ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΠΟΔΟΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ
ΠΡΟΣΑΡΜΟΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ
ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΚΤΥΑΚΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ
ΕΦΑΡΜΟΓΩΝ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

ΕΥΑΓΓΕΛΟΣ Δ. ΣΑΚΚΟΠΟΥΛΟΣ

ΕΠΙΒΛΕΠΩΝ: ΚΑΘΗΓΗΤΗΣ Κ. ΑΘΑΝΑΣΙΟΣ ΤΣΑΚΑΛΙΔΗΣ

Δεκέμβριος 2005

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΠΟΔΟΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ
ΠΡΟΣΑΡΜΟΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ
ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΚΤΥΑΚΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ
ΕΦΑΡΜΟΓΩΝ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

ΕΥΑΓΓΕΛΟΣ Δ. ΣΑΚΚΟΠΟΥΛΟΣ

ΚΑΘΗΓΗΤΗΣ Κ. ΑΘΑΝΑΣΙΟΣ ΤΣΑΚΑΛΙΔΗΣ (ΕΠΙΒΛΕΠΩΝ)

Τριμελής Συμβουλευτική Επιτροπή (αλφαβητικά):

ΑΝ. ΚΑΘΗΓΗΤΗΣ Κ. ΙΩΑΝΝΗΣ ΓΑΡΟΦΑΛΑΚΗΣ (ΜΕΛΟΣ ΤΡΙΜΕΛΟΥΣ)

ΚΑΘΗΓΗΤΗΣ Κ. ΑΘΑΝΑΣΙΟΣ ΤΣΑΚΑΛΙΔΗΣ (ΕΠΙΒΛΕΠΩΝ)

ΕΠΙΚ. ΚΑΘΗΓΗΤΗΣ Κ. ΙΩΑΝΝΗΣ ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ (ΜΕΛΟΣ ΤΡΙΜΕΛΟΥΣ)

Επταμελής Εξεταστική Επιτροπή (αλφαβητικά):

ΑΝ. ΚΑΘΗΓΗΤΗΣ Κ. ΙΩΑΝΝΗΣ ΓΑΡΟΦΑΛΑΚΗΣ (ΜΕΛΟΣ ΤΡΙΜΕΛΟΥΣ)

ΕΠΙΚ. ΚΑΘΗΓΗΤΗΣ Κ. ΧΡΗΣΤΟΣ ΜΑΚΡΗΣ

ΚΑΘΗΓΗΤΗΣ Κ. ΣΠΥΡΙΔΩΝ ΛΥΚΟΘΑΝΑΣΗΣ

ΚΑΘΗΓΗΤΗΣ Κ. ΓΕΩΡΓΙΟΣ ΠΑΥΛΙΔΗΣ

ΚΑΘΗΓΗΤΗΣ Κ. ΠΑΝΑΓΙΩΤΗΣ ΠΙΝΤΕΛΑΣ

ΚΑΘΗΓΗΤΗΣ Κ. ΑΘΑΝΑΣΙΟΣ ΤΣΑΚΑΛΙΔΗΣ (ΕΠΙΒΛΕΠΩΝ)

ΕΠΙΚ. ΚΑΘΗΓΗΤΗΣ Κ. ΙΩΑΝΝΗΣ ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ (ΜΕΛΟΣ ΤΡΙΜΕΛΟΥΣ)

Δεκέμβριος 2005

**UNIVERSITY OF PATRAS
SCHOOL OF ENGINEERING
COMPUTER ENGINEERING AND INFORMATICS DEPARTMENT**

PHD DISSERTATION

**EFFICIENT ALGORITHMS AND ADAPTIVE
TECHNIQUES FOR NET-CENTRIC
INFORMATION SYSTEMS AND WEB
APPLICATIONS MANAGEMENT**

EVANGELOS D. SAKKOPOULOS

ACADEMIC SUPERVISOR: PROF. ATHANASIOS K. TSAKALIDIS

DECEMBER 2005

«Όπου υπάρχει θέληση, υπάρχει ο τρόπος»
in Moby Dick by Herman Melville

ΠΡΟΛΟΓΟΣ

Πριν παρουσιάσω τα αποτελέσματα της διδακτορικής διατριβής μου, αισθάνομαι την υποχρέωση να αναφερθώ σε ορισμένους από τους ανθρώπους (συναδέλφους, συνεργάτες και φίλους) που γνώρισα και συνεργάστηκα μαζί τους κατά τη διάρκεια των χρόνων εκπόνησής της.

Αρχικά θα ήθελα να σημειώσω ότι θεωρώ τιμή μου τη συνεργασία με τον Καθηγητή κ. Αθανάσιο Τσακαλίδη και υποχρέωσή μου να τον ευχαριστήσω θερμά. Στάθηκε δίπλα μου ως γνήσιος Πανεπιστημιακός Δάσκαλος και μου έδωσε την ευκαιρία να μάθω τι σημαίνει έρευνα και πόσο σημαντικό είναι να θέτεις και να κατακτάς συνεχώς υψηλότερους στόχους. Τον ευχαριστώ που με στήριξε στα πρώτα μου επιστημονικά βήματα και που με τίμησε δείχνοντας μου πολλές φορές την εμπιστοσύνη του. Ελπίζω να στάθην αξίος των προσδοκιών που είχε για μένα και ελπίζω να αισθάνεται περήφανος για το μαθητή του. Τον διαβεβαιώνω ότι θα προσπαθώ πάντα να τον κάνω ακόμη περισσότερο περήφανο και θα ήθελα να ξέρει ότι πάντα θα τον σέβομαι και θα τον υπολογίζω στα επόμενα βήματά μου.

Ευχαριστώ επίσης θερμά τα άλλα δύο μέλη τις τριμελούς συμβουλευτικής επιτροπής της διδακτορικής αυτής διατριβής τον Αναπλ. Καθηγητή κ. Γιάννη Γαροφαλάκη και τον Επιμ. Καθηγητή κ. Γιάννη Χατζηλυγερούδη για τη συνεργασία που είχαμε και το ενδιαφέρον που έδειξαν στην ερευνητική προσπάθειά μου.

Ευχαριστώ ακόμη τον Καθηγητή κ. Σπυρίδων Λυκοθανάση, τον Καθηγητή κ. Γεώργιο Παυλίδη, τον Καθηγητή κ. Παναγιώτη Πιντέλα για τη πρόθυμη συμμετοχή τους ως μέλη της εξεταστικής επιτροπής.

Θα ήθελα ιδιαίτερα να ευχαριστήσω τον Επιμ. Καθηγητή Χρήστο Μακρή που καθ' όλη τη διάρκεια της προσπάθειάς μου στάθηκε δίπλα μου ως φίλος και αναντικατάστατος ερευνητικός συνεργάτης. Θα ήθελα ακόμη να ευχαριστήσω το Δρ. Σπύρο Σιούτα για την εποικοδομητική μας συνεργασία και τη συμπαράστασή του.

Επίσης ευχαριστώ τον Επιμ. Καθηγητή Σπύρο Συρμακέση και το Δρ. Γιάννη Τζήμα για την πολύτιμη συνεργασία, βοήθεια και συμπαράστασή τους τόσο στις ερευνητικές όσο και τις επαγγελματικές μου προσπάθειες.

Θα ήθελα ακόμη να ευχαριστήσω όλους τους συναδέλφους, συνεργάτες και φίλους που βρεθήκαμε και ζήσαμε μαζί στο ΤΜΗΥΠ και την Ερευνητική Μονάδα 5 του ΕΑΙΤΥ όλα αυτά τα χρόνια. Υπήρξατε όλοι άξιοι συμπαράστατες στον αγώνα μου με τον τρόπο σας και θα είστε πάντα στη σκέψη μου.

Ένα μεγάλο ευχαριστώ στο φίλο Γιάννη Παναγή με τον οποίο συμπορευτήκαμε και περάσαμε μαζί αξέχαστες στιγμές σε όλη τη διάρκεια των χρόνων εκπόνησης της διατριβής αυτής.

Ξεχωριστή θέση έχει η σύντροφός μου Πούλια και την ευχαριστώ, που επέμεινε υπομονετικά να συμπαραστέκεται σε όλες μου τις προσπάθειες.

Άφησα για το τέλος τους γονείς μου Δημήτριο και Ασημούλα και την αδερφή μου Ρούλη, αφανείς ήρωες σε κάθε μου προσπάθεια. Τα λόγια δεν είναι αρκετά για να τους εκφράσω την ευγνωμοσύνη μου. Τους αφιερώνω το παρόν πόνημα ως ελάχιστο δείγμα της αγάπης μου.

ΕΠΙΤΕΛΙΚΗ ΣΥΝΟΨΗ

Στα πλαίσια της διδακτορικής μας διατριβής ασχοληθήκαμε με προβλήματα διαχείρισης δικτυακών πληροφοριακών συστημάτων που βασίζονται σε τεχνολογίες παγκόσμιου ιστού (network-centric information systems, netcentric information systems, web information systems). Η έννοια της δικτυο-κεντρικής προσέγγισης (netcentric) προσπαθεί να αποδώσει την τάση να χρησιμοποιείται η δικτυακή υποδομή και τεχνολογία όλο και περισσότερο στα πληροφοριακά συστήματα και τις εφαρμογές παγκόσμιου ιστού για να παρέχουν, να δημοσιοποιούν, να διαμοιράζουν και να επικοινωνούν online υπηρεσίες και πληροφορίες.

Κύριος στόχος της διατριβής είναι α) η διασφάλιση της ποιότητας κατά την εξυπηρέτηση, β) η μείωση του χρόνου εντοπισμού και γ) η εξατομίκευση υπηρεσιών και πληροφοριών σε δικτυακά πληροφοριακά περιβάλλοντα και εφαρμογές που βασίζονται σε τεχνολογίες μηχανικής Παγκόσμιου Ιστού.

Σε πρώτο επίπεδο, οι αποδοτικοί αλγόριθμοι που αναπτύξαμε αφορούν τις υπηρεσίες Web Services¹ που έχουν σχεδιαστεί να υποστηρίζουν διαλειτουργική αλληλεπίδραση μεταξύ μηχανών με χρήση δικτυακής υποδομής. Πρόκειται ένα τεχνολογικό πλαίσιο το οποίο προτυποποιήθηκε από το W3 Consortium (<http://www.w3.org>) και γνωρίζει την ευρεία υποστήριξη τόσο της επιστημονικής κοινότητας τεχνολογιών πληροφορικής και επικοινωνιών όσο και των επαγγελματιών μηχανικών Η/Υ και της βιομηχανίας πληροφορικής παγκοσμίως.

Αναλυτικότερα στο πρώτο μέρος της διατριβής δίνουμε αρχικά μία νέα κατηγοριοποίηση και συγκριτική παρουσίαση των λύσεων και προβλημάτων που αφορούν αποδοτικές λύσεις αλγορίθμων διαχείρισης και αναζήτησης υπηρεσιών. Στη συνέχεια, εισάγουμε μια σειρά από νέους αποδοτικούς αλγορίθμους διαχείρισης και αναζήτησης υπηρεσιών που διασφαλίζουν την ποιότητα της παρεχόμενης υπηρεσίας και βελτιώνουν την πολυπλοκότητα στο χρόνο εντοπισμού μιας υπηρεσίας. Συνολικά στο πρώτο μέρος παρουσιάζουμε:

- Αποδοτικούς αλγορίθμους δυναμικής επιλογής Web Service που λαμβάνουν υπόψη μη λειτουργικές προδιαγραφές για ποιότητα και απόδοση κατά την προσπάθεια χρήσης (consumption) του Web Service (QoWS enabled WS discovery).
- Αποδοτικούς αλγορίθμους διαχείρισης και αναζήτησης υπηρεσιών δικτυο-κεντρικών πληροφοριακών συστημάτων οι οποίοι βασίζονται σε αποκεντρωποιημένες δικτυακές λύσεις ειδικά σχεδιασμένες για WS καταλογογράφηση (decentralized WS discovery).

¹ Από το λεξικό όρων (<http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>).

Σε δεύτερο επίπεδο, δίνουμε αποδοτικές προσαρμοστικές μεθόδους για την εξατομίκευση των αποτελεσμάτων αναζήτησης πληροφοριών στον Παγκόσμιο Ιστό. Με τον τρόπο αυτό επιτυγχάνουμε βελτίωση της απόδοσης τόσο για τις εσωτερικές λειτουργίες διαχείρισης και αναζήτησης των δικτυακών πληροφοριακών συστημάτων όσο και του τελικού αποτελέσματος, της πληροφoρίας δηλαδή, που παρουσιάζουν τα συστήματα αυτά στον τελικό χρήστη.

Συγκεκριμένα, στο *δεύτερο μέρος της διατριβής* εισάγουμε μια σειρά από τρεις αλγορίθμους εξατομίκευση των αποτελεσμάτων αναζήτησης, οι οποίοι βασίζονται σε τεχνικές μετρίων συνδέσμων (link metrics). Το κύριο πλεονέκτημα των τεχνικών που προτείνουμε είναι ότι επιτρέπουν, με τη χρήση μιας αρκετά απλής μεθοδολογίας, την εξατομίκευση των αποτελεσμάτων αναζήτησης, χωρίς να επιβαρύνονται οι χρήστες σε όγκο αποθήκευσης ή με καθυστερήσεις λόγω χρόνου εκτέλεσής τους. Επιτυγχάνουμε εξατομικευμένη αναζήτηση εφαρμόζοντας τεχνικές ανάλυσης και επεξεργασίας συνδέσμων *όχι* στο γράφο ιστού αλλά για πρώτη φορά σε αρκετά *μικρότερους* εξατομικευμένους γράφους που σχηματίζονται από διαθέσιμες σημασιολογικές ταξονομίες. Συνοψίζοντας τα ερευνητικά αποτελέσματα του δεύτερου μέρους παρουσιάζουμε τα ακόλουθα:

- Αποδοτικοί αλγόριθμοι για εξατομικευμένη αναζήτηση πληροφορίας (personalized searching) στον Παγκόσμιο Ιστό.
- Μηχανισμός προσαρμοστικής παρουσίασης αποτελεσμάτων αναζήτησης με χρήση πολλαπλών επιπέδων κατηγοριοποίησης.
- Επέκταση των αλγορίθμων για μηχανισμούς στοχευμένης συλλογής σελίδων (focused web crawlers) που αποτελούν εναλλακτική της εξατομικευμένης αναζήτησης πληροφοριών.

Τέλος στο *τρίτο και τελευταίο μέρος της διατριβής* παρουσιάζουμε μια σειρά από εφαρμογές, αρχιτεκτονικές και λειτουργικά πλαίσια τα οποία αφορούν δικτυακά πληροφοριακά περιβάλλοντα στα οποία εφαρμόζουμε τεχνικές διαχείρισης υπηρεσιών και μηχανισμούς εξατομίκευσης πληροφοριών. Ο κύριος στόχος της παρουσίασης των λύσεων αυτών είναι να επιδειχθεί ότι οι προτεινόμενοι αποδοτικοί αλγόριθμοι, που παρουσιάστηκαν στα προηγούμενα κεφάλαια, έχουν εφαρμογή σε πολλαπλά προβλήματα διαφορετικών επιστημονικών και τεχνολογικών πεδίων που χρησιμοποιούν δικτυακά πληροφοριακά συστήματα και εφαρμογές παγκόσμιου ιστού.

EXECUTIVE SUMMARY

In our PhD dissertation we dealt with performance issues in network - centric information systems, netcentric information systems and web information systems. Netcentric approach attempts to depict the augmenting tendency to use the network communication in information systems and web applications in order to provide, to publish, to distribute and to communicate online services and information.

The key aim of our doctoral thesis is a) the quality at the service provision, v) the reduction of discovery time and c) the personalization of services and information in network information systems and applications that are based on web engineering technologies.

Initially, we studied, designed and implemented efficient algorithms concerning Web Services technologies² that have been designed to facilitate interoperable service integration using network infrastructure. Web Services Architecture has been standardized by W3 Consortium (<http://www.w3.org>) as the technological framework and it has received the wide support of the information technology scientific community as well as the information technology (IT) professionals and industry worldwide.

In the first section we introduce a new categorization and comparative presentation of the available algorithmic solutions for service management and discovery. Then, we introduce a series of new efficient algorithms that ensure quality of service provision and improve time complexity in service discovery. Overall in the first part of the thesis we present:

- Efficient algorithms for dynamic Web Service selection taking into account non-functional specifications (Quality of Web Service – QoWS) and performance issues during Web Service (WS) consumption attempt (i.e. QoWS enabled WS discovery).
- Efficient algorithms for service management and discovery in network centric information systems that are based on decentralized network approaches specifically designed for WS discovery.

In the sequel, we propose efficient adaptive methods for personalized web searching. In this way we provide performance improvement both for the internal management and discovery functionality of web based net-centric

² From the W3C dictionary of terms (<http://www.w3.org/TR/2004/SOUP-20040211/#webservice>).

information systems as well as for the systems' output that is the end-user information.

In particular, *in the second section*, we introduce a series of three new algorithms for personalized searching. The proposed algorithms are mainly based on link metrics techniques. Their main advantage is that they allow, with the use of a simple methodology, search results personalization, with minimum overhead in terms of storage volume and computation time. We achieve personalized search using link analysis in a personalized graph much *smaller* one than the whole web graph. The personalized graph is shaped taking advantage of semantic taxonomies. Summarizing the novel research results of this second section are the following:

- Efficient algorithms for personalized web information searching.
- Adaptive presentation mechanisms of search results with the use of multiple levels of novel categorization.
- Extension that allows the adoption of the algorithms for the case of focused web crawling mechanisms, which constitute an alternative personalized searching approach.

Finally in the *third and last section of our thesis*, we present a series of applications, architectures and frameworks of different web based net-centric information environments cases, in which we apply our techniques for service management and personalized information discovery. The main objective of this presentation is to show that the efficient algorithms presented in the previous sections, have multiple potentials of application in problems of different research and technological areas using web based net-centric informative systems and web applications. Cases presented include network management information systems, e-learning approaches, semantic mining and multimedia retrieval systems, web content and structure maintenance solutions and agricultural information systems.

ΠΕΡΙΕΧΟΜΕΝΑ

<i>Αριθμός/ Τίτλος Κεφαλαίου</i>	<i>Σελίδα</i>
Πρόλογος.....	<i>ix</i>
Επιτελική Σύνοψη.....	<i>xi</i>
Executive Summary.....	<i>xiii</i>
Περιεχόμενα	<i>xv</i>
Λίστα Σχημάτων.....	<i>xix</i>
Λίστα Πινάκων	<i>xxi</i>
Λίστα Αλγορίθμων	<i>xxii</i>
Δημοσιευμένες εργασίες που υποστηρίζουν τη διατριβή.....	<i>xxiii</i>
Διεθνή Επιστημονικά Περιοδικά με κρίση	<i>xxiii</i>
Κεφάλαια σε Διεθνή Επιστημονικά Βιβλία με κρίση	<i>xxiii</i>
Διεθνή Επιστημονικά Συνέδρια με κρίση	<i>xxiv</i>
0 ΕΙΣΑΓΩΓΗ	<i>27</i>
0.1 Εισαγωγικά Θέματα και Διάρθρωση της παρούσας Διατριβής.....	<i>27</i>
Μέρος Α': Αποδοτικοί Αλγόριθμοι Διαχείρισης και Αναζήτησης σε Δικτυακά Πληροφοριακά Συστήματα με Web Services	<i>39</i>
1 Τεχνολογίες μηχανικής παγκόσμιου ιστού (Web Engineering) για αναζήτηση υπηρεσιών (Web Services) σε δικτυακά πληροφοριακά Συστήματα (Net-Centric Information Systems)	<i>39</i>
1.1 Εισαγωγή.....	<i>39</i>
1.2 Προαπαιτούμενα	<i>40</i>
1.3 Τεχνολογίες αναζήτησης υπηρεσιών Web Services.....	<i>43</i>
1.4 Κύριοι Παράγοντες – Παίχτες στο Παιχνίδι της Ανακάλυψης	<i>46</i>
1.4.1 Λύσεις Προ-Web Service	<i>46</i>
1.4.2 Οι βασικοί φορείς στις διαδικασίες ανακάλυψης.....	<i>47</i>
1.5 Αρχιτεκτονική: Όψεις και Προσεγγίσεις	<i>51</i>
1.5.1 Επίπεδα Αυτοματοποίησης – Χειροκίνητες και Έξυπνες Λύσεις	<i>51</i>
1.5.2 Κεντροποιημένες (Centralized) και Αποκεντροποιημένες (decentralized) Λύσεις	<i>51</i>
1.5.2.1 Κεντροποιημένες (Centralized) Υπηρεσίες	<i>51</i>
1.5.2.2 Αποκεντροποιημένες Λύσεις.....	<i>51</i>
1.5.3 Ακολουθώντας τα Πρότυπα και Αναθεωρώντας τα.....	<i>51</i>
1.5.3.1 Ακολουθώντας την Προδιαγραφή UDDI	<i>51</i>
1.5.3.2 Προσεγγίσεις που Αναθεωρούν τα Πρότυπα	<i>51</i>
1.5.4 Τεχνολογικά Περιβάλλοντα	<i>51</i>

1.6	Μοντέλα Δεδομένων.....	51
1.6.1	Μοντελοποίηση με Τεχνικές Ανάκτησης Πληροφορίας.....	51
1.6.2	Σημασιολογική Προσέγγιση	51
1.6.3	Προσεγγιστική Ανάκτηση.....	51
1.7	Θέματα Διασφάλισης Ποιότητας.....	51
1.7.1	Παράμετροι QoWS.....	51
1.7.2	Ανακάλυψη WS με Απαίτηση Ποιοτικών Χαρακτηριστικών	51
1.7.3	Παράδοση QoWS	51
1.8	Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις.....	51
Βιβλιογραφία Κεφαλαίου 1.....		51
2	Δυναμική διαχείριση και ανακάλυψη Web Services με βάση λειτουργικά και ποιοτικά χαρακτηριστικά.....	51
2.1	Εισαγωγή.....	51
2.2	Περιγραφή του Προβλήματος	51
2.3	Σχετικές εργασίες	51
2.3.1	Απομακρυσμένη κλήση μεθόδων.....	51
2.3.2	Τεχνικές Ανακάλυψης Web Service.....	51
2.4	Οδικός Χάρτης της Προσέγγισής μας	51
2.5	Πρώτο βήμα επιλογής με χρήση υπολογιστικής γεωμετρίας	51
2.6	Προσαρμοστική επιλογή.....	51
2.6.1	Προπαρασκευαστική δράση: Διατήρηση του ιστορικού προφίλ των WS.....	51
2.6.2	Online δράσεις: Αποδοτική επιλογή.....	51
2.7	Εκτίμηση πολυπλοκότητας λόγω χρήσης της επιλογής contour.....	51
2.8	Πειραματικά αποτελέσματα	51
2.8.1	Μεθοδολογία του πειράματος.....	51
2.8.2	Αξιολόγηση των μετρήσεων.....	51
2.8.3	Αξιολόγησης της επίδρασης του αλγορίθμου contour	51
2.9	Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις.....	51
Βιβλιογραφία Κεφαλαίου 2		51
3	Αναζήτηση Υπηρεσιών Ιστού με Δικτυοκεντρικές προσεγγίσεις	51
3.1	Εισαγωγή.....	51
3.2	Περιγραφή του Προβλήματος	51
3.3	Παρουσίαση της προσέγγισης του NIPPERS.....	51
3.4	Προαπαιτούμενα: Το Στατικό Interpolation Search Tree	51
3.5	Η δομή δεδομένων NIPPERS.....	51
3.5.1	Οι κόμβοι στο δίκτυο NIPPERS	51
3.5.2	Ανάλυση του αλγορίθμου αναζήτησης	51
3.5.3	Εισαγωγή/Διαγραφή των Στοιχείων/Κόμβων στο NIPPERS.....	51
3.6	Ανεκτικότητα σε σφάλματα του NIPPERS	51
3.7	Εξασφάλιση Ποιότητας στην Ανακάλυψη Web Service	51

3.8	Αξιολόγηση και Περιγραφή της Συνεισφοράς του NIPPERS	51
3.9	Παρουσίαση της προσέγγισης του BDT-GRID	51
3.10	Προαπαιτούμενα: Ιεραρχικοί και βασισμένοι σε δέντρα αλγόριθμοι.....	51
3.10.1	Ιεραρχικά Πρωτόκολλα.....	51
3.10.2	Ένα παράδειγμα μιας DNS αναζήτησης.....	51
3.10.3	Λανθάνουσα μνήμη στο DNS	51
3.10.4	Πλεονασμός και ανεκτικότητα σε σφάλματα στο DNS	51
3.11	Η αρχιτεκτονική BDT-GRID	51
3.11.1	Το BDT-Grid δίκτυο.....	51
3.11.2	Κόμβοι στο BDT-Grid	51
3.11.3	Πολυπλοκότητα Αναζήτησης BDT-GRID.....	51
3.11.4	Ανεκτικότητα σε σφάλματα του BDT-GRID	51
3.11.5	Συμμετοχή και Αποχώρηση στο BDT-GRID	51
3.12	Αξιολόγηση και Περιγραφή της Συνεισφοράς του BDT-GRID	51
3.13	Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις	51
	<i>Βιβλιογραφία Κεφαλαίου 3.....</i>	<i>51</i>
	Μέρος Β': Αποδοτικοί Αλγόριθμοι για Εξατομικευμένη Αναζήτηση	
	Πληροφοριών σε Δικτυακά Πληροφοριακά Περιβάλλοντα	51
4	<i>Αποδοτικοί Αλγόριθμοι Εξατομικευμένης Αναζήτησης στο Web.....</i>	<i>51</i>
4.1	Εισαγωγή.....	51
4.2	Περιγραφή του Προβλήματος	51
4.3	Σχετικές εργασίες	51
4.4	Αλγόριθμοι εξατομίκευσης αναζήτησης	51
4.5	Προαπαιτούμενα	51
4.6	Αλγόριθμος Off-Line	51
4.7	Αλγόριθμος OnLine	51
4.8	Σημασιολογική συσχέτιση με Κόμβους Εικονικών Κατηγοριών	51
4.9	Μεθοδολογία υλοποίησης εξατομίκευσης κατηγοριών.....	51
4.9.1	Αρχικοποίηση	51
4.9.2	Υπολογισμός της Σπουδαιότητας Κατηγορίας	51
4.10	Οδικός χάρτης πειραματικής διαδικασίας και αξιολόγησης	51
4.11	Μεθοδολογία πειραμάτων.....	51
4.11.1	Φάση Αρχικοποίησης.....	51
4.11.2	Φάση Ερωτήματος.....	51
4.12	Αξιολόγηση των αποτελεσμάτων	51
4.12.1	Γράφοι Ακρίβειας-Ανάκλησης.....	51
4.12.2	Η επίδραση της εισαγωγής εικονικών κόμβων.....	51
4.13	Μια επέκταση: στοχευμένη συλλογή σελίδων	51
4.14	Προσαρμοστική αναζήτηση με χρήση SVM κατηγοριοποιητών.....	51
4.14.1	Προφίλ πολλαπλών επιπέδων	51
4.14.2	Μηχανισμός προσαρμοστικής αναζήτησης	51

4.15	Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις.....	51
<i>Βιβλιογραφία Κεφαλαίου 4</i>		<i>51</i>
Μέρος Γ': Αποδοτική εφαρμογή προτεινόμενων αλγορίθμων σε Δικτυακά		
Πληροφοριακά Συστήματα		51
5	<i>Μελέτη και παρουσίαση περιπτώσεων εφαρμογής.....</i>	<i>51</i>
5.1	Εισαγωγή - Μελέτη και παρουσίαση περιπτώσεων εφαρμογής	51
5.2	Διαχείριση ασύρματων και ενσύρματων δικτυακών πόρων και συσκευών	51
5.3	Ηλεκτρονική μάθηση και εκπαίδευση (e-learning)	51
5.4	Σημασιολογική εξόρυξη και διαχείριση πολυμεσικών δεδομένων	51
5.5	Πλαίσιο επανοργάνωσης δομής και εξατομίκευσης περιεχομένου εφαρμογών ιστού.....	51
5.6	Δικτυακά πληροφοριακά συστήματα γεωπονίας.....	51
5.7	Συνεισφορά και συμπεράσματα.....	51
<i>Βιβλιογραφία Κεφαλαίου 5</i>		<i>51</i>
Ανακεφαλαίωση Συμπερασμάτων, Συνεισφοράς και Ανοικτών Προβλημάτων..		
Συμπεράσματα και συνεισφορά της διδακτορικής διατριβής.....		51
Ανοικτά προβλήματα - Μελλοντικές κατευθύνσεις έρευνας		51
<i>Ευρετήριο.....</i>		<i>51</i>
<i>Γλωσσάρι</i>		<i>51</i>

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Αριθμός/ Τίτλος

Σελίδα

Σχήμα 1: Το μοντέλο των Web Services	42
Σχήμα 2: Κατηγοριοποίηση (α) των προσεγγίσεων της ανακάλυψης Web Service	50
Σχήμα 3: Κατηγοριοποίηση (β) των προσεγγίσεων της ανακάλυψης Web Service	50
Σχήμα 4: Κατηγοριοποίηση με βάση την αρχιτεκτονική προσέγγιση.....	51
Σχήμα 5: Ταξινόμηση με βάση την Επεκτασιμότητα και την Ευελιξία Εντοπισμού	51
Σχήμα 6: Η αρχιτεκτονική QoWS broker	51
Σχήμα 7: Γενική άποψη του προβλήματος επιλογής Web Service	51
Σχήμα 8: Οδικός χάρτης αλγορίθμου δυναμικής επιλογής WS	51
Σχήμα 9: Το πρώτο (1 ^ο) βήμα δυναμικής επιλογής WS με QoS	51
Σχήμα 10: Παράδειγμα ενημέρωσης contour	51
Σχήμα 11: Προσαρμοστική επιλογή (Offline βήμα) της δυναμικής επιλογής WS με QoS.....	51
Σχήμα 12: Προσαρμοστική επιλογή (Online βήματα) της δυναμικής επιλογής WS με QoS	51
Σχήμα 13: Πειραματικά αποτελέσματα με δύο σύνολα WS σύντομα σε διάρκεια εκτέλεσης και δικτυακής καθυστέρησης.....	51
Σχήμα 14: Πειραματικά αποτελέσματα με δύο σύνολα WS μεγάλης διάρκειας εκτέλεσης και δικτυακής καθυστέρησης.....	51
Σχήμα 15: Συγκριτική αξιολόγηση της επίδρασης της επιλογής contour στα σύνολα WS 1 και 2	51
Σχήμα 16: Το κέρδος του χρόνου απόκρισης της WS στα πειραματικά αποτελέσματα	51
Σχήμα 17: Το σύστημα NIPPERS από την εργασία μας [17]	51
Σχήμα 18: Διαχωρισμός Δικτυακού Γράφου.....	51
Σχήμα 19: Η αρχιτεκτονική NIPPERS ως κατανεμημένο εκθετικό δέντρο αναζήτησης (Distributed Exponential Search Tree).....	51
Σχήμα 20: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Weibull (smooth) κατανομή	51
Σχήμα 21: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Beta (smooth) κατανομή	51
Σχήμα 22: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Normal (smooth) κατανομή.....	51
Σχήμα 23: Το σύστημα BDT-Grid από την εργασία μας [10].....	51
Σχήμα 24: Σύγκριση απόδοσης αναζήτησης με BDT-Chord.....	51
Σχήμα 25: Παράδειγμα εξατομίκευσης με χρήση εικονικών κόμβων κατηγοριών.	51
Σχήμα 26: Εισαγωγή εικονικών κόμβων.	51
Σχήμα 27: Αποτελέσματα P-R των Αρχικών (Κατάταξη Google) και Τελικών (Εξατομικευμένη Κατάταξη) Αποτελεσμάτων.	51
Σχήμα 28: Αποτελέσματα P-R των Αρχικών (Κατάταξη Google) και Τελικών (Εξατομικευμένη Κατάταξη) Αποτελεσμάτων.	51

Σχήμα 29: Η αύξηση του επιπέδου ακρίβειας αυξάνεται ιδιαίτερα στο 20% της Ανάκλησης	51
Σχήμα 30: Το απόλυτο κέρδος στην ακρίβεια για πολιτικές εισαγωγή Ενός και Πολλαπλών Εικονικών κόμβων στον semi-offline αλγόριθμο.....	51
Σχήμα 31: Το σχετικό κέρδος στην ακρίβεια για πολιτικές εισαγωγή Ενός και Πολλαπλών Εικονικών κόμβων στον semi-offline αλγόριθμο.....	51
Σχήμα 32: Πολυεπίπεδο πλαίσιο διαχείρισης προφίλ	51
Σχήμα 33: Αποτελέσματα κατηγοριοποίησης με βάση πλήρες κείμενο και σύνοψη	51

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

<i>Αριθμός/ Τίτλος</i>	<i>Σελίδα</i>
Πίνακας 1: Συγκριτική Παρουσίαση Αποτελεσμάτων NIPPERS, BDT-GRID και Chord	30
Πίνακας 2: Τύποι του Καταλόγου UDDI.....	48
Πίνακας 3: Οι παράμετροι QoS και οι αντίστοιχες μονάδες μέτρησης.	51
Πίνακας 4: Περιπτώσεις/Προβλήματα και τεχνικές ανακάλυψης που τα αντιμετωπίζουν.	51
Πίνακας 5: Ένα παράδειγμα σχήματος εκτέλεσης WS	51
Πίνακας 6: Πειραματικά αποτελέσματα ομαδοποιημένα σε σύνολα	51
Πίνακας 7: Σύγκριση απόδοσης NIPPERS.....	51
Πίνακας 8: Παράδειγμα απάντησης σε ένα ερώτημα DNS.....	51
Πίνακας 9: Σύγκριση BDT-GRID με την πλέον διαδεδομένη αποδοτική αρχιτεκτονική	51
Πίνακας 10: Συνολικός συγκριτικός πίνακας συνεισφοράς δικτυοκεντρικών λύσεων NIPPERS και BDT-grid με γνωστές τεχνικές.	51
Πίνακας 11: Συγκριτική Παρουσίαση Αποτελεσμάτων NIPPERS, BDT-GRID και Chord	51

ΛΙΣΤΑ ΑΛΓΟΡΙΘΜΩΝ

<i>Αριθμός/ Τίτλος</i>	<i>Σελίδα</i>
Αλγόριθμος 1: Ο αλγόριθμος της επιλογής contour.....	51
Αλγόριθμος 2: Αλγόριθμος online επιλογής.....	51
Αλγόριθμος 3: Αλγόριθμος online επιλογής με χρήση multicast μηχανισμού	51
Αλγόριθμος 4: Ψευδο-κώδικας για τον αλγόριθμο αναζήτησης NIPPERS	51
Αλγόριθμος 5: Ψευδο-κώδικας για τον αλγόριθμο εισαγωγής NIPPERS	51
Αλγόριθμος 6: Ψευδο-κώδικας για τον αλγόριθμο διαγραφής NIPPERS.....	51
Αλγόριθμος 7: Ψευδο-κώδικας για τη λειτουργία ανακατασκευής NIPPERS.....	51
Αλγόριθμος 8: Ψευδο-κώδικας για BDT-Grid αναζητήσεις	51
Αλγόριθμος 9: Ψευδο- κώδικας για εισαγωγή στο BDT-Grid	51
Αλγόριθμος 10: Ψευδο- κώδικας για διαγραφή στο BDT-Grid.....	51
Αλγόριθμος 11: Ψευδο-κώδικας για τη λειτουργία ανακατασκευής BDT-Grid.....	51
Αλγόριθμος 12: Αλγόριθμος Experiment_i	51
Αλγόριθμος 13: Αλγόριθμος Offline_Categories.....	51
Αλγόριθμος 14: Αλγόριθμος Update_Weights	51
Αλγόριθμος 15: Αλγόριθμος Online_Categories	51
Αλγόριθμος 16 : Αλγόριθμος Semi_Offline_Categories	51
Αλγόριθμος 17: Προ-επεξεργασία στοχευμένης συλλογής σελίδων	51
Αλγόριθμος 18: Ανάθεση βάρους για ταξινομητή στοχευμένης συλλογής σελίδων	51

ΔΗΜΟΣΙΕΥΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

ΠΟΥ ΥΠΟΣΤΗΡΙΖΟΥΝ ΤΗ ΔΙΑΤΡΙΒΗ

Στη συνέχεια δίνουμε τη λίστα με τις δημοσιευμένες εργασίες μας που υποστηρίζουν την παρούσα διδακτορική διατριβή συνολικά. Ακολουθούν στην εισαγωγή περισσότερες λεπτομέρειες για τις εργασίες και τον τρόπο που αυτές εντάσσονται στη διάρθρωση της διατριβής.

Σύμφωνα με την πάγια τακτική των μελών ΔΕΠ του Τμήματος Μηχανικών Η/Υ & Πληροφορικής του Πανεπιστημίου Πατρών, οι ερευνητές αναφέρονται αλφαβητικά στις επιστημονικές εργασίες.

Διεθνή Επιστημονικά Περιοδικά με κρίση

- [I11]. Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Efficient and adaptive discovery techniques of web services handling large data sets”, in J. Systems and Software (JSS), Elsevier Science, corrected proof, in press.
- [I12]. Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Category ranking for personalized search”, in the J. of Data and Knowledge Engineering (DKE), Elsevier Science, to appear.
- [I13]. Dimitrios Kanellopoulos, Evangelos Sakkopoulos and Athanasios Tsakalidis (2005), “Semantic Mining and Web Service discovery techniques for media resources management”, in the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol. 1 (1), ISSN (Online): 1744-263X - ISSN (Print): 1744-2621, Inderscience Publishers, corrected proof, in press.
- [I14]. John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis (2006), “Optimization mechanism for Web Search Results using Topic Knowledge”, in the International Journal of Knowledge and Learning (IJKL), Vol. 2 (1&2), ISSN (Online): 1741-1017 - ISSN (Print): 1741-1009, Inderscience Publishers, corrected proof, in press.

Κεφάλαια σε Διεθνή Επιστημονικά Βιβλία με κρίση

- [I15]. Yannis Panagis, Evangelos Sakkopoulos, Spiros Sioutas and Athanasios Tsakalidis, “Web Service Integration and Management Strategies for Large Scale Datasets”, book chapter in ed. Prof. Zongmin Ma, Database Modelling for Industrial Data Management: Emerging Technologies and Applications, Idea Group Inc., 2006, pp. 217-243.

Διεθνή Επιστημονικά Συνέδρια με κρίση

- [Σ1].Christos Makris, Evangelos Sakkopoulos, Spiros Sioutas, Peter Triantafillou and Athanasios Tsakalidis, Bill Vassiliadis, "NIPPERS: Network of InterPolated PeERS for Web Service Discovery", in the proceedings of the 2005 IEEE International Conference on Information Technology: Coding & Computing, Track Next Generation Web and Grid Systems (IEEE ITCC 2005), 2005 in Las Vegas, USA, Vol. 2, pp. 193-198.
- [Σ2].John Garofalakis, Evangelos Sakkopoulos, Spiros Sirmakessis, Athanasios Tsakalidis "Integrating Adaptive Techniques into Virtual University Learning Environment", IEEE International Conference on Advanced Learning Technologies (IEEE ICALT), September 9- 12, 2002, Kazan Tatarstan, Russia, pp. 28 -33.
- [Σ3].Eythimios Dimopoulos, Athanasios Panousis, Evangelos Sakkopoulos and Athanasios Tsakalidis, "WiFiAdmin: An Intelligent Web-based Management Environment for Wireless Networks", in the IEEE International Conference on Information Technology and Applications 2005 (IEEE ICITA), Sydney 4-7/7/2005, pp. 508-513.
- [Σ4].John Garofalakis, Evangelos Sakkopoulos, Spiros Sirmakessis, Athanasios Tsakalidis "Integrating Adaptive Techniques with Web Services", IEEE International Conference on Information Technology: Coding & Computing (IEEE ITCC), April 28-30, 2003, Las Vegas, Nevada, USA, pp. 415-419.
- [Σ5].Evangelos Sakkopoulos, Athanasios Tsakalidis "Utilizing Complementary Know-How: Advanced Fine Arts meeting Information Technology to provide a Virtual University for artists - students and alumni", ACM SIGUCCS User Services conference 2002, November 22, 2002, Providence, Rhode Island, USA, pp. 122-129.
- [Σ6].John Garofalakis, Theofanis Matsoukas, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis. "Personalization Techniques for Web Search Results Categorization", in the proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE EEE 2005), 29 March - 1 April 2005 in Hong Kong, China, pp. 148-151.
- [Σ7].John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis. "Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?". International Workshop on Web Engineering, "Hypermedia Development & Web Engineering Principles and Techniques: Put them in use.", in conjunction with ACM Hypertext 2004, (ACM HT04), Santa Cruz, August 10, 2004, URL: http://www.ht04.org/workshops/WebEngineering/HT04WE_Garofalakis.pdf
- [Σ8].Lampros Drossos, Spyros Sioutas, Evangelos Sakkopoulos, Spiros Sirmakessis. «BDT-Grid: An Efficient Scalable Peer-to-Peer Lookup System for Web Service Discovery», International Workshop on Peer to Peer and Service Oriented Hypermedia: Techniques and Systems in conjunction with ACM HyperText 2005, to appear in the ACM DL HT05 Proceedings, Salzburg, Austria.
- [Σ9].Eleni Christopoulou, John Garofalakis, Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, "Automating Restructuring of Web Applications", ACM Hypertext 2002, June 11-15, 2002, College Park, Maryland, USA., ACM 1-58113-477-0/02/0006.
- [Σ10].Poulia Adamopoulou, Dimitrios Kanellopoulos, Evangelos Sakkopoulos and Athanasios Tsakalidis, "Semantic learning interventions using web services technology", IASTED 2005

International Conference Web Based Education, (WBE 2005), Web Based Teaching and Learning Technologies track, 2/2005, pp. 528-533

- [Σ11].Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, An Algorithmic Framework for Adaptive Web Content, in the Int WS on Adaptive and Personalized Semantic Web, in conjunction with ACM Hypertext 2005, (ACM HT05), proceedings published by Springer Verlag, Studies in Computational Intelligence, 2005, to appear
- [Σ12].Aglaia Liopa-Tsakalidis, Pantelis Barouhas, Athanasios Kouloupoulos, Evangelos Sakkopoulos, Giannis Tzimas, Leonidas Panagiotopoulos, “FloatNet: An Intelligent Environment for Tobacco Transplants Production”, in the proceedings of ITAFE05 - “International Congress on Information Technologies in Agriculture, Food and Environment”, 2005.

0.1 Εισαγωγικά Θέματα και Διάθροση της παρούσας Διατριβής

Το Διαδίκτυο (Internet), το οποίο στην πλέον δημοφιλή του έκφανση το αντιλαμβανόμαστε ως Παγκόσμιο Ιστό (World Wide Web), έχει εξελιχθεί με ραγδαίους ρυθμούς σε μια καθολική πηγή πληροφόρησης για κάθε σχεδόν θεματικό αντικείμενο αλλά παράλληλα και σε πλατφόρμα για τη σύγχρονη επιχειρηματική ηλεκτρονική δραστηριότητα. Αποτελεί το de facto κυρίαρχο μέσο πληροφόρησης και παροχής υπηρεσιών μέσω των σύγχρονων πληροφοριακών συστημάτων τόσο για τους χρήστες όσο και για τις επιχειρήσεις αντίστοιχα [8], [25]. Δισεκατομμύρια πηγές πληροφόρησης και σημεία παροχής υπηρεσιών μέσω Web είναι ήδη διαθέσιμα σήμερα, ενώ συνεχώς αυξάνονται σε αριθμό και ανανεώνεται το περιεχόμενό τους. Αυτή η συνεχής ανανέωση και αύξηση της ποσότητας των διαθέσιμων πηγών αλλά και των προσφερόμενων υπηρεσιών περιπλέκει τις διαδικασίες για διαχείριση και εντοπισμό της κατάλληλης υπηρεσίας ή πληροφορίας στο Web. Συχνά οδηγείται κανείς σε αναποτελεσματικές επιλογές και καθυστέρηση ή αποτυχία εντοπισμού της επιθυμητής υπηρεσίας ή πληροφορίας. Το πρόβλημα αυτό είναι ιδιαίτερα ευρύ και εμφανίζεται τόσο στις περιπτώσεις τυπικής αναζήτησης πληροφορίας σε μορφή σελίδων του Διαδικτύου από χρήστες όσο και στις περιπτώσεις διαχείρισης και αναζήτησης υπηρεσιών για τη διεκπεραίωση διαδικασιών ή λογικής από επιχειρήσεις με τεχνολογίες Παγκόσμιου Ιστού. Ως αποτέλεσμα απαιτούνται μεθοδική μελέτη και συστηματικές τεχνικές για την αποδοτική αντιμετώπισή του.

Συγκεκριμένα, οι υπηρεσίες Παγκόσμιου Ιστού (Web Services) [2] και συνολικότερα οι αρχιτεκτονικές πληροφοριακών συστημάτων βασισμένες σε Web Services είναι ένα από τα νεότερα μέλη στην περιοχή της μηχανικής Παγκόσμιου Ιστού (Web Engineering). Έχουν ήδη εξελιχθεί σε μια κοινά αποδεκτή πλατφόρμα -πρότυπο κατά W3C- για την επιχειρησιακή συνεργασία και τη διαλειτουργική επικοινωνία επόμενης γενιάς. Πρόκειται για την κυρίαρχη λύση στη διαμοίραση πληροφορίας αλλά και απομακρυσμένης εκτέλεσης διαδικασιών και αντικαθιστά παλαιότερες λύσεις όπως ήταν οι τεχνολογίες CORBA (Common Object Request Broker Architecture), RMI (Remote Method Invocation), και DCOM (Distributed Component Object Model) για να αναφέρουμε ορισμένες. Όλες αυτές οι λύσεις παρουσίασαν σημαντικά προβλήματα στην αποδοχή τους τόσο από τους επαγγελματίες της πληροφορικής όσο και από τη βιομηχανία και την ερευνητική κοινότητα. Αντίθετα οι Web Services βασίζονται στην ευρύτατη αποδοχή προτύπων XML [1] που υποστηρίζουν λειτουργίες όπως η περιγραφή υπηρεσιών (π.χ. WSDL [43]), η επικοινωνία (π.χ. SOAP [42]) και η καταλογογράφηση (π.χ. UDDI [16]) και έχουν σχεδιαστεί να επιτρέπουν την απευθείας επικοινωνία και ανταλλαγή δεδομένων σε εφαρμογές ανεξάρτητα από γλώσσα, πλατφόρμα και τοποθεσία. Αποτελούν τη βάση για δημιουργία και σύνθεση δικτυακών επιχειρηματικών εφαρμογών

βελτιώνοντας τη διαλειτουργικότητα ανάμεσα στις επιχειρήσεις. Αναμένεται κατακόρυφη αύξηση της διάθεσης νέων Web Services, ακόμη ευρύτερη υιοθέτησή τους και κατά συνέπεια θα υπάρχει αυξημένη ανάγκη για βελτίωση στις διαδικασίες διαχειρίσής και αναζήτησής τους. Παρόλο που η διαχείριση και η αναζήτηση Web Services με συγκεκριμένες επιχειρηματικές και λειτουργικές προδιαγραφές υποστηρίζονται από τους καταλόγους Web Services (WS registries, UDDI), η δυναμική διαχείριση και αναζήτηση είναι σε πρώιμο μόνο στάδιο ακόμη.

Η επιστημονική κοινότητα έχει αφιερώσει πολύ χρόνο και ακόμη επικεντρώνεται σε προσπάθειες ανάπτυξης κατάλληλων τεχνικών που έχουν στόχο να βελτιώνουν την ποιότητα και να αυξήσουν την απόδοση στις διαδικασίες διαχείρισης, αναζήτησης και εξατομίκευσης πληροφοριών για δικτυακά πληροφοριακά συστήματα (netcentric information systems) [45]. Καθορισμός της έννοιας της ποιότητας σε Web Services γίνεται πρωταρχικά στην εργασία [24].

Υπάρχει μία σειρά από ερευνητικές εργασίες που αφορούν στη διαχείριση και αναζήτηση Web Service βασισμένη σε λέξεις κλειδιά ή περιγραφή που να υποστηρίζει ποιοτικά χαρακτηριστικά κάποιου είδους ([18], [19], [39], [40], [44]). Συστηματική αντιμετώπιση πραγματοποιείται όμως μόλις το 2004-2005 στην εργασία [14].

Αντικείμενο του πρώτου (1^{ου}) κεφαλαίου της παρούσας διατριβής είναι η αναλυτική και συγκριτική παρουσίαση των λύσεων και προβλημάτων που αφορούν στη διαχείριση και αναζήτηση Web Service για δικτυακά πληροφοριακά συστήματα [Π5] με έμφαση σε εκείνες τις προσεγγίσεις που αφορούν τη διαχείριση ποιότητας υπηρεσίας. Μελετήσαμε και οργανώσαμε τις λύσεις αυτές σε μια νέα κατηγοριοποίηση [Σ7] που περιλαμβάνει σειρά από αναλυτικούς πίνακες και συγκριτικούς χάρτες σύμφωνα με τις εργασίες μας:

[Π5] Yannis Panagis, Evangelos Sakkopoulos, Spiros Sioutas and Athanasios Tsakalidis, "Web Service Integration and Management Strategies for Large Scale Datasets", book chapter in ed. Prof. Zongmin Ma, Database Modelling for Industrial Data Management: Emerging Technologies and Applications, Idea Group Inc., 2006, pp. 217-243.

[Σ7] John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis. "Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?". International Workshop on Web Engineering, "Hypermedia Development & Web Engineering Principles and Techniques: Put them in use.", in conjunction with ACM Hypertext 2004, (ACM HT04), Santa Cruz, August 10, 2004

Στο δεύτερο (2^ο) κεφάλαιο της διατριβής παρουσιάζουμε αλγόριθμους [Π1], [Σ4] για αποδοτική και προσαρμοστική επιλογή Web Service που λαμβάνουν υπόψη πέρα

από τα λειτουργικά χαρακτηριστικά και μη λειτουργικές προδιαγραφές για ποιότητα (Quality of Service-QoS) κατά την προσπάθεια σύνδεσης αλλά και χρήσης (consumption) της Web Service. Τέτοιες προδιαγραφές μπορούν να περιλαμβάνουν παραμέτρους όπως το χρόνο εκτέλεσης όσο και το χρόνο αναμονής. Στη διδακτορική διατριβή, εισάγουμε λύσεις που επιλέγουν προσαρμοστικά την πλέον κατάλληλη Web Service με βάση πραγματικού χρόνου, μετρήσιμα στοιχεία.

Περιπτώσεις που αντιμετωπίζονται αφορούν λειτουργίες επιχειρησιακής συνεργασίας μεταξύ οργανισμών και φορέων που κατά συνέπεια συνδέονται σημασιολογικά στα πλαίσια μιας συγκεκριμένης επιχειρηματικής λογικής και δραστηριότητας. Ως αποτέλεσμα, οι συμμετέχοντες φορείς χρειάζεται να αναπτύξουν προγραμματιστικές διεπαφές για να διαχειριστούν και να επιλέξουν Web Services με συγκεκριμένη λειτουργικότητα ο ένας από τον άλλο. Επιπλέον οι όποιες αναζητήσεις Web Services απαιτούνται, πραγματοποιούνται μέσα σε συγκεκριμένα όρια καταλόγων WS που διαθέτουν συγκεκριμένοι συνεργάτες που θα πραγματοποιήσουν τις επιμέρους εργασίες. Στόχος μας είναι να αντιμετωπιστεί με τον αποδοτικότερο τρόπο η διαδικασία διαχείρισης και αναζήτησης για δυναμική επιλογή της πλέον κατάλληλης Web Service πριν ακριβώς από τη στιγμή της εκτέλεσης εξασφαλίζοντας παράλληλα με τις λειτουργικές απαιτήσεις και απαιτήσεις ποιότητας υπηρεσίας. Το κεφάλαιο στριίζεται στις ακόλουθες δημοσιευμένες εργασίες μας:

[Π1] Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Efficient and adaptive discovery techniques of web services handling large data sets”, in J. Systems and Software (JSS), Elsevier Science, corrected proof, in press.

[Σ4] John Garofalakis, Evangelos Sakkopoulos, Spiros Sirmakessis, Athanasios Tsakalidis “Integrating Adaptive Techniques with Web Services“, IEEE International Conference on Information Technology: Coding & Computing (IEEE ITCC), April 28-30, 2003, Las Vegas, Nevada, USA, pp. 415-419.

Στο τρίτο (3ο) κεφάλαιο εισάγουμε αλγορίθμους αναζήτησης Web Services με χρήση πρωτότυπων δικτυακών λύσεων που χρησιμοποιούν μορφές αποκεντριοποιημένων και ομότιμων δικτύων (decentralized - peer based networks) ειδικά σχεδιασμένων για καταλογογράφηση WS. Έως σήμερα, κατάλογοι βασισμένοι στο πρότυπο Καθολικής Περιγραφής, Ανακάλυψης και Ενσωμάτωσης (Universal Description, Discovery and Integration standard, UDDI) αποτελούν την κυρίαρχη τεχνολογική υποδομή για αναζήτηση Web Services. Οι κατάλογοι αυτοί έχουν συνήθως κεντριοποιημένη αρχιτεκτονική όπως στις εργασίες [11][26][27]. Ωστόσο, η προσέγγιση αυτή συναντά ήδη εμπόδια [20] μιας και ο αριθμός των Web Services αυξάνει και γίνονται όλο και περισσότερο δυναμικές. Εναλλακτικές προσεγγίσεις προτείνουν τη δημιουργία καταλόγων με βάση κατανεμημένες αρχιτεκτονικές στις οποίες κυριαρχούν οι λύσεις με υπάρχοντα ομότιμα δίκτυα (Peer to Peer) [12], [15], [28], [29], [37].

Αρχικά παρουσιάζουμε έναν αλγόριθμο όπου κόμβοι που αποθηκεύουν πληροφορίες για τις Web Services μπορούν αποδοτικά να εντοπιστούν χρησιμοποιώντας μια νέα δομή δεδομένων δεικτοδότησης για ομότιμα δίκτυα, η οποία υποστηρίζει κλιμάκωση αποδοτικά και την έχουμε ονομάσει NIPPERS.

Μία δεύτερη προσέγγιση αναζήτησης Web Services που εισάγουμε, είναι το δίκτυο βασισμένο στο BDT-GRID (Balanced Distributed Tree). Η λύση αυτή προσαρμόζει αποδοτικά επερωτήσεις ανανεώσεων καθώς κόμβοι προσθαφαιρούνται στο σύστημα και μπορεί να απαντήσει σε ερωτήματα ακόμη και όταν το σύστημα τροποποιείται συνεχώς.

Συγκριτικά αποτελέσματα παρουσιάζονται στον πίνακα παρακάτω:

Αρχιτεκτονικές για ανακάλυψη Web Services	Αναζήτηση	Ενημέρωση	Πληροφορία Δρομολόγησης
CHORD by Stoica et al [38]	$O(\log n)$	$O(\log^2 n)$ w.h.p.	$O(\log n)$ nodes
NIPPERS [Σ1]	$O(\log \log n)$ αναμενόμενη	$O(\log \log n)$ επιμερισμένη αναμενόμενη $O(n)$ χειρότερη περίπτωση	Εκθετικά μειούμενο
BDT-GRID [Σ8]	$O(\sqrt{\log n / \log \log n})$	$O(\sqrt{\log n / \log \log n})$ επιμερισμένη $\Theta(n)$ χειρότερη περίπτωση	Εκθετικά αυξανόμενο

Πίνακας 1: Συγκριτική Παρουσίαση Αποτελεσμάτων NIPPERS, BDT-GRID και Chord

Τα παραπάνω αποτελέσματα υποστηρίζονται από τις ακόλουθες δημοσιευμένες εργασίες μας:

[Σ1] Christos Makris, Evangelos Sakkopoulos, Spiros Sioutas, Peter Triantafillou and Athanasios Tsakalidis, Bill Vassiliadis, "NIPPERS: Network of InterPolated PeERS for Web Service Discovery", in the proceedings of the 2005 IEEE International Conference on Information Technology: Coding & Computing, Track Next Generation Web and Grid Systems (IEEE ITCC 2005), 2005 in Las Vegas, USA, Vol. 2, pp. 193-198.

[Σ8] Lampros Drossos, Spyros Sioutas, Evangelos Sakkopoulos, Spiros Sirmakessis. «BDT-Grid: An Efficient Scalable Peer-to-Peer Lookup System for Web Service Discovery», International Workshop on Peer to Peer and Service Oriented Hypermedia: Techniques and Systems in conjunction with ACM HyperText 2005, to appear in the ACM DL HT05 Proceedings, Salzburg, Austria.

Στη συνέχεια, στο τέταρτο (4^ο) κεφάλαιο της διδακτορικής διατριβής και στα πλαίσια της εξυπηρέτησης των αναγκών των τελικών χρηστών μέσα από δικτυακά πληροφοριακά συστήματα (netcentric information systems) εισάγαμε αλγορίθμους και τεχνικές για εξατομικευμένη [4], [3], [6] αναζήτηση πληροφορίας. Οι διαθέσιμες μηχανές αναζήτησης πληροφοριών ανεξάρτητα από την αποτελεσματικότητα στην επιλογή ορθών αποτελεσμάτων παρέχουν στο χρήστη μια πλειάδα από πιθανές απαντήσεις προτάσεις που δεν είναι πάντοτε αποδοτικά σχετικές με το αρχικό υποβληθέν ερώτημα. Αυτό το σενάριο συμβαίνει σε περιπτώσεις που υποβάλλονται γενικά ερωτήματα αναζήτησης και δημιουργούνται φαινόμενα πολυσημίας. Περιπτώσεις δηλαδή όπου ζητείται ένας όρος που λαμβάνει πολλαπλές έννοιες ανάλογα με τα συμφραζόμενα και το περιεχόμενο στο οποίο βρίσκεται. Για παράδειγμα ο όρος “spider” περιλαμβάνει τόσο τα web robots όσο και τα έντομα ή αντίστοιχα ο όρος “jaguar” περιλαμβάνει είτε το θηλαστικό ή αποτελέσματα σχετικά με αυτοκίνητα. Οι μηχανές αναζήτησης αντιμετωπίζουν μόνο μερικώς το πρόβλημα βαθμολογώντας τα αποτελέσματα με βάση τη δημοτικότητα των σελίδων στο γράφο του web. Ωστόσο μεγάλο ποσοστό των χρηστών χρειάζεται να σπαταλούν μεγάλο χρόνο αναζήτησης ψάχνοντας μέσα στη λίστα των αποτελεσμάτων. Έχει βρεθεί ότι το 85% των χρηστών επικεντρώνεται στις δέκα πρώτες προτάσεις και δεν τροποποιεί το ερώτημα εάν δε δοκιμάσει πρώτα το 75% των δυνατών επιλογών. Πρέπει να λάβουμε όμως υπόψη ότι οι χρήστες τις περισσότερες φορές έχουν στο μυαλό τους συγκεκριμένες κατηγορίες αποτελεσμάτων που τους αφορούν και ταιριάζουν στις ανάγκες του συγκεκριμένου ερωτήματος. Η τελευταία παρατήρηση παίζει ακόμη ίσως και υποσυνείδητα μόνο ρόλο στον τρόπο αποκλεισμού ανεπιθύμητων αποτελεσμάτων-προτάσεων.

Παλιότερες προσπάθειες περιλαμβάνουν διάφορες τεχνικές βελτίωσης της αναζήτησης πληροφορίας. Αρχικά η ένδειξη της κατηγορίας σε κάθε προτεινόμενη απάντηση παρουσιάζεται από μηχανές όπως το Yahoo!, το A9 και το Google. [30], [33], [36]. Συσταδοποίηση των αποτελεσμάτων σε κατηγορίες πραγματοποιείται από την Adutoria [31] και το Vivisimo [34]. Περιβάλλοντα όπως η Letizia και το CiteSeer δημιουργούν προφίλ χρηστών (profile) ώστε να προτείνουν έγγραφα με βάση αυτά.

Εμείς με τη σειρά μας εστιάσαμε σε τεχνικές που παρουσιάζουν τα αποτελέσματα που αντιπροσωπεύουν το χρήστη υψηλότερα στη λίστα των επιστρεφόμενων εγγράφων μετά από αναζήτηση. Οι τεχνικές αυτές σχετίζονται με την ερευνητική περιοχή της εξατομικευμένης αναζήτησης (personalized search) και την προσαρμοστική παρουσίαση αποτελεσμάτων αναζήτησης. Στην περιοχή καθοριστικό σημείο υπήρξε η παλαιότερη εργασία του επηρεαζόμενου από κατηγορίες page-rank (topic sensitive page-rank) [14] που είναι μια τροποποίηση του γνωστού αλγορίθμου ταξινόμησης αποτελεσμάτων αναζήτησης page-rank [21]. Στην τεχνική αυτή, εισάγεται η επιτυχημένη τροποποίηση της βαθμολόγησης των αποτελεσμάτων με βάση την κατηγορία από όπου προέρχονται, με περιορισμένα όμως πειραματικά αποτελέσματα. Επιπλέον στην εργασία των Liu et al. [13] προτείνεται μια τεχνική που χρησιμοποιεί το ιστορικό προφίλ του χρήστη (τύπου

Rocchio) για να αντιστοιχίσει ερωτήματα σε κατηγορίες. Στην περίπτωση αυτή οι κατηγορίες του δημιουργούμενου προφίλ βασίζονται στο ODP (Open Directory Project) [17] που πρόκειται για το μεγαλύτερο και πιο δημοφιλή κατάλογο κατηγοριών.

Στη διδακτορική διατριβή, προτείναμε μια σειρά από τρεις αλγορίθμους [Π2] που βασίζονται και αυτοί σε τεχνικές μετρικής συνδέσμων για να εξατομικεύσουμε τα αποτελέσματα της αναζήτησης ιστού. Το μεγαλύτερο πλεονέκτημα των τεχνικών που προτείνουμε είναι ότι επιτρέπουν, με τη χρήση μιας αρκετά απλής μεθοδολογίας, την εξατομικευση των αποτελεσμάτων αναζήτησης, χωρίς να επιβαρύνονται οι χρήστες σε όγκο αποθήκευσης ή με καθυστερήσεις λόγω χρόνου εκτέλεσής τους. Επιτυγχάνουμε εξατομικευμένη αναζήτηση εφαρμόζοντας τεχνικές ανάλυσης και επεξεργασίας συνδέσμων *όχι* στο γράφο ιστού αλλά για πρώτη φορά σε αρκετά *μικρότερους* εξατομικευμένους γράφους που σχηματίζονται από διαθέσιμες σημασιολογικές ταξινομίες.

Για να αξιολογήσουμε τους αλγορίθμους που παρουσιάζονται, εκτελέσαμε μια σειρά από πειράματα με βάση τις δημοφιλείς λίστες κατηγοριών του ODP και τα αποτελέσματα ερωτημάτων του Google. Οι καμπύλες ακρίβειας και ανάκλησης των αποτελεσμάτων δείχνουν ότι η στρατηγική που περιγράφεται είναι αποτελεσματική και αποδοτική.

Στο ίδιο κεφάλαιο παρουσιάζουμε και κάνουμε χρήση ενός νέου πολυεπίπεδου προφίλ [Σ2] για την καταγραφή των στοιχείων συμπεριφοράς του χρήστη, το οποίο έχει μεταξύ άλλων δυνατότητα συνεργατικής αλληλεπίδρασης και χρονικής απόσβεσης των στοιχείων του προφίλ.

Επιπλέον για να ολοκληρώσουμε όλες τις πτυχές της εξατομικευμένης αναζήτησης παρουσιάζουμε έναν επιπλέον μηχανισμό προσαρμοστικής παρουσίασης αποτελεσμάτων αναζήτησης [Π4], [Σ6] χρησιμοποιώντας επί τόπου (online) κατηγοριοποίηση με χρήση SVM (Support Vector Machine) [23], [7] κατηγοριοποιητή για τα αποτελέσματα που δεν έχουν προ-κατηγοριοποιηθεί από τη μηχανή αναζήτησης. Το πλεονέκτημα της μεθόδου είναι ο συνδυασμός προσωπικών αλλά και κοινών κατηγοριοποιητών για βελτιστοποίηση των αποτελεσμάτων.

Τα παραπάνω αποτελέσματα υποστηρίζονται από τις ακόλουθες δημοσιευμένες εργασίες μας:

[Π2] Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Category ranking for personalized search”, in the J. of Data and Knowledge Engineering (DKE), Elsevier Science, to appear.

[Π4] John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis (2006), “Optimization mechanism for Web Search Results using Topic Knowledge”, in the International Journal of Knowledge and Learning (IJKL), Vol. 2 (1&2), ISSN

(Online): 1741-1017 - ISSN (Print): 1741-1009, Inderscience Publishers, corrected proof, in press.

[Σ2] John Garofalakis, Evangelos Sakkopoulos, Spiros Sirmakessis, Athanasios Tsakalidis “Integrating Adaptive Techniques into Virtual University Learning Environment“, IEEE International Conference on Advanced Learning Technologies (IEEE ICALT), September 9- 12, 2002, Kazan Tatarstan, Russia, pp. 28 -33.

[Σ6] John Garofalakis, Theofanis Matsoukas, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis. “Personalization Techniques for Web Search Results Categorization”, in the proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE EEE 2005), 29 March - 1 April 2005 in Hong Kong, China, pp. 148-151.

Τέλος, στο πέμπτο (5^ο) κεφάλαιο της διατριβής θα παρουσιαστούν μια σειρά από εφαρμογές, αρχιτεκτονικές και λειτουργικά πλαίσια τα οποία αφορούν δικτυακά πληροφοριακά περιβάλλοντα στα οποία εφαρμόζουμε τεχνικές διαχείρισης υπηρεσιών και μηχανισμούς εξατομίκευσης πληροφοριών.

Πιο συγκεκριμένα θα παρουσιαστούν λύσεις για δικτυακή διαχείριση ασύρματων και ενσύρματων πόρων και συσκευών, για πληροφοριακό σύστημα βασισμένο σε τεχνολογίες Παγκόσμιου Ιστού με web services που διαχειρίζονται πολυμεσικές πληροφορίες, λύσεις για περιβάλλοντα εκπαίδευσης από απόσταση – εικονικού πανεπιστημίου που χρησιμοποιούν δυναμικό προσαρμοστικό μοντέλο χρήστη για τη διαχείριση, εμφάνιση περιεχομένου και αναζήτηση καθώς και λύσεις όπου πραγματοποιείται δυναμική αναπροσαρμογή του παγκόσμιου ιστού ώστε να διευκολύνεται ο εντοπισμός της τελικής επιθυμητής πληροφορίας. Κλείνοντας το τελευταίο κεφάλαιο, θα δοθούν στοιχεία για την περίπτωση ενός δικτυακού πληροφοριακού συστήματος γεωπονίας.

Οι παραπάνω εφαρμογές, αρχιτεκτονικές και λειτουργικά πλαίσια υποστηρίζονται από τις ακόλουθες δημοσιευμένες εργασίες μας:

[Π3] Dimitrios Kanellopoulos, Evangelos Sakkopoulos and Athanasios Tsakalidis (2005), “Semantic Mining and Web Service discovery techniques for media resources management”, in the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol. 1 (1), ISSN (Online): 1744-263X - ISSN (Print): 1744-2621, Inderscience Publishers, corrected proof, in **press**.

[Σ3] Eythimios Dimopoulos, Athanasios Panousis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “WiFiAdmin: An

Intelligent Web-based Management Environment for Wireless Networks”, in the IEEE International Conference on Information Technology and Applications 2005 (IEEE ICITA), Sydney 4-7/7/2005, pp. 508-513.

[Σ5] Evangelos Sakkopoulos, Athanasios Tsakalidis “Utilizing Complementary Know-How: Advanced Fine Arts meeting Information Technology to provide a Virtual University for artists - students and alumni”, ACM SIGUCCS User Services conference 2002, November 22, 2002, Providence, Rhode Island, USA, pp. 122-129.

[Σ9] Eleni Christopoulou, John Garofalakis, Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Automating Restructuring of Web Applications“, ACM Hypertext 2002, June 11-15, 2002, College Park, Maryland, USA., ACM 1-58113-477-0/02/0006.

[Σ10] Poulia Adamopoulou, Dimitrios Kanellopoulos, Evangelos Sakkopoulos and Athanasios Tsakalidis, “Semantic learning interventions using web services technology”, IASTED 2005 International Conference Web Based Education, (WBE 2005), Web Based Teaching and Learning Technologies track, 2/2005, pp. 528-533

[Σ11] Christos Makris, Yannis Panagis, Evangelos Sakkopoulos and Athanasios Tsakalidis, An Algorithmic Framework for Adaptive Web Content, in the Int WS on Adaptive and Personalized Semantic Web, in conjunction with ACM Hypertext 2005, (ACM HT05), proceedings published by Springer Verlag, Studies in Computational Intelligence, 2005

[Σ12] Aglaia Liopa-Tsakalidis, Pantelis Barouchas, Athanasios Koulopoulos, Evangelos Sakkopoulos, Giannis Tzimas, Leonidas Panagiotopoulos, “FloatNet: An Intelligent Environment for Tobacco Transplants Production”, in the proceedings of ITAFE05 - “International Congress on Information Technologies in Agriculture, Food and Environment”, 2005.

Βιβλιογραφία Εισαγωγής

- [1] Austin, D., Barbir, A., Ferris, C. and Garg, S. (eds.): Web Service Architecture Requirements. W3C Working Group Notes (2004) <http://www.w3.org/TR/wsa-reqs>
- [2] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion M., Ferris, C. and Orchard, D. (eds.): Web Services Architecture. W3C WG Note. <http://www.w3.org/TR/ws-arch/>
- [3] Brusilovsky, P. Maybury, M.T., “From adaptive hypermedia to the adaptive web”, Commun. ACM 45(5): 30-33, 2002
- [4] Brusilovsky, P., “Adaptive Hypermedia”, User Model. User-Adapt. Interact. 11(1-2): 2001, pp. 87-110
- [5] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. Introduction to Algorithms, 2nd Edition. The MIT Press, McGraw-Hill Book Company.
- [6] De Bra, P., Brusilovsky, P., Houben, G.J., “Adaptive hypermedia: from systems to framework”, ACM Comput. Surv. 31(4es): 12, 1999
- [7] Dumais, S.T., Platt, J. Heckerman, D., Sahami, M., “Inductive learning algorithms and representations for text categorization”, proceedings of ACM CIKM, 98, pp. 148-155
- [8] Gonsalves, A., “Survey says number of Web Services Projects Growing”, URL <http://www.techweb.com/wire/26801009>
- [9] Gruber, T.R. (1993) ‘A translation approach to portable ontologies’, Knowledge Acquisition, Vol.5, No.2, pp.199-220.
- [10] Haveliwala T.H, “Topic-Sensitive PageRank”, proceedings of the 11th International World Wide Web Conference (WWW2002), 2002, pp. 517–526.
- [11] Lacey, P.: Uddi & Dynamic Web Service Discovery. <http://www.ddj.com/articles/2004>
- [12] Li, Y., Zou, F., Wu, Z., Ma, F.: PWSD: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network, In Proc. APWeb04, LNCS 3007. (2004) 291-300
- [13] Liu F., Cl. Yu, W. Meng, “Personalized Web Search by Mapping User Queries to Categories”, in Proc. CIKM, pp.558—565
- [14] Liu, Y., Ngu, A., and Zeng, L. QoS Computation and Policing in Dynamic Web Service Selection. In proc. *World Wide Web Conference* (WWW2004), pp. 66 – 73, 2004.
- [15] Montebello, M., C. Abela, C.: DAML Enabled Web Services and Agents in the Semantic Web. Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services and Database Systems , LNCS. (2003) 46 – 58
- [16] OASIS UDDI Specifications TC - Committee Specifications <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>
- [17] Open Directory Project <http://www.dmoz.org>
- [18] Ouzzani M., Bouguettaya A.: Efficient Access to Web Services. IEEE Internet Computing, March/April (2004) 34-44
- [19] Ouzzani, M., Efficient Delivery of Web Services, PhD Thesis, Virginia Tech University. (2004)
- [20] Overhage, S.: On Specifying Web Services Using UDDI Improvements. 3rd Annual International Conference on Object-Oriented and Internet-based Technologies, Concepts, and Applications for a Networked World Net.ObjectDays, Germany (2002)

- [21]Page, L., Brin, S., Motwani, R., and Winograd, T., “The pagerank citation ranking: Bringing order to the web”, Technical report, Stanford Digital Libraries, 1998.
- [22]Patil, A., Oundhakar, S., Sheth, A. and Verma, K., (2004) ‘METEOR-S Web Service Annotation Framework’, Proceedings of WWW 2004 Conference, pp. 553-562.
- [23]Platt, J., “Fast training of support vector machines using sequential minimal optimization”, Advances in Kernel Methods -Support Vector Learning. B. Scholkopf, C. Burges, and A. Smola, eds., MIT Press, 1999, pp. 185-208
- [24]Ran S.: A Model for Web Services Discovery with QoS, ACM, ACM SIGecom Exchanges, Volume 4 , Issue 1 Spring. (2003) 1 - 10.
- [25]Roby, A., Murphet, T., “Web Services: IT efficiency today, powerful business tomorrow”, Accenture survey report, 2005, URL http://www.accenture.com/xdoc/en/services/integration/insights/web_survey.pdf
- [26]Rompothong, P., Senivongse, Tw.: A Query Federation of UDDI Registries, ISICT (2003)
- [27]Sajjanhar, A., Hou, J., Zhang, Y.: Algorithm for Web Services Matching, In Proc. APWeb 2004, Lecture notes in Computer Science 3007, Springer Verlag, (2004) 665-670
- [28]Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A scalable and ontology-based P2P infrastructure for semantic web services. Proc. 2nd Int. Conf. Sweden. (2002) 104–111.
- [29]Schmidt, C., Parashar, M. A.: Peer-to-Peer Approach to Web Service Discovery. World Wide Web: Internet and Web Information Systems, 7. (2004) 211-229
- [30]Search Engine: A9 Search <http://www.a9.com>
- [31]Search Engine: Adutopia <http://www.adutopia.com/>
- [32]Search Engine: CiteSeer <http://www.citeseer.com>
- [33]Search Engine: Google Search <http://www.google.com>
- [34]Search Engine: Vivisimo <http://www.vivisimo.com/>
- [35]Search Engine: Wisenut <http://www.wisenut.com/>
- [36]Search Engine: Yahoo! <http://www.yahoo.com>
- [37]Sivashanmugam, K., Verma, K., Mulye, R., Zhong, Z., and Sheth, A.: Speed-R: Semantic P2P environment for diverse Web Service registries, <http://webster.cs.uga.edu/~mulye/SemEnt/Speed-R.html>. (2004)
- [38]Stoica, I., Morris, R., Karger, D., Kaashoek M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for Internet applications. ACM SIGCOMM, San Diego, (2001)
- [39]Tao, Y. and Lin, K.-J. The design of QoS Broker Algorithms for QoS-capable Web Services. In proc. of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE’04), pp. 17-24, 2004.
- [40]Tao, Y., and Lin, K.-J. Service Selection Algorithms for Web Services with end-to-end QoS Constraints. In *Proceedings of the IEEE International Conference on E-Commerce Technology, CEC2004*, pp. 129-136, 2004.
- [41]W3C Semantic Web <http://www.w3.org/2001/sw/>
- [42]W3C SOAP 1.2 (2003) <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>
- [43]W3C WSDL (2004). URL <http://www.w3.org/TR/wsdl>

- [44] Zhuge, H. and Liu, J.: Flexible retrieval of Web Services. *The Journal of Systems and Software*, 70, 2004, pp. 107–116.
- [45] Jahnke, J.H, Kontogiannis, K., Stroulia, E., Tilley, S.R., Wong, K.: 3rd International Workshop on Net-Centric Computing (NCC 2001): Theme: Migrating to the Web. ICSE 2001, pp. 766-767

1 ΤΕΧΝΟΛΟΓΙΕΣ ΜΗΧΑΝΙΚΗΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ (WEB ENGINEERING) ΓΙΑ ΑΝΑΖΗΤΗΣΗ ΥΠΗΡΕΣΙΩΝ (WEB SERVICES) ΣΕ ΔΙΚΤΥΑΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ (NET-CENTRIC INFORMATION SYSTEMS)

Πρόλογος – Σύνοψη Κεφαλαίου

Η εισαγωγή στην ανάπτυξη λογισμικού των Web Services είναι μία από τις σημαντικότερες εξελίξεις στις τεχνολογίες μηχανικής παγκόσμιου ιστού τα τελευταία χρόνια.. Η ευρεία αποδοχή και αναγνωρισμένη αξία τους βασίζεται στο γεγονός ότι παρέχουν μια απάντηση στο πρόβλημα της ανάπτυξης δικτυακών πληροφοριακών συστημάτων και εφαρμογών ανεξάρτητη από το τεχνολογικό περιβάλλον που υιοθετείται. Εξίσου σημαντικοί είναι οι μηχανισμοί που επιτρέπουν την ανακάλυψη των υπηρεσιών με βάση τα λειτουργικά και μη χαρακτηριστικά τους. Στο κεφάλαιο παρουσιάζουμε αρχικά την αρχιτεκτονική των Web Services και εισάγουμε στη συνέχεια μια νέα οργάνωση και κατηγοριοποίηση των τεχνικών ανακάλυψής τους.

Το παρόν κεφάλαιο έχει την ακόλουθη δομή: Στην ενότητα 1.1 παρουσιάζεται μια σύντομη εισαγωγή και στην 1.2 δίνονται χρήσιμα προαπαιτούμενα στοιχεία. Στοχευμένες λεπτομέρειες για τις τεχνικές ανακάλυψης Web Services παρουσιάζονται αναλυτικά στην ενότητα 1.3. Στην ενότητα 1.4 περιγράφονται οι κύριοι παράγοντες στον τομέα της ανακάλυψης. Πτυχές που αφορούν τις διαφορετικές αρχιτεκτονικές εξετάζονται στην ενότητα 1.5. Τα μοντέλα δεδομένων που διευκολύνουν την ανακάλυψη συζητούνται στην παράγραφο 1.6. Η ποιότητα παροχής Web Services παρουσιάζεται στην παράγραφο 1.7. Τέλος, μια συνολική αξιολόγηση μαζί με μελλοντικά βήματα και τα συμπεράσματα παρουσιάζεται στην ενότητα 1.8.

1.1 Εισαγωγή

Οι υπηρεσίες παγκόσμιου ιστού, γνωστές με το όνομα Web Services (εν συντομία WS), έχουν χαρακτηρίσει τις σύγχρονες μεθοδολογίες μηχανικής Παγκόσμιου Ιστού και υποστηρίζονται ευρέως από προμηθευτές και χρήστες της τεχνολογίας της πληροφορικής (Information Technology-IT). Οι Web Services είναι μια από τις λίγες αρχιτεκτονικές που υιοθετήθηκαν ομόφωνα από τη βιομηχανία ανάπτυξης λογισμικού. Από τα πρώτα προσχέδια των τεχνικών προδιαγραφών των Web Services (WS Specifications), η αρχιτεκτονική των Web Services καθιερώθηκε ως το κυρίαρχο πρότυπο για κατανεμημένη ανάπτυξη λογισμικού.

Η μεγάλη εξάπλωσή τους βασίζεται στο γεγονός ότι αποτελούνται από συστάσεις και πρότυπα διεθνών οργανισμών (όπως το W3C και ο OASIS) βασισμένα στην XML. Εν συντομία, οι WS είναι διαλειτουργικά τμήματα λογισμικού που μπορούν να χρησιμοποιηθούν στην ενοποίηση εφαρμογών και στην ανάπτυξη εφαρμογών με λειτουργικά δομοστοιχεία (component based). Με άλλα λόγια, οι Web Services είναι συλλογές λειτουργιών –κομμάτια μεγαλύτερων εφαρμογών- που είναι

διαθέσιμες από απόσταση (remotely available) μέσω συνήθων διαδικτυακών πρωτοκόλλων, χωρίς να τίθενται περιορισμοί στις πλατφόρμες των δύο πλευρών επικοινωνίας.

Η αρχιτεκτονική Web Services έχει ήδη προτυποποιηθεί [56] και αποτελείται από τρία βασικά λειτουργικά υποσυστήματα (components): Την Γλώσσα Περιγραφής Web Services (WSDL) [56], μια γλώσσα που επιτρέπει τον επίσημο λειτουργικό χαρακτηρισμό των παρεχόμενων λειτουργιών, το SOAP (παλαιότερα γνωστό ως Simple Object Access Protocol όρος που πλέον καταργήθηκε) [49], ένα πρωτόκολλο που ορίζει τη μορφή της ανταλλαγής πληροφορίας, και το Universal Description, Discovery and Integration (UDDI) [29], ο οποίος είναι ένας κατάλογος με περιγραφές Web Services.

Όλα τα τρία υποσυστήματα που αναφέρθηκαν καθορίζονται χρησιμοποιώντας επεκτάσεις της απλής XML γλώσσας. Κάθε WS συναλλαγή λαμβάνει χώρα πάνω από εγκαθιδρυμένα διαδικτυακά πρωτόκολλα όπως το HTTP [16] και το FTP. Η υιοθέτηση των ανεξάρτητων από πλατφόρμα πρωτοκόλλων είναι το γεγονός που διευκόλυνε την ευρεία αποδοχή των Web Services ως μια πλατφόρμα που υλοποιεί ένα ευρύ φάσμα εφαρμογών. Οι εφαρμογές αυτές κυμαίνονται από πολύπλοκες υπηρεσίες όπως επιχειρηματικές συνεργασίες και διαχείριση σχέσεων πελατών (customer relationship management), έως πιο περιορισμένες υπηρεσίες όπως ο έλεγχος της τιμής των μετοχών και η κράτηση ή η επιβίβαση check in σε μια πτήση.

1.2 Προαπαιτούμενα

Στη συνέχεια παρέχουμε ορισμένες εισαγωγικές πληροφορίες για τις Web Services και αυτά που οδήγησαν στις ανάπτυξή τους, μαζί με βασικά στοιχεία και σχετικά πρότυπα. Μια βασική έννοια στη σύγχρονη ανάπτυξη λογισμικού είναι η Service Oriented Architecture (SOA) [56]. Η SOA είναι μια αρχιτεκτονική όπου το υλοποιημένο κατανεμημένο σύστημα έχει τα ακόλουθα γνωρίσματα:

1. Παρέχει διαφάνεια λειτουργίας
2. Λειτουργεί με επικοινωνία βασισμένη σε μηνύματα (message driven)
3. Οι υπηρεσίες του συνοδεύονται από περιγραφές υπηρεσίας
4. Είναι ανεξάρτητο από πλατφόρμα.

Τα πιο γνωστά πρωτόκολλα που περιλαμβάνει η αρχιτεκτονική των Web Services είναι τα HTTP, XML, SOAP, WSDL, και UDDI. Το HTTP και η XML είναι de facto αποδεκτά πρότυπα για την ευέλικτη και ανεξάρτητη πλατφόρμας ανταλλαγή πληροφοριών. Το HTTP και παρόμοια πρωτόκολλα μεταφοράς πληροφορίας είναι απλά οι ενδιάμεσοι για τη μεταφορά δεδομένων μεταξύ απομακρυσμένων Web Services (Το TCP/IP έχει τον ίδιο ρόλο, για παράδειγμα, για μεταφορά HTTP δεδομένων). Η XML, λόγω της επεκτασιμότητάς της, παρέχει ένα wrapper για τις

απαιτούμενες λειτουργικότητες της αρχιτεκτονικής των Web Services. Τα υπόλοιπα πρωτόκολλα βασίζονται τις προδιαγραφές τους πάνω στην XML και το HTTP.

SOAP [49] είναι ένα πρωτόκολλο που ορίζει τη μορφή της ανταλλαγής μηνυμάτων. Αυτή η ανταλλαγή πραγματοποιείται όταν ανακαλύπτεται, δεσμεύεται ή καταναλώνεται μια Web Service.

Web Service Description Language [56], (εν συντομία WSDL) είναι μια γλώσσα που περιγράφει τις λειτουργίες μιας Web Service και παρέχει επιπλέον πληροφορίες σχετικά με τους τρόπους που μπορεί να προσπελαστεί, τα σημεία που μπορεί να βρεθεί, κλπ.

Το Universal Description Discovery and Integration [29] (εν συντομία UDDI) ορίζει μια ξεχωριστή οντότητα (ένα κατάλογο) που μεσολαβεί στην αναπτυξιακή διαδικασία φιλοξενώντας τις περιγραφές των Web Services.

Στη συνέχεια παρουσιάζεται η οργάνωση της αρχιτεκτονικής των Web Services με μεγαλύτερη λεπτομέρεια.

Το τυπικό μοντέλο με βάση το οποίο λειτουργούν οι Web Services είναι το ονομαζόμενο «δημοσίευση-έυρεση-δέσμευση» (publish-find-bind). Αυτό το μοντέλο αποτελείται από τρεις οντότητες: τον πάροχο, τον καταναλωτή, και τον κατάλογο. Ο πάροχος εγγράφει τις υπηρεσίες του στον κατάλογο UDDI χρησιμοποιώντας το SOAP/WSDL και περιμένει για αιτήματα υπηρεσίας. Μια διαδικασία καταναλωτή υποβάλλει ερωτήματα στον κατάλογο για μια Web Service χρησιμοποιώντας μηνύματα SOAP. Σημειώνουμε ότι τα ερωτήματα που παρέχονται από τα UDDI προτυποποιημένα APIs αναζήτησης απευθύνονται μόνο στις λειτουργικές απαιτήσεις της WS που επιθυμεί ο καταναλωτής. Οι προσπάθειες να προστεθεί περισσότερη νοημοσύνη συζητούνται σε ξεχωριστή παράγραφο. Αφού ανακτηθούν τα αποτελέσματα, η διαδικασία κατανάλωσης καθορίζει τη μέθοδο δέσμευσης και στη συνέχεια ζητάει από τον πάροχο μέσω μηνυμάτων, την εκτέλεση μίας συγκεκριμένης υπηρεσίας. Στην ιδανική περίπτωση ο πάροχος απαντά, εκτελεί και επιστρέφει αποτελέσματα σε SOAP μηνύματα. Η διαδικασία παρουσιάζεται το σχήμα 1.

Όπως συζητήθηκε, οι Web Services χρησιμοποιούνται ευρέως στην κατανεμημένη ανάπτυξη εφαρμογών. Ωστόσο, ίσως δεν είναι ακόμα τόσο ξεκάθαρο κατά πόσο η αρχιτεκτονική των Web Services είναι κατάλληλη για ανάπτυξη εφαρμογών σε επιχειρησιακά και βιομηχανικά περιβάλλοντα. Στη συνέχεια παρέχονται επιχειρήματα για τη χρήση Web Services και επισημαίνονται ορισμένα από τα μειονεκτήματά τους.

Τα σύγχρονα συστήματα επιχειρησιακής πληροφορίας γίνονται όλο και πιο σύνθετα και είναι δύσκολο να συντηρηθούν. Ένας τρόπος για να μειώσουν τα μεγάλα επιχειρησιακά δίκτυα το μέγεθος της συνολικής πολυπλοκότητας είναι η κατανομή της υπολογιστικής προσπάθειας μεταξύ φυσικά-διαχωρισμένων υπολογιστικών κόμβων. Συνεπώς, προκύπτει η ανάγκη εκτέλεσης απομακρυσμένων υπολογιστικών διαδικασιών. Υπάρχουν αρκετά παραδείγματα τέτοιων πρακτικών, όπως η

ηλεκτρονική τραπεζική (Internet banking) και λύσεις ηλεκτρονικού εμπορίου (e-Commerce), διαδικασίες τιμολόγησης τηλεφωνικών κλήσεων ή εκτέλεση στατιστικών αναλύσεων στη θυγατρική μιας εταιρίας από το κεντρικό υποκατάστημα και αντίστροφα.

Σχήμα 1: Το μοντέλο των Web Services

Μια άλλη ανάγκη που συναντάται παντού είναι ο συντονισμός της εκτέλεσης διαδικασίας μεταξύ διαφορετικών οντοτήτων. Ένα τυπικό παράδειγμα είναι το εξής: Μια διαδικασία αγοραστή ενεργοποιεί ένα αίτημα σε μια υπηρεσία πώλησης με χρήση πιστωτικής κάρτας. Στην πλευρά του πωλητή χρειάζεται να επικυρωθεί το πιστωτικό όριο του αγοραστή καλώντας μια τρίτη διαδικασία. Αυτό δείχνει μια περίπτωση όπου εμπλέκονται πολλοί συμβαλλόμενοι. Ο συντονισμός των διαδικασιών μπορεί επίσης να συμβεί μέσα στον ίδιο οργανισμό. Για παράδειγμα, ας υποθέσουμε ότι ένας τοπικός χειριστής αερογραμμών χρειάζεται να ελέγξει τις προϋποθέσεις ενός πελάτη για ένα πρόγραμμα συχνού ταξιδιώτη -frequent flyer bonus- πριν την κράτηση. Ο χειριστής τότε πρέπει να καλέσει μια απομακρυσμένη διαδικασία για να συλλέξει δεδομένα για το συγκεκριμένο πελάτη πριν καλέσει την κατάλληλη κλήση στην διαδικασία κράτησης.

Ένας επιπλέον σημαντικός παράγοντας είναι η ευελιξία. Ο όρος «ευελιξία» μπορεί να μεταφραστεί με πολλαπλούς τρόπους, όπως για παράδειγμα η ευκολία ενσωμάτωσης επιχειρησιακών διαδικασιών, η εισαγωγή νέων εξωτερικών συνεργατών, ή ακόμα και η ανεξαρτησία από τις πλατφόρμες των μηχανών στα άκρα επικοινωνίας.

Το πλαίσιο των Web Services, σε έντονη αντίθεση με τα κατανεμημένα πρωτόκολλα που εμφανίστηκαν προηγούμενα παρέχει ικανοποιητικές απαντήσεις σε όλες τις προηγούμενες απαιτήσεις. Οι Web Services παρέχουν έναν ευέλικτο, εννοιολογικά

απλό, και ανεξάρτητο από πλατφόρμα τρόπο διαμοίρασης διαδικασιών και εκτέλεσης απομακρυσμένων διαδικασιών. Οι Web Services είναι ευκολότερο να καταναλωθούν και δεν απαιτούν μεγάλη λίστα a priori συμφωνιών στις μεθόδους επικοινωνίας (loosely coupled). Επιπλέον, επιτρέπεται η ενοποίηση νέων διαδικασιών ή συνεργατών και οι επεκτάσεις σε πρότυπα, καθώς έχουν χτιστεί με βάση τις ευέλικτες XML προδιαγραφές. Αυτό το γνώρισμα είναι που επιτρέπει την ανεξαρτησία πλατφόρμας μεταξύ των WS, παρέχοντας έτσι επιπλέον βαθμούς ελευθερίας. Όσον αφορά το συντονισμό των απομακρυσμένων διαδικασιών, αυτό υποστηρίζεται ενδογενώς από τις πρόσφατες προδιαγραφές της αρχιτεκτονικής των Web Services υπό τον όρο “Web Service Choreography” [57].

Παρόλα αυτά η αρχιτεκτονική WS, στις τρέχουσες προδιαγραφές της, παρουσιάζει ορισμένα μειονεκτήματα. Καταρχάς, ακόμα και εάν το μοντέλο διαχείρισης που υποστηρίζει είναι κατανεμημένο, το κομμάτι της ανακάλυψης υπηρεσιών είναι κεντροποιημένο (centralized). Οι κατάλογοι UDDI λαμβάνουν όλα τα αιτήματα κατανάλωσης WS, τα οποία μπορεί να οδηγούν σε σφάλματα μοναδικού σημείου και συμφορήσεις (bottleneck). Επιπλέον, οι οντότητες που αποθηκεύονται στον κατάλογο παρέχουν μόνο τεχνικά προσανατολισμένες περιγραφές των WS. Επομένως, η διαδικασία κατανάλωσης συνήθως χρειάζεται να δοκιμάσει επαναλαμβανόμενες φορές «trial and error» ώστε να εντοπίζει ποια δικτυακή διαδικασία θα δεσμεύσει. Ένα άλλο μειονέκτημα είναι ότι τα χαρακτηριστικά Ποιότητας Υπηρεσίας (QoWS) δεν περιλαμβάνονται στις WSDL περιγραφές. Αυτό μπορεί να δημιουργήσει προβληματικές διασυνδέσεις σε πολλές περιπτώσεις καθώς δεν υπάρχει κάτι που να υποδηλώνει ότι, για παράδειγμα, ένας σύνδεσμος είναι απαρχαιωμένος και έχει μεταφερθεί ή ότι απλά μια υπηρεσία δεν είναι πια διαθέσιμη. Ο Orth [30] επίσης αναφέρει την «υπευθυνότητα» ως «τρύπα» της αρχικής προδιαγραφής, από την άποψη ότι η επιβολή ενός διαχειριστικού σχήματος στη χρήση μιας WS δεν είναι εύκολο έργο.

1.3 Τεχνολογίες αναζήτησης υπηρεσιών Web Services

Έχοντας γνωρίσει περισσότερες λεπτομέρειες για την τεχνολογία των WS συνειδητοποιεί κανείς ότι καθώς η ζήτηση για κατανάλωση WS αυξάνεται, προκύπτει μια σειρά ερωτημάτων/ προβλημάτων που έχουν ανάγκη αντιμετώπισης, σχετικά με τις μεθόδους και τις διαδικασίες ανακάλυψης της «καταλληλότερης» προς χρήση WS. Το κεφάλαιο αυτό στοχεύει στη διεξοδική και κριτική παρουσίαση, εξέταση και ανάλυση των διαφορετικών τεχνολογιών, τεχνικών και μηχανισμών που εξυπηρετούν την αναζήτηση υπηρεσιών σε δικτυακά πληροφοριακά συστήματα.

Αρχικά πρέπει να οριστεί η σημασία του όρου «μηχανισμός αναζήτησης». Μια πρώτη περιγραφή των μηχανισμών αναζήτησης για τους παρόχους υπηρεσιών παρουσιάζεται στο [52] ως μια διαδικασία αντιστοίχισης. Είναι η διαδικασία εύρεσης ενός κατάλληλου παρόχου υπηρεσιών (service provider) για έναν πελάτη υπηρεσιών (service requester) μέσω ενός ενδιάμεσου πράκτορα (middle agent) [7]. Η διαδικασία περιλαμβάνει τα παρακάτω γενικά βήματα: α) Οι πάροχοι υπηρεσιών

δημοσιεύουν-διαφημίζουν (publish) τις λειτουργικές δυνατότητές τους στους ενδιαμέσους πράκτορες, β) οι ενδιαμέσοι πράκτορες αποθηκεύουν αυτήν την πληροφορία, γ) ένας πελάτης μπορεί να ρωτά τον ενδιαμέσο πράκτορα εάν γνωρίζει κάποιο πάροχο που να ικανοποιεί βέλτιστα τις ζητούμενες δυνατότητες, και δ) ο ενδιαμέσος πράκτορας, με στόχο να απαντήσει, προσπαθεί να αντιστοιχήσει το αίτημα με τις αποθηκευμένες διαφημίσεις και επιστρέφει ένα υποσύνολο των αποθηκευμένων διαφημίσεων των παρόχων.

Μια πιο σύγχρονη προσέγγιση [4] ορίζει ότι ο μηχανισμός ανακάλυψης WS με την ευρύτερη έννοια είναι «ο εντοπισμός μιας επεξεργάσιμης από μηχανή περιγραφής μιας WS που μπορεί να ήταν προηγουμένως άγνωστη και η οποία ικανοποιεί συγκεκριμένα λειτουργικά κριτήρια». Είναι μια υπηρεσία υπεύθυνη για την διαδικασία ανακάλυψης, ένας λογικός ρόλος, που θα μπορούσε να εκτελεστεί είτε από τον αιτούντα, ή από τον πάροχο είτε από κάποιο άλλο ενδιαμέσο πράκτορα.

Η κύρια χρήση των WS έως τώρα περιλαμβάνει την κλήση υπηρεσιών από απόσταση, με την αποστολή, και στις περισσότερες περιπτώσεις, τη λήψη μηνυμάτων. Αυτή η κατάσταση παρόλα αυτά δεν είναι αποδοτική λόγω των αναγκών που προβάλλουν σύνθετες εφαρμογές στα δικτυακά πληροφοριακά περιβάλλοντα. Υπάρχουν νέες απαιτήσεις εφαρμογών που χρειάζονται αφ' ενός πρόσβαση σε σύνθετες WS (composite WS) και αφετέρου πρέπει να είναι σε θέση να επιλέξουν μεταξύ μιας αφθονίας παρεχόμενων Web Services με την ίδια λειτουργικότητα αλλά διαφορετικά μη λειτουργικά – ποιοτικά χαρακτηριστικά (όπως π.χ. η δικτυακή καθυστέρηση της απάντησης). Προκειμένου να διευκρινιστεί καλύτερα η κατάσταση παρουσιάζουμε ορισμένα τυπικά σενάρια πιθανών εφαρμογών Web Service:

- Προμηθευτές δεδομένων, που παρέχουν δεδομένα όπως ενημερώσεις χρηματιστηριακών μετοχών.
- Ενοποίηση/ολοκλήρωση (integration) ενδοεπιχειρησιακών διαδικασιών, όπως αυτές που στέλνουν μια εντολή διεκπεραίωσης από ένα τμήμα της εταιρείας σε ένα άλλο. Ενοποίηση επιχειρησιακών εφαρμογών, για παράδειγμα, ενοποίηση της βάσης δεδομένων ηλεκτρονικών διευθύνσεων μιας εταιρείας με τη βάση δεδομένων ανθρωπίνου δυναμικού (HR)
- Ενοποίηση συστημάτων και διαδικασιών με απομακρυσμένους συνεργάτες, ακόμα και με ανταγωνιστές.
- Εφαρμογές Helpdesk, που δίνουν τη δυνατότητα στους χειριστές τεχνικής βοήθειας να αναζητήσουν/ τροποποιήσουν τα εσωτερικά δεδομένα των πελατών.

Καθώς οι WS που βρίσκονται σε καταλόγους (repositories) μπορεί να συνοδεύονται από ένα πλήθος πληροφοριών, οι μέθοδοι για τον περιορισμό των αποτελεσμάτων της αναζήτησης στις WS εκείνες που ταιριάζουν με κάποια συγκεκριμένη λειτουργική προδιαγραφή μπορεί να είναι αρκετά πολύπλοκες.

Οι μηχανισμοί ανακάλυψης Web Services επιτρέπουν την πρόσβαση σε καταλόγους υπηρεσιών που μπορούν να αποθηκεύσουν πληροφορίες για τις ίδιες τις επιχειρήσεις - παρόχους, τις υπηρεσίες και άλλες περαιτέρω λεπτομέρειες. Υπό αυτήν την έννοια, τέτοιοι μηχανισμοί θα πρέπει να έχουν τη δυνατότητα να ανακτήσουν υπηρεσίες από ένα ευρύ φάσμα πληροφοριών σχετικά με τους ίδιους τους παρόχους υπηρεσιών, εκτός από τις διαφημιζόμενες υπηρεσίες τους (π.χ. υπηρεσίες εταιρειών μιας συγκεκριμένης γεωγραφικής – δικτυακής περιοχής (domain .gr)).

Επιπλέον, υπάρχει ανάγκη για δυναμικές δομές ανακάλυψης που θα είναι ενημερωμένες με σύγχρονο (online) τρόπο, παρέχοντας αποδοτικές και διαθέσιμες επιλογές από Web Services. Ο μηχανισμός ανακάλυψης θα πρέπει να παρέχει ένα πλήθος από δυνατότητες ανακάλυψης τόσο κατά το χρόνο ανάπτυξης μιας εφαρμογής που απαιτεί χρήση υπηρεσιών WS όσο και κατά το χρόνο εκτέλεσής της. Κατά τη διάρκεια της ανάπτυξης, μπορεί κανείς να αναζητήσει σε έναν WS κατάλογο πληροφορίες για τις διαθέσιμες υπηρεσίες. Κατά την εκτέλεση, οι εφαρμογές-πελάτες μπορούν να χρησιμοποιήσουν τον κατάλογο για να ανακαλύψουν όλα τα στιγμιότυπα (instances) μιας web service που ταιριάζουν με μια δεδομένη διεπαφή.

Το κύριο εμπόδιο που επηρεάζει αρνητικά τους μηχανισμούς ανακάλυψης Web Services είναι η ετερογένεια μεταξύ των υπηρεσιών. Η αναδυόμενη αρχιτεκτονική των Web Services (Web Services Architecture by W3C) θεωρεί μια προσέγγιση υψηλού επιπέδου για την ανακάλυψη [1]. Κάθε λύση που εξετάζεται σε αυτό το κεφάλαιο προσπαθεί να υπερνικήσει τις διαφορετικές πτυχές αυτής της ετερογένειας προκειμένου να επιλέξει την πιο κατάλληλη λειτουργικά και ποιοτικά Web Service που είναι διαθέσιμη. Ο προσδιορισμός των διαφορετικών ειδών ετερογένειας δίνει μια αίσθηση για αυτά που πρέπει να ληφθούν υπόψη προκειμένου να αποφευχθούν ή να μετριαστούν τα εμπόδια [33]:

- Τεχνολογικές ετερογένειες (διαφορετικές πλατφόρμες ή διαφορετικοί τύποι δεδομένων).
- Σημασιολογικές ετερογένειες (όροι και έννοιες προκαθορισμένου πεδίου εφαρμογών (domain-specific) οι οποίοι μπορεί να διαφέρουν μεταξύ τους, ειδικά όταν αναπτύσσονται από διαφορετικούς προμηθευτές).
- Πραγματολογικές ετερογένειες (διαφορετική ανάπτυξη διαδικασιών προκαθορισμένου πεδίου εφαρμογών και διαφορετικές αντιλήψεις σχετικά με την υποστήριξη έργων προκαθορισμένου πεδίου εφαρμογών).

Λαμβάνοντας υπόψη τα παραπάνω, το κεφάλαιο αυτό συνολικά παρουσιάζει κριτικά τις υπάρχουσες λύσεις για την ανακάλυψη Web Services και θέτει μελλοντικούς στόχους. Στη συνέχεια περιγράφονται οι κύριοι παράγοντες που συμμετέχουν στους μηχανισμούς ανακάλυψης.

1.4 Κύριοι Παράγοντες – Παίχτες στο Παιχνίδι της Ανακάλυψης

Οι μηχανισμοί Ανακάλυψης WS (WS Discovery) περιλαμβάνουν μια σειρά από καταλόγους, λίστες, λύσεις με ενδιαμέσους πράκτορες (agent-based) και λύσεις ομότιμων δικτύων (Peer to Peer -P2P). Το επικρατέστερο μοντέλο είναι η καταλογογράφηση με το πρότυπο Universal Description Discovery and Integration (UDDI) το οποίο σε τρέχον χρόνο βρίσκεται στην έκδοση 3 [29]. Μπορεί να θεωρηθεί σχετικά ώριμο δεδομένου ότι δεν έχει αλλάξει σημαντικά από την πρώτη έκδοση του. Στις ακόλουθες υποενότητες παρουσιάζονται οι διαφορετικοί κύριοι παράγοντες προκειμένου να υπάρξει μια πρώτη ταξινόμηση των διαθέσιμων λύσεων.

1.4.1 Λύσεις *Pro-Web Service*

Η αρχιτεκτονική Web Service είναι μια, σχετικά νέα, λύση σε ένα πλήθος παλιών προβλημάτων διαμοιρασμού δεδομένων και απομακρυσμένων κλήσεων.

Καθώς τα προβλήματα αυτά είναι αριστά παλιά, έχουν παρουσιαστεί αρκετές μεθοδολογίες οι οποίες οδήγησαν στην δημιουργία της Κατανεμημένης Ανάπτυξης Εφαρμογών (Distributed Application Development). Η Κατανεμημένη Ανάπτυξη Εφαρμογών είναι η τέχνη και η μηχανική της μεταφοράς δεδομένων από μια μηχανή σε μια άλλη. Η ανταλλαγή των δεδομένων και τα είδη των υπολογισμών που εκτελούνται μπορούν θεωρητικά να είναι οποιουδήποτε είδους: εντολές αγοράς, δεδομένα πελατών, μαθηματικοί υπολογισμοί, κλπ. Υπάρχουν αρκετές τεχνολογίες για την υλοποίηση εφαρμογών που μπορούν να στείλουν δεδομένα εμπρός και πίσω. Η CORBA (Common Object Request Broker Architecture), η RMI (Remote Method Invocation), και το DCOM (Distributed Component Object Model) είναι μερικές από αυτές.

Όλες οι προαναφερθείσες προσεγγίσεις δε κατόρθωσαν να κερδίσουν ευρεία υιοθέτηση από την κοινότητα της ανάπτυξης εφαρμογών καθώς παρουσίαζαν σημαντικές δυσκολίες. Χαρακτηριστικό παράδειγμα είναι το DCOM, το κατανεμημένο αντίστοιχο του Microsoft COM (Component Object Model). Η χρήση του DCOM περιορίζεται ουσιαστικά σε πλατφόρμες Windows, παρόλο που είναι εφικτό να αναπτυχθούν διαλειτουργικές (cross-platform) λύσεις. Η αρχιτεκτονική Common Object Request Broker Architecture, ή CORBA, είναι επίσης μια κατανεμημένη τεχνολογία που έχει σχεδιαστεί για να διευκολύνει την ανάπτυξη σε ετερογενή περιβάλλοντα. Εντούτοις, η ανάπτυξη εφαρμογών CORBA είναι ιδιαίτερα περίπλοκη, με αποτέλεσμα η CORBA να μην έχει παραμείνει πραγματικά δημοφιλής ούτε να έχει καθολική υιοθέτηση.

Οι απαιτήσεις για μια κατανεμημένη αρχιτεκτονική, η οποία θα βασιζόταν σε απλά, ανεξάρτητα από πλατφόρμα και διαδεδομένα πρωτόκολλα, προέκυψαν από τις δυσκολίες για την υιοθέτηση οποιασδήποτε από τις ήδη υπάρχουσες. Επομένως, η κοινότητα ανάπτυξης εφαρμογών οδηγήθηκε να επινοήσει την αρχιτεκτονική των Web Services.

1.4.2 Οι βασικοί φορείς στις διαδικασίες ανακάλυψης

Οι κατάλογοι των Web Services είναι η τεχνολογική βάση που κυριαρχεί στις υλοποιήσεις μηχανισμών ανακάλυψής τους. Είναι εξειδικευμένες αποθήκες (storage) που εφαρμόζουν ένα πλαίσιο προδιαγραφών ως μετασχήμα (metaschema). Ειδικότερα, πριν από το πρότυπο UDDI, οι οργανισμοί δε διέθεταν μια κοινή προσέγγιση για να δημοσιεύσουν πληροφορίες σχετικά με τα προϊόντα και τις web services που παρείχαν στους πελάτες και τους συνεργάτες τους. Το UDDI καθιέρωσε την πρώτη ομοιόμορφη μέθοδο που περιλάμβανε τις λεπτομέρειες για την ενοποίηση των ήδη υπαρχόντων συστημάτων και των διαδικασιών μεταξύ των επιχειρήσεων και συνεργατών.

Το UDDI επιτρέπει στις επιχειρήσεις να ανακαλύψουν και να διαμοιραστούν πληροφορίες όσον αφορά τις web services και άλλες ηλεκτρονικές και μη ηλεκτρονικές υπηρεσίες που εγγράφονται σε έναν κατάλογο (registry). Μια υπηρεσία καταλόγου UDDI είναι και η ίδια μια WS, η οποία διαχειρίζεται τις πληροφορίες για τους παρόχους υπηρεσιών, τις υλοποιήσεις υπηρεσιών, και τα μεταδεδομένα υπηρεσιών. Προκειμένου να εντοπιστεί μια web service που χρησιμοποιεί το UDDI, απαιτούνται πολλές πληροφορίες σχετικά με την υπηρεσία. Οι πληροφορίες που απαιτούνται περιλαμβάνουν τις περιγραφικές λέξεις – κλειδιά (key words) και μέρος του ονόματος της υπηρεσίας και των απαιτήσεων του χρήστη, προκειμένου να επιλεγεί η κατάλληλη υπηρεσία από τα αποτελέσματα του καταλόγου. Τα μέχρι στιγμής διαθέσιμα εργαλεία αναζήτησης είναι πολύ απλά και δε λαμβάνουν υπόψη οποιεσδήποτε συσχετίσεις μεταξύ των web services και των ποιοτικών χαρακτηριστικών κάθε web service, αναγκάζοντας τον χρήστη να επαναλάβει την αναζήτηση από την αρχή χρησιμοποιώντας νέες λέξεις κλειδιά.

Η υλοποίηση ενός καταλόγου UDDI μπορεί να εξυπηρετεί διαφορετικούς σκοπούς χρήσης (Πίνακας 2)

Οι προδιαγραφές UDDI περιλαμβάνουν:

1. SOAP APIs που επιτρέπουν τις ερωτήσεις και τη δημοσίευση πληροφοριών,
2. XML αναπαράσταση για το μοντέλο δεδομένων καταλόγου και τη μορφή των SOAP μηνυμάτων,
3. WSDL ορισμοί διεπαφών του SOAP και
4. APIs ορισμού για διάφορα τεχνικά μοντέλα που διευκολύνουν συστήματα κατηγορίας να αναγνωρίσουν και να κατηγοριοποιήσουν τις UDDI εγγραφές.

Τύπος	Περιγραφή
Δημόσιο	Η ερώτηση και η αντιστοίχιση πληροφοριών είναι δημόσιες σε όλους τους καταναλωτές web service. Υπό αυτήν την έννοια το δημόσιο UDDI είναι το ίδιο μια WS. Η δημοσίευση πληροφοριών στον κατάλογο υποστηρίζεται μέσω ασφαλών καναλιών (π.χ. https), αλλά αυτό δεν ανατρέπει το δημόσιο χαρακτήρα του. Υποστηρίζεται η ανταλλαγή δεδομένων με άλλους καταλόγους.
Προστατευόμενο	Η έννοια της εμπιστοσύνης μεταξύ των συνεργατών χαρακτηρίζει αυτό το είδος των καταλόγων. Τέτοιοι κατάλογοι υλοποιούνται σε ένα closed-group περιβάλλον με ελεγχόμενη πρόσβαση σε τρίτους. Τα διοικητικά χαρακτηριστικά μπορούν να μεταβιβαστούν σε έμπιστα συμβαλλόμενα μέρη. Η ανταλλαγή δεδομένων με άλλους καταλόγους επιτρέπεται μόνο εάν διευκρινίζεται ρητά κάτι τέτοιο.
Ιδιωτικό	Πρόκειται για πλήρως εξασφαλισμένους απομονωμένους καταλόγους. Είναι συνήθως κατάλογοι εξαρτώμενοι από το πεδίο εφαρμογής σε ένα εσωτερικό δίκτυο. Η ανταλλαγή δεδομένων με άλλους (εξωτερικούς) καταλόγους δεν επιτρέπεται.

Πίνακας 2: Τύποι του Καταλόγου UDDI

Χρησιμοποιώντας τις παραπάνω προδιαγραφές, αναγνωρίζεται ευρέως, αν και δεν αναφέρεται συγκεκριμένα στο πρότυπο, ότι υπάρχουν τρεις τύποι πληροφοριών που υποστηρίζονται από τον κατάλογο. Αυτοί οι τύποι περιλάμβαναν την εγγραφή των λευκών, κίτρινων και πράσινων σελίδων.

- Οι Λευκές Σελίδες (White pages) περιλαμβάνουν βασικές πληροφορίες επαφών και περιγραφικά αναγνωριστικά όπως όνομα οργανισμού, διεύθυνση, πληροφορίες επαφής, και άλλα μοναδικά ids.
- Οι Κίτρινες Σελίδες (Yellow pages) περιγράφουν μια web service χρησιμοποιώντας διαφορετικές ταξινομίες (taxonomies). Έτσι είναι πιθανό να ανακαλυφθεί μια Web Service με βάση την κατηγορία της.
- Οι Πράσινες Σελίδες (Green pages) περιλαμβάνουν τις τεχνολογικές πληροφορίες που περιγράφουν τις συμπεριφορές και τις συναρτήσεις υποστήριξης μιας Web Service.

Πριν προχωρήσουμε στην λογική της λύσης Ομότιμων δικτύων για την ανακάλυψη WS, θα πρέπει επίσης να περιληφθεί στον τύπο καταλόγου η απλουστευμένη εφαρμογή μιας απλής σελίδας – λίστας. Εν συντομία, πρόκειται για μια σελίδα στο Διαδίκτυο που δεν είναι κάτι παραπάνω από μια λίστα αναφορών για Web Services. Τέτοιες συλλογές (compilations) δεν είναι ούτε επιτακτικές ούτε επικυρωμένες. Είναι συνήθως συγκεντρωμένες συλλογές δημοσιευμένων πληροφοριών από τους φορείς παροχής υπηρεσιών (που συνήθως χρησιμοποιούν αυτοματοποιημένους μηχανισμούς συλλογής όπως web spiders).

Οι πλατφόρμες ομότιμων δικτύων (P2P) παρέχουν έναν καλό πεδίο δραστηριοποίησης για την υλοποίηση μηχανισμών ανακάλυψης Web Services. Ένα P2P δίκτυο παρέχει την υποδομή για τη δρομολόγηση και τον εντοπισμό δεδομένων

σε ένα αποκεντριοποιημένο (decentralized), αυτό-οργανωμένο περιβάλλον στο οποίο κάθε κόμβος (peer) ενεργεί όχι μόνο ως κόμβος που παρέχει τη δρομολόγηση και την υπηρεσία εντοπισμού δεδομένων, αλλά και ως κεντρικός υπολογιστής που παρέχει την πρόσβαση σε υπηρεσίες. Ένα P2P δίκτυο μπορεί να θεωρηθεί ως ένα πλήρες κατανεμημένο υπολογιστικό μοντέλο. Πρόσφατα προτεινόμενα P2P συστήματα περιλαμβάνουν το CAN [40], το Pastry [42], το Chord [48] και το NIPPERS [25]. Τα παραπάνω συστήματα ταξινομούν το δίκτυο των κόμβων σε ένα δακτύλιο. Σε κάθε κόμβο ανατίθεται ένα αναγνωριστικό (id) που προέρχεται από ένα καθολικό χώρο διευθύνσεων (global/universal address space). Στους κόμβους ανατίθεται επίσης μια σειρά από κλειδιά από τον καθολικό χώρο διευθύνσεων για τα οποία είναι αρμόδιοι. Κάθε κόμβος αποθηκεύει επίσης βοηθητικές πληροφορίες προκειμένου να δρομολογήσει κατάλληλα ευρέσεις κλειδιών (key lookups). Συνήθως, μια εύρεση κλειδιού αρχίζει σε έναν κόμβο. Σε αυτήν την περίπτωση, ο κόμβος συμβουλευτεί τον πίνακα εύρεσης κλειδιού του προκειμένου να δρομολογήσει επιτυχώς το ερώτημα στον κόμβο που αποθηκεύει το ερωτούμενο κλειδί. Στην περίπτωση της Chord δρομολόγησης με τη βοήθεια πίνακα εύρεσης κλειδιού, προσομοιώνεται η δυαδική αναζήτηση στο χώρο διευθύνσεων όλων των κόμβων, επομένως ένα αίτημα σε ένα δίκτυο N κόμβων μπορεί να δρομολογηθεί σε $O(\log N)$ μηνύματα. Οι συμμετέχοντες κόμβοι (hosts) στο P2P δίκτυο δημοσιεύουν τις περιγραφές υπηρεσιών τους και ως αποτέλεσμα οι χρήστες έχουν πρόσβαση στις ενημερωμένες υπηρεσίες Web Services. Στη συνέχεια συζητούνται εν συντομία οι διάφορες αρχιτεκτονικές πτυχές του Chord, ενώ στην παράγραφο 1.6 αναφέρονται περισσότερες λεπτομέρειες για το μοντέλο δεδομένων που χρησιμοποιεί.

Στο [19] παρουσιάζεται μια αρχιτεκτονική που ονομάζεται Ανακάλυψη Web Service με βάση Ομότιμα Δίκτυα (P2P-based Web Service Discovery-PWSD). Οι συγγραφείς χρησιμοποιούν ένα Chord πρωτόκολλο ομότιμων δικτύων ως επικάλυψη (overlay), το οποίο αποτελείται από Κόμβους παροχής Υπηρεσίας (Service Peers-SP). Κάθε SP αντιστοιχίζεται σε αρκετές Λογικές Μηχανές (Logical Machines), όπου διαφορετικές μηχανές αντιστοιχούν στο ίδιο υλικό. Κάθε Λογική Μηχανή διατηρεί τις απαραίτητες διεπαφές για να αντιστοιχήσει και να αναζητήσει WS στο ομότιμο δίκτυο. Οι Περιγραφές Υπηρεσίας (Service Descriptions) κατανέμονται με χρήση hashing και δρομολογούνται στο Chord δίκτυο.

Ένα παρόμοιο μηχανισμό χρησιμοποιεί το σύστημα Speed-R [47]. Είναι ένα σύστημα αποθήκευσης και ανάκτησης WS που χρησιμοποιεί οντολογίες και μια υποδομή ομότιμων δικτύων. Κάποιοι κόμβοι στο P2P υποσύστημα ανατίθενται σε καταλόγους οι οποίοι με τη σειρά τους χωρίζονται σύμφωνα με την περιοχή εφαρμογής τους. Μια οντολογία ανατίθεται σε κάθε περιοχή. Το σύστημα ομότιμων δικτύων υλοποιήθηκε με χρήση της πλατφόρμας JXTA [13]. Η αρχιτεκτονική του βασίζεται στην ανάθεση ρόλων σε κόμβους από έναν κεντρικό ελεγκτή (για παράδειγμα ορισμένοι κόμβοι έχουν αναλάβει το ρόλο του ελέγχου ενημερώσεων και τη διάδοσή τους), επομένως το σύστημά τους είναι ευάλωτο σε δυσλειτουργία λόγω αποτυχίας μοναδικού σημείου (single point failure).

Σχήμα 2: Κατηγοριοποίηση (α) των προσεγγίσεων της ανακάλυψης Web Service

Σχήμα 3: Κατηγοριοποίηση (β) των προσεγγίσεων της ανακάλυψης Web Service

Κλείνοντας αυτήν την ενότητα, οι κατάλογοι και οι λύσεις με τεχνολογία ομότιμων δικτύων φαίνεται να είναι οι βασικοί φορείς για ανακάλυψη Web Service. Η υλοποίηση αυτών των μηχανισμών περιλαμβάνει αρκετές διαφορετικές πλευρές που παρουσιάζονται αναλυτικά στη συνέχεια. Ένας συνολικός γράφος ταξινόμησης

(taxonomization graph) παρουσιάζεται στο σχήμα 2 και το σχήμα 3. Όλες οι κατηγορίες που παρουσιάστηκαν απεικονίζονται στα στρογγυλεμένα ορθογώνια, ενώ οι αποδοτικότερες λύσεις σε κάθε κατηγορία περιλαμβάνονται στα κανονικά ορθογώνια ως φύλλα.

Τα δύο σχήματα παρέχονται ως σκιαγράφηση των διαθέσιμων μεθόδων. Δίνουν έμφαση μόνο στην προσπάθειά μας για κατηγοριοποίηση και ο αναγνώστης μπορεί να το χρησιμοποιήσει ως κατευθυντήριο χάρτη για να επιλέξει τις πιο κατάλληλες ιδιότητες και τις ικανότητες αντιστοίχησης, πριν εφαρμόσει μια λύση ανακάλυψης WS. Επιπλέον, η επιστημονική κοινότητα μπορεί να βρει για πρώτη φορά μια τόσο αναλυτική επισκόπηση των τεχνικών αναζήτησης και αντιστοίχησης WS.

1.5 Αρχιτεκτονική: Όψεις και Προσεγγίσεις

Υπάρχουν προσεγγίσεις της διαδικασίας ανακάλυψης WS που βασίζονται σε διαφορετικές αρχιτεκτονικές αντιλήψεις. Στις ακόλουθες ενότητες διαχωρίζουμε τους μηχανισμούς διαχείρισης WS σύμφωνα με:

1. το επίπεδο αυτοματοποίησης που παρέχουν,
2. τοπολογικά ζητήματα από την άποψη της συμμετοχής δικτύου,
3. συμμόρφωση με τα πρότυπα και τις συστάσεις, και,
4. τις διαθέσιμες πλατφόρμες.

1.5.1 Επίπεδα Αυτοματοποίησης – Χειροκίνητες και Έξυπνες Λύσεις

Μια αρχική προσέγγιση αυτού του τύπου κατηγοριοποίησης παρουσιάζεται στο [4] την αρχιτεκτονική των WS από την επιτροπή W3C. Η ανακάλυψη Web Service η οποία μπορεί να πραγματοποιείται με το χέρι (π.χ. από προγραμματιστές κατά τη διάρκεια του χρόνου ανάπτυξης) αλλά και αυτόματα (όπως από αυτό-συνδεδεμένες εφαρμογές κατά τη διάρκεια του χρόνου εκτέλεσης), είναι μια διαδικασία τριών φάσεων που αποτελείται από αναζήτηση Web service, αξιολόγηση και επιλογή Web services για τη διαδικασία διαμόρφωσης.

Στη χειροκίνητη διαδικασία ανακάλυψης, ένας χρήστης προσπαθεί μέσα από μια υπηρεσία ανακάλυψης (τυπικά σε χρόνο σχεδιασμού) να εντοπίσει και να επιλέξει μια περιγραφή υπηρεσίας που να καλύπτει τα επιθυμητά λειτουργικά και άλλα κριτήρια που θέτει. Στην έξυπνη αυτοματοποιημένη ανακάλυψη, ένας κατ' αντιστοιχία πράκτορας εκτελεί τη διαδικασία και αξιολογεί τα αποτελέσματα των αναζητήσεων, είτε σε χρόνο σχεδιασμού είτε σε χρόνο εκτέλεσης..

Οι διάφορες UDDI διαδικασίες και μηχανισμοί καλύπτουν απλώς λειτουργικές πτυχές της διαχείρισης δεδομένων και θέματα διατήρησης αντιγράφων. Είναι σχεδιασμένες και είναι κατάλληλες για τις ρητά δημοσιευμένες αλλαγές στα δεδομένα καταλόγου, οι οποίες γίνονται από τους χειριστές ή τους εκδότες. Ενώ αυτές οι διαδικασίες μπορούν να θεωρηθούν ως μια προσέγγιση για την αυτόματη

διαχείριση των αλλαγών στον κατάλογο, δεν αντιπροσωπεύουν μια λύση για το πρόβλημα της δυναμικής κλήσης υπηρεσιών ή της ανοχής σε σφάλματα (fault tolerance).

Για να ξεπεραστεί μερικώς το πρόβλημα της χειροκίνητης επιλογής, μια εναλλακτική “hacker” προσέγγιση, η οποία προτείνεται στο [21], είναι η αναβολή της απόφασης για το ποια υπηρεσία θα δεσμευθεί μέχρι το χρόνο εκτέλεσης, υποβάλλοντας ερωτήματα στο UDDI για τα σημεία πρόσβασης των υπηρεσιών που είναι γνωστά ότι παρέχουν υπηρεσία σύμφωνα με κάποιο συγκεκριμένο WSDL.

Στο [59] συζητείται ένας μηχανισμός που εκφράζει τη δυνατότητα των υπηρεσιών Web portal. Αυτός ο μηχανισμός μπορεί να χρησιμοποιηθεί για την ανίχνευση και την κατανάλωση υπηρεσιών Web portal. Επιτρέπει τον ακριβή εντοπισμό καθώς επίσης και την αυτοματοποιημένη ανακάλυψη και την κλήση των Web portal υπηρεσιών, αν και δεν παρέχεται εκτενής πειραματική υποστήριξη.

Μια πρωτότυπη online προσέγγιση ανακάλυψης υπηρεσίας με διασφάλιση ποιότητας (QoS) που αναπτύχθηκε στα πλαίσια αυτής της διδακτορικής διατριβής αναλύεται στο επόμενο κεφάλαιο 2. Η τεχνική βασίζεται στην αντιστοίχιση καλύτερης περίπτωσης χρησιμοποιώντας συνδυασμένα ιστορικά και QoS δεδομένα.

1.5.2 Κεντροποιημένες (Centralized) και Αποκεντροποιημένες (decentralized) Λύσεις

1.5.2.1 Κεντροποιημένες (Centralized) Υπηρεσίες

Ένας κατάλογος με περιγραφές και πληροφορίες για WS μπορεί να θεωρηθεί ως μια «αυθεντία» (authoritative) και τεχνολογικά είναι σε αδρές γραμμές μία κεντρικά ελεγχόμενη αποθήκη πληροφοριών (information storage). Ο κυρίαρχος εκπρόσωπος αυτής της κατηγορίας είναι ο κατάλογος UDDI [29]. Μια απλή έκδοση ενός καταλόγου είναι η κεντροποιημένη (centralised) υπηρεσία ευρετηρίου με συνδέσμους – δείκτες (index). Ένα ευρετήριο είναι μια συλλογή ή ένας οδηγός σε περεταίρω πληροφορίες που υπάρχουν κάπου αλλού. Δεν αποτελεί «αυθεντία» και δεν ελέγχει κεντρικά τις πληροφορίες στις οποίες έχει αναφορές. Η βασική διαφορά μεταξύ των δύο προσεγγίσεων δεν είναι απλά η διαφορά μεταξύ του ίδιου του καταλόγου και ενός ευρετηρίου με αναφορές. Πράγματι, το UDDI θα μπορούσε να χρησιμοποιηθεί ως μέσο για να υλοποιηθεί ένα ευρετήριο πραγματοποιώντας μια τυχαία διαπέραση σε σελίδες του Παγκόσμιου Ιστού και καταγράφοντας τα αποτελέσματα σε έναν κατάλογο UDDI. Η βασική διαφορά έγκειται σε θέματα ελέγχου: Ποιος ελέγχει τι και πώς ανακαλύπτονται οι περιγραφές υπηρεσιών; Στο μοντέλο καταλόγου, αυτό το ελέγχει ο ιδιοκτήτης του καταλόγου. Στο μοντέλο δεικτών, δεδομένου ότι ο οποιοσδήποτε μπορεί να δημιουργήσει ένα δείκτη, η αγορά καθορίζει ποιοι δείκτες θα γίνουν δημοφιλείς. Επομένως, είναι τελικά η αγορά που ελέγχει τι και πως θα ανακαλυφθούν οι περιγραφές υπηρεσιών [4].

1.5.2.2 Αποκεντριοποιημένες Λύσεις

Υπάρχει μια απλή, αλλά γνωστή και διαδεδομένη, προσέγγιση αποκεντριοποίησης. Είναι η περίπτωση όπου δημόσια διαθέσιμοι κόμβοι UDDI συνδέονται μεταξύ τους και διαμορφώνουν μια υπηρεσία η οποία, ενώ εμφανίζεται να είναι εικονικά ένα ενιαίο συστατικό, αποτελείται από έναν αυθαίρετο αριθμό κόμβων (operator nodes). Μια τέτοια λύση καλείται συνήθως είτε ως σύννεφο UDDI είτε ως ομοσπονδία (cloud or federation) [41]. Ένας κόμβος είναι αρμόδιος για τα δεδομένα που δημοσιεύονται σε αυτόν, πράγμα το οποίο σε όρους UDDI, είναι «ο επιστάτης εκείνου του μέρους των δεδομένων».

Σε περιπτώσεις τέτοιων αρχιτεκτονικών προσεγγίσεων των καταλόγων, θέματα που αφορούν την ακεραιότητα των δεδομένων αντιμετωπίζονται και προλαμβάνονται αποτελεσματικά με την χρήση διαδικασιών αντιγραφής δεδομένων, που υπάρχουν προτυποποιημένες στο UDDI.

Περισσότερο περίτεχνες και αποτελεσματικές αποκεντριοποιημένες λύσεις έχουν επίσης προταθεί. Τέτοιες είναι τα συστήματα [45], [47] τα οποία βασίζονται σε τεχνολογίες ομότιμων δικτύων (P2P) και σε οντολογίες για να δημοσιεύσουν και να αναζητήσουν περιγραφές Web Services. Μια λύση ομότιμων δικτύων έχει επίσης προταθεί στο [22]. Σε αυτή την περίπτωση, παρουσιάζεται ένα P2P σύστημα δεικτοδότησης και σχετική P2P αποθήκευση που υποστηρίζει μεγάλης κλίμακας, αποκεντριοποιημένες, πραγματικού χρόνου δυνατότητες αναζήτησης. Οι βασισμένες σε πράκτορα λύσεις περιλαμβάνουν την προσέγγιση της εργασίας [26]. Αυτή η προσέγγιση στοχεύει να περιγράψει ένα περιβάλλον που ονομάζεται DASD (DAML Agents for Service Discovery) όπου οι αιτούντες και οι πάροχοι WS μπορούν να ανακαλύψουν ο ένας τον άλλον με την ενδιάμεση δράση μιας υπηρεσίας αντιστοίχισης (matchmaking service). Δύο πρωτότυπες λύσεις που χρησιμοποιούν αποκεντριοποιημένη λογική αναπτύχθηκαν στα πλαίσια αυτής της διδακτορικής διατριβής και αναλύονται στο κεφάλαιο 3 της παρούσας διατριβής.

Μακρινοί πρόγονοι των κατανεμημένων καταλόγων εύρεσης είναι τα πρωτόκολλα εύρεσης Whois++ [55] και rWhois [43]. Τα δύο πρωτόκολλα παρέχουν online εύρεση στοιχείων ατόμων, πληροφοριών επιχειρήσεων, δικτυακών διευθύνσεων μηχανών κλπ. Η βασική ιδιότητά τους είναι η ιεραρχική και κατανεμημένη αρχιτεκτονική τους, σε παρόμοιο πνεύμα αλλά διαφορετικό πλαίσιο με τα σύγχρονα αποκεντριοποιημένα πρωτόκολλα εύρεσης.

1.5.3 Ακολουθώντας τα Πρότυπα και Αναθεωρώντας τα

1.5.3.1 Ακολουθώντας την Προδιαγραφή UDDI

Υπάρχουν προτάσεις βελτίωσης της απόδοσης και λειτουργικότητας των διαδικασιών ανακάλυψης που δεν ανατρέπουν σε τίποτε τις προδιαγραφές του UDDI και κατά συνέπεια είναι οι πλέον εφικτές και εφαρμόσιμες. Συγκεκριμένα, προκειμένου να εμπλουτιστεί η προδιαγραφή UDDI, οι πληροφορίες που απαιτούνται για την εύρεση WS προτείνεται να προστίθενται στα τμήματα διοικητικών πληροφοριών, δηλαδή τις λευκές και κίτρινες σελίδες που ήδη υπάρχουν [21]. Όταν η υψηλή απόδοση αποτελεί

προϋπόθεση, στην ποιότητα της υπηρεσίας συμπεριλαμβάνονται δεδομένα (όπως ο αναμενόμενος χρόνος μεταξύ αποτυχιών, ο μέγιστος χρόνος απόκρισης, η μέγιστη ρυθμαπόδοση, κλπ), τα οποία είναι κρίσιμα για να αξιολογήσουν την επιλογή του. Για να είναι σε θέση η διαδικασία ανακάλυψης να πληροφορηθεί για τις προδιαγραφές ασφάλειας μιας Web service, οι σχετικές λεπτομέρειες πρέπει να περιλαμβάνονται και αυτές.

Μια περαιτέρω πτυχή στην κατεύθυνση της βελτίωσης της ανακάλυψης είναι πραγματοποίηση βελτιώσεων στις πράσινες σελίδες [33]. Η συμπεριφορά των Web Services (συγκεκριμένα των περιλαμβανομένων μεθόδων τους) πρέπει να τεκμηριωθεί διευκρινίζοντας τις συνθήκες που απαιτούνται πριν και μετά την εκτέλεσή τους. Επομένως, ως βελτίωση προτείνεται ο σχεδιασμός με συμβόλαιο (by contract) [40]. Μια «προ-συνθήκη» εκφράζει τους περιορισμούς κάτω από τους οποίους μια κληθείσα μέθοδος επιστρέφει τα σωστά αποτελέσματα. Μια «μετα-συνθήκη» περιγράφει την κατάσταση που προκύπτει ως αποτέλεσμα της εκτέλεσης μιας μεθόδου και έτσι εγγυάται ότι θα ικανοποιήσει συγκεκριμένους όρους. Περιορισμοί σχετικά με τη διατεταγμένη κλήση των μεθόδων Web service μπορούν να εμφανιστούν μεταξύ των Web services. Καλούνται περιορισμοί συντονισμού και βοηθούν στη διαμόρφωση (configuring) μιας Web service στη βάση των διαδικασιών εφαρμογής.

1.5.3.2 Προσεγγίσεις που Αναθεωρούν τα Πρότυπα

Στα πλαίσια της ανάπτυξης λύσεων για τη βελτίωση της αναζήτησης έχουν προταθεί λύσεις που επαυξάνουν και αναθεωρούν πολλές φορές σημαντικά τα υπάρχοντα πρότυπα.

Παρότι παρέχει βασική υποστήριξη για απομακρυσμένη κλήση υπηρεσιών, το UDDI δεν υποστηρίζει δυναμική κλήση υπηρεσιών μέσα σε ένα δίκτυο κατανεμημένων υπηρεσιών. Η βασική προσέγγιση του ενεργού UDDI [18] είναι μια επέκταση της υπάρχουσας υποδομής UDDI χωρίς να απαιτούνται αλλαγές στις δομές δεδομένων ή τα ίδια τα APIs, αλλά χρησιμοποιώντας μιας τελειώς νέα web service που παίζει το ρόλο ενός ενδιάμεσου (man-in-the-middle). Αυτή η λύση παρέχει μια προσέγγιση με βάση proxy με σκοπό να παρέχει δυναμική ενημέρωση καταλόγου.

Μια διαφορετική χρήση για Web Service ανακάλυψη παρουσιάζεται στο [27]. Συγκεκριμένα, για να χτιστεί ένα ανοιχτό, μεγάλης κλίμακας και διαλειτουργικό, multi-agent σύστημα στα πλαίσια του Grid computing, έχει γίνει μια προσπάθεια για την ενοποίηση τεχνολογιών πρακτόρων (agent) με Web Services. Το Grid πρόβλημα ορίζεται ως ο ευέλικτος, ασφαλής, συντονισμένος διαμοιρασμός πόρων, ανάμεσα σε δυναμικές ομάδες ατόμων, οργανισμών και πόρων [11]. Μια επέκταση του καταλόγου UDDI χρησιμοποιείται, με επιπρόσθετες πληροφορίες (μετα-δεδομένα) σχετικά με πράκτορες και αναζήτηση βασισμένη σε οντολογίες (pattern-matching) προκειμένου να προσαρμοστεί το είδος αναζήτησης που απαιτείται για τον εντοπισμό μιας υπηρεσίας σύμφωνα με τις λειτουργίες που υποστηρίζει. Η

προτεινόμενη επέκταση του UDDI περιέχει WSDL περιγραφές όλων των πρακτόρων που έχουν εγγραφεί. Κατ' αυτό τον τρόπο επιτρέπονται η δυναμική ανακάλυψη και η κλήση υπηρεσιών από λογισμικό μέσω κοινής ορολογίας και κοινής έννοιας.

Στο [48] παρουσιάζεται η γλώσσα WSDL-S, η οποία δημιουργήθηκε επεκτείνοντας την WSDL 2.0. Αυτή η προσπάθεια μπορεί να χρησιμοποιηθεί τόσο σε κεντρικοποιημένους όσο και αποκεντρωποιημένους καταλόγους και ταυτόχρονα παρέχει μια ενδιαφέρουσα επέκταση των προτύπων επιτρέποντας συμβατότητα προς τα πίσω. Ειδικότερα, αυτή η γλώσσα παρέχει χώρο για σημασιολογικό σχολιασμό των web services και της λειτουργικότητάς τους. Ο σχολιασμός μπορεί είτε να παράγεται αυτόματα από αντίστοιχο σχολιασμό στον πηγαίο κώδικα μιας web service είτε να εισάγεται έπειτα στο αρχείο της βασισμένης σε WSDL περιγραφής.

Μια πρόταση, που παρακάμπτει το UDDI και πηγάζει από την περιοχή των βάσεων δεδομένων, είναι η ευέλικτη προσέγγιση ανάκτησης Web Service [61]. Σε αυτήν υιοθετείται ένας ορθογώνιος χώρος υπηρεσιών (orthogonal service space). Ο βαθμός ομοιότητας μεταξύ υπηρεσιών μετريέται με βάση τους συσχετισμούς μεταξύ των λειτουργιών που παρέχουν οι υπηρεσίες αυτές. Οι συγγραφείς παρέχουν μια SQL-like, ευέλικτη γλώσσα διατύπωσης ερωτημάτων και μια εφαρμογή που υλοποιεί μια web service, τον «ευέλικτο κατάλογο». Ο παρεχόμενος νοητός χώρος υπηρεσιών βοηθά τους χρήστες να εστιάσουν τις προθέσεις τους. Αφ' ετέρου, οι χρήστες μπορούν να περιγράψουν τον προσορισμό ανάκτησής τους με ευέλικτο τρόπο, επιλέγοντας τις σχέσεις ειδίκευσης και καθορίζοντας τον βαθμό ομοιότητας μεταξύ της πηγής και του στόχου.

Μια τεχνική απόφασης αντιστοίχισης με βάση κανόνες (rule based decision matching technique) προτείνεται στο [10] όπου παρουσιάζονται ακριβέστερα αποτελέσματα αντιστοίχισης, σε σύγκριση με τις υπάρχουσες συντακτικές τεχνικές αντιστοίχισης, καθώς βασίζεται στη λειτουργική σημασιολογία των web services. Χρησιμοποιούνται δίκτυα σεναρίου για να περιγράψουν μερικώς την εξωτερική συμπεριφορά των web services. Προτείνεται μια πολλαπλή προσέγγιση αντιστοίχισης με βάση τη σημασιολογία για δίκτυα σεναρίου, υιοθετώντας την έννοια της λειτουργικής αντικατάστασης.

Στο [60] προτείνεται μια λύση που αντικαθιστά τις περιγραφές των WS που είναι βασισμένες σε πρότυπα WSDL με μια εναλλακτική τους θεώρηση που αποκαλείται WS-Net. Το WS-Net είναι μια εκτελέσιμη αρχιτεκτονική γλώσσα περιγραφής που ενσωματώνει τη σημασιολογία του Colored Petri-net με το ύψος και την καταννοησιμότητα των αντικειμενοστραφών εννοιών. Το WS-Net περιγράφει κάθε λειτουργικό υποσύστημα μιας web service σε τρία επίπεδα: το δίκτυο διεπαφής δηλώνει τις υπηρεσίες που παρέχει το λειτουργικό υποσύστημα σε άλλα λειτουργικά υποσυστήματα, το δίκτυο διασύνδεσης διευκρινίζει τις υπηρεσίες που αποκτά το λειτουργικό υποσύστημα για να ολοκληρώσει την αποστολή του, και το δίκτυο λειτουργικότητας περιγράφει τις εσωτερικές λειτουργικές συμπεριφορές του λειτουργικού υποσυστήματος. Η προσέγγιση αυτή βοηθά στην ενίσχυση της

αξιοπιστίας των προσανατολισμένων σε web-services εφαρμογών. Εντούτοις, η μετατροπή με το χέρι των προδιαγραφών WSDL σε προδιαγραφές WS-Net δεν είναι μια απλή διαδικασία.

Το σχήμα 4 παρουσιάζει προσεγγίσεις που αναφέρθηκαν παραπάνω σύμφωνα με την αρχιτεκτονική τους. Στην τομή των αξόνων υπάρχουν οι δυνατότητες αναζήτησης του προτύπου UDDI [29]. Καθώς διατρέχουμε τον οριζόντιο άξονα, οι τεχνικές τείνουν να παρέχουν ευρύτερες τροποποιήσεις στα τρέχοντα πρότυπα. Στην κάθετη κατεύθυνση και προς τα επάνω αυξάνεται η αποκεντριοποίηση, επομένως εμφανίζονται λύσεις βασισμένες σε ομότιμα δίκτυα P2P και Grid-like αρχιτεκτονικές. Στο επάνω δεξιό τεταρτημόριο τοποθετούνται οι πιο «αντι-UDDI» λύσεις. Σε φθίνουσα σειρά, οι επικρατέστερες από τις λύσεις που παρουσιάζονται σε αυτήν την ενότητα, εκτός από το κλασσικό UDDI, είναι

- Οι τεχνικές ομότιμων δικτύων.
- Οι πολλά υποσχόμενοι παρόλα αυτά όχι ευρέως υλοποιημένοι μηχανισμοί ανακάλυψης βασισμένοι σε Grid.
- Τέλος για το απώτερο μέλλον, σημασιολογικές τεχνικές με βάση ομότιμα δίκτυα.

Σχήμα 4: Κατηγοριοποίηση με βάση την αρχιτεκτονική προσέγγιση

1.5.4 Τεχνολογικά Περιβάλλοντα

Προσπαθώντας να ικανοποιήσουν την ανάγκη για λύσεις ανακάλυψης WS, σημαντικά τεχνολογικά περιβάλλοντα περιλαμβάνουν εγγενώς στις εγκαταστάσεις τους δυνατότητες αναζήτησης WS. Το λειτουργικό Windows 2003 έχει προ-εγκατεστημένο ένα UDDI κατάλογο, δεδομένου ότι πολλοί προμηθευτές J2EE

εγκαθιστούν υλοποιήσεις του UDDI στους δικούς τους εξυπηρετητές εφαρμογών (application servers) [21]. Τα διαφορετικά τεχνολογικά περιβάλλοντα θα είναι κατά συνέπεια το αντικείμενο της παρούσας ενότητας, με στόχο να περιγραφούν οι δημοφιλέστερες πλατφόρμες που έχουν το μεγαλύτερο μερίδιο της αγοράς ανάπτυξης εφαρμογών και λύσεων για δικτυακά πληροφοριακά περιβάλλοντα.

Java 2 Enterprise Edition (J2EE). (<http://java.sun.com/j2ee/index.jsp>) Η Sun Microsystems προωθεί το Java API της για καταλόγους XML (JAXR) ως γενικού σκοπού API για επικοινωνία με πολλαπλούς τύπους καταλόγου. Το JAXR επιτρέπει στους πελάτες του πρόσβαση στις Web Services η οποία παρέχεται από έναν πάροχο Web Services που δημοσιεύει Web Services που δημιουργήθηκαν βάση μιας υλοποίησης των JAXR προδιαγραφών.

Microsoft .NET. (<http://www.microsoft.com/net/>) Αρχικά, η Microsoft είχε την ανακάλυψη των Web Services με DISCO στην μορφή ενός αρχείου ανακάλυψης (DISCO). Ένα δημοσιευμένο αρχείο DISCO είναι ένα XML έγγραφο που συνδέεται με τις άλλες πηγές που περιγράφουν την Web Service. Παρά την ευρεία εξάπλωση του UDDI, η Microsoft το έχει υποστηρίξει με σκοπό να μεγιστοποιήσει την διαλειτουργικότητα μεταξύ των λύσεων σε ένα σύνολο από προδιαγραφές διαλειτουργικότητας. Εκτός από την παροχή ενός .NET UDDI εξυπηρετητή, το UDDI πακέτο ανάπτυξης εφαρμογών παρέχει υποστήριξη για το Visual Studio .NET και εξαρτάται από το πλαίσιο .NET. Προϊόντα όπως το Microsoft Office (XP και 2003) παρέχουν υποστήριξη για ανακάλυψη υπηρεσιών μέσω UDDI.

Java-Based APIs. Οι προδιαγραφές UDDI δεν καθορίζουν άμεσα ένα Java-based API για πρόσβαση σε έναν UDDI κατάλογο. Η προδιαγραφή Programmer's API ορίζει μόνο μια σειρά από μηνύματα SOAP που μπορεί να δεχτεί ένας κατάλογος UDDI. Επομένως, ένας προγραμματιστής Java που θέλει να προσπελάσει έναν κατάλογο UDDI μπορεί να το κάνει με ένα πλήθος τρόπων:

1. χρησιμοποιώντας ένα Java-based SOAP API,
2. χρησιμοποιώντας ένα custom Java-based UDDI client API είτε
3. χρησιμοποιώντας το JAXR.

Simple Web Services API (SWSAPI). (<http://aspn.activestate.com/ASPN/WebServices/SWSAPI>) Αρκετές κορυφαίες βιομηχανίες του Ανοικτού Λογισμικού (Open Source) όπως η PHP, Perl και η Python έχουν συνεργαστεί για να αναπτύξουν ένα API που να είναι αρκετά απλό για να χρησιμοποιηθεί ακόμα και από νέους χρήστες και επιπλέον να μπορεί να χειριστεί και σύνθετες Web Services. Σήμερα, το API επικεντρώνεται στα εξής προβλήματα:

1. ανάγνωση και εγγραφή WSDL αρχείων και URLs,
2. caching τοπικά απομακρυσμένων WSDL αρχείων,

3. κλήση υπηρεσιών που περιγράφονται από την WSDL (invocation) και
4. δυνατότητα προγραμματιστικής πρόσβασης στα εσωτερικά συστατικά του WSDL (introspection). Με το πέρασμα του χρόνου θα υπάρξουν σχετικές προδιαγραφές για την αντιμετώπιση server side θεμάτων, θεμάτων XML σχήματος κλπ.

Η J2EE και το .NET φαίνεται να διαθέτουν την πλειοψηφία των τρεχουσών λύσεων βασισμένων σε Web Services, γεγονός που οφείλεται κυρίως στην ενοποίηση και την υποστήριξη ισχυρών πλατφόρμων ανάπτυξης όπως το JBuilder και το Visual Studio .NET που απολαμβάνουν ευρεία αποδοχή από τους προγραμματιστές εφαρμογών.

1.6 Μοντέλα Δεδομένων

Μια σημαντική πτυχή της ανακάλυψης Web Services αφορά το ζήτημα της μοντελοποίησης (modeling) των υπηρεσιών. Ο όρος μοντέλο σε αυτό το πλαίσιο αναφέρεται στην αναπαράσταση των Web Services, μια διαδικασία που πραγματοποιείται πριν από την ανακάλυψή τους. Σε αυτή την ενότητα παρουσιάζουμε δύο εναλλακτικές απόψεις: πρώτα την προσέγγιση μέσα από το πεδίο της ανάκτησης πληροφορίας (information retrieval) και, στη συνέχεια, τη σημασιολογική προσέγγιση.

1.6.1 Μοντελοποίηση με Τεχνικές Ανάκτησης Πληροφορίας

Το απλούστερο μοντέλο δεδομένων είναι ο *Catalogue/Keyword Based*. Αυτό το πρότυπο ακολουθείται από το τρέχον UDDI πρότυπο και το μηχανισμό ανακαλύψεων που υποστηρίζει. Εν συντομία, η περιγραφή που συνοδεύει κάθε Web Service αποθηκεύεται στον κατάλογο UDDI μαζί με την τεχνική περιγραφή (tModel) που παρέχει τη λειτουργικότητα της υπηρεσίας. Το στάδιο ανάκτησης περιλαμβάνει ένα χρήστη ή ένα πρόγραμμα αναζήτησης, που εισάγουν ένα ερώτημα στον κατάλογο. Το ερώτημα αποτελείται από λέξεις κλειδιά, οι οποίες αντιστοιχίζονται σε αποθηκευμένες περιγραφές. Οι αντιστοιχημένες Web Services επιστρέφονται έπειτα ως σύνολο υποψήφιων επιλογών. Στη συνέχεια ο χρήστης τις επεξεργάζεται προκειμένου να βρει ποια από αυτές ανταποκρίνεται πραγματικά στις ανάγκες του ή, αυτό που τείνει να είναι μια συχνή περίπτωση, υποβάλλει εκ νέου μια άλλη ερώτηση.

Η παραπάνω προσέγγιση που ακολουθείται από τους τρέχοντες UDDI καταλόγους, μοιάζει με το κλασσικό boolean μοντέλο ανάκτησης πληροφορίας [2]. Παρά την απλότητα και την ευκολία εφαρμογής του, πάσχει από μειονεκτήματα μιας και προκύπτουν είτε πολλά επιστρεφόμενα αποτελέσματα είτε πολύ περιορισμένα.

Ένα ακόμη μειονέκτημα της προσέγγισης αυτής είναι το γεγονός ότι συνήθως ένας κεντροποιημένος κατάλογος φιλοξενεί την πλειοψηφία των περιγραφών και επομένως λαμβάνει εκατομμύρια αιτημάτων με αποτέλεσμα να μετατρέπεται σε σημείο καθυστέρησης (bottleneck). Οι αποκεντρωποιημένες προτάσεις έχουν ήδη συζητηθεί στην παράγραφο 1.4.2.2.

Μια προσέγγιση για την αντιμετώπιση των προβλημάτων της αναζήτησης με λέξεις κλειδιά προτείνεται στο [44]. Η βασική ιδέα στην προσέγγιση αυτή, η οποία αποτελεί κυρίαρχη προσέγγιση στο πεδίο της ανάκτησης πληροφορίας (Information Retrieval – IR), είναι η αναπαράσταση των περιγραφών υπηρεσιών ως διανύσματα εγγράφου (document vector). Ένα κείμενο περιγραφής, κατά συνέπεια, αντιστοιχεί σε ένα διάνυσμα $\vec{d}$ στο διανυσματικό χώρο που περιλαμβάνει όλους τους όρους που χρησιμοποιούνται στα κείμενα περιγραφής υπηρεσίας. Προχωρούν ένα βήμα περαιτέρω με την αναπαράσταση όλων των διανυσμάτων εγγράφου, ως στήλες σε ένα πίνακα A όρων – εγγράφων (term-document matrix).

Μια ακόμη τεχνική από το πεδίο ανάκτησης πληροφορίας εφαρμόζεται στη συνέχεια, η οποία μετασχηματίζει τη μήτρα A επιτυγχάνοντας μια αναπαράσταση της συλλογής εγγράφων από τις σημαντικότερες σημασιολογικές έννοιές της, ή όπως ονομάζεται λανθάνουσα σημασιολογική δεικτοδότηση (Latent Semantic Indexing - LSI) [3]. Αυτή η μέθοδος αποδεικνύεται ότι είναι σε θέση να επιστρέψει έγγραφα της μοντελοποιημένης συλλογής κειμένων, τα οποία συσχετίζονται περισσότερο με τη σημασιολογία του διατυπωμένου ερωτήματος, ανεξάρτητα από την ακριβή αντιστοίχιση ή όχι με τους όρους του ερωτήματος.

Όταν εφαρμόστηκε το LSI στην ανακάλυψη Web Service παρατηρήθηκε ότι τα διανύσματα περιγραφής που προκύπτουν από το μετασχηματισμό της αρχική μήτρας, αντιστοιχήθηκαν πιο κοντά στην αναπαράσταση διανυσματικού χώρου του ερωτήματος, από ό,τι στις αντίστοιχες αναπαραστάσεις απλών περιγραφών με λέξεις κλειδιά.

Μια Web Service που μοντελοποιείται ως d -διάστατο διάνυσμα, μπορεί επίσης να θεωρηθεί ως ένα σημείο σε d -διαστάσεις. Από αυτήν την άποψη μια γεωμετρική δομή δεδομένων για δεικτοδότηση πολυδιάστατων δεδομένων μπορεί να επεκταθεί στη δεικτοδότηση και την υποβολή ερωτήσεων σε Web Services. Αντί για τη μεταφορά αυτού του προβλήματος σε μεγάλες διαστάσεις, οι Schmidt και Parashar [46], χρησιμοποίησαν έναν μετασχηματισμό ο οποίος αντιστοιχεί σημεία που βρίσκονται σε μεγάλες διαστάσεις, σε αριθμούς. Αυτός ο μετασχηματισμός καλείται καμπύλη πλήρωσης χώρου (space-filling). Πολλές space-filling καμπύλες έχουν προταθεί (δείτε [12]), αλλά ανάμεσα σε αυτές, η καμπύλη Hilbert έχει τη σημαντική ιδιότητα ότι τα παρακείμενα διαστήματα αντιστοιχούνται σε κοντινές περιοχές στις d -διαστάσεις.

Στο σύστημα των Schmidt και Parashar [46], δημιουργείται ένα μοναδικό id για κάθε Web Service, μέσω της αντιστοίχισης της καμπύλης Hilbert. Τα ids αποθηκεύονται στη συνέχεια σε ένα Chord [50] από κόμβους Web Service. Επομένως, η αποθήκευση και ανάκτηση της WS κληρονομεί την ικανότητα εξισορρόπησης φορτίου και τη δυναμική φύση του Chord. Το κύριο πλεονέκτημα του μοντέλου που ακολουθείται από το [46] είναι ότι μπορεί να υποστηρίξει αποτελεσματικά ερωτήματα μερικής αντιστοίχισης. Αυτά τα ερωτήματα πραγματοποιούνται αποτελεσματικά κυρίως λόγω των ιδιοτήτων συσταδοποίησης της

καμπύλης Hilbert. Η διαδικασία επερωτήσεων ενισχύεται επιπλέον από ευριστικές τεχνικές βελτιστοποίησης των ερωτημάτων που υποβάλλονται.

Οι Li et. al. [22] συνδύασαν την αντιστοίχιση λέξεων-κλειδιών με την αποθήκευση ομότιμων δικτύων παρουσιάζοντας ένα σύστημα που αντιστοιχίζει επίσης τις XML περιγραφές υπηρεσίας (WSDL) πάνω από ένα ομότιμο δίκτυο, χρησιμοποιώντας κατακερματισμένο κατακερματισμό. Από αυτήν την άποψη οι κόμβοι λειτουργούν τόσο ως πάροχοι υπηρεσιών όσο και ως γεννήτριες αιτημάτων. Οι XML περιγραφές υπηρεσίας διατρέχονται με στόχο την εξαγωγή λέξεων- κλειδιών υπηρεσιών και οι λέξεις- κλειδιά κατακερματίζονται με τη συνάρτηση MD5 hash. Το P2P πρωτόκολλο δικτύου είναι και στην περίπτωση αυτή ένα Chord [50] και το τροποποιημένο σύστημα ονομάζεται XChord. Η πολιτική διανομής Chord επιβάλλεται για να δρομολογήσει τις παραγόμενες κατακερματισμένες περιγραφές σε κόμβους. Ένα ερώτημα που αρχίζει σε έναν κόμβο, αναλύεται επίσης σε λέξεις – κλειδιά. Το XChord αποδεικνύεται πιο σταθερό, ισοζυγισμένο και καταναλώνει λιγότερο χώρο σύμφωνα με τα πραγματοποιημένα πειράματα.

1.6.2 Σημασιολογική Προσέγγιση

Πρόσφατες εργασίες έχουν εστιάσει στην εκτέλεση σημασιολογικής αντιστοίχισης για ανακάλυψη Web Services. Αυτή η εξέλιξη είναι όλο και πιο σημαντική δεδομένου ότι φαίνεται να είναι σε θέση να αντιμετωπίσει μερικές από τις ανεπάρκειες των καταλόγων UDDI. Το κυρίαρχο πρόβλημα είναι οι περιορισμοί που τίθενται από την αντιστοίχιση λέξεων-κλειδιών η οποία δεν επιτρέπει την ανάκτηση μιας WS με παρόμοια λειτουργικότητα, καθώς δύο περιγραφές WSDL μπορούν να χρησιμοποιηθούν για να περιγράψουν την ίδια υπηρεσία αλλά με διαφορετικές λέξεις. Εντούτοις, όταν μοντελοποιούνται Web Services με οντολογίες μπορεί να χρησιμοποιηθεί η σημασιολογική αναπαράσταση των εννοιών και των σχέσεών τους και έτσι να εκτελείται η σημασιολογική αντιστοίχιση. Οι σημασιολογικές περιγραφές των Web Services μπορούν να ληφθούν με τη χρήση των γλωσσών DAML-S [6] ή OWL-S [34].

Μια πρώιμη αναγνώριση της σπουδαιότητας της σημασιολογικής περιγραφής των Web Services βρίσκεται στην εργασία των Dogac et al. [8]. Σε αυτή, συζητούνται τα πλεονεκτήματα της περιγραφής της σημασιολογίας υπηρεσιών μέσω γλωσσών για οντολογίες και περιγράφονται οι τρόποι συσχετισμού της σημασιολογίας που ορίζονται με τις υπηρεσίες που διαφημίζονται στους καταλόγους υπηρεσιών.

Σε παράλληλη δράση, οι Paolucci et al. [35] παρουσίασαν ένα πλαίσιο για να επιτρέψουν στην WSDL και το UDDI να εκτελέσουν σημασιολογική αντιστοίχιση. Οι Web Services μοντελοποιούνται ως οντολογίες, ή προφίλ υπηρεσίας (Service Profiles) καθώς ονομάζονται, με τη χρήση της DAML-S [6]. Τυπικά, ένα προφίλ υπηρεσίας περιέχει τις πληροφορίες για τον πάροχο, λειτουργικές ιδιότητες όπως τη γεωγραφική θέση και λειτουργικές περιγραφές όπως οι είσοδοι/ έξοδοι της υπηρεσίας. Με τη διατήρηση των ιεραρχιών οντολογίας, είναι δυνατό να εκτελεσθεί σημασιολογική αντιστοίχιση, η οποία εκτελείται στη συνέχεια με την εκμετάλλευση

των ικανοτήτων υπαγωγής (subsumption) της DAML. Προκειμένου να συνδυαστούν οι οντολογίες με τον UDDI κατάλογο, οι συγγραφείς καθορίζουν ένα χωριστό επίπεδο, το DAML-S Matchmaker. Ο Matchmaker δεν επεκτείνει καμία από τις κατηγορίες σελίδων UDDI, αλλά αντιμετωπίζεται ως πρόσθετο, το οποίο αναλαμβάνει τη σημασιολογική αντιστοίχιση και την αντιστοίχιση των οντολογιών στις περιγραφές UDDI. Η σημασιολογική αντιστοίχιση, ειδικά όταν χρησιμοποιείται DAML-S, έχει διάφορα πλεονεκτήματα. Καταρχήν παρέχει ευελιξία αντιστοίχισης, επειδή επιστρέφονται αποτελέσματα που μπορούν να διαφέρουν συντακτικά με το ερώτημα εισόδου. Παρέχει επίσης ακρίβεια, δεδομένου ότι καμία αντιστοίχιση δεν εκτελείται εκτός αν προκύπτει από την ιεραρχία, και τέλος μπορεί να υποστηριχθεί η έννοια του βαθμού αντιστοίχισης.

Μια πρόσφατη εξέλιξη ήταν η εισαγωγή μιας νέας γλώσσας, της OWL-S, που συνδυάζει το σημασιολογικό σχολιασμό (annotation) των Web Services με την ανακάλυψη τους και την κλήση με WSDL και SOAP (δες [51]). Αυτή η προσέγγιση οδήγησε στο λειτουργικό υποσύστημα OWL-S Matchmaker.

Η προσέγγιση της προσθήκης επεκτάσεων DAML στις UDDI περιγραφές υιοθετείται επίσης στην εργασία [47]. Αντί για την παροχή ενός ξεχωριστού επιπέδου, εκτελείται μια απλούστερη σύνθεση ενεργοποιώντας τις WSDL περιγραφές και οι UDDI εγγραφές περιέχουν σημασιολογικές πληροφορίες. Αυτές οι πληροφορίες είναι απλά η αντιστοίχιση μεταξύ των WSDL εγγράφων και των DAML+OIL οντολογιών. Στην περίπτωση του UDDI, παρέχονται διαφορετικά tModels για να αναπαραστήσουν τη λειτουργικότητα, την είσοδο την έξοδο, κλπ. Εκτελείται μια διαδικασία αντιστοίχισης που χρησιμοποιεί πρότυπα και εικμεταλλεύεται τις σημασιολογικές περιγραφές για να παρέχει τη σημασιολογική αντιστοίχιση.

Οι Moreau et al. [27] πραγματοποίησαν επίσης ένα είδος σημασιολογικής αντιστοίχισης, αλλά σε διαφορετικό περιεχόμενο. Στην εργασία τους περιγράφουν πράκτορες που εκτελούν Grid υπολογισμούς, ως WSs. Επομένως, μετασχηματίζουν οντολογίες πρακτόρων σε XML και εκτελείται σημασιολογική αντιστοίχιση. Στην προσέγγιση του [26] ακολουθείται ένας συνδυασμός σημασιολογικού σχολιασμού των Web Services και agent-based δημοσίευση και ανακάλυψη.

Το σχήμα 5 παρουσιάζει τις προσεγγίσεις που αναφέρθηκαν σύμφωνα με την αναπαράσταση και τη μοντελοποίηση της διαδικασίας ανακάλυψης των Web Services. Στην τομή των αξόνων έχουν τοποθετηθεί οι τυπικές δυνατότητες αναζήτησης του UDDI [33]. Καθώς κινούμαστε στον οριζόντιο άξονα, οι τεχνικές τείνουν να παρέχουν καλύτερη επεκτασιμότητα στα τρέχοντα πρότυπα ανακάλυψης. Μια πολύτιμη τεχνική είναι αυτή των Schmidt και Parashar, η τεχνική PSWD που είναι μια λύση βασισμένη σε ομότιμα δίκτυα και προτείνεται στο [46]. Στην κάθετη κατεύθυνση και προς τα άνω αυξάνεται η ευελιξία αντιστοίχισης και εμφανίζονται μηχανισμοί βασισμένοι στο διανυσματικό-χώρο. Στο επάνω δεξιά τεταρτημόριο εμφανίζονται οι πιο ευέλικτες και επεκτάσιμες λύσεις, αν και οι τεχνικές σημασιολογικής αντιστοίχισης είναι ελπιδοφόρες αλλά μάλλον αποτελούν

στρατηγικές του απώτερου μέλλοντος. Σε φθίνουσα σειρά, οι κυρίαρχες λύσεις που παρουσιάζονται εδώ, εκτός από το κλασικό UDDI, είναι:

1. Τα πολλαπλά κριτήρια αναζήτησης που είναι το βασικό και πρώτο πλεονέκτημα σε αυτήν την κατηγορία. Η αναζήτηση με βάση το διανυσματικό χώρο μπορεί να είναι ιδιαίτερα αποτελεσματική και θα ενσωμάτωνε τις παραδοσιακές πολιτικές ανάκτησης πληροφορίας.
2. Η τεχνική PWSD που είναι πραγματικά επεκτάσιμη. Το σύστημα NIPPERS (κεφάλαιο 3) μπορεί επίσης να περιληφθεί στην ίδια κατηγορία.
3. Επίσης σε αυτήν την περίπτωση περιλαμβάνονται πολλά υποσχόμενες αλλά παρόλα αυτά όχι τόσο σταθερές λύσεις που είναι προσανατολισμένες σε σημασιολογικές τεχνικές.

Σχήμα 5: Ταξινόμηση με βάση την Επεκτασιμότητα και την Ευελιξία Εντοπισμού

Ενδιαφέρον πλαίσια εφαρμογής για τη σημασιολογική αντιστοίχιση προτείνονται ακόμη στα [15] και [33]. Η πρώτη εργασία παρέχει ένα πλαίσιο που χρησιμοποιεί οντολογίες για να ανακαλύψει («δεσμεύσει» σε αυτήν την περίπτωση) τις Web Services που ταιριάζουν καλύτερα σε μια συγκεκριμένη περιοχή λειτουργίας (επιθυμητό σύνολο διαδικασιών). Τα διαθέσιμα δεδομένα αναπαριστούνται με τις οντολογίες επιχειρησιακών περιοχών και οι διαθέσιμες λειτουργίες με τις οντολογίες λειτουργίας. Οι σχέσεις γενίκευσης και στα δύο μοντέλα κωδικοποιούνται σε μορφή XML και έτσι είναι δεσμευτικές σχέσεις. Η δέσμευση μπορεί επομένως να εκτελεσθεί από μια δεσμευτική οντολογία που αποφασίζει τι ταιριάζει που σύμφωνα με τις δεσμευτικές σχέσεις. Από την άλλη μεριά, ο Overhage στο [33] προτείνει την εφαρμογή των σημασιολογικών περιγραφών ως επέκταση του πρωτοκόλλου UDDI, με την ονομασία E-UDDI. Το E-UDDI εισάγει τις «Μπλε Σελίδες», τμήματα που περιέχουν τις σημασιολογικές περιγραφές των Web Services τα οποία υλοποιούνται στην DAML-S. Το μοντέλο που περιγράφεται στο [33], παρέχει επίσης επεκτάσεις στο τμήμα πράσινων σελίδων του UDDI, με την προσθήκη της ικανότητας να

καθορίζονται οι περιορισμοί στην ακολουθία εκτέλεσης της WS. Εντούτοις το E-UDDI φαίνεται να είναι περισσότερο ένα όραμα παρά ένα υλοποιήσιμο σύστημα. Το σχήμα 5, παρουσιάζει μια γραφική ταξινόμια των Μοντέλων Ανακάλυψης.

Μια αναλυτική προσέγγιση για σημασιολογικό σχολιασμό (semantic annotation) και τη δημοσίευση των Web Services εξετάζεται στα [37], [38], [48], τα οποία είναι σχετικά με το ενοποιημένο πλαίσιο METEOR-S Semantic Web Service Annotation Framework (MWSAF). Αρκετές μέθοδοι και εργαλεία συζητούνται με σκοπό να επιτρέψουν στους προγραμματιστές να ενσωματώσουν τις σημασιολογικές περιγραφές των Web Services κατά τη διάρκεια της ανάπτυξης του πηγαίου κώδικα. Αυτή η προσέγγιση επηρεάζει το μηχανισμό σχολιασμού που παρέχεται από την προγραμματιστική γλώσσα Java, αλλά μπορεί επίσης να υλοποιηθεί με τους αντίστοιχους μηχανισμούς σχολιασμού του C# .NET. Οι συγγραφείς επαληθεύουν τα αποτελέσματα υλοποιώντας έναν σχεδιασθή σημασιολογικών Web Services για το σχολιασμό πηγαίου κώδικα και μια γεννήτρια σημασιολογικών περιγραφών για την παραγωγή πλούσιων περιγραφών των Web Services.

Μια ακόμη εφαρμογή των εξελίξεων της έξυπνης επιλογής WS βρίσκεται στο [17] και είναι δανεισμένη από την περιοχή έρευνας πολυμέσων. Υποστηρίζεται ότι οι σημασιολογικές Web Services μπορούν να χρησιμεύσουν ως μια βασική τεχνολογία που θα επιτρέψει την επιτυχία των στόχων του Universal Multimedia Access (UMA), όπου οι χρήστες θα μπορούν να καταναλώσουν οποιοδήποτε πόρο πολυμέσων, οπουδήποτε, οποιαδήποτε στιγμή, το οποίο είναι και το κινητήριο όραμα των προσπαθειών προτυποποίησης της ομάδας τυποποίησης ISO/IEC Moving Picture Experts Group (MPEG).

1.6.3 Προσεγγιστική Ανάκτηση

Σε πολλές περιπτώσεις τόσο σε προβλήματα ανάκτηση πληροφορίας όσο και στην ανακάλυψη Web Services, είναι απαραίτητη η μερική/ προσεγγιστική αναζήτηση με βάση την ομοιότητα. Ο λόγος είναι ότι αν δεν είναι δυνατή η αντιστοίχιση με βάση την ομοιότητα, πρέπει να είναι γνωστές οι ακριβείς περιγραφές των υπηρεσιών. Η αναζήτηση ομοιότητας εμφανίζεται με δύο μορφές. Είτε ένας χρήστης υποβάλλει μια ερώτηση και ανακτά παρόμοια (μερική απάντηση) αποτελέσματα ή μόλις ανακτηθεί μια Web Service ένας χρήστης ίσως θελήσει να αναζητήσει Web Services με παρόμοιες λειτουργίες ή παρόμοια είσοδο/έξοδο.

Οι Sajjanhar et al. [44] εκτέλεσαν αντιστοίχιση ομοιότητας με την αναπαράσταση κειμένων περιγραφής υπηρεσίας όπως περιγράφεται στο τμήμα 1.6.1 και επομένως είχαν τη δυνατότητα να χρησιμοποιήσουν αλγόριθμους με βάση το LSI για να υπολογίσουν την ομοιότητα μεταξύ υπηρεσιών στο επίπεδο περιγραφής. Αυτό επιτυγχάνεται καθώς το LSI έχει παρατηρηθεί ότι είναι αποτελεσματικό στην αντιστοίχιση διανυσμάτων εννοιολογικά παρόμοιων εγγράφων, σε γειτνιάζοντα σημεία ενός χαμηλών- διαστάσεων διανυσματικού χώρου, που επεκτείνεται από τις υπονοούμενες έννοιες (δείτε [3]).

Μια άλλη μορφή επερωτήσεων ομοιότητας βρίσκεται στην υλοποίηση των Schmidt και Parashar [46], όπου επιτρέπεται να καθοριστούν ερωτήματα με τη χρήση wildcard χαρακτήρων. Για παράδειγμα, χρησιμοποιώντας τις συμβάσεις του [46], υποθέτουμε ότι ο χώρος λέξεων κλειδιών των περιγραφών Web Service διαιρείται σε δύο διαστάσεις, τον επιχειρησιακό τομέα και τη λειτουργικότητα της παρεχόμενης υπηρεσίας. Κατά συνέπεια ερωτήματα της μορφής (*travel, booking*), (*stock**, *), επιτρέπονται. Αυτά τα ερωτήματα επιτρέπονται λόγω της αντιστοίχισης των σημείων δεδομένων με την καμπύλη Hilbert. Με αυτήν την αντιστοίχιση, περιοχές που είναι υποψήφιες απαντήσεις για τα ερωτήματα με wildcard, αντιστοιχίζονται σε διαδοχικές περιοχές, τις *συστάδες*, και επιπλέον στα διαδοχικά μονοδιάστατα διαστήματα. Μόλις προσδιοριστούν οι συστάδες, τα ερωτήματα δρομολογούνται στους όμοιους κόμβους στην επικάλυψη, αποθηκεύοντας αυτές τις συστάδες. Ένα ευριστικό αναπτύσσεται επίσης για να περικλύει ορισμένους κόμβους, μειώνοντας έτσι το κόστος επιπλέον μηνυμάτων (overhead) για την ανακοίνωση συστάδων (cluster notification).

Στο Woogole [9] η προσεγγιστική ανάκτηση αντιμετωπίζεται σε διαφορετικό πλαίσιο, ωστόσο οι συγγραφείς υιοθετούν τεχνικές συσταδοποίησης. Η συσταδοποίηση διαμορφώνεται βάσει του σχηματισμού κανόνων συσχετίσεως³ μεταξύ των όρων που αποτελούν τα ονόματα λειτουργιών. Επομένως, οι συστάδες σχηματίζονται μεταξύ συσχετισμένων παραμέτρων με προσοχή στη διασφάλιση υψηλής δια-συστάδας και χαμηλής ομοιότητας διά-συστάδων. Διαισθητικά, όταν όλες οι παράμετροι μέσα σε ένα σύνολο συμπίπτουν συχνά, τότε μπορούν να αντιστοιχούν σε παρόμοιες λειτουργίες.

Οι συγγραφείς υπολογίζουν τα εξής τέσσερα είδη ομοιότητας: ομοιότητα ονόματος παραμέτρου εισόδου/εξόδου, ομοιότητα έννοιας εισόδου/εξόδου, ομοιότητα περιγραφής λειτουργίας και ομοιότητα περιγραφής WS. Ας θεωρήσουμε αρχικά την ομοιότητα ονόματος παραμέτρου I/O. Σε αυτό το είδος ομοιότητας, οι λειτουργίες χαρακτηρίζονται από το σύνολο λέξεων, που περιλαμβάνουν τα ονόματα παραμέτρων. Οι ομοιότητες υπολογίζονται με τη χρήση της μετρικής TF/IDF, δείτε [2]. Ανάλογες διαδικασίες χρησιμοποιούνται στην περιγραφή λειτουργιών και την περιγραφή WS. Για την ομοιότητα εννοιών I/O οι λέξεις υποκαθίστανται από τις συστάδες τους πριν την εφαρμογή των μετρικών TF/IDF. Αυτοί οι τέσσερις τύποι ομοιότητας συνδυάζονται γραμμικά σε ένα ενιαίο συνολικό βαθμό ομοιότητας.

1.7 Θέματα Διασφάλισης Ποιότητας

Η διασφάλιση και παροχή ποιότητας WS (Quality of WS Provisioning-QoWS), είναι ένα ζήτημα το οποίο δεν έχει αντιμετωπιστεί διεξοδικά στην περιοχή των WS έως τις μέρες μας. Επομένως, δεν υπάρχει ούτε σαφής ορισμός ούτε μια συνολικά αποδεκτή αντίληψη για την QoWS. Μια πρώιμη εργασία καθορισμού της QoS [39]

³ Ένας κανόνας συσχέτισης της μορφής $x \rightarrow y$, περιγράφει συχνή συνύπαρξη δύο στοιχείων x και y και το βαθμό στον οποίο η ύπαρξη του x υπονοεί την ύπαρξη του y .

επιχείρησε να αναδείξει τα θέματα που σχετίζονται με τις μη λειτουργικές ιδιότητες υπηρεσιών. Υποστηρίζεται ότι ένα αυξημένο επίπεδο πληροφοριών ιδιοτήτων υπηρεσίας θα διευκόλυνε την ακριβέστερη λήψη αποφάσεων από έναν αιτούντα υπηρεσιών. Πρόσφατες εργασίες παρόλα αυτά, παρουσιάζουν μια προσπάθεια να καθοριστούν αυστηρότερα οι παράμετροι QoWS και οι μέθοδοι παράδοσης. Σχεδιασμοί συστημάτων παρουσιάζονται επίσης στα [53], [54] που παραδίδουν QoWS στις εφαρμογές τελικού χρήστη.

Παράμετρος QoWS	Μονάδα
Καθυστέρηση	Συνήθως milliseconds
Διαθεσιμότητα	Πραγματική τιμή στο [0,1]
Ρυθμαπόδοση	Πραγματική τιμή στο [0,1]
Κόστος εκτέλεσης	Πραγματική τιμή
Φήμη	Πραγματική τιμή στο $[0, r^4]$
Αυθεντικοποίηση	{TRUE, FALSE}
Κρυπτογράφηση	{TRUE, FALSE}
Έλεγχος Πρόσβασης	{TRUE, FALSE}

Πίνακας 3: Οι παράμετροι QoWS και οι αντίστοιχες μονάδες μέτρησης.

1.7.1 Παράμετροι QoWS

Οι Ran [39], Ouzzani [32], Ouzzani και Bouguettaya [31], και Liu et al. [23] έδωσαν έμφαση στις κυρίαρχες παραμέτρους που καθορίζουν την ποιότητα Web Service. Αν και υπάρχει διαφωνία στην καθιέρωση συγκεκριμένων ποιοτικών κριτηρίων, παρουσιάζονται στη συνέχεια οι ποιοτικές παράμετροι που είναι κοινές για την πλειοψηφία της βιβλιογραφίας:

- **Υπολογιστικά Κριτήρια.** Ενδιαφερόμαστε για παραμέτρους όπως:
 - *Καθυστέρηση:* ο μέσος χρόνος στον οποίο μια λειτουργία επιστρέφει αποτελέσματα μετά την κλήση της. Περιλαμβάνει δικτυακές καθυστερήσεις και καθυστερήσεις εκτέλεσης.
 - *Διαθεσιμότητα:* η πιθανότητα να τρέχει μια υπηρεσία
 - *Ρυθμαπόδοση:* το ποσοστό των ολοκληρωμένων web services σε μια χρονική περίοδο.
- **Επιχειρησιακά Κριτήρια.** Αναφέρονται σε κριτήρια που σχετίζονται με επιχειρησιακές πλευρές της κλήσης υπηρεσιών
 - *Κόστος Εκτέλεσης,* μετράει το κόστος της κλήσης υπηρεσιών

⁴ Η τιμή του r καθορίζεται ανά εφαρμογή. Στο Ouzzani [32] έχει τιμή 10, ενώ στο [23] δίνεται με τιμή 5

- ο *Φήμη/Reputation*, μετράει το βαθμό ικανοποίησης, ο οποίος ανατίθεται από εξυπηρετημένους χρήστες.
- **Κριτήρια Ασφαλείας.** Αναφέρονται στους περιορισμούς ασφαλείας που πρέπει να ικανοποιούνται κατά τη διάρκεια του χρόνου εκτέλεσης. Αυτοί περιλαμβάνουν:
 - ο *Αυθεντικοποίηση/Authentication*, υποδεικνύει εάν παρέχεται ταυτοποίηση χρηστών
 - ο *Κρυπτογράφηση*, υποδεικνύει εάν υποστηρίζεται η κρυπτογράφηση μηνυμάτων
 - ο *Έλεγχος Πρόσβασης*, υποδεικνύει εάν η πρόσβαση είναι περιορισμένη σε εγκεκριμένες ομάδες χρηστών.

Ο Ran [39] παρουσιάζει ένα πλαίσιο τροποποίησης του tModel για την ενοποίηση των QoWS χαρακτηριστικών.

Οι παραπάνω παράμετροι δεν είναι όλες μετρήσιμες. Παρόλα αυτά, εάν μια παράμετρος δεν είναι άμεσα μετρήσιμη, μια boolean τιμή μπορεί να αναπαραστήσει την απουσία (false) ή την παρουσία (true) της συγκεκριμένης απαίτησης ποιότητας. Ο Πίνακας 3 συνοψίζει τις ποιοτικές παραμέτρους και δίνει τις κατάλληλες μονάδες μέτρησης για κάθε μια από αυτές.

1.7.2 Ανακάλυψη WS με Απαίτηση Ποιοτικών Χαρακτηριστικών

Η ανάγκη για ανακάλυψη WS με απαίτηση ποιοτικών χαρακτηριστικών (QoS aware) προκύπτει τόσο στην παροχή μιας απλής Web Service όσο και σε σύνθετες Web Service. Στην περίπτωση των σύνθετων WS ένα χαρακτηριστικό σενάριο περιλαμβάνει ένα χρήστη που εκτελεί ένα ερώτημα, το οποίο μεταφράζεται διαφανώς σε ένα σύνολο από WSs που συνήθως χρειάζεται να εκτελεστεί με μια συγκεκριμένη σειρά/ αλληλουχία. Αυτή η ακολουθία εκτέλεσης αναφέρεται επίσης ως *σχέδιο εκτέλεσης* (execution plan) κατ' αναλογία με τις εκτελέσεις σύνθετων ερωτημάτων σε βάσεις δεδομένων. Στη συνέχεια παρουσιάζονται οι μηχανισμοί για την υποστήριξη της εκτέλεσης WS, οι οποίοι είναι συμβατοί με τα κριτήρια QoWS που τίθενται στην προηγούμενη παράγραφο.

Μια WS μπορεί πιθανώς να παρέχεται σε περισσότερα από ένα διακριτά σημεία πρόσβασης (πολλαπλές εναλλακτικές συνδέσεις από έναν πάροχο ή πολλαπλές υλοποιήσεις από διαφορετικούς παρόχους). Επομένως, τόσο στην ακολουθία εκτέλεσης όσο και στην παροχή μιας απλής Web Service, το πρώτο βήμα περιλαμβάνει την κατασκευή ενός συνόλου από όλες τις πιθανές Web Service(s), οι *υποψήφιοι επιλογής* (selection candidates). Στη συνέχεια, στην περίπτωση της ακολουθίας εκτέλεσης, πρέπει να κατασκευαστεί ένας *γράφος εξάρτησης* (dependency graph) [32], για να μοντελοποιήσει τη συγκεκριμένη σειρά των WSs μέσα στο σχέδιο εκτέλεσης, δηλαδή κάθε WS αντιστοιχεί σε ένα μοναδικό κόμβο και μια κατευθυνόμενη ακμή από την WS_i στην WS_j, δηλώνει ότι η *i* πρέπει να εκτελεστεί πριν την *j* σε ένα συγκεκριμένο σχέδιο εκτέλεσης. Προφανώς, είναι επίσης καθοριστικό να οριστεί μια

συνάρτηση ταξινόμησης στους υποψηφίους επιλογής, η οποία να μετρήσει την παρεχόμενη QoWS.

Η πρόκληση εντοπίζεται στην επινόηση μιας συνάρτησης βαθμολόγησης που να συνδυάζει τα κριτήρια QoWS κατά δίκαιο τρόπο προκειμένου να παραχθεί ένα ενιαίο αποτέλεσμα για κάθε Web Service ή σχέδιο εκτέλεσης, αντίστοιχα. Οι Liu et al. στην εργασία [23] παρουσιάζουν μία προσέγγιση όπου αναπαριστούν κάθε WS ως ένα διάνυσμα m διαστάσεων, όπου m ο αριθμός των παραμέτρων που λαμβάνονται υπόψη. Στη συνέχεια οι υποψήφιοι επιλογής αναπαριστούνται ως μια μήτρα, $Q: n \times m$, όπου n είναι ο αριθμός των WSs με παρόμοια λειτουργικότητα. Αυτό το μητρώο υφίσταται διαδοχικές κανονικοποιήσεις με σκοπό να φέρουν τα διαφορετικά κριτήρια στο ίδιο επίπεδο και να συσταδοποιήσουν τα κριτήρια σε ομάδες. Επομένως, έπειτα από τις απαραίτητες κανονικοποιήσεις, τα κριτήρια QoWS ομαδοποιούνται και κανονικοποιούνται, και ένας μοναδικός βαθμός QoWS για την Web Service ws_i προκύπτει από τη σχέση

$$QoWS(ws_i) = q_i \cdot w^T \quad (1)$$

όπου η παραπάνω σχέση δείχνει το εσωτερικό γινόμενο μεταξύ του ποιοτικού διανύσματος q_i για την υπηρεσία ws_i και ενός χρήστη που παρείχε το διάνυσμα w , από βάρη σημασίας για κάθε ποιοτική κατηγορία. Προφανώς, θα επιλεγεί η WS που θα επιτύχει τον υψηλότερο βαθμό QoWS.

Όταν πρόκειται να επιλεγεί το βέλτιστο σχέδιο εκτέλεσης για ένα αίτημα υπηρεσίας η κατάσταση είναι η ακόλουθη: έχουμε τα πολλαπλά σχέδια εκτέλεσης, κάθε ένα από τα οποία συνοδεύεται με ένα βαθμό QoWS, ο οποίος είναι ένας γραμμικός συνδυασμός των βαθμολογήσεων για τις ξεχωριστές WSs. Αυτό που ζητείται είναι η επιλογή του σχεδίου εκτέλεσης που θα μεγιστοποιήσει τη συνολική ποιότητα υπηρεσίας. Είναι σχεδόν προφανές ότι το παραπάνω πρόβλημα καταλήγει τελικά σε γραμμική βελτιστοποίηση. Αυτή η περίπτωση ισχύει στα [31, 32]. Για κάθε σχέδιο εκτέλεσης p , διακρίνουμε τις αρνητικές παραμέτρους QoWS τις οποίες δηλώνουμε με neg, και τις θετικές QoWS παραμέτρους που τις οποίες δηλώνουμε με pos. Έπειτα για κάθε σχέδιο p , η σχέση (2) παρέχει τη συνολικά παρεχόμενη QoWS

$$F(p) = \left(\sum_{Q_i \in neg} \frac{Q_i^{\max} - Q_i}{Q_i^{\max} - Q_i^{\min}} + \sum_{Q_i \in pos} \frac{Q_i - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}} \right) \quad (2)$$

όπου το Q_i δηλώνει τη μετρούμενη τιμή της παραμέτρου και $Q_i^{\max}$ ($Q_i^{\min}$) είναι η μέγιστη (ελάχιστη) τιμή όλων των διαθέσιμων επιλογών για την υπηρεσία i . Πρέπει να τονίσουμε ότι η τιμή για κάθε Q_i προστίθεται κατάλληλα πάνω από όλες τις συμμετέχουσες υπηρεσίες στο p . Η βαθμολογία στο (2) μοιάζει με τη λογική κανονικοποίησης του [23]. Η συνολική αντικειμενική συνάρτηση ορίζεται στο (3).

$$C = \left(\sum_{Q_i \in neg} w_i \cdot \frac{Q_i^{\max} - Q_i}{Q_i^{\max} - Q_i^{\min}} + \sum_{Q_i \in pos} w_i \cdot \frac{Q_i - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}} \right) \quad (3)$$

όπου W_i είναι το βάρος που παρέχεται από ένα χρήστη ο οποίος δείχνει προτίμηση σε συγκεκριμένες παραμέτρους QoS. Η συνάρτηση κόστους μπορεί να συντονιστεί περαιτέρω για να περιλάβει τον βαθμό ταιριάσματος (*matching degree*), ο βαθμός δηλαδή στον οποίο ένα σχέδιο εκτέλεσης ικανοποιεί το ερώτημα χρήστη, και την τιμή QoS, ένα αποτέλεσμα που υπολογίζεται για κάθε WS λειτουργία για να μετρήσει την απόκλιση της από την αρχικά υποσχόμενη ποιότητα.

Ο συγγραφέας του [32], σημειώνει ότι εάν κανείς θεωρούσε ολόκληρο το χώρο λύσης για τη βέλτιστη λύση, θα απαιτούταν εκθετικός χρόνος για το πλήθος των συμμετεχουσών διαδικασιών. Επομένως, παρουσιάζονται αρκετά ευριστικά που μπορούν να υπολογίσουν τη βέλτιστη λύση με αποδοτικό τρόπο.

Οι Ταο και Lin [54] παρουσίασαν ένα διαφορετικό αλγόριθμο για την επιλογή του βέλτιστου σχεδίου εκτέλεσης. Έστω ότι ενδιαφερόμαστε για ένα σχέδιο εκτέλεσης που αποτελείται από μια ακολουθία k διαδοχικά εκτελούμενων Web Services $\{S_1, S_2, \dots, S_k\}$, ή ένα “pipeline” όπως καλείται στο [54]. Για κάθε υπηρεσία καθορίζεται ένα σύνολο υποψήφιων σημείων πρόσβασης, και μια συνάρτηση χρησιμότητας (utility function). Η συνάρτηση χρησιμότητας καλύπτει τους παράγοντες κέρδους και το κόστος για κάθε υποψήφιο επιλογής. Όσο υψηλότερη είναι η συνάρτηση χρησιμότητας τόσο υψηλότερη είναι και η QoS που παρέχεται. Εάν είναι επιθυμητό η επιλεγμένη ακολουθία εκτέλεσης να έχει συνολικό χρόνο εκτέλεσης κάτω από ένα κατώτατο όριο R , τότε ολόκληρο το πρόβλημα μπορεί να διατυπωθεί ως *Multiple Choice Knapsack* (MCK). Στο MCK έχουμε διάφορες κλάσεις, κάθε μια με ένα βάρος και ένα κέρδος, και ένα knapsack με περιορισμένη ικανότητα. Ο στόχος είναι να συλλεχθεί ένα στοιχείο από κάθε κλάση, έτσι ώστε το συνολικό κέρδος να μεγιστοποιηθεί, και παρόλα αυτά το συνολικό βάρος να μην υπερβαίνει την χωρητικότητα του knapsack. Η μείωση της επιλογής της ακολουθίας εκτέλεσης στο MCK μπορεί να γίνει ως εξής:

- Κάθε βήμα εκτέλεσης αντιστοιχεί σε μια κλάση στο MCK
- Οι υποψήφιοι επιλογής για κάθε υπηρεσία είναι τα στοιχεία στις κλάσεις του MCK, όπου ο χρόνος απόκρισης κάθε υποψηφίου είναι το βάρος αυτού του στοιχείου και η συνάρτηση χρησιμότητας είναι το κέρδος της επιλογής στοιχείου
- Ο στόχος είναι να μεγιστοποιηθεί η συνολική χρησιμότητα κρατώντας το συνολικό χρόνο απόκρισης κάτω από R , την knapsack capacity.

Οι Ταο και Lin [54], δίνουν έναν αλγόριθμο δυναμικού προγραμματισμού και ορισμένες παραλλαγές για να λύσουν το MCK, και συνεπώς να λύσουν αποτελεσματικά την επιλογή ακολουθίας υπηρεσιών. Η επέκταση της pipeline επιλογής σε μη γραμμικές συνθέσεις Web Services, δηλ σε σύνολα WS σε μορφή κατευθυνόμενου άκυκλου γράφου, πραγματοποιείται μετά από απαρίθμηση όλων των διακριτών μονοπατιών μήκους k , και την εκτέλεση ενός αλγορίθμου σύντομου μονοπατιού για να αποφασίσουν το βέλτιστο μονοπάτι εκτέλεσης.

1.7.3 Παράδοση QoWS

Μια ορθογώνια άποψη σε σχέση με την ανακάλυψη QoWS είναι η παράδοση QoWS. Μια σειρά από σχετικές δημοσιεύσεις [5], [53] παρουσιάζουν αυτό που ονομάζεται ενδιάμεσος ποιότητας υπηρεσίας (*QoS Broker*). Ο QoS Broker υλοποιείται ως ξεχωριστό λειτουργικό υποσύστημα, μεταξύ του καταλόγου UDDI και του τελικού χρήστη. Αποτελείται από μια βάση δεδομένων που αποθηκεύει πληροφορίες QoS των υπηρεσιών, ένα Διαχειριστή Πληροφοριών Ποιότητας Υπηρεσίας (QoS Information Manager) που διαχειρίζεται τις αποθηκευμένες πληροφορίες, έναν Διαχειριστή Διαπραγμάτευσης Ποιότητας Υπηρεσίας (QoS Negotiation Manager) που εκτελεί το έργο της αναζήτησης κατάλληλων Web Services, για να εξασφαλίσει ότι δεσμεύονται αρκετοί πόροι, να επιβεβαιώσει και να επιτρέψει τη δέσμευση, και τον Αναλυτή Ποιότητας Υπηρεσίας (QoS Analyzer) που αλληλεπιδρά με την DB για να υπολογίσει και να αποθηκεύσει τις QoS βαθμολογίες για κάθε παρεχόμενη WSs. Μια γενική επισκόπηση ενός QoS broker φαίνεται στο σχήμα 6.

Σχήμα 6: Η αρχιτεκτονική QoWS broker
δανεισμένη από τους Chen et al. [5]

Ο QoS broker χρειάζεται να χρησιμοποιεί δύο είδη αλγορίθμων: ένα για την ανακάλυψη κατάλληλων υπηρεσιών ως προς τις μη λειτουργικές προδιαγραφές ποιότητα και ένα για να διασφαλίσει τη δέσμευση πόρων. Οι αλγόριθμοι του πρώτου είδους εμπίπτουν στην κατηγορία αλγορίθμων που παρουσιάστηκαν στην προηγούμενη παράγραφο, παρόλα αυτά δεν παρουσιάζονται οι συγκεκριμένες επιλογές των [5], [53]. Οι συγγραφείς του [53] έχουν αναπτύξει αρκετές στρατηγικές δέσμευσης για ομογενή δέσμευση πόρων (HQ) και μη ομογενή δέσμευση (RQ), δηλαδή ευνοϊκή αντιμετώπιση ορισμένων αιτημάτων σε σχέση με άλλα.

1.8 Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις

Οι μηχανισμοί ανακάλυψης Web Service που παρουσιάστηκαν, προσπαθούν να ενισχύσουν την απόδοση κατά τη διαδικασία της αντιστοίχισης και δέσμευσης WS με βάση τόσο τα λειτουργικά όσο και μη χαρακτηριστικά της. Μεταξύ άλλων, οι μηχανισμοί ανακάλυψης θα πρέπει να επιτρέπουν την αναζήτηση και την αξιολόγηση, βασισμένοι αποκλειστικά στην εξωτερική άποψη (interface) μιας Web Service. Η αξιολόγηση είναι βασισμένη στη λήψη αποφάσεων πολλαπλών κριτηρίων [20].

Επιπλέον, ιδιαίτερη σημασία πρέπει να δοθεί στον καθορισμό των μετρικών QoWS. Αυτό είναι σημαντικό δεδομένου ότι θα επιτρέψει την περεταίρω βελτίωση των μηχανισμών ανακάλυψης WS προκειμένου να επιτευχθούν τα κατώτερα επίπεδα απόδοσης, ασφάλειας και διαθεσιμότητας στα αποτελέσματα αντιστοίχισης και δέσμευσης. Θα είναι τότε δυνατή η υποστήριξη της εξισορρόπησης φορτίου στην παράδοση WS αρχίζοντας από τη στιγμή της επιλογής. Αυτό μπορεί να βελτιώσει σημαντικά την απόδοση, ειδικά στις περιπτώσεις όπου συναντιέται υπερβολικά αυξημένος φόρτος εργασίας.

Περιγραφή Περίπτωσης/Προβλήματος	Τεχνική Ανακάλυψης
Ανοχή σε Single point failure– Εξισορρόπηση φορτίου	[41] UDDI Federation
Χαμηλή διαθεσιμότητα ή σύντομη εμφάνιση κόμβων που παρέχουν την υπηρεσία	[22] Ομότιμα Δίκτυα, [31] Αποδοτική Πρόσβαση
Χειροκίνητη αναζήτηση του προγραμματιστή χρησιμοποιώντας απλή διεπαφή	[29] UDDI, [61] Ευέλικτη Ανάκτηση
Δυναμική/ Προσαρμοστική ανακάλυψη	[24] Δυναμική επιλογή
Υποστήριξη Grid	[27] GRID, [25] NIPPERS
Ενοποίηση σημασιολογικών δυνατοτήτων	[45] P2P
Πλήρης βιομηχανική υποστήριξη κατά την εγκατάσταση, ανάπτυξη και υποστήριξη	[29] UDDI
Υποστήριξη ποιότητας υπηρεσίας - QoS	[5] & [53] QoS broker, [23] QoS framework
Ενσωμάτωση γνωστών τεχνικών Ανάκτησης Πληροφορίας	[44] Αντιστοίχιση Διανυσματικού Χώρου
QoS aware σχεδιασμός εκτέλεσης	[32], Γραμμική βελτιστοποίηση

Πίνακας 4: Περιπτώσεις/Προβλήματα και τεχνικές ανακάλυψης που τα αντιμετωπίζουν.

Η εισαγωγή των μπλε σελίδων σύμφωνα με το [9] μπορεί να θεωρηθεί ως το επόμενο βήμα προς αυτήν την κατεύθυνση, αν λάβουμε υπόψη την αυξανόμενη ανάγκη για την προσπάθεια πραγματοποίησης του Σημασιολογικού Ιστού. Όταν κατευθυνόμαστε προς τα επάνω στον άξονα αποκεντριοποιημένων λύσεων στο σχήμα 4, οι λύσεις ομότιμων δικτύων [22], [25], [31], [46] έχουν δείξει την καλύτερη απόδοση έως τώρα. Επιπλέον, στο σχήμα 5 οι περιπτώσεις των [25] και [46] παρέχουν ευρεία επεκτασιμότητα και επιλογές για εξισορρόπηση φόρτου δικτύου και σταθερότητα υπηρεσίας τα οποία είναι ένα καθαρά ανταγωνιστικό πλεονέκτημα σε σχέση με την απλοϊκή βασισμένη σε λέξεις κλειδιά αναζήτηση του UDDI ακόμα και στην κατανεμημένη έκδοσή της. Επιπλέον, από την άποψη της ευελιξίας το μοντέλο

διανυσματικού χώρου παρέχει μια σταθερή βάση για προσεγγίσεις καλύτερες από το Συνολικά, μια service oriented λύση ανακάλυψης WS μπορεί να υποστηριχθεί από τους απαραίτητους κοινοτικούς μηχανισμούς για την επόμενη γενιά αποτελεσματικής αναζήτησης για WSs.

Μια συμπερασματική παρουσίαση των προσεγγίσεων που θα προσπαθούσε να παρέχει μια γενική ταξινόμηση των μηχανισμών που συζητήθηκαν ίσως παραπλανούσε το χρήστη προς ενδεχομένως λανθασμένες υποθέσεις. Εντούτοις, παρακάτω παρουσιάζεται μια τελευταία κατηγοριοποίηση (Πίνακας 4), όπου οι τεχνικές ομαδοποιούνται με βάση τις ιδιαίτερες περιπτώσεις και τα προβλήματα που αντιμετωπίζουν.

Η έρευνα για τους μηχανισμούς ανακάλυψης πρέπει να φθάσει σε αποδοτικές δομές που να είναι εφαρμόσιμες όχι μόνο σε WSs, αλλά και σε διαδικτυακά ή άλλα τμήματα λογισμικού γενικά. Αυτό θα απαιτούσε ορισμένες πρόσθετες προδιαγραφές σχετικά με την πλατφόρμα, τις απαιτήσεις συστήματος, το είδος επαναχρησιμοποίησης, το είδος κώδικα, και το εύρος εφαρμογής [33]. Αυτές οι προδιαγραφές θα μπορούσαν να προστεθούν ενδεχομένως μέσα στις πληροφορίες καταλόγων στην ενότητα των μεταδεδομένων διαχειριστικού χαρακτήρα. Συνεπώς, θα μπορούσε τελικά να επιτευχθεί ακόμη και μια ενοποιημένη προδιαγραφή των τμημάτων λογισμικού (που θα μπορούσε να είναι η βάση για τους μελλοντικούς καταλόγους component και τα CASE εργαλεία).

Ολοκληρώνοντας αυτήν την ενότητα, διάφορες προσεγγίσεις έχουν προταθεί χρησιμοποιώντας διαφορετικές προσεγγίσεις. Εκτός από το UDDI, ανερχόμενες αποκεντριοποιημένες απόψεις όπως οι λύσεις ομότιμων δικτύων είναι αρκετά ελπιδοφόρες. Είναι επίσης δυνατή η αναβάθμιση των μοντέλων δεδομένων με τη διαμόρφωση τεχνικών και οντολογιών Ανάκτησης Πληροφορίας, ειδικά λαμβάνοντας υπόψη ότι η έρευνα για τον Σημασιολογικό Ιστό είναι ιδιαίτερα δημοφιλής. Οι μηχανισμοί ανακάλυψης WS έχουν έναν ακόμα σημαντικότερο ρόλο από την αναζήτηση στον Ιστό, καθώς διευκολύνουν την ανάγκη για συνεργασία μεταξύ των επιχειρησιακών διαδικασιών και των καταναλωτών, πάνω από ευρέως αποδεκτά πρότυπα Ιστού.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ 1

1. Austin, D., Barbir, A., Ferris, C. and Garg, S. (eds.): Web Service Architecture Requirements. W3C Working Group Notes (2004) <http://www.w3.org/TR/wsa-reqs>
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, Addison-Wesley, (1999)
3. Berry, M. W., Dumais, S. T. and O'Brien, G. W.: Using linear algebra for intelligent information retrieval. SIAM Review, 37(4). (1995) 573-595
4. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion M., Ferris, C. and Orchard, D. (eds.): Web Services Architecture. W3C WG Note. <http://www.w3.org/TR/ws-arch/>
5. Chen, H., Tao, Y. and Lin, K.-J. QCWS: An Implementation of QoS-Capable Multimedia Web Services. In proc. of the IEEE Fifth International Symposium on Multimedia Software Engineering (ISMSE'03), 2003.
6. DAML-S Coalition: DAML-S: Web Service Description for the Semantic Web, Proc. 1st Int'l Semantic Web Conf. (ISWC 02), 2002.
7. Decker, K., Sycara, K., Williamson, M.: Middle-Agents for the Internet. Proc. 15th IJCAI. Nagoya, Japan. (1997) 578-583
8. Dogac, A., Laleci, G., Kabak, Y. and Cingil, I.: Exploiting Web Service Semantics: Taxonomies vs. Ontologies. In IEEE Data Engineering Bulletin, Vol. 25, No. 4, December 2002.
9. Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J., Similarity Search for Web Services. In proc. of the 30th International Conference on Very Large Databases (VLDB2004), pp. 372-383, 2004.
10. Elgedawy, I., Tari, Z. and Winikoff, M.: Scenario Matching Using Functional Substitutability in Web Services. In X. Zhou et al. (Eds.): WISE 2004, LNCS 3306, pp. 59–65, 2004.
11. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid. Enabling scalable virtual organizations. International Journal of Supercomputer Applications. (2001)
12. Gaede, V., Gunther, O.: Multidimensional Access Methods. ACM CSUR, 30(2). (1998)
13. Gong L., "JXTA: A Network Programming Environment". IEEE Internet Computing, (5)3:88-95, May/June 2001.
14. Greiner, U. and Rahm, E.: Quality-Oriented Handling of Exceptions in Web-Service-Based Cooperative Processes, Proc. of EAI-Workshop 2004 - Enterprise Application Integration, Oldenburg. GITO-Verlag, Berlin, pp. 11-18, Feb. 2004.
15. Hu, Z.: Using Ontology to Bind Web Services to the Data Model of Automation Systems Revised Papers from the NODE 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems. (2002) 154 – 168.
16. HTTP, Hypertext transfer protocol W3C (2000) URL <http://www.w3.org/Protocols/>
17. Jannach, D., Leopold, K., Timmerer, C. and Hellwagner, H.: Toward Semantic Web Services for Multimedia Adaptation. In X. Zhou et al. (Eds.): WISE 2004, LNCS 3306, pp. 641–652, 2004.
18. Jeckle, M., Zengler, B.: Active UDDI - an Extension to UDDI for Dynamic and Fault-Tolerant Service Invocation, Revised Papers from the NODE 2002 Web and Database-Related Workshops on Web, Web-Services and Database Systems , LNCS. (2003) 91-99
19. Jinghai, R. and Xiaomeng, S.: A Survey of Automated Web Service Composition Methods. In J. Cardoso and A. Sheth (Eds.): SWSWPC 2004, LNCS 3387, pp. 43–54, 2004.
20. Konito, J.: A Case Study in Applying a Systematic Method for COTS Selection. In: Proc., 18th Int. Conf. on Soft. Eng. (ICSE). IEEE Computer Society Press (1996) 201–209
21. Lacey, P.: Uddi & Dynamic Web Service Discovery. <http://www.ddj.com/articles/2004>
22. Li, Y., Zou, F., Wu, Z., Ma, F.: PWSD: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network, In Proc. APWeb04, LNCS 3007. (2004) 291-300
23. Liu, Y., Ngu, A., and Zeng, L. QoS Computation and Policing in Dynamic Web Service Selection. In proc. World Wide Web Conference (WWW2004), pp. 66 – 73, 2004.

24. Makris, Ch., Panagis, Y., Sakkopoulos, E., Tsakalidis, A.: Efficient search algorithm for large scale web service data, Research Academic Computer Technology Institute Technical Report, No. CTI TR 2004/09/01.
25. Makris, Ch., Sakkopoulos, E., Sioutas, S., Triantafillou, P., Tsakalidis, A., Vassiliadis, B.: NIPPERS: Network of InterPolated PeERS for Web Service Discovery. In the Proceedings of the 2005 IEEE International Conference on Information Technology: Coding & Computing (IEEE ITCC 2005), Track Next Generation Web and Grid Systems, USA. (2005)
26. Montebello, M., C. Abela, C.: DAML Enabled Web Services and Agents in the Semantic Web. Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services and Database Systems , LNCS. (2003) 46 – 58
27. Moreau, L., Avila-Rosas, A., Dialani, V., Miles, S. and Liu, X. Agents for the Grid: A Comparison with Web Services (part II: Service Discovery). In Proceedings of Workshop on Challenges in Open Agent Systems , Italy. (2002) 52-56
28. O’Sullivan, J., Edmond, D. and Ter Hofstede, A.: What’s in a Service? Towards Accurate Description of Non-Functional Service Properties. In Distributed and Parallel Databases, 12, pp. 117–133, 2002 Kluwer Academic Publishers.
29. OASIS UDDI Specifications TC - Committee Specifications <http://www.oasis-open.org/committees/uddi-spec/doc/tcpspecs.htm>
30. Orth, Th. (2002). The Web Services Framework: A Survey of WSDL, SOAP and UDDI. Master Thesis, TU Wien, Wien, Austria.
31. Ouzzani M., Bouguettaya A.: Efficient Access to Web Services. IEEE Internet Computing, March/April (2004) 34-44
32. Ouzzani, M., Efficient Delivery of Web Services, PhD Thesis, Virginia Tech University. (2004)
33. Overhage, S.: On Specifying Web Services Using UDDI Improvements. 3rd Annual International Conference on Object-Oriented and Internet-based Technologies, Concepts, and Applications for a Networked World Net.ObjectDays, Germany (2002)
34. OWL-S Specifications, <http://www.daml.org/services/owl-s/1.0/>
35. Paolucci, M., Kawamura, T., Payne, T. R., Sycara, K.: Semantic Matching of Web Services Capabilities. In Proceedings of the 1st Int. Semantic Web Conference (ISWC2002). (2002)
36. Patel, Ch., Supekar, K. and Lee, Y.: A QoS Oriented Framework for Adaptive Management of Web Service based Workflows. In the Proceedings of Database and Expert Systems 2003 conference (DEXA), LNCS, Springer-Verlag, 2003, pp. 826 – 835.
37. Patil, A., Oundhakar, S., Sheth, A. and Verma, K.: METEOR-S Web service Annotation Framework. In the Proceedings of the World Wide Web Conference, July 2004.
38. Rajasekaran, P., Miller, J., Verma, K. and Sheth, A.: Enhancing Web Services Description and Discovery to Facilitate Composition. In J. Cardoso and A. Sheth (Eds.): SWSWPC 2004, LNCS 3387, 2004.
39. Ran S.: A Model for Web Services Discovery with QoS, ACM, ACM SIGecom Exchanges, Volume 4 , Issue 1 Spring. (2003) 1 - 10.
40. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. Proceedings of ACM SIGCOMM’01, San Diego, September (2001)
41. Rompothong, P., Senivongse, Tw.: A Query Federation of UDDI Registries, ISICT (2003)
42. Rowstron, A., Druschel, P.: Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems. Proc. of the 18th IFIP/ACM Middleware, Germany. (2001)
43. rWhois RFC, <http://rfc.net/rfc2167.html>
44. Sajjanhar, A., Hou, J., Zhang, Y.: Algorithm for Web Services Matching, In Proc. APWeb 2004, Lecture notes in Computer Science 3007, Springer Verlag, (2004) 665-670
45. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A scalable and ontology-based P2P infrastructure for semantic web services. Proc. 2nd Int. Conf. P2P’02, Sweden. (2002) 104–111.
46. Schmidt, C., Parashar, M. A.: Peer-to-Peer Approach to Web Service Discovery. World Wide Web: Internet and Web Information Systems, 7. (2004) 211-229

47. Sivashanmugam, K., Verma, K., Mulye, R., Zhong, Z., and Sheth, A.: Speed-R: Semantic P2P environment for diverse Web Service registries, <http://webster.cs.uga.edu/~mulye/SemEnt/Speed-R.html>. (2004)
48. Sivashanmugam, K., Verma, K., Sheth, A. and Miller, J.: Adding Semantics to Web Services Standards. In the Proceeding of the 1st International Conference on Web Services (ICWS'03), Las Vegas, Nevada, pp. 395-401, June 2003.
49. SOAP. W3C SOAP Version 1.2 Part 1: Messaging Framework (W3C). Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., & Frystyk Nielsen, H., (eds.). (2003) Retrieved January 3, 2005, from W3C URL: <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
50. Stoica, I., Morris, R., Karger, D., Kaashoek M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for Internet applications. ACM SIGCOMM'01, San Diego, (2001)
51. Sycara K., Dynamic Discovery, Invocation and Composition of Semantic Web Services, Third Hellenic Conference on AI, SETN 2004. LNCS 3025, Springer (2004) 3-12
52. Sycara, K., Klusch, M., Widoff, S., Lu, J.: Dynamic service matchmaking among agents in open information environments, ACM SIGMOD Record, v.28 n.1. (1999) 47-53
53. Tao, Y. and Lin, K.-J. The design of QoS Broker Algorithms for QoS-capable Web Services. In proc. of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), pp. 17-24, 2004.
54. Tao, Y., and Lin, K.-J. Service Selection Algorithms for Web Services with end-to-end QoS Constraints. In Proceedings of the IEEE International Conference on E-Commerce Technology, CEC2004, pp. 129-136, 2004.
55. Whois++ RFC, <http://rfc.net/rfc1835.html>
56. WSArchitecture, Web Services Architecture (W3C). Booth, D. et al. (eds.), (2004). Retrieved January 3, 2005, from W3C URL: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>
57. WSChoreography, Web Services Choreography Description Language Version 1.0 (W3C). 2004 Kavantzaz, N., Burdett, D., Ritzinger, G., Fletcher, T., & Lafon, Y. (eds.). Retrieved January 3, 2005, URL: <http://www.w3.org/TR/ws-cdl-10/>
58. WSDL. W3C Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language (W3C). Chinnici, R., Gudgin, M., Moreau, J.-J., Schlimmer, J., & Weerawarana, S., (eds.). (2004) Retrieved January 3, 2005 URL: <http://www.w3.org/TR/wsd120.htm>
59. Yu, H., Mine, T. and Amamiya, M.: Towards Automatic Discovery of Web Portals – Semantic Description of Web Portal Capabilities. In J. Cardoso and A. Sheth (Eds.): SWSWPC 2004, LNCS 3387, San Diego, California, USA, July 6, 2004.
60. Zhang, J., Chung, J.Y., Chang, C.K. and Kim, S.: WS-Net: A Petri-net Based Specification Model for Web Services. In the Proceedings of ICWS 2004, pp. 420-427.
61. Zhuge, H. and Liu, J.: Flexible retrieval of Web Services. The Journal of Systems and Software, 70, 2004, pp. 107–116.

2 ΔΥΝΑΜΙΚΗ ΔΙΑΧΕΙΡΙΣΗ ΚΑΙ ΑΝΑΚΑΛΥΨΗ WEB SERVICES ΜΕ ΒΑΣΗ ΛΕΙΤΟΥΡΓΙΚΑ ΚΑΙ ΠΟΙΟΤΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Πρόλογος – Σύνοψη Κεφαλαίου

Αρχιετές προσπάθειες έχουν γίνει σχετικά με την αναζήτηση και την εύρεση μιας Web Service με βάση λέξεις κλειδιά και περιγραφές. Παρόλα αυτά, πρωτότυπη είναι η προσπάθεια για αποδοτική επιλογή Web Service που θα λαμβάνει υπόψη, εκτός από τα λειτουργικά χαρακτηριστικά, τους παράγοντες ποιότητας και απόδοσης τη στιγμή της κατανάλωσης-σύνδεσης με μια Web Service σε δίκτυο καλύτερης προσπάθειας (best-effort) όπως το Διαδίκτυο. Τέτοιοι παράγοντες μπορούν να περιλαμβάνουν το χρόνο εκτέλεσης της υπηρεσίας σε συγκεκριμένες συνθήκες λειτουργίας του εξυπηρετητή-παρόχου (διαθέσιμη μνήμη, διαθέσιμότητα υπολογιστικής ισχύος κλπ) καθώς και το χρόνο απόκρισης. Η δική μας προτεινόμενη προσέγγιση επιλέγει προσαρμοστικά την αποδοτικότερη WS μεταξύ των εναλλακτικών λύσεων με βάση πραγματικού χρόνου, βελτιστοποιημένες και μετρήσιμες παραμέτρους λειτουργίας του παροχέα.

Το παρόν κεφάλαιο έχει την ακόλουθη δομή: Στην ενότητα 2.1 παρουσιάζεται η εισαγωγή και στην ενότητα 2.2 το πρόβλημα που αντιμετωπίστηκε. Στην ενότητα 2.3 παρουσιάζονται σύντομα ορισμένες σχετικές εργασίες που επεκτείνουν το κεφάλαιο 1 της διατριβής που παρουσιάστηκε ήδη. Στην ενότητα 2.4 παρουσιάζεται μια σύντομη περιγραφή της αλγοριθμικής προσέγγισης μας που στόχο έχει να αποτελέσει έναν οδικό χάρτη στον οποίο μπορεί να αναφέρεται ο αναγνώστης για την καλύτερη παρακολούθηση της περαιτέρω ανάλυσης. Στην ενότητα 2.5 παρουσιάζεται το πρώτο βήμα της προσέγγισης μας με χρήση υπολογιστικής γεωμετρίας. Στη συνέχεια στη 2.6 περιγράφεται το κύριο μέρος της αποδοτικής και προσαρμοστικής επιλογής με βάση ποιοτικές απαιτήσεις. Η ενότητα 2.7 περιλαμβάνει την αξιολόγηση σε επίπεδο πολυπλοκότητας. Στη συνέχεια, η ενότητα 2.8 περιλαμβάνει στοιχεία υλοποίησης, τη διαδικασία πειραματικής αξιολόγησης και τα αποτελέσματά της. Τέλος, το δεύτερο κεφάλαιο ολοκληρώνεται με την ενότητα 2.9 όπου δίνονται μελλοντικές ερευνητικές κατευθύνσεις.

2.1 Εισαγωγή

Οι Web Services (WS) παρέχουν ένα κοινώς αποδεκτό πλαίσιο για αλληλεπίδραση μεταξύ υπηρεσιών, με βάση τα υπάρχοντα δικτυακά πρωτόκολλα και τα ανοιχτά πρότυπα που βασίζονται στην XML. Το πλαίσιο των Web Services είναι ένα από τα νεότερα μέλη στον τομέα της εφαρμοσμένης μηχανικής Ιστού (web engineering) [4] και αποτελεί την πλέον υποσχόμενη πλατφόρμα για δικτυοκεντρικά πληροφοριακά περιβάλλοντα. Οι WS περιλαμβάνουν μια σειρά από τεχνολογίες επικοινωνίας και κωδικοποίησης δεδομένων, οι κυριότερες εκ των οποίων είναι: το πρωτόκολλο επικοινωνίας, η περιγραφή υπηρεσίας, και η ανακάλυψη υπηρεσίας. Ήδη για τις αντίστοιχες τεχνολογίες έχουν αναπτυχθεί, και ακόμη εξελίσσονται, πολλές προδιαγραφές όπως το πρωτόκολλο επικοινωνίας SOAP [18], η γλώσσα περιγραφής Web Services (WSDL) [19] και το πρότυπο για τον κατάλογο εγγραφής (registry) και τις μεθόδους ανακάλυψης Universal Description, Discovery and Integration

(UDDI) [17]. Η περιοχή της ανακάλυψης WS είναι η πλέον ενδιαφέρουσα, δεδομένου ότι η ζήτηση για κατανάλωση WS αυξάνεται συνεχώς παράλληλα με την εξάπλωση και την εισδοχή νέων δικτυοκεντρικών πληροφοριακών συστημάτων. Υπάρχει κατά συνέπεια μεγάλη επιστημονική προσπάθεια ώστε να ανακαλυφθούν οι αποδοτικότεροι αλγόριθμοι ανακάλυψης αλλά και να προσδιοριστούν οι μέθοδοι εκείνες που είναι οι καταλληλότερες ανά περίπτωση.

Η συμπεριφορά και η χρήση των WS επηρεάζεται από πολλούς παράγοντες. Οι Web Services έχουν θεωρητικά σχεδιαστεί για να υποστηρίζουν τους παρακάτω τέσσερις τύπους συμπεριφορών:

1. αίτημα-απόκριση (request-response),
2. μόνο αίτημα (request-only),
3. μόνο απόκριση (response-only) και
4. ενημέρωση κατ' απαίτηση-(solicit-response).

Ωστόσο, η μεγάλη πλειοψηφία των WS, όπως και οι διαδικασίες επιχειρησιακής λογικής, είναι του τύπου αιτήματος-απόκρισης και μόνο ένας περιορισμένος αριθμός ακολουθεί τη συμπεριφορά μονόδρομης αποστολής μηνυμάτων⁵ [2].

Το παρόν κεφάλαιο εξετάζει τα προβλήματα που εμφανίζονται κυρίως σε στην πρώτη και μεγαλύτερη κατηγορία, όπου ο χρόνος και το κόστος διεκπεραίωσης μια ερώτησης και αποστολής της απάντησης από μία Web Service παίζει σημαντικό ρόλο τόσο σε χρονικό όσο και οικονομικό επίπεδο (time and cost sensitive) για τον τελικό αποδέκτη. Διαισθητικά, ο σημαντικότερος παράγοντας για τον τελικό χρήστη έχει να κάνει με τον απαραίτητο συνολικά χρόνο που χρειάζεται μια Web Service για να παράγει και να παραδώσει τα αποτελέσματά της, ο χρόνος εκτέλεσης δηλαδή της Web Service. Υπάρχουν περιπτώσεις όπου οι Web Services πρέπει να εξετάσουν πολύ μεγάλο αριθμό δεδομένων ή μπορεί να χρειαστούν μεγάλο χρονικό διάστημα εκτέλεσης για άλλους λόγους (χαμηλή διαθεσιμότητα υπολογιστικών πόρων ή δικτύου της υποδομής) πριν επιστρέψουν τα αποτελέσματα στον πελάτη/χρήστη. Η διάρκεια εκτέλεσης των Web Services ποικίλει ανάλογα με την τρέχουσα κατάσταση της υποδομής που την υποστηρίζει και εξαρτάται από τους διαθέσιμους πόρους. Κατά συνέπεια ο πελάτης μπορεί να περιμένει ασύγχρονα την απάντηση μιας Web Service, η οποία να καθυστερήσει τόσο πολύ να απαντήσει, ώστε να αποφασίσει να ζητήσει τα αποτελέσματα από μια άλλη διαθέσιμη Web Service που τον καλύπτει λειτουργικά αλλά βρίσκεται σε μια διαφορετική θέση με υψηλότερη διαθεσιμότητα και επομένως συντομότερο πιθανό χρόνο ανταπόκρισης.

Τέτοιες Web Services συναντιούνται ιδιαίτερα στα επιχειρησιακά περιβάλλοντα όπου εκτελούνται εντατικές σε χρόνο και χώρο συναλλαγές (time and space intensive)

⁵ Στη συντριπτική πλειοψηφία των υλοποιήσεων χρησιμοποιείται το πρωτόκολλο HTTP το οποίο σε κάθε περίπτωση απαιτεί ανταλλαγή τουλάχιστον δύο μηνυμάτων αίτηση και στην περίπτωση (request-only).

μεταξύ πελατών και των παρεχόμενων υπηρεσιών/ προϊόντων. Ένα χαρακτηριστικό παράδειγμα στις τηλεπικοινωνιακές επιχειρήσεις είναι οι Web Services που παρουσιάζουν τις online αναλύσεις των τηλεφωνικών κλήσεων ενός λογαριασμού πελάτη. Λαμβάνοντας υπόψη ότι υπάρχουν τηλεπικοινωνιακοί φορείς με εκατομμύρια πελατών και δισεκατομμύρια καθημερινών τηλεφωνικών συναλλαγών καθημερινά η ανάγκη για αποδοτικούς αλγορίθμους γίνεται ολοένα και μεγαλύτερη. Σε αυτήν την περίπτωση, μια Web Service πρέπει να μπει σε μια χρονοβόρα διαδικασία αναζήτησης που απαιτεί υπομονή εκ μέρους του WS καταναλωτή (WS consumer). Αυτό θα συμβεί επειδή θα χρειαστεί να πραγματοποιηθεί η αναζήτηση σε όλες τις μη τιμολογημένες κλήσεις (unbilled) των συναλλαγών μιας ολόκληρης ημέρας για να παρουσιάσει αναλυτικά τα τηλεφωνήματα κάθε πελάτη. Επιπλέον, αυτή η χρονοβόρα Web Service μπορεί να υπόκειται και σε επιπλέον καθυστερήσεις οι οποίες εξαρτώνται από τις προδιαγραφές διαθεσιμότητας και απόδοσης των πόρων της υποδομής του φορέα. Συνεπώς είναι σημαντικό για την αποτελεσματικότητα της τελικής επιλογής Web Service να ληφθούν υπόψη αυτές οι μη λειτουργικές προδιαγραφές ποιότητας τόσο κατά την επιλογή όσο και κατά τη δέσμευση της Web Service.

2.2 Περιγραφή του Προβλήματος

Το πρόβλημα που αντιμετωπίζουμε με τους προτεινόμενους αλγορίθμους [8], [6] εντοπίζεται στην προσπάθεια για αποδοτική διαχείριση και τελική επιλογή Web Service σε ένα δικτυακό πληροφοριακό σύστημα που θα λαμβάνει υπόψη, εκτός από τα λειτουργικά χαρακτηριστικά της WS, και τους παράγοντες ποιότητας και απόδοσης τη στιγμή της κατανάλωσης-σύνδεσης με μια Web Service σε δίκτυο καλύτερης προσπάθειας (best-effort) όπως το Διαδίκτυο. Τέτοιοι παράγοντες μπορούν να περιλαμβάνουν το χρόνο εκτέλεσης της υπηρεσίας σε συγκεκριμένες συνθήκες λειτουργίας του εξυπηρετητή-παρόχου (διαθέσιμη μνήμη, διαθεσιμότητα υπολογιστικής ισχύος κλπ) καθώς και το χρόνο απόκρισης.

Συγκεκριμένα, έχουν γίνει ήδη προσπάθειες για την αναζήτηση και ανακάλυψη μιας Web Service με βάση λέξεις κλειδιά ή/και τη λειτουργική περιγραφή της. Παρόλα αυτά, στον τομέα της αρχιτεκτονικής και της ανακάλυψης Web Services δεν έχει προταθεί μέχρι στιγμής καμία εργασία σχετικά με την online δυναμική επιλογή WS τη στιγμή της διαδικασίας δέσμευσης και κατανάλωσής της. Προηγούμενη εργασία με τίτλο Object Management Group (OMG) Trading Service, η οποία τυποποιήθηκε στα μέσα της δεκαετίας του 90 και βασίστηκε στην έρευνα που πραγματοποιήθηκε κατά τη διάρκεια της προηγούμενης δεκαετίας, υποστηρίζει δυναμικές ιδιότητες, αλλά εκτός του ότι είναι δύσκολο να εφαρμοστεί, έχει υιοθετηθεί μόνο μερικώς, για λόγους που τελικά οδήγησαν και στην ανάπτυξη και την ευρεία αποδοχή των τεχνολογιών Web Service για την υλοποίηση υπηρεσιών σε δικτυοκεντρικά πληροφοριακά περιβάλλοντα. Σε αυτήν την εργασία, ο χρόνος απόκρισης δεν αναφέρεται μόνο στην ελαχιστοποίηση του χρονικού κόστους της ανταλλαγής μηνυμάτων αλλά περιλαμβάνει τη συνολική διάρκεια της δέσμευσης και κατανάλωσης της WS από έναν πελάτη σε ένα δικτυακό πληροφοριακό περιβάλλον.

Η προτεινόμενη προσέγγιση επιλέγει προσαρμοστικά την αποδοτικότερη WS μεταξύ των διαφορετικών εναλλακτικών λύσεων με βελτιστοποιημένες, πραγματικού χρόνου, μετρήσιμες παραμέτρους-παράγοντες [8], [6]. Μια αφηρημένη περίπτωση του προβλήματος απεικονίζεται στο σχήμα 7.

Σχήμα 7: Γενική άποψη του προβλήματος επιλογής Web Service

Οι αλγόριθμοί μας έχουν προκύψει κατά τη διάρκεια μελέτης και εργασίας στο δικτυακό πληροφοριακό περιβάλλον μεγάλου ιδιωτικού παρόχου εθνικού επιπέδου τηλεπικοινωνιών στην Ελλάδα. Συγκεκριμένα η ενασχόλησή μας με την περίπτωση αυτή ανέδειξε το πρόβλημα και καθόρισε τις παραμέτρους της φύσης του αποτελώντας έτσι το κίνητρο για την μελέτη και επίλυση του προβλήματος της δυναμικής διαχείρισης και επιλογής υπηρεσιών σε δικτυακά πληροφοριακά περιβάλλοντα. Γενικά, οι τηλεπικοινωνιακοί φορείς χρησιμοποιούν πύλες/εξυπηρετητές για να παρέχουν υπηρεσίες σταθερής και κινητής τηλεφωνίας, οι οποίες έχουν την ικανότητα να καταγράφουν τις συναλλαγές κλήσης είτε σε απλά αρχεία είτε άμεσα σε σχεσιακές βάσεις δεδομένων. Κατά συνέπεια, υπάρχουν πολλαπλές επιχειρησιακές ανάγκες για κατανάλωση Web Services που παραδίδουν αναφορές τιμολόγησης και ανταλλαγής δεδομένων όπως αναλυτικές καταγραφές του πελάτη/κλήσης (customer/call detailed records-cdr). Οι καταγραφές cdr είναι το ίχνος μιας συναλλαγής τηλεπικοινωνιακής υπηρεσίας και περιλαμβάνει την τιμολόγηση και τις πληροφορίες επικοινωνίας (προέλευση κλήσης, προορισμός, διάρκεια, κόστος, κλπ). Κάθε συναλλαγή παράγει πολλαπλές εγγραφές cdr οι οποίες αποθηκεύονται σε ΒΔ, παράγοντας κατ' αυτό τον τρόπο σταδιακά ένα τεράστιο ποσό δεδομένων. Ο αρχικός χειρισμός ενός τέτοιου μεγάλου συνόλου δεδομένων γίνεται συνήθως με χρονοπρογραμματιζόμενες αυτοματοποιημένες

(batch) διαδικασίες. Ωστόσο, το βαθύτερο κίνητρο είναι η παροχή προηγμένων Web Services προστιθέμενης αξίας που θα διευκολύνουν την online επιχειρησιακή κατανάλωση WS που παρέχουν ενημερωμένες πληροφορίες ενός λογαριασμού πελάτη ακόμη και για τις μη τιμολογημένες cdr συναλλαγές που δεν έχουν ακόμα προ-επεξεργαστεί (batch preprocessing).

Επιπλέον, οι πληροφορίες σχετικά με τη βασισμένη σε Web Services κατανάλωση των συναλλαγών των πελατών είναι απαραίτητες και στους συνεργάτες. Η λύση που σχεδιάστηκε έπρεπε να παρέχει τις απαραίτητες πληροφορίες συναλλαγών cdr για να επιτρέψει την ασύγχρονη ανίχνευση και την πιθανή πρόληψη της πλαστής υπερχρέωσης (faked overcharging). Τέτοια υπερφόρτωση μπορεί να προκληθεί χειροκίνητα από κακόβουλους χρήστες ή αυτόματα από ιούς υπολογιστών σε περιπτώσεις κλοπής (“hacked”, “snooped”) κωδικών πρόσβασης τηλεφωνικών λογαριασμών. Γενικά, στις υπηρεσίες τηλεφωνίας σε κάθε πελάτη ανατίθενται ένας ή περισσότεροι προσωπικοί αριθμοί αναγνώρισης και αντίστοιχοι κωδικοί πρόσβασης για την πιστοποίηση στο σύστημα, ακριβώς όπως στις περιπτώσεις της βασισμένης σε προπληρωμένη κάρτα τηλεφωνίας. Οι τηλεπικοινωνιακοί φορείς (συμπεριλαμβανομένων των Παρόχων Υπηρεσιών Διαδικτύου - ISP) χρειάζονται μέτρα για να προστατεύσουν τους πελάτες τους από απροσδόκητες διακυμάνσεις στους λογαριασμούς τους. Κατά συνέπεια, έχουν αναπτυχθεί και δημοσιευθεί διάφορες υπηρεσίες με σκοπό να ικανοποιήσουν τις ανάγκες των επιχειρησιακών συνεργατών και των συνεταιίρων στα διαφορετικά σημεία της παρουσίας (POP) του εταιρικού δικτύου.

Για να παράσχουμε ποιότητα υπηρεσίας (QoS) σε τέτοιου είδους νέες υπηρεσίες προστιθέμενης αξίας, είχαμε το κίνητρο να μεγιστοποιήσουμε τη ρυθμαπόδοση και να μειώσουμε το χρόνο αναμονής για εκτέλεση Web Service με αλγόριθμους που περιγράφονται στο παρόν κεφάλαιο για αποδοτική αναίχλυνση και δέσμευση της καταλληλότερης Web Service, έτσι ώστε να διατηρηθεί η αίσθηση της “online” απόκρισης.

2.3 Σχετικές εργασίες

Στην παρούσα ενότητα θα παρουσιαστούν σύντομα παλαιότερες σχετικές εργασίες. Εκτενής ανάλυση έχει ήδη παρουσιαστεί στο πρώτο κεφάλαιο της διατριβής και εδώ γίνεται μία σύντομη μόνο αναφορά στις χαρακτηριστικότερες και πλησιέστερες προϋπάρχουσες προσεγγίσεις με αυτήν του παρόντος κεφαλαίου.

2.3.1 Απομακρυσμένη κλήση μεθόδων

Πριν προχωρήσουμε μια σύντομη επισκόπηση θα αποκαλύψει τις διαφορετικές πτυχές και τις σχετικές τεχνολογίες. Μια σημαντική έννοια στη σύγχρονη ανάπτυξη λογισμικού είναι αυτή της Service Oriented Architecture ή SOA [20]. Το SOA είναι μια σχετικά νέα προσπάθεια να αντιμετωπιστούν τα παλαιά προβλήματα που αφορούν στη διαμοίραση δεδομένων και την απομακρυσμένη κλήση μεθόδων. Χαρακτηριστικά σενάρια, όπου εμφανίζονται οι προηγούμενες ενέργειες,

περιλαμβάνουν επιχειρήσεις που διαμοιράζονται πληροφορίες με πελάτες, τμήματα επιχειρήσεων που διαμοιράζονται δεδομένα με άλλα τμήματα και εφαρμογές τελικών χρηστών που συνεργάζονται με άλλες εφαρμογές τελικών χρηστών. Όλες αυτές οι περιπτώσεις αντιμετωπίζονται επίσης από τη λύση SOA.

Αυτοί οι στόχοι έχουν εμφανιστεί από τα πρώτα χρόνια της ενδοεπικοινωνίας υπολογιστών και έχουν παρουσιαστεί διάφορες μεθοδολογίες οι οποίες οδήγησαν στην εισαγωγή της αποκαλούμενης *ανάπτυξης καταναεμημένων εφαρμογών*. Η ανάπτυξη καταναεμημένων εφαρμογών εστιάζει στις μεθόδους εφαρμοσμένης μηχανικής για να πάρει δεδομένα από έναν κόμβο σε άλλον. Στην πραγματικότητα, έχουν υπάρξει πολλές τεχνολογίες για την δημιουργία εφαρμογών που μπορούν να στείλουν τα δεδομένα εμπρός και πίσω, όπως η CORBA (Common Object Request Broker Architecture), η RMI (Remote Method Invocation), και το DCOM (Distributed Component Object Model).

Κάθε ένα από τα παραπάνω έχει αποτύχει για διάφορους λόγους, συμπεριλαμβανομένων των πολύπλοκων απαιτήσεων υλοποίησης και υποδομής, των πρωτόκολλων επικοινωνίας, την έλλειψη υποστήριξης από μεγάλες επιχειρήσεις λογισμικού και των περιορισμών πλατφόρμας. Επομένως, η κοινότητα ανάπτυξης δικτυοκεντρικών πληροφοριακών συστημάτων οδηγήθηκε στην ανάπτυξη της αρχιτεκτονικής των Web Services.

2.3.2 Τεχνικές Ανακάλυψης Web Service

Αν και ο τομέας της ανακάλυψης Web Service είναι μάλλον νέος, ακόμα νεότερος από τις ίδιες τις Web Services, έντονη έρευνα έχει αφιερωθεί πρόσφατα. Η προσπάθεια στην πλειονότητα των προσεγγίσεων έχει αφιερωθεί στην ενίσχυση των μηχανισμών ανακάλυψης προκειμένου να υπερνικηθεί η ανεπάρκεια της βασισμένης σε λέξεις κλειδιά (keyword-based) αντιστοίχισης, στην οποία συχνά ο χρήστης δε μπορεί να ανακαλύψει την Web Service που επιθυμεί.

Σε αυτό το πνεύμα, οι Sajjanhar et al. [16] πρότειναν την αναπαράσταση των Περιγραφών Υπηρεσίας (Service Descriptions) ως κείμενα, και αντιστοιχίζοντας τα παραγόμενα διανύσματα σε ένα χώρο ιδιοτήτων μικρών διαστάσεων παρατήρησαν καλύτερη ανάκτηση Web Services. Μια ενοποιημένη λύση έχει προταθεί στο [27], όπου παρουσιάστηκε μια ευέλικτη γλώσσα διατύπωσης ερωτημάτων τύπου SQL για να υποστηρίξει την ευέλικτη ανάκτηση των υπηρεσιών. Έχει βασιστεί στη χρήση του Service Grid [24], [25], [26], το οποίο οργανώνει τις υπηρεσίες σε ένα κανονικοποιημένο και ορθογώνιο πολυδιάστατο χώρο υπηρεσιών.

Η διαδικασία αντιστοίχισης σε όλες τις παραπάνω περιπτώσεις, δε λαμβάνει υπόψη της μη λειτουργικές παραμέτρους όπως ο χρόνος απόκρισης, Εντούτοις, σε ένα πραγματικό περιβάλλον κλήσης υπηρεσιών, πτυχές όπως ο χρόνος απόκρισης είναι ιδιαίτερα σημαντικές. Τα χαρακτηριστικά όπως ο χρόνος εκτέλεσης συναντιούνται υπό τον όρο Ποιότητα Υπηρεσίας (Quality of Web Service - QoWS). Στο [15] παρουσιάζονται ορισμένα πιο επιθυμητά χαρακτηριστικά τα οποία ορίζουν την QoWS. Μια παρόμοια προσπάθεια έγινε στο [14] όπου αντιμετωπίζεται για πρώτη

φορά το θέμα της ανακάλυψης Web Services η οποία ικανοποιεί συγκεκριμένους ποιοτικούς περιορισμούς. Αν και οι λεπτομέρειες επιλογής είναι κάπως ασαφείς, παρουσιάζεται ένα ενοποιημένο πλαίσιο ανακάλυψης QoWS.

Επιπλέον, στο [23] παρουσιάζεται ένας αποδοτικός αλγόριθμος για την επιλογή του βέλτιστου σχεδίου εκτέλεσης όταν εκτελείται ένα διάγραμμα ροής διαδικασίας Web Service. Αντιμετωπίζονται οι περιπτώσεις όπου το ενδιαφέρον στρέφεται στα σχέδια εκτέλεσης που αποτελούνται από μια ακολουθία k διαδοχικά εκτελούμενων Web Services $\{S_1, S_2, \dots, S_k\}$ ή ένα “pipeline” όπως ονομάζεται. Η επιλογή μιας ακολουθίας εκτέλεσης με συνολικό χρόνο εκτέλεσης μικρότερο από ένα κατώφλι R μετασχηματίζεται σε ένα Multiple Choice Knapsack (MCK). Στο [22] παρέχεται ένας δυναμικός αλγόριθμος και παραλλαγές για την επίλυση του MCK και την αποτελεσματική επιλογή της πιο κατάλληλης ακολουθίας υπηρεσιών. Επιπλέον, στα [21] και [23] παρουσιάζεται ένας μηχανισμός που ονομάζεται QoS Broker (ενδιάμεσος) για Web Services. Είναι ένας ενδιάμεσος μεταξύ του καταλόγου UDDI και του τελικού χρήστη ο οποίος εκτελεί όλα τα απαιτούμενα έργα για να εντοπίσει και να εξασφαλίσει την παράδοση των καταλληλότερων Web Services.

Στο [7] παρουσιάζεται μια γενικευμένη πλευρά του προκλητικού στόχου να επινοηθεί μια συνάρτηση βαθμολόγησης που να συνδυάζει τα κριτήρια QoWS κατά δίκαιο τρόπο. Ο στόχος είναι η παραγωγή μιας ενιαίας βαθμολόγησης για κάθε Web Service ή ενός σχεδίου εκτέλεσης, αναπαριστώντας κάθε WS ως ένα διάνυσμα m διαστάσεων, όπου το m είναι ο αριθμός παραμέτρων που λαμβάνονται υπόψη.

Συνολικά, εκτενής και συγκριτική έρευνα για τους μηχανισμούς Ανακάλυψης Web Service παρουσιάζεται στο κεφάλαιο 1 της παρούσας διατριβής.

2.4 Οδικός Χάρτης της Προσέγγισής μας

Στην παρούσα ενότητα θα παρουσιαστούν οι βασικοί αλγόριθμοι που αποτελούν την επίλυση που δώσαμε στο πρόβλημα της δυναμικής επιλογής Web Services με βάση μη λειτουργικά/ποιοτικά χαρακτηριστικά. Η περίπτωση αυτή εφαρμόζεται στα δικτυακά πληροφοριακά συστήματα και συναντιέται ευρέως στις περιπτώσεις επιχειρησιακής συνεργασίας και επικοινωνίας μεταξύ φορέων προκειμένου να εκτελεστούν μια συγκεκριμένη σειρά διαδικασιών, οι οποίες συνεπώς συσχετίζονται σημασιολογικά. Κατά συνέπεια, απαιτείται η ανάπτυξη προγραμματιστικών διεπαφών για να ανακαλυφθούν και να καταναλωθούν Web Services και συγκεκριμένα τείνουν να εκτελούν ερωτήματα για Web Services με παρόμοια λειτουργία που άπτεται της επιχειρησιακής τους δραστηριότητας. Επιπλέον, πρέπει να σημειωθεί ότι τέτοιες διαδικασίες αναζήτησης διεξάγονται μέσα σε ένα συγκεκριμένο εύρος συνεργατών - παρόχων Web Service και επομένως τα στοιχεία είναι πεπερασμένα και η προτεινόμενη διαδικασία τερματίζει. Σε τέτοιες περιπτώσεις, ο στόχος μας είναι να παρέχουμε μια αποδοτική διαδικασία ανακάλυψης για τη δυναμική επιλογή της καταλληλότερης Web Service. Κάτι τέτοιο θα συνεπάγονταν τη συγκέντρωση των χαρακτηριστικών απόδοσης όλων των διαθέσιμων WS και έπειτα απλά την επιλογή μιας WS που να μεγιστοποιεί ή να ελαχιστοποιεί μια συγκεκριμένη μετρική.

Ο αλγόριθμος περιλαμβάνει δύο λειτουργικά υποσυστήματα που συνθέτουν το τελικό βήμα της επιλογής Web Service. Η γενική λογική του αλγορίθμου απεικονίζεται στο σχήμα 8 και περιγράφηκε στην εργασία μας [8] και [6]. Σε αυτό το σχήμα παρουσιάζεται η αλληλεπίδραση μεταξύ των λειτουργικών υποσυστημάτων της λύσης.

Σχήμα 8: Οδικός χάρτης αλγορίθμου δυναμικής επιλογής WS

Πριν προχωρήσουμε στην ανάλυση κάθε λειτουργικού υποσυστήματος, παρουσιάζεται μια περίληψη του αλγορίθμου. Αυτή η ενότητα χρησιμεύει ως ένας οδικός χάρτης (roadmap) για τη διευκόλυνση του αναγνώστη, ενώ οι απαραίτητες λεπτομέρειες μπορούν να βρεθούν στις υποενότητες που ακολουθούν.

(1) Επιλογή Contour: Αρχικά ο προτεινόμενος αλγόριθμος λαμβάνει υπόψη δύο παραμέτρους και με βάση αυτές, εκτελεί μια αποδοτική επιλογή των καλύτερων υποψηφίων Web Services. Το βήμα αυτό απεικονίζεται στο σχήμα 8a και συγκεκριμένα, οι παράμετροι περιλαμβάνουν:

- Τη δικτυακή απόσταση, η οποία είναι η μέση παρατηρημένη καθυστέρηση (latency) δικτύου μεταξύ του πελάτη και της Web Service.
- Τον αριθμό άλλων διακριτών Web Services, οι οποίες σχετίζονται λειτουργικά με την Web Service από την πλευρά του επιχειρησιακού περιβάλλοντος. Η παράμετρος αυτή συμβάλλει στην περαιτέρω μείωση του αριθμού των υποψηφίων Web Services.

(2) Προσαρμοστική επιλογή: Ως δεύτερο λειτουργικό υποσύστημα προτείνεται η διαδικασία *προσαρμοστικής επιλογής*. Είναι βασισμένη στις online τιμές των παραμέτρων ποιότητας μιας Web Service (QoWS ratings). Αυτά είναι υπολογίσιμοι παράγοντες όπως η διαθέσιμη μνήμη, κ.λπ., που σχετίζονται με την ποιότητα και τις συνθήκες διαθεσιμότητας της υποδομής που υλοποιεί την WS κατά την στιγμή εκτέλεσης. Με το πέρας του χρόνου και έπειτα από έναν αριθμό εκτελέσεων

παρατηρείται ο μέσος χρόνος εκτέλεσης της κάθε υπηρεσίας. Σε αυτό το σημείο για να επιβεβαιώσουμε τη μεγιστοποίηση της ποιότητας στην επιλεγμένη Web Service, χρησιμοποιείται το εταιρικό δίκτυο για να συλλέξει ενημερωμένες μετρήσεις της WS και να εκτιμήσει τα αναμενόμενα χαρακτηριστικά QoWS με βάση τα προηγούμενως καταγεγραμμένα σχέδια εκτέλεσης-προφίλ. Η εναλλακτική λύση του σχεδιασμού τακτικών δυναμικών ενημερώσεων των QoWS ιδιοτήτων του UDDI μπορεί να χρησιμοποιηθεί στα περιβάλλοντα όπου η κυκλοφορία εταιρικού δικτύου είναι ήδη υψηλή.

Αυτή η διαδικασία περιλαμβάνει μια προπαρασκευαστική και τρεις online δυναμικές ενέργειες:

(α) Δημιουργία και ενημέρωση των τιμών παραμέτρων ποιότητας (QoWS ratings). Η υπηρεσία αυτή διατηρεί ιστορικά προφίλ των WS και αποτελεί μια προπαρασκευαστική δράση η οποία καταγράφει τις τιμές παραμέτρων ποιότητας καθώς επίσης και τον αντίστοιχο χρόνο εκτέλεσης της WS και τα αποθηκεύει ως ένα ιστορικό προφίλ (από εδώ και στο εξής αποκαλούμενο σχέδιο εκτέλεσης της WS – WS execution scheme). Ειδικότερα αποθηκεύει τη διάρκεια που παίρνει μια WS για να επεξεργαστεί ένα αίτημα κάτω από τις συγκεκριμένες τιμές παραμέτρων ποιότητας καθώς επίσης και τις ίδιες τις τιμές των παραμέτρων ποιότητας. Αυτή η δράση χρησιμεύει ως μια επιπρόσθετη (add-in) υπηρεσία που πραγματοποιείται πριν από τη διαδικασία επιλογής. Αυτή η προπαρασκευαστική δράση διατηρεί αποθηκευμένα τα σχέδια εκτέλεσης της WS. Οι λεπτομέρειες της ενέργειας αυτής της προσέγγισης περιλαμβάνονται στην παράγραφο 2.6.1.

(β) Online/πραγματικού χρόνου αίτημα των τρεχουσών (τελευταίων) τιμών παραμέτρων ποιότητας WS. Αυτή η ενέργεια χρησιμοποιεί ένα πράκτορα υπηρεσίας που επιστρέφει τις τιμές συνθηκών του WS που εξετάζεται. Ο πράκτορας μπορεί να ρυθμιστεί να επιστρέφει οσοδήποτε τιμές αρκεί να έχουν προβλεφθεί να περιλαμβάνονται στο σχήμα εκτέλεσης της WS. Αυτή η online δράση επιστρέφει την τρέχουσα τιμή μετρήσιμων παραγόντων του WS. Η διαδικασία συμπεριλαμβάνεται ως μέρος της κύριας online δράσης.

(γ) Ο online υπολογισμός του αναμενόμενου χρόνου εκτέλεσης της WS με βάση τις τρέχουσες τιμές παραμέτρων ποιότητας (QoWS). Αυτή η δράση βρίσκει τον κοντινότερο στο σχέδιο εκτέλεσης της WS χρόνο απόκρισης που είναι και ο πλέον αναμενόμενος. Εκτελεί την αντιστοίχιση με βάση την Ευκλείδεια απόσταση μεταξύ των τιμών παραμέτρων ποιότητας (QoWS ratings) στο σχέδιο εκτέλεσης της WS. Αυτή η online δράση επιστρέφει τον *αναμενόμενο χρόνο εκτέλεσης* της WS με βάση τις τιμές διάφορων μετρήσιμων παραγόντων διαθέσιμων στο WS POP. Ενεργεί ως μέρος της κύριας online δράσης (περιγράφεται στο σχήμα 8d).

(δ) Διαχείριση και διεκπεραίωση των online ενεργειών και παραγωγή του αποτελέσματος επιλογής. Αυτή είναι η τελευταία online δράση που διαχειρίζεται τις προηγούμενες δύο και εκτελεί τα κύρια βήματα της προσαρμοστικής επιλογής του προτεινόμενου αλγορίθμου. Επιστρέφει την τελικά επιλεγμένη Web Service που θα

χρησιμοποιηθεί. Η λεπτομερής περιγραφή της δράσης μπορεί να βρεθεί στην παράγραφο 2.6.2.

2.5 Πρώτο βήμα επιλογής με χρήση υπολογιστικής γεωμετρίας

Η εκτίμηση της απόδοσης για κάθε κόμβο που παρέχει web services είναι μια υπολογιστικά απαιτητική διαδικασία και άρα πρέπει να την εφαρμόσουμε σε όσο λιγότερους κόμβους είναι δυνατό. Επομένως, χρειάζεται η εφαρμογή ευριστικών μεθόδων που να επιτρέπει τον αποκλεισμό συγκεκριμένων κόμβων ως πρώτο βήμα. Αυτό το ευριστικό καλείται *επιλογή contour*. Πριν προχωρήσουμε στην επιλογή, παρέχουμε ορισμένες χρήσιμες έννοιες (Σχήμα 9).

Σχήμα 9: Το πρώτο (1^ο) βήμα δυναμικής επιλογής WS με QoS

Έστω ότι το S δηλώνει ένα σύνολο από υποψήφιους κόμβους, έτσι ώστε κάθε κόμβος παρέχει ένα σύνολο από Web Services τις οποίες ενδιαφερόμαστε να καταναλώσουμε. Για ένα τυχαίο στοιχείο $s_i \in S$, ορίζουμε δύο παραμέτρους: τη δικτυακή του απόσταση d_i από τον κόμβο που υποβάλλει το ερώτημα και τον αριθμό f_i των διακριτών Web Services που παρέχει. Η επιλογή της απόστασης βασίζεται στην ανάγκη να αποκλειστούν αρχικά οι κόμβοι που έχουν μεγαλύτερη απόσταση από το σημείο εισόδου μας, επομένως ο χρόνος εκτέλεσης συν τη δικτυακή καθυστέρηση μπορεί πιθανώς να προκαλέσουν υπερβολικά υψηλό συνολικό χρόνο εκτέλεσης. Η επιλογή του αριθμού των παρεχόμενων Web Services προκύπτει ακολουθώντας τη λογική ότι εάν ένας συγκεκριμένος κόμβος παρέχει συγκεκριμένο αριθμό Web Services, τότε αυξάνεται η πιθανότητα να εξυπηρετεί ένα μεγάλο υποσύνολο του απαιτούμενου συνόλου Web Services, μέσα στους ίδιους κόμβους, ειδικότερα λαμβάνοντας υπόψη ότι οι καταναλωτές λειτουργούν μέσα σε ένα συγκεκριμένο επιχειρησιακό πλαίσιο (π.χ. Web Services για τουρισμό: κράτηση εισιτηρίων, κράτηση δωματίων ξενοδοχείου, κλπ.). Η ευριστική μέθοδος επιλογής με χρήση contour, επιλέγει κόμβους που είναι μέγιστοι σε σχέση τόσο με τη δικτυακή απόσταση όσο και με τον αριθμό των συναρτήσεων.

Ειδικότερα, κάθε s_i , μπορεί να αναπαρίσταται μοναδικά ως ένα σημείο $(-d_i, f_i)$ στο επίπεδο xy , όπου η απόσταση αντιστοιχεί στην συντεταγμένη x και ο αριθμός των Web Services, στην συντεταγμένη y . Ένα σημείο $s_i \in S \subseteq \mathbb{R}^2$, $s_i(x_i, y_i)$ καλείται

κυρίαρχο σημείο (maximal) εάν δεν υπάρχει άλλο σημείο $s_j(x_j, y_j)$ τέτοιο ώστε να ισχύει $x_j \geq x_i$ και $y_j \geq y_i$. Μπορεί εύκολα να επαληθευτεί ότι μέγιστα σημεία σχηματίζουν μια κλίμακα στο επίπεδο, η οποία στο εξής θα ονομάζεται *contour* (Σχήμα 10). Το σύνολο $M \subseteq S$ των μέγιστων σημείων αντιστοιχεί στους *servers* που δεν μπορούν να ξεπεραστούν ούτε στον αριθμό των υπηρεσιών ούτε και στην κοντινότερη απόσταση από οποιοδήποτε άλλο *server* που παρέχει μια συγκεκριμένη υπηρεσία. Οποιοδήποτε από αυτά τα σημεία (*servers*) θα μπορούσε πιθανώς να οδηγήσει σε μια καλή απόφαση όσον αφορά στην υπηρεσία που ζητήθηκε.

Δεδομένου ενός συνόλου S σημείων σε ένα επίπεδο, το υποσύνολο M των μέγιστων σημείων, μπορεί να υπολογιστεί με τον αλγόριθμο **CONTOUR_CONSTRUCTION**. Ο αλγόριθμος 1 χρησιμοποιεί ένα ισοζυγισμένο δυαδικό δέντρο αναζήτησης T , για να αποθηκεύσει σημεία στο M σύμφωνα με την x συντεταγμένη τους. Θεωρούμε ότι το T είναι φυλλο-προσανατολισμένο, που σημαίνει ότι τα σημεία αποθηκεύονται μόνο στα φύλλα των δέντρων ενώ στους εσωτερικούς κόμβους αποθηκεύονται μόνο συγκεκριμένες πληροφορίες δρομολόγησης. Για πιο εκτενείς πληροφορίες για την έννοια των φυλλο-προσανατολισμένων δέντρων, μπορεί κανείς να ανατρέξει στο βιβλίο του Mehlhorn [9]. Κάθε φύλλο v , εκτός από την συντεταγμένη x των σημείων p που αποθηκεύει, κρατάει επίσης έναν δείκτη στο p . Επιπλέον θεωρούμε ότι κάθε φύλλο δέντρου v , περιέχει δείκτες $v.left$ και $v.right$ στα αριστερά και δεξιά αδέρφια του (*siblings*), αντίστοιχα. Ο αλγόριθμος **CONTOUR_CONSTRUCTION** που παρουσιάζεται κατασκευάζει μια αναπαράσταση *contour* με δεδομένο ένα σύνολο σημείων στο επίπεδο.

Contour_Construction (x)

Input: A set of points M

Output: The set of maximal points M

1. Sort points in S according to their x -coordinates

(* Let $s_1, s_2, \dots, s_n$ denote the sorted order *)

2. **for** $i=1$ **to** n

(* Current points in $s_i(x_i, y_i)$ *)

3. search T with x_i ,
(* Search will terminate at a leaf u storing point $p(x_i, y)$
and having $x \leq x_i$. *)

4. $cur \leftarrow u$;

5. **if** ($u.y < y_i$) **insert** (T, x_i);

6. **while** ($u.y < y_i$) **do**

7. $cur \leftarrow u.left$;

8. **delete** (T, u);

9. $u \leftarrow cur$;

10. **end -while**

11. **end-for**

Αλγόριθμος 1: Ο αλγόριθμος της επιλογής *contour*

Η διαδικασία που πραγματοποιείται κατά τη διάρκεια της κατασκευής *contour* απεικονίζεται καλύτερα στον Αλγόριθμο 1. Το σχήμα 10 παρουσιάζει το *contour* (συνεχής γραμμή) του συνόλου σημείων $S = \{(-13, 2), (-11, 13), (-10, 7), (-9, 11), (-$

$8,3$), $(-7,5)$, $(-4, 5)$, $(-3,7)$ }. Αφού έχουν επεξεργαστεί τα πρώτα επτά σημεία, η εισαγωγή του $s_8(-3, 7)$ στο δέντρο T , προηγείται από μια αναζήτηση (γραμμή 3), η οποία τελειώνει σε ένα φύλλο που αποθηκεύει το -4 . Στη συνέχεια αποθηκεύουμε το s_8 και αναζητούμε στα αριστερά του ώστε να αποκλείσουμε οποιαδήποτε σημεία στα οποία υπερσχύει το πρόσφατα εισαχθέν (s_8) (γραμμές 6–10). Αυτό οδηγεί στη διαγραφή των $(-7, 5)$ και $(-4, 5)$ από το T . Αυτή η περίπτωση παρουσιάζεται στο σχήμα 10b όπου τα διαγραμμένα φύλλα είναι σκιασμένα.

Σχήμα 10: Παράδειγμα ενημέρωσης contour

Το contour του $S = \{(-13, 2), (-11,13), (-10,5), (-9,11), (-8,3), (-7,5), (-4, 5), (-3,7)\}$. (a) Το contour μετά την εισαγωγή των επτά πρώτων σημείων. (b) Ενημέρωση του contour μετά την εισαγωγή του $s_8(-3, 7)$. Τα φύλλα που διαγράφηκαν από το T υποσκιαζονται. Το νέο contour φαίνεται με συνεχή γραμμή και το παλιό με διακεκομμένη. Τα αντίστοιχα στιγμιότυπα του T φαίνονται και στις δύο περιπτώσεις

Το πάνω όριο της πολυπλοκότητας κατασκευής του contour με τον αλγόριθμο CONTOUR_CONSTRUCTION μπορεί να προκύψει ως εξής:

Προφανώς οι αναζητήσεις και οι εισαγωγές σε ένα ισοζυγισμένο δυαδικό δέντρο αναζήτησης n -κόμβων παίρνει χρόνο $O(\log n)$, για ένα σύνολο $O(n \log n)$ συνολικού χρόνου. Παρόλα αυτά, μετά την εισαγωγή ενός σημείου s_i στη γραμμή 5, ίσως χρειαστεί η σάρωση των φύλλων του δέντρου και η διαγραφή οποιονδήποτε σημείων αποθηκεύονται στα φύλλα, των οποίων οι συντεταγμένες x (και y) είναι κυρίαρχες από το s_i . Ίσως χρειαστεί να γίνουν έως και $O(n)$ τέτοιες λειτουργίες μετά από μια εισαγωγή, με κόστος $O(n \log n)$. Ωστόσο, το συνολικό κόστος όλων των εκτελέσεων των γραμμών 6–10 δεν είναι $O(n^2 \log n)$. Ας υποθέσουμε ότι οι γραμμές 6–10, στο i -οστό σημείο εισαγωγής, διαγράφουν $k_i \geq 0$ φύλλα, όπου $i = 1, \dots, n$. Επομένως, στην i -οστή επανάληψη το κόστος είναι $O(k_i \log n)$.

Παρατηρούμε ωστόσο ότι $\sum_{i=1}^n k_i = n$, καθώς κάθε σημείο μπορεί να διαγραφεί το πολύ μια φορά, μιας και όταν ένα σημείο διαγραφεί δεν εισάγεται ξανά. Επομένως το

συνολικό κόστος της εκτέλεσης των γραμμών 6–10, πάνω σε όλα τα πιθανά i είναι $\sum_{i=1}^n O(k_i \log n) = O(n \log n)$. Η παραπάνω συζήτηση οδηγεί στο παρακάτω λήμμα.

Λήμμα 2.1. Η CONTOUR_CONSTRUCTION για n σημεία ολοκληρώνεται σε χρόνο $O(n \log n)$.

Σχήμα 11: Προσαρμοστική επιλογή (Offline βήμα) της δυναμικής επιλογής WS με QoS

2.6 Προσαρμοστική επιλογή

Καθώς η επιλογή contour έχει ολοκληρωθεί επιτυχώς, υπάρχει μια λίστα από περισσότερες από μια υποψήφιες Web Services για την τελική επιλογή. Στις παραγράφους που ακολουθούν θα περιγραφεί η βελτιστοποίηση της αντιστοίχισης των καταγεγραμμένων Web Services με βάση την online ανατροφοδότηση τιμών παραμέτρων ποιότητας. Το επόμενο βήμα στην προτεινόμενη μέθοδο επιλογής του κεφαλαίου είναι η προσαρμοστική επιλογή. Αρχικά παρουσιάζεται μια προπαρασκευαστική δράση, όπου καταγράφετε το προφίλ των Web Services. Στη συνέχεια παρουσιάζονται αναλυτικά οι online δράσεις.

2.6.1 Προπαρασκευαστική δράση: Διατήρηση του ιστορικού προφίλ των WSs

Η πρώτη προπαρασκευαστική δράση του αλγορίθμου προσαρμοστικής επιλογής καταγράφει τα ιστορικά προφίλ των WSs (βήμα (a) στο σχήμα 11). Κάθε Web Service ενδιαφέροντος αποθηκεύεται με ένα σχήμα εκτέλεσης QoWS (το ιστορικό προφίλ δηλαδή της WS) που περιλαμβάνει λεπτομέρειες για κάθε διαφορετικό σχέδιο εκτέλεσης. Τα σχέδια εκτέλεσης περιλαμβάνουν λεπτομέρειες (τιμές παραμέτρων ποιότητας) σχετικά με την Web Service που εκτελείται σε ένα συγκεκριμένο περιβάλλον (μετρήσιμες παράμετροι ποιότητας και διαθεσιμότητας της υποδομής). Επιπλέον, κάθε εκτέλεση WS χρειάζεται κάποιο χρόνο για να εκτελεστεί, ο οποίος ονομάζεται *χρόνος εκτέλεσης*. Οι πιο κρίσιμοι μετρήσιμοι παράγοντες που μπορεί να επιμηκύνουν το χρόνο εκτέλεσης μπορεί να είναι το ποσοστό της μέσης χρησιμοποίησης cpu, το ποσοστό της μέσης χρησιμοποίησης

μνήμης, ο μέσος αριθμός διαδικασιών και ο μέσος αριθμός συναλλαγών αρχείων. Επίσης, μπορεί κανείς να λάβει υπόψη τους μετρήσιμους παράγοντες πραγματικού χρόνου (πλήρης λίστα των παραμέτρων είναι διαθέσιμη στα [11], [12]). Η ένταξη σε μικρότερο ή μεγαλύτερο βαθμό των αποθηκευμένων παραγόντων στο σχέδιο εκτέλεσης δεν επηρεάζει τη συμπεριφορά του αλγορίθμου επιλογής, καθώς η αποθήκευση είναι μια προκαταρκτική δράση.

Κατά συνέπεια το σχήμα εκτέλεσης QoWS περιλαμβάνει μια λίστα από διαφορετικά σενάρια που περιγράφουν το χρόνο εκτέλεσης ως μια συνάρτηση αποθηκευμένων παραγόντων ποιότητας και διαθεσιμότητας της υποδομής. Σε περίπτωση πιθανής αλλαγής υποδομής (λόγω αναβάθμισης υλικού για παράδειγμα), το σχήμα εκτέλεσης QoWS θα πρέπει να συνταχθεί ξανά.

Οι αποθηκευμένες παράμετροι αποτελούνται από μετρητές σε επίπεδο συστήματος, υπηρεσίας και εφαρμογών. Υπάρχουν ορισμένοι γενικοί μετρητές όπως η χρήση CPU, η κατάρληψη μνήμης, η χρησιμοποίηση I/O σε δίσκο και ταυτόχρονα διαδικασίες της ίδιας εφαρμογής στο σύστημα (επίπεδο εφαρμογής), αιτήματα συναλλαγών σε άλλες WSs (επίπεδο υπηρεσίας) και άλλα. Οι πληροφορίες των σχημάτων εκτέλεσης QoWS όλων των WSs στο δίκτυο αποθηκεύονται σε μια βάση δεδομένων μαζί με τον κατάλογο των διαθέσιμων WS.

Σχήμα 12: Προσαρμοστική επιλογή (Online βήματα) της δυναμικής επιλογής WS με QoS

2.6.2 Online δράσεις: Αποδοτική επιλογή

Μετά το προπαρασκευαστικό πρώτο βήμα του αλγορίθμου προσαρμοστικής επιλογής που περιλαμβάνει την αποθήκευση των QoWS σχημάτων των Web Services, παρουσιάζουμε τις online δράσεις του αποδοτικού αλγορίθμου προσαρμοστικής επιλογής που επιλέγει τις WS (τα βήματα που περιλαμβάνονται παρουσιάζονται στο σχήμα 12). Οι συμμετέχοντες των online δράσεων είναι:

1. Η request agent service των τρεχουσών τιμών παραμέτρων ποιότητας
2. Ο αλγόριθμος υπολογισμού του αναμενόμενου χρόνου εκτέλεσης των WS με βάση τα αποτελέσματα των ανακτημένων τιμών
3. Ο αλγόριθμος διαχείρισης που εκτελεί την επιλογή.

Έχουμε δημιουργήσει μια υπηρεσία πράκτορα (agent service) που μπορεί να επικοινωνήσει με κάθε υποδομή που φιλοξενεί κάποιο WS και να ανακτήσει τις τρέχουσες (τελευταίες) παραμέτρους απόδοσης. Η υπηρεσία αυτή είναι μία WS και η ίδια που δημοσιεύει τα εργαλεία μέτρησης της απόδοσης που παρέχονται από το λειτουργικό σύστημα και τις ειδικές εφαρμογές όπως τα συστήματα διαχείρισης ΒΔ. Η λειτουργία αυτή επιστρέφει ως αποτέλεσμα τις τρέχουσες τιμές των επιλεγμένων παραμέτρων της υποδομής που υλοποιεί την WS. Ένα παράδειγμα των αποθηκευμένων τιμών παραμέτρων ποιότητας παρουσιάζει ο πίνακας 5.

Ο πίνακας 5 παρουσιάζει ένα παράδειγμα ενός σχήματος εκτέλεσης WS (Web Service execution scheme). Περιλαμβάνει τον αριθμό ταυτοποίησης της WS (id), τον καταγεγραμμένο χρόνο εκτέλεσης σε δευτερόλεπτα, ένα αναγνωριστικό ημερομηνίας και ώρας και διάφορα υπολογίσιμα μέτρα σε επίπεδο συστήματος, εφαρμογής και υπηρεσίας. Όπως έχει ήδη περιγραφεί, καταγράφονται τα χαρακτηριστικά εκτέλεσης για αυτήν την WS σε πολλαπλές διακριτές χρονικές στιγμές.

Καταγεγραμμένη τιμή χρόνου εκτέλεσης ανά WS			Επίπεδο Συστήματος				Επίπεδο Εφαρμογής		Επίπεδο Υπηρεσίας
WS id	Χρόνος Εκτέλεσης (s)	Date time id	Αριθμός διεργασιών	Μέγιστο % CPU	Μέσο % διαθέσιμης μνήμης	%	Αριθμός νημάτων εφαρμογής	I/O read write x 10 ³	Αριθμός άλλων WS διεκπεραιώσεων
12	61	0	2	5	82		1	≈200	0
12	53	1	1	6	91		1	≈180	0
12	244	2	13	25	56		2	≈1000	0
12	351	3	41	80	20		10	≈1500	0
12	157	4	9	43	74		8	≈800	0

Πίνακας 5: Ένα παράδειγμα σχήματος εκτέλεσης WS

Έχοντας διατηρήσει τα ιστορικά προφίλ, οι τρέχουσες τιμές παραμέτρων ποιότητας αντιστοιχίζονται με τον αναμενόμενο χρόνο εκτέλεσης με βάση το ιστορικό προφίλ της υποδομής. Η αντιστοίχιση εκτελείται με τον υπολογισμό της Ευκλείδειας απόστασης μεταξύ του διάνυσματος τρεχουσών τιμών παραμέτρων ποιότητας και του διάνυσματος αποθηκευμένου προφίλ.

Ειδικότερα, ας θεωρήσουμε το διάνυσμα προφίλ $h = [h_1, h_2, \dots, h_p]$, που αποτελείται από p αποθηκευμένες παραμέτρους και ένα διάνυσμα $c = [c_1, c_2, \dots, c_p]$ των τρεχουσών παραμέτρων υπηρεσίας.

Η Ευκλείδεια απόσταση L_2 μεταξύ h και c είναι $L_2 = \sqrt{\sum_{i=1}^p (c_i - h_i)^2}$.

Το διάνυσμα $h_{\min}$ που ελαχιστοποιεί το L_2 παρέχει μια εκτίμηση του τρέχοντος χρόνου εκτέλεσης. Πιθανή χρησιμοποίηση των μετρικών απόστασης L_1 ή L_∞ θα παρείχε παρόμοια αποτελέσματα.

Μια εναλλακτική διαδικασία αντιστοίχισης θα μπορούσε να χρησιμοποιεί την ευκλείδεια απόσταση μεν αλλά να χρησιμοποιεί τεχνικές χρονικής γήρανσης/απόσβεσης της αξίας των καταγραφών του προφίλ. Με τον τρόπο αυτό θέλουμε να απεικονίσουμε τη μειούμενη αξιοπιστία σε συνάρτηση με το πέρασμα του χρόνου αξία των παλαιότερων καταγραφών.

Στη συνέχεια, ο μηχανισμός που διαχειρίζεται τις παραπάνω δράσεις εκτελεί την κύρια προσαρμοστική επιλογή WS. Αρχικά, εκτελείται το βήμα κλαδέματος (pruning) της παραγράφου 2.5. Αυτό το βήμα μειώνει το χώρο των υποψηφίων για επιλογή Web Services σε αυτές τις υπηρεσίες που ανήκουν στο contour. Έστω ότι αυτό το σύνολο είναι το W' . Για μια συγκεκριμένη Web Service στο W' , οι προηγούμενοι χρόνοι εκτέλεσης καταγράφονται και όλοι οι χρόνοι εκτέλεσης ταξινομούνται σε αύξουσα σειρά. Χρησιμοποιούμε δύο πίνακες ET και r για αποθήκευση των εκτιμώμενων χρόνων εκτέλεσης των υποψηφίων Web Services και των πραγματικών χρόνων εκτέλεσης κατά τη στιγμή της επιλογής, αντίστοιχα. Προκειμένου να προσδιοριστούν οι αρχικές τιμές για τις εγγραφές του ET μπορούμε να χρησιμοποιήσουμε δύο στρατηγικές ανάλογα με το φόρτο δικτύου:

1. Έστω ότι ο ενδιαμέσος QoWS ρωτάει όλους τους κόμβους ταυτόχρονα για τον τρέχον φόρτο εργασίας, συλλέγει απαντήσεις, αντιστοιχίζει με τα αποθηκευμένα προφίλ και ορίζει τον εκτιμώμενο χρόνο εκτέλεσης στο κοντινότερο προφίλ σε σχέση με τη μετρική L_2 .
2. Ορίζουμε το ET[i] στο μέσο όλων των χρόνων εκτέλεσης που είναι αποθηκευμένοι στο προφίλ υπηρεσίας.

Έστω ότι το $\min$ δηλώνει τον ελάχιστο χρόνο εκτέλεσης. Αρχικά, το $\min$ παίρνει την ελάχιστη τιμή που αποθηκεύεται στο ET. Οι εγγραφές του r υπολογίζονται online σε χρόνο εκτέλεσης ρωτώντας κάθε κόμβο τις κατάλληλες παραμέτρους και χρησιμοποιώντας ξανά τη μετρική L_2 . Σημειώνουμε ξανά ότι θα μπορούσε να χρησιμοποιηθεί ένας μηχανισμός με δυνατότητα αποστολής μηνυμάτων multicast (ταυτόχρονης εκπομπής) για να ερωτηθούν συστάδες από κόμβους που υπάρχουν στο δίκτυο τοπικής περιοχής για γρηγορότερες απαντήσεις και εφόσον το επιτρέπει η δικτυακή υποδομή.

Για κάθε i , έχουμε υπολογίσει την καθυστέρηση δικτύου $L[i]$ που αντιστοιχεί στο χρόνο επικοινωνίας που απαιτείται για να επικοινωνήσει με τον server i . Έστω επίσης ότι το L δηλώνει το άθροισμα όλων των δικτυακών καθυστερήσεων.

Η επιλογή βασίζεται σε δύο γεγονότα. Πρώτον, εάν η συνολική καθυστέρηση δικτύου L , που απαιτείται για να ερωτηθεί κάθε κόμβος, είναι μικρότερη από την τρέχουσα εκτίμηση ελάχιστης εκτέλεσης, τότε μπορούμε να ξανα-υπολογίσουμε όλους τους χρόνους εκτέλεσης με την ελπίδα ότι μπορεί να βρούμε ένα νέο ελάχιστο. Εάν το L είναι μεγαλύτερο από τον ελάχιστο αναμενόμενο χρόνο εκτέλεσης, δε μας συμφέρει να ρωτήσουμε όλους τους κόμβους ελπίζοντας σε μια ακριβέστερη εκτίμηση, καθώς θα ξοδέψουμε περισσότερο χρόνο για να ρωτήσουμε παρά ακολουθώντας το καλύτερο σενάριο εκτέλεσης. Ο αλγόριθμος πρέπει επομένως να καθορίσει ένα κριτήριο παύσης. Συνεχίζει να αναζητά ενημερωμένους εκτιμώμενους χρόνους για όσο χρόνο η καθυστέρηση που έχει πληρωθεί μέχρι στιγμής συν την καθυστέρηση της ερώτησης του επόμενου κόμβου είναι μικρότερη από τον ελάχιστο χρόνο που εμφανίστηκε μέχρι στιγμής. Οποιαδήποτε επιπλέον βήματα δεν θα έχουν καμία σημασία, ακόμα και αν βρεθεί ένας μικρότερος πραγματικός χρόνος, καθώς έχουμε ήδη ξοδέψει περισσότερο χρόνο στην αναζήτηση μιας απάντησης απ' ότι στην εκτέλεση της Web Service με ελάχιστο χρόνο εκτέλεσης.

Ο τελικός αλγόριθμος επιλογής ADAPTIVE_SELECTION παρουσιάζεται στον Αλγόριθμος 2. Ο υπολογισμός της πιο ενημερωμένης εκτίμησης εκτέλεσης παρουσιάζεται στο Αλγόριθμος 2 με την εντολή “compute” (γραμμές 6 και 9).

Σημειώνουμε σε αυτό το σημείο ότι ο αλγόριθμός που παρουσιάζεται μεροληπτεί προσωρινά υπέρ συγκεκριμένων ιστοτόπων που παρέχουν Web Services, καθώς είναι βέλτιστοι σε σχέση με συγκεκριμένα κριτήρια. Παρόλα αυτά, η κατάσταση αλλάζει κατά τη διάρκεια της εκτέλεσης εάν συγκεκριμένοι κόμβοι λαμβάνουν περισσότερη κίνηση και επομένως απαντούν με μεγαλύτερους χρόνους εκτέλεσης.

Online Questions Adaptive Selection

Input: A Web Service description request WSD and a set of candidates W'

Output: min, the WS with best expected execution time

```

1.   if (  $L < \min$  )
      (* Αν ο ελάχιστος χρόνος εκτέλεσης είναι μεγαλύτερος του κόστους
        τότε ρώτησε όλους τους κόμβους *)
2.   ask all nodes for their current workload, compute new
    estimations for each candidate and store them to ET;
3.    $\min \leftarrow$  the new minimum execution time estimate;
4.   else
5.      $j \leftarrow 1$ ;
6.     compute  $r[1]$ ;
7.      $\min \leftarrow r[1]$ ;
8.   while (  $\sum_{i=1}^j L[i] + L[j+1] < \min$  )
9.     compute  $r[j+1]$ ;
10.    if (  $r[j+1] < \min$  )
11.       $\min \leftarrow r[j+1]$ ;
12.    end-while
13.  end-if
14.  return min;
```

Αλγόριθμος 2: Αλγόριθμος online επιλογής

Όπως δίνεται εδώ ο αλγόριθμος περιγράφει τόσο την περίπτωση στην οποία έχουμε τον απαραίτητο χρόνο να επικοινωνούμε με όλους τους κόμβους (γραμμή 2) όσο και την περίπτωση όπου το κριτήριο παύσης αναγκάζει τον αλγόριθμο να σταματήσει.

Στην εναλλακτική περίπτωση των multicast ερωτημάτων ο Αλγόριθμος 2 μετατρέπεται ως εξής:

Multicast Questions Adaptive Selection

Input: A Web Service description request WS_D and a set of candidates W'

Output: min , the WS with best expected execution time

1. **if** ($L_{max} < min$)
2. ask all nodes for their current workload using a multicast mechanism, collect responses, compute new estimations for each candidate and store them to r ;
3. $min \leftarrow$ minimum expected execution time stored in vector r ;
4. **else**
5. for each of the candidate Web Services compute the average of all its execution times recorded in the service profile and store each average to the corresponding entry of ET ;
6. **end-if**
7. **return** min ;

Αλγόριθμος 3: Αλγόριθμος online επιλογής με χρήση multicast μηχανισμού

Ο υπολογισμός της πιο ενημερωμένης εκτίμησης εκτέλεσης παρουσιάζεται επίσης στον Αλγόριθμο 3 με την εντολή “compute” (γραμμές 2 και 5).

2.7 Εκτίμηση πολυπλοκότητας λόγω χρήσης της επιλογής contour

Η λογική της χρήσης της ευριστικής μεθόδου επιλογής με χρήση contour είναι διπλή:

1. Το σύνολο των υποψήφιων κόμβων Web Service μειώνεται σημαντικά στη μέση περίπτωση,
2. είναι εγγυημένο ότι το συνολικό κόστος επικοινωνίας λόγω δικτυακής καθυστέρησης μειώνεται.

Στη συνέχεια, περιγράφουμε εν συντομία μια ανάλυση της εκτίμησης απόδοσης, η οποία στοχεύει να διευκρινίσει τη διαίσθηση για την επιλογή contour. Επιπλέον η ανάλυση αυτή τεκμηριώνεται πειραματικά στην επόμενη παράγραφο.

Επομένως, έστω m ο αριθμός των υποψήφιων κόμβων Web Service, m' ο αριθμός των κόμβων που ανήκουν στο contour, t ο αριθμός των εγγραφών στον πίνακα που αποθηκεύει τα τρέχοντα διανύσματα προφίλ της συγκεκριμένης WS και έστω l το μέγιστο κόστος επικοινωνίας λόγω καθυστέρησης δικτύου. Τότε, η χρονική πολυπλοκότητα χωρίς την ευριστική μέθοδο επιλογής με χρήση contour θα είναι:

$O(m \log m)$ {για ταξινόμηση των υποψηφίων με χρήση π.χ., mergesort (βλ. [3], [9])} + $O(mt)$ {για προσπέλαση των κελιών στον πίνακα} + $O(l)$

Αντίστοιχα η χρονική πολυπλοκότητα όταν εφαρμόζεται η λογική της επιλογής contour θα είναι: $O(m \log m)$ {για τον υπολογισμό του contour} + $O(m' \log m')$ {για ταξινόμηση των υποψηφίων} + $O(m't)$ {για προσπέλαση των κελιών του πίνακα} + $O(l')$.

Καθώς το P είναι πάντα το πολύ 1 προκύπτει ότι η προεπεξεργασία της επιλογής contour είναι σίγουρα δικαιολογημένη όταν

$$mt > m' \log m' + m't.$$

Παρόλα αυτά πρέπει να σημειωθεί ότι:

- 1) στην γενική περίπτωση το P είναι αυστηρά μικρότερο από το 1, και
- 2) ο κυρίαρχος όρος στην επιλογή contour είναι το κόστος επικοινωνίας λόγω δικτυακής καθυστέρησης και καθώς η λογική της επιλογής contour εγγυάται ότι το κόστος μειώνεται, μπορεί να αναμένεται ότι η συνολική χρονική πολυπλοκότητα (η οποία καθορίζεται από το κόστος επικοινωνίας) θα μειωθεί.

Κατά συνέπεια κρίνεται σκόπιμη και επωφελής η επιλογή contour ως πρώτο βήμα της προσέγγισής μας.

2.8 Πειραματικά αποτελέσματα

Το τεχνολογικό περιβάλλον που χρησιμοποιείται για την υλοποίηση των προτεινόμενων μηχανισμών όπως επίσης και την αξιολόγησή τους είναι το πλαίσιο Microsoft .NET έκδοση 1.1 και η γλώσσα προγραμματισμού C# [13]. Οι πειραματικοί υπολογισμοί εκτελέστηκαν σε Windows 2003 Intel Pentium IV (3 GHz) και μνήμη RAM 1 GB. Επίσης χρησιμοποιήθηκε το σύστημα διαχείρισης βάσεων δεδομένων MS SQL 2000 με sp3. Η λύση που προτάθηκε δε χάνει τη γενικότητά της και μπορεί να υλοποιηθεί σε μια σειρά από άλλες πλατφόρμες που υλοποιούν web services. Οι κύριοι λόγοι που οδήγησαν σε αυτές τις επιλογές ήταν η προώθηση της συμβατότητας μέσω μιας στρατηγικής επιλογής της τεχνολογικής πλατφόρμας όπως επίσης και η προσπάθεια για διευκόλυνση της διαλειτουργικότητας με υπάρχουσες υπηρεσίες και υποδομές.

Ένα εταιρικό δίκτυο τηλεπικοινωνιακού φορέα είναι ένα γνωστό και ισοζυγισμένο δίκτυο όπου η μέση δικτυακή καθυστέρηση είναι πολύ μικρότερη από το μέσο χρόνο εκτέλεσης μιας Web Service που χειρίζεται μεγάλα σύνολα δεδομένων. Γενικά, αυτή είναι η περίπτωση των περισσότερων επιχειρησιακών διαδικασιών σε μεγάλα εταιρικά ή άλλα δίκτυα που χρησιμοποιούν Web Services οι οποίες χειρίζονται μεγάλα ποσά δεδομένων πριν υπολογίσουν την απάντησή τους. Τα αποτελέσματα που το επισημαίνουν αυτό παρουσιάζονται στη συνέχεια. Ωστόσο, συγκριτικά πειραματικά αποτελέσματα παρουσιάζονται και για περιπτώσεις όπου η καθυστέρηση δικτύου είναι κοντά στο χρόνο εκτέλεσης, πράγμα που δείχνει ότι η

λύση που προτείνουμε λειτουργεί αποτελεσματικά και στις δύο περιπτώσεις. Τέλος, για να αξιολογήσουμε μια συγκεκριμένη επίδραση του αλγοριθμικού βήματος της επιλογής contour, παρουσιάζουμε επίσης αντίστοιχα πειραματικά συγκριτικά αποτελέσματα.

2.8.1 Μεθοδολογία του πειράματος

Η μεθοδολογία των πειραμάτων παρουσιάζεται στην ακόλουθη λίστα από παραδοχές και λεπτομέρειες σχεδιασμού.

Το σχήμα εκτέλεσης QoWS ανά Web Service που χρησιμοποιήθηκε αφορά καταγραφές μετρητών για μια λίστα δώδεκα (12) διαφορετικών Web Services σε εξήντα έξη (66) διαφορετικά στιγμιότυπα (instances) και το καθένα αναπαριστά διαφορετικές συνθήκες/καταστάσεις της υποδομής.

- 1 Οι WS είναι ανεξάρτητες μεταξύ τους. Σύνθετες Web Services θα ήταν αποδεκτές στα πειράματα εφόσον δεν χρησιμοποιούσαν κάποια άλλη συμμετέχουσα WS.
- 2 Διαφορετικά χαρακτηριστικά QoWS συμμετέχουν στα σχήματα εκτέλεσης. Χωρίς να επηρεάζεται η γενικότητα, καθώς μπορούν να χρησιμοποιηθούν περισσότερες QoWS, υποθέτουμε ότι καταγράφονται πέντε (5) παράμετροι ποιότητας WS:
 - 2.1 η χρησιμοποίηση της CPU,
 - 2.2 η χρησιμοποίηση της μνήμης,
 - 2.3 λειτουργίες I/O δίσκου,
 - 2.4 ο αριθμός των διαδικασιών της ίδιας εφαρμογής και
 - 2.5 ο αριθμός των διαδικασιών συστήματος που τρέχουν στην υποδομή

Όλες οι παράμετροι κανονικοποιήθηκαν για να διευκολύνουν τους υπολογισμούς ως εξής: $QoWS \in [0, 1]$.

- 3 Η διάρκεια εκτέλεσης όπως επίσης και οι τιμές της καθυστέρησης δικτύου κυμαίνονται ως εξής: χρόνοι εκτέλεσης WS, $ET \in [10, 3600]$ και δικτυακή καθυστέρηση, $nt \in [1, 60]$.
- 4 Η ίδια λίστα WS εισάχθηκε σε μια υλοποίηση UDDI καταλόγου (UDDI υπηρεσία σε Windows 2003 Server 10). Όλες οι WS αναγνωρίζονται με βάση το μοναδικό Web Service κλειδί τους το οποίο χρησιμοποιείται για να επιστρέψει στον καταναλωτή την κατάλληλη αναφορά σε WSDL για τη σύνδεση.
- 5 Επιλέγονται τυχαία αιτήματα κατανάλωσης Web Service. Τα τυχαία αιτήματα υπηρεσίας δεν επηρεάζουν το αποτέλεσμα της επιλογής καθώς θα είχαμε τα ίδια αποτελέσματα ακόμα και αν τα αιτήματα ήταν συγκεκριμένης κατανομής. Επιπλέον, η δημιουργία τυχαίων αιτημάτων εγγυάται ότι ο αλγόριθμος δεν

επηρεάζεται από συγκεκριμένες επιχειρησιακές διαδικασίες, από το περιεχόμενο δηλαδή του αιτήματος. Τόσο η UDDI υλοποίηση όσο και ο προτεινόμενος αλγόριθμος, κάτω από τις ίδιες συνθήκες/καταστάσεις της υποδομής, επέστρεψαν υποψήφια στιγμιότυπα επιλογής WS με την ίδια λειτουργικότητα.

Set ID	Χρόνοι Εκτέλεσης	Χρόνοι Δικτύου	Μέση απόκλιση μεταξύ επιλογής UDDI και online επιλογής
WS set 1	10-60	1-30	+5,29
WS set 2	30-320	1-30	+33,42
WS set 3	60-600	1-30	+69,18
WS set 4	180-3600	10-60	+399,85

Πίνακας 6: Πειραματικά αποτελέσματα ομαδοποιημένα σε σύνολα

Σχήμα 13: Πειραματικά αποτελέσματα με δύο σύνολα WS σύντομα σε διάρκεια εκτέλεσης και δικτυακής καθυστέρησης.

Σχήμα 14: Πειραματικά αποτελέσματα με δύο σύνολα WS μεγάλης διάρκειας εκτέλεσης και δικτυακής καθυστέρησης

Στη συνέχεια παρουσιάζουμε συγκριτικά διαγράμματα τα οποία απεικονίζουν τα αποτελέσματα χρησιμοποιώντας δύο άξονες:

- ο άξονας των y απεικονίζει το χρόνο εκτέλεσης της WS που επιλέχθηκε από κάθε μέθοδο και
- ο άξονας των x απεικονίζει τα ερωτήματα για διαφορετικές WS τυχαίας επιλογής.

Τα διαγράμματα στο σχήμα 13 και το σχήμα 14 παρουσιάζουν τα αποτελέσματα που επιστρέφει η τυπική (χωρίς QoWS) αναζήτηση στο UDDI σε σύγκριση με τα

αποτελέσματα του προτεινόμενου αλγορίθμου που διασφαλίζουν QoWS. Για να διευκολυνθεί η συζήτηση των πειραματικών αποτελεσμάτων, τα έχουμε συγκεντρώσει σε τέσσερις κατηγορίες. Τα αποτελέσματα κάθε κατηγορίας έχουν παρόμοια χαρακτηριστικά εκτέλεσης. Τα αποτελέσματα διαφορετικών κατηγοριών έχουν διαφορετικά όρια διακύμανσης στο χρόνο εκτέλεσης και/ή τις καθυστερήσεις δικτύου για να προσπελάσουν μια WS. Τα διαφορετικά σύνολα παρουσιάζονται στον πίνακα 6.

2.8.2 Αξιολόγηση των μετρήσεων

Στο σχήμα 13 και το σχήμα 14, η διακεκομμένη γραμμή απεικονίζει το χρόνο εκτέλεσης των WS που επιλέγονται από τον αλγόριθμό μας σε σύγκριση με το ET των WS που βασίζεται σε μια UDDI επιλογή. Καθώς η διακεκομμένη γραμμή είναι στις περισσότερες περιπτώσεις κάτω από τη γραμμή του UDDI, φαίνεται ότι η προτεινόμενη επιλογή καταλήγει γενικά σε καλύτερους χρόνους εκτέλεσης. Ειδικότερα, ο πίνακας 6 παρουσιάζει τη μέση απόκλιση των αποτελεσμάτων για κάθε ένα από τα τέσσερα σύνολα WS. Τα πειραματικά αποτελέσματα υποδεικνύουν ότι σε κάθε σύνολο αντιστοιχεί μια βελτίωση χρόνου απόκρισης WS σε ποσοστό 25,29%, 42,71%, 57,89%, και 66,47% αντίστοιχα. Το μέσο κέρδος όλων των συνόλων είναι 48,09%. Τα πειράματα αποκάλυψαν ότι ο προτεινόμενος αλγόριθμος επιστρέφει βελτιωμένα αποτελέσματα όταν αυξάνεται ο χρόνος εκτέλεσης και ο χρόνος δικτύου μιας WS. Τα αποτελέσματα του σχήμα 16 ενισχύουν την υπόθεση ότι ο προτεινόμενος αλγόριθμος παρέχει καλύτερο χρόνο απόκρισης μέσω της online επιλογής WS. Ο προτεινόμενος αλγόριθμος παρέχει αποτελεσματικά μεγιστοποιημένες επιλογές απόδοσης WS και επίσης επιβεβαιώνει την διαθεσιμότητα της WS.

2.8.3 Αξιολόγησης της επίδρασης του αλγορίθμου contour

Για να αποδείξουμε την επίδραση του αλγορίθμου αρχικής επιλογής με χρήση του contour (βήμα 1, Σχήμα 8), αξιολογούμε το κέρδος χρησιμοποιώντας συγκριτικά αποτελέσματα σε διαφορετικές περιπτώσεις. Στο σχήμα 15 η επιλογή εκτελείται σε Web Services των συνόλων 1 και 2. Το σχήμα 15 αναπαριστά τη συμπεριφορά του συνολικού χρόνου εκτέλεσης μετά την online διαδικασία επιλογής της παραγράφου 2.6 με και χωρίς το αρχικό βήμα κλαδέματος (pruning) της παραγράφου 2.5. Η σύγκριση έδειξε ότι η επιλογή με βάση το contour παρέχει ένα αρχικό κατώφλι για να μειώσει το χώρο αναζήτησης σε αυτές τις WS που έχουν ελαχιστοποιήσει τη δικτυακή απόσταση (απόκριση) και να μεγιστοποιήσει τη λειτουργικότητα. Ειδικότερα, στο σύνολο WS 1 όπου η δικτυακή απόσταση είναι κοντά στο χρόνο εκτέλεσης, το κέρδος είναι 8,37%. Στη δεξιά πλευρά στο σχήμα 15, όπου χρησιμοποιείται το σύνολο WS 2, καταγράφεται ένα ακόμα μεγαλύτερο κέρδος 33,57%. Αυτά τα πειράματα επικυρώνουν ότι το αρχικό βήμα κλαδέματος μειώνει σημαντικά το χώρο αναζήτησης WS, καταλήγοντας επομένως κατά μέσο όρο σε μεγαλύτερη βελτίωση του συνολικού χρόνου εκτέλεσης.

Σχήμα 15: Συγκριτική αξιολόγηση της επίδρασης της επιλογής contour στα σύνολα WS 1 και 2

2.9 Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις

Σε αυτό το κεφάλαιο ασχοληθήκαμε με το πρόβλημα της εύρεσης αποτελεσματικών λύσεων για την επιλογή και δέσμευση μιας Web Services, σύμφωνα με προκαθορισμένες λειτουργικές απαιτήσεις εξασφαλίζονται παράλληλα απαιτούμενα ποιοτικά χαρακτηριστικά. Προτείνουμε μια νέα συνολική προσέγγιση με τις εργασίες μας [8] και [6] για το πρόβλημα της δυναμικής επιλογής Web Services και συγκεκριμένα δώσαμε δύο αποδοτικές αλγοριθμικές προσεγγίσεις (βλ. Αλγόριθμος 2, Αλγόριθμος 3). Τονίζεται ότι οι μέθοδοι που εισήχθησαν πέρα από τα λειτουργικά χαρακτηριστικά που απαιτούνται, λαμβάνουν υπόψη παράγοντες ποιότητας υπηρεσίας (QoS) (μη λειτουργικά χαρακτηριστικά) και τους μεγιστοποιούν με σκοπό να επιλέξουν δυναμικά κατά τη στιγμή της προσπάθειας κατανάλωσης την καταλληλότερη Web Service.

Σχήμα 16: Το κέρδος του χρόνου απόκρισης της WS στα πειραματικά αποτελέσματα

Εκτός από τη θεωρητική υποστήριξη, παρουσιάστηκε σειρά πειραματικών μελετών που εκτελέστηκαν χρησιμοποιώντας ως δοκιμή διαδικασίες από ελληνικό φορέα

τηλεπικοινωνιών που διαχειρίζονται μεγάλου όγκου δεδομένα για την παραγωγή αναφορών και αναλύσεων για τους τελικούς πελάτες, τους επιχειρηματικούς συνεργάτες καθώς και για ενδοεπιχειρησιακές αναφορές και παρακολούθηση δεικτών. Τα αποτελέσματα που παρουσιάστηκαν παραπάνω δείχνουν ότι στις περιπτώσεις WS που διαχειρίζονται μεγάλου όγκου δεδομένα σε ένα δικτυακό πληροφοριακό σύστημα, ο αποδοτικός αλγόριθμος επιλογής που εισάγαμε είναι πιο αποτελεσματική και αποδοτική λύση σε σύγκριση με την υπάρχουσα μέθοδο αναζήτησης στο UDDI μιας και διασφαλίζει τα ποιοτικά χαρακτηριστικά. Στην προτεινόμενη λύση ο συνολικός χρόνος εκτέλεσης που προκύπτει από τον υπολογισμό του βέλτιστου σχήματος εκτέλεσης είναι μικρότερος από το χρόνο που προκύπτει από την επιλογή του καθιερωμένου UDDI. Συγκεκριμένα τα συγκριτικά αποτελέσματα παρουσιάζουν βελτίωση του χρόνου απόκρισης έως και 66,47%. Το μέσο κέρδος σε όλες τις περιπτώσεις είναι 48,09% όπως φαίνεται στο σχήμα 15. Τα πειράματα αποκάλυψαν επίσης ότι ο προτεινόμενος αλγόριθμος επιστρέφει βελτιωμένα αποτελέσματα όσο αυξάνεται ο χρόνος εκτέλεσης ή ο χρόνος καθυστέρησης δικτύου για την πρόσβαση και εκτέλεση μιας WS.

Η επίτευξη βελτιωμένης απόδοσης μετά την ενοποίηση μηχανισμού multicast για τη συλλογή τιμών ιδιοτήτων από τους κόμβους βελτιώνει ακόμη περισσότερο το αναμενόμενο κέρδος μιας και οι ιδιότητες συλλέγονται για πολλές μαζί περιπτώσεις WS ταυτόχρονα. Κατά συνέπεια σε περιπτώσεις αποδοτικών δικτύων που δεν έχουν πρόβλημα φόρτου και δίνουν πολύ μικρές δικτυακές καθυστερήσεις προτείνουμε τη δεύτερη εκδοχή του αλγορίθμου μας.

Μελλοντικά βήματα περιλαμβάνουν την εξέταση της απόδοσης της προτεινόμενης μεθοδολογίας με διαφορετικές λύσεις σχετικά με το πρώτο βήμα του κλαδέματος (pruning) πέρα από την προτεινόμενη επιλογή contour. Στην περίπτωση αυτή σκοπός είναι να πραγματοποιηθεί μια μελέτη των παραγόντων εκείνων που μπορούν να μειώσουν το εύρος της δυναμικής αναζήτησης κατά τη στιγμή της κατανάλωσης ώστε να βελτιστοποιηθεί στη συνέχεια η online επιλογή.

Ένα επιπλέον βήμα είναι η αξιοποίηση της προτεινόμενης στρατηγικής για την επιλογή WS με την υποστήριξη ομότιμων δικτυοκεντρικών λύσεων όπως αυτών που περιγράφονται στο επόμενο κεφάλαιο. Η ευρεία διάδοση των ομότιμων δικτύων μπορεί να αποτελέσει τη βάση για ένα νέο κατάλογο εγγραφής WS πλήρως αποκεντριοποιημένο. Κατά συνέπεια η μελέτη της δυνατότητας ολοκλήρωσης της τεχνικής μας στην περίπτωση αυτών των δικτύων ώστε να υποστηρίζεται δυναμική επιλογή κατά τη στιγμή της κατανάλωσης θα μπορούσε να πολλαπλασιάσει το κέρδος ως προς τα ποιοτικά χαρακτηριστικά και να μειώσει καταλυτικά τους χρόνους αναμονής εκτέλεσης.

Μία ακόμη κατεύθυνση στην οποία οι αλγόριθμοι επιλογής Web Service του κεφαλαίου μπορούν να οδηγήσουν σε περαιτέρω βελτιώσεις είναι στο πρόβλημα της βελτίωσης αλληλουχίας από WS στα πλαίσια μιας ροής επιχειρησιακών διαδικασιών. Στην περίπτωση αυτή διακρίνονται δύο υποκατηγορίες προβλημάτων.

Η πρώτη περιλαμβάνει τις εμφωλευμένες WS (component WS) όπου μία διαδικασία WS είναι η διεπαφή απλά από ένα σύνολο άλλων WS που εκτελούν μια ολοκληρωμένη λειτουργία. Στις εμφωλευμένες WS, οι δικτυακές καθυστερήσεις είναι μικρές γιατί οι WS βρίσκονται συνήθως σε μικρής κλίμακας δικτυακά περιβάλλοντα. Στην περίπτωση αυτή είναι χρήσιμο να εξεταστεί η δυνατότητα βελτίωσης των αποδόσεων για δυναμική επιλογή με ποιοτικά χαρακτηριστικά με την τεχνική μας που διαθέτει multicast μιας και το δικτυακό κόστος είναι χαμηλό.

Η δεύτερη ομάδα ενδιαφέροντος αφορά ροές που αποτελούν αλληλουχίες WS μιας επιχειρησιακής διαδικασίας. Στις περιπτώσεις αυτές κάθε βήμα της ροής χρειάζεται να επιλεγεί η βέλτιστη δυνατή ποιοτικά WS που ταιριάζει για τη λειτουργία του συγκεκριμένου βήματος. Η επέκταση των τεχνικών μας για να υποστηρίξουν πολλαπλά βήματα θα έδινε μια αποδοτική λύση στο πρόβλημα αυτό.

Τέλος, μία σημαντική και πολλά υποσχόμενη ερευνητική κατεύθυνση στην περιοχή περιλαμβάνει την ενσωμάτωση γνώσης από οντολογίες. Η σημασιολογική συσχέτιση των διαδικασιών σε επιχειρησιακό επίπεδο μπορεί να αντικατοπτριστεί σε γνώση για τα βήματα και την επικοινωνία ανάμεσα σε WS. Ως αποτέλεσμα είναι ενδιαφέρον να βρεθούν τεχνικές ώστε να βελτιώνουν τις λειτουργικές αναζητήσεις με βάση μια περισσότερο ασαφή περιγραφή της λειτουργίας αλλά και των ποιοτικών χαρακτηριστικών που είναι επιθυμητά.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ 2

1. Austin, D., Barbir, A., Ferris, C. and Garg, S. (eds.): Web Service Architecture Requirements. W3C Working Group Notes (2004) <http://www.w3.org/TR/wsa-reqs>
2. Ballinger, K., 2004. ".NET Web Services: Architecture and Implementation with .NET", Microsoft .NET Development Series, Addison-Wesley.
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2001. Introduction to Algorithms, second ed. The MIT Press/McGraw-Hill Book Company, Cambridge, MA/New York.
4. Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., White, B., 2002. Web engineering. J. Web Eng. 1 (1), 3–17.
5. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., 2004. WebService Discovery mechanisms: looking for a needle in a haystack? In: International Workshop on Web Engineering, Hypermedia Development and Web Engineering Principles and Techniques: Put them in use, in conjunction with ACM Hypertext 2004. Extended version accepted in the Journal of Web Engineering, Rinton Press to appear in 2006. Available from the URL: http://www.ht04.org/workshops/WebEngineering/HT04WE_Garofalakis.pdf.
6. Garofalakis, J., Sakkopoulos, E., Sirmakessis, S., Tsakalidis, A., "Integrating Adaptive Techniques with Web Services", IEEE International Conference on Information Technology: Coding & Computing (IEEE ITCC), April 28-30, 2003, Las Vegas, Nevada, USA.
7. Liu, Y., Ngu, A.H.H., Zeng, L., 2004. QoS computation and policing in dynamic Web Service selection. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (Eds.), Proceedings of the 13th international conference on World Wide Web-Alternate Track Papers & Posters, WWW 2004, New York, NY, USA, May 17–20, 2004. pp. 66–73.
8. Makris, C., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., "Efficient and adaptive discovery techniques of web services handling large data sets", in J. Systems and Software, Elsevier Science, corrected proof, in press.
9. Mehlhorn, K., 1984. Sorting and Searching Data Structures and Algorithms, vol. 1. Springer Verlag, Berlin.
10. Microsoft, 2003. Uddi, windows 2003 implementation. <<http://www.microsoft.com/windowsserver2003/technologies/webapp/uddi/default.mspx>>.
11. Microsoft, 2004a. Performance counters and objects. <http://microsoft.com/windowsxp/home/using/productdoc/en/SAG_MPmonperf_06.asp>.
12. Microsoft, 2004b. Performance monitoring, browsing counters. <http://msdn.microsoft.com/library/en-us/perfmon/base/getting_counter_information.asp>.
13. Microsoft, 2005. C Sharp Programming Language Specification. <<http://msdn.microsoft.com/library/en-us/csspec/html/CSharpSpecStart.asp>>.
14. Ouzzani, M., Bouguettaya, A., 2004. Efficient access to Web Services. IEEE Internet Comput. 8 (2), 34–44.
15. Ran, S., 2003. A model for Web Services discovery with QoS. SIGecom Exch. 4 (1), 1–10.
16. Sajjanhar, A., Hou, J., Zhang, Y., 2004. Algorithm for Web Services matching. In: Advanced Web Technologies and Applications, Proceedings of the 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, April 14–17, 2004. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (Eds.), Lecture Notes in Computer Science, vol. 3007. Springer, Berlin, pp. 291–300.
17. UDDI, 2004. UDDI Version 3.0.2. <<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>>.
18. W3C, 2003. SOAP W3C Recommendation Documents. <<http://www.w3.org/TR/SOAP>>.
19. W3C, 2004a. Web Service Description Language. <<http://www.w3.org/TR/wsdl>>.
20. W3C, 2004b. Web Services Architecture. <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>.

21. Yu, T., Lin, K.-J., 2004a. The design of QoS broker algorithms for QoS-capable Web Services. *Int. J. Web Service Res.* 1 (4), 33–50.
22. Yu, T., Lin, K.-J., 2004b. Service selection algorithms for Web Services with end-to-end QoS constraints. In: 2004 IEEE International Conference on Ecommerce Technology (CEC 2004), 6–9 July 2004, San Diego, CA, USA. pp. 129–136.
23. Yu, T., Lin, K.-J., 2005. A broker-based framework for QoS-aware Web Service composition. In: 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), 29 March–1 April 2005, Hong Kong, China. pp. 22–29.
24. Zhuge, H., 2002a. Clustering soft-devices in the semantic grid. *IEEE Comput. Sci. Eng.* 4 (6), 60–62.
25. Zhuge, H., 2002b. A knowledge grid model and platform for global knowledge sharing. *Expert Syst. Appl.* 22 (4), 313–320.
26. Zhuge, H., 2002c. Vega-kg: A way to the knowledge web. In: Proceedings of 11th International World Wide Web Conference (WWW2002). Available from: <<http://www2002.org/CDROM/poster/53.pdf>>.
27. Zhuge, H., Liu, J., 2004. Flexible retrieval of Web Services. *J. Syst. Softw.* 70 (1–2), 107–116.

3 ΑΝΑΖΗΤΗΣΗ ΥΠΗΡΕΣΙΩΝ ΙΣΤΟΥ ΜΕ ΔΙΚΤΥΟΚΕΝΤΡΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ

Πρόλογος – Σύνοψη Κεφαλαίου

Η τεχνολογία πλαίσιο των Web services αποτελεί μια πολλά υποσχόμενη και πλέον καθιερωμένη λύση για την κατασκευή δικτυακών πληροφοριακών συστημάτων. Σε αυτό το κεφάλαιο, προτείνουμε δύο νέες προσεγγίσεις που αντιμετωπίζουν το πρόβλημα της αναζήτησης υπηρεσιών ιστού με αποδοτικές δικτυοκεντρικές προσεγγίσεις (netcentric). Ονομάζουμε την πρώτη προσέγγιση NIPPERS [17] (Network of InterPolated PeERS for Web Service Discovery) και είναι μια λύση που αντιμετωπίζει το πρόβλημα της ανακάλυψης Web services χρησιμοποιώντας μια επεκτάσιμη και σταθερή (robust) δομή δεικτοδότησης δεδομένων για δικτυοκεντρικά συστήματα. Στο ίδιο κεφάλαιο παρουσιάζουμε μια εναλλακτική λύση το BDT-GRID (Balanced Distributed Tree) [10] που αντιμετωπίζει αποδοτικότερα το ίδιο πρόβλημα κατά την επανοργάνωση της δομής.

Το παρόν κεφάλαιο έχει την ακόλουθη δομή: Στην ενότητα 3.1 μια σύντομη εισαγωγή και στην 3.2 παρουσιάζεται το πρόβλημα και σχετικές εργασίες (αναλυτική βιβλιογραφία δείτε στο κεφάλαιο 1). Στο 3.4 παρουσιάζεται αρχικά η λύση NIPPERS [17]. Στην ενότητα 3.4 δίνονται στοιχεία του στατικού interpolation search tree από τη βιβλιογραφία. Στην ενότητα 3.5 παρουσιάζεται η δομή δεδομένων NIPPERS. Ακολουθεί στην ενότητα 3.6 η ανάλυση της ανεκτικότητάς της σε σφάλματα. Στην ενότητα 3.7 σημειώνεται η δυνατότητα που δίνει το NIPPERS στην εξασφάλιση ποιότητας υπηρεσίας WS. Η παρουσίαση του αλγορίθμου αυτού κλείνει στην ενότητα 3.8 με την αξιολόγηση της δομής σε σχέση με γνωστά ομότιμα δίκτυα. Ακολουθεί στην ενότητα 3.9 η παρουσίαση της εναλλακτικής προτεινόμενης λύσης BDT-GRID [10]. Στην ενότητα 3.10 παρουσιάζονται σύντομα στοιχεία για ιεραρχικούς δικτυοκεντρικούς αλγορίθμους. Ακολουθεί η ενότητα 3.11 με την αρχιτεκτονική, την παρουσίαση και την ανάλυση του DBT-GRID. Στην ενότητα 3.12 δίνεται η αξιολόγηση της δομής σε σχέση με γνωστά ομότιμα δίκτυα. Το κεφάλαιο ολοκληρώνεται με την ενότητα 3.13 που δίνει τη συνεισφορά μας, τα συνολικά συμπεράσματα, συγκριτικό πίνακα των λύσεων που προτείναμε σε σχέση με γνωστές δικτυοκεντρικές προσεγγίσεις καθώς και μελλοντικές κατευθύνσεις έρευνας.

3.1 Εισαγωγή

Η παρούσα διδακτορική διατριβή ασχολείται συνολικά με αλγορίθμους διαχείρισης και αναζήτησης υπηρεσιών και πληροφοριών σε δικτυοκεντρικά περιβάλλοντα (netcentric), τα οποία περισσότερο αυστηρά ορίζονται [35] ως «τα δικτυακά κατανεμημένα περιβάλλοντα όπου οι εφαρμογές και τα δεδομένα διακινούνται κυρίως μέσω εξυπηρετητών και κόμβων αποκεντριοποιημένων – ομότιμων δικτύων. Αυτό βρίσκεται στην αντίθετη όχθη από την αντιμετώπιση όπου ισχυροί υπολογιστές στηρίζονται σε τοπικά μέσα και πηγές. Κατά μία άποψη τα δικτυακά περιβάλλοντα αυτά μοιάζουν με τα πρώτα συστήματα mainframe και τερματικών. Ωστόσο υπάρχουν κύριες και σημαντικές διαφοροποιήσεις. Τα δικτυακά πληροφοριακά συστήματα και εφαρμογές βασίζονται σε υπηρεσίες και πληροφορίες που βρίσκονται και εκτελούνται σε διαφορετικά τεχνολογικά περιβάλλοντα (πλατφόρμες), σε κινητά

δεδομένα που προσπελάνονται μέσω ταχύτατου δικτύου και σε χαμηλού κόστους πελάτες-συσκευές για την τοπική επεξεργασία». Μία βασικές προσεγγίσεις δικτυοκεντρικών λύσεων είναι οι Web Services (WS), «το όραμα μιας χαλαρούς σύζευξης «loosely coupled» αλληλεπίδρασης μεταξύ λειτουργικών υποσυστημάτων (components), προγραμμάτων, και εφαρμογών», οι οποίες αποτελούν πλέον κατευθυντήριο οδηγό για την επιχειρησιακή ολοκλήρωση [5]. Παρόλο που όπως έχουμε σημειώσει τα WS είναι από τα νεότερα μέλη της μηχανικής παγκόσμιου ιστού (web engineering) [8], ήδη τα υπάρχοντα πρωτόκολλα Ιστού και τα πρόσφατα εισαχθέντα ανοικτά πρότυπα XML παρέχουν ένα ισχυρό πλαίσιο για το σχεδιασμό, την εφαρμογή και τη δημοσίευση WS. Πολλά ελπιδοφόρα βήματα έχουν γίνει ώστε η τεχνολογία των WS να γίνει μια παγκοσμίως υποστηριζόμενη λύση για αλληλεπίδραση μεταξύ εφαρμογών (A2A – application to application).

Στο κεφάλαιο αυτό, εστιάζουμε στους αλγόριθμους και τεχνικές διαχείρισης και αναζήτησης Web Services που βοηθούν την ανακάλυψη των υπηρεσιών οι οποίες ικανοποιούν ένα συγκεκριμένο επιχειρησιακό και λειτουργικό προφίλ. Αυτές οι πληροφορίες πρέπει να είναι διαθέσιμες με έναν προσιτό τρόπο για επεξεργασία από μηχανές (machine processing). Ωστόσο, η αληθινά δυναμική ανακάλυψη των Web services δεν έχει ακόμα επιτευχθεί.

Έως σήμερα, η πλειοψηφία των Web services αναπτύσσεται κυρίως για ενδο-επιχειρηματική χρήση. Ωστόσο, οι Web services προβλέπεται να υιοθετηθούν ευρύτερα στα επόμενα έτη, επιτρέποντας πολύ ευρύτερη ενδοεπιχειρησιακής και διεπιχειρησιακής ολοκλήρωσης. Κατά συνέπεια, θα υπάρξει μια αυξανόμενη ζήτηση για αυτοματοποιημένη ανακάλυψη υπηρεσιών, η οποία θα διευκολύνει την περαιτέρω αλληλεπίδραση των Web services με ακόμα λιγότερη ανθρώπινη προσπάθεια [5].

Προς το παρόν, οι Web services διαφημίζονται κυρίως στους καταλόγους. Αυτοί οι κατάλογοι είναι βασισμένοι στο πρότυπο Universal Description, Discovery and Integration (UDDI) [31] και αποτελούν το κυρίαρχο τεχνολογικό περιβάλλον για την ανακάλυψη Web services (WS Discovery). Αναμένεται να δημιουργηθεί ένας μεγάλος αριθμός εφαρμογών καταλόγων/αποθηκών για διά-επιχειρησιακή επικοινωνία, κάθε μια από τις οποίες θα εστιάζει στην καταχώρηση των υπηρεσιών ενδιαφέροντος στις αντίστοιχες ομάδες. Κατά συνέπεια, εκτός από τον μεγάλο αριθμό διαφορετικών web services που μπορούν να επιλεγούν, υπάρχει επίσης η πρόκληση της επιλογής από έναν μεγάλο αριθμό καταλόγων κατά τη διάρκεια της ανακάλυψης και της επιλογής υπηρεσιών. Επιπλέον, η επερχόμενη απαίτηση για ποιότητα της WS (QoWS) εισάγει ακόμη περισσότερες διαδικασίες ελέγχων και επιλογής.

3.2 Περιγραφή του Προβλήματος

Οι Web Services είναι μία σχετικά νέα τεχνολογία και ίσως ο τομέας της ανακάλυψης Web Services μπορεί να θεωρηθεί ακόμα νεότερος. Ωστόσο, πολλή προσπάθεια έχει ήδη αφιερωθεί στην συγκεκριμένη επιστημονική περιοχή. Μια πολλά υποσχόμενη προσέγγιση για την ανακάλυψη Web Services είναι να θεωρούνται οι κόμβοι δικτύων

ως κόμβοι (peers) που διαμοιράζονται πληροφορίες και είναι σε θέση να ρωτήσουν άλλους κόμβους. Το πρόβλημα μας απασχολεί και αντιμετωπίζουμε στο παρόν κεφάλαιο είναι η ανακάλυψη WS σε αποκεντριοποιημένα δικτυακά περιβάλλοντα (δικτυοκεντρικά πληροφοριακά περιβάλλοντα).

Στην κατεύθυνση επίλυσης του προβλήματος αυτού, έχουν προταθεί ορισμένες δομημένες P2P αρχιτεκτονικές που βασίζονται σε κατανεμημένους hash πίνακες – distributed hash tables- (DHTs) και έχουν γίνει πολύ δημοφιλείς στις κοινότητες του κατανεμημένου υπολογισμού και δικτύωσης. Οι περισσότερες P2P λύσεις βασίζονται σε τέτοια DHT-based συστήματα. Συνολικά, αυτά τα συστήματα παρέχουν αποδοτική επεξεργασία των διαδικασιών εντοπισμού έτσι ώστε, δεδομένου ενός ερωτήματος με ένα κλειδί αντικειμένου (object key), εντοπίζουν (δρομολογούν το ερώτημα σε) τον όμοιο κόμβο που αποθηκεύει το αντικείμενο. Τα DHT-based συστήματα χαρακτηρίζονται επίσης ως δομημένα P2P συστήματα επειδή, γενικά, στηρίζονται στις αναζητήσεις ενός κατανεμημένου hash πίνακα, ο οποίος επιβάλλει μια εικονική δομή στο σύστημα που προκύπτει από τον τρόπο που οι κόμβοι καθορίζουν τους γείτονές τους. Μη δομημένα P2P συστήματα όπως το Gnutella [25], το MojoNation [4] κ.α. δεν δημιουργούν τέτοιες δομές, δεδομένου ότι οι γείτονες των κόμβων καθορίζονται ad-hoc. Υπάρχουν διάφορες P2P DHT αρχιτεκτονικές όπως το Chord [30], το CAN [24], το Pastry [27], το Tapestry [34] και το P-Grid [1]. Το CAN και το Chord είναι τα πιο ευρέως χρησιμοποιούμενα, ενώ το P-Grid είναι μια επεκτάσιμη δομή πρόσβασης βασισμένη σε ένα εικονικό κατανεμημένο δέντρο αναζήτησης. Χρησιμοποιεί τεχνικές τυχαιότητας για να δημιουργήσει και να διατηρήσει τη δομή προκειμένου να παρέχει αποκεντριοποίηση.

Το Speed-R [29] είναι ένα P2P σύστημα αποθήκευσης και ανάκτησης WS βασισμένο σε οντολογία. Οι κατάλογοι ανατίθενται σε ορισμένους κόμβους στο P2P υποσύστημα, οι οποίοι στη συνέχεια χωρίζονται σύμφωνα με τη περιοχή εφαρμογής τους. Κάθε περιοχή εφαρμογής (domain) αναπαρίσταται χρησιμοποιώντας μια οντολογία. Η P2P υποδομή Speed-R είναι βασισμένη στην εφαρμογή JXTA. Η αρχιτεκτονική του Speed-R σχεδιάστηκε με βάση την υπόθεση ότι σε κάθε κόμβο ανατίθεται ένας ρόλος, όπως για παράδειγμα οι ενημερώσεις ελέγχου και η διάδοση τους ή η αποδοχή των νεοεισαχθέντων στοιχείων. Κατά συνέπεια αυτό το σύστημα μπορεί να πάσχει από τον κίνδυνο αποτυχίας μοναδικού σημείου (single point failure hazard).

Μία ακόμη τεχνική ανακάλυψης Web Services με βάση Ομότιμα Δίκτυα (PWSD) παρουσιάστηκε στο [15]. Σε αυτήν, το πρωτόκολλο Chord P2P το οποίο χρησιμοποιείται ως overlay, αποτελείται από Κόμβους Υπηρεσίας (Service Peers-SP). Κάθε SP αντιστοιχίζεται σε αραιές Λογικές Μηχανές (Logical Machines) (Διαφορετικές μηχανές αντιστοιχούν στο ίδιο υλικό). Κάθε Λογική Μηχανή διατηρεί τις απαραίτητες διεπαφές για να αντιστοιχίσει και να αναζητήσει WS στο ομότιμο δίκτυο.

Αναλυτική σύγκριση και παρουσίαση των διαφορετικών διαθέσιμων τεχνικών αναζήτησης WS με χρήση ομότιμων δικτύων έχει παρουσιαστεί στην ενότητα 1.5.2.2 αλλά και συνολικά στο πρώτο κεφάλαιο της παρούσας διατριβής.

3.3 Παρουσίαση της προσέγγισης του NIPPERS

Αρχικά παρουσιάζουμε μια νέα αποδοτική δομή ανακάλυψης-αναζήτησης για Ανακάλυψη Web Services Ομότιμων δικτύων, που ονομάζουμε NIPPERS [17] (Network of InterPolated PeERS for Web Service Discovery). Το NIPPERS παρέχει υποστήριξη για την επεξεργασία:

(α) ερωτημάτων ακριβούς αντιστοίχισης της μορφής: «δεδομένου ενός κλειδιού, εντοπίστε τον κόμβο που περιέχει το κλειδί» και

(β) ερωτημάτων περιοχής της μορφής «δεδομένης μιας περιοχής κλειδιών, εντοπίστε τον κόμβο/κόμβους που περιέχουν τα κλειδιά που ανήκουν σε αυτήν την περιοχή».

Το NIPPERS είναι βασικά μια κατανεμημένη έκδοση της δενδρικής δομής Interpolation Search Tree [18]. Τα αποτελέσματα από τη θεωρητική ανάλυση δείχνουν ότι το κόστος επικοινωνίας των διαδικασιών ερώτησης και ενημέρωσης κλιμακώνονται διπλο-λογαριθμικά με τον αριθμό των κόμβων NIPPERS. Επιπλέον, το σύστημα είναι σταθερό (robust) όσον αφορά τις αποτυχίες, καλύπτοντας με αυτόν τον τρόπο τις απαιτήσεις QoWS.

3.4 Προαπαιτούμενα: Το Στατικό Interpolation Search Tree

Πριν παρουσιάσουμε τις λεπτομέρειες της προτεινόμενης λύσης NIPPERS, δίνεται μια προκαταρκτική εισαγωγή σχετικά με το static interpolation search tree. Αυτό το δέντρο είναι μια στατική, και βελτιωμένη έκδοση των δέντρων αναζήτησης που χρησιμοποιούνται στο [3]. Ένα δέντρο αυτού του είδους που αποθηκεύει n στοιχεία, μπορεί να χαρακτηριστεί πλήρως από τρεις μη μειούμενες (non decreasing) συναρτήσεις $H(n)$, $R(n)$ και $I(n)$, όπου το $H(n)$ δηλώνει το ύψος του δέντρου, το $R(n)$ δηλώνει τον out-degree της ρίζας και το $I(n)$ καθορίζεται με τους τύπους $I(n)=n \cdot g(H(n))$. Η συνάρτηση $g(n)$ πρέπει να ικανοποιεί τη συνθήκη:

$$\sum_{i=1}^{\infty} g(i) = \Theta(n)$$

Για να εγγυηθεί το ύψος του $H(n)$, πρέπει να βεβαιώσει ότι $n/R(n)=H^{-1}(H(n)-1)$. Τα παιδιά της ρίζας έχουν $n'=\Theta(n/R(n))$ φύλλα. Το ύψος τους θα είναι $H(n')=H(n)-1$, και το out-degree τους είναι $R(n')=H^{-1}(H(n)-1)/H^{-1}(H(n)-2)$, και $I(n')=n' \cdot g(H(n'))$. Γενικά, θεωρούμε έναν εσωτερικό κόμβο v σε βάθος i και υποθέτουμε ότι τα φύλλα n_i αποθηκεύονται στο υποδέντρο που έχει ρίζα το v . Τότε, ισχύει ότι:

$$R(n_i)=\Theta(H^{-1}(H(n)-i+1)/H^{-1}(H(n)-i)) \text{ και } I(n_i)=n_i \cdot g(H(n)-i).$$

Ο κόμβος v σχετίζεται με ένα πίνακα $REF[1..R(n_i)]$ στοιχείων δειγμάτων (ένα στοιχείο δείγμα για κάθε ένα από υποδέντρα του) και ένα πίνακα ID που αποθηκεύει

ένα σύνολο στοιχείων δειγμάτων που προσεγγίζουν την αντίστροφη συνάρτηση κατανομής. Με τη χρησιμοποίηση του πίνακα ID, μπορούμε να παρεμβάλουμε τον πίνακα REF για να καθορίσουμε το υποδέντρο στο οποίο θα συνεχιστεί η διαδικασία αναζήτησης.

Ειδικότερα, ο πίνακας ID για τον κόμβο v είναι ένας πίνακας $ID[1..m]$, όπου:

$m = I(n_i)$, με $ID[i] = j$ εάν $REF[j] < 1 + i(u-l)/m \leq REF[j+1]$,

όπου τα l και u είναι τα μέγιστα και ελάχιστα σημεία αντίστοιχα που αποθηκεύονται στο υποδέντρο που έχει ρίζα το v . Έστω ότι το x είναι το στοιχείο που ζητάμε. Για να παρεμβάλουμε το REF, υπολογίζουμε το δείκτη $j = ID[(m(x-l)/(u-l))]$, και στη συνέχεια ψάχνουμε τον πίνακα REF από το $REF[j+1]$ μέχρι να εντοπιστεί το κατάλληλο υποδέντρο. Για κάθε κόμβο, διατηρούμε ρητά τους δείκτες παιδιά και γονείς. Οι απαραίτητες πληροφορίες δεικτών μπορούν εύκολα να ενσωματωθούν στην κατασκευή του στατικού interpolation search tree.

Σχήμα 17: Το σύστημα NIPPERS από την εργασία μας [17]

3.5 Η δομή δεδομένων NIPPERS

Κάθε δικτυακός κόμβος στο NIPPERS λειτουργεί ως ένας κόμβος Web Service, ο οποίος όχι μόνο επιτρέπει την κατανάλωση WS, αλλά επίσης λειτουργεί ως ένας κόμβος στο P2P overlay δίκτυο. Η αρχιτεκτονική δικτύου του NIPPERS που υποστηρίζει την ανακάλυψη web service απεικονίζεται στο σχήμα 17.

Ο προτεινόμενος αλγόριθμος ανακάλυψης και επιλογής WS καθορίζει πώς να δρομολογηθούν τα ερωτήματα στους κόμβους οι οποίοι μπορούν να διευκολύνουν

τα αιτήματα κατανάλωσης υπηρεσίας. Στο NIPPERS ακολουθώντας το παράδειγμα των καταλόγων, ένα αίτημα WS περιγράφεται χρησιμοποιώντας XML σύμφωνα με τις περιγραφές WSDL που αποθηκεύονται στους ενδιάμεσους κόμβους WS (node peers). Άλλοι αλγόριθμοι δρομολόγησης και εντοπισμού, όπως το Chord, υποστηρίζουν μόνο ερωτήματα ακριβούς-αντιστοίχισης. Μια λύση είναι να παρασχεθούν ορισμένες επεκτάσεις για την υποστήριξη της βασισμένης σε XML υπό συνθήκες αντιστοίχισης. Στο NIPPERS, έχουμε λάβει υπόψη τέτοιες περιπτώσεις. Επιπλέον, το NIPPERS υποστηρίζει μερική αντιστοίχιση. Αυτό το είδος αντιστοίχισης είναι ιδιαίτερα χρήσιμο προκειμένου να παρέχεται εγγύηση για την ποιότητα της επιλογής της WS.

Το NIPPERS παρέχει μια δομή μορφής δέντρου επάνω στην οποία μπορεί να εκτελεστεί η βασισμένη σε κλειδιά αναζήτηση. Από την άποψη της χρήσης του εύρους ζώνης, η αναζήτηση κλιμακώνεται πολύ καλά, καθώς δεν πραγματοποιείτε καμία broadcasting ή άλλη δραστηριότητα κατανάλωσης εύρους ζώνης. Δεδομένου ότι όλες οι αναζητήσεις βασίζονται σε κλειδιά, υπάρχουν δύο δυνατότητες: είτε (α) κάθε κόμβος εφαρμόζει τον ίδιο αλγόριθμο, ο οποίος μεταφράζει μια λέξη κλειδί σε ένα δυαδικό κλειδί ή (β) μια άλλη υπηρεσία παρέχει το δυαδικό κλειδί. Αυτή η υπηρεσία δέχεται ερωτήματα με βάση λέξεις κλειδιά και μπορεί να απαντήσει με το αντίστοιχο κλειδί. Η δεύτερη προσέγγιση είναι πιο ακριβής. Είναι επίσης πιθανό να χρησιμοποιήσει μια πιο κεντριοποιημένη υλοποίηση μιας τέτοιας υπηρεσίας. Για το υπόλοιπο της κεφαλαίου, υποθέτουμε ότι το κλειδί είναι διαθέσιμο. Στις επόμενες παραγράφους περιγράφουμε έναν αλγόριθμο για την πρώτη περίπτωση. Επίσης υποθέτουμε ότι ένα σύνολο κλειδιών σε κάθε κόμβο διατηρεί μια καθολική σειρά.

3.5.1 Οι κόμβοι στο δίκτυο NIPPERS

Το NIPPERS [17] ως δέντρο αναζήτησης είναι ένα υπονοούμενο εικονικό Interpolation Search Tree T. Πιο συγκεκριμένα, η δομή αποτελείται από δύο επίπεδα όπου το κορυφαίο επίπεδο είναι ένα στατικό Interpolation Search Tree, και το κατώτατο επίπεδο είναι μια οικογένεια δοχείων (buckets). Η δομή διατηρείται εκτελώντας αυξητικά τις καθολικές ανακατασκευές. Έστω n_0 ο αριθμός των στοιχείων που αποθηκεύονται κατά την διάρκεια της πιο πρόσφατης ανακατασκευής. Μετά από αυτόν τον χρόνο, όταν ο αριθμός των αναπροσαρμογών υπερβαίνει το rn_0 , όπου το r είναι μια αυθαίρετη σταθερά, τότε αναδημιουργείται ολόκληρη η δομή δεδομένων. Έστω n ο αριθμός των αποθηκευμένων στοιχείων αυτή τη στιγμή. Μετά από την αναδημιουργία, ο αριθμός των δοχείων είναι ίσος με $n/\ln n$, ενώ κάθε δοχείο περιλαμβάνει $\Theta(\ln n)$ στοιχεία. Κατά τη διάρκεια των επόμενων εισαγωγών/διαγραφών, το ανώτερο επίπεδο παραμένει άθικτο ενώ μόνο τα δοχεία επηρεάζονται. Σημειώνουμε ότι κατά τη διάρκεια αυτών των διαδικασιών ενημέρωσης, δεν είναι προφανής ο τρόπος που θα περιοριστεί το μέγεθος των κώδων. Στο [14], παρουσιάστηκε μια ιδέα γενικού επιστημονικού ενδιαφέροντος: η μοντελοποίηση των εισαγωγών/διαγραφών ως συνδυαστικό παιχνίδι δοχείων και σφαιρών (bins and balls), το μέγεθος κάθε δοχείου είναι αναμενόμενο w.h.p. $\text{polylog}(n)$, για τα στοιχεία που προέρχονται από μια ομαλή κατανομή [18].

```

Node p *NIPPERSgrid_search (p, k)
{
  int j;
  bool move_right=false;
  if ( p is responsible for this k)
    return p;
  if (someone else is responsible)
    check whether k is to the left or right of p;
  \\ έστω ότι το k είναι στα δεξιά του p\\
  p_next=father(p)
   $S_0 = |T_{p\_next}|$ 
  While (k > REFp_next[( $\sqrt{S_0}$ )]&&move_right=false) {
    p'=right_sibling of p_next;
     $S' = |T_{p'}|$ ;
    if (k <= REFp'[( $\sqrt{S'}$ )])
    {
      p_next= p';
      move_right=true;
    }
    else
      p_next=father (p_next);
     $S_0 = |T_{p\_next}|$ 
    host = send_search(p_next, k);
  }
  While ( p_next is not a key_host)
  {
    j=search (k,p_next);
    \\ To search (key, node) δηλώνει τη διαδικασία
    \\που επιστρέφει μια θέση ακεραίου j
    \\η οποία δείχνει τον κατάλληλο απόγονο στον οποίο
    \\πρέπει να συνεχίσουμε την περαιτέρω αναζήτηση
    p_next=&REF[j];
    host = send_search(p_next, k);
  }
  p=p_next;
  return p;
}

```

Αλγόριθμος 4: Ψευδο-κώδικας για τον αλγόριθμο αναζήτησης NIPPERS

Στη συνέχεια, θα δείξουμε πώς αυτή η δομή δεδομένων μπορεί εικονικά να εφαρμοστεί σε ένα ομότιμο δίκτυο. Έστω ότι το p_i δηλώνει έναν ενδιαμέσο κόμβο (peer node). Διακρίνουμε μεταξύ των κόμβων-φύλλων (leaf_peer) και των ενδιαμέσων κόμβων (node_peers): Εάν το p_i είναι ένας κόμβος-φύλλο τότε διατηρεί ένα σύνολο S_i των ταξινομημένων k-bit δυαδικών κλειδιών $k_i = b_1...b_k$, όπου το k είναι λιγότερο ή ίσο με το n_i , για κάποια δεσμευμένη σταθερά n_i που είναι η ίδια για όλα τα p_i . Αυτό το ταξινομημένο σύνολο κλειδιών δείχνει το διάστημα κλειδιών (key space) για το οποίο ένας κόμβος είναι αρμόδιος και είναι ισοδύναμο με το σύνολο στοιχείων που αποθηκεύονται σε ένα δοχείο της εικονικής δομής.

Εάν p_i είναι ένας ενδιάμεσος κόμβος του NIPPERS-Grid δικτύου, συσχετίζεται με έναν εσωτερικό κόμβο του ανώτερου επιπέδου της εικονικής δομής δεδομένων. Έτσι περιέχει δύο πίνακες: έναν τοπικό πίνακα REF των δειγμάτων στοιχείων, ένα για κάθε ένα από τα υποδέντρα του, και έναν πίνακα ID. Χρησιμοποιώντας τον πίνακα ID είναι δυνατό να παρεμβληθεί ο πίνακας REF για να καθορίσει το υποδέντρο στο οποίο θα συνεχιστεί η διαδικασία αναζήτησης.

Έστω ότι οι κόμβος-φύλλο₁, κόμβος-φύλλο₂,... κόμβος-φύλλο_n είναι οι κόμβοι-φύλλα που ταξινομούνται σύμφωνα με τον αριθμό τους. Τότε, τα σύνολα κλειδιών S_j διατηρούν μια καθολική σειρά. Αυτό σημαίνει ότι

$$\forall S_j, S_m, 1 \leq j \leq n, 1 \leq m \leq n, j \neq m,$$

$$\text{εάν } \min\{S_j\} < \min\{S_m\} \text{ τότε } \max\{S_j\} < \min\{S_m\}.$$

Κατά συνέπεια, ταξινομούμε τα σύνολα κλειδιών για να παρέχουμε μια φυλλο-προσανατολισμένη δομή δεδομένων (βλ. Σχήμα 17). Τέλος, για κάθε κόμβο διατηρούμε ρητά το γονέα, το παιδί, και τους αδερφούς (sibling) δείκτες. Οι δείκτες στους αδερφούς (sibling) κόμβους θα αναφερθούν εναλλακτικά ως σύνδεσμοι επιπέδου. Οι απαραίτητες πληροφορίες δεικτών μπορούν να ενσωματωθούν εύκολα στην κατασκευή του δέντρου αναζήτησης NIPPERS.

3.5.2 Ανάλυση του αλγορίθμου αναζήτησης

Θεώρημα 3.1: Θεωρούμε το δέντρο αναζήτησης NIPPERS [17] με παραμέτρους $R(s_0) = (s_0)^{1-1/\delta}$, όπου $0 < \delta < 1$, και $s_0 = n$. Τότε, τα ερωτήματα ακριβούς αντιστοίχισης απαιτούν $O(\log \log n)$ αναμενόμενο αριθμό βημάτων (hops) για τα στοιχεία που προέρχονται από μια uniform κατανομή [18].

Απόδειξη: Έστω ότι ένας κόμβος εκτελεί μια αναζήτηση για το κλειδί k . Για να εκτελέσει την αναζήτηση, εγκαθίσταται μια σύνδεση σε ένα κόμβο p στο NIPPERS δίκτυο και εκτελείται η κλήση `NIPPERSgrid_search(p, k)` (βλ. Αλγόριθμος 4): προχωρώντας προς τη ρίζα, έστω ότι φθάσαμε στον κόμβο u . Ελέγχουμε εάν το k είναι απόγονος του u ή του δεξιού γείτονα του u στο ίδιο επίπεδο ψάχνοντας στον πίνακα REF του u ή του δεξιού γείτονα του u αντίστοιχα. Αν όχι, τότε προχωράμε στον πατέρα του u . Διαφορετικά γυρίζουμε πίσω και αναζητάμε το k με το συνηθισμένο τρόπο. Ας υποθέσουμε ότι γυρίζουμε πίσω στον κόμβο w ύψους h . Έστω ότι v είναι εκείνος ο γιος του w που είναι στο μονοπάτι για τον κόμβο p . Τότε όλοι οι απόγονοι του δεξιού γείτονα του v βρίσκονται μεταξύ του κόμβου p και του κλειδιού k . Το υποδέντρο με ρίζα στο δεξιό γείτονα του v είναι ένα δέντρο NIPPERS για $n' \leq n$ στοιχεία, και το ύψος του είναι $h = \Theta(\log \log n)$. Έτσι, προκύπτει ότι η χρονική πολυπλοκότητα είναι αυτή που δηλώνεται στο Θεώρημα 3.1.

Παρατήρηση 3.1: Αξιοποιώντας τη σειρά μεταξύ των `key_sets` στα φύλλα, τα Ερωτήματα Περιοχής (Range Queries) της μορφής $[k_l, k_r]$ απαιτούν

$$O(\log \log n + |A|) \text{ βήματα (hops),}$$

όπου $|A|$ είναι ο αριθμός των κόμβων-φύλλων μεταξύ των κόμβων που είναι υπεύθυνοι για τα k_l, k_r αντίστοιχα.

```
Procedure INSERT_leaf_peer
{
  \\ Έστω T ένα υπάρχον NIPPERS-Grid δέντρο και  $v_0, v_1, \dots, v_k$  το μονοπάτι
  από το νέο κόμβο  $v_0$  στη ρίζα root  $v_k$  \\
  store the overflowed key in  $v_0$ ;
   $C(v_0)=0$ ;
   $isize(v_0)=1$ ;
  \\  $isize(v)$  δηλώνει την αρχική τιμή (βάρος) του υποδέντρου  $T_v$  την
  τελευταία φορά που μηδενίστηκε ο μετρητής  $C(v)$  \\
  for ( $i=1$ ;  $i \leq k$ ;  $++i$ )  $C(v_i)=C(v_i)+1$ ;
  Find the maximum  $i$  such as:  $C(v_i) \leq isize(v_i)/4$ ;
  If (there exists such  $i$ )
    Rebuild( $T_{v_i}$ );
}
```

Αλγόριθμος 5: Ψευδο-κώδικας για τον αλγόριθμο εισαγωγής NIPPERS

```
Procedure DELETE_leaf_peer
{ \\ Έστω ότι T είναι το υπάρχον NIPPERS-Grid δέντρο και  $v_0, v_1, \dots, v_k$ 
  είναι το μονοπάτι από το διαγραμμένο κόμβο  $v_0$  στη ρίζα  $v_k$  \\
  transfer the few remaining keys of  $v_0$  to the left or right neighbors;
  for ( $i=1$ ;  $i \leq k$ ;  $++i$ )  $C(v_i)=C(v_i)+1$ ;
  Find the maximum  $i$  such as:  $C(v_i) \leq isize(v_i)/4$ ;
  If (there exists such  $i$ )
    Rebuild( $T_{v_i}$ );
}
```

Αλγόριθμος 6: Ψευδο-κώδικας για τον αλγόριθμο διαγραφής NIPPERS

```
Procedure Rebuild( $T_v$ )
{
  Build a new NIPPERS-Grid structure for  $T_v$ ;
   $\forall v \in T$  Set  $C(v)=0$ ;
   $\forall v \in T$  define  $isize(v)$ ;
}
```

Αλγόριθμος 7: Ψευδο-κώδικας για τη λειτουργία ανακατασκευής NIPPERS

3.5.3 Εισαγωγή/Διαγραφή των Στοιχείων/Κόμβων στο NIPPERS

Όταν θέλουμε να εισάγουμε/ διαγράψουμε ένα κλειδί (βλ. Αλγόριθμος 5, Αλγόριθμος 6) από τη δομή ψάχνουμε αρχικά για τον κόμβο (`key_host_peer`) που είναι αρμόδιος για αυτό (χρησιμοποιώντας έναν αριθμό $O(\log \log n)$ hops) και έπειτα απλά εισάγουμε/διαγράφουμε τον κατάλληλο κόμβο. Μετά από $O(n)$ εισαγωγές/διαγραφές εκτελούμε όπως αναφέρθηκε ήδη σφαιρική ανακατασκευή. Ωστόσο υπάρχουν μερικές περιπτώσεις που ένας κόμβος-φύλλο υπερχειλίζει ή υποχειλίζει (underflows). Στην πρώτη περίπτωση, πρέπει να εισάγουμε εκεί κοντά

ένα νέο κόμβο-φύλλο. Στη δεύτερη περίπτωση, πρέπει να διαγράψουμε τον κόμβο-φύλλο μετακινώντας πρώτα τα λίγα κλειδιά που απομένουν προς τους αριστερούς ή δεξιούς γείτονες. Επιλέξαμε να εξετάσουμε αυτές τις εισαγωγές/ διαγραφές κόμβων με την υιοθέτηση μιας σειράς τοπικών ανακατασκευών που εφαρμόζονται χρησιμοποιώντας τους κατάλληλους μετρητές $C()$ πάνω (attached) στους κόμβους. Αυτές οι τοπικές ανακατασκευές μπορούν ομοίως να χειριστούν τις εισαγωγές στοιχείων και τις διαγραφές (κατά συνέπεια με το [18]) καθιστώντας έτσι τη σφαιρική ανακατασκευή απαρχαιωμένη.

Με την υιοθέτηση αυτής της σειράς τοπικών ανακατασκευών μπορεί να αποδειχθεί (η απόδειξη παραλείπεται δεδομένου ότι είναι αριστά όμοια με το [18]) ότι οι εισαγωγές και οι διαγραφές απαιτούν $O(\log \log n)$ αναμενόμενο (expected) επιμερισμένο (amortized) αριθμό hops για τα ομαλά κατανεμημένα στοιχεία.

Η αναδημιουργία του υποδέντρου T , απαιτεί τοπικότητα δικτύου. Στην περίπτωση όπου οι κόμβοι οργανώνονται σε σχετικούς επιχειρησιακούς τομείς, τότε επιτυγχάνεται η τοπικότητα δικτύου. Εάν δεν υπάρχει τέτοια τοπικότητα τότε διακυβεύεται ένας σοβαρός παράγοντας κόστους επικοινωνίας. Προκειμένου να υποστηριχθούν όλα τα είδη WS P2P οργάνωσης καταλόγου, προτείνουμε τη χρήση ενός τοπολογικού διαχωρισμού (partition) δικτύων. Κατ' αυτό τον τρόπο ελαχιστοποιούνται επίσης οι κίνδυνοι αστοχίας μονού σημείου.

3.6 Ανεκτικότητα σε σφάλματα του NIPPERS

Γενικά, για να παρέχουμε ανοχή στις αποτυχίες, υιοθετήθηκε ένα πρωτόκολλο αντιγραφής δεδομένων έτσι ώστε σε μια αποτυχία κόμβων το σύστημα να βρίσκει έναν κοντινό κόμβο ο οποίος αποθηκεύει ένα αντίγραφο των επιθυμητών δεδομένων. Στη δομή μας, για κάθε κόμβο, φτιάχνουμε k πλεονάζοντες κόμβους όπου καθένας από αυτούς αποθηκεύει ένα αντίγραφο (replicated) ενός στοιχείου δεδομένων, όπου $k > 1$ είναι μια μικρή θετική σταθερά. Υποθέτουμε ότι το δίκτυο είναι « k -robust», που σημαίνει ότι η ταυτόχρονη αποτυχία όλων αυτών των κόμβων είναι αδύνατη, κατά συνέπεια τουλάχιστον ένας κόμβος είναι ζωντανός στο δίκτυο.

Χρησιμοποιούμε την ιδέα του τοπολογικού χωρίσματος σε ένα δίκτυο γράφων G (Σχήμα 18). Διαιρούμε το γράφο G σε έναν εκθετικό αριθμό (n^α , $0 < \alpha < 1$) υπογράφων G_i , $1 \leq i \leq n^\alpha$. Για λόγους απλότητας, ορίζουμε $\alpha = 1/2$. Κάθε ένας από τους υπογράφους G_i έχει k αντιπροσωπευτικούς κόμβους, όπου $k > 1$ μια μικρή θετική σταθερά. Υποθέτουμε ότι κάθε ένας υπογράφος είναι k -robust, που σημαίνει ότι το ταυτόχρονο κλείσιμο όλων των αντιπροσωπευτικών κόμβων είναι αδύνατο, κατά συνέπεια τουλάχιστον ένας από τους k - κόμβους είναι ενεργός στο δίκτυο.

Σχήμα 18: Διαχωρισμός Δικτυακού Γράφου

Χωρίζουμε κάθε υπογράφο επαναληπτικά εωσότου βρούμε ένα μικρό τοπολογικό υπογράφο (Σχήμα 19) χωρίς περισσότερους από c κόμβους, όπου $c > 1$, μια μικρή θετική σταθερά. Ως αποτέλεσμα, δημιουργείται μια Κατανεμημένη Εκθετική δομή. Κάθε κόμβος αντιπροσωπεύει ένα υπογράφο και διατηρεί τις κομβο-συνδέσεις (host-links) με υπογράφους που έχουν χωριστεί. Ο αριθμός των κομβο-συνδέσεων μειώνεται εκθετικά ενώ προχωρούμε στο δέντρο. Οι κομβο-συνδέσεις δημιουργούνται μεταξύ των αντιπροσωπευτικών κόμβων κάθε υπογράφου.

Το επόμενο βήμα είναι να αποθηκευτούν τα ταξινομημένα σύνολα κλειδιών (key_sets) στα φύλλα της παραπάνω τοπολογικής δομής. Η δομή είναι βασισμένη σε κλειδιά, και έτσι, οι πληροφορίες κλειδιών πρέπει να αποθηκευτούν σε όλους τους κόμβους μέχρι τη ρίζα. Εκτελούμε την αναζήτηση, την ένωση, και τις λειτουργίες φύλλων με τον ίδιο τρόπο.

3.7 Εξασφάλιση Ποιότητας στην Ανακάλυψη Web Service

Όπως σημειώθηκε και στο κεφάλαιο 1, η Ποιότητα Παροχής WS (Quality of WS Provisioning) είναι ένα θέμα που παραμελείται, τουλάχιστον στο μεγαλύτερο μέρος της έρευνας στην περιοχή των WS [22],[12],[13]. Κατά συνέπεια, δεν υπάρχει ούτε ξεκάθαρος ορισμός ούτε συνολικά αποδεκτές αντιλήψεις για την QoWS. Ωστόσο, πρόσφατες εργασίες [21], [23] δίνουν νέο έδαφος σε μία προσπάθεια να καθοριστούν μερικές από τις παραμέτρους QoWS και τις μεθόδους παράδοσης WS. Αντίστοιχα η προσέγγισή μας [16], η οποία αναλύθηκε στο προηγούμενο κεφάλαιο 2, είναι μια τεχνική δυναμικής ανακάλυψης η οποία επιλέγει την αποδοτικότερη WS λαμβάνοντας υπόψη μη λειτουργικές και πραγματικού χρόνου υπολογίσιμες QoWS παραμέτρους.

Η επιχειρησιακή συμπεριφορά έχει άμεση σχέση με την QoWS από την άποψη του κόστους εκτέλεσης και της αξιοπιστίας του φορέα παροχής υπηρεσιών. Τέλος, υπάρχουν περιορισμοί μετα-δεδομένων που πρέπει να ικανοποιηθούν σχετικά με τις

UDDI/ WSDL παραμέτρους όπως η θέση, συγκεκριμένες επιχειρήσεις κ.λπ. Χρησιμοποιώντας το NIPPERS, κάποια ερωτήματα μπορούν να ανακτήσουν μια σειρά από διαφορετικές περιγραφές Μοντέλου. Κατ' αυτό τον τρόπο, έχουμε προτείνει ένα αποδοτικό δικτυοκεντρικό περιβάλλον για τον πρόσθετο έλεγχο των χαρακτηριστικών QoWS όπου μπορεί να εκτελεστεί εκτενής επικύρωση της QoWS προκειμένου να επιλεγεί η τελική αποδοτικότερη επιλογή WS.

Σχήμα 19: Η αρχιτεκτονική NIPPERS ως κατανεμημένο εκθετικό δέντρο αναζήτησης (Distributed Exponential Search Tree)

3.8 Αξιολόγηση και Περιγραφή της Συνεισφοράς του NIPPERS

Στη συνέχεια αξιολογούμε το πρωτόκολλο NIPPERS μέσω προσομοίωσης. Ο προσομοιωτής παράγει αρχικά K κλειδιά τα οποία προέρχονται από μια ομοιόμορφη κατανομή. Μετά από τη διαδικασία αρχικοποίησης ο προσομοιωτής ταξινομεί τα κλειδιά και επιλέγει ως αντιπροσώπους δοχείου (bucket) το 1^ο κλειδί, το $\ln n$ -οστό κλειδί, το $2\ln n$ -οστό κλειδί, κλπ. Προφανώς δημιουργεί N δοχεία ή N Leaf_nodes όπου $N=K/\ln n$. Με τη μοντελοποίηση των εισαγωγών/διαγραφών όπως το συνδυαστικό παιχνίδι δοχείων και σφαιρών (bins and balls) που παρουσιάζεται στο [14], το μέγεθος κάθε δοχείου (host peer) αναμένεται να είναι με μεγάλη πιθανότητα

$$\Theta(\ln n) \text{ (w.h.p.)}.$$

Συγκρίνουμε την απόδοση του NIPPERS [17] με τα αποτελέσματα από το γνωστό CHORD που παρουσιάζεται στο [6] και [30]. Πιο συγκεκριμένα, αξιολογούμε το μήκος του μονοπατιού αναζήτησης αυτών των δύο αρχιτεκτονικών. Προκειμένου να γίνει κατανοητή στην πράξη η απόδοση δρομολόγησης αυτών των δύο πρωτοκόλλων, προσομοιώσαμε ένα δίκτυο με $N=2^k$ κόμβους, που αποθηκεύουν $K=100 \times 2^k$ κλειδιά. Ορίσαμε ότι η παράμετρος k κυμαίνεται από 3 έως 14 και πραγματοποιήσαμε ένα ξεχωριστό πείραμα για κάθε τιμή. Κάθε κόμβος σε ένα πείραμα επέλεξε ένα τυχαίο σύνολο κλειδιών από το σύστημα για να ρωτήσει, και μετρήσαμε το μήκος μονοπατιού που απαιτήθηκε για να επιλύσει κάθε ερώτημα. Για τα πειράματα θεωρήσαμε τεχνητά σύνολα στοιχείων. Η δημιουργία τους βασίστηκε σε διάφορες κατανομές όπως οι Uniform, Regular, Weibull, Beta και Normal. Για καθεμία από αυτές τις κατανομές αξιολογήσαμε το μήκος μονοπατιού για ερωτήματα αναζήτησης. Έπειτα υπολογίσαμε τις μέσες τιμές των παραπάνω διαδικασιών για όλες τις πειραματικές δοκιμές. Τα παρακάτω σχήματα απεικονίζουν το μέσο μήκος μονοπατιών για κατανομή Weibull, Beta και Normal (Σχήμα 20, σχήμα 21, σχήμα 22) αντίστοιχα.

Από τα πειραματικά αποτελέσματα συμπεραίνουμε ότι το NIPPERS είναι περισσότερο αποδοτικό κατά την αναζήτηση. Η αποδοτικότητα κατά την αναζήτηση αποδείχτηκε νωρίτερα και είναι προφανής. Η σχετική ισοδυναμία στην συμπεριφορά ως προς την εξισορρόπηση φορτίου είναι αναμενόμενη αφού στην πράξη ο χώρος του πίνακα δρομολόγησης που χρησιμοποιεί κάθε κόμβος στο Chord διατηρεί $\log^2 N$ εγγραφές και κάθε φύλλο-κόμβος του NIPPERS χρειάζεται $\text{polylog}(n)$.

Σχήμα 20: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Weibull (smooth) κατανομή

Σχήμα 21: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Beta (smooth) κατανομή

Σχήμα 22: Μήκος μονοπατιού για ερωτήματα αναζήτησης όταν κόμβοι εισέρχονται και φεύγουν ακολουθώντας την Normal (smooth) κατανομή

Για λόγους σύγκρισης και απεικόνισης της συνεισφοράς μας με το NIPPERS [17], παρουσιάζεται η αξιολόγηση μιας στοιχειώδους λειτουργίας σε σχέση με το διαδεδομένο και πλέον αποδοτικό CHORD [30] (Πίνακας 7).

Αρχιτεκτονικές Ομότιμων Δικτύων	Μηνύματα Αναζήτησης	Μηνύματα Ενημέρωσης	Πληροφορίες Δρομολόγησης
CHORD [30]	$O(\log n)$	$O(\log^2 n)$ με μεγάλη πιθανότητα	$O(\log n)$ κόμβοι
NIPPERS [17]	$O(\log \log n)$ αναμενόμενο	$O(\log \log n)$ αναμενόμενο amortised $O(n)$ χειρότερη περίπτωση	Μειούμενο Εκθετικά

Πίνακας 7: Σύγκριση απόδοσης NIPPERS

Η αρχιτεκτονική μας παρέχει βελτιωμένο κόστος αναζήτησης για τις ερωτήσεις ακριβούς-αντιστοίχισης, από $O(\log n)$ στο DHTs σε $O(\log \log n)$ στο NIPPERS, καθώς και μια επαρκή και απλή λύση στο πρόβλημα του ερωτήματος περιοχής. Τα ερωτήματα ενημέρωσης όπως τα αιτήματα εγγραφής και διαγραφής της WS δεν εκτελούνται τόσο συχνά όσο μια σύνδεση και αποσύνδεση χρήστη σε ένα χαρακτηριστικό P2P δίκτυο παράδοσης δεδομένων. Οι Web Services είναι λογισμικό που αναπτύσσεται για να υποστηρίξει τις επιχειρησιακές δομές και τις διαδικασίες που αναμένονται να είναι διαθέσιμες στους καταλόγους ανακάλυψης WS περισσότερο από ότι το χρονικό διάστημα μιας συνόδου χρήστη P2P. Το NIPPERS προσαρμόζεται πολύ καλά σε διαφορετικές συνθήκες στην επιμερισμένη (amortized) περίπτωση και είναι καλύτερο από το Chord στις αναμενόμενες επιχειρησιακού προσανατολισμού αδύναμες - αραιές ενημερώσεις. Το NIPPERS δεν προσαρμόζεται καλά στη χειρότερη περίπτωση, η οποία ωστόσο δεν συναντιέται τυπικά στις εφαρμογές WS καταλόγου. Επιπλέον, ένα σχήμα ανεκτικότητας σε σφάλματα (fault tolerance schema) είναι διαθέσιμο στην υποστήριξη με την ακρίβεια μια στοιχειώδους επιχειρησιακής λύση web services.

Συμπερασματικά και προτού συνεχίσουμε στην παρουσίαση του δεύτερου αλγορίθμου, ως το σημείο αυτό παρουσιάσαμε μια δομή ανακάλυψης Web Services πάνω από ένα ομότιμο δίκτυο που ονομάζουμε NIPPERS. Είναι ένα πρωτόκολλο που λύνει με αποκεντριοποιημένο τρόπο το προκλητικό πρόβλημα του καθορισμού κόμβου που αποθηκεύει ένα Web Service ή τους κόμβους που αποθηκεύουν ένα σύνολο στοιχείων WS, οι όποιοι ικανοποιούν ένα κριτήριο περιοχής.

3.9 Παρουσίαση της προσέγγισης του BDT-GRID

Μετά την παρουσίαση του NIPPERS [17] προχωρήσαμε σε μία ακόμη προσέγγιση με σκοπό να αντιμετωπίσουμε με ακόμη μεγαλύτερη επιτυχία τις περιπτώσεις όπου απαιτούνται αναδιοργανώσεις/ ανακατασκευές. Υπενθυμίζουμε ότι στην αποκεντριοποιημένη ανακάλυψη των Web Services οι κόμβοι δικτύων που χρησιμοποιούνται θεωρούνται ως κόμβοι που διαμοιράζονται πληροφορίες και είναι σε θέση να ρωτήσουν άλλους κόμβους.

Γενικά, τα βασισμένα σε DHT συστήματα παρέχουν αποδοτική επεξεργασία των διαδικασιών δρομολόγησης/εντοπισμού που, λαμβάνοντας υπόψη μια ερώτηση για ένα id εγγράφου, εντοπίζουν (δρομολογούν το ερώτημα σε) τον όμοιο κόμβο που αποθηκεύει αυτό το έγγραφο. Κατά συνέπεια, παρέχουν υποστήριξη για τα

ερωτήματα ακριβούς αντιστοίχισης. Τα βασισμένα σε DHT συστήματα αναφέρονται ως δομημένα P2P συστήματα επειδή γενικά στηρίζονται στις αναζητήσεις ενός κατανεμημένου hash πίνακα, ο οποίος δημιουργεί μια δομή στο σύστημα που προκύπτει από τον τρόπο που οι κόμβοι καθορίζουν τους γείτονές τους.

Εκτός από τα DHT δομημένα P2P συστήματα, υπάρχουν και άλλα που χτίζουν κατανεμημένες, επεκτάσιμες δομές δεικτοδότησης για να δρομολογήσουν τέτοια ερωτήματα, όπως το P-Grid. Το P-Grid [1] είναι μια επεκτάσιμη δομή πρόσβασης που βασίζεται σε ένα εικονικό κατανεμημένο δέντρο αναζήτησης. Χρησιμοποιεί πιθανοτικές τεχνικές για να δημιουργήσει και να διατηρήσει τη δομή με σκοπό να παρέχει πλήρη αποκεντριοποίηση.

Εδώ παρουσιάζουμε το BDT-GRID (Balanced Distributed Tree) [10] μια νέα αποδοτική δομή grid για αποκεντριοποιημένη ανακάλυψη Web Service. Το BDT-GRID παρέχει υποστήριξη για την επεξεργασία

(α) Ερωτημάτων Ακριβούς Αντιστοίχισης της μορφής «δεδομένου ενός κλειδιού, αντιστοίχισε το κλειδί σε έναν κόμβο», και

(β) Ερωτήματα Περιοχής της μορφής «αντιστοίχισε τους κόμβους των οποίων τα κλειδιά ανήκουν στη συγκεκριμένη περιοχή».

Το BDT-GRID χρησιμοποιεί ένα εικονικό Εικαστικό Δέντρο Αναζήτησης για να οδηγήσει τις βασισμένες σε κλειδί αναζητήσεις. Ο εντοπισμός δεδομένων μπορεί να εφαρμοστεί εύκολα πάνω από το BDT συσχετίζοντας ένα κλειδί με κάθε στοιχείο δεδομένων, και αποθηκεύοντας το ζευγάρι στοιχείων κλειδί/δεδομένο στον κόμβο στον οποίο αντιστοιχεί το κλειδί. Υποθέτουμε ότι κάθε κόμβος αποθηκεύει ένα ταξινομημένο σύνολο κλειδιών και ο αλγόριθμος αντιστοίχισης τρέχει με τέτοιο τρόπο που τα τοπικά ταξινομημένα key_sets είναι επίσης ανεξάρτητα (disjoint) μεταξύ τους. Το BDT-Grid προσαρμόζει αποτελεσματικά τα ερωτήματα ενημέρωσης καθώς οι κόμβοι συνδέονται και αφήνουν (join and leave) το σύστημα, και μπορεί να απαντήσει στις ερωτήσεις ακόμα κι αν το σύστημα αλλάζει συνεχώς. Τα αποτελέσματα από τη θεωρητική ανάλυση δείχνουν ότι το κόστος επικοινωνίας των διαδικασιών ερώτησης και αναπροσαρμογής συμπεριφέρονται διπλολογαριθμικά με τον αριθμό των BDT-GRID κόμβων. Επιπλέον, το σύστημά μας είναι σταθερό στις περιπτώσεις σφαλμάτων (failure).

3.10 Προαπαιτούμενα: Ιεραρχικοί και βασισμένοι σε δέντρα αλγόριθμοι

Στη συνέχεια παρουσιάζονται εν συντομία οι ιεραρχικοί και βασισμένοι σε δέντρα αλγόριθμοι που είναι χρήσιμοι στο ευρύτερο πλαίσιο των ομότιμων δικτύων.

3.10.1 Ιεραρχικά Πρωτόκολλα

Τα ιεραρχικά πρωτόκολλα δεν είναι νέα, αλλά παρέχουν μια ενδιαφέρουσα προσέγγιση στην ισορροπία μεταξύ της επεκτασιμότητας και της απόδοσης. Η πιο γνωστή υπηρεσία σε χρήση σήμερα που χρησιμοποιεί ένα ιεραρχικό πρωτόκολλο

είναι το DNS (Domain Name Service)[9]. Ο σκοπός του DNS είναι να μεταφράσει ένα φιλικό στον άνθρωπο όνομα περιοχής, όπως το www.ietf.org, στην αντίστοιχη διεύθυνση IP του (σε αυτήν την περίπτωση 4.17.168.6). Η αρχιτεκτονική του DNS αποτελείται από τα εξής:

- Εξυπηρετητές ονομάτων ρίζας (Root name servers)
- Εξυπηρετητές άλλων ονομάτων (Other name servers)
- Πελάτες

Οι εξυπηρετητές άλλων ονομάτων μπορούν επίσης να ταξινομηθούν ως κύριοι (authorative) για ορισμένες περιοχές (όπως για το Πανεπιστήμιο Πατρών (upatras.gr) ο DNS εξυπηρετητής nic.upatras.gr). Το Διαδίκτυο στα αρχικά του στάδια ανάγκασε όλους τους κόμβους (hosts) να διατηρήσουν ένα αντίγραφο ενός αρχείου με όνομα `hosts.txt`, το οποίο περιείχε όλες τις απαραίτητες μεταφράσεις. Καθώς το μέγεθος του δικτύου αυξήθηκε, οι συχνές αλλαγές του αρχείου έγιναν ανέφικτες. Η εισαγωγή του DNS το 1994 επίλυσε αυτό το πρόβλημα και έχει λειτουργήσει επιτυχώς από τότε.

3.10.2 Ένα παράδειγμα μιας DNS αναζήτησης

Έστω ότι ένας host βρίσκεται στην περιοχή sourceforge.net. Το ακόλουθο σενάριο παρουσιάζει πως θα έμοιαζε μια DNS αναζήτηση στην πράξη.

Εάν ένας χρήστης στον προαναφερθέντα host, στην περιοχή sourceforge.net, κατευθύνει τον διαφυλλιστή Ιστού του στη διεύθυνση <http://www.ietf.org>, ο διαφυλλιστής εκδίδει μια DNS αναζήτηση για το όνομα www.ietf.org.

Το αίτημα στέλνεται στον τοπικό εξυπηρετητή (name server) της περιοχής sourceforge.net.

Ο name server στο sourceforge.net δεν έχει τη δυνατότητα να απαντήσει άμεσα στο ερώτημα, αλλά γνωρίζει την διεύθυνση των root name servers και επικοινωνεί με έναν από αυτούς.

Υπάρχουν 13 root name servers (10 στις ΗΠΑ, 1 στο Ηνωμένο Βασίλειο, 1 στη Σουηδία και 1 στην Ιαπωνία) [26]. Ο root name server γνωρίζει τη διεύθυνση ενός name server από την περιοχή `org`. Αυτή η διεύθυνση στέλνει σε απάντηση της ερώτησης του τοπικού name server στο sourceforge.net.

Ο name server στο sourceforge.net ρωτάει τον name server της περιοχής `org`, αλλά ούτε αυτός έχει την απάντηση, όμως ο name server της περιοχής `org` γνωρίζει το όνομα και τη διεύθυνση του authoritative name server της περιοχής ietf.org.

Ο name server στο sourceforge.net επικοινωνεί με τον name server στο ietf.org και ρωτάει ξανά τη διεύθυνση του www.ietf.org. Αυτή τη φορά η απάντηση βρίσκεται και επιστρέφεται η IP διεύθυνση 4.17.168.6.

Ο διαφυλλιστής μπορεί να συνεχίσει τη δουλειά του ανοίγοντας μια σύνδεση στο σωστό host.

Σημειώνουμε εδώ ότι ένα ερώτημα που στέλνεται σε ένα name server μπορεί να είναι είτε αναδρομικό (recursive) είτε επαναληπτικό. Ένα αναδρομικό ερώτημα προκαλεί ο name server να συνεχίσει να ρωτάει άλλους name servers μέχρι να λάβει μια απάντηση, η οποία μπορεί να είναι ότι το όνομα δεν υπάρχει. Ένα επαναληπτικό ερώτημα επιστρέφει άμεσα μια απάντηση στο host που υποβάλλει το ερώτημα. Εάν δε δοθεί μια συγκεκριμένη απάντηση, δίνονται υποδείξεις για το ποιοι servers να ερωτηθούν.

3.10.3 Λανθάνουσα μνήμη στο DNS

Η αποθήκευση στη λανθάνουσα μνήμη παίζει έναν σημαντικό ρόλο στο DNS. Στο παραπάνω παράδειγμα ο τοπικός name server θα αποθηκεύει στη λανθάνουσα μνήμη (caching) τις διευθύνσεις που λαμβάνονται για τον name server της περιοχής org και της περιοχής ietf.org επίσης, καθώς και την τελική απάντηση, τη διεύθυνση του www.ietf.org. Αυτό προκαλεί οι επόμενες μεταφράσεις του www.ietf.org να απαντώνται άμεσα από τον τοπικό name server, και οι μεταφράσεις άλλων hosts στην περιοχή ietf.org μπορεί να παρακάμψουν τον root name server και τον org server. Η μετάφραση μιας διεύθυνσης όπως το www.gnu.org παρακάμπτει τον root name server και ρωτά απευθείας τον name server για την περιοχή org.

3.10.4 Πλεονασμός και ανεκτικότητα σε σφάλματα στο DNS

Για να γίνει το DNS ανεκτικό σε σφάλματα (fault tolerant), οποιοσδήποτε name server μπορεί να κρατήσει ένα σύνολο καταχωρήσεων ως απάντηση σε μια ερώτηση (πλεονασμός – redundancy). Ένας name server μπορεί να απαντήσει σε μια ερώτηση όπως «Ποια είναι η διεύθυνση του www.gnu.org» (Πίνακας 8), ο οποίος παρέχει τα ονόματα των name servers της περιοχής gnu.org. Τα αποτελέσματα προκύπτουν χρησιμοποιώντας την utility dig που είναι διαθέσιμη στα περισσότερα Unix based λειτουργικά συστήματα. Στην πραγματικότητα η απάντηση είναι πολύ πιο σύντομη.

ANSWER SECTION:

gnu.org.	86385	IN	NS	nic.cent.net.
gnu.org.	86385	IN	NS	ns1.gnu.org.
gnu.org.	86385	IN	NS	ns2.gnu.org.
gnu.org.	86385	IN	NS	ns2.cent.net.
gnu.org.	86385	IN	NS	ns3.gnu.org.

ADDITIONAL SECTION:

nic.cent.net.	79574	IN	A	140.186.1.4
ns1.gnu.org.	86373	IN	A	199.232.76.162
ns2.gnu.org.	86385	IN	A	195.68.21.199
ns2.cent.net.	79574	IN	A	140.186.1.14

Πίνακας 8: Παράδειγμα απάντησης σε ένα ερώτημα DNS

Το παράδειγμα δείχνει ότι η περιοχή `gnu.org` εμφανίζεται να έχει πέντε name servers (NS), οι διευθύνσεις των τεσσάρων από τους οποίους είναι γνωστές σε εμάς. Το ερώτημα της διεύθυνσης του `www.gnu.org`, μπορεί να σταλεί σε οποιονδήποτε από τους τέσσερις servers. Αυτό σημαίνει ότι μπορούμε να λάβουμε μια απάντηση στο ερώτημά μας εφόσον τουλάχιστον ένας από τους name servers είναι διαθέσιμος.

3.11 Η αρχιτεκτονική BDT-GRID

Το BDT-GRID [10] που προτείνουμε παρέχει μια δομή τύπου Balanced Distributed Tree όπου μπορεί να εκτελεστεί αναζήτηση βασισμένη σε κλειδιά με σκοπό να ανακαλύψει και να επιλέξει ένα ζητούμενο Web Service κλειδί ή URI. Από την πλευρά της χρησιμοποίησης του εύρους ζώνης η αναζήτηση αποδίδει (scales) πολύ καλά καθώς δε συμβαίνουν broadcasting ερωτήματα ή άλλες δραστηριότητες που να καταναλώνουν εύρος ζώνης κατά τη διάρκεια των αναζητήσεων. Καθώς όλες οι αναζητήσεις βασίζονται σε κλειδιά υπάρχουν δυο πιθανότητες:

- Έστω ότι κάθε host υλοποιεί τον ίδιο αλγόριθμο μετάφρασης, ο οποίος μεταφράζει μια ακολουθία λέξεων-κλειδιών σε ένα δυαδικό κλειδί.
- Έστω ότι μια άλλη υπηρεσία παρέχει το δυαδικό κλειδί. Αυτή η υπηρεσία δέχεται ερωτήματα με βάση λέξεις –κλειδιά και μπορεί να απαντήσει με το αντίστοιχο κλειδί.

Η δεύτερη προσέγγιση είναι πιο ακριβής. Είναι επίσης πιθανό να χρησιμοποιηθεί μια πιο κεντριοποιημένη υλοποίηση μιας τέτοιας υπηρεσίας. Από εδώ και στο εξής υποθέτουμε ότι το κλειδί είναι διαθέσιμο. Η εργασία περιγράφει έναν αλγόριθμο για την πρώτη περίπτωση. Επίσης υποθέτουμε ότι ένα σύνολο κλειδιών σε κάθε host διατηρεί μια καθολική διάταξη (global order).

3.11.1 Το BDT-Grid δίκτυο

Το BDT-Grid σύμφωνα με την εργασία [10] είναι ένα ζυγισμένο δέντρο κατανομής T όπου ο βαθμός των κόμβων στο επίπεδο I ορίζεται να είναι

$$d(i) = t(i),$$

όπου το $t(i)$ δείχνει τον αριθμό των κόμβων που υπάρχουν στο επίπεδο i .

Αυτό πρέπει να ισχύει για $i \geq 1$, ενώ $d(0) = 2$ και $t(0) = 1$. Είναι εύκολο να δούμε ότι έχουμε επίσης

$$t(i) = t(i-1)d(i-1),$$

οπότε βάζοντας μαζί τα διάφορα λειτουργικά υποσυστήματα, μπορούμε να σταματήσουμε την επανάληψη και να επιτύχουμε

$$i \geq 1: d(i) = 2^{2^{i-1}}, t(i) = 2^{2^i}.$$

Ένα από τα πλεονεκτήματα αυτού του δέντρου είναι ότι το ύψος του είναι

$$O(\log \log n),$$

όπου n είναι ο αριθμός των στοιχείων που αποθηκεύονται σε αυτό.

3.11.2 Κόμβοι στο BDT-Grid

Διαχωρίζουμε τις περιπτώσεις των κόμβων-φύλλων και των ενδιάμεσων κόμβων: Εάν ένας κόμβος i , ο οποίος εφεξής δηλώνεται ως p_i , είναι ένας `key_host_peer` (φύλλο) του δικτύου BDT-Grid [10] ισχύουν τα εξής:

- Ένας αριθμός ταξινομημένων δυαδικών κλειδιών k -bit $k_i = b_1 \dots b_k$, όπου το k είναι μικρότερο ή ίσο με n_i , για κάποια δεσμευμένη σταθερά n_i που είναι ίδια για όλα τα p_i . Αυτό το ταξινομημένο σύνολο κλειδιών δείχνει το διάστημα κλειδιών για τα οποία ο κόμβος είναι αρμόδιος. Έστω K ο αριθμός των k -bit δυαδικών κλειδιών και n ο αριθμός των `key_host_peers`. Ενώ μπορούμε αρχικά να καταναλώνουμε τα κλειδιά με τέτοιο τρόπο έτσι ώστε κάθε `host peer` (φύλλο) να αποθηκεύει ένα φορτίο $\Theta(K/n)$ κλειδιών, δεν είναι καθόλου προφανές πώς να περιορίσουμε το φορτίο των `host peers`, κατά τη διάρκεια των διαδικασιών ενημέρωσης. Στο [14] παρουσιάζεται μια ιδέα γενικού επιστημονικού ενδιαφέροντος: μοντελοποιώντας τις εισαγωγές/διαγραφές ως συνδυαστικό παιχνίδι δοχείων και σφαιρών (bins and balls), το μέγεθος κάθε `host κόμβου` αναμένεται να είναι με μεγάλη πιθανότητα (w.h.p.) $\Theta(\ln n)$, για τα κλειδιά που προέρχονται από μια άγνωστη κατανομή.
- Τα `key_sets` $S_j = \{k_i \mid 1 \leq i \leq \Theta(K/n)\}, 1 \leq j \leq n$ διατηρούν μια global σειρά. Αυτό σημαίνει ότι, $\forall S_j, S_q, 1 \leq j \leq n, 1 \leq q \leq n, j \neq q$, εάν $\min\{S_j\} < \min\{S_q\}$ τότε $\max\{S_j\} < \min\{S_q\}$. Επομένως, ταξινομούμε τα παραπάνω `key_sets` παρέχοντας μια φύλλο-προσανατολισμένη δομή δεδομένων όπως φαίνεται στο σχήμα 23.

Εάν p_i είναι ένας ενδιάμεσος κόμβος (ρίζα ή εσωτερικός κόμβος) του δικτύου BDT-Grid σχετίζεται με τα ακόλουθα:

- Ένας τοπικός πίνακας στοιχείων δειγμάτων REF_{p_i} , ένα για κάθε υποδέντρο του. Ο πίνακας REF ονομάζεται πίνακας αναφοράς (reference table) του κόμβου και η έκφραση $REF_{p_i}[r]$ δηλώνει το σύνολο των διευθύνσεων στον δείκτη r στον πίνακα. Κάθε πίνακας REF οργανώνεται βάσει Anderson και Thorup [2] σε γραμμικό χώρο, λύση που επιτυγχάνει ένα βέλτιστο χρόνο χειρότερης περίπτωσης $O(\sqrt{\log n / \log \log n})$ για τις διαδικασίες δυναμικής ενημέρωσης και αναζήτησης, όπου n ο αριθμός των αποθηκευμένων στοιχείων. Θα χρησιμοποιήσουμε αυτήν την λύση ως βάση στη ρουτίνα αναζήτησης στον τοπικό πίνακα κάθε κόμβου δικτύου.

Για κάθε κόμβο p_i διατηρούμε ρητά τους δείκτες γονέα, παιδιού, και αδελφών. Οι δείκτες στους αδελφούς κόμβους (sibling nodes) θα αναφέρονται εναλλακτικά ως

συνδέσεις επιπέδων. Οι απαραίτητες πληροφορίες δεικτών μπορούν εύκολα να ενσωματωθούν στην κατασκευή του δέντρου αναζήτησης BDT-Grid.

Σχήμα 23: Το σύστημα BDT-Grid από την εργασία μας [10]

3.11.3 Πολυπλοκότητα Αναζήτησης BDT-GRID

Θεώρημα 3.2: Έστω ένα BDT-GRID δίκτυο σύμφωνα με την εργασία μας [10]. Τότε, οι λειτουργίες Ακριβούς αντιστοίχισης (Exact Match) απαιτούν $O(\sqrt{\log n / \log \log n})$ βήματα (hops) όπου το n δηλώνει τον τρέχον αριθμό κόμβων.

Απόδειξη: Υποθέτουμε ότι ένας **key_host_peer** p εκτελεί μια αναζήτηση για το κλειδί k. Αρχικά ελέγχουμε εάν το k είναι στα αριστερά ή τα δεξιά του p, έστω ότι το k είναι στα δεξιά του p. Έπειτα προχωρώντας προς τη ρίζα έστω ότι φθάνουμε στον κόμβο u. Ελέγχουμε εάν το k είναι απόγονος του u ή του δεξιού γείτονα του u στο ίδιο επίπεδο, αναζητώντας στον πίνακα REF του u ή του δεξιού γείτονα του u αντίστοιχα. Αν όχι, τότε προχωράμε στον πατέρα του u. Διαφορετικά γυρίζουμε πίσω και ψάχνουμε για το k με τον συνηθισμένο τρόπο. Έστω ότι γυρίζουμε πίσω στον κόμβο w ύψους h και v είναι ο γιος του w που είναι στο μονοπάτι προς τον κόμβο p. Τότε όλοι οι απόγονοι του δεξιού γείτονα του v βρίσκονται μεταξύ του

κόμβου p και του κλειδιού k . Το υποδέντρο T_w είναι ένα BDT δέντρο για $n' \leq n$ στοιχεία, και το ύψος του είναι $h = \Theta(\log \log n)$.

Οπότε, πρέπει να επισκεφθούμε το κατάλληλο μονοπάτι αναζήτησης $w, w_1, w_2, \dots, w_r$ από τον εσωτερικό κόμβο w στον κόμβο-φύλλο w_r . Σε κάθε κόμβο αυτού του μονοπατιού πρέπει να ψάξουμε για το κλειδί k που χρησιμοποιεί τους δείκτες REF_{w_i} , $1 \leq i \leq r$ και $r = O(\log \log n)$, καταναλώνοντας χρόνο χειρότερης περίπτωσης $O(\sqrt{\log d(w_i) / \log \log d(w_i)})$, όπου $d(w_i)$ είναι ο βαθμός του κόμβου w_i . Αυτό μπορεί να εκφραστεί με το παρακάτω άθροισμα:

$$\sum_{i=1}^{r=O(\log \log d)} \sqrt{\frac{\log d(w_i)}{\log \log d(w_i)}}$$

Έστω L_1, L_r τα επίπεδα των W_1 και W_r αντίστοιχα. Τότε,

$$d(w_1) = 2^{2^{L_1}} \text{ και } d(w_r) = 2^{2^{L_r}}$$

Αλλά,

$$L_r = O(\log \log n).$$

Το προηγούμενο άθροισμα μπορεί άρα να εκφραστεί ως εξής:

$$\sqrt{\frac{2^{L_1}}{L_1}} + \sqrt{\frac{2^{L_1+1}}{L_1+1}} + \dots + \sqrt{\frac{\log n}{\log \log n}} = O\left(\sqrt{\frac{\log n}{\log \log n}}\right)$$

Παρατήρηση 3.2.: Αξιοποιώντας τη σειρά μεταξύ των key_sets στα φύλλα είναι προφανές ότι τα Ερωτήματα Περιοχής της μορφής $[k_\ell, k_r]$ απαιτούν $O(\sqrt{\log n / \log \log n} + |A|)$ hops, όπου A είναι το σύνολο απαντήσεων. Σε ένα τέτοιο ερώτημα, οι hosts των οποίων τα κλειδιά ανήκουν στην περιοχή $[k_\ell, k_r]$ μπορούν να βρεθούν ψάχνοντας πρώτα στη BDT-Grid δομή για το k_ℓ και να εκτελώντας έπειτα μια διαπέραση in-order στο δέντρο από το k_ℓ στο k_r .

Για να εκτελέσει την αναζήτηση καθιερώνεται μια σύνδεση σε έναν κόμβο p στο BDT-GRID και εκτελείται η κλήση `bdtgrid_search(p, k)`. Η συνάρτηση `bdtgrid_search` παρουσιάζεται στον Αλγόριθμος 8.

3.11.4 Ανεκτικότητα σε σφάλματα του BDT-GRID

Τι θα συμβεί όταν κάποιοι εσωτερικοί κόμβοι έχουν κλείσει; Στις εφαρμογές αποθήκευσης δεδομένων στη λαθάνουσα μνήμη (data caching), είναι χρήσιμο να λυθεί το γενικότερο πρόβλημα εύρεσης ενός κοντινού κόμβου που έχει πραγματικά ένα αντίγραφο των επιθυμητών δεδομένων. Στη δομή μας, κατά συνέπεια, εξοπλίζουμε κάθε κόμβο με k πλεονάζοντες κόμβους, ο καθένας από τους οποίους αποθηκεύει αντίγραφα (replicated) ενός στοιχείου δεδομένων, όπου $k > 1$ μια μικρή

θετική σταθερά. Επίσης υποθέτουμε ότι κάθε κόμβος είναι k -robust, που σημαίνει ότι το ταυτόχρονο κλείσιμο όλων αυτών των κόμβων είναι αδύνατο, κατά συνέπεια τουλάχιστον ένας είναι ενεργός στο δίκτυο.

```

Node p *bdtgrid_search (p, k)
{
  int j;
  bool move_right=false;

  if (p=host_key && p is responsible for this k)
    return p;
  if (someone else is responsible)
  {
    Check whether k is to the left or right of p;
    \\ έστω ότι το k είναι στα δεξιά του p\\

    p_next=father(p)

    while (k > REFpnext[right_most] && move_right = false)
    {
      p'=right_sibling of p_next;

      if (k <= REFp[right _ most ])
      {
        p_next= p';
        move_right=true;
      }
      else
        p_next=father (p_next);

      host = send_search(p_next, k);
    }
    while (p_next is not a key_host)
    {
      j=search (k, p_next);
      \\ Όπου η search (key, node) δηλώνει τη διαδικασία [2]
      \\που επιστρέφει μια ακέραια θέση j που δείχνει
      \\τον κατάλληλο απόγονο στον οποίο πρέπει
      \\να συνεχίσουμε την αναζήτηση

      p_next =&REFpnext [j];
      host = send_search(p_next, k);
    }
    p=p_next;
    return p;
  }
}

```

Αλγόριθμος 8: Ψευδο-κώδικας για BDT-Grid αναζητήσεις

3.11.5 Συμμετοχή και Αποχώρηση στο BDT-GRID

Στην περίπτωση της υπερχείλισης του `key_host_peer` πρέπει να εισάγουμε εκεί κοντά ένα νέο `host_peer`. Στη δεύτερη περίπτωση της υποχείλισης (underflow) πρέπει να σημειώσουμε ως διαγραφμένο το `key_host_peer` μετακινώντας αρχικά τα λίγα εναπομείναντα κλειδιά στους αριστερούς και δεξιούς γείτονες. Προφανώς μετά

από ένα σημαντικό αριθμό λειτουργιών join/leave απαιτείται μια καθολική ανακατασκευή (global rebuilding) διαδικασία για τον καθαρισμό των πλεοναζόντων κόμβων και την επαναζύγιση της δομής BDT.

```

Procedure INSERT_host_peer (p)
{
  Insert a new leaf node p;
  counter=counter+1;
  p_next=father(p);

  While (p_next !=root)
  {
    update REFp_next ;
    //πρόσθεσε ένα ακόμα σύνδεσμο σύμφωνα με τον αλγόριθμο που
    παρουσιάζεται στο [2] //

    p_next=father(p_next);
  }
  if counter =  $\Theta(n)$  then Rebuild(T);
}

```

Αλγόριθμος 9: Ψευδο- κώδικας για εισαγωγή στο BDT-Grid

```

Procedure DELETE_host_peer p
{
  search for p;
  // σύμφωνα με τη ρουτίνα bdtgrid_search //
  mark p ;
  counter=counter+1;
  if counter =  $\Theta(n)$  then Rebuild(T);
}

```

Αλγόριθμος 10: Ψευδο- κώδικας για διαγραφή στο BDT-Grid

```

Procedure Rebuild(T)
{
  Build a new BDT_Grid structure;
  Counter=0;
}

```

Αλγόριθμος 11: Ψευδο-κώδικας για τη λειτουργία ανακατασκευής BDT-Grid

Θεώρημα 3.3: Θεωρούμε ένα BDT-GRID δίκτυο όπως στην εργασία μας [10]. Τότε οι λειτουργίες join και leave απαιτούν $O(\sqrt{\log n / \log \log n})$ επιμερισμένο αριθμό hops όπου το n δηλώνει τον τρέχον αριθμό κόμβων.

Απόδειξη: Μια λειτουργία join (εισαγωγής) επηρεάζει το μονοπάτι από το νέο κόμβο φύλλο στη ρίζα του BDT-GRID. Σε κάθε μονοπάτι-κόμβο w_i ($1 \leq i \leq c \log \log n$ όπου c είναι μια σταθερά) πρέπει να ενημερώσουμε τον δείκτη REF_{w_i} . Αυτή η διαδικασία απαιτεί χρόνο

$O(\sqrt{\log d(w_i) / \log \log d(w_i)})$, όπου $d(w_i)$ είναι ο βαθμός του κόμβου w_i .

Αυτό μπορεί να εκφραστεί με το ακόλουθο άθροισμα:

$$\sum_{i=1}^{r=O(\log \log d)} \sqrt{\frac{\log d(w_i)}{\log \log d(w_i)}} = \sqrt{\frac{\log n}{\log \log n}}.$$

Η λειτουργία leave (διαγραφής) απαιτεί

$O(\sqrt{\log n / \log \log n})$ hops για την ανίχνευση του κόμβου και

$O(1)$ χρόνο για να σημειώσει αυτόν τον κόμβο ως διαγραμμένο.

Μετά από $\Theta(n)$ λειτουργίες ενημέρωσης πρέπει να ξαναχτίσουμε την κύρια ισόρροπα κατανομημένη δομή (Balanced Distributed backbone). Κατανέμοντας τις $\Theta(n)$ ανακατασκευές στις επόμενες $\Theta(n)$ ενημερώσεις, προκύπτει το επιμερισμένο όριο του θεωρήματος.

3.12 Αξιολόγηση και Περιγραφή της Συνεισφοράς του BDT-GRID

Για λόγους σύγκρισης, ο πίνακας 9 παρουσιάζει την αξιολόγηση/πολυπλοκότητα μιας βασικής λειτουργίας συγκριτικά με την πλέον διαδεδομένη αποδοτική αρχιτεκτονική Chord.

Αρχιτεκτονικές Ομότιμων Δικτύων	Μηνύματα Αναζήτησης	Μηνύματα Ενημέρωσης	Πληροφορίες Δρομολόγησης
CHORD [30]	$O(\log n)$	$O(\log^2 n)$ με μεγάλη πιθανότητα	$O(\log n)$ κόμβους
BDT-GRID [10]	$O(\sqrt{\log n / \log \log n})$	$O(\sqrt{\log n / \log \log n})$ επιμερισμένη $\Theta(n)$ χειρότερη περίπτωση	Εκθετικά αυξανόμενη

Πίνακας 9: Σύγκριση BDT-GRID με την πλέον διαδεδομένη αποδοτική αρχιτεκτονική

Η συμβολή μας προβλέπει βελτιωμένο κόστος αναζήτησης για τα ερωτήματα αριθμούς-αντιστοίχησης, από $O(\log n)$ στο DHT σε $O(\sqrt{\log n / \log \log n})$ στο BDT-GRID, καθώς και μια επαρκή και απλή λύση στο πρόβλημα του ερωτήματος περιοχής. Update Queries όπως τα αιτήματα εγγραφής και διαγραφής WS δεν εκτελούνται τόσο συχνά όσο μια σύνδεση και αποσύνδεση χρήστη σε ένα τυπικό P2P δίκτυο παράδοσης δεδομένων. Οι Web Services είναι λογισμικό που αναπτύσσεται για να υποστηρίξει τις επιχειρησιακές δομές και τις διαδικασίες οι οποίες αναμένεται να είναι διαθέσιμες στους καταλόγους ανακάλυψης WS περισσότερο από το χρονικό διάστημα μιας τυπικής συνόδου ενός χρήστη ομότιμου δικτύου μεταφοράς δεδομένων.

Το BDT-GRID συμπεριφέρεται πολύ καλά στην amortized περίπτωση και είναι καλύτερο από το Chord στις αναμενόμενες επιχειρησιακές σπάνιες ανανεώσεις των

Web Services. Το BDT-GRID δε συμπεριφέρεται καλά στη χειρότερη περίπτωση λόγω μιας πιθανότητας ανακατασκευής που λειτουργεί ως πλεονάζων φόρτος (overhead), το οποίο ωστόσο δε συναντάται συχνά στις περιπτώσεις υλοποιήσεων καταλόγων WS. Επιπλέον, ένα σχήμα ανεκτικότητας σε σφάλματα είναι διαθέσιμο για να υποστηρίξει με πιστότητα μια στοιχειώδη επιχειρησιακή λύση web services.

Σχήμα 24: Σύγκριση απόδοσης αναζήτησης με BDT-Chord

Στη συνέχεια κατ' αναλογία με τη λύση NIPPERS που εισαγάγαμε στα [17] και παρουσιάστηκε νωρίτερα στο παρόν κεφάλαιο, αξιολογούμε το πρωτόκολλο BDT από την εργασία μας [10] μέσω προσομοίωσης και ακολουθούμε την ίδια διαδικασία που περιγράφεται στην ενότητα 3.8. Ο προσομοιωτής χρησιμοποιεί τον κώδικα που περιγράφει ο Αλγόριθμος 8. Συγκρίνουμε την απόδοση του προσομοιωτή BDT με τα αποτελέσματα από το δημοσιευμένο πιο γνωστό προσομοιωτή CHORD που παρουσιάζεται στο [6].

Πιο συγκεκριμένα, αξιολογούμε το μήκος του μονοπατιού αναζήτησης αυτών των δύο αρχιτεκτονικών. Προκειμένου να γίνει κατανοητή στην πράξη η απόδοση δρομολόγησης αυτών των δύο πρωτοκόλλων, προσομοιώσαμε ένα δίκτυο με $N=2^k$ κόμβους, που αποθηκεύουν $K=100 \times 2^k$ κλειδιά. Ορίσαμε ότι η παράμετρος k κυμαίνεται από 3 έως 14 και πραγματοποιήσαμε ένα ξεχωριστό πείραμα για κάθε τιμή. Κάθε κόμβος σε ένα πείραμα επέλεξε ένα τυχαίο σύνολο κλειδιών από το σύστημα για να ρωτήσει, και μετρήσαμε το μήκος μονοπατιού που απαιτήθηκε για να επιλύσει κάθε ερώτημα. Για τα πειράματα θεωρήσαμε τεχνητά σύνολα στοιχείων. Η δημιουργία τους βασίστηκε σε διάφορες κατανομές όπως οι Uniform, Regular, Weibull, Beta και Normal. Για καθεμία από αυτές τις κατανομές αξιολογήσαμε το μήκος μονοπατιού για ερωτήματα αναζήτησης και το μέγιστο φορτίο κάθε κόμβου-φύλλου αντίστοιχα. Έπειτα υπολογίσαμε τις μέσες τιμές των παραπάνω διαδικασιών για όλες τις πειραματικές δοκιμές. Το σχήμα 24 απεικονίζει το μέσο μήκος μονοπατιών αντίστοιχα.

3.13 Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις

Ο συνδυασμός των αρχιτεκτονικών ομοτίμων δικτύων και της τεχνολογίας των Web Services έχει ένα αποτέλεσμα προστιθέμενης αξίας: τα προβλήματα λειτουργικής ενοποίησης υπερνικούνται λόγω της αρχιτεκτονικής παγκόσμιας πληροφορίας (universal information) που παρέχεται από τις WS, και επιπλέον, η άμεση και ανεκτική σε σφάλματα πρόσβαση στους υπολογιστικούς πόρους επιτυγχάνεται χάρη στην αρχιτεκτονική υποδομής δικτύων που παρέχει το P2P. Η αποκεντριοποίηση της ανακάλυψης WS, η όποια αναμένεται να είναι μια από τις ευνοϊότερες εκβάσεις αυτού του συνδυασμού, θα αυξήσει την ανοχή σε σφάλματα (fault tolerance) και την συνολική αποδοτικότητα αναζήτησης.

Στην περίπτωση μιας δομής ανακάλυψης Web Services πάνω από ένα P2P δίκτυο, ένα στοιχείο web service (ή ένα σύνολο WS στοιχείων) που ικανοποιούν το κριτήριο περιοχής αποθηκεύονται σε έναν κόμβο (ή κόμβους αντίστοιχα). Προκειμένου να ανακαλυφθεί η WS που ικανοποιεί τα κριτήρια αναζήτησης, πρέπει να καθοριστούν αυτοί οι κόμβοι. Σε αυτήν το κεφάλαιο εισαγάγαμε και αναλύσαμε δύο προσεγγίσεις λύσεις αυτού του είδους: τον αποδοτικό αλγόριθμο NIPPERS σύμφωνα με την εργασία μας [17] και τον BDT-GRID από την εργασία [10], που αντιμετωπίζουν αυτό το προκλητικό πρόβλημα με αποκεντριοποιημένο τρόπο.

Δικτυοκεντρικές Λύσεις	Αναζήτηση	Ενημέρωση	Πληροφορία Δρομολόγησης
CHORD [30]	$O(\log n)$	$O(\log^2 n)$ w.h.p.	$O(\log n)$ κόμβοι
F-Chord (α) [11]	$O(\log n / \log \log n)$	$O(\log n)$	
LPRS-Chord [33]	Ελαφρώς καλύτερη από $O(\log n)$		
NIPPERS [17]	$O(\log \log n)$ αναμενόμενη	$O(\log \log n)$ επιμερισμένη αναμενόμενη $O(n)$ χειρότερη περίπτωση	Εκθετικά μειούμενη
BDT-GRID [10]	$O(\sqrt{\log n / \log \log n})$ χειρότερη περίπτωση	$O(\sqrt{\log n / \log \log n})$ επιμερισμένη $\Theta(n)$ χειρότερη περίπτωση	Εκθετικά αυξανόμενη

Πίνακας 10: Συνολικός συγκριτικός πίνακας συνεισφοράς δικτυοκεντρικών λύσεων NIPPERS και BDT-grid με γνωστές τεχνικές.

Οι προσεγγίσεις του NIPPERS και του BDT-GRID έχουν κοινό αλγοριθμικό σχεδιασμό. Διαφέρουν ωστόσο σε ό,τι αφορά την πολυπλοκότητά τους. Ο πίνακας 10 παρουσιάζει συνολικά τις παρόμοιες διαθέσιμες λύσεις της βιβλιογραφίας και τους προτεινόμενους αλγορίθμους. Το NIPPERS έχει καλύτερους μεν χρόνους πολυπλοκότητας αλλά είναι αναμενόμενοι (expected), υπονοώντας ότι το σύστημα εξαρτάται από την τάξη (class) της κατανομής. Συγκεκριμένα η πολυπλοκότητα αυτή ανήκει στην οικογένεια των ομοιόμορφων κατανομών. Αυτό ήταν κύριο μειονέκτημα, καθώς στην πραγματικότητα WS εφαρμογές πραγματοποιούν διαδικασίες

ενημέρωσης (update queries) όπως εγγραφή WS και διαγραφή σπάνια μεν αλλά σε συνάρτηση με αυθαίρετα τυχαίες τιμές. Για το λόγο αυτό αναπτύχθηκε η λύση του BDT-GRID, η οποία επιτυγχάνει πολυπλοκότητα χειρότερης περίπτωσης ανεξάρτητα από την τάξη της κατανομής. Επιπλέον σημειώνουμε ότι η διαφορά μεταξύ της πολυπλοκότητας $O(\sqrt{\log n / \log \log n})$ και $O(\log \log n)$ είναι αμελητέα.

Τα μελλοντικά σχέδια επέκτασης περιλαμβάνουν έρευνα για τις σημασιολογικά βασισμένες P2P λύσεις ([20],[19]) προκειμένου να υποστηρίξουμε τις σημασιολογικά εμπλουτισμένες Web Services και τις αντίστοιχες οντολογίες τους ([7], [32]). Ήδη έχουμε εξετάσει τις δυνατότητες της χρήσης οντολογιών για αναζήτηση WS παροχής εκπαιδευτικού περιεχομένου στην εργασία μας [28] και εξετάζουμε την υλοποίηση μιας παρόμοιας προσέγγισης που να χρησιμοποιεί αποκεντρωμένη πλατφόρμα αναζήτησης όπως αυτές που περιγράψαμε στο παρόν κεφάλαιο.

Επιπλέον οι μελλοντικές κατευθύνσεις μπορούν να περιλαμβάνουν την εφαρμογή και την πειραματική αξιολόγηση του NIPPERS και του BDT σε άλλες περιοχές όπως

- ο διαμοιρασμός αρχείων,
- τα time-shared διαθέσιμα συστήματα αποθήκευσης,
- οι κατανεμημένοι δείκτες για έγγραφα και
- μεγάλης κλίμακας κατανεμημένες υπολογιστικές πλατφόρμες επιστημονικών υπολογισμών (υπολογισμοί με GRID για μετεωρολογία, αστρονομία).

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ 3

1. Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Ponceva, M., Schmidt, R.: P-Grid: a self-organizing structured P2P system. SIGMOD Record 32(3): 29-33 (2003)
2. Anderson, A., Thorup, M.: Tight(er) Worst – Case Bounds on Dynamic Searching and Priority Queues, ACM STOC 2000.
3. Andersson, A., Mattson, C., “Dynamic Interpolation Search in $o(\log \log n)$ time”, in Proc of ICALP’93, 1993.
4. Bakker, A., Amade, E., Ballintijn, G., Kuz, I., Verkaik, P., van der Wijk, I., et al, “The Globe Distribution Network”, in Proc. of the USENIX Annual Conf., 2000, pp. 141 – 152.
5. Borenstein, J., Fox, J., “Semantic Discovery of Web services: A step towards fulfillment of the vision”, Web Services Journal, URL <http://www.sys-con.com/webservices>, 2004.
6. Dabek, F., Brunskill, E., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I., Balakrishnan, H., “Building P2P systems with Chord, a distributed location service”, in Proc. of the 8th IEEE Workshop on Hot Topics in Operating Systems, Elmau/Oberbayern, Germany, May 2001, pp. 71-76.
7. DAML Services (DAML-S / OWL-S). Semantic Web Services Architecture (SWSA) Committee. URL <http://www.daml.org/services>
8. Deshpande Y., Murugesan, A., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., White, B., “Web Engineering”, Journal of Web Engineering, Vol. 1, No. 1, 2002, pp. 3-17.
9. DNS, 2005. Domain Name System URL <http://www.dns.net>
10. Drosos, L., Sioutas, S., Sakkopoulos, S., Sirmakessis, S.. «BDT-Grid: An Efficient Scalable Peer-to-Peer Lookup System for Web Service Discovery», International Workshop on Peer to Peer and Service Oriented Hypermedia: Techniques and Systems in conjunction with ACM HyperText 2005, to appear in the ACM HT’05 Proceedings, Salzburg, Austria.
11. G. Cordasco, L. Gargano, M. Hammar, A. Negro, V. Scarano, “Non-uniform deterministic routing on F-Chord(α)”, in Proc. of First International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P’04) Volendam (The Netherlands) October 08, 2004, pp. 16-21
12. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., "Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?". International Workshop on Web Engineering, "Hypermedia Development & Web Engineering Principles and Techniques: Put them in use.", in conjunction with ACM Hypertext 2004, Santa Cruz, August 10, 2004.
13. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., “Contemporary Web Service Discovery Mechanisms”, Journal of Web Engineering special issue for revised and extended papers from the Web Engineering workshop at ACM HyperText 2004, in press.
14. Kaporis, Makris, Ch., Sioutas, S., Tsakalidis, A., Tsichlas, K., Zaroliagis, Ch., “Improved Bounds for Finger Search on a RAM”, LNCS 2832, pp 325-336, 11th Annual European Symposium on Algorithms (ESA 2003) – Budapest, 15-20 September, 2003.
15. Li, Y., Zou, F., Wu, Z., Ma, F. , “PWSD: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network”, in Proc. APWeb’04, LNCS 3007, 2004, pp. 291-300.
16. Makris, C., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., “Efficient and adaptive discovery techniques of web services handling large data sets”, in J. Systems and Software, Elsevier Science, corrected proof, in press.
17. Makris, Ch., Sakkopoulos, E., Sioutas, S., Triantafillou, P., Tsakalidis, A., Vassiliadis, B., "NIPPERS: Network of InterPolated PeERS for Web Service Discovery", in the proceedings of the 2005 IEEE International Conference on Information Technology: Coding & Computing (IEEE ITCC 2005), Track Next Generation Web and Grid Systems, 2005 in Las Vegas, USA, Vol. 2, pp. 193-198.
18. Mehlhorn, K., Tsakalidis, A., “Dynamic Interpolation Search”, Journal of the ACM 40(3), 621-634. 1993.

19. Nejdl, W., Siberski, W. & Sintek, M. (2003) Design Issues and Challenges for RDF- and Schema-Based Peer-to-Peer Systems, SIGMOD Record.
20. Nejdl, W., Wolpers, M. Siberski, W., Schmitz, C., Schlosser, M. Brunkhorst, I. Löser, A. (2004) Super-peer-based routing strategies for RDF-based peer-to-peer networks, Journal of Web Semantics, 177-186.
21. Ouzzani, M., Bouguettaya, A., "Efficient Access to Web Services", IEEE Internet Computing, March/April 2004, pp. 34-44.
22. Panagis, Y., Sakkopoulos, E. Sioutas, S., Tsakalidis, A., "Web Service Integration and Management Strategies for Large Scale Datasets", in the Database Modeling for Industrial Data Management: Emerging Technologies and Applications, to be published by Idea Group Inc. in 2005.
23. Ran, S., "A Model for Web Services Discovery with QoS", ACM SIGecom Exchanges, Volume 4 , Issue 1 Spring 2003, pp. 1 - 10.
24. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S., "A Scalable Content-Adressable Network", in ACM SIGCOMM 2001, pp. 161-172
25. Ripeanu, M., Iamnitchi, A., Foster, I.T., «Mapping the Gnutella Network». IEEE Internet Computing 2002, Vol. 6, No 1, pp. 50-57
26. Root Name Servers, 2005. URL <http://www.root-servers.org/>
27. Rowstron, A., Druschel, P., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", Lecture Notes in Computer Science 2218, 2001, pp. 329-350
28. Sakkopoulos, E., Kanellopoulos, D., Tsakalidis, A., "Semantic Mining and Web Service discovery techniques for media resources management", in the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol. 1 (1), ISSN (Print): 1744-2621, Inderscience Publishers, 2005, in press.
29. Sivashanmugam, K., "Speed-R: Semantic P2P environment for diverse Web Service registries", URL <http://webster.cs.uga.edu/~mulye/SemEnt/Speed-R.html>, 2004.
30. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M.F., Dabek, F., Balakrishnan, H., Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans. Netw. 11(1): 17-32 (2003)
31. UDDI OASIS Specifications TC - Committee Specifications, 2005 URL <http://www.uddi.org>
32. Web Services Modeling Ontology URL <http://www.wsmo.org/>
33. Zhang, H., Goel, A., Govindan, R., "Incrementally Improving the Lookup Latency of Distributed Hash Table Systems", in Proceedings of ACM SIGMETRICS, June 2003.
34. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiawicz, J.D., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE Journal on Selected Areas in Communications, Vol 22, No 1, 2004, pp. 41-53
35. Jahnke, J.H, Kontogiannis, K., Stroulia, E., Tilley, S.R., Wong, K.: 3rd International Workshop on Net-Centric Computing (NCC 2001): Theme: Migrating to the Web. ICSE 2001, pp. 766-767

*Μέρος Β': Αποδοτικοί Αλγόριθμοι για
Εξατομίκευση Αναζήτησης Πληροφοριών σε
Δίκτυα και Πληροφορικά Περιβάλλοντα*

4 ΑΠΟΔΟΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΕΞΑΤΟΜΙΚΕΥΜΕΝΗΣ ΑΝΑΖΗΤΗΣΗΣ ΣΤΟ WEB

Πρόλογος – Σύνοψη Κεφαλαίου

Στα πλαίσια της εξυπηρέτησης των αναγκών των τελικών χρηστών μέσα από δικτυακά πληροφοριακά συστήματα (netcentric information systems) αναπτύχθηκαν τεχνικές για διαχείριση και εξατομίκευση (personalization) κατά την αναζήτηση πληροφορίας. Στο κεφάλαιο αυτό της διδακτορικής διατριβής παρουσιάζονται νέες προσεγγίσεις που εξατομικεύουν τα αποτελέσματα. Προτείνονται λύσεις που παρουσιάζουν τα αποτελέσματα που αντιπροσωπεύουν το χρήστη υψηλότερα στη λίστα των επιστρεφόμενων εγγράφων μετά από αναζήτηση. Οι τεχνικές αυτές σχετίζονται με την ερευνητική περιοχή της εξατομικευμένης αναζήτησης αλλά και της προσαρμοστικής παροχής αποτελεσμάτων αναζήτησης.

Στο επόμενο ενότητα παρουσιάζουμε μία νέα λύση προς την κατεύθυνση αυτή. Εισάγουμε την αναζήτηση βασισμένη στη βαθμολόγηση τοπικού συνόλου κατηγοριών που αποτελούν το προφίλ εξατομίκευσης του χρήστη. Παρουσιάζουμε νέους αποδοτικούς αλγόριθμους [25] που εκμεταλλεύονται τις κατηγορίες των επιλεγόμενων σελίδων για να εξατομικεύσουν το αποτέλεσμα της αναζήτησης. Η πειραματική αξιολόγηση με πραγματικούς χρήστες έδειξε ότι οι προτάσεις μας είναι αποδοτικές. Επιπλέον δείχνουμε ότι οι τεχνικές μας μπορούν να χρησιμοποιηθούν στην περίπτωση στοχευμένης συλλογής δεδομένων που θεωρείτε ως εναλλακτική εξατομικευμένη πρόσβαση σε πληροφορία.

Τέλος, παρουσιάζουμε ένα ακόμη μηχανισμό [15], [14] προσαρμοστικής παροχής αποτελεσμάτων αναζήτησης χρησιμοποιώντας online κατηγοριοποίηση με χρήση SVM (Support Vector Machine) [23], [7] κατηγοριοποιητή για τα αποτελέσματα που δεν έχουν προ-κατηγοριοποιηθεί από τη μηχανή αναζήτησης. Στην περίπτωση μας πέρα από κοινούς κεντρικούς κατηγοριοποιητές συνδυάζονται και εξατομικευμένες εκδόσεις τους για την περαιτέρω βελτίωση των αποτελεσμάτων.

Το παρόν κεφάλαιο έχει την ακόλουθη δομή: Στην ενότητα 4.1 δίνεται εισαγωγή και στην 4.2 παρουσιάζεται το πρόβλημα. Στην ενότητα 4.3 δίνονται σχετικές εργασίες εξατομίκευσης κατά την αναζήτηση. Στις ενότητες 4.4 έως 4.9 εισάγονται και αναλύονται τρεις νέοι αλγόριθμοι για αναζήτηση βασισμένη στη βαθμολόγηση τοπικού συνόλου κατηγοριών που αποτελούν το προφίλ εξατομίκευσης του χρήστη. Στην ενότητα 4.10 οργανώνουμε την πειραματική διαδικασία, στην 4.11 τους εξετάζουμε πειραματικά και στην 4.12 αξιολογούμε τα αποτελέσματά τους. Στην ενότητα 4.13 δίνουμε μια επένταση με εφαρμογή στη στοχευμένη συλλογή σελίδων. Στην ενότητα 4.14 παρουσιάζουμε έναν ακόμη μηχανισμό προσαρμοστικής παροχής αποτελεσμάτων χρησιμοποιώντας online κατηγοριοποίηση με χρήση SVM και αξιολογούμε τη συμπεριφορά του. Τέλος το κεφάλαιο κλείνει με την ενότητα 4.15 όπου αναλύεται η συνεισφορά, τα συμπεράσματα και τα μελλοντικά ερευνητικά βήματα της περιοχής.

4.1 Εισαγωγή

Οι σύγχρονες μηχανές αναζήτησης, παρά την αποτελεσματικότητά τους στον καθορισμό των επιθυμητών αποτελεσμάτων, παρέχουν στο χρήστη ένα πλήθος από πιθανώς σχετικές ή άσχετες απαντήσεις. Αυτό το σενάριο συμβαίνει συχνότερα στην περίπτωση που ο χρήστης παρέχει γενικούς όρους αναζήτησης οι οποίοι προκαλούν

φαινόμενα πολυσημίας, πολλαπλότητας δηλαδή εννοιών. Η «πολυσημία» αφορά το φαινόμενο στο οποίο ένας όρος εμφανίζεται σε διαφορετικά περιεχόμενα με διαφορετική έννοια. Για παράδειγμα, ο όρος “spider” περιγράφει τόσο web robots όσο αυτοκίνητα αλλά και έντομα, ενώ ο όρος “jaguar” μπορεί να υπάρχει σε περιεχόμενα σχετικά με ζώα ή σχετικά με αυτοκίνητα.

Οι μηχανές αναζήτησης αντιμετωπίζουν μερικώς αυτό το πρόβλημα κατατάσσοντας τις σελίδες που επιστρέφονται ως αποτελέσματα της αναζήτησης με βάση τη δημοτικότητα που τους έχει αποδοθεί στον δικτυακό γράφο [17]. Συνεπώς, οι χρήστες συνήθως διατρέχουν τα αποτελέσματα, ώστε να μπορέσουν να εντοπίσουν τις απαντήσεις που τους ενδιαφέρουν. Ωστόσο, η πλειοψηφία των χρηστών αποφεύγει μεγάλης διάρκειας ή αριθμού αποτελεσμάτων περιηγήσεις. Όπως παρατηρήθηκε στο [32], το 85% των χρηστών επικεντρώνεται στα δέκα πρώτα αποτελέσματα της αναζήτησης, ενώ για το 75% των ερωτημάτων οι χρήστες δεν αναθεωρούν το ερώτημα αναζήτησης μετά την πρώτη προσπάθεια. Ένα επιπλέον γεγονός είναι ότι οι χρήστες τις περισσότερες φορές έχουν εκ των προτέρων κατά νου συγκεκριμένες κατηγορίες αποτελεσμάτων που ταιριάζουν καλύτερα στις πληροφοριακές τους ανάγκες. Το γεγονός αυτό (έστω και υποσυνείδητα) παίζει καθοριστικό ρόλο στο να αποκλείσουν οι χρήστες τα ανεπιθύμητα αποτελέσματα ήδη πριν υποβάλλουν το ερώτημα για την αναζήτηση.

Ενόψει αυτού του γεγονότος, εκατοντάδες μηχανές αναζήτησης συμπεριλαμβανομένου και των πλέον γνωστών όπως το Google και το Yahoo!, τοποθετούν αναγνωριστικές ετικέτες στα αποτελέσματα αναζήτησης ανάλογα με την κατηγορία στην οποία ανήκουν [44]. Η διατήρηση των κατηγοριών αναζήτησης και η τοποθέτηση ετικετών στα αποτελέσματα είναι μια ημιαυτόματη διαδικασία. Εκτελείται κυρίως με ανθρώπινη παρέμβαση, η οποία καταλήγει σε μια μερική μόνο κάλυψη των πληροφοριών που έχουν ετικέτες, λόγω του τεράστιου όγκου των ιστοσελίδων που ανακλύπουν καθημερινά στο διαδίκτυο. Αυτή η χειροκίνητη διαδικασία διευκολύνεται εν μέρει από την ανάπτυξη του Open Directory Project (ODP) [44], μια δημόσια διαθέσιμη ιεράρχηση των θεματικών κατηγοριών σε μορφή Resource Description Framework (RDF). Κάθε URL κατατάσσεται συνήθως σε μια ODP κατηγορία. Στην περίπτωση μας χρησιμοποιήθηκε η Google Web Service για να παρέχει ένα διαφανές, αναβαθμισμένο ιστοτόπο αναζήτησης Google, ο οποίος επιστρέφει κατηγορίες για τις διαδικασίες αξιολόγησης σε όλες τις περιπτώσεις. (Ο βασικός ιστοτόπος του Google δεν παρουσιάζει πάντα την κατηγορία, ωστόσο το Google Web Service API την επιστρέφει κάθε φορά. Επιπλέον, το τμήμα καταλόγου του Google (Google directory section) παρουσιάζει επίσης την κατηγοριοποίηση όπως κάνει εξ' ορισμού και το Yahoo!.)

4.2 Περιγραφή του Προβλήματος

Παρόλα αυτά, οι διαθέσιμες κατηγοριοποιήσεις που θα μπορούσαν επίσης να βοηθήσουν το έργο του χρήστη, είναι αρκετά χαοτικές και επομένως αποθαρρύνουν

τους χρήστες από το να αναγνωρίσουν τις κατάλληλες κατηγορίες πριν προχωρήσουν στην αναζήτηση.

Αυτό το κεφάλαιο στοχεύει να παρουσιάσει μια σειρά από αλγορίθμους που θα επιτρέψουν σε κάθε χρήστη να έχει έναν εξατομικευμένο υπογράφο σημαντικά μικρότερο από το συνολικό γράφο κατηγοριών ODP. Σημειώνεται ότι αυτός ο υπογράφος είναι δυνατό να δημιουργείτε χωρίς να απαιτείται καμία ιδιαίτερη προσπάθεια από τους χρήστες. Παράγεται αυτόματα μετά από μια περίοδο καταγραφής των περιηγήσεων των χρηστών στα παλαιότερα αποτελέσματα αναζήτησης, τα οποία θεωρούμε ότι αντικατοπτρίζουν τα ενδιαφέροντά τους.

Ειδικότερα, με βάση τα αποτελέσματα αναζήτησης, ο χρήστης μπορεί να επιλέξει μια σελίδα σε μια προσπάθεια να ικανοποιήσει κάποια συγκεκριμένη πληροφοριακή ανάγκη. Η λύση που προτείνουμε καταγράφει το ιστορικό περιηγήσεων έμμεσα αναγνωρίζοντας τις κατηγορίες που επιλέγει ένας χρήστης. Αυτό επιτρέπει την κατασκευή ενός πολύ μικρότερου υποσυνόλου κατηγοριών σε σχέση με ολόκληρο το ODP και επιπλέον επιτρέπει την υλοποίηση μιας νέας λογικής κατάταξης των κατηγοριών. Η κατάταξη κατηγοριών συνδυάζεται συνεπώς με τα αποτελέσματα αναζήτησης και με αποτέλεσμα η ταξινόμηση μιας μηχανής αναζήτησης να παράγει αποτελέσματα αναζήτησης που προσαρμόζονται καλύτερα στο προφίλ αναζήτησης του χρήστη. Οι αλγόριθμοι αυτοί σχετίζονται σχετίζεται με άλλες προσεγγίσεις που εκμεταλλεύονται τη χειροκίνητη ή αυτόματη κατηγοριοποίηση των σελίδων με σκοπό να εξάγουν γνώση από το δίκτυο σύμφωνα και με τις εργασίες μας [15], [14]. Ωστόσο όπως προτείνουμε στο [25], μπορούμε να εκμεταλλευόμαστε αυτήν την πληροφορία διαφορετικά για καλύτερη ποιότητα αναζήτησης σε δικτυακά πληροφοριακά περιβάλλοντα και το διαδίκτυο εφαρμόζοντας άμεσα τεχνικές μετρικής συνδέσμων (link metric) στο γράφο κατηγοριών. Η λύση που προτείνουμε ελέγχει την ιστορία περιηγήσεων των χρηστών και ενημερώνει το προφίλ τους όποτε οι συνήθειες περιήγησης τους αλλάζουν. Όταν ο χρήστης υποβάλλει μια ερώτηση, την επόμενη φορά τα αποτελέσματα αναζήτησης προσαρμόζονται με βάση το νέο προφίλ χρήστη.

Πραγματοποιήσαμε μια εκτενή πειραματική μελέτη των προτεινόμενων τεχνικών, όπου η μεθοδολογία μας επέδειξε αυξημένη ακρίβεια στη λίστα των επιστρεφόμενων αποτελεσμάτων αναζήτησης. Τέλος, παρουσιάζουμε τον τρόπο που οι τεχνικές μας μπορούν να προσαρμοστούν προκειμένου να σχεδιαστούν μηχανισμοί στοχευμένης στο θέμα συλλογής σελίδων(topic-focused crawlers) [8], [20].

4.3 Σχετικές εργασίες

Στην ενότητα αυτή παρουσιάζονται υπάρχουσες εργασίες για εξατομικευμένη αναζήτηση σε δικτυακά πληροφοριακά περιβάλλοντα και εκ των υστέρων (post)-κατηγοριοποίηση αποτελεσμάτων.

Η αναζήτηση στον ιστό, όπως έχει εξελιχθεί με την ραγδαία εξάπλωση του Παγκόσμιου Ιστού (World Wide Web), είναι μια περιοχή που έχει ερευνηθεί

λεπτομερώς. Έχουν παρουσιαστεί πολλές τεχνικές αναζήτησης (δείτε [19], [29]) με πολυάριθμες παραλλαγές. Δύο από τις πιο πρακτικά και θεωρητικά επιτυχημένες είναι οι τεχνικές που παρουσιάζονται στα [5] και [23]. Στην περιοχή των μηχανών αναζήτησης, ένας δείκτης κατηγορίας σε κάθε έγγραφο που επιστρέφεται παρουσιάζεται στα Yahoo!, το A9 και το Google ([39], [37], [43]). Σύμφωνα με το [4] αυτός ο δείκτης διευκολύνει τις αναζητήσεις, καθώς οι περισσότεροι χρήστες τείνουν να ακολουθήσουν τις υψηλά καταταγμένες σελίδες ακόμα και αν είναι πιθανώς άσχετες με το αρχικό αίτημα. Τα WiseNut, Adutoria και Vivisimo ([42], [38], [41]) παρέχουν συσταδοποίηση των αποτελεσμάτων σε κατηγορίες. Το Teoma [40] επίσης συσταδοποιεί τα αποτελέσματα του και παρέχει βελτιώσεις των ερωτημάτων. Τα περιβάλλοντα Metasearch υλοποιούν στρατηγικές που αντιστοιχούν τα ερωτήματα χρήστη σε συλλογές [26], [21]. Σε παρόμοιο πλαίσιο, τα προφίλ χρήστη κατασκευάζονται είτε έμμεσα είτε άμεσα στα περιβάλλοντα αναζήτησης όπως παλαιότερα έχουν κάνει το Letizia και το CiteSeer. Αυτά τα προφίλ βοηθούν το έργο του χρήστη παρέχοντας προτάσεις αναζήτησης ανάλογα με το που περιηγείται σε τρέχον χρόνο ο χρήστης.

Η στοχευμένη συλλογή σελίδων (Focused crawling) είναι μια άλλη τεχνική για να φιλτραριστούν τα μη συσχετιζόμενα έγγραφα. Οι μηχανισμοί στοχευμένης συλλογής σελίδων (Focused crawlers) [11], [8], [10], [27], [20] απορρίπτουν σημαντικά μέρη του γράφου του ιστού (web graph) και επικεντρώνονται μόνο σε έγγραφα που κρίνονται σχετικά με μια προκαθορισμένη ιεραρχία κατηγορίας. Η σχετικότητα επιτυγχάνεται με τη χρησιμοποίηση στρατηγιών κατηγοριοποίησης κειμένου. Οι Chakrabarti et al. [8] αναφέρουν μεθόδους για κατηγοριοποίηση εγγράφων υπερκειμένου με βάση τους υπερσυνδέσμους. Οι μέθοδοί τους αποδείχθηκε ότι ξεπερνούν τους παραδοσιακούς ταξινομητές κειμένων.

Η εργασία μας συσχετίζεται περισσότερο με την έρευνα στην προσαρμοσμένη και εξατομικευμένη αναζήτηση Ιστού. Ένα σημαντικό βήμα σε αυτήν την περιοχή ήταν η εργασία του Haveliwala [17] που εισάγει την «εξαρτημένη ταξινόμηση (PageRank) από το θέμα (topic sensitive)». Η εξαρτημένη από το θέμα ταξινόμηση σελίδων είναι ουσιαστικά μια μέθοδος που κατευθύνει το PageRank προς συγκεκριμένα θέματα. Σε αυτήν την εργασία, αναφέρεται μια επιτυχημένη μεροληψία (biasing) του PageRank προς συγκεκριμένα θέματα, ωστόσο διεξάγεται λίγος πειραματισμός σε ένα εξατομικευμένο πλαίσιο. Στο [9] προτείνονται αλγόριθμοι για τη δημιουργία εξατομικευμένων εγγράφων αυθεντίας (authority) για να αντιστοιχούν περισσότερο στο εσωτερικό μοντέλο του χρήστη, ακολουθώντας τις συνθήκες του αλγορίθμου HTS του Kleinberg [23]. Οι Liu et al. [24] προτείνουν επίσης μια τεχνική για την αντιστοίχιση ερωτημάτων σε κατηγορίες προκειμένου να βοηθηθεί η διαδικασία αναζήτησης. Η τεχνική τους διατηρεί πληροφορίες για το ιστορικό αναζήτησης χρήστη με την κατασκευή ενός Rocchio-based προφίλ χρήστη. Παράγουν επίσης το γενικό βασισμένο στο ODP προφίλ και συνδυάζουν και τα δύο παραγόμενα προφίλ για να ενισχύσουν τα αποτελέσματα αναζήτησης. Μια άλλη σχετική τεχνική που λαμβάνει υπόψη με έμμεσο τρόπο της γενικής συμπεριφορά και επισκεψιμότητα

κατά την περιήγηση του χρήστη για να προσαρμόσει τα αποτελέσματα αναζήτησης παρουσιάζεται στο [33].

4.4 Αλγόριθμοι εξατομίκευσης αναζήτησης

Εισάγουμε μία νέα τεχνική εξατομίκευσης [25] που χρησιμοποιεί ένα εξατομικευμένο υποσύνολο της δομής κατηγοριών ODP που συνδέεται με τις ανάγκες πληροφόρησης των χρηστών. Αυτό το υποσύνολο υποβάλλεται σε επεξεργασία τοπικά σε κάθε χρήστη, με στόχο την ενίσχυση των γενικών αποτελεσμάτων που προσφέρονται από τις μηχανές αναζήτησης. Η εξατομικευμένη αναζήτηση σε δικτυακά περιβάλλοντα και δει στο Web έχει απασχολήσει μια σειρά από εργασίες και προσεγγίσεις που την αντιμετωπίζουν με τεχνικές στη μεριά του εξυπηρετητή (server side) [11], [8], [10], [27]. Στην πραγματικότητα οι τεχνικές αυτές προσπαθούν να αντλήσουν τις πληροφορίες από το δίκτυο και να τις οργανώσουν, ώστε να μπορεί κανείς να έχει πρόσβαση σε αυτές αποδοτικά. Ωστόσο δε λαμβάνουν υπόψη τους τις προσωπικές απαιτήσεις και επιθυμίες του κάθε χρήστη. Για να εξατομικευτούν τα αποτελέσματα αναζήτησης μπορούμε να θεωρήσουμε ότι συνολικά χρησιμοποιούνται τρεις μορφές προσεγγίσεων:

- Η άμεση ανάδραση από το χρήστη (user feedback) [31] της σχετικότητας των σελίδων
- Η καταγραφή δημογραφικών και ενδιαφερόντων μετά από εγγραφή του χρήστη
- Χρήση της βαθμολόγησης σελίδων από προηγούμενους χρήστες με συνεργατικές μεθόδους.

Αυτές οι τεχνικές απαιτούν εγγραφή, ανάδραση ή βαθμολόγηση και επομένως χρόνο που συνήθως οι χρήστες δεν είναι πρόθυμοι να ξοδέψουν [33]. Μια υποσχόμενη πρόσφατη αντιμετώπιση για την εξατομίκευση των αποτελεσμάτων λαμβάνει υπόψη τις συνολικές επισκέψεις του διαφυλλιστή του χρήστη (web browsing history) ανά ημέρα [33]. Ωστόσο η ανάλυση των στοιχείων του χρήστη γίνεται ασύγχρονα την επόμενη ημέρα λόγω μεγάλου όγκου δεδομένων που πρέπει να περάσουν από επεξεργασία και επιπλέον λαμβάνουν υπόψη όλες τις επισκέψεις του χρήστη δημιουργώντας ένα πολύ μεγάλο σε όγκο προφίλ που πρέπει να αποθηκεύεται τοπικά.

Εμείς έχουμε ως κύριο στόχο να εκμεταλλευτούμε μία πληροφορία που υπάρχει ήδη για τις σελίδες, την κατηγοριοποίησή τους, και που ο χρήστης χρησιμοποιεί έμμεσα κατά την επιλογή τους. Η βασική μας ιδέα εντοπίζεται στην εισαγωγή και μετακύληση των βαρών σημασίας σε εκείνα τα μέρη της δομής ODP που αντιστοιχούν στις προσωπικές προτιμήσεις του χρήστη καθώς και η σημασιολογική διασύνδεση των κατηγοριών που ο χρήστης συνδυάζει κατά την πλοήγηση στα αποτελέσματα της αναζήτησής του [25]. (Υπενθυμίζουμε ότι το ODP χρησιμοποιείται από όλες τις σύγχρονες μηχανές αναζήτησης ως βάση για την

κατηγοριοποίηση των σελίδων τους ακόμη και για τις εκδόσεις τοπικών συστημάτων (desktop search engines) [44]. Έτσι επιτυγχάνουμε να εξατομικεύουμε τη λίστα αποτελεσμάτων με αποτελεσματικό και γρήγορο τρόπο στην πλευρά του χρήστη μιας και τα δεδομένα που επεξεργαζόμαστε είναι πολύ λιγότερα. Με τον τρόπο αυτό δεν απαιτείται ούτε από τη μηχανή αναζήτησης να κρατά κεντρικά προφίλ χρηστών οπότε κερδίζουμε σε χώρο αποθήκευσης, ούτε να απαιτείται αυθεντικοποίηση οπότε κερδίζουμε σε αριθμό βημάτων (κλικ) κατά την αναζήτηση [33]. Η εισαγωγή και ενημέρωση των βαρών πραγματοποιείται με έμμεσο τρόπο και με βάση τη συμπεριφορά και επισκεψιμότητα στοχευμένα για την περιήγηση σε αποτελέσματα προηγούμενων αναζητήσεων. Τα αποτελέσματα της τοπικής επεξεργασίας συνδυάζονται στη συνέχεια με τη σφαιρική γνώση για να παραγάγουν μια συνολική εξατομικευμένη ταξινόμηση των αποτελεσμάτων. Στην ακόλουθη υποενότητα περιγράφουμε λεπτομερέστερα το θεωρητικό πλαίσιο που χρησιμοποιείται και τα αλγοριθμικά βήματα των τριών προτεινόμενων αλγορίθμων το υλοποιούν.

4.5 Προαπαιτούμενα

Το γενικό προτεινόμενο πλαίσιο είναι το ακόλουθο: Θεωρούμε μια τυχαία μηχανή αναζήτησης που χρησιμοποιεί ένα κατευθυνόμενο άκυκλο γράφο κατηγοριών – directed acyclic graph (DAG) $G(V,E)$, με στόχο να κατηγοριοποιήσει ιστοσελίδες. Ο γράφος G αποτελείται από κόμβους $v \in V$ που δηλώνουν κατηγορίες και κάθε ακμή $(v_i, v_j) \in E$ δηλώνει ότι το v_j είναι υποκατηγορία της v_i και της ανατίθεται ένα βάρος $d(v_i, v_j) \in [0,1]$. Αυτό δηλώνει μια σημασιολογική συσχέτιση μεταξύ των κατηγοριών v_i, v_j . Επιπλέον θεωρείται ότι κάθε ιστοσελίδα έχει ετικέτα με μια συγκεκριμένη κατηγορία. Σε αντίθετη περίπτωση πραγματοποιούμε επί τόπου κατηγοριοποίηση σύμφωνα με την προσέγγιση μας που παρουσιάζουμε σε επόμενη υποενότητα παρακάτω και στις εργασίες [15], [14].

Συνολικά, η προτεινόμενη προσέγγιση εισάγει την ιδέα της αυξητικής δημιουργίας, στην πλευρά του πελάτη, ενός υπερ-γράφου G' ενός κατάλληλα επιλεγμένου υπο-γράφου G_{sub} του G . Αυτός ο υπο-γράφος μπορεί να κατασκευαστεί με την καταχώρηση των προτιμήσεων των χρηστών έμμεσα ή άμεσα (αφήνοντας δηλαδή το χρήστη να δηλώσει τις προτιμήσεις του). Στην ακραία περίπτωση ισχύει $G_{sub} \equiv G$. Σε κάθε κατηγορία v του G' θα οριστεί ένα βάρος σχετικότητας-σημασίας $\beta(v) > 0$. Αυτά τα βάρη χρησιμοποιούνται προκειμένου να ταξινομηθούν οι σελίδες που επιστρέφονται στον τελικό χρήστη, κατά την υποβολή μιας ερώτησης. Ειδικότερα, η κατάταξη (ranking) μιας σελίδας p στο σύνολο αποτελεσμάτων ενός τυχαίου ερωτήματος χρήστη θα δοθεί από μια συνάρτηση της μορφής: $\phi(\beta(\gamma(p)), \sigma(p))$

Στην παραπάνω συνάρτηση,

- $\gamma(p)$ είναι η κατηγορία στην οποία ανήκει μια σελίδα p ,

Σχήμα 25: Παράδειγμα εξατομίκευσης με χρήση εικονικών κόμβων κατηγοριών
Οι γκρι κόμβοι δείχνουν τα μέρη του ODP που ο χρήστης ίσως βρει ενδιαφέροντα. Τα μέρη των μονοπατιών που φαίνονται με διακεκομμένες γραμμές παραλείπονται λόγω έλλειψης χώρου. Οι τετράγωνοι κόμβοι αναπαριστούν τους κόμβους εικονικών κατηγοριών που προστέθηκαν.

- $\sigma(p)$ είναι η σχετικότητα-σημασία σύμφωνα με τον αλγόριθμο ταξινόμησης της μηχανής, και
- η συνάρτηση $\phi(\cdot)$ δείχνει πώς η τελική ταξινόμηση θα είναι προκατειλημμένη προς την ταξινόμηση μηχανής ή την σπουδαιότητα κατηγορίας που ορίζεται (για παράδειγμα $\phi(\alpha, \beta) = (\alpha + \beta)/2$).

Γενικά, έχουμε εισαγάγει τη συνάρτηση $\phi(\cdot)$ που συνδυάζει τη ταξινόμηση της μηχανής αναζήτησης και την προτεινόμενη εξατομικευμένη ταξινόμησης μας προκειμένου να παρασχεθεί καλύτερη εξελιξιμότητα (scalability) της λειτουργικής διαμόρφωσης της λύσης μας (solution configuration). Στη συνέχεια προτείνονται αρχικά δύο μέθοδοι για την κατασκευή του G' . Οι δύο προσεγγίσεις έχουν παρόμοια προσέγγιση.

Ο πρώτος αλγόριθμος, ο offline όπως τον αποκαλούμε στην εργασία [25], επινοήθηκε με σκοπό να επιτρέψει την ασύγχρονη δημιουργία του προφίλ χρήστη μετά από μια περίοδο παρατήρησης σχετικά με τις συνήθειες αναζήτησης του χρήστη. Ο γράφος G' κατασκευάζεται μετά από αυτήν την περίοδο παρατήρησης και παραμένει ίδιος για όση διάρκεια εκτιμάτε ότι δεν έχουν συμβεί μεγάλες αλλαγές στις προτιμήσεις αναζήτησης. Αυτή η προσέγγιση μπορεί να εφαρμοστεί επανειλημμένα για να δημιουργήσει και να ενημερώνει το προφίλ σχετικά με τις προτιμήσεις του χρήστη σε μια ιστορική διαδρομή. Αυτό μπορεί να πραγματοποιείτε είτε σε προκαθορισμένες περιόδους (μετά από κάθε σύνοδο περιήγησης, κάθε μέρα είτε έπειτα από αίτημα του χρήστη (άμεση κλήση της μεθόδου από το χρήστη). Σε κάθε περίπτωση η τεχνική μας είναι δυνατό να

διαμορφωθεί ανάλογα και σύμφωνα με την επιθυμία του χρήστη. Τεχνικές χρονοπρογραμματισμού ανανέωσης βέβαια προφίλ δίνονται στην εργασία [33].

Η δεύτερη προσέγγιση, η online όπως την ονομάζουμε στην εργασία [25], ενημερώνει το G' μετά από κάθε ερώτημα. Με τον τρόπο αυτό υπάρχει μια άμεση συσχέτιση των επιλεγόμενων αποτελεσμάτων, και έμμεσα των κατηγοριών τους, με τις ιδιαίτερες προτιμήσεις του χρήστη στην περίοδο μιας συγκεκριμένης συνόδου κατά οποία ο χρήστης αναζητά στοχευόμενα μια πληροφορία. Αυτές οι κατηγορίες μπορεί να μη συνδέονται άμεσα στον αρχικό γράφο G , ωστόσο οι επιλογές του χρήστη αναδεικνύουν μια σημασιολογική συσχέτιση την οποία εκμεταλλευόμαστε σε πραγματικό χρόνο με σκοπό να παρουσιάσουμε μια εξατομικευμένη λίστα κατάταξης των αποτελεσμάτων αναζήτησης του στα αμέσως επόμενα ερωτήματα.

Η λογική πίσω από το G' είναι να δημιουργηθεί ένας νέος εξατομικευμένος γράφος κατηγορίας από μια γενική ιεραρχία κατηγοριών. Κατά συνέπεια η τεχνική μας είναι δυνατό να επεκταθεί στο μέλλον ώστε να παρέχει εξατομίκευση με βάση έννοιες και κατηγορίες από σημασιολογικές ιεραρχίες.

Το G' είναι στην πλειοψηφία των περιπτώσεων ένα πολύ μικρό μέρος της συνολικής γενικής ιεραρχίας, το οποίο περιλαμβάνει τις κατηγορίες που απεικονίζουν καλύτερα τις εξατομικευμένες προτιμήσεις αναζήτησης με βάση τον έμμεσο έλεγχο του ιστορικού περιήγησης των χρηστών στους καταλόγους αποτελέσματος των μηχανών αναζήτησης. Η μεθοδολογία μας είναι βασισμένη στην ακόλουθη παρατήρηση: σε κάθε προσπάθεια αναζήτησης, ανεξάρτητα από το πραγματικό ερώτημα, οι χρήστες επιλέγουν να επισκεφτούν μόνο μερικά από τα επιστρεφόμενα αποτελέσματα, τα οποία δεν ανήκουν απαραίτητα στην ίδια κατηγορία [25]. Αυτό δείχνει ότι υπάρχει μια έμμεση συσχέτιση των κατηγοριών για τις συγκεκριμένες προτιμήσεις χρηστών. Αυτές οι κατηγορίες δεν μπορούν να συνδεθούν άμεσα στον αρχικό γράφο κατηγορίας G , εντούτοις ο χρήστης θα επιθυμούσε να τους συνδέσει προκειμένου να πάρει μια εξατομικευμένη λίστα ταξινόμησης των αποτελεσμάτων αναζήτησης του/της στα μελλοντικά ερωτήματα κατά τη διάρκεια μιας συγκεκριμένης συνόδου.

Επιπλέον, οι χρήστες μπορεί να επιλέξουν να συσχετίσουν μια κατηγορία v , η οποία ήταν της μακροχρόνιας προτίμησής τους και επομένως να έχει διατηρήσει μια τιμή $\beta(v)$, με μια κατηγορία w η οποία μόλις κέρδισε την προσοχή τους. Θεωρούμε για παράδειγμα ένα χρήστη που αναζητά πάντα για αποτελέσματα ποδοσφαίρου και σε μια συγκεκριμένη στιγμή αποφασίζει να αναζητήσει ταινίες σχετικές με ποδόσφαιρο. Σε αυτήν την περίπτωση μπορεί να βρει σχετικά αποτελέσματα τόσο σε κατηγορίες σχετικές με ποδόσφαιρο όσο και σε κατηγορίες σχετικές με ταινίες. Η παραπάνω συζήτηση παρουσιάζει δύο ανάγκες: την ανάγκη για συσχετισμό ασύνδετων κατηγοριών και την ανάγκη να διαδώσει τη σπουδαιότητα μεταξύ τους. Για να καλύψουμε την ανάγκη αυτή εισάγουμε την έννοια των εικονικών κόμβων κατηγορίας, που παρουσιάζεται στην παράγραφο 4.3.4 και σύμφωνα με την εργασία [25].

4.6 Αλγόριθμος Off-Line

Υποθέτουμε ότι εκτελούνται k υπο-πειράματα, τα οποία είναι σύνοδοι περιήγησης (browsing sessions) του χρήστη που συνεπάγονται την υποβολή ενός ερωτήματος και την περιήγηση στα ανακτημένα αποτελέσματα. Κατά τη διάρκεια αυτών των υπο-πειραμάτων το ιστορικό περιήγησης του χρήστη παρακολουθείται έμμεσα χωρίς καμία ιδιαίτερη προσπάθεια από τους χρήστες. Συγκεκριμένα, καταγράφονται οι κατηγορίες των σελίδων που επιλέγονται κατά τη διάρκεια των υπο-πειραμάτων. Αρχικά έχουμε $G' = G_{sub}$.

Θεωρούμε το i -οστό υπο-πείραμα, $i \leq k$ το οποίο περιγράφεται που δίνει ο αλγόριθμος 12. Με την ολοκλήρωση των k πειραμάτων, το $\beta(v)$ λαμβάνεται ως μέσος όρος των ενδιάμεσων τιμών που λαμβάνονται από τα υπο-πειράματα.

Experiment i

Έστω ότι S_i είναι το σύνολο των σελίδων αποτελέσματος και ότι $a_i(p)$ είναι η τιμή σπουδαιότητας που ανατίθεται στη σελίδα p από το χρήστη ($a_i(p) \in [0,1]$).

Σημειώστε ότι η ανάθεση σπουδαιότητας εκτελείται έμμεσα χωρίς καμία ιδιαίτερη προσπάθεια από το χρήστη συνολικά στις τιμές $\beta_i(v)$ όπως περιγράφεται παραπάνω.)

Βήμα 1. Υποβάλλουμε ένα ερώτημα, συλλέγουμε τα S_i και επιτρέπουμε στους χρήστες να ορίσουν το $a_i(p)$.

Βήμα 2. Για κάθε κατηγορία v του G' , ορίζουμε

$$\beta_i(v) = \frac{1}{n} \sum_{j=1}^n a_i(p_j) ,$$

όπου το $p_j \in S_i$, και το p_j ανήκει στην κατηγορία v , όπου n είναι ο αριθμός των σελίδων του S_i με κατηγορία v . (Σημειώστε ότι το p_j ανήκει στην κατηγορία v και όχι στις υποκατηγορίες της).

Βήμα 3. Υποθέτουμε ότι $v_1, v_2, \dots, v_l$ είναι όλες οι κατηγορίες των οποίων το βάρος έχει υπολογιστεί μεγαλύτερο από ένα προκαθορισμένο κατώφλι $thres1$. Δημιουργούμε έναν εικονικό κόμβο v που θα είναι ο κοινός πατέρας των $v_1, v_2, \dots, v_l$ και θα έχει αρχικό βάρος $\beta(v) = 0$.

Χειρισμός Εξαιρέσεων: Στην ειδική περίπτωση όπου ένας κανονικός κόμβος ODP υπάρχει ήδη ως κοινός πατέρας των $v_1, v_2, \dots, v_l$, δε χρειάζεται να δημιουργηθεί κανένας εικονικός κόμβος και αυτό το βήμα παραλείπεται

Αλγόριθμος 12: Αλγόριθμος Experiment i

Το κατώφλι $thres1$ διασφαλίζει ότι μόνο οι κορυφαίες προτιμώμενες κατηγορίες διασυνδέονται με έναν κοινό πατέρα - εικονικό κόμβο. Αυτό το κατώφλι καθορίζεται συνήθως έμμεσα ορίζοντας ένα μέγιστο αριθμό συσχετισμένων κατηγοριών κάτω από έναν εικονικό κόμβο. Η λογική πίσω από την εισαγωγή του κατωφλίου είναι ότι μόνο οι πολύ συχνά επισκεπτόμενες κατηγορίες είναι σημαντικά σημασιολογικά συσχετισμένες και επομένως κρίσιμες για την εξατομίκευση. Στα πραγματοποιημένα

πειράματα ένας εικονικός κόμβος ήταν περιορισμένος για να δεχτεί το πολύ πέντε (5) κατηγορίες (τιμή `thres1`).

Συνεπώς, ενημερώνουμε κάθε κατηγορία $v \in G'$, ως εξής:

$$\beta(v) = \beta(v) + \sum_{i=1}^{outdegree(v)} d(v, v_i) \beta(v_i) + \sum_{i=1}^{indegree(v)} d(v_i, v) \beta(v_i) \quad (1)$$

Αυτό το άθροισμα πρέπει να υπολογιστεί επαναληπτικά αρκετές φορές εωσότου να διαδοθούν τα σωστά βάρη σε ολόκληρο τον κόμβο.

Offline_Categories (G_{sub})

Έστω k μια προκαθορισμένη μικρή σταθερά, και G' ο γράφος που αυξανόμενα αντικαθιστά το G_{sub} .

```

G' = Gsub ;
for i=1 to k do
    Experiment_i ;
for each v ∈ G'
     $\beta(v) = \frac{1}{k} \sum_{i=1}^k \beta_i(v)$  ;
Update_Weights(G') ;
return G' ;

```

Αλγόριθμος 13: Αλγόριθμος Offline_Categories

Update_Weights(G)

```

for i=1 to c do
    for each v ∈ G
        update  $\beta(v)$  according to equation (1) (pre-updated values of  $\beta(v_{i-1})$ 
        are only used)
return G;

```

Αλγόριθμος 14: Αλγόριθμος Update_Weights

Η προκαθορισμένη μικρή σταθερά k είναι ουσιαστικά ο αριθμός των ερωτημάτων – περιόδου περιήγησης – πριν θεωρήσουμε το τέλος μιας “συνόδου περιήγησης”. Όπως αναφέρθηκε ήδη στην εισαγωγή, η offline προσέγγιση μπορεί να εφαρμοστεί επαναληπτικά για να δημιουργήσει και να ενημερώσει το προφίλ του χρήστη σχετικά με τις προτιμήσεις του χρήστη είτε σε προκαθορισμένες περιόδους είτε έπειτα από αίτημα του χρήστη.

4.7 Αλγόριθμος OnLine

Ο online αλγόριθμος συνεπάγεται την εκτέλεση των υπολογισμών που περιγράφηκαν παραπάνω, ωστόσο, ο αλγόριθμος 12 Experiment_i εκτελείται για κάθε ερώτημα χρήστη και όχι μετά από ένα αριθμό συνόδων. Επομένως, ο τροποποιημένος κώδικας για τον online αλγόριθμο φαίνεται ως αλγόριθμος 15.

Online_Categories (G_{sub})

Έστω G' ο γράφος που αυξανόμενα αντικαθιστά το G_{sub} .

Συνεπάγεται ότι αυτός ο αλγόριθμος εκτελείται με διαφανή τρόπο σε κάθε ερώτημα χρήστη

$G' = G_{sub}$;

Experiment_i ;

Update_Weights(G') ;

return G' ;

Αλγόριθμος 15: Αλγόριθμος Online_Categories

4.8 Σημασιολογική συσχέτιση με Κόμβους Εικονικών Κατηγοριών

Στόχος μας όπως είπαμε είναι επιπλέον να αποδώσουμε τη σημασιολογική συσχέτιση που υποδεικνύει η επιλογή αποτελεσμάτων που ανήκουν σε ασύνδετες κατηγορίες στην ιεραρχία ODP. Για το λόγο αυτό εισάγουμε την έννοια των κόμβων εικονικών κατηγοριών [25]. Συγκεκριμένα οι λεπτομέρειες των κόμβων αυτών και σε ό,τι αφορά τα επιμέρους στοιχεία των προηγούμενων αλγορίθμων εξηγούνται περαιτέρω.

Ο αλγόριθμος 12 Experiment_i περιλαμβάνει το βήμα 3 κατά το οποίο προστίθεται ένας επιπλέον κόμβος-κατηγορία ως ένας κοινός άμεσος πρόγονος κάποιων κατηγοριών στις οποίες δόθηκε μεγαλύτερη σημασία από ότι άλλες σε ένα συγκεκριμένο ερώτημα. Αυτή η επιλογή επιβάλλει έναν έμμεσο συσχετισμό αυτών των κατηγοριών ($v_1, v_2, \dots, v_i$) στα πλαίσια ενός συγκεκριμένου ερωτήματος. Ο συσχετισμός αυτός προκύπτει από το γεγονός ότι ο χρήστης αναθέτει υψηλούς βαθμούς σε όλες τις κατηγορίες (μεγαλύτερες από το *thres1*). Επομένως δημιουργείται ένας νέος κόμβος κατηγορίας που διασυνδέει σημασιολογικά και ουσιαστικά στον εξατομικευμένο γράφο τις i κατηγορίες ως υποκατηγορίες του.

Αυτός ο εικονικός κόμβος κατηγορίας θεωρείται ως ένα κανάλι κατανομής «σπουδαιότητας» μεταξύ των απογόνων του, άμεσων και έμμεσων. Ο αλγόριθμος 14 Update_Weights υλοποιεί τη διαδικασία αυτή που κατανέμει τη σπουδαιότητα που προσδίδει ο χρήστης από έναν κόμβο στους γειτνιάζοντες κόμβους του, ενημερώνοντας σε κάθε επανάληψη το βάρος κάθε κατηγορίας. Η ενημέρωση εκτελείται με βάση ότι ένας κόμβος u κερδίζει από κάθε γείτονα v , ένα ποσό σπουδαιότητας p που ισούται με $\beta(v) \times d(u, v)$ (ή $\beta(v) \times d(v, u)$). Η τελευταία ποσότητα βρίσκεται σε κάθε άθροισμα της σχέσης (1) παραπάνω και μεταφράζεται ως εξής: *σπουδαιότητα γείτονα $\times$ συσχετισμός κατηγορίας*.

Το σχήμα 25 απεικονίζει με λεπτομέρειες τη διαδικασία που πραγματοποιείται τοπικά. Οι γκρι κόμβοι αντιπροσωπεύουν τα ενδιαφέροντα υποσύνολα του ODP. Οι τετραγωνικοί κόμβοι είναι εικονικές κατηγορίες που συσχετίζουν τις διαφορετικά πιθανώς ανεξάρτητες κατηγορίες και επιτρέπουν μια ροή των βαρών σπουδαιότητας μεταξύ τους. Ο κόμβος v_i μπορεί να αντιπροσωπεύσει μια ερώτηση του τύπου

«Τίτλοι Ταινιών Bollywood Soundtracks» ενώ ο v_2 μια ερώτηση όπως «Τίτλοι ταινιών στους κινηματογράφους του Λονδίνου».

Η ενημέρωση συνεχίζει έως ότου όλοι οι κόμβοι του γράφου να περιέχουν τις τελικές τιμές τους. Θεωρούμε ότι:

- Ο γράφος για κάθε χρήστη είναι ένα ζυγισμένο δυαδικό δέντρο με κατά προσέγγιση 650,000 κόμβους το μέγιστο (αυτός είναι και ο συνολικός αριθμός όλων των κατηγοριών του ODP [44]).
- Η ενημέρωση παίρνει την τελική της μορφή μετά από έναν μικρό αριθμό βημάτων c , όπου c είναι κοντά στη διάμετρο D του γράφου.

Σύμφωνα με την πρώτη υπόθεση το D ισούται με το ύψος του δέντρου, το οποίο είναι λογαριθμικό στον αριθμό των κόμβων. Αυτό σημαίνει ότι:

$$c \approx \lceil \log_2 650,000 \rceil = 20$$

Ωστόσο, έχουμε παρατηρήσει κατά τη διάρκεια των πειραμάτων αξιολόγησης ότι στην πράξη αρκεί ένας μικρότερος αριθμός επαναλήψεων για να πάρουν τα βάρη τις κατά προσέγγιση τελικές τους τιμές. Στην πειραματική αξιολόγηση, δώδεκα (12) επαναλήψεις ήταν αρκετές για να ενημερώσουν τα βάρη κατηγοριών και να εκτελεστεί επιτυχημένη εξατομίκευση των αποτελεσμάτων αναζήτησης. Συγκρίναμε αυτά τα πειραματικά αποτελέσματα με νέα που λάβαμε χρησιμοποιώντας μόνο οκτώ (8) κύκλους ενημέρωσης και τα αποτελέσματα ήταν επίσης επαρκή στην επιστροφή κατάλληλων εξατομικευμένων λιστών ταξινόμησης των αποτελεσμάτων αναζήτησης σε σχεδόν το μισό χρόνο επεξεργασίας (45% βελτίωση). Παρατηρήσαμε 99% ομοιότητα με τη λίστα αποτελεσμάτων μεταξύ των δύο περιπτώσεων, η οποία δείχνει ότι η υπόθεσή μας είναι έγκυρη και υπολογιστικά αποτελεσματική.

4.9 Μεθοδολογία υλοποίησης εξατομίκευσης κατηγοριών

Όπως αναφέρθηκε ήδη, η μεθοδολογία που προτείνουμε εκμεταλλεύεται τις κατηγορίες του ODP για να διευκολύνει την εξατομικευμένη αναζήτηση. Οι κατηγορίες του ODP είναι διαθέσιμες σε μορφή RDF [45]. Η ποσότητα των πληροφοριών κατηγορίας που διατηρείται στο ODP είναι ιδιαίτερα μεγάλη και φτάνει τις 650.000.

Στο σημείο αυτό πρέπει να αναφερθεί ότι ο αλγόριθμος αναζήτησης πάνω στον οποίον πειραματιστήκαμε είναι ένα υβρίδιο που συνδυάζει ιδέες και από τις δύο προσεγγίσεις των 4.6 και 4.7. Ο υλοποιημένος αλγόριθμος 16 *Semi_Offline_Categories* πραγματοποιεί υπολογισμό των βαρών σπουδαιότητας των κατηγοριών του ODP εκτελείται online σε κάθε υπο-πειράμα, εκτελώντας ένα σταθερό αριθμό, έστω k , online πειραμάτων. Μετά από αυτό το στάδιο, τα βάρη στις κατηγορίες του τοπικού γράφου G είναι σταθερά για το διάστημα μιας συνόδου συνδυάζοντας έτσι τη λογική από την offline έκδοση. Αυτή η προσέγγιση είναι απαιτεί λιγότερη υπολογιστική ισχύ από ό,τι η online χωρίς να χάνει το στοιχείο της άμεσης απεικόνισης των επιλογών μέσα στη ίδια τη σύνοδο.

Semi_Offline_Categories (G_{sub})

Έστω k μια μικρή προκαθορισμένη σταθερά, και G' ο γράφος που αντικαθιστά αυξητικά το G_{sub} .

```

 $G' = G_{sub}$  ;
for  $i=1$  to  $k$  do
    Experiment_ $i$  ;
    Update_Weights( $G'$ ) ;
return  $G'$  ;

```

Αλγόριθμος 16 : Αλγόριθμος Semi_Offline_Categories

4.9.1 Αρχικοποίηση

Ανεξάρτητα από το ποιο είδος κατηγοριοποίησης έχει επιλεγεί, πρέπει να οριστεί στο πρώτο βήμα το υποσύνολο του ODP όπου λειτουργούν οι αλγόριθμοι. Η διαδικασία μπορεί να περιγραφεί στα παρακάτω βήματα:

Βήμα 1. Στην περίπτωση που οι προτιμώμενες κατηγορίες χρήστη παρέχονται ήδη (από προηγούμενη καταγραφή δραστηριότητας περιήγησης) προχωράμε στο επόμενο βήμα. Διαφορετικά, πρέπει να προηγηθεί έμμεση παρακολούθηση των προηγούμενων δραστηριοτήτων περιήγησης στα αποτελέσματα του χρήστη.

Βήμα 2. Απομονώνουμε τον υπογράφο που προκύπτει G_{sub} των κατηγοριών ενδιαφέροντος του χρήστη από όλες τις ODP κατηγορίες.

Βήμα 3. Για κάθε ακμή $e(u,v)$, ορίζουμε $d(u,v) = 1/(\text{outdegree}(u))$.

Το Βήμα 3 υπολογίζει σημασιολογικούς συσχετισμούς κατηγοριών στον γράφο που προκύπτει. Παρόλα αυτά, οι υπολογισμοί που εκτελούνται πραγματικά στην υλοποίησή μας υιοθετούν την υπόθεση ότι όσο πιο γενική είναι η κατηγορία, επομένως έχει πολλά παιδιά, τόσο λιγότερο θα σχετίζεται με τα παιδιά της. Αυτή η επιλογή υπαγορεύθηκε από την απουσία επαρκών text περιγραφών κατηγορίας. Αν αυτό δεν ισχύει, μπορεί να διατηρηθεί ένας πίνακας όρων-κατηγορίας A . Μπορούμε να υπολογίσουμε τον πίνακα $C=A^T \cdot A$, το στοιχείο $C(i,j)$ του οποίου κρατάει το συσχετισμό των κατηγοριών i και j .

4.9.2 Υπολογισμός της Σπουδαιότητας Κατηγορίας

Ο αλγόριθμος 12 Experiment_ i απαιτεί από το χρήστη να αναθέσει μια τιμή σχετικότητας $a_i(p)$ σε κάθε μια από τις επιστρεφόμενες σελίδες p που αυτός θεωρεί σημαντικές. Η αρχική συνάρτηση κατάταξης ανεξάρτητα από τη σύνθεσή της δεν επηρεάζει σε μεγάλο βαθμό το τελικό αποτέλεσμα. Επομένως, επιλέγουμε να αναθέσουμε βάρη έμμεσα, αφήνοντας κάθε «ψήφο» (επιλογή από το χρήστη μιας σελίδας αποτελέσματος) για μια κατηγορία v , να συνεισφέρει στην ίδια βαθμολογία. Έτσι παίρνουμε:

$$\beta(v_i) = \frac{\# \text{σελίδων κατηγορίας } i}{\# \text{σελίδων που επισκέφθηκε ο χρήστης}}$$

4.10 Οδικός χάρτης πειραματικής διαδικασίας και αξιολόγησης

Υποβάλαμε τους αλγορίθμους που παρουσιάστηκαν σε μια σειρά πειραματικών αξιολογήσεων για να ελέγξουμε και να αξιολογήσουμε στη συνέχεια την αποδοτικότητα και αποτελεσματικότητά τους. Εκατόν ένα (101) εθελοντές σπουδαστές του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πάτρας (ΑΤΕΙ Πάτρας, Ελλάδα) συμμετείχαν στην πειραματική αξιολόγηση κατά τη διάρκεια του ακαδημαϊκού έτους του 2004-2005. Οι σπουδαστές είχαν μια μέση γνώση χρήσης του Διαδικτύου που τους επέτρεψε να χρησιμοποιήσουν τις μηχανές αναζήτησης με ευχέρεια. Πραγματοποιήσαμε τα πειράματα σε δύο φάσεις σύμφωνα με τους αλγορίθμους: Προηγήθηκε φάση αρχικοποίησης (έμμεση καταγραφή του ιστορικού περιήγησης χρηστών (ερωτήματα και αποτελέσματα μόνο) και συνεχίστηκε με τη φάση ταξινόμησης ερωτήματος (εκτελέσεις του προτεινόμενου αλγορίθμου). Όλοι οι σπουδαστές συμμετείχαν και στις δύο φάσεις.

Αρχικά, η φάση αρχικοποίησης περιλάμβανε τις ακόλουθες δράσεις:

α) Εκκίνηση της εξατομίκευσης βάζοντας τους σπουδαστές να υποβάλλουν ερωτήματα και να επιλέξουν τις απαντήσεις που προτιμούν ενώ οι επιλογές τους καταγράφονταν.

β) Στη συνέχεια, δημιουργήθηκε ένα εξατομικευμένο δέντρο κατηγορίας (γράφος G) κατάλληλο για τον υπολογισμό βαρών κατηγορίας και εισήχθησαν εικονικοί κόμβοι σύμφωνα με τους προτεινόμενους αλγορίθμους. Αυτή η φάση υλοποιεί τη semi-offline μεθοδολογία της παραγράφου 4.9. Τα αναφερόμενα “ k πειράματα” είναι τυχαία ερωτήματα-σύνοδοι αναζήτησης των συμμετεχόντων εθελοντών φοιτητών στην πρώτη φάση, διαφορετικά για κάθε χρήστη ($k \in [5,12]$).

Μετά την ολοκλήρωση της αρχικοποίησης, συνεχίσαμε τα πειράματα βάζοντας τους σπουδαστές να υποβάλλουν περισσότερα ερωτήματα με σκοπό να συγκρίνουμε την ταξινόμηση των αποτελεσμάτων που ανακτήθηκαν από τη μηχανή αναζήτησης με την εξατομικευμένη έκδοση της κατάταξης που υπολογίζεται από τους αλγορίθμους μας. Μετά την ολοκλήρωση των πράξεων της πρώτης φάσης, δεν έγιναν επιπλέον ενημερώσεις βάρους κατηγορίας και ένα δεύτερο διαφορετικό σύνολο ερωτημάτων υποβλήθηκε στη μηχανή αναζήτησης. Αυτό ήταν απαραίτητο μόνο για να πραγματοποιήσουμε μέτρηση της Ακρίβειας-Ανάκλησης [2] και δεν απαιτείται από τη λειτουργία της μεθόδου μας.

Τέλος, συγκρίναμε τους γράφους Ακρίβειας-Ανάκλησης (Precision-Recall) [2] μεταξύ των αρχικών βαθμολογήσεων που επιστράφηκαν από τη μηχανή αναζήτησης χωρίς εξατομίκευση και τη βαθμολόγηση μετά την εμπλοκή των αλγορίθμων εξατομίκευσης. Διαφορετικές υλοποιήσεις των προτεινόμενων αλγορίθμων δοκιμάστηκαν χρησιμοποιώντας εναλλακτικές πολιτικές δημιουργίας εικονικών

κόμβων είτε με απλούς (έναν) είτε με πολλαπλούς κόμβους. Λεπτομέρειες για κάθε πειραματικό βήμα συζητούνται στη συνέχεια.

4.11 Μεθοδολογία πειραμάτων

Σε αντίθεση με τη χρήση μιας συλλογής εγγράφων δοκιμής που μπορεί να γίνει σε συστήματα ανάκτησης πληροφορίας, στα δικτυακά πληροφοριακά συστήματα αλλά και στο Διαδίκτυο χρησιμοποιείται για απεικόνιση της πληροφορίας ο ιστός που αποτελεί «ζωντανά» “live” δεδομένα τα οποία αλλάζουν συνεχώς, εμποδίζοντας τα πειράματα να είναι μπορούν να επαναληφθούν ακριβώς τα ίδια από τη μια αναζήτηση σε μία άλλη και από τον ένα χρήστη σε έναν άλλο. Επιπλέον, πιστεύεται ότι ένα σύνολο από δημοφιλή ερωτήματα ιστού και τα επιθυμητά αποτελέσματά τους αλλάζουν σημαντικά με την πάροδο του χρόνου και ότι αυτές οι αλλαγές έχουν σημαντική επίδραση στην αξιολόγηση της μηχανής αναζήτησης. Οι Hawking, et al. επισημαίνουν ότι “Οι αποδόσεις των μηχανών αναζήτησης μπορεί να ποικίλουν σημαντικά πάνω από διαφορετικά σύνολα ερωτημάτων και κατά τη διάρκεια του χρόνου” [18].

Οποιαδήποτε μηχανή αναζήτησης θα μπορούσε να χρησιμοποιηθεί για τα πειράματα. Χωρίς να χάνεται η γενικότητα της μεθόδου που προτείνουμε [25], στην περίπτωση των πειραμάτων μας, χρησιμοποιήθηκαν αποτελέσματα από τη μηχανή αναζήτησης Google. Η επιλογή αυτή έγινε κυρίως λόγω της υποστήριξης που δίνει η μηχανή ως προς τις προγραμματιστικές διεπαφές. Τα πειράματα των χρηστών πραγματοποιήθηκαν σε εργαστηριακές αίθουσες πληροφορικής του ΑΤΕΙ Πατρών χρησιμοποιώντας επιτραπέζιους προσωπικούς υπολογιστές Pentium III 800 MHz με 256MB RAM για κάθε χρήστη.

Οι ιδιότητες του γράφου του ιστού, συνολικά, διαφέρουν από αυτές οποιουδήποτε δείκτη κατηγορίας ή δέντρου ταξινόμησης / οντολογίας. Ωστόσο, είναι αδύνατο να πάρουμε ένα σταθερό υποσύνολο του ιστού και να παράγουμε το γράφο κατηγορίας από αυτόν. Ως αποτέλεσμα, χρησιμοποιούνται μόνο κατηγορίες που υπάρχουν στο ευρύτερα χρησιμοποιούμενο κατάλογο του ODP. Άλλες σχέσεις τύπου ανήκει σε (IS-A), οντολογίες ή ταξινομίες σε συγκεκριμένες περιοχές εφαρμογής μπορούν επίσης να χρησιμοποιηθούν χωρίς να επηρεάσουν τη γενικότητα των αλγορίθμων που παρουσιάστηκαν. Ειδικότερα, οι πληροφορίες ιεραρχίας κατηγοριών ODP, ενημερωμένες με το ιστορικό προηγούμενων τροποποιήσεων των κατηγοριών που δίνεται στο ODP, αναλύθηκαν και μετατράπηκαν σε ένα σύστημα βάσης δεδομένων (Enterprise Edition Microsoft SQL server 2000 sp3). Στα πειράματα, αποτελέσματα χρησιμοποιήθηκαν μόνο όταν άνηκαν σε ODP κατηγορίες, περιλαμβάνοντας αποτελέσματα που προέρχονταν κυρίως από το τμήμα καταλόγου των μηχανών αναζήτησης (το Google χρησιμοποιήθηκε σε αυτήν την περίπτωση όπως έχουμε ήδη αναφέρει). Ωστόσο η γενικότητα δεν επηρεάζεται. Μπορεί κανείς να εκτελέσει τεχνικές κατηγοριοποίησης για να ταξινομήσει αυτόματα ένα αποτέλεσμα μηχανής αναζήτησης on-the-fly ώστε να ληφθούν υπόψη οι αταξινόμητες σελίδες στους προτεινόμενους αλγορίθμους (δείτε τα [8], [12], [15],

[14]). Στα πειράματα που πραγματοποιήθηκαν, οι προτεινόμενες τεχνικές εκμεταλλεύονται μόνο την έμμεση είσοδο του χρήστη – τα επιλεγμένα URLs. Επιπλέον, θα ήταν επίσης δυνατό να ληφθεί υπόψη η άμεση αξιολόγηση χρήστη των επιλεγμένων URLs. Ωστόσο μια τέτοια αξιολόγηση αυξάνει τον αριθμό των βημάτων και γενικά αποφεύγεται [33]. Παρόλο που η μελέτη των δεδομένων από άμεσα βήματα βαθμολόγησης (κλικ) από κάποιους θεωρείτε χρήσιμη υπάρχει μία καταγεγραμμένη αδυναμία, καθώς οι χρήστες είναι πιο πιθανό να κάνουν κλικ σε υψηλά καταταγμένα έγγραφα ανεξάρτητα από την ποιότητά τους [32].

Επιπλέον, όπως δηλώθηκε στην παράγραφο 4.5, όταν η v_j είναι μια υποκατηγορία της v_i τότε της ανατίθεται ένα βάρος $d(v_i, v_j) \in [0,1]$ που εκφράζει τη σημασιολογική συσχέτιση μεταξύ των κατηγοριών v_i, v_j . Εδώ η υπόθεση είναι ότι:

$$d(v_i, v_j) = 1 / |\text{απόγονοι της } v_i|$$

Ως τρόπος βελτίωσης της συνολικής απόδοσης κατά την εκτέλεση, η προεπεξεργασία του γράφου ODP εκτελέστηκε για να υπολογίσει τους απογόνους κάθε κατηγορίας εκ των προτέρων.

Οι χρήστες υπέβαλαν συνολικά 58 διαφορετικά ερωτήματα ιστού κατά τη φάση αρχικοποίησης, η οποία στο εξής δηλώνεται ως σύνολο ερωτημάτων Q_1 . Για τη δεύτερη φάση, η οποία ονομάζεται φάση ερωτήματος, οι χρήστες υπέβαλαν συνολικά 96 διαφορετικά ερωτήματα. Στο εξής, το δεύτερο σύνολο ερωτημάτων θα δηλώνεται ως σύνολο ερωτημάτων Q_2 . Κάθε χρήστης έπρεπε να υποβάλλει τουλάχιστον πέντε (5) ερωτήματα κατά τη διάρκεια της πρώτης φάσης και τουλάχιστον δέκα (10) ερωτήματα στη δεύτερη φάση. Κατά μέσον όρο, κάθε χρήστης υπέβαλλε 5,94 ερωτήματα κατά τη διάρκεια της φάσης ερωτήματος και 16,93 ερωτήματα κατά τη δεύτερη πειραματική φάση. Οι χρήστες ήταν ελεύθεροι να υποβάλλουν ερωτήματα σύμφωνα με τις προσωπικές τους προτιμήσεις και ενδιαφέροντα και στις δύο περιπτώσεις. Προκειμένου να καταγραφεί η δραστηριότητα των χρηστών, με τη συγκατάθεση τους κατά τη διάρκεια του πειράματος, μια διαφανής στο χρήστη δικτυακή εφαρμογή κατέγραφε τα ερωτήματα του χρήστη και τις επιλογές των αποτελεσμάτων αναζήτησης σε μια βάση δεδομένων. Τα ερωτήματα που υποβλήθηκαν ήταν σχετικά σύντομα καθώς τα ερωτήματα ιστού έχουν συνήθως μήκος κατά προσέγγιση 2,3 λέξεων. Αυτό είναι σύμφωνο με μετρήσεις που έχουν γίνει σε καταγραφές χρήσης των μηχανών αναζήτησης (search engine query logs) όπου το μέσο ερώτημα έχει μήκος 2,21 όρους [32]. Οι σπουδαστές υπέβαλλαν διαφορετικά ερωτήματα στις δύο φάσεις. Η παρακολούθηση επίσης έδειξε ότι υπήρχε επικάλυψη στα ερωτήματα που υποβλήθηκαν από τους σπουδαστές, το οποίο είναι φυσικό καθώς οι σπουδαστές μπορεί να έχουν παρόμοια ενδιαφέροντα/ερωτήματα κατά νου. Κάτι τέτοιο όμως δεν έχει επιπτώσεις στον πειραματισμό των αλγορίθμων καθώς ξεχωριστοί εξατομικευμένοι γράφοι κατηγοριών δημιουργήθηκαν για κάθε χρήστη.

4.11.1 Φάση Αρχικοποίησης

Η ιδέα στην παρούσα φάση είναι να απεικονιστεί ένα αρχικό ιστορικό επιλογών των χρηστών ως ζυγισμένο αποτέλεσμα στις κατηγορίες προτίμησης. Εν συντομία, αυτό επιτυγχάνεται πραγματοποιώντας μια σειρά ερωτημάτων και καταγράφοντας έμμεσα τη δραστηριότητα περιήγησης των χρηστών. Λαμβάνοντας υπόψη τις επιλογές χρηστών, ο αλγόριθμός μας αρχικοποιεί και δημιουργεί το εξατομικευμένο δέντρο κατηγορίας κάθε χρήστη. Η προσέγγισή μας ακολουθεί την υπόθεση ότι στη λίστα των αποτελεσμάτων που επιλέγονται από το χρήστη (Relative Set), συγκεκριμένες σελίδες έχουν επισκεφτεί οι οποίες ανήκουν στις κατηγορίες και επομένως οι κατηγορίες εμφανίζονται με μια συγκεκριμένη συχνότητα, η οποία είναι η έμμεση βαθμολογία που αναθέτει ο χρήστης στην κατηγορία.

Στη φάση αρχικοποίησης τέθηκαν διαφορετικά ερωτήματα (του συνόλου Q_1) από τους χρήστες και καταγράψαμε:

Σχήμα 26: Εισαγωγή εικονικών κόμβων.

- Το σύνολο R των σχετικών σελίδων κάθε ερωτήματος (Relative Set, οι σελίδες που επισκέφθηκαν οι χρήστες)
- Το PageRank κάθε σελίδας που επιστράφηκε ως αποτέλεσμα (μετασχηματισμένο στο πεδίο $[0,1]$)

Στο σημείο αυτό θα διευκρινίσουμε τι συμβαίνει μεταξύ δύο διαδοχικών εκτελέσεων της $Experiment_i$ (Αλγόριθμος 12 - υποβολή ερωτήματος).

Στην j -οστή εκτέλεση της $Experiment_i$, υπολογίστηκε ένα βάρος $\beta_j(v)$ για την κατηγορία v . Σημειώνουμε ότι προηγούμενα βάρη $\beta_i(v)$ μπορεί επίσης να υπάρχουν για το v για ορισμένες τιμές $i=1, \dots, j-1$. Σε αυτήν την περίπτωση το τελικό βάρος σπουδαιότητας $\bar{\beta}_j(v)$ αυτής της κατηγορίας υπολογίζεται ως εξής:

$$\bar{\beta}_j(v) = \begin{cases} \beta_j(v) & , j=1 \\ \frac{\beta_j(v) + \sum_{i=1}^{j-1} \bar{\beta}_i(v)}{k+1} & , j>1 \end{cases}$$

όπου το k δηλώνει τον αριθμό των φορών που επιλέχθηκαν αποτελέσματα από την κατηγορία v στα πειράματα $1, \dots, j-1$ (το k μπορεί να είναι $j-1$). Με απλά λόγια, εξάγαμε τους μέσους όρους όλων των διαθέσιμων τιμών του $\beta(v)$ τη στιγμή της j -οστής επανάληψης.

Στη συνέχεια υπάρχει δυνατότητα προσθήκης των νέων εικονικών κόμβων που εισάγουμε [25] πάνω από διαδοχικές εκτελέσεις της Experiment_i. Όλες οι πιθανές διαμορφώσεις παρουσιάζονται στο σχήμα 26. Υποθέτουμε ότι στο βήμα 3 είχαμε επιλέξει τρεις σελίδες με $\beta(v) > thresh1$, και ένας εικονικός κόμβος (Virtual Category – VU) δημιουργείται ('Virtual1' στο σχήμα 26α).

Στην περίπτωση που οι ίδιες σελίδες εμφανίζονται πάλι σε κάποιο μεταγενέστερο βήμα, απλά αγνοείται η δημιουργία του εικονικού κόμβου εφόσον υπάρχει ήδη. Εάν ένα υποσύνολό τους (αντίστοιχα ισχύει και για υπερσύνολο) εμφανίζεται τότε δημιουργείται ένας νέος κόμβος (τα παραδείγματα εμφανίζονται στα σχήμα 26β και σχήμα 26γ), αντίστοιχα. Σύμφωνα με τον κανόνα χειρισμού εξαιρέσεων της παραγράφου 4.6 (Αλγόριθμος 12-Experiment_i), οι εικονικοί κόμβοι δημιουργούνται μόνο όταν καμία υπάρχουσα κατηγορία ODP ή προϋπάρχων εικονικός κόμβος δε διασυνδέει τις κατηγορίες που εμπλέκονται. Σε αυτήν την εξαιρετική περίπτωση η εντολή δημιουργίας του εικονικού κόμβου VU απλά παρακάμπτεται.

4.11.2 Φάση Ερωτήματος

Σε αυτή τη φάση, η πειραματική αξιολόγηση έφτασε στο επόμενο στάδιο όπου πλέον δεν εκτελέστηκαν ενημερώσεις κατηγοριών ή δημιουργία εικονικών κόμβων στους εξατομικευμένους γράφους κατηγορίας χρήστη (προφίλ χρήστη). Οι μόνες λειτουργίες που εκτελέστηκαν ήταν η υποβολή νέων ερωτημάτων και η παρουσίαση των απαντήσεων αποτελέσματος με τον εξατομικευμένο τρόπο του προτεινόμενου αλγορίθμου. Ένα νέο και διακριτό σύνολο ερωτημάτων Q_2 υποβλήθηκε στη μηχανή αναζήτησης. Έχοντας κατά νου την περιγραφή της παραγράφου 4.5, θυμηθείτε ότι το σύστημα ανάκτησης χρησιμοποιεί μια συνάρτηση βαθμολόγησης $\phi(\beta(\gamma(p)), \sigma(p))$, η οποία είναι συνδυασμός μιας συνάρτησης βαθμολόγησης $\sigma(p)$ της γενικής μηχανής αναζήτησης και της σπουδαιότητας $\beta(\gamma(p))$ της κατηγορίας σελίδας που ανήκει το p . Για τους σκοπούς του πειράματος ορίζουμε την ακόλουθη εναλλακτική κατάταξη που χρησιμοποιεί τα αποτελέσματα του εξατομικευμένου γράφου κατηγορίας:

$$\phi(\beta(\gamma(p)), \sigma(p)) = (\beta(\gamma(p)) + \sigma(p)) / 2,$$

όπου $\sigma(p)$ είναι το PageRank του p ($\in [0,1]$), το p ανήκει στην κατηγορία v και ισχύει $v \in G$.

Μετά από κάθε ερώτημα τα αποτελέσματα βαθμολογήθηκαν σύμφωνα με τη $\phi()$. Για να αξιολογηθούν τα αποτελέσματα και να υπολογιστεί η ακρίβεια σε προκαθορισμένα επίπεδα ανάκλησης εφαρμόσαμε τη μέθοδο της ανάδρασης σχετικότητας (relevance feedback) [31], στην οποία ζητήθηκε από τους χρήστες αφού εξέτασαν τις απαντήσεις που ανακτήθηκαν να σημειώσουν ποιες από αυτές ήταν πιο σχετικές. Η λίστα των σχετικών αποτελεσμάτων για κάθε ερώτημα επιλέχθηκε χρησιμοποιώντας ως οδηγό τη συμπεριφορά του χρήστη όπως καταγράφηκε έμμεσα κατά τη διάρκεια της φάσης αρχικοποίησης. Συνολικά, ο βασικός στόχος αυτής της φάσης είναι η παρακολούθηση της ανάδρασης σχετικότητας του χρήστη η οποία θα βοηθούσε στην αξιολόγηση της απόδοσης του συστήματος.

4.12 Αξιολόγηση των αποτελεσμάτων

4.12.1 Γράφοι Ακρίβειας-Ανάκλησης

Τα πειραματικά αποτελέσματα απεικονίζονται χρησιμοποιώντας την προσέγγιση της Ακρίβειας και Ανάκλησης [1]. Θεωρούμε ένα αίτημα ερωτήματος Q_i και το σύνολο αποτελεσμάτων R σχετικών εγγράφων (οι απαντήσεις ερωτημάτων που επιλέγονται από τους σπουδαστές). Τότε το $|R|$ δηλώνει τον αριθμό των URLs – εγγράφων στο σύνολο. Αν υποθέσουμε ότι μια δεδομένη στρατηγική επιστρέφει ένα σύνολο εγγράφων απάντησης A , τότε $|A|$ είναι ο αριθμός αυτών των URLs-εγγράφων (τα αποτελέσματα της μηχανής αναζήτησης Google). Τα μέτρα ανάκλησης και ακρίβειας ορίζονται ως εξής:

$$Recall = \frac{|R \cap A|}{|R|}, Precision = \frac{|R \cap A|}{|A|}$$

Κατά τη διάρκεια των πειραμάτων, οι απαντήσεις-αποτελέσματα ήταν το πολύ εκατό (100) για κάθε ερώτημα. Στην πραγματικότητα, υπάρχει μια γενική αίσθηση ότι η πλειονότητα των χρηστών θα έψαχνε τον ιστό μόνο για ένα μικρό αριθμό υψηλά συσχετισμένων σελίδων και ότι θα περιοριζόταν να κοιτάζει μόνο ένα μέρος των αποτελεσμάτων. Η αξιολόγηση απαιτεί ακρίβεια χάραξης των καμπύλων ακρίβειας και ανάκλησης. Στα πειράματα, εξετάσαμε την ακρίβεια με ένα βήμα ανάκλησης 10% με στόχο να πάρουμε τους γράφους ακρίβειας και ανάκλησης.

Επιπλέον, για να αξιολογήσουμε την απόδοση του αλγορίθμου στα υποβεβλημένα ερωτήματα, υπολογίσαμε τους μέσους όρους των τιμών ακρίβειας σε κάθε επίπεδο ανάκλησης ως εξής:

$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

όπου $\bar{P}(r)$ είναι η μέση ακρίβεια στο επίπεδο ανάκλησης r , N_q είναι ο αριθμός των ερωτημάτων που χρησιμοποιούνται, και $P_i(r)$ είναι η ακρίβεια στο επίπεδο ανάκλησης r για το i -οστό ερώτημα.

Οι καμπύλες Ακρίβειας-Ανάκλησης (Precision-Recall) έχουν υπολογιστεί για κάθε βασισμένο σε ερώτημα υπο-πείραμα. Οι καμπύλες μέσης ακρίβειας και ανάκλησης παρουσιάζονται στο σχήμα 27 και το σχήμα 28 για τη φάση ερωτήματος, όπου εξατομικεύσαμε την τελική κατάταξη των αποτελεσμάτων των ερωτημάτων με βάση τη δραστηριότητα περιήγησης σε προηγούμενα αποτελέσματα, χωρίς καμία ιδιαίτερη προσπάθεια από την πλευρά του χρήστη, αντίθετα με την αρχική κατάταξη της μηχανής αναζήτησης. Οι χρήστες περιηγήθηκαν-επέλεξαν τα πιο κατάλληλα αποτελέσματα που επέστρεψαν από αυτήν την άποψη ανάδραση σχετικότητας.

Σχήμα 27: Αποτελέσματα P-R των Αρχικών (Κατάταξη Google) και Τελικών (Εξατομικευμένη Κατάταξη) Αποτελεσμάτων.

Η πρώτη καμπύλη ονομάζεται “Αρχική” και έχει τη σκούρα χρωματισμένη γραμμή στο κάτω μέρος του σχήμα 28. Προκύπτει από τις αρχικές επιλογές με βάση το χρήστη στη λίστα κατάταξης των URL που ανακτήθηκαν από το ερώτημα (ή το μέσο όλων των ερωτημάτων στο σύνολο). Η άλλη καμπύλη ονομάζεται “Τελική” και απεικονίζει την καμπύλη P-R που προκύπτει από τη λίστα αποτελεσμάτων της στρατηγικής κατάταξης που προτείνουμε.

Η μέση P-R προσέγγιση του συνόλου ερωτημάτων υποδεικνύει ότι η στρατηγική μας εξατομικεύει αποτελεσματικά την εμπειρία αναζήτησης του χρήστη. Τα επίπεδα ακρίβειας αυξήθηκαν σημαντικά μεταξύ του πρώτου 20% της ανάκλησης, ενώ κατά μέσο όρο ο προτεινόμενος αλγόριθμος εξατομίκευσης αυξάνει την ακρίβεια έως 21,48% των ανακτημένων εγγράφων κατά τη διάρκεια των πειραμάτων. Αυτό δείχνει ότι η προσέγγισή μας παρέχει αποτελεσματικά εξατομίκευση.

Σχήμα 28: Αποτελέσματα P-R των Αρχικών (Κατάταξη Google) και Τελικών (Εξατομικευμένη Κατάταξη) Αποτελεσμάτων.

Σχήμα 29: Η αύξηση του επιπέδου ακρίβειας αυξάνεται ιδιαίτερα στο 20% της Ανάκλησης

4.12.2 Η επίδραση της εισαγωγής εικονικών κόμβων

Προκειμένου να ποσοτικοποιηθεί η επίδραση των εικονικών κόμβων που εισάγονται στον εξατομικευμένο υπο-γράφο του χρήστη (που προκύπτει από την τιμή του *thres1*), εκτελέσαμε επίσης μια εναλλακτική προσέγγιση αξιολόγησης. Κατά τη διάρκεια των πειραμάτων, εισάγονται πολλαπλοί εικονικοί κόμβοι από τον semi-offline αλγόριθμο, ένας ανά σύνοδο ερωτήματος “experiment_i” για κάθε χρήστη. Για να υπολογίσουμε την επίδραση των εικονικών κόμβων στον αλγόριθμο, υπολογίσαμε την ακρίβεια-ανάκληση σύμφωνα με μια εναλλακτική του semi-offline αλγορίθμου, όπου δημιουργείται μόνο ένας εικονικός κόμβος στο τέλος για όλες τις συνόδους ερωτήματος “Experiment_i”, ελαχιστοποιώντας κατ' αυτό τον τρόπο τον αριθμό των εικονικών κόμβων που εμφανίζονται.

Σχήμα 30: Το απόλυτο κέρδος στην ακρίβεια για πολιτικές εισαγωγή Ενός και Πολλαπλών Εικονικών κόμβων στον semi-offline αλγόριθμο

Τα αποτελέσματα δείχνουν ότι ο προτεινόμενος semi-offline αλγόριθμος συμπεριφέρεται καλύτερα από την εναλλακτική λύση όπου ένας μόνο εικονικός κόμβος συνενώνει όλες οι προτιμήσεις ανά χρήστη. Το αναλυτικό κέρδος στην επίδραση εξατομίκευσης απεικονίζεται στο σχήμα 30. Βλέπουμε στο σχήμα 31 ότι στο πρώτο 20% της ανάκλησης με τη χρήση VU η ακρίβεια αυξάνεται κατά 15% σε σχέση με την εναλλακτική ενώ συνολικά έχουμε κατά μέσο όρο αύξηση 33,5% πράγμα που σημαίνει ότι τα αποτελέσματα επηρεάζονται θετικά από τη χρήση των εικονικών κόμβων.

Γενικά θεωρείται ότι οι μηχανές αναζήτησης είναι βελτιστοποιημένες ώστε να ανακτούν υψηλά συσχετισμένα έγγραφα με υψηλή ακρίβεια σε χαμηλά επίπεδα ανάκλησης, χαρακτηριστικά που είναι επιθυμητά για την υποστήριξη συχνών ερωτήσεων αναζήτησης. Σε αυτές τις περιπτώσεις, η διαφορά στα επίπεδα ακρίβειας μπορεί να αναμένεται να είναι μικρότερη. Ωστόσο οι εικονικοί κόμβοι που εισάγονται διευκολύνουν τη βελτιωμένη αναδιάταξη των αποτελεσμάτων χρησιμοποιώντας τις γειτνιάζουσες σχετικές κατηγορίες, καθώς αυτοματοποιούν περεταίρω την εξατομίκευση των γραφών κατηγορίας και συνδέουν σημασιολογικά και σύμφωνα με τις επιλογές του χρήστη ακόμη και ασυσχέτιστες μεταξύ τους κατηγορίες του γενικού γραφού ODP.

4.13 Μια επέκταση: στοχευμένη συλλογή σελίδων

Η τεχνική μας μπορεί επίσης να εφαρμοστεί με σκοπό να υλοποιήσει μηχανισμούς στοχευμένης συλλογής σελίδων (focused web crawlers)[8]. Οι μηχανισμοί στοχευμένης συλλογής σελίδων προτείνονται ως λύση στην αδυναμία να εκτελέσουμε αποτελεσματικά εξαντλητική συλλογή σελίδων λόγω του αυξανόμενου μεγέθους και περιεχομένου του ιστού [20]. Οι μηχανισμοί στοχευμένης συλλογής σελίδων αναζητούν μόνο ένα υποσύνολο του ιστού που σχετίζεται με συγκεκριμένα θέματα και παρέχουν μια πιθανή λύση στο πρόβλημα της κλιμάκωσης.

Σχήμα 31: Το σχετικό κέρδος στην ακρίβεια για πολιτικές εισαγωγή Ενός και Πολλαπλών Εικονικών κόμβων στον semi-offline αλγόριθμο

Το πρόβλημα της στοχευμένης συλλογής σελίδων καθορίζεται ως εξής:

Έστω ότι μας δίνεται ένα σύνολο υποδειγματικών εγγράφων που καθορίζουν έμμεσα τα θέματα για τα οποία ενδιαφέρεται ο χρήστης και καλούμαστε να παρέχουμε μια στρατηγική συλλογής σελίδων η οποία θα ευνοεί τις σελίδες που σχετίζονται θεματικά με τα έμμεσα καθορισμένα θέματα. Η επιθυμητή στρατηγική συλλογής σελίδων πρέπει να αποτελείται από δύο μέρη: έναν *ταξινομητή* (*classifier*) ο οποίος ταξινομεί τις εξερχόμενες συνδέσεις ιστοσελίδας σύμφωνα με τη σχετικότητα τους με το σύνολο θεμάτων, και ένας διαχωριστής (*distiller*) που εντοπίζει τα καλά σημεία πρόσβασης για την έναρξη της συλλογής σελίδων [8].

Προσαρμόζουμε τους αλγορίθμους μας [25] που έχουμε ήδη παρουσιάσει στις προηγούμενες ενότητες για να σχεδιαστεί το λειτουργικό υποσύστημα του ταξινομητή ως εξής:

Έστω ότι G είναι ο γράφος που χρησιμοποιείται για να κατηγοριοποιήσει τις ιστοσελίδες και S το σύνολο των υποδειγματικών εγγράφων που δίνονται. Το λειτουργικό υποσύστημα ταξινομητή αρχικά θα εκτελέσει τα ακόλουθα βήματα προ-επεξεργασίας:

Focused_Crawler_Init

Βήμα 1. Στο σύνολο S εφαρμόζουμε έναν αλγόριθμο μετρικής συνδέσμων ([5] ή [23]) και υπολογίζει για κάθε σελίδα $p \in S$ το αντίστοιχο βάρος της $w(p)$.

Βήμα 2. Για κάθε κατηγορία v του G , υπολογίζει το

$$w(v) = \frac{1}{n(v)} \sum_{j=1}^{n(v)} w(p_j^v),$$

όπου $p_1^v, \dots, p_{n(v)}^v$ είναι οι σελίδες του S που ανήκουν στην κατηγορία v και $n(v) = |S|$.

Βήμα 3. Update_Weights(G);

Αλγόριθμος 17: Προ-επεξεργασία στοχευμένης συλλογής σελίδων

Αφού εκτελεστεί ο αλγόριθμος 17, κάθε κατηγορία στο G θα έχει ένα βάρος που καθορίζει την σχετικότητα των θεμάτων που αναπαρίστανται από την κατηγορία με τα θέματα που δόθηκαν στα υποδειγματικά έγγραφα. Συνεπώς, ο μηχανισμός συλλογής σελίδων μπορεί να ταξινομήσει τους συνδέσμους εξόδους (outgoing link) μιας σελίδας χρησιμοποιώντας απλά ως μέτρο σχετικότητας το βάρος της κατηγορίας της ιστοσελίδας όπου δείχνει ο σύνδεσμος. Ως σύνδεσμοι εξόδου θεωρούνται οι σύνδεσμοι εκείνοι που οδηγούν σε σελίδα διαφορετικής περιοχής (domain).

Μια αδυναμία της παραπάνω προσέγγισης προέρχεται από το γεγονός ότι είναι δυνατό μια σελίδα να μην ανήκει σε μια κατηγορία. Σε αυτήν την περίπτωση μπορούμε είτε να χρησιμοποιήσουμε μια προσέγγιση, όπως αυτή που περιγράφεται στο [8] για να εντοπίσει μία ή περισσότερες κατηγορίες που ταιριάζουν καλύτερα με το θέμα της σελίδας ή μπορούμε να εξετάσουμε μια διαφορετική προσέγγιση προκειμένου να υπολογιστεί η σχετικότητα ενός εξερχόμενου συνδέσμου. Σε αυτήν την προσέγγιση (δείτε επίσης [28]) χρησιμοποιούμε τους συνδυασμούς των ταξονομιών και των θησαυρών που αφορούν τους όρους που περιλαμβάνονται στις ιστοσελίδες (ένα παράδειγμα ενός τέτοιου θησαυρού είναι το Wordnet) και συνεχίζουμε ως εξής:

Έστω ότι O είναι η ταξονομία που σχετίζεται με διάφορους όρους και έστω ότι το S δηλώνει και πάλι το σύνολο των υποδειγματικών εγγράφων. Σε μια αρχική φάση του υποσυστήματος ταξινόμησης θα εκτελέσουμε τα παρακάτω βήματα:

Focused_Crawler_Process

Βήμα 4. Υπολογίζουμε για κάθε όρο v και έγγραφο p στην συλλογή, ένα βάρος $w(p^v)$ που δηλώνει τη σημασία του όρου v για την περιγραφή του περιεχομένου του p (μια βολική επιλογή για αυτό θα ήταν η μετρική $tf-idf$, δείτε [1]).

Βήμα 5. Για κάθε όρο v του O , υπολογίζουμε:

$$w(v) = \frac{1}{n(v)} \sum_{j=1}^{n(v)} w(p_j^v),$$

όπου $p_1^v, \dots, p_{n(v)}^v$ είναι οι σελίδες του S που περιέχουν τον όρο v και $n(v) = |S|$.

Βήμα 6. $Update_Weights(O)$;

Αλγόριθμος 18: Ανάθεση βάρους για ταξινομητή στοχευμένης συλλογής σελίδων

Αφού εκτελεστούν τα βήματα που δίνει ο αλγόριθμος 18, κάθε όρος στην ταξονομία θα έχει ένα βάρος που δηλώνει την ομοιότητα του με τα θέματα που αναπαρίστανται από τα υποδειγματικά έγγραφα. Τότε, για ένα δεδομένο σύνδεσμο εξόδου η μετρική σχετικότητας θα είναι ίση με το ζυγισμένο άθροισμα της σχετικότητας των όρων που περιέχονται στο κείμενο του συνδέσμου (anchor text) (ή σε ένα παράθυρο ορισμένων χαρακτήρων γύρω από το anchor text όπως γίνεται στο [8]).

4.14 Προσαρμοστική αναζήτηση με χρήση SVM κατηγοριοποιητών

Στη συνέχεια παρουσιάζουμε ένα μηχανισμό σύμφωνα με τις εργασίες μας [14], [15] ο οποίος συνεισφέρει στην προσαρμοστική και εξατομικευμένη αναζήτηση πραγματοποιώντας κατηγοριοποίηση στα αποτελέσματα και παρουσιάζοντάς τα ομαδοποιημένα.

Ο μηχανισμός παρέχει πολλαπλές δυνατότητες:

1. Είναι δυνατό να χρησιμοποιηθεί από τους αλγορίθμους εξατομικευσης online, offline και semi-offline των προηγούμενων ενοτήτων, στις περιπτώσεις όπου πρέπει να ληφθούν υπόψη, αποτελέσματα που δεν έχουν κατηγοριοποιηθεί εξ αρχής από τη μηχανή αναζήτησης.
2. Επιπλέον μπορεί να χρησιμοποιηθεί ως αυτόνομος μηχανισμός προσαρμοστικής αναζήτησης. Στην περίπτωση αυτή, οι χρήστες είναι δυνατό να επιλέξουν τις κατηγορίες εκείνες στις οποίες επιθυμούν να ομαδοποιήσουν τα αποτελέσματα. Επιπλέον, συντηρούμε ένα ιστορικό προφίλ για τους χρήστες με τις παλαιότερες κατηγορίες αποτελεσμάτων που έχουν επιλέξει προκειμένου να ανιχνεύονται σωτηρηρά οι προτιμήσεις τους.

Για να μεγιστοποιήσουμε το αποτέλεσμα τις εξατομικευσης (personalization, adaptive web) [6] ως προφίλ έχουμε χρησιμοποιήσει τη λύση προφίλ πολλαπλών επιπέδων που έχουμε εισάγει στην εργασία [16].

4.14.1 Προφίλ πολλαπλών επιπέδων

Προτού αναλύσουμε περεταίρω την προσέγγιση προσαρμοστικής αναζήτησης με χρήση κατηγοριοποιητή θα δώσουμε λεπτομέρειες αναφορικά με το προφίλ πολλαπλών επιπέδων που έχουμε εισάγει με την εργασία [16] για την περίπτωση χρηστών εικονικών πανεπιστημίων (virtual university).

Ένα προφίλ καταγράφει πληροφορίες αναφορικά με τις ενέργειες και την κατάσταση γνώσης ενός χρήστη [3]. Αυτές οι πληροφορίες είναι ζωτικής σημασίας για την εξατομικευμένη λειτουργία ενός συστήματος. Η έμμεση συλλογή των ενεργειών χρησιμοποιείται συχνότερα για την καταγραφή προφίλ, δεδομένου ότι απαιτεί την ελάχιστη συμμετοχή στη διαδικασία υλοποίησής του από τους ίδιους τους χρήστες. Εντούτοις, τα στοιχεία αυτά είναι κάποιες φορές δύσκολο να συλλεχθούν μιας και δεν είναι πάντοτε εύκολο να αντιπροσωπευθούν οι δυνατότητες του χρήστη. Επιπλέον, η φύση του Ιστού επιβάλλει ορισμένους περιορισμούς ως προς τις λεπτομέρειες και την ακριβή αντίληψη που μπορεί να καταγραφεί από οποιοδήποτε δικτυακό περιβάλλον για ένα χρήστη. Το περιβάλλον που παρουσιάζεται στο παρόν κεφάλαιο χρειάζεται να καταγράφει μόνο την αναλυτική πορεία επιλογών αποτελεσμάτων αποθήκευσης και τις λέξεις κλειδιά των αναζητήσεων που πραγματοποιήθηκαν ανά σύνοδο.

Σχήμα 32: Πολυεπίπεδο πλαίσιο διαχείρισης προφίλ

Στο προφίλ που προτείναμε υπάρχουν πολλαπλά επίπεδα για την καταγραφή κάθε δραστηριότητας και ένα κοινό διαχειριστικό επίπεδο που διασυνδέει τα επίπεδα αυτά. Στη περίπτωση του εικονικού πανεπιστημίου το προφίλ ήταν τόσο ένα εργαλείο απεικόνισης του μοντέλου μάθησης κατά την εκπαιδευτική διαδικασία όσο και η καταγραφή μιας σειράς από επιπλέον δραστηριοτήτων του χρήστη σε επιπλέον υπηρεσίες επικοινωνίας και συνεργασίας. Για την εξατομικευμένη αναζήτηση χρησιμοποιήσαμε τις επιπλέον αυτές υπηρεσίες προσαρμόζοντάς τις κατάλληλα. Για παράδειγμα τα καταγραφόμενα ερωτήματα ανά σύνοδο καταγράφονται σε διαφορετικό επίπεδο του προφίλ από τις επιλογές κατηγοριών που έγιναν αλλά και τους τίτλους των σελίδων που επισκέφτηκε ο χρήστης κατά την αναζήτησή του.

Στη συνέχεια πραγματοποιείται μια συνεργατική συσχέτιση ανάμεσα σε αναζητήσεις με τη χρήση ενός επιπέδου δικτυακής διασύνδεσης μεταξύ προφίλ διαφορετικών χρηστών. Στη διαδικασία αυτή κοινές περιγραφές (σύμφωνα με τις τεχνικές από το WordNet [34], [35]) ίδιων επιπέδων προφίλ (διαφορετική δραστηριότητα) μεταξύ διαφορετικών χρηστών θεωρούνται διασυνδεδεμένες και επομένως οι αντίστοιχες επιλογές σελίδων κοινού ενδιαφέροντος. Εφόσον οι διασυνδεδεμένες επιλογές είναι σημαντικές σε αριθμό (περισσότερες από 10) τότε η κοινή γνώση μεταφέρεται στο τοπικό επίπεδο διασύνδεσης κάθε εμπλεκόμενου χρήστη. Με τον τρόπο αυτό είναι δυνατό να βελτιωθούν τα επίπεδα τόσο της κατηγοριοποίησης όσο και της τελικής εξατομικεύσης με τη χρήση των εικονικών κόμβων.

Επιπρόσθετα στο προφίλ που προτείναμε [16] υπάρχει πρόβλεψη ώστε να αντιμετωπίζονται καταγραφές και έννοιες που απαρχαιώνονται με την πάροδο του χρόνου. Όσα χαρακτηριστικά που καταγράφηκαν δεν χρησιμοποιηθούν για

καθορισμένο μέγιστο χρονικό διάστημα απενεργοποιούνται με την ένδειξη απαρχαιωμένα υποδεικνύοντας ότι μάλλον ο χρήστης δεν ενδιαφέρεται πλέον για την έννοια αυτή. Ωστόσο οι εγγραφές αυτές δεν αφαιρούνται από το προφίλ τελείως για τυχόν μελλοντική χρήση πριν από ένα ευρύτερο καθοριζόμενο διάστημα. Στην περίπτωση μας καθορίσαμε τα δύο διαστήματα σε ένα ημερολογιακό μήνα. Η διαδικασία της απαρχαίωσης μπορεί να μετατραπεί εύκολα ώστε να λειτουργεί με βάση την υπόθεση ότι μία έννοια που ακολουθεί την ομοιόμορφη κατανομή χρονικά είναι περισσότερο χρήσιμη από άλλες [1]. Αν και η συγκεκριμένη υπόθεση δεν έχει ούτε επιβεβαιωθεί ούτε και απορριφθεί, παραμένει το γεγονός ότι η επιρροή του χρόνου πάνω στις έννοιες του προφίλ είναι σημαντικής σημασίας.

4.14.2 Μηχανισμός προσαρμοστικής αναζήτησης

Μετά από την ολοκλήρωση των δραστηριοτήτων της μηχανής αναζήτησης, ένας κατηγοριοποιητής υψηλής απόδοσης [13], [30] χρησιμοποιείται για την τελική κατηγοριοποίηση –ταξινόμηση – και ομαδοποίηση των αποτελεσμάτων στις επιθυμητές θεματικές κατηγορίες. Στη δική μας περίπτωση το λειτουργικό υποσύστημα της κατηγοριοποίησης είναι διαθέσιμο πολλαπλώς. Κύρια συνεισφορά μας στο μηχανισμό που εισάγουμε είναι η δυνατότητα κάθε χρήστη να εκπαιδεύσει επιπρόσθετα την προσωπική του έκδοση κατηγοριοποιητή πέρα από τη χρήση της κοινής έκδοσης [15], [14]. Με τον τρόπο αυτό είναι δυνατό να εκπαιδεύσει τον κατηγοριοποιητή στοχοθετημένα και σύμφωνα με τα ιδιαίτερά του ενδιαφέροντα.

Γενικά για να αυξηθεί η ακρίβεια του μηχανισμού όπως και σε κάθε κατηγοριοποιητή, είναι ουσιαστικό να εκπαιδευτεί το SVM με χαρακτηριστικές σελίδες κάθε θεματικής περιοχής που ενδιαφέρει. Στην περίπτωση μας χρησιμοποιήσαμε την ιεραρχία κατηγοριών του ODP. Πέρα από τις κατηγορίες το ODP συνοδεύεται από μια σειρά σελίδων ανά κατηγορία που μπορούν να χρησιμοποιηθούν για την εκπαίδευση του SVM. Η κεντρική λογική του SVM είναι να ταξινομεί ένα νέο αποτέλεσμα σε μία κατηγορία σύμφωνα με τα προηγούμενα κατηγοριοποιημένα δείγματα.

Η εκπαίδευση του κατηγοριοποιητή είναι κρίσιμη για την αποδοτική λειτουργία του μηχανισμού. Οι χρήστες μπορούν να χρησιμοποιούν αρχικά ένα κοινό SVM κατηγοριοποιητή ο οποίος έχει εκπαιδευτεί από την πρώτη εγκατάσταση του μηχανισμού. Εμείς εισάγουμε ένα επιπλέον επίπεδο εξειδίκευσης έτσι ώστε να είναι δυνατό για τους χρήστες να παρέχουν πρόσθετη κατάρτιση μιας προσωπικής έκδοσης κατηγοριοποιητή που λειτουργεί επαυξητικά με τον κοινό. Κάθε χρήστης μπορεί να παρέχει στον προσωπικό του κατηγοριοποιητή έγγραφα που αντιπροσωπεύουν τις προσωπικές του θεματικές κατηγορίες ώστε να εκτελεσθεί πρόσθετη εξειδίκευση στοχευμένα.

Κατά συνέπεια η χρήση των κατάλληλων δειγμάτων είναι ένα πολύ κρίσιμο βήμα στο στάδιο της εκμάθησης του SVM. Αυτά τα δείγματα συμμετέχουν ουσιαστικά στις επόμενες διαδικασίες κατηγοριοποίησης νέων εγγράφων – αποτελεσμάτων αναζήτησης και επομένως θα πρέπει να επιλέγονται με προσοχή από τους χρήστες.

Σχήμα 33: Αποτελέσματα κατηγοριοποίησης με βάση πλήρες κείμενο και σύνοψη

Για ακόμη ακριβέστερα αποτελέσματα, ο μηχανισμός μας διευκολύνει τους χρήστες κατά την εκπαίδευση του κατηγοριοποιητή με ειδικά σχεδιασμένη διεπαφή. Ο χρήστης είναι σε θέση να καθορίσει τον αριθμό δειγμάτων που θα χρησιμοποιηθούν, καθώς επίσης και τον αριθμό δειγμάτων που έχουν χρησιμοποιηθεί ήδη για την εκπαίδευση της κατηγορίας. Με τον τρόπο αυτό το σύστημα αποφεύγει να χρησιμοποιήσει για δεύτερη φορά τα ίδια δείγματα. Η διαδικασία εκπαίδευσης εκτός από συγκεκριμένα έγγραφα επιλογής του χρήστη μπορεί να πραγματοποιήσει αναζήτηση δειγμάτων με χρήση λέξεων κλειδιών απευθείας στις σελίδες του καταλόγου ODP ή και σε γενική αναζήτηση σε κάποια μηχανή όπως το Google. Ως βασική πλατφόρμα υλοποίησης του κατηγοριοποιητή χρησιμοποιήσαμε τη βιβλιοθήκη libSVM 2,4 [46].

Ο μηχανισμός που προτείνουμε διαθέτει επιπλέον δύο επιλογές λειτουργίας. Στην πρώτη, για να διευκολυνθεί σε ταχύτητα η κατηγοριοποίηση υπάρχει η δυνατότητα να οριστεί ως χρήσιμο κείμενο (είσοδος) εκπαίδευσης η σύνοψη (περίληψη) κάθε εγγράφου που εντοπίζεται κατά την αναζήτηση και όχι ολόκληρο το σώμα κειμένου του εγγράφου. Η περίληψη αυτή δίνεται από κάθε μηχανή αναζήτησης με τη μορφή μιας σύντομης παραγράφου μαζί με τη λίστα των αποτελεσμάτων αναζήτησης και επομένως δεν απαιτείτε επιπλέον δικτυακή μεταφορά δεδομένων. Ως δεύτερη εναλλακτική επιλογή λειτουργίας για όσους επιθυμούν να επιμείνουν σε ασφαλέστερη κατηγοριοποίηση, είναι διαθέσιμη η εκπαίδευση στο σύνολο του κειμένου κάθε προκείμενου σελίδας. Η δεύτερη επιλογή έχει μεγαλύτερο υπολογιστικό κόστος και επομένως περισσότερο χρόνο αναμονής για την κατηγοριοποίηση.

Η διαίσθηση πίσω από αυτή τη διαφοροποίηση ελέγχθηκε με πειραματική αξιολόγηση σε εργαστηριακό περιβάλλον από 14 χρήστες-εθελοντές. Στην υλοποίησή χρησιμοποιήθηκε μια πολιτική λανθάνουσας αποθήκευσης (caching strategy) για την επιπλέον επιτάχυνση κατά την κατηγοριοποίηση [36].

Στο σχήμα 33 παρουσιάζονται τα αποτελέσματα της αξιολόγησης ακρίβειας των δύο λειτουργικών επιλογών. Βλέπουμε στον κάθετο άξονα το ποσοστό ευστοχίας κατά την κατηγοριοποίησης για μια σειρά από μονολεκτικές αναζητήσεις (οριζόντιος άξονας) που αξιολογήθηκαν. Κατά μέσο όρο ο κατηγοριοποιητής αποδίδει εύστοχη κατηγοριοποίηση σε ποσοστό 89,29% με χρήση πλήρες κειμένου και σε 52,14% με χρήση της σύνοψης (περίληψης) μόνο. Το ποσοστό ευστοχίας με χρήση της σύνοψης αυξάνεται με τη χρήση του προσωπικού κατηγοριοποιητή για κάθε χρήστη έως και 35% επιπλέον.

4.15 Συνεισφορά, συμπεράσματα και μελλοντικές κατευθύνσεις

Η εξατομίκευση των αποτελεσμάτων αναζήτησης ιστού μπορεί να επιτευχθεί όταν εκτελείται προσαρμογή των αποτελεσμάτων της αναζήτησης στις απαιτήσεις του χρήστη. Προτείνουμε μια σειρά από τρεις αλγορίθμους σύμφωνα με την εργασία [25] που βασίζονται σε τεχνικές μετρικής συνδέσμων για να εξατομικεύσουμε τα αποτελέσματα της αναζήτησης ιστού. Για να αξιολογήσουμε τους αλγορίθμους που παρουσιάστηκαν, εκτελέσαμε μια σειρά από πειράματα με βάση τις δημοφιλείς λίστες κατηγοριών του Open Directory Project και τα αποτελέσματα ερωτημάτων του Google. Οι καμπύλες ακρίβειας και ανάκλησης των αποτελεσμάτων δείχνουν ότι η στρατηγική που περιγράφεται είναι αποτελεσματική και αποδοτική ιδιαίτερα στο πρώτο 20% της ανάκλησης, ενώ κατά μέσο όρο ο προτεινόμενος αλγόριθμος εξατομίκευσης αυξάνει την ακρίβεια έως 21,48% των ανακτημένων εγγράφων.

Θεωρούμε ότι το μεγαλύτερο πλεονέκτημα των τεχνικών που προτείνουμε είναι ότι επιτρέπουν, με τη χρήση μιας αρκετά απλής μεθοδολογίας, την εξατομίκευση των αποτελεσμάτων αναζήτησης, χωρίς να επιβαρύνονται οι χρήστες σε όγκο αποθήκευσης ή με καθυστερήσεις λόγω χρόνου εκτέλεσής τους. Κατά συνέπεια ένα κεντρικό πλεονέκτημα της μεθόδου μας είναι η δυνατότητα να παράγουμε εξατομικευμένα αποτελέσματα τα οποία να «ακολουθούν»-επηρεάζονται άμεσα από τις τρέχουσες επιλογές του χρήστη και όχι ασύγχρονα όπως σε παλαιότερες μεθόδους. Επιτυγχάνουμε αυτό το σκοπό εφαρμόζοντας τεχνικές ανάλυσης και επεξεργασίας συνδέσμων όχι στο γράφο ιστού αλλά για πρώτη φορά σε αρκετά μικρότερους εξατομικευμένους γράφους που σχηματίζονται από διαθέσιμες σημασιολογικές ταξονομίες / οντολογίες.

Εφαρμόσαμε τις τεχνικές μας στην πλέον έγκυρη λίστα κατηγοριών (category pool) ODP [44] που χρησιμοποιείτε από το σύνολο των μηχανών αναζήτησης για κατηγοριοποίηση. Ωστόσο μπορεί ομοίως να εφαρμοστεί με υιοθέτηση άλλων λιστών κατηγοριών αλλά και να επεκταθεί για την επεξεργασία με σημασιολογικές λειτουργίες πάνω σε οντολογίες. Συγκεκριμένα, η αποτελεσματικότητα της τεχνικής μας είναι διπλή:

α) άμεση ταξινόμηση των κατηγοριών που επηρεάζουν την κατηγοριοποίηση και ταξινόμηση των αποτελεσμάτων αναζήτησης και

β) ενίσχυση του εξατομικευμένου γράφου κατηγοριών με βασισμένους στο χρήστη εικονικούς κόμβους κατηγορίας που διασυνδέουν σημασιολογικά σχετιζόμενες κατηγορίες.

Ο εξατομικευμένος γράφος κατηγορίας προκύπτει έμμεσα από τις κατηγορίες των επιλεγμένων αποτελεσμάτων αναζήτησης ή άμεσα με επιλογή του χρήστη. Οι εικονικές κατηγορίες που εισάγονται ως κόμβοι λειτουργούν ως ένα κανάλι που κατανέμει “ίσχυρές” συσχετίσεις μεταξύ των απογόνων. Αυτή η διαδικασία κατανέμει σπουδαιότητα από ένα κόμβο στους γειτνιάζοντες κόμβους του, ενημερώνοντας το βάρος κάθε κατηγορίας σε κάθε επανάληψη.

Επιπλέον για την καταγραφή των στοιχείων συμπεριφοράς του χρήστη κάνουμε χρήση ενός πολυεπίπεδου προφίλ το οποίο έχει δυνατότητα διαχείρισης και χρονικής απόσβεσης των στοιχείων του προφίλ σύμφωνα με την εργασία μας [16]. Ενημερώνουμε το προφίλ κυρίως με αυτόματο τρόπο και με βάση την έμμεση παρατήρηση των κινήσεων του χρήστη. Ωστόσο δίνουμε στους χρήστες τη δυνατότητα ως εναλλακτική να τροποποιήσουν ρητά το προφίλ αναζήτησής τους, εφόσον το επιθυμούν για να βελτιωθούν ακόμη περισσότερο τα αποτελέσματα.

Οι δυνατότητες της μεθόδου και η εφαρμογή των αλγορίθμων που εισάγαμε, δείξαμε ότι είναι μπορούν να προσαρμοστούν για το σχεδιασμό και την υλοποίηση μηχανισμού στοχευμένης στο θέμα συλλογής σελίδων (focused crawler). Τέτοιοι μηχανισμοί θεωρούνται ότι ανήκουν στην περιοχή της εξατομικευμένης αναζήτησης και αποτελούν εναλλακτική προσέγγιση για τη λύση του προβλήματος [20]. Εμείς εισάγαμε μία λύση η οποία μπορεί επομένως να βοηθήσει και προς τις δύο αυτές εναλλακτικές κατευθύνσεις.

Στη συνέχεια προτείναμε και παρουσιάσαμε ένα μηχανισμό σύμφωνα με τις εργασίες μας [14], [15] ο οποίος συνεισφέρει στην προσαρμοστική και εξατομικευμένη αναζήτηση πραγματοποιώντας κατηγοριοποίηση στα αποτελέσματα και παρουσιάζοντάς τα ομαδοποιημένα. Ο μηχανισμός αυτός παρέχει πολλαπλές δυνατότητες:

- Είναι δυνατό να χρησιμοποιηθεί από τους αλγορίθμους εξατομικεύσης online, offline και semi-offline
- Επιπλέον μπορεί να χρησιμοποιηθεί ως αυτόνομος μηχανισμός προσαρμοστικής αναζήτησης.

Για να μεγιστοποιήσουμε το αποτέλεσμα τις εξατομικεύσης (personalization, adaptive web) [6] ως προφίλ έχουμε χρησιμοποιήσει τη λύση προφίλ πολλαπλών επιπέδων που έχουμε εισάγει στην εργασία [16].

Μελλοντικές ερευνητικές κατευθύνσεις περιλαμβάνουν την ενασχόληση με τεχνικές που θα ενισχύουν τις δομές εξατομικευμένων κατηγοριών χρησιμοποιώντας αυτοματοποιημένες τεχνικές δόμησης των εικονικών κόμβων όπως:

- Υποστήριξη πολλαπλών επιπέδων εικονικών κόμβων για το συσχετισμό περαιτέρω εικονικών κατηγοριών από προηγούμενες συνόδους χρηστών.
- Χρήση της γνώσης του συνεργατικού προφίλ χρήστη για να περιλάβει εύστοχους «συνεργατικούς εικονικούς κόμβους» στη δομή εξατομικευμένων κατηγοριών.
- Αυτόνομη, βασισμένη σε ομότιμα δίκτυα ενημέρωση μεταξύ των χρηστών για το εξατομικευμένο γράφο εξατομικευμένης κατηγορίας, ώστε να αποφευχθούν τυχόν κίνδυνοι αποτυχίας μοναδικού σημείου.
- Υπολογισμός της ταχύτητας προσαρμογής του αλγορίθμου για να παρακολουθεί άμεσα και με ευστοχία τις αλλαγές στην εστίαση των συνηθειών αναζήτησης των χρηστών. Ένας τέτοιος υπολογισμός θα απαιτήσει πειραματική παρακολούθηση και μέτρηση απαιτεί πιο μεγάλη πειραματική περίοδο και τη συμμετοχή ακόμη περισσότερων χρηστών.

Επιπλέον κατευθύνσεις περιλαμβάνουν την ενεργοποίηση μετα-σημασιολογικών σχέσεων εφόσον οι κατηγορίες κάθε σελίδας θεωρηθούν ότι ανήκουν έννοιες οντολογίας. Κάτι τέτοιο μπορεί να αποτελέσει ένα μελλοντικό ανώτερο επίπεδο συσχέτισης των εννοιών που θα βασιστεί στην έννοια των εικονικών κόμβων που προτείνουμε και θα ενεργοποιείτε χρησιμοποιώντας εννοιολογικές σχέσεις.

Ακόμη, πειραματική μελέτη πρόσθετων πολιτικών διαχείρισης λανθάνουσας μνήμης για την ελαχιστοποίηση του αριθμού εκτέλεσης κατηγοριοποίησης τοπικά σε κάθε χρήστη. Αυτό στηρίζεται στη διαίσθηση ότι οι χρήστες επισκέπτονται από καιρό σε καιρό σελίδες σε ίδιες τοποθεσίες που ενώ έχουν αλλάξει τα περιεχόμενά τους η θεματική κατηγοριοποίησή τους παραμένει η ίδια και δεν χρειάζεται επαναληπτική επεξεργασία (π.χ. αθλητικές σελίδες εφημερίδων).

Μια εναλλακτική ερευνητική κατεύθυνση είναι η δικτυοκεντρική εξατομίκευση στις κατηγορίες σε περιβάλλοντα κοινοτήτων που λειτουργούν στη βάση τους με ομότιμα δίκτυα. Παράλληλη επέκταση μιας τέτοιας προσέγγισης μπορεί να είναι η εφαρμογή της κατηγοριοποίησης για την υποστήριξη αναζήτησης υπηρεσιών Web Services. Η κατηγοριοποίηση στην περίπτωση αυτή μπορεί να βασιστεί πέρα σε υποστηριζόμενες σημασιολογικές περιγραφές που πλέον καταχωρούνται σε μεταδεδομένα.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ 4

1. Arampatzis, A.T., van der Weide, T.P., Koster, C.H.A. and van Bommel, P. (2000): Term selection for filtering based distribution of terms over time, RIAO' 2000 Conference Proceedings, Vol. 2, 12-14/4, Paris, France, 1221-1237
2. Baeza-Yates, R. and Ribeiro-Neto. Modern Information Retrieval. ACM Press Series/Addison Wesley, New York, May 1999.
3. Beck, J., Stern, M., and Haugsjaa, E. (1996) Applications of AI in Education, ACM Crossroads.
4. Boyan, J., Freitag, D., and Joachims, T. A machine learning architecture for optimizing web search engines. In AAAI'96 (August, 1996) Workshop on Internet Based Information Systems. URL: http://www.cs.cornell.edu/People/tj/publications/boyan_et_al_96a.pdf
5. Brin S. and L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proc. of the 7th World-Wide Web WWW Conference, 1998.
6. Brusilovsky, P. and Maybury, M.T., "From Adaptive Hypermedia to the Adaptive Web", Communications of the ACM Journal, 45(5), May 2002, pp. 31-33.
7. Chakrabarti S., B. Dom, P. Indyk, Enhanced Hypertext Categorization Using Hyperlinks, In Proceedings of the ACM International Conference on Management of Data SIGMOD98, 1998, pp. 1-12
8. Chakrabarti, S., van der Berg, M., and Dom, B., "Focused crawling: a new approach to topic-specific web resource discovery", in Computer Networks Vol. 31(11-16), 1999, pp. 1623-1640
9. Chang H. , D. Cohn, and A. K. McCallum, "Learning to Create Customized Authority Lists", proceedings of the 17th International Conference on Machine Learning (ICML 2000), 2000, pp. 49-54.
10. De Bra, P., Houben, G.J., Kornatzky, Y., Post, R., "Information Retrieval in Distributed Hypertexts", proceedings of RIAO-94 Conference, pp. 481-492, New York, 1994.
11. Diligenti, M., Coetze, F., Lawrence, S., Miles, C. and Gori, M., Focused Crawling using Context Graphs, 26th International Conference on Very Large Databases, VLDB 2000, pp. 527-534.
12. Dumais, S.T, Cutrell E. and Chen H. Bringing order to the web: Optimizing search by showing results in context. In Proceedings of CHI'01, Human Factors in Computing Systems, April 2001, pp. 277-283.
13. Dumais, S.T., Platt, J., Heckerman, D., Sahami, M., "Inductive learning algorithms and representations for text categorization", proceedings of ACM CIKM98, Nov. 1998, pp. 148-155.
14. Garofalakis, J., Matsoukas, Th., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., "Personalization Techniques for Web Search Results Categorization", in the proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE EEE 2005), 29 March - 1 April 2005 in Hong Kong, China, pp. 148-151.
15. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A. "Optimization mechanism for Web Search Results using Topic Knowledge", in the International Journal of Knowledge and Learning (IJKL), Vol. 2 (1&2), 2006, ISSN (Print): 1741-1009, Inderscience Publishers, in press.
16. Garofalakis, J., Sakkopoulos, E., Sirmakessis, S., Tsakalidis, A.K., "Integrating Adaptive Techniques into Virtual University Learning Environment", IEEE International Conference on Advanced Learning Technologies (ICALT'02), Kazan, Russia, 2002, pp. 28-33.
17. Haveliwalla T.H., "Topic-Sensitive PageRank", proceedings of the 11th International World Wide Web Conference (WWW2002), 2002, pp. 517-526.
18. Hawking, D., Craswell, N., and Griffiths, K. Which Search Engine is Best at Finding Online Services? In Proceedings of WWW10 (Hong Kong, May 2001), Posters. Actual poster in <http://pigfish.vic.cmis.csiro.au/~nickc/pubs/www10actualposter.pdf>.

19. Henzinger M., Algorithmic Challenges in Web Search Engines, *Internet Mathematics* 1, 1, 2003, pp.115-126
20. Henzinger, M., "Hyperlink analysis on the world wide web", in *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, 2005, pp. 1-3
21. Howe A.E. and D. Dreilinger, "SavvySearch: A meta-search engine that learns which search engines to query", *AI Magazine*, 18(2), 1997, pp. 19-25.
22. Jansen, B., Spink, A., and Saracevic, T. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2) (2000), 207-227.
23. Kleinberg J.M., "Authoritative Sources in a Hyperlinked Environment", *Journal of the ACM*, 46(5), 1999, pp. 604-632.
24. Liu F., Cl. Yu, W. Meng, "Personalized Web Search by Mapping User Queries to Categories", in *Proc. CIKM*, pp.558-565
25. Makris, C., Panagis, Y., Sakkopoulos E., Tsakalidis, A. "Category ranking for personalized search", in the *Data and Knowledge Engineering Journal*, Elsevier Science.
26. Meng W., C. Yu, K. Liu, "Building Efficient and Effective Metasearch Engines", *ACM Computing Surveys*, 34(1), March 2002, pp.48-89.
27. Michael Herscovici, Michal Jacovi, Yoëlle S. Maarek, Dan Pelleg, Menachem Shtalhim, Sigalit Ur: The Shark-Search Algorithm. An Application: Tailored Web Site Mapping, *proceedings of the WWW7 Conference*, 1998, pp. 317-326
28. Nguyen, B., Vazirgiannis, M., Varlamis, I., and Halkidi, M., "THESUS: Organizing Web document collections based on link semantics". *VLDB journal*, special issue on "Semantic Web", vol. 12, number 4, pp. 320-332, November 2003
29. Ozsoyoglu G., A. Al-Hamdani *Web Information Resource Discovery: Past, Present, and Future*, *ISCIS2003*, pp. 9—18, LNCS 2869, 2003.
30. Platt, J., "Fast training of support vector machines using sequential minimal optimization", *Advances in Kernel Methods -Support Vector Learning*. B. Scholkopf, C. Burges, and A. Smola, eds., MIT Press, 1999, pp. 185-208.
31. Rocchio, J., "Relevance Feedback in Information Retrieval", in G. Salton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
32. Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. Analysis of a very large web search engine query log. *SIGIR Forum* 33(1) (Fall, 1999), pp. 6-12.
33. Sugiyama, K., Hatano, K., Yoshikawa, K. "Adaptive web search based on user profile constructed without any effort from users", in the *proceedings of the WWW Conference*, 2004, pp. 675-684
34. WordNet, Princeton University Cognitive Science Laboratory - <http://wordnet.princeton.edu/>
35. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, p. 133–138, Las Cruces, New Mexico.
36. Zeng, H., He, Q.C., Chen, Z., Ma, W.Y., "Learning To Cluster Search Results", in the 27th Annual International ACM SIGIR Conference (SIGIR'2004), July 2004, pp. 210-217
37. A9 Search <http://www.a9.com>
38. Adutopia <http://www.adutopia.com/>
39. Google Search <http://www.google.com>
40. Teoma <http://www.teoma.com/>
41. Vivisimo <http://www.vivisimo.com/>
42. Wisenut <http://www.wisenut.com/>
43. Yahoo! <http://www.yahoo.com>
44. ODP dmoz <http://www.dmoz.org/about.html>
45. ODP dumps <http://rdf.dmoz.org>
46. libSVM <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

5 ΜΕΛΕΤΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΠΕΡΙΠΤΩΣΕΩΝ ΕΦΑΡΜΟΓΗΣ

5.1 Εισαγωγή - Μελέτη και παρουσίαση περιπτώσεων εφαρμογής

Στα πλαίσια της παρούσας διατριβής εισάγαμε μια σειρά αποδοτικών αλγορίθμων και τεχνολογιών στα προηγούμενα κεφάλαια με σκοπό τη δυναμική διαχείριση και την εξατομίκευση της αναζήτησης υπηρεσιών και πληροφοριών σε δικτυοκεντρικά περιβάλλοντα. Μαζί παρουσιάσαμε ήδη τα πειραματικά τους αποτελέσματα σε συγκεκριμένες υλοποιήσεις που τις υποστηρίζουν. (περίπτωση διαχείρισης δεδομένων μεγάλου όγκου τηλεπικοινωνιακού φορέα κεφάλαιο 2, ομότιμο δίκτυο διεκπεραίωσης αναζήτησης υπηρεσιών WS κεφάλαιο 3 και λύση εξατομικευμένης αναζήτησης πληροφοριών κεφάλαιο 4).

Για να ολοκληρώσουμε την παρουσίαση του έργου μας θα παρουσιάσουμε στο παρόν κεφάλαιο την περιγραφή μιας σειράς περιπτώσεων εφαρμογής [5], [6], [7], [8], [12], [13], [14] σε περιβάλλοντα τα οποία καλύπτουν ένα ευρύ φάσμα πεδίων αναγκών. Στόχος είναι να δείξουμε τη γενικότητα των μεθόδων που έχουν παρουσιαστεί και τη δυνατότητα εφαρμογής τους σε διαφορετικά επιστημονικά πεδία, επιχειρησιακά προβλήματα και περιοχές εφαρμογών.

5.2 Διαχείριση ασύρματων και ενσύρματων δικτυακών πόρων και συσκευών

Στις μέρες μας υπάρχει μεγάλη αύξηση της χρήσης εναλλακτικών δικτύων επικοινωνίας, κυρίως ασύρματων, τα οποία όμως έχουν έλλειψη ολοκληρωμένης αντιμετώπισης για τη διαχείριση από κοινού πόρων και της κοινότητας χρηστών τους. Παρουσιάσαμε μια νέα λύση βασισμένη σε τεχνολογίες παγκόσμιου ιστού για τη διαχείριση μητροπολιτικού δικτύου ασύρματης επικοινωνίας [8]. Το περιβάλλον αυτό πλήρως δικτυοκεντρικό υποστηρίζει τεχνικές εξατομίκευσης πληροφοριών που αναφέρονται στη διαχείριση και παρακολούθηση των ασύρματων κόμβων. Χρησιμοποιήθηκαν αρχιτεκτονικές που προτείναμε για το προφίλ εξατομίκευσης [9] χρηστών και το περιβάλλον λειτουργεί για τη διαχείριση του δικτύου εδώ και δύο ήδη χρόνια στην πόλη της Πάτρας. Η αξιολόγηση του συστήματος από τους χρήστες έδειξε ότι ο μηχανισμός παράδοσης πληροφοριών ήταν εύστοχος και αποδοτικός.

5.3 Ηλεκτρονική μάθηση και εκπαίδευση (e-learning)

Στα πλαίσια των εφαρμογών ηλεκτρονικής μάθησης και εκπαίδευσης έχουν υιοθετηθεί και υλοποιηθεί πολλές και ενδιαφέρουσες προσεγγίσεις. Στην περίπτωση

μας και σε συνεργασία με την Ανώτατη Σχολή Καλών Τεχνών (ΑΣΚΤ Αθηνών) υλοποιήσαμε εκπαιδευτικό περιβάλλον [7] με σκοπό να θέσουμε κάτω από μια κοινή ομπρέλα όλους εκείνους του ερευνητές, επιστήμονες, καλλιτέχνες, ακαδημαϊκούς που ενδιαφέρονται για την καλλιέργεια και εκπαίδευση των ψηφιακών μορφών τέχνης. Παρουσιάσαμε μια λύση με την οποία οι φοιτητές της ΑΣΚΤ προπτυχιακοί, μεταπτυχιακοί αλλά και απόφοιτοι είναι δυνατό να επικοινωνούν τις ιδέες τους και να έχουν πρόσβαση στη γνώση με το περιβάλλον Ευρυδίκη II. Πρόκειται για ένα βασισμένο σε τεχνολογίες παγκόσμιου ιστού περιβάλλον το οποίο για να διευκολύνει την πρόσβαση στη γνώση χρησιμοποιεί τεχνικές εξατομίκευσης που δόθηκαν στο κεφάλαιο 4 για να παραδώσει με μεγαλύτερη αμεσότητα πληροφορία ιστού σχετική με τα μαθήματα της σχολής. Το περιβάλλον έχει δοκιμαστεί για την παράδοση πληροφορίας που αφορά μαθήματα ιστορίας της τέχνης σε μια σειρά από διαφορετικές θεματικές περιόδους της ελληνικής ζωγραφικής.

5.4 Σημασιολογική εξόρυξη και διαχείριση πολυμεσικών δεδομένων

Στα πλαίσια της επέκτασης των αλγορίθμων που παρουσιάζονται στη διδακτορική διατριβή στο πεδίο εφαρμογών που χρησιμοποιούν τεχνολογίες του σημασιολογικού ιστού (semantic web) [2], [4] παρουσιάζουμε μια προσέγγιση [5], [6] η οποία επεκτείνει τις δυνατότητες διαχείρισης και παράδοσης πολυμεσικού web περιεχομένου για μονογραφίες επαγγελματικού προσανατολισμού σε ένα πληροφοριακό σύστημα εκπαίδευσης με σημασιολογικά περιγραφόμενες Web Services [3].

Η προτεινόμενη λύση παρουσιάστηκε στις εργασίες μας [5], [6] και εφαρμόζει τις τεχνικές αναζήτησης WS που εισάγαμε με σκοπό τη διαχείριση πολυμεσικής πληροφορίας ενός πληροφοριακού συστήματος εκπαίδευσης. Η κεντρική ιδέα στην οποία βασίσαμε την αρχιτεκτονική μας είναι η λειτουργική συμμετοχή όλων των εμπλεκόμενων φορέων να είναι βασισμένη πλήρως στην τεχνολογία WS. Με στόχο ώστε όλοι οι πάροχοι του πολυμεσικού web περιεχομένου στο σύστημα εκπαίδευσης να διαμοιράζονται πηγές πληροφόρησης και ως αποτέλεσμα να αυξάνουν τη συνολική αποθηκευτική δυνατότητα του συστήματος. Για να είναι δυνατός ο γρήγορος και αποδοτικά εύστοχος εντοπισμός μιας WS ενός παρόχου που δημοσιεύει επαγγελματικές μονογραφίες ορίσαμε και ακολουθούμε μια σημασιολογική περιγραφή για τις WS με βάση μία προτεινόμενη οντολογία. Επιπλέον καθορίσαμε μια σειρά από μετρικές ποιοτικής αξιολόγησης της κάθε μονογραφίας τις οποίες θεωρήσαμε ως χαρακτηριστικά ποιότητας (QoWS). Κατά συνέπεια είναι εφικτή η δυναμική αξιολόγηση και αναζήτηση του περιεχομένου που διαμοιράζεται μέσα από WS στο δικτυοκεντρικό αυτό σύστημα από τους φορείς με χρήση αλγορίθμων αναζήτησης από το κεφάλαιο 2.

Στο δικτυοκεντρικό αυτό σύστημα μπορούν να συμμετέχουν φορείς που προσφέρουν επαγγελματικές μονογραφίες για την ενημέρωση μαθητών που σπουδάζουν στην τεχνική εκπαίδευση και κατάρτιση ή που επιθυμούν να ενημερωθούν για αυτή (ΚΕΚ, ΙΕΚ, ΚΕΣΥΠ). Μόλις ένας χρήστης περιγράψει τη μονογραφία – επάγγελμα που

επιθυμεί ενεργοποιείτε ο μηχανισμός αναζήτησης WS που αναζητά βασιζόμενος στις σημασιολογικές περιγραφές τους. Για την δυναμική επιλογή βασιζόμαστε στο μοντέλο ποιοτικής αξιολόγησης το οποίο τροφοδοτείται από τους χρήστες αμέσως μετά από τη χρήση μιας πολυμεσικής μονογραφίας.

Μελλοντικές κατευθύνσεις που προέκυψαν από την ενασχόλησή μας με το σύστημα αυτό είναι η μελέτη προς την κατεύθυνση της αυτοματοποιημένης επιλογής του κατάλληλου WS μέσα από ένα έξυπνο μηχανισμό επιλογής ο οποίος πέρα από τα ποιοτικά χαρακτηριστικά θα στηρίζει και μέρη τουλάχιστον των λειτουργικών προδιαγραφών μέσα από διαδικασίες εξόρυξης γνώσης με οντολογίες για WS.

5.5 Πλαίσιο επανοργάνωσης δομής και εξατομίκευσης περιεχομένου εφαρμογών ιστού

Μία επιπλέον κατηγορία εφαρμογών με την οποία ασχοληθήκαμε ανήκει στην περιοχή έρευνας και εφαρμογών της εξατομίκευσης περιεχομένου και επανοργάνωσης ιστοχώρων (web site). Καθώς πολλοί σχεδιαστές και διαχειριστές ιστοχώρων χρειάζονται να εντοπίζουν λανθασμένες σχεδιαστικές επιλογές ή τροποποιήσεις στο σχεδιασμό που θα αναβάθμιζαν την προσπελασιμότητα του ιστοχώρου τους, πολύ έρευνα έχει πραγματοποιηθεί για τεχνικές εξατομίκευσης περιεχομένου και επανοργάνωσης ιστοχώρων [10], [11]. Εμείς προτείνουμε με τη σειρά μας, σε συνδυασμό με τη μελέτη μας για θέματα εξατομικευμένης αναζήτησης (κεφάλαιο 4), ένα αλγοριθμικό πλαίσιο [13] για την προσαρμοστική παρουσίαση περιεχομένου σύμφωνα με τις απαιτήσεις του χρήστη το οποίο συνεισφέρει α) έναν ασύγχρονο αλγόριθμο αναδιοργάνωσης στηριγμένο σε μετρικές μας που λαμβάνουν υπόψη ποιοτικά και δομικά χαρακτηριστικά του ιστοχώρου και β) έναν online μηχανισμό εξατομίκευσης για να αναδειξουμε τις δημοφιλείς και πρόσφατες περιοχές του γράφου του ιστοχώρου σε μια προσαρμοστική λίστα προτάσεων ανάλογη για κάθε χρήστη. Αυτές οι τεχνικές ουσιαστικά καθορίζουν ανακατεύθυνση συνδέσμων, σχολιασμό σελίδων, μετατροπές και αναταξινόμησης του γράφου των ιστοχώρων.

Οι τροποποιήσεις αυτές προτείνονται να πραγματοποιηθούν όμως μόνο χειροκίνητα πράγμα που κάνει ενδιαφέρον τη μελέτη μηχανισμών για αυτοματοποίηση της αναδιοργάνωσης. Προς τη κατεύθυνση αυτή υπάρχουν εργασίες από το χώρο της μοντελοποίησης λογισμικού αλλά παρ' όλ' αυτά καμία από αυτές τις προσπάθειες δεν ολοκληρώνεται σε αυτοματοποιημένο επίπεδο με τη διαδικασία εξατομίκευσης περιεχομένου και αναδιοργάνωσης ιστοχώρων. Εμείς προτείνουμε ένα πλαίσιο διερεύνησης των δυνατοτήτων αυτοματοποίησης βασισμένο στην XML με σκοπό μια πρώτη συνεισφορά προς την κατεύθυνση αυτή και βάση για μελλοντική έρευνα [12].

5.6 Δικτυακά πληροφοριακά συστήματα γεωπονίας

Μία εναλλακτική εφαρμογή δικτυοκεντρικής προσέγγισής μας έρχεται από τον ευρύτερο χώρο της γεωπονίας. Σχεδιάσαμε ένα περιβάλλον για τη διαχείριση της καλλιέργειας καπνού σε επιπλέον σύστημα. Τέτοια συστήματα καλλιέργειας θεωρούνται τα πλέον σύγχρονα και έχουν ήδη εμφανιστεί από τα μέσα της δεκαετίας

του 1980 (από την Speedling, Inc). Στην Ελλάδα περίπου το 90% του καπνού παράγεται σε τέτοια θερμοκηπιακά πλωτά συστήματα. Η παραγωγή με τα συστήματα αυτά εξαρτάται από πολλούς παράγοντες όπως η ποιότητα του νερού (συγκεντρώσεις, ασβεστίου, βαρέων μετάλλων) αλλά και το μέσο καλλιέργειας. Κατά συνέπεια απαιτείται η διαχείριση των συγκεκριμένων χαρακτηριστικών που διοχετεύονται στο νερό ανάλογα με το είδος του καλλιεργούμενου μέσου. Ως αποτέλεσμα κρίνεται σκόπιμο να υποστηρίξουμε τις διαδικασίες της καλλιέργειας με ολοκληρωμένα δικτυακά πληροφοριακά περιβάλλοντα. Σε συνεργασία με το ΑΤΕΙ Μεσολογγίου και το τμήμα Θερμοκηπιακών καλλιεργειών και Ανθοκομίας, προτείναμε το δικτυακό πληροφοριακό σύστημα FloatNET [14] το οποίο έχει ως σκοπό να δώσει στον παραγωγό την πρόσβαση σε συμβουλευτικές υπηρεσίες και διαχειριστική βοήθεια για την καλλιέργεια καπνού με το εν λόγω σύστημα εξατομικευμένο στις ανάγκες του [9], ώστε να ενθαρρύνουμε την ανάπτυξη και την υποστήριξή τους στην Ελλάδα.

5.7 Συνεισφορά και συμπεράσματα

Συνολικά στο τελευταίο αυτό κεφάλαιο αναφερθήκαμε σε μια σειρά από εφαρμογές, αρχιτεκτονικές και λειτουργικά πλαίσια τα οποία αφορούν δικτυακά πληροφοριακά περιβάλλοντα στα οποία εφαρμόζουμε τεχνικές διαχείρισης υπηρεσιών και μηχανισμούς εξατομικευσης πληροφοριών. Παρουσιάσαμε εργασίες μας από ένα ευρύ φάσμα πεδίων εφαρμογής που περιλαμβάνει ενασχόληση με:

1. Δικτυακή διαχείριση ασύρματων και ενσύρματων πόρων και συσκευών
2. Ηλεκτρονική μάθηση και εκπαίδευση
3. Σημασιολογική εξόρυξη και διαχείριση πολυμεσικών δεδομένων με WS
4. Εξατομίκευση περιεχομένου
5. Δικτυακά πληροφοριακά συστήματα γεωπονίας

Στόχος της παρουσίασης των παραπάνω λύσεων είναι η επιβεβαίωση της δυνατότητας εφαρμογής των αποδοτικών αλγορίθμων, που παρουσιάστηκαν στα τρία προηγούμενα κεφάλαια και αποτελούν τα κύρια ερευνητικά αποτελέσματα της παρούσας διατριβής, σε πολλαπλά προβλήματα και τεχνολογικά πεδία που χρησιμοποιούν δικτυακά πληροφοριακά συστήματα και εφαρμογές παγκόσμιου ιστού.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ 5

1. Austin, D., Barbir, A., Ferris, C. and Garg, S. (eds.): Web Service Architecture Requirements. W3C Working Group Notes (2004) <http://www.w3.org/TR/wsa-reqs>
2. W3C Semantic Web <http://www.w3.org/2001/sw/>
3. Patil, A., Oundhakar, S., Sheth, A. and Verma, K., (2004) 'METEOR-S Web Service Annotation Framework', Proceedings of WWW 2004 Conference, pp. 553-562.
4. Gruber, T.R. (1993) 'A translation approach to portable ontologies', Knowledge Acquisition, Vol.5, No.2, pp.199-220.
5. Sakkopoulos, E., Kanellopoulos, D., A. Tsakalidis, "Semantic Mining and Web Service discovery techniques for media resources management", in the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol. 1 (1), ISSN (Online): 1744-263X - ISSN (Print): 1744-2621, Inderscience Publishers, corrected proof, in press.
6. Adamopoulou, P., Kanellopoulos, D., Sakkopoulos, E. Tsakalidis, A., "Semantic learning interventions using web services technology", IASTED 2005 International Conference Web Based Education, (WBE 2005), Web Based Teaching and Learning Technologies track, 2/2005, pp. 528-533
7. Sakkopoulos, E., Tsakalidis, A., "Utilizing Complementary Know-How: Advanced Fine Arts meeting Information Technology to provide a Virtual University for artists - students and alumni", ACM SIGUCCS User Services conference 2002, November 22, 2002, Providence, Rhode Island, USA.
8. Dimopoulos, E., Panousis, A., Sakkopoulos, E., Tsakalidis, A. "WiFiAdmin: An Intelligent Web-based Management Environment for Wireless Networks", in the IEEE International Conference on Information Technology and Applications 2005 (IEEE ICITA), 4/2005, Sydney 4-7/7/2005.
9. Garofalakis, J., Sakkopoulos, E., Sirmakessis, S., Tsakalidis, A.K., "Integrating Adaptive Techniques into Virtual University Learning Environment", IEEE International Conference on Advanced Learning Technologies (ICALT02), Kazan, Russia, 2002, pp. 28-33.
10. Garofalakis, J., Kappos, P., Mourloukos, D. "Web Site Optimization Using Page Popularity", IEEE Internet Computing, July-August 1999, pp. 22-29.
11. Srikant, R., Yang, Y., "Mining Web Logs to improve Website organization", WWW10, 2001, pp 430-437
12. Christopoulou, E., Garofalakis, J., Makris, C., Panagis, Y., Sakkopoulos, E., Tsakalidis, A. , "Automating Restructuring of Web Applications", ACM Hypertext 2002, June 11-15, 2002, College Park, Maryland, USA., ACM 1-58113-477-0/02/0006.
13. Makris, C., Panagis, Y., Sakkopoulos, E., Tsakalidis, A., "An Algorithmic Framework for Adaptive Web Content", in the Int WS on Adaptive and Personalized Semantic Web, in conjunction with ACM Hypertext 2005, (ACM HT05), proceedings published by Springer Verlag, Studies in Computational Intelligence, 2005
14. Liopa-Tsakalidis, A., Barouchas, P., Kouloupoulos, A., Sakkopoulos, A., Tzimas, G., Panagiotopoulos, L., "FloatNet: An Intelligent Environment for Tobacco Transplants Production", in the proceedings of ITAFE05- "International Congress on Information Technologies in Agriculture, Food and Environment", 2005.

ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ, ΣΥΝΕΙΣΦΟΡΑΣ ΚΑΙ ΑΝΟΙΚΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ

Για να ολοκληρώσουμε την παρουσίαση της διδακτορικής εργασίας μας κλείνουμε τη διδακτορική διατριβή μας με την ανακεφαλαίωση των αποτελεσμάτων που παρουσιάστηκαν και μια σειρά από ανοικτά προβλήματα. Η παρούσα ανακεφαλαίωση λειτουργεί ως μία σύνοψη της ανάλυσης αποτελεσμάτων, της συνεισφοράς και των μελλοντικών κατευθύνσεων που παρουσιάζονται αναλυτικότερα στο τέλος κάθε κεφαλαίου.

Στόχος είναι να παρέχουμε έναν οδικό χάρτη της συνεισφοράς της εργασίας μας και των ανοικτών προβλημάτων καθώς η ανάλυση των συμπερασμάτων μας και τα επόμενα ερευνητικά βήματα. Σημειώνουμε ότι η βιβλιογραφία αναφέρεται στις εργασίες που υποστηρίζουν τη διατριβή και βρίσκεται στην ενότητα δημοσιευμένες εργασίες (βλ. σελ. xxiii).

Συμπεράσματα και συνεισφορά της διδακτορικής διατριβής

Αρχικά, πραγματοποιήσαμε μια ανάλυση των αλγοριθμικών τεχνικών που είναι διαθέσιμες για τη διαχείριση και αναζήτηση υπηρεσιών και πληροφοριών για δικτυακά πληροφοριακά συστήματα σύμφωνα με την εργασία μας [Π5]. Ιδιαίτερη έμφαση δώσαμε σε εκείνες τις προσεγγίσεις που εφαρμόζουν διασφάλιση ή κατ'ελάχιστο διαχείριση ποιότητας υπηρεσίας (QoS). Μελετήσαμε και οργανώσαμε τις λύσεις αυτές σύμφωνα με μια νέα κατηγοριοποίηση μέρος της οποίας παρουσιάσαμε στην εργασία [Σ7]. Περιλαμβάνουμε σειρά από συγκριτικούς χάρτες και πίνακες που παρουσιάζουν συμπερασματικά ποιες λύσεις είναι οι πλέον ενδεδειγμένες σε μια σειρά από δημοφιλή προβλήματα. Πρόκειται ουσιαστικά για μια εποικοδομητική ανάλυση τεχνολογιών μηχανικής παγκόσμιου ιστού (web engineering) που αφορούν αναζήτηση υπηρεσιών (web services) σε δικτυακά πληροφοριακά συστήματα (net-centric information systems).

Στη συνέχεια στο δεύτερο κεφάλαιο, παρουσιάζουμε νέους αλγόριθμους σύμφωνα με τις εργασίες μας [Π1] και [Σ4] για την αποδοτική επιλογή υπηρεσίας (Web Service) που λαμβάνουν υπόψη πέρα από τα λειτουργικά χαρακτηριστικά και μη λειτουργικές προδιαγραφές για ποιότητα (Quality of Service-QoS) κατά την προσπάθεια σύνδεσης αλλά και χρήσης (consumption) της Web Service. Επιπλέον παρουσιάσαμε εναλλακτικές αποδοτικές αλγοριθμικές προσεγγίσεις (βλ. Αλγόριθμος 2, Αλγόριθμος 3) για την υποστήριξη παραδοσιακής αλλά και multicast-enabled δικτυακής υποδομής. Τονίζεται ότι οι μέθοδοι που εισήχθησαν λαμβάνουν υπόψη πέρα από τα λειτουργικά χαρακτηριστικά που απαιτούνται, παραμέτρους ποιότητας υπηρεσίας (QoWS) (μη λειτουργικά χαρακτηριστικά) και τους μεγιστοποιούν με

σκοπό να επιλέξουν δυναμικά κατά τη στιγμή της προσπάθειας κατανάλωσης την καταλληλότερη Web Service.

Πειραματικά αποτελέσματα έδειξαν ότι στις περιπτώσεις WS που διαχειρίζονται μεγάλου όγκου δεδομένα σε ένα δικτυακό πληροφοριακό σύστημα, ο αποδοτικός αλγόριθμος επιλογής που εισάγαμε είναι πιο αποτελεσματική και αποδοτική λύση σε σύγκριση με την υπάρχουσα μέθοδο αναζήτησης στο UDDI και επιπλέον διασφαλίζει τα ποιοτικά χαρακτηριστικά κατά την επιλογή. Συγκεκριμένα τα συγκριτικά αποτελέσματα παρουσιάζουν βελτίωση του χρόνου απόκρισης έως και 66,47%. Το μέσο κέρδος σε όλες τις περιπτώσεις φτάνει στο 48,09% όπως φαίνεται στο σχήμα 15.

Στο κεφάλαιο 3 εισάγουμε αλγορίθμους αναζήτησης Web Services με χρήση πρωτότυπων δικτυακών λύσεων (decentralized - peer based networks) ειδικά σχεδιασμένων για καταλογογράφηση WS. Παρουσιάζουμε δύο αλγορίθμους όπου κόμβοι που αποθηκεύουν πληροφορίες για τις Web Services μπορούν αποδοτικά να εντοπιστούν χρησιμοποιώντας μια νέες δομές δεδομένων δεικτοδότησης. Ονομάσαμε τις προσεγγίσεις αυτές NIPPERS και BDT-GRID (Balanced Distributed Tree) σύμφωνα με τις εργασίες μας [Σ1] και [Σ8]. Οι δύο προσεγγίσεις έχουν κοινό αλγοριθμικό σχεδιασμό, αλλά διαφέρει η πολυπλοκότητά τους. Ο πίνακας 10 παρουσιάζει συνολικά την πολυπλοκότητα για τους προτεινόμενους αλγορίθμους σε σύγκριση με δημοφιλείς λύσεις της βιβλιογραφίας.

Συνοπτικά αποτελέσματα παρουσιάζονται στον πίνακα 11 παρακάτω σε σύγκριση με το γνωστό Chord:

Δικτυοκεντρικές Λύσεις	Αναζήτηση	Ενημέρωση	Πληροφορία Δρομολόγησης
CHORD [30]	$O(\log n)$	$O(\log^2 n)$ w.h.p.	$O(\log n)$ κόμβοι
NIPPERS [Σ1]	$O(\log \log n)$ αναμενόμενη	$O(\log \log n)$ επιμερισμένη αναμενόμενη $O(n)$ χειρότερη περίπτωση	Εκθετικά μειούμενη
BDT-GRID [Σ8]	$O(\sqrt{\log n / \log \log n})$ χειρότερη περίπτωση	$O(\sqrt{\log n / \log \log n})$ επιμερισμένη $\Theta(n)$ χειρότερη περίπτωση	Εκθετικά αυξανόμενη

Πίνακας 11: Συγκριτική Παρουσίαση Αποτελεσμάτων NIPPERS, BDT-GRID και Chord

Το NIPPERS έχει καλύτερους μεν χρόνους πολυπλοκότητας αλλά είναι αναμενόμενοι (expected), υπονοώντας ότι το σύστημα εξαρτάται από την τάξη (class) της κατανομής. Για το λόγο αυτό αναπτύχθηκε η λύση του BDT-GRID, η οποία επιτυγχάνει πολυπλοκότητα χειρότερης περίπτωσης ανεξάρτητα από την τάξη της κατανομής. Επιπλέον σημειώνουμε ότι η διαφορά μεταξύ της πολυπλοκότητας $O(\sqrt{\log n / \log \log n})$ και $O(\log \log n)$ είναι αμελητέα.

Στο κεφάλαιο 4, εισάγαμε αλγορίθμους για εξατομικευμένη αναζήτηση, ώστε να ολοκληρώσουμε τη συνεισφορά μας στο πεδίο διαχείρισης πληροφοριών στην περίπτωση αυτή για τους τελικούς χρήστες δικτυακών πληροφοριακών συστημάτων (web based/ netcentric information systems). Προτείνουμε μια σειρά από τρεις αλγορίθμους σύμφωνα με την εργασία μας [Π2] για να εξατομικεύσουμε τα αποτελέσματα της αναζήτησης ιστού (web searching). Το μεγαλύτερο πλεονέκτημα των τεχνικών που προτείνουμε είναι ότι επιτρέπουν, με τη χρήση μιας αρχικά απλής μεθοδολογίας, την εξατομίκευση των αποτελεσμάτων αναζήτησης, χωρίς να επιβαρύνονται οι χρήστες σε όγκο αποθήκευσης ή με καθυστερήσεις λόγω χρόνου εκτέλεσής τους. Επιτυγχάνουμε εξατομικευμένη αναζήτηση εφαρμόζοντας τεχνικές ανάλυσης και επεξεργασίας συνδέσμων *όχι* στο γράφο ιστού αλλά για πρώτη φορά σε αρχικά *μικρότερους* εξατομικευμένους γράφους που σχηματίζονται από διαθέσιμες σημασιολογικές ταξονομίες. Συγκεκριμένα, η αποτελεσματικότητα της τεχνικής μας περιλαμβάνει την άμεση ταξινόμηση των κατηγοριών που επηρεάζουν την κατηγοριοποίηση και ταξινόμηση των αποτελεσμάτων αναζήτησης και την ενίσχυση του εξατομικευμένου γράφου κατηγοριών με εικονικούς κόμβους κατηγορίας που διασυνδέουν κατηγορίες σημασιολογικά σχετιζόμενες σύμφωνα με το χρήστη.

Για να αξιολογήσουμε τους αλγορίθμους που παρουσιάζονται, εκτελέσαμε μια σειρά από πειράματα με βάση τις δημοφιλείς λίστες κατηγοριών του ODP και τα αποτελέσματα ερωτημάτων του Google. Οι καμπύλες ακρίβειας και ανάκλησης των αποτελεσμάτων δείχνουν ότι η στρατηγική που περιγράφεται είναι αποτελεσματική και αποδίδει ιδιαίτερα στο πρώτο 20% της ανάκλησης, ενώ η κατά μέσο όρο ακρίβεια αυξάνει έως 21,48%.

Στο ίδιο κεφάλαιο παρουσιάζουμε και κάνουμε χρήση ενός νέου πολυεπίπεδου προφίλ [Σ2] για την καταγραφή των στοιχείων συμπεριφοράς του χρήστη, το οποίο έχει μεταξύ άλλων δυνατότητα συνεργατικής αλληλεπίδρασης και χρονικής απόσβεσης των στοιχείων του προφίλ. Επιπλέον για να ολοκληρώσουμε όλες τις πτυχές της εξατομικευμένης αναζήτησης παρουσιάζουμε έναν επιπλέον μηχανισμό προσαρμοστικής παρουσίασης αποτελεσμάτων αναζήτησης [Π4], [Σ6] με χρήση επί τόπου (online) κατηγοριοποίηση με χρήση SVM (Support Vector Machine) κατηγοριοποιητή. Το πλεονέκτημα της μεθόδου είναι ο συνδυασμός προσωπικών αλλά και κοινών κατηγοριοποιητών για βελτιστοποίηση των αποτελεσμάτων.

Τέλος, στο κεφάλαιο 5 παρουσιάσαμε μια σειρά από εφαρμογές, αρχιτεκτονικές και λειτουργικά πλαίσια τα οποία αφορούν δικτυακά πληροφοριακά περιβάλλοντα στα οποία εφαρμόζουμε τεχνικές διαχείρισης υπηρεσιών και μηχανισμούς εξατομίκευσης πληροφοριών. Παρουσιάσαμε λύσεις για δικτυακή διαχείριση ασύρματων και ενσύρματων πόρων και συσκευών, για web service based διαχείριση πολυμεσικών πληροφοριών με σημασιολογική αξιολόγησή τους, για εκπαίδευση από απόσταση με χρήση προσαρμοστικού μοντέλου χρήστη για τη διαχείριση, εμφάνιση περιεχομένου και αναζήτηση καθώς και λύσεις όπου πραγματοποιείται δυναμική αναπροσαρμογή του περιεχομένου εξυπηρετητών παγκόσμιου ιστού (web server content restructuring), ώστε να διευκολύνεται συνολικά για τους επισκέπτες ο εντοπισμός της

επιθυμητής πληροφορίας. Κλείνοντας το τελευταίο κεφάλαιο, δίνονται στοιχεία για τη μελέτη περίπτωσης ενός δικτυακού πληροφοριακού συστήματος γεωπονίας.

Ανοικτά προβλήματα - Μελλοντικές κατευθύνσεις έρευνας

Καθώς αυξάνονται σε αριθμό τα δικτυακά πληροφοριακά περιβάλλοντα αλλά και η συνολική πληροφορία που διανέμεται μέσα από εφαρμογές παγκόσμιου ιστού η έρευνα για περισσότερο εξελιγμένους τους μηχανισμούς διαχείρισης και ανακάλυψης υπηρεσιών και πληροφορίας είναι αναγκαίο να φθάσει σε αποδοτικές δομές που να είναι εφαρμόσιμες όχι μόνο σε web services ή web data, αλλά και σε διαδικτυακά ή άλλα τμήματα λογισμικού γενικά. Αυτό θα απαιτούσε ορισμένες πρόσθετες προδιαγραφές σχετικά με την πλατφόρμα, τις απαιτήσεις συστήματος, το είδος επαναχρησιμοποίησης, το είδος κώδικα, και το εύρος εφαρμογής [33]. Αυτές οι προδιαγραφές θα μπορούσαν να προστεθούν ενδεχομένως μέσα στις πληροφορίες καταλόγων στην ενότητα των μεταδεδομένων διαχειριστικού χαρακτήρα. Συνεπώς, αποτελεί ανοικτό πρόβλημα μια ενοποιημένη προδιαγραφή των τμημάτων λογισμικού (που θα μπορούσε να είναι η βάση για τους μελλοντικούς καταλόγους component και τα CASE (computer aided software engineering) εργαλεία.

Μία επιπλέον κατεύθυνση αποτελεί η αξιοποίηση στρατηγικών για την επιλογή WS με διασφάλιση παραμέτρων ποιότητας με την υποστήριξη ομότιμων δικτυοκεντρικών λύσεων. Η ευρεία διάδοση των ομότιμων δικτύων μπορεί να αποτελέσει τη βάση για ένα νέο κατάλογο εγγραφής WS πλήρως αποκεντρωποιημένο. Κατά συνέπεια η μελέτη της δυνατότητας ολοκλήρωσης της τεχνικής μας στην περίπτωση αυτών των δικτύων ώστε να υποστηρίζεται δυναμική επιλογή κατά τη στιγμή της κατανάλωσης θα μπορούσε να πολλαπλασιάσει το κέρδος ως προς τα ποιοτικά χαρακτηριστικά και να μειώσει καταλυτικά τους χρόνους αναμονής εκτέλεσης.

Μία ακόμη κατεύθυνση στην οποία οι αλγόριθμοι επιλογής Web Service του κεφαλαίου μπορούν να οδηγήσουν σε περαιτέρω βελτιώσεις είναι στο πρόβλημα της βελτίωσης αλληλουχίας από WS στα πλαίσια μιας ροής επιχειρησιακών διαδικασιών. Στην περίπτωση αυτή διακρίνονται δύο υποκατηγορίες προβλημάτων.

Η πρώτη περιλαμβάνει τις εμφωλευμένες υπηρεσίες (component WS) όπου μία διαδικασία WS είναι η διεπαφή απλά από ένα σύνολο άλλων WS που εκτελούν μια ολοκληρωμένη λειτουργία. Στις εμφωλευμένες WS, οι δικτυακές καθυστερήσεις είναι μικρές γιατί οι WS βρίσκονται συνήθως σε μικρής κλίμακας δικτυακά περιβάλλοντα. Στην περίπτωση αυτή είναι χρήσιμο να εξεταστεί η δυνατότητα βελτίωσης των αποδόσεων για δυναμική επιλογή με ποιοτικά χαρακτηριστικά με την τεχνική μας που διαθέτει multicast μιας και το δικτυακό κόστος είναι χαμηλό.

Η δεύτερη ομάδα ενδιαφέροντος προβλήματος αφορά ροές που αποτελούν αλληλουχίες υπηρεσιών (ή ροές) (WS workflows) μιας επιχειρησιακής διαδικασίας. Στις περιπτώσεις αυτές κάθε βήμα της ροής χρειάζεται να επιλεγεί η βέλτιστη δυνατή ποιοτικά WS που ταιριάζει για τη λειτουργία του συγκεκριμένου βήματος. Η

επέκταση των τεχνικών μας για να υποστηρίξουν πολλαπλά βήματα θα έδινε μια αποδοτική λύση στο πρόβλημα αυτό.

Μία υποσχόμενη ερευνητική κατεύθυνση περιλαμβάνει την ενσωμάτωση γνώσης από τη σημασιολογική συσχέτιση των διαδικασιών σε επιχειρησιακό επίπεδο. Είναι ανοικτό πρόβλημα η αυτοματοποιημένη μετατροπή ώστε να αντικατοπτριστεί μία επιχειρησιακή ροή σε γνώση για τα βήματα και την επικοινωνία ανάμεσα σε υπηρεσίες WS. Στην κατεύθυνση αυτή, ανοικτή ακόμη παραμένει η συνολική αναβάθμιση των μοντέλων δεδομένων με τη διαμόρφωση σημασιολογικών τεχνικών και οντολογιών από την περιοχή της Ανάκτησης Πληροφορίας. Οι μηχανισμοί ανακάλυψης WS έχουν έναν ακόμα σημαντικότερο ρόλο από την αναζήτηση στον Ιστό, καθώς διευκολύνουν την ανάγκη για συνεργασία μεταξύ των επιχειρησιακών διαδικασιών και των καταναλωτών, πάνω από ευρέως αποδεκτά πρότυπα Ιστού.

Η εφαρμογή των τεχνικών μας σε άλλες περιοχές έχουν επιπλέον ενδιαφέρον μιας και θα ήταν ενδιαφέρον να αξιολογηθούν τα NIPPERS και BDT για την περίπτωση διαμοίρασης αρχείων, time-shared διαθέσιμων συστημάτων αποθήκευσης και μεγάλης κλίμακας επιστημονικών υπολογισμών σε κατανεμημένες υπολογιστικές πλατφόρμες (υπολογισμοί με GRID).

Επιπλέον, μελλοντικές ερευνητικές κατευθύνσεις περιλαμβάνουν τη μελέτη τεχνικών για την ενίσχυση της απόδοσης δομών εξατομικευμένων κατηγοριών χρησιμοποιώντας αυτοματοποιημένες τεχνικές δόμησης των εικονικών κόμβων που προτείναμε. Είναι ενδιαφέρουσα η ανάπτυξη δομής υποστηρίξης πολλαπλών επιπέδων εικονικών κόμβων για το χρονικό αλλά και χωρικό συσχετισμό των εικονικών κατηγοριών σύμφωνα με το χρόνο (διακοπές, εργασία) και τη θέση (σπίτι, γραφείο, από φορητή συσκευή) από όπου πραγματοποιείτε η σύννοδος ενός χρήστη. Επιπλέον ενδιαφέρουσα είναι η αξιοποίηση της γνώσης ενός συνεργατικού προφίλ χρήστη για να περιλάβει δυναμικούς βάρους «συνεργατικούς εικονικούς κόμβους» στη δομή ε εξατομικευμένων κατηγοριών. Ανοικτό πρόβλημα στην περίπτωση αυτή είναι η αντιμετώπιση της αποτυχίας μοναδικού σημείου (single point failure), η οποία θα μπορούσε ενδεχομένως να βρει τη λύση της μέσα από μια αποκεντριοποιημένη δομή ομότιμων δικτύων.

Μία ακόμη ερευνητική κατεύθυνση είναι η εργασία για την ανάπτυξη πρόσθετων πολιτικών διαχείρισης λανθάνουσας μνήμης για την ελαχιστοποίηση του αριθμού εκτέλεσης κατηγοριοποίησης τοπικά σε κάθε χρήστη. Αυτό στηρίζεται στη διαίσθηση ότι οι χρήστες επισκέπτονται από καιρό σε καιρό σελίδες σε ίδιες τοποθεσίες που ενώ έχουν αλλάξει τα περιεχόμενά τους η θεματική κατηγοριοποίησή τους παραμένει η ίδια και δεν χρειάζεται επαναληπτική επεξεργασία (π.χ. αθλητικές σελίδες εφημερίδων).

Τέλος, πρόκληση αποτελεί η έρευνα για την ενεργοποίηση μετα-σημασιολογικών σχέσεων εφόσον οι κατηγορίες κάθε σελίδας θεωρηθούν ότι είναι έννοιες οντολογίας. Κάτι τέτοιο θα μπορούσε να αποτελέσει ένα ανώτερο επίπεδο συσχέτισης των

εννοιών που θα βασιστεί στην έννοια των εικονικών κόμβων που προτείνουμε και θα ενεργοποιείτε χρησιμοποιώντας εννοιολογικές σχέσεις.

EYPETHPIO

A

Agricultural Information Systems, Xvi
Amortized, 114

B

BDT-GRID, 119, 120, 123
Bottleneck, 45

C

Caching, 163
CASE, 178
Centralized, 45
Chord, 51
COM, 48
Common Object Request Broker
Architecture. *Βλ.* Corba
Component Object Model. *Βλ.* COM
Component WS, 178
Computer Aided Software Engineering, 178
Contour, 84, 86, 87
Contour_Construction. *Βλ.* Contour
CORBA, 48

D

DAML, 55
DAML-S, 62
DCOM, 48
DHT, 107
Distributed Application Development, 48
Distributed Component Object Model. *Βλ.*
DCOM
Domain Name Service, 121

E

E-Learning, Xvi, 169
Expected, 114

F

Fault Tolerance, 131
Focused Crawlers, 138
Focused Web Crawlers, 157

G

Google, 149
Google Web Service, 136
GRID, 179

H

Hashing, 51
HTS, 138

I

Instances, 96
Interpolation Search Tree, 108

J

J2EE, 59
Java 2 Enterprise Edition, 59
JAXR, 59
JXTA, 51

L

Latent Semantic Indexing, 61
Loosely Coupled, 45
LSI, 61

M

Machine Processing, 106
Maximal, 87
METEOR-S, 65
Microsoft .NET, 59
Multicast, 92
Multimedia, Xvi

N

Network Management Information Systems,
Xvi
NIPPERS, 51, 108, 109, 112, 116

O

ODP. *Βλ.* Open Directory Project
Open Directory Project, 34, 136
OWL-S, 62

P

Pagerank, 33, 138
 Personalized Search, 33
 Precision-Recall, 148
 Profile, 33
 Pruning. *Βλ.* Κλάδεμα

Q

Qows, 45, 66, 82
 Quality Of Web Service, Xv

R

Ranking, 140
 Remote Method Invocation. *Βλ.* RMI
 Request-Only, 78
 Request-Response, 78
 Resource Description Framework, 136
 Response-Only, 78
 RMI, 48

S

Semantic Annotation, 65
 Semantic Mining, Xvi
 Semantic Web, 170
 Service Oriented Architecture, 42
 Sibling, 112
 Single Point Failure, 51, 107
 SOAP, 42
 Solicit-Response, 78
 Space-Filling, 61
 SWSAPI, 59

T

TF/IDF, 66
 Topic Sensitive Page-Rank, 33

U

UDDI, 42, 43, 45, 48, 77, 78
 Universal Description, Discovery And
 Integration. *Βλ.* UDDI

V

Virtual University, 160

W

Web Engineering, 41, 175
 Web Service Choreography, 45

Web Service Description Language. *Βλ.*
 WSDL
 Web Service Execution Scheme, 91
 Wrapper, 42
 WS Execution Scheme, 85
 WS Workflows, 178
 WSDL, 42, 43

A

Αλληλουχίες Υπηρεσιών, 178
 Ανάκτηση Πληροφορίας, 60
 Αποτυχία Μοναδικού Σημείου, 51
 Ασύρματη Συσκευή, 169

Δ

Διαφυλλιστής, 121, 122

E

Εμπαιδευτικό Περιβάλλον, 170
 Εμφωλευμένες Υπηρεσίες, 178
 Επανοργάνωσης Ιστοχώρων, 171
 Επιμερισμένο, 114
 Ευκλείδεια Απόσταση, 85, 92

Z

Ζωγραφικής, 170

I

Ιεραρχικά Πρωτόκολλα, 120
 Ιστορικό Προφίλ, 85

K

Καμπύλη Hilbert, 61
 Κατανεμημένη Ανάπτυξη Εφαρμογών, 48
 Κλαδέμα, 92
 Κυρίαρχο Σημείο, 87

M

Μητροπολιτικού Δικτύου, 169
 Μηχανισμοί Στοχευμένης Συλλογής Σελίδων,
 138

Π

Ποιότητα Υπηρεσίας, 45, 66, 82
 Πολιτική Λανθάνουσας Αποθήκευσης, 163
 Προσαρμοστική Επιλογή, 84
 Προσεγγιστική Αναζήτηση, 65
 Προφίλ, 34

Προφίλ Πολλαπλών Επιπέδων, 160, 165

Στοχευμένη Συλλογή Σελίδων, 157

Σ

Φ

Σημασιολογικού Ιστού, 170
Στιγμιότυπα, 96

Φύλλο-Προσανατολισμένο, 87

ΓΛΩΣΣΑΡΙ

Δικτυοκεντρικά περιβάλλοντα (netcentric)

Τα δικτυακά κατανεμημένα περιβάλλοντα όπου οι εφαρμογές και τα δεδομένα διακινούνται κυρίως μέσω εξυπηρετητών και κόμβων αποκεντρωμένων – ομότιμων δικτύων. Αυτό βρίσκεται στην αντίπερα όχθη από την αντιμετώπιση όπου ισχυροί υπολογιστές στηρίζονται σε τοπικά μέσα και πηγές. Κατά μία άποψη τα δικτυακά περιβάλλοντα αυτά μοιάζουν με τα πρώτα συστήματα mainframe και τερματιών. Ωστόσο υπάρχουν κύριες και σημαντικές διαφοροποιήσεις. Τα δικτυακά πληροφοριακά συστήματα και εφαρμογές βασίζονται σε υπηρεσίες και πληροφορίες που βρίσκονται και εκτελούνται σε διαφορετικά τεχνολογικά περιβάλλοντα (πλατφόρμες), σε κινητά δεδομένα που προσπελάνονται μέσω ταχύτατου δικτύου και σε χαμηλού κόστους πελάτες-συσκευές για την τοπική επεξεργασία

Agent

Ένας agent είναι ένα πρόγραμμα που δρα εκ μέρους ενός ιδιώτη ή μιας εταιρίας.

Agent-Αιτών

Είναι ένας agent που επιθυμεί να αλληλεπιδράσει με έναν agent-παροχέα με σκοπό να ζητήσει την εκτέλεση μιας λειτουργίας εκ μέρους του ιδιοκτήτη του, δηλαδή της οντότητας αιτούντα.

Agent-Παροχέας

Είναι ένας agent ο οποίος είναι ικανός και εξουσιοδοτημένος να εκτελεί τις λειτουργίες που σχετίζονται με ένα service εκ μέρους του ιδιοκτήτη του, δηλαδή της οντότητας παροχέα.

Binding

Συσχέτιση μεταξύ ενός περιβάλλοντος αλληλεπίδρασης, ενός πρωτοκόλλου και μιας μορφής δεδομένων. Ένα binding καθορίζει το πρωτόκολλο και τη μορφή των δεδομένων που θα χρησιμοποιηθεί στη μετάδοση των μηνυμάτων σύμφωνα με το συγκεκριμένο περιβάλλον υλοποίησης.

Χαρτογράφηση του περιβάλλοντος υλοποίησης και των σχετικών με αυτό λειτουργιών σε μια συγκεκριμένη μορφή μηνύματος και πρωτοκόλλου μετάδοσης.

End-Point

Είναι η συσχέτιση ανάμεσα σε ένα binding και σε μια διεύθυνση δικτύου, που καθορίζεται από ένα URI, και η οποία χρησιμοποιείται για την επικοινωνία με

Registry Service	<p>κάποιο στιγμιότυπο ενός service. Ένα end-point καθορίζει μια συγκεκριμένη τοποθεσία πρόσβασης ενός service χρησιμοποιώντας ένα συγκεκριμένο πρωτόκολλο και μορφή δεδομένων.</p> <p>Επίσημη, κεντρική αποθήκη πληροφοριών.</p>
Service Ανακάλυψης	<p>Ένα service είναι μια αφηρημένη πηγή που αναπαριστά την ικανότητα εκπλήρωσης κάποιων διαδικασιών που σχηματίζουν συναφείς λειτουργίες από τη μεριά των οντοτήτων του παροχέα και του αιτούντα. Η υλοποίησή του γίνεται μέσω ενός agent-παροχέα.</p> <p>Δείτε επίσης και το Web Service.</p> <p>Το service ανακάλυψης είναι ένα service που διευκολύνει τους agents να ανακτήσουν τις περιγραφές των web services.</p>
SOAP	<p>Είναι το επίσημο σύνολο των συμβάσεων που διέπουν τους κανόνες μορφοποίησης και επεξεργασίας ενός μηνύματος SOAP. Οι συμβάσεις αυτές περιλαμβάνουν την αλληλεπίδραση ανάμεσα σε κόμβους SOAP που δημιουργούν και δέχονται τα μηνύματα SOAP για τον σκοπό της μεταφοράς πληροφορίας κατά μήκος του μονοπατιού μηνυμάτων SOAP. Η τρέχουσά του έκδοση είναι η 1.2.</p>
UDDI	<p>Το UDDI ή όπως ονομάζεται εν συντομία το Universal Description, Discovery and Integration είναι ένα πρότυπο δημιουργίας καταλόγων στον οποίο καταχωρούνται τα Web Services και οι υπηρεσίες τις οποίες παρέχουν. Η τρέχουσα του έκδοση είναι η 3.0.</p>
Web Service	<p>Ένα Web service είναι ένα σύστημα λογισμικού σχεδιασμένο για να υποστηρίζει δια-λειτουργική αλληλεπίδραση από μηχανή σε μηχανή πάνω σε ένα δίκτυο. Έχει ένα περιβάλλον που περιγράφεται σε μια μορφή επεξεργάσιμη από μηχανή, το WSDL. Τα άλλα συστήματα αλληλεπιδρούν με το Web service με τρόπο που καθορίζεται από την περιγραφή του χρησιμοποιώντας μηνύματα SOAP, που μεταφέρονται συνήθως μέσω HTTP με μια αρχικοποίηση XML σε συνδυασμό και με άλλα Web-σχετιζόμενα στάνταρ.</p>
WSD	<p>Το WSD είναι μια προδιαγραφή, επεξεργάσιμη από τη μηχανή, του περιβάλλοντος του web service και είναι γραμμένη σε WSDL.</p>
WSDL	<p>Πρόκειται για τη γλώσσα περιγραφής Web Services (Web Service Description Language). Είναι μια</p>

	<p>συλλογή από end-points. Η τρέχουσά της έκδοση είναι η 2.0.</p>
XML	<p>Η XML (Extensible Markup Language) είναι μια απλή και πολύ ευέλικτη μορφή κειμένου που προέκυψε από το SGML (ISO 8879). Αρχικά σχεδιάστηκε για να ικανοποιήσει τις απαιτήσεις των ηλεκτρονικών εκδόσεων μεγάλης κλίμακας, παίζει όμως και έναν αυξανόμενο σημαντικό ρόλο στην ανταλλαγή ευρείας ποικιλίας δεδομένων πάνω στο διαδίκτυο και αλλού.</p>
Ανακάλυψη	<p>Είναι η πράξη του εντοπισμού μιας περιγραφής επεξεργάσιμης από μηχανή ενός web service το οποίο ήταν άγνωστο μέχρι τότε και που συμμορφώνεται σε κάποια λειτουργικά κριτήρια. Περιλαμβάνει το ταίριασμα λειτουργικών και άλλων κριτηρίων με τις περιγραφές των web services με σκοπό τον εντοπισμό της κατάλληλης πηγής που διαθέτει το κατάλληλο web service.</p>
Αρχιτεκτονική	<p>1. Η αρχιτεκτονική ενός προγράμματος ή ενός υπολογιστικού συστήματος είναι η δομή του συστήματος. Η δομή αυτή περιλαμβάνει τα τμήματα λογισμικού, τις εξωτερικές ιδιότητες αυτών των τμημάτων, τις μεταξύ τους σχέσεις και τους περιορισμούς στη χρήση τους.</p> <p>2. Είναι μια αφηρημένη περιγραφή των τμημάτων ενός συστήματος λογισμικού κατά τη διάρκεια μιας φάσης λειτουργίας του. Ένα σύστημα μπορεί να αποτελείται από πολλά επίπεδα αφάιρεσης και πολλές φάσεις λειτουργίας, καθεμιά με τη δική της αρχιτεκτονική.</p>
Ιδιώτης ΈΗ Εταιρεία	<p>Μπορεί να είναι οι ιδιοκτήτες ενός agent που παρέχουν ή αναζητούν web services.</p>
Κόμβος SOAP	<p>Η ενσάρκωση της λογικής διαδικασίας που είναι απαραίτητη για τη μετάδοση, τη λήψη, την επεξεργασία και την αναμετάδοση μηνυμάτων SOAP σύμφωνα με το σύνολο των συμβάσεων που ορίζονται. Ένας κόμβος SOAP είναι υπεύθυνος για την επιβολή των κανόνων που διέπουν την ανταλλαγή μηνυμάτων SOAP. Έχει πρόσβαση στα services που παρέχονται από τα συγκεκριμένα πρωτόκολλα μέσω ενός ή περισσότερων bindings.</p>
Μήνυμα	<p>Ένα μήνυμα είναι η βασική μονάδα δεδομένων που στέλνεται από έναν agent σε έναν άλλο.</p>

Μήνυμα SOAP	<p>Η βασική μονάδα επικοινωνίας μέσω ενός web service και ενός αιτούντα. Είναι τα δεδομένα που διακινούνται από και προς το web service σαν μια, ενιαία μετάδοση.</p> <p>Δείτε επίσης και στα μηνύματα SOAP.</p> <p>Η βασική μονάδα επικοινωνίας μεταξύ των κόμβων SOAP.</p>
Μονοπάτι Μηνυμάτων SOAP	<p>Είναι το σύνολο των κόμβων SOAP μέσα από τους οποίους περνάει ένα μήνυμα SOAP. Περιλαμβάνει τον αρχικό αποστολέα SOAP, κανέναν ή περισσότερους διαμεσολαβητές και έναν τελικό SOAP αποδέκτη.</p>
Οντότητα Αιτών	<p>Ο ιδιώτης ή η εταιρία που επιθυμεί να χρησιμοποιήσει ένα web service μιας οντότητας παροχέα.</p>
Οντότητα Παροχέας Περιγραφή Service	<p>Ο ιδιώτης ή η εταιρία που παρέχει ένα web service.</p> <p>Η περιγραφή ενός service είναι ένα σύνολο αρχείων που περιγράφουν το περιβάλλον αλληλεπίδρασης και τη σημασιολογία ενός service.</p>
Πρωτόκολλο	<p>Ένα σύνολο επίσημων κανόνων που περιγράφουν το πως γίνεται η μετάδοση των δεδομένων, κυρίως πάνω σε ένα δίκτυο. Χαμηλού επιπέδου πρωτόκολλα ορίζουν τα ηλεκτρικά και φυσικά στάνταρ που πρέπει να τηρηθούν, τη σειρά των bit και byte, την μετάδοση, τον εντοπισμό και τη διόρθωση λαθών μιας ροής από bit.</p> <p>Υψηλού επιπέδου πρωτόκολλα χειρίζονται τη μορφοποίηση των δεδομένων, συμπεριλαμβανομένης και της σύνταξης των μηνυμάτων, το διάλογο μεταξύ τερματικού και υπολογιστή, τα σύνολα χαρακτηριστών, την ακολουθία των μηνυμάτων, κλπ.</p>
Σημασιολογία	<p>Η σημασιολογία ενός service είναι η συμπεριφορά που αυτό αναμένεται να έχει όταν αλληλεπιδρά με κάποιο άλλο service. Εκφράζει ένα συμβόλαιο ανάμεσα στην οντότητα παροχέα και την οντότητα αιτούντα καθώς επίσης και το αποτέλεσμα αυτής της συνεργασίας. Η σημασιολογία ενός service μπορεί να περιγράφεται επίσημα σε μια μορφή αναγνώσιμη από μηχανή, μπορεί να αναγνωρίζεται αλλά να μην ορίζεται επίσημα ή μπορεί να ορίζεται ανεπίσημα μέσω μιας συμφωνίας ανάμεσα στις δύο οντότητες.</p>

Copyright © Ευάγγελος Σακκίopoulos, 2005
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πατρών.