

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εποπτεία Επίδοσης σε Συστήματα Ψηφιακών Βιβλιοθηκών

Σωτήριος Τερζής

Μεταπτυχιακή Εργασία

Ηράκλειο, Σεπτέμβριος 1997

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΠΟΠΤΕΙΑ ΕΠΙΔΟΣΗΣ ΣΕ ΣΥΣΤΗΜΑΤΑ ΨΗΦΙΑΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εργασία που υποβλήθηκε από τον
Σωτήριο Τερζή
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Σωτήριος Τερζής
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Χρήστος Νικολάου, Αναπληρωτής Καθηγητής, Επόπτης

Σαράντος Καπιδάκης, Επισκέπτης Καθηγητής, Επιβλέπων Ερευνητής

Καίτη Χούστη, Αναπληρώτρια Καθηγήτρια, Μέλος

Απόστολος Τραγανίτης, Αναπληρωτής Καθηγητής, Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Σεπτέμβριος 1997

Εποπτεία Επίδοσης σε Συστήματα Ψηφιακών Βιβλιοθηκών

Σωτήριος Τερζής

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

ΠΕΡΙΛΗΨΗ

Οι ψηφιακές βιβλιοθήκες είναι συστήματα που παρέχουν εύκολη πρόσβαση και διαχείριση σε συλλογές αντικειμένων (κειμένου, ήχου, εικόνες και βίντεο) σε ψηφιακή μορφή, διασκορπισμένων στο παγκόσμιο δίκτυο (**internet**). Η διάδοση των συστημάτων αυτών τα τελευταία χρόνια έφερε στο προσκήνιο τα θέματα επίδοσης τους. Τα συστήματα αυτά χαρακτηρίζονται από την κατανομή τους σε έναν συνεχώς αυξανόμενο αριθμό ετερογενών κόμβων και την κατασκευή τους με την ολοκλήρωση προϋπαρχόντων συστημάτων πάνω από τον παγκόσμιο δικτυακό ιστό (**World Wide Web**), γεγονός που συνεπάγεται σημαντικές διακυμάνσεις στην απόδοσή τους. Τα χαρακτηριστικά αυτά επιβάλλουν μια νέα οπτική στα θέματα εποπτείας επίδοσης.

Στην παρούσα εργασία σχεδιάστηκε και υλοποιήθηκε ένας επόπτης επίδοσης για συστήματα ψηφιακής βιβλιοθήκης. Ο επόπτης αυτός αποτελεί την βάση για ένα πλαίσιο μελέτης της επίδοσης των συστημάτων αυτών και υποστήριξης μηχανισμών όπως η εγγύηση επιπέδων ποιότητας υπηρεσίας (**quality of service levels**), η δυναμική αναπροσαρμογή των χρονικών περιθωρίων (**timeouts**) και η εξισορρόπηση φόρτου (**load balancing**). Στην συνέχεια, για να γίνουν πιο σαφείς οι δυνατότητες που παρέχει ο επόπτης επίδοσης, τον χρησιμοποιήσαμε στην μελέτη επίδοσης του συστήματος ψηφιακής βιβλιοθήκης *NCSTR-L-Dienst* καταλήγοντας σε μια σειρά από προτάσεις για την βελτίωση της αρχιτεκτονικής του συστήματος. Τέλος, σχεδιάστηκαν ένας μηχανισμός για την δυναμική αναπροσαρμογή των χρονικών περιθωρίων και δοκιμάστηκαν κάποιες ενδεικτικές πολιτικές εξισορρόπησης φόρτου στο σύστημα *NCSTR-L-Dienst*.

Επόπτης: Χρήστος Νικολάου

Επιβλέπων Ερευνητής: Σαράντος Καπιδάκης

Performance Monitoring in Digital Library Systems

Sotirios Terzis

Master of Science Thesis

Department of Computer Science
University of Crete

ABSTRACT

Digital Libraries are systems for the management and transparent access of digital objects (text, sounds, images and videos) scattered throughout the **Internet**. In the recent years, the wide spread of those systems leads to an increased consideration of performance issues. What characterises digital library systems is their distribution to tens or hundreds of heterogeneous nodes and the fact that are integration of existing systems over the **World Wide Web**. These characteristics lead to significant performance variations and impose a new approach to performance monitoring.

In this work, we designed and implemented a performance monitor for digital library systems. This monitor provides the necessary framework for performance studies and support of mechanisms for quality of service guarantees, dynamic adaptation of timeout and load balancing in digital library systems. As an example of its abilities, the monitor was used for a performance study of the *NCSTR-L-Dienst* system, which led to the suggestion of various improvements in system's architecture. Finally, we designed a mechanism for dynamic adaptation of timeouts and tested some strategies for load balancing for the *NCSTR-L-Dienst* system.

Supervisor: Christos Nikolaou

Supervising Researcher: Sarantos Kapidakis

Ευχαριστίες

Ευχαριστίες

Η παρούσα εργασία δεν θα είχε πραγματοποιηθεί χωρίς την σημαντική βοήθεια ορισμένων ανθρώπων στα διάφορα στάδια της πορείας της, τους οποίους και θα ήθελα να ευχαριστήσω.

Ο επόπτης καθηγητής κ. Χρήστος Νικολάου μου έδωσε την δυνατότητα να ασχοληθώ με τα θέματα των ψηφιακών βιβλιοθηκών καθοδηγώντας με στις βασικές επιλογές της εργασίας μου. Τον ευχαριστώ, ακόμα, και για την δυνατότητα που μου έδωσε να παρουσιάσω την εργασία μου σε άλλους ερευνητές με την συμμετοχή μου σε Ευρωπαϊκά και Διεθνή Συνέδρια. Ο κ. Σαράντος Καπιδάκης, ο επιβλέπων ερευνητής της εργασίας μου, υπήρξε πολύτιμος σύμβουλος σε όλα τα στάδια της εργασίας και βοήθησε σημαντικά στην λύση των πρακτικών προβλημάτων. Η κα. Καίτη Χούστη και ο κ. Απόστολος Τραγανίτης, μέλη της εξεταστικής επιτροπής, έκαναν εκτενή και ουσιώδη σχόλια, συμβάλλοντας στην βελτίωση της παρουσίας του θέματος.

Ακόμη, σημαντικός ήταν ο ρόλος του Σταύρου Παπαδάκη του οποίου οι γνώσεις σε *Perl* αποτέλεσαν πολύτιμη βοήθεια στα διάφορα στάδια της υλοποίησης και ιδιαίτερα στην μεταφορά της αρχικής υλοποίησης του επόπτη σε *Perl*. Ο Αντώνης Σιδηρόπουλος ήταν πολύτιμος βοηθός ως διαχειριστής του *Dienst* εξυπηρετητή, του Ινστιτούτου Πληροφορικής (ΙΠ) του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΤΕ). Ο Αντώνης Χατζησταματίου βοήθησε σημαντικά κάνοντας την πρώτη υλοποίηση του επόπτη, καθώς και με την σχεδίαση και υλοποίηση ενός γραφικού περιβάλλοντος για την απεικόνιση των μετρήσεων που διεξάγει ο επόπτης επίδοσης. Ο Γιώργος Σαπουντζής βοήθησε με την σχεδίαση και υλοποίηση ενός γραφικού περιβάλλοντος για την παρακολούθηση της δρομολόγησης των ερωτήσεων του συστήματος *NCSTRL-Dienst*. Ο Jakka Sairamesh υπήρξε σημαντικός βοηθός στις πρώτες φάσεις της σχεδίασης και ιδιαίτερα στα θέματα ανάλυσης επίδοσης, ειδικώς του συστήματος *Dienst*, καθώς και με τα εποικοδομητικά του σχόλια στις διάφορες φάσεις της σχεδίασης. Ο Μανόλης Μαραζάκης αποτέλεσε πάντα πολύτιμη πηγή σχολίων για τα διάφορα στάδια της σχεδίασης, της υλοποίησης καθώς και για την συγκέντρωση της σχετικής βιβλιογραφίας. Ακόμα, μαζί με την Πηνελόπη Κωνσταντά - Φανουράκη αποτέλεσε σημαντική πηγή ενθάρρυνσης. Τέλος, η στοιχειοθεσία του παρόντος κείμενου οφείλεται στις γνώσεις του Σταύρου Παπαδάκη για το *Microsoft Office*.

Επίσης, θα ήθελα να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική και οικονομική υποστήριξη που μου παρείχαν κατά την διάρκεια των σπουδών μου.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	i
ABSTRACT.....	ii
Ευχαριστίες.....	iii
Περιεχόμενα.....	v
Κατάλογος Πινάκων, Γραφημάτων και Μαθηματικών Τύπων	ix
Κατάλογος Σχημάτων.....	xi
Κεφάλαιο 1. Εισαγωγή.	1
1.1. Το θέμα της εργασίας.	2
1.2. Γιατί ένας δικός μας επόπτης επίδοσης;	3
1.3. Το σύστημα <i>NCSTRL - Dienst</i>	4
1.4. Συνοπτικά συμπεράσματα.....	4
1.5. Η οργάνωση του κειμένου.	5
Κεφάλαιο 2. Σχετικές Εργασίες.	7
Κεφάλαιο 3. Ανάλυση Επίδοσης και Αρχιτεκτονική Επόπτη Επίδοσης.	11
3.1. Ο ορισμός των παραμέτρων επίδοσης.....	11
3.2. Η αρχιτεκτονική μέτρησης και αποθήκευσης των παραμέτρων επίδοσης. ..	12
3.2.1. Ο τρόπος μέτρησης των παραμέτρων επίδοσης.....	12
3.2.2. Η αρχιτεκτονική του επόπτη επίδοσης.	13
3.2.3. Θέματα Υλοποίησης.	15
Κεφάλαιο 4. Εφαρμογές της Εποπτείας Επίδοσης.....	17
4.1. Οπτικοποίηση παραμέτρων επίδοσης.....	17
4.2. Εγγύηση ποιότητας υπηρεσίας.	18
4.3. Εξισορρόπηση φόρτου.....	19

4.4. Δυναμική αναπροσαρμογή των χρονικών περιθωρίων.	20
Κεφάλαιο 5. Το Σύστημα NCSTRL – Dienst.	23
5.1. Η βασική αρχιτεκτονική του συστήματος <i>Dienst</i>	23
5.1.1. Το μοντέλο εγγράφων.	24
5.1.2. Η αρχιτεκτονική του εξυπηρετητή.....	24
5.1.3. Η διεπιφάνεια χρήσης του συστήματος.	26
5.2. Οι <i>NCSTRL</i> μετατροπές του συστήματος.....	26
5.2.1. Η αρχιτεκτονική των εξυπηρετητών περιοχής.....	27
5.2.2. Το σύστημα των αναγνωριστικών εγγράφων.....	28
Κεφάλαιο 6. Ανάλυση Επίδοσης του Συστήματος <i>NCSTRL – Dienst</i>.	29
6.1. Μοντελοποίηση μιας αίτησης αναζήτησης.....	29
6.1.1. Τμήματα υποβολής αίτησης.....	31
6.1.2. Εκκίνηση παράλληλης αναζήτησης.	31
6.1.3. Επεξεργασία αίτησης στον τοπικό κόμβο.....	32
6.1.4. Επεξεργασία αίτησης στον μακρινό κόμβο 1.....	33
6.1.5. Επεξεργασία αίτησης στον μακρινό κόμβο 2.....	33
6.2. Μοντέλο συστήματος με εξυπηρετητή υποστήριξης.	34
6.3. Αίτηση με ενημέρωση των καταλόγων.	35
Κεφάλαιο 7. Εφαρμογές της Εποπτείας Επίδοσης στο Σύστημα <i>NCSTRL-Dienst</i>.	39
7.1. Αποτελέσματα προκαταρκτικής μελέτης επίδοσης του συστήματος <i>NCSTRL-Dienst</i>	40
7.1.1. Πλατφόρμα πειραμάτων.....	40
7.1.2. Πειραματικά αποτελέσματα.....	40
7.2. Επικοινωνία μεταξύ των εξυπηρετητών και χρονικά περιθώρια.	43
7.2.1. Τροποποιημένο πρωτόκολλο επικοινωνίας του συστήματος <i>NCSTRL-Dienst</i>	44
7.3. Επέκταση του πρωτοκόλλου <i>Dienst</i>	46
7.4. Εξισορρόπηση φόρτου.....	47
7.4.1. Αλγόριθμοι δυναμικής δρομολόγησης ερωτήσεων.....	48
7.4.2. Θέματα υλοποίησης.	50

7.4.3. Πειραματική μελέτη.	50
Πλατφόρμα Πειραμάτων.....	51
Ανάλυση των Αρχείων Καταγραφής του Cornell.	51
Πειραματικά Αποτελέσματα.....	53
 Κεφάλαιο 8. Συμπεράσματα και Μελλοντικές Κατευθύνσεις.	61
 Βιβλιογραφία	63
 Παράρτημα 1. Το Πρωτόκολλο Επικοινωνίας του Επόπτη Επίδοσης.....	67
Επικοινωνία με τις διαδικασίες μετρήσεων.	67
Επικοινωνία με την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης.	70
Επικοινωνία με τον «έξω κόσμο».	71
 Παράρτημα 2. Αρχεία Καταγραφής Μετρήσεων.	73
 Παράρτημα 3. Παράδειγμα Χρήσης του Συστήματος NCSTRL-Dienst.	77
 Παράρτημα 4. Το Πρωτόκολλο <i>Dienst</i>	79
 Παράρτημα 5. Οι Παράμετροι Επίδοσης του Συστήματος <i>NCSTRL-Dienst</i>	81
 Παράρτημα 6. Προσθήκες και Αλλαγές στον Κώδικα του Συστήματος <i>NCSTRL-Dienst</i>	83
 Παράρτημα 7. Λεξικό Τεχνικών Όρων.	85

Κατάλογος Πινάκων, Γραφημάτων και Μαθηματικών Τύπων

(1) ΔΥΝΑΜΙΚΑ ΠΡΟΣΑΡΜΟΖΟΜΕΝΟ ΧΡΟΝΙΚΟ ΠΕΡΙΘΩΡΙΟ.....	21
(2) ΑΝΙΣΟΤΗΤΑ ΤCΗΕΒΙΣΕV.....	21
(3) ΧΡΟΝΟΣ ΤΟΠΙΚΗΣ ΈΝΑΝΤΙ ΜΑΚΡΙΝΗΣ ΑΝΑΖΗΤΗΣΗΣ.....	41
(4) ΜΕΓΕΘΟΣ ΜΝΗΜΗΣ ΚΑΙ ΧΡΟΝΟΣ ΑΝΑΖΗΤΗΣΗΣ.....	41
(5) ΜΕΓΕΘΟΣ ΒΑΣΗΣ ΚΑΙ ΧΡΟΝΟΣ ΑΝΑΖΗΤΗΣΗΣ.....	41
(6) ΧΡΟΝΙΚΑ ΠΕΡΙΘΩΡΙΑ ΚΑΙ ΧΡΟΝΟΣ ΕΞΥΠΗΡΕΤΗΣΗΣ ΑΙΤΗΣΗΣ.....	42
(7) ΜΕΓΕΘΟΣ ΒΑΣΗΣ ΚΑΙ ΚΑΘΥΣΤΕΡΗΣΗ ΣΤΟ ΔΙΚΤΥΟ.....	43
(8) ΧΡΟΝΙΚΟ ΠΕΡΙΘΩΡΙΟ ΑΝΑΖΗΤΗΣΗΣ ΣΤΟΝ ΕΞΥΠΗΡΕΤΗΤΗ ΥΠΟΣΤΗΡΙΞΗΣ,.....	45
(9) ΒΕΛΤΙΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΔΥΝΑΜΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ.....	49
(10) ΤΡΟΠΟΠΟΙΗΜΕΝΟΣ ΑΛΓΟΡΙΘΜΟΣ ΔΥΝΑΜΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ.....	49
(11) ΨΕΥΔΟΚΑΘΥΣΤΕΡΗΣΗ ΔΙΚΤΥΟΥ.....	51
(12) ΑΙΤΗΣΕΙΣ ΑΝΑ ΚΑΤΗΓΟΡΙΑ ΥΠΗΡΕΣΙΑΣ.....	52
(13) ΕΙΔΗ ΑΙΤΗΣΕΩΝ ΔΙΕΠΙΦΑΝΕΙΑΣ ΧΡΗΣΗΣ (UI).....	52
(14) ΑΡΙΘΜΟΣ ΛΕΞΕΩΝ ΚΛΕΙΔΙΩΝ ΣΤΙΣ ΑΙΤΗΣΕΙΣ QUERYNF.....	53
(15) ΔΡΟΜΟΛΟΓΗΣΗ ΕΡΩΤΗΣΕΩΝ ΠΡΩΤΗΣ ΠΑΡΑΛΛΑΓΗΣ 1.....	54
(16) ΠΙΝΑΚΑΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΠΡΩΤΗΣ ΠΑΡΑΛΛΑΓΗΣ 1.....	55
(17) ΔΡΟΜΟΛΟΓΗΣΗ ΕΡΩΤΗΣΕΩΝ ΠΡΩΤΗΣ ΠΑΡΑΛΛΑΓΗΣ 2.....	55
(18) ΠΙΝΑΚΑΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΠΡΩΤΗΣ ΠΑΡΑΛΛΑΓΗΣ 2.....	56
(19) ΔΡΟΜΟΛΟΓΗΣΗ ΕΡΩΤΗΣΕΩΝ ΔΕΥΤΕΡΗΣ ΠΑΡΑΛΛΑΓΗΣ 1.....	56
(20) ΠΙΝΑΚΑΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΔΕΥΤΕΡΗΣ ΠΑΡΑΛΛΑΓΗΣ 1.....	57
(21) ΔΡΟΜΟΛΟΓΗΣΗ ΕΡΩΤΗΣΕΩΝ ΔΕΥΤΕΡΗΣ ΠΑΡΑΛΛΑΓΗΣ 2.....	57
(22) ΠΙΝΑΚΑΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΔΕΥΤΕΡΗΣ ΠΑΡΑΛΛΑΓΗΣ 2.....	58
(23) ΕΞΕΛΙΞΗ ΤΩΝ ΕΠΙΛΟΓΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΗΝ ΔΕΥΤΕΡΗ ΠΑΡΑΛΛΑΓΗ ΠΕΙΡΑΜΑ 1 ΓΙΑ ΤΗΝ ΑΡΧΗ ΈΚΔΟΣΗΣ Α1.....	58
(24) ΕΞΕΛΙΞΗ ΤΩΝ ΕΠΙΛΟΓΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΗΝ ΔΕΥΤΕΡΗ ΠΑΡΑΛΛΑΓΗ ΠΕΙΡΑΜΑ 2 ΓΙΑ ΤΗΝ ΑΡΧΗ ΈΚΔΟΣΗΣ Α9.....	59
(25) ΥΠΗΡΕΣΙΕΣ ΚΑΙ ΟΝΟΜΑΤΑ ΑΙΤΗΣΕΩΝ ΤΟΥΣ.....	80

Κατάλογος Σχημάτων

ΣΧΗΜΑ 1: ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΛΥΣΗΣ ΕΠΙΔΟΣΗΣ.....	12
ΣΧΗΜΑ 2: ΜΟΝΤΕΛΟ ΛΕΙΤΟΥΡΓΙΑΣ ΕΠΟΠΤΗ ΕΠΙΔΟΣΗΣ.....	12
ΣΧΗΜΑ 3: ΜΗΧΑΝΙΣΜΟΣ ΜΕΤΡΗΣΗΣ ΠΑΡΑΜΕΤΡΩΝ ΕΠΙΔΟΣΗΣ.....	13
ΣΧΗΜΑ 4: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΠΟΠΤΗ ΕΠΙΔΟΣΗΣ.....	14
ΣΧΗΜΑ 5: ΟΠΤΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ ΕΠΙΔΟΣΗΣ.....	17
ΣΧΗΜΑ 6: ΕΡΓΑΛΕΙΟ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΕΝ ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΕΠΙΔΟΣΗΣ.....	18
ΣΧΗΜΑ 7: ΟΠΤΙΚΟΠΟΙΗΣΗ ΕΝ ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΕΠΙΛΟΓΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΕΞΙΣΟΡΡΟΠΗΣΗΣ ΦΟΡΤΟΥ.....	20
ΣΧΗΜΑ 8: ΤΟ ΜΟΝΤΕΛΟ ΕΓΓΡΑΦΩΝ ΤΟΥ DIENST.....	24
ΣΧΗΜΑ 9: Η ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΤΩΝ ΕΞΑΡΤΗΜΑΤΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ DIENST.....	25
ΣΧΗΜΑ 10: ΠΑΡΑΔΕΙΓΜΑ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ DIENST.....	29
ΣΧΗΜΑ 11: ΡΟΗ ΕΞΥΠΗΡΕΤΗΣΗΣ ΜΙΑΣ ΕΡΩΤΗΣΗΣ ΣΤΟ ΣΥΣΤΗΜΑ DIENST.....	30
ΣΧΗΜΑ 12: ΡΟΗ ΕΞΥΠΗΡΕΤΗΣΗΣ ΑΙΤΗΣΗΣ ΜΕ ΧΡΗΣΗ ΕΞΥΠΗΡΕΤΗΤΗ ΥΠΟΣΤΗΡΙΞΗΣ.....	34
ΣΧΗΜΑ 13: ΡΟΗ ΕΞΥΠΗΡΕΤΗΣΗΣ ΑΙΤΗΣΗΣ ΜΕ ΕΝΗΜΕΡΩΣΗ ΤΩΝ ΚΑΤΑΛΟΓΩΝ.....	35
ΣΧΗΜΑ 14: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑΣ DIENST – ΕΠΟΠΤΗ ΕΠΙΔΟΣΗΣ.....	39
ΣΧΗΜΑ 15: ΤΡΕΧΟΥΣΑ ΧΡΗΣΗ ΤΩΝ ΧΡΟΝΙΚΩΝ ΠΕΡΙΘΩΡΙΩΝ ΣΤΟ ΣΥΣΤΗΜΑ NCSTR-L-DIENST.....	45
ΣΧΗΜΑ 16: ΠΡΟΤΕΙΝΟΜΕΝΗ ΧΡΗΣΗ ΤΩΝ ΧΡΟΝΙΚΩΝ ΠΕΡΙΘΩΡΙΩΝ ΣΤΟ ΣΥΣΤΗΜΑ NCSTR-L-DIENST.....	46
ΣΧΗΜΑ 17: ΣΕΛΙΔΑ ΜΕ ΤΙΣ ΠΑΡΑΜΕΤΡΟΥΣ ΕΠΙΔΟΣΗΣ ΕΝΟΣ ΕΞΥΠΗΡΕΤΗΤΗ.....	46
ΣΧΗΜΑ 18: ΣΤΑΤΙΚΗ ΔΡΟΜΟΛΟΓΗΣΗ ΤΩΝ ΕΡΩΤΗΣΕΩΝ ΣΤΟ ΣΥΣΤΗΜΑ NCSTR-L-DIENST.....	48
ΣΧΗΜΑ 19: ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ NCSTR-L-DIENST.....	77
ΣΧΗΜΑ 20: ΑΝΑΖΗΤΗΣΗ ΜΕ ΒΑΣΗ ΤΑ ΠΕΔΙΑ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΓΓΡΑΦΗΣ.....	77
ΣΧΗΜΑ 21: ΣΕΛΙΔΑ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΑΝΑΖΗΤΗΣΗΣ.....	78
ΣΧΗΜΑ 22: ΣΕΛΙΔΑ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΓΓΡΑΦΗΣ ΕΓΓΡΑΦΟΥ.....	78

Στους γονείς μου, Γιάννη και Μαρία.

Κεφάλαιο 1. Εισαγωγή.

Η εκρηκτική ανάπτυξη της παγκόσμιας δικτύωσης (**internet**) τα τελευταία χρόνια δίνει την δυνατότητα σε όλο και μεγαλύτερο αριθμό ανθρώπων από όλο τον κόσμο να έχει πρόσβαση σε απεριόριστο όγκο ψηφιοποιημένων πληροφοριών. Ο τεράστιος αυτός όγκος πληροφοριών, όμως, είναι στο μεγαλύτερο μέρος του ανοργάνωτος, διάσπαρτος σε διάφορα μέρη (**sites**) και με διάφορες μορφές (**formats**). Το γεγονός αυτό δυσχεραίνει ιδιαίτερα την διαχείριση της πληροφορίας, ενώ οι χρήστες έχοντας συχνά να αντιμετωπίσουν μια "Βαβέλ" από αρχιτεκτονικές και μορφές εγγράφων, αδυνατούν να αξιοποιήσουν την πληροφορία αυτή. Για την αντιμετώπιση αυτών των προβλημάτων προτάθηκαν ως λύση οι ψηφιακές βιβλιοθήκες.

Οι ψηφιακές βιβλιοθήκες είναι, ουσιαστικά, αρχιτεκτονικές για την εύκολη διαχείριση και πρόσβαση σε οργανωμένες συλλογές ψηφιοποιημένων αντικειμένων με την μορφή κειμένου, εικόνας, μουσικής, ταινιών κτλ. Οι αρχιτεκτονικές αυτές βασίζονται σε μεγάλο βαθμό στο μοντέλο της παραδοσιακής βιβλιοθήκης με τα βιβλία. Υπάρχουν, δηλαδή, μια σειρά από συλλογές ψηφιοποιημένων αντικειμένων ("βιβλίων"). Η κάθε συλλογή αποτελείται από αντικείμενα με κάποια λογική συσχέτιση, π.χ. ίδιου τύπου ή από τον ίδιο παραγωγό - εκδότη. Οι συλλογές αυτές είναι αποθηκευμένες σε κάποιο χώρο (**repository**), ο οποίος είναι συχνά κατακευματισμένος σε μεγάλο αριθμό από υπολογιστές στο δίκτυο. Στον χρήστη της βιβλιοθήκης παρέχεται η δυνατότητα να ψάξει για κάποιο αντικείμενο, που να ικανοποιεί μια σειρά από κριτήρια αναζήτησης, καθώς και να "δει" κάποιο αντικείμενο. Στους διαχειριστές της βιβλιοθήκης παρέχονται μια σειρά από εργαλεία που διευκολύνουν την διαχείριση των συλλογών.

Από τα παραπάνω, είναι σαφές πως οι ψηφιακές βιβλιοθήκες είναι απαραίτητες για την αξιοποίηση της διαθέσιμης πληροφορίας, μέσω των μηχανισμών διαχείρισης και αναζήτησης που παρέχουν, και το άνοιγμα του παγκόσμιου δικτύου στους απλούς χρήστες, μέσω των απλοποιημένων μεθόδων αλληλεπίδρασης που προσφέρουν. Η σπουδαιότητα του θέματος των ψηφιακών βιβλιοθηκών οδήγησε, τα τελευταία χρόνια, σημαντικό μέρος των ερευνητικών προσπαθειών στην επιστήμη των υπολογιστών να επικεντρωθεί προς αυτές. Είναι χαρακτηριστικό ότι τόσο στις Ηνωμένες Πολιτείες, όσο και στην Ευρώπη αποτελεί κεντρικό στόχο οργανωμένων ερευνητικών πρωτοβουλιών. Τέτοιες πρωτοβουλίες είναι η αμερικανική Εθνική Ερευνητική Πρωτοβουλία (**National Research Initiative - NRI**) για ψηφιακές βιβλιοθήκες του αμερικανικού Εθνικού Ιδρύματος Ερευνών (**National Science Foundation - NSF**) και της υπηρεσίας Προηγμένων Ερευνητικών Προγραμμάτων του αμερικανικού υπουργείου Άμυνας (**Defense Advanced Research Project Agency - DARPA**), και η ερευνητική πρωτοβουλία για ψηφιακές βιβλιοθήκες του Ευρωπαϊκού Ερευνητικού Συνδέσμου για την Πληροφορική και τα Μαθηματικά (**European Research Consortium for Informatics and Mathematics - ERCIM**).

Στα πλαίσια αυτά η Ομάδα Παράλληλων και Κατανεμημένων Συστημάτων – Πλειάδες, του Ινστιτούτου Πληροφορικής (ΙΠ), του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΤΕ), εξετάει τις ερευνητικές τις δραστηριότητες στο θέμα των ψηφιακών βιβλιοθηκών. Μέρος της δραστηριότητας αυτής αποτελεί και η παρούσα εργασία.

1.1. Το θέμα της εργασίας.

Τα συστήματα ψηφιακών βιβλιοθηκών είναι ουσιαστικά πλατφόρμες που δίνουν την εικόνα ενιαίου χώρου σε μια συλλογή ετερογενών πηγών πληροφορίας, καθώς και μηχανισμούς για το ψάξιμο, την ανάκληση και την διαχείριση της πληροφορίας. Τα συστήματα αυτά συνήθως βασίζονται σε προϋπάρχουσες συλλογές πληροφοριών, και στους μηχανισμούς που παρέχονται από τον παγκόσμιο δικτυακό ιστό (**World Wide Web**).

Κύρια χαρακτηριστικά των συστημάτων αυτών είναι, πρώτον, ο μεγάλος αριθμός των κόμβων στους οποίους είναι κατανεμημένα και, δεύτερον, η μεγάλη διασπορά των κόμβων αυτών στο παγκόσμιο δίκτυο. Τα χαρακτηριστικά αυτά, σε συνδυασμό με το γεγονός ότι σε πολλές περιπτώσεις οι κόμβοι και κυρίως το διασυνδεδετικό δίκτυο δεν είναι αφιερωμένα αποκλειστικά στο σύστημα ψηφιακής βιβλιοθήκης αλλά είναι κοινόχρηστα, έχουν ως αποτέλεσμα η κατάσταση των συστημάτων αυτών να είναι ιδιαίτερα ευμετάβλητη. Από την άλλη, ο απλός χρήστης των συστημάτων αυτών δεν κατανοεί τις ιδιαιτερότητες αυτές και θεωρεί την αστάθεια στο χρόνο απόκρισης των συστημάτων βιβλιοθήκης ουσιαστικό μειονέκτημα. Συνεπώς, είναι αναγκαίο για την καλύτερη εξυπηρέτηση των χρηστών αλλά και την καλύτερη διαχείριση των συστημάτων να έχουμε την δυνατότητα να παρακολουθούμε δυναμικά την επίδοσή τους. Την δυνατότητα αυτή την παρέχει ένας επόπτης επίδοσης.

Παρόλο που έχει γίνει σημαντική δουλειά στα θέματα επίδοσης των κατανεμημένων συστημάτων γενικότερα (βλέπε και το επόμενο κεφάλαιο), η εποπτεία επίδοσης στα συστήματα ψηφιακών βιβλιοθηκών είναι ένα νέο και ιδιαίτερα σημαντικό θέμα. Οι ιδιαιτερότητες των συστημάτων αυτών απαιτούν σε ορισμένα σημεία διαφορετική προσέγγιση και απλοποιήσεις. Το πρώτο βήμα προς την κατεύθυνση αυτή είναι η ανάλυση επίδοσης των ψηφιακών βιβλιοθηκών και η σχεδίαση και υλοποίηση ενός επόπτη επίδοσης. Ο επόπτης παρακολουθεί ουσιαστικά μια σειρά παραμέτρων επίδοσης, οι οποίες ορίζονται με βάση την ανάλυση επίδοσης του συστήματος ψηφιακής βιβλιοθήκης, και παρέχει πρόσβαση στις παραμέτρους αυτές. Με την χρήση των τιμών των παραμέτρων είναι δυνατόν να υλοποιήσουμε μια σειρά από πολιτικές που βελτιώνουν την απόδοση του συστήματος τόσο για τον χρήστη όσο και για τον διαχειριστή του. Τέτοιες υπηρεσίες είναι η εγγύηση συγκεκριμένων επιπέδων ποιότητας υπηρεσίας (**quality of service levels**), η εξισορρόπηση φόρτου (**load balancing**) και η δυναμική αναπροσαρμογή των χρονικών περιθωρίων (**dynamic adaptation of timeouts**).

Η εξισορρόπηση φόρτου είναι ουσιαστικά η δυναμική δρομολόγηση των υποερωτήσεων μιας αίτησης του χρήστη προς τους διάφορους εξυπηρετητές που απαρτίζουν το σύστημα ψηφιακής βιβλιοθήκης. Σκοπός της εξισορρόπησης φόρτου είναι να μπορεί το σύστημα να διαγνώσει τα σημεία καθυστέρησης και να τα αποφεύγει, όταν αυτό είναι δυνατό. Οι αποφάσεις δρομολόγησης βασίζονται στις τιμές των παραμέτρων που προσφέρει ο επόπτης επίδοσης.

Οι ψηφιακές βιβλιοθήκες ως ευρέως κατανεμημένα συστήματα πάνω από το παγκόσμιο δίκτυο, βασίζουν την ορθή λειτουργία τους σε μια σειρά από μέγιστα χρονικά περιθώρια (**timeouts**), που εξασφαλίζουν ότι το σύστημα δεν πρόκειται να κολλήσει εξαιτίας κάποιας ανωμαλίας στην λειτουργία του διασυνδεδετικού δικτύου. Η

δυναμική αναπροσαρμογή των χρονικών περιθωρίων εξασφαλίζει, πρώτον, ότι η αναμονή σε περίπτωση ανωμαλίας στο δίκτυο θα είναι η ελάχιστη δυνατή και, δεύτερον, ότι ακόμα και οι κόμβοι με κακή συνδεσιμότητα (κακής ποιότητας σύνδεση δικτύου) θα χρησιμοποιούνται, για να έχουμε όσο το δυνατό πιο πλήρεις απαντήσεις στις αιτήσεις του χρήστη. Οι τιμές των χρονικών περιθωρίων τίθενται με βάση τόσο τις παραμέτρους που μετρά ο επόπτης επίδοσης όσο και τις απαιτήσεις του χρήστη για την πληρότητα της απάντησης στην αίτηση του.

Η ποιότητα υπηρεσίας ορίζεται να είναι η διασφάλιση της ικανοποίησης των, βασικά χρονικών, απαιτήσεων του χρήστη. Ειδικότερα, ο χρήστης μπορεί να θέσει περιορισμούς στον χρόνο απόκρισης του συστήματος. Το σύστημα πρέπει να μπορεί να απαντήσει μια ερώτηση του χρήστη μέσα στα χρονικά πλαίσια που προσδιόρισε αυτός. Με την χρήση των μετρήσεων του επόπτη κάθε εξυπηρετητής του συστήματος είναι σε θέση να εκτιμήσει εάν το δοθέν χρονικό περιθώριο είναι αρκετό για την εξυπηρέτηση της υποερώτησης που του ανατέθηκε. Από την άλλη το σύστημα ψηφιακής βιβλιοθήκης έχοντας τις εκτιμήσεις για τους διάφορους εξυπηρετητές είναι σε θέση να προχωρήσει στην δρομολόγηση των υποερωτήσεων και να θέσει τις τιμές των χρονικών περιθωρίων κατά τρόπο που ο συνολικός χρόνος απόκρισης να ικανοποιεί τον χρονικό περιορισμό του χρήστη.

Στα πλαίσια της παρούσας εργασίας σχεδιάσαμε και υλοποιήσαμε έναν επόπτη επίδοσης. Ακόμα, παρουσιάζουμε μια σειρά από μηχανισμούς που υλοποιούνται πάνω από αυτόν, όπως η εξισορρόπηση του φόρτου, η δυναμική αναπροσαρμογή των χρονικών περιθωρίων και η εξασφάλιση ποιότητας υπηρεσίας. Στην συνέχεια προχωρήσαμε σε μια ανάλυση επίδοσης του συστήματος NCSTR-Dienst για τον προσδιορισμό των παραμέτρων επίδοσης του. Για την παρακολούθηση των παραμέτρων αυτών χρησιμοποιήθηκε ο προτεινόμενος επόπτης επίδοσης, για να ελέγξουμε την λειτουργία του. Με βάση τα στοιχεία που συγκεντρώσαμε για το σύστημα στηρίξαμε μια σειρά από προτάσεις για την βελτίωση της σχεδίασης και της υλοποίησης του. Τέλος, για να καταδειχτούν οι μηχανισμοί για την εξισορρόπηση φόρτου και την δυναμική αναπροσαρμογή των χρονικών περιθωρίων μερικές ενδεικτικές πολιτικές υλοποιήθηκαν με βάση αυτούς στο σύστημα.

1.2. Γιατί ένας δικός μας επόπτης επίδοσης;

Τα τελευταία χρόνια σημαντική προσπάθεια έχει καταβληθεί από ερευνητές στα θέματα εποπτείας επίδοσης για καταναμημένα συστήματα, όπως γίνεται σαφές και από το Κεφάλαιο 2. Η δουλειά αυτή έχει οδηγήσει σε μια σειρά από διαφορετικές προτάσεις καθώς και μια σειρά από εμπορικά προϊόντα. Βλέποντας κάποιος τον όγκο αυτό της δουλειάς θα μπορούσε να αναρωτηθεί γιατί δεν χρησιμοποιείται κάποια από τις προτάσεις ή τα προϊόντα αυτά. Μια τέτοια παρατήρηση θα παράβλεπε ουσιαστικά την ιδιαιτερότητα που παρουσιάζουν τα συστήματα ψηφιακής βιβλιοθήκης.

Όπως αναφέρθηκε και παραπάνω τα συστήματα ψηφιακής βιβλιοθήκης είναι, ή φιλοδοξούν να γίνουν, μεγάλης κλίμακας συστήματα, με εκατοντάδες ή και χιλιάδες κόμβους. Επίσης, τα συστήματα αυτά είναι ευρέως καταναμημένα πάνω στο παγκόσμιο δίκτυο. Αποτελούν συνήθως ενώσεις διαφόρων προγραμμάτων και εξυπηρετητών στους οποίους οι διαχειριστές του συστήματος βιβλιοθήκης δεν έχουν έλεγχο. Τέλος, λειτουργούν σε ένα εχθρικό περιβάλλον μια που ούτε το υλικό ούτε το λογισμικό που χρησιμοποιούν δεν είναι αποκλειστικής χρήσης από το σύστημα βιβλιοθήκης. Συνεπώς, ο βασικός ρόλος ενός επόπτη επίδοσης στα συστήματα αυτά είναι να βοηθά στην εξαγωγή συμπερασμάτων σχετικά με τις καθυστερήσεις του δικτύου που χρησιμοποιούν και τον χρόνο απόκρισης των διαφόρων εξαρτημάτων που τα αποτελούν.

Γίνεται λοιπόν φανερό ότι ένας επόπτης επίδοσης για ένα σύστημα ψηφιακής βιβλιοθήκης οφείλει να είναι σχετικά απλός και ευέλικτος, καθώς το σύστημα βιβλιοθήκης δεν έχει στην πράξη την πολύπλοκους περιορισμούς στην λειτουργία, ούτε τον πλήρη έλεγχο που έχει ένα καταναμημένο σύστημα προς τα εξαρτήματα του. Πέρα, όμως, από αυτό ο επόπτης επίδοσης σε ένα σύστημα βιβλιοθήκης πρέπει να παρέχει και ένα ευέλικτο πλαίσιο εργασίας για την υλοποίηση υπηρεσιών προστιθέμενης αξίας (π.χ. επίπεδα ποιότητας υπηρεσίας) από τους διαχειριστές του συστήματος. Για τους παραπάνω λόγους, καθώς και γιατί για διευκόλυνση κατά την διαδικασία ανάπτυξης θέλαμε να έχουμε πλήρη έλεγχο στον κώδικα αποφασίσαμε να προχωρήσουμε στην υλοποίηση ενός δικού μας επόπτη επίδοσης στα πλαίσια αυτής της εργασίας. Η ανάγκη για την δημιουργία ενός δικού μας επόπτη επίδοσης γίνεται περισσότερο εμφανής στην αντιπαράθεση των χαρακτηριστικών του με αυτά των υπάρχοντων εποπτών επίδοσης κατά την εξέταση των σχετικών εργασιών στο επόμενο κεφάλαιο.

1.3. Το σύστημα *NCSTR*L - *Dienst*.

Στην εργασία αυτή, ως σύστημα για την δοκιμή του προτεινόμενου επόπτη επίδοσης και την παρουσίαση των μηχανισμών που αυτός παρέχει, επιλέχθηκε το σύστημα Δικτυακής Βιβλιοθήκης για Τεχνικές Αναφορές στην Επιστήμη Υπολογιστών (**Networked Computer Science Technical Report Library - NCSTR**L). Το σύστημα NCSTR L βασίζεται σε μια σειρά από προηγούμενες προσπάθειες στα πλαίσια του προγράμματος Τεχνικές Αναφορές στην Επιστήμη Υπολογιστών (**Computer Science Technical Reports - CS-TR**) του NSF και κυρίως τα συστήματα WATERS [6] και Dienst [3]. Το NCSTR L είναι ουσιαστικά μια προσπάθεια για την δημιουργία της βασικής υποδομής για μια ψηφιακή βιβλιοθήκη για τεχνικές αναφορές. Τρία στοιχεία το χαρακτηρίζουν: Πρώτον, είναι ένα σύστημα που ήδη λειτουργεί και τυγχάνει ευρείας αποδοχής μια και βασίζεται σε προϋπάρχοντα συστήματα, δεύτερον, αποτελεί ένα όχημα για την εξερεύνηση θεμάτων οργάνωσης και πολιτικής, και τρίτον, είναι ένα πλαίσιο εργασίας για την εισαγωγή και χρήση νέων τεχνολογιών στις ψηφιακές βιβλιοθήκες. Εξάλλου, όπως στοιχειοθετείται στο [9], η αρχιτεκτονική του NCSTR L, που περιγράφουν οι Kahn και Wilensky στο [8], παρέχει ένα πλαίσιο εργασίας για την κατανόηση των ιδεών "κλειδιά" στην αρχιτεκτονική των ψηφιακών βιβλιοθηκών. Τέλος, το σύστημα NCSTR L-Dienst υιοθετήθηκε και από το ERCIM ως η βασική πλατφόρμα της Ερευνητικής του Πρωτοβουλίας για Ψηφιακές Βιβλιοθήκες μέσω του προγράμματος SAMOS. Άρα, η εποτεία επίδοσης στο σύστημα αυτό έχει (ελπίζουμε) ευρύτερο ενδιαφέρον. Οι παραπάνω λόγοι οδήγησαν στην επιλογή του συστήματος NCSTR L-Dienst.

1.4. Συνοπτικά συμπεράσματα.

Η αλματώδης ανάπτυξη των συστημάτων ψηφιακής βιβλιοθήκης τα τελευταία χρόνια έφερε στο προσκήνιο τα θέματα επίδοσης τους. Συγχρόνως, οι ιδιαιτερότητες των συστημάτων αυτών κάνουν αναγκαία μια νέα προσέγγιση των θεμάτων επίδοσης, γενικότερα. Στην εργασία αυτή προτείνεται μία πρώτη τέτοια προσέγγιση καθώς και ένα πλαίσιο εργασίας με βάση την εποπτεία επίδοσης που δίνει νέες δυνατότητες στα συστήματα βιβλιοθήκης. Οι δυνατότητες αυτές είναι επιθυμητές και χρήσιμες τόσο από τους διαχειριστές όσο και από τους χρήστες των συστημάτων αυτών. Επίσης, με την χρήση αυτού του πλαισίου εργασίας σε ένα υπάρχον και αρκετά διαδεδομένο σύστημα ψηφιακής βιβλιοθήκης, το σύστημα NCSTR L-Dienst, οι δυνατότητες αυτού

του πλαισίου εργασίας γίνονται περισσότερο φανερές. Από την άλλη, μέσω της εποπτείας επίδοσης στο παραπάνω σύστημα προκύπτουν μια σειρά από βελτιώσεις στην αρχιτεκτονική του, οι οποίες θα το κάνουν ικανό να ανταποκριθεί ικανοποιητικά στο αυξανόμενο του μέγεθος και φόρτο λειτουργίας. Τέλος, ανοίγεται ένας δρόμος για την βαθύτερη μελέτη των επιμέρους θεμάτων, όπως η εγγύηση ποιότητας υπηρεσίας, η εξισορρόπηση φόρτου, η δυναμική αναπροσαρμογή των χρονικών περιθωρίων κ.α..

1.5. Η οργάνωση του κειμένου.

Το κείμενο, μετά την παρούσα εισαγωγή, ξεκινά, στο δεύτερο κεφάλαιο, με μια παρουσίαση της υπάρχουσας δουλειάς στα θέματα της εποπτείας επίδοσης και των θεμάτων των ψηφιακών βιβλιοθηκών, γενικότερα. Επίσης, παρουσιάζονται οι βασικοί στόχοι της παρούσας εργασίας σε σχέση με τις υπάρχουσες εργασίες.

Στο επόμενο κεφάλαιο, το τρίτο, περιγράφεται η διαδικασία ανάλυσης επίδοσης ενός συστήματος ψηφιακής βιβλιοθήκης και προσδιορισμού των παραμέτρων επίδοσης που θα παρακολουθεί ο επόπτης. Στην συνέχεια του κεφαλαίου, περιγράφεται η αρχιτεκτονική του επόπτη επίδοσης καθώς και οι μηχανισμοί που αυτός παρέχει.

Στο τέταρτο κεφάλαιο περιγράφονται μια σειρά από εφαρμογές των μηχανισμών του επόπτη επίδοσης. Οι εφαρμογές που περιγράφονται είναι η οπτικοποίηση των παραμέτρων επίδοσης, η εγγύηση ποιότητας υπηρεσίας, η εξισορρόπηση φόρτου και η δυναμική αναπροσαρμογή των χρονικών περιθωρίων.

Στο πέμπτο κεφάλαιο έχουμε την περιγραφή του συστήματος NCSTRL – Dienst, σύστημα δοκιμής του επόπτη επίδοσης. Στο κεφάλαιο αυτό περιγράφεται, αρχικά η βασική αρχιτεκτονική του συστήματος Dienst, το μοντέλο εγγράφων του συστήματος και οι υπηρεσίες που παρέχει. Στην συνέχεια περιγράφονται οι μετατροπές της αρχιτεκτονικής που κρίθηκαν απαραίτητες για την συμμόρφωση του συστήματος με την αρχιτεκτονική του NCSTRL, η αρχιτεκτονική των περιοχών (**regional architecture**) και το σύστημα αναγνωριστικών εγγράφων (**handle system**).

Το έκτο κεφάλαιο περιλαμβάνει την ανάλυση επίδοσης του συστήματος NCSTRL – Dienst, σαν συγκεκριμένη εφαρμογή της διαδικασίας που περιγράφηκε προηγούμενα. Η ανάλυση της επίδοσης γίνεται με τον ορισμό μιας σειράς παραμέτρων επίδοσης και τον προσδιορισμό των σχέσεων αυτών μέσω αλγεβρικών παραστάσεων. Στόχος της ανάλυσης είναι ο προσδιορισμός των ενδιαφερόντων παραμέτρων που θα αποτελέσουν και τις παραμέτρους που θα παρακολουθεί ο επόπτης επίδοσης.

Στο έβδομο κεφάλαιο παρουσιάζονται συμπεράσματα από την εποπτεία επίδοσης του συστήματος NCSTRL-Dienst, ενώ περιγράφεται και μια σειρά από τροποποιήσεις στην σχεδίαση και την υλοποίηση του συστήματος. Στην συνέχεια, περιγράφεται η υλοποίηση ορισμένων ενδεικτικών πολιτικών εξισορρόπησης φόρτου και δυναμικής αναπροσαρμογής των χρονικών περιθωρίων για το συγκεκριμένο σύστημα και παρουσιάζονται συμπεράσματα από την χρήση τους.

Το όγδοο κεφάλαιο συνοψίζει τα συμπεράσματα τις παρούσας εργασίας και προτείνει μια σειρά από πιθανά θέματα προς διερεύνηση στο μέλλον, τα οποία είτε αφορούν τον ίδιο τον επόπτη είτε τις εφαρμογές του.

Η εργασία ολοκληρώνεται με 7 παραρτήματα. Το πρώτο παράρτημα περιγράφει το πρωτόκολλο επικοινωνίας με τον επόπτη επίδοσης. Το δεύτερο δίνει ένα παράδειγμα αρχείου καταγραφής των μετρήσεων. Το τρίτο δίνει ένα παράδειγμα της χρήσης του συστήματος NCSTRL-Dienst. Το τέταρτο περιγράφει το πρωτόκολλο του συστήματος NCSTRL – Dienst. Το πέμπτο περιέχει τον ορισμό των παραμέτρων

επίδοσης για το σύστημα NCSTRL – Dienst. Το έκτο περιγράφει αναλυτικά τις αλλαγές και τις προσθήκες που έγιναν στον κώδικα του συστήματος NCSTRL-Dienst για την παρακολούθηση της επίδοσης και τις εφαρμογές της. Το τελευταίο παράρτημα είναι ένα ελληνοαγγλικό λεξικό τεχνικών όρων που στόχο έχει την διευκόλυνση των αναγνωστών στην κατανόηση του παρόντος κειμένου και την σύνδεση του κειμένου με την ορολογία της διεθνούς βιβλιογραφίας.

Κεφάλαιο 2. Σχετικές Εργασίες.

Το θέμα της εποπτείας επίδοσης στα συστήματα ψηφιακών βιβλιοθηκών φέρνει σε σύγκλιση δύο παράλληλους μέχρι τώρα τομείς έρευνας, τον τομέα της εποπτείας επίδοσης στα καταναμημένα συστήματα και τον τομέα των ψηφιακών βιβλιοθηκών. Έτσι, στο χώρο της εποπτείας, και γενικότερα διαχείρισης επίδοσης, σε καταναμημένα συστήματα υπάρχει σημαντική δουλειά τόσο από ερευνητικές ομάδες πανεπιστημίων και ερευνητικών ινστιτούτων [18, 19, 21, 22, 23, 24, 26, 27, 28] όσο και εταιριών. Πέρα μάλιστα από την ερευνητική δουλειά κυκλοφορούν στην αγορά και μια σειρά προϊόντων διαχείρισης επίδοσης καταναμημένων συστημάτων [20, 29, 30, 31, 32, 33]. Από την άλλη, τα τελευταία χρόνια, όπως προαναφέρθηκε, υπάρχει και σημαντικό ενδιαφέρον για τα θέματα των ψηφιακών βιβλιοθηκών. Έτσι, υπάρχουν σήμερα στον κόσμο αρκετές ερευνητικές ομάδες που ασχολούνται με το θέμα, προτείνοντας είτε ολοκληρωμένες λύσεις (πλήρη συστήματα ψηφιακής βιβλιοθήκης)[7, 35, 36, 37, 38, 39, 40, 41, 42, 43] είτε λύσεις για επιμέρους προβλήματα.

Η εποπτεία, γενικότερα, καταναμημένων συστημάτων ξεκίνησε αρχικά με την μεταφορά των μεθόδων που υπήρχαν από την εποπτεία σειριακών προγραμμάτων. Στα τελευταία χρόνια, όμως, έχουν γίνει σημαντικά άλματα με αποτέλεσμα ο ίδιος ο επόπτης του συστήματος να είναι αρκετές φορές καταναμημένος. Η λειτουργικότητα που προσφέρουν οι επόπτες συστήματος διαιρείται στις εξής κατηγορίες [18]:

- Αξιοπιστία (**dependability**) που περιλαμβάνει ανοχή λαθών (**fault-tolerance**) και ασφάλεια (**safety**).
- Βελτίωση Επίδοσης (**performance enhancement**) που περιλαμβάνει δυναμική διαμόρφωση συστήματος, δυναμική ρύθμιση προγραμμάτων και εν λειτουργία καθοδήγηση (**on-line steering**).
- Έλεγχος Ορθότητας (**correctness checking**) που είναι η παρακολούθηση μιας εφαρμογής με σκοπό την εξασφάλιση της συνέπειας της με κάποιο τυπικό μοντέλο (**formal model**). Μπορεί να χρησιμοποιηθεί για τον εντοπισμό λαθών εκτέλεσης ή σαν τεχνική επαλήθευσης.
- Ασφάλεια (**security**) που είναι η παρακολούθηση με σκοπό τον εντοπισμό παραβιάσεων του συστήματος, όπως παράνομη είσοδο ή προσπάθεια πρόσβασης αρχείου.
- Έλεγχος (**control**) που περιλαμβάνει τις περιπτώσεις εκείνες που ο επόπτης είναι ενσωματωμένος με το σύστημα που εποπτεύει και παρέχει πρόσθετη λειτουργικότητα στο σύστημα.
- Αποσφαλμάτωση (**debugging**) και δοκιμή (**testing**) όταν οι τεχνικές εποπτείας συστήματος χρησιμοποιούνται για την εξαγωγή συγκεκριμένων τιμών από το υπό έλεγχο σύστημα.
- Εκτίμηση Επίδοσης (**performance evaluation**) χρησιμοποίηση του επόπτη με σκοπό την εξαγωγή δεδομένων του συστήματος που

αναλύονται αργότερα με στόχο την αξιολόγηση της απόδοσης του συστήματος.

Η παρούσα εργασία επικεντρώνεται βασικά σε δύο από τις παραπάνω κατηγορίες εποπτείας, στην εκτίμηση επίδοσης και στην βελτίωση της απόδοσης, εποπτεία επίδοσης. Εξάλλου, ο προτεινόμενος επόπτης επίδοσης είναι ένας εν λειτουργία επόπτης επίδοσης (**on-line performance monitor**), αφού έχει τα απαραίτητα χαρακτηριστικά [18]: (α) είναι ένας εξωτερικός παρατηρητής, (β) εποπτεύει ένα σύστημα σε πλήρη λειτουργία και (γ) προορίζεται να είναι μόνιμος.

Από την άλλη οι επόπτες επίδοσης ανάλογα με το νοηματικό μοντέλο στο οποίο βασίζεται η λειτουργία τους χωρίζονται [19] σε επόπτες κατασκευής προφίλ προγράμματος (**program profiling**) όπως το *Parasight* της Encore Corp. [20], επόπτες συμβάντων (**event based**) όπως το σύστημα Pablo [21, 22] που παρέχει ένα περιβάλλον για την συλλογή, ανάλυση και παρουσίαση δεδομένων επίδοσης και το JEWEL [23] ένα καταναμημένο σύστημα μετρήσεων που μπορεί και προσαρμόζεται στον προσδιορισμό κάθε εφαρμογής (**application-specific adaptations**), τέλος σε επόπτες που βασίζονται σε μοντελοποίηση πληροφοριών, όπως αυτόν που παρουσίασε ο Snodgrass στην διδακτορική του διατριβή [24], και η συλλογή και ανάκληση των πληροφοριών γίνεται με ενημερώσεις και ερωτήσεις σε μία χρονική βάση δεδομένων. Στην παρούσα εργασία ακολουθήθηκε η προσέγγιση της εποπτείας συμβάντων, μια που αυτό που θέλαμε ήταν να παρακολουθήσουμε την ενεργοποίηση των διαφόρων εξαρτημάτων του συστήματος ψηφιακής βιβλιοθήκης καθώς και τα μηνύματα που ανταλλάσσουν μεταξύ τους.

Τα συμβάντα που μπορεί να παρακολουθεί ένας επόπτης μπορεί να είναι γεγονότα στο επίπεδο του υλικού, στο επίπεδο των διαδικασιών και στο επίπεδο της εφαρμογής [18]. Τα γεγονότα, όμως, που βρίσκονται στο επίπεδο του υλικού και των διεργασιών είναι αδύνατο να παρακολουθηθούν με τρόπο ανεξάρτητο από το λειτουργικό σύστημα και τη μηχανή στην οποία συμβαίνουν. Η ανάγκη αυτή οδήγησε στην δημιουργία της παγκόσμιας αρχιτεκτονικής μέτρησης (**Universal Measurement Architecture – UMA**) [25], η οποία υιοθετήθηκε ως πρότυπο από το X/Open το 1997, ενώ ήδη κυκλοφορεί μια εμπορική υλοποίηση από την Amdahl [34]. Στην παρούσα εργασία περιοριστήκαμε σε συμβάντα στο επίπεδο της εφαρμογής, γιατί η μεταφερσιμότητα (**portability**) του επόπτη επίδοσης ήταν ουσιαστικής σημασίας.

Δύο άλλα σημαντικά θέματα στην εποπτεία επίδοσης είναι ο καθορισμός των παραμέτρων επίδοσης και η συλλογή των δεδομένων επίδοσης. Ο καθορισμός των παραμέτρων επίδοσης, από την στιγμή που η παρακολούθηση του συστήματος γίνεται στο επίπεδο της εφαρμογής, γίνεται με την ανάλυση επίδοσης του συστήματος από τον διαχειριστή του για τον προσδιορισμό των τμημάτων ενδιαφέροντος. Επίσης, η συλλογή των δεδομένων γίνεται με την αποστολή μηνυμάτων από το σύστημα στον επόπτη επίδοσης, τα οποία εισάγονται από τον διαχειριστή του συστήματος στην αρχή και στο τέλος των ενδιαφερόντων τμημάτων. Τέλος, παρόλο που επόπτης κάνει μιας μορφής ανάλυση των δεδομένων εν λειτουργία (όπως υπολογισμός ελαχίστου, μεγίστου, μέσης τιμής και τυπικής απόκλισης) δίνεται η δυνατότητα για παραπέρα ανάλυση της λειτουργίας μέσω των αρχείων καταγραφής μετρήσεων που διατηρεί το σύστημα.

Όσον αφορά τα εμπορικά προϊόντα, παρόλο που είναι ιδιαίτερα ισχυρά εργαλεία για την εποπτεία επίδοσης αλλά και την διαχείριση του συστήματος, μπορούν να εργαστούν σε ένα περιορισμένο αριθμό λειτουργικών συστημάτων και μηχανών (βλέπε παραπάνω και την επιχειρηματολογία για το πρότυπο **UMA**). Έτσι, δεν θα μπορούσαμε να είχαμε επιλέξει κάποιο από αυτά πρώτον διότι δεν καλύπτουν

την ανάγκη για μεταφερσιμότητα και δεύτερον το λογισμικό τους δεν είναι διαθέσιμο για τροποποιήσεις που πιθανόν θα ήταν αναγκαίες.

Από την άλλη, τα συστήματα ψηφιακών βιβλιοθηκών είναι συστήματα που προσφέρουν πρόσβαση σε ψηφιοποιημένη πληροφορία διαφόρων μορφών. Έτσι, το σύστημα Informedia [46] είναι μία βιβλιοθήκη με ταινίες, το σύστημα VARIATIONS [36] με μουσικά κομμάτια, το σύστημα NCSTRL-Dienst [3] με τεχνικές αναφορές. Βασικό χαρακτηριστικό των συστημάτων αυτών είναι το ότι συνδυάζουν διάφορες πηγές πληροφορίας με την σύνδεση τους μέσω του παγκόσμιου ιστού, με χαρακτηριστικά παραδείγματα το σύστημα Alexandria [45], που έχει εικόνες δορυφόρων, χάρτες και μετρήσεις αποθεμάτων ύδατος, και το Infobus [48], που συνδέει όλους τους δημοφιλείς εξυπηρετητές, π.χ. Altavista, το σύστημα Alexandria και το σύστημα βιβλιοθήκης του πανεπιστημίου του Michigan. Επίσης, τα περισσότερα από αυτά τα συστήματα είναι κατανομημένα σε μεγάλη έκταση, π.χ. το σύστημα NCSTRL-Dienst είναι εγκατεστημένο σε πάνω από 15 χώρες καλύπτοντας ολόκληρες τις Η.Π.Α. και μεγάλο μέρος της Ευρώπης. Τέλος, ο όγκος των πληροφοριών που περιέχουν είναι της τάξεως των Gigabytes, χαρακτηριστική περίπτωση το σύστημα Alexandria που έχει εικόνες δορυφόρων, χάρτες κ.α..

Στην σχεδίαση των συστημάτων ψηφιακής βιβλιοθήκης έχουν χρησιμοποιηθεί διάφορα μοντέλα. Έτσι, το σύστημα του πανεπιστημίου Stanford [48] είναι ουσιαστικά ένα μέσο μεταφοράς πληροφορίας μεταξύ των δημοφιλέστερων πηγών στο παγκόσμιο δίκτυο, όπως μηχανές αναζήτησης (**search engines**) αλλά και τα συστήματα ψηφιακής βιβλιοθήκης άλλων πανεπιστημίων, όπως του πανεπιστημίου Michigan [47] και του πανεπιστημίου της Καλιφόρνιας στην Santa Barbara [45]. Για την μεταφορά της η πληροφορία εισάγεται σε αντικείμενα περιεχομένου (**container object**). Από την άλλη, στο πανεπιστήμιο του Michigan [47] το σύστημα ψηφιακής βιβλιοθήκης είναι ουσιαστικά μια ομάδα από πράκτορες (**agents**), οι οποίοι εκτελούν κάποιες εργασίες και συνεργάζονται μεταξύ τους για την εξυπηρέτηση των χρηστών.

Στην παρούσα εργασία θεωρούμε το σύστημα ψηφιακής βιβλιοθήκης να αποτελείται από μία σειρά εξυπηρετητές, οι οποίοι αλληλεπιδρούν για την εξυπηρέτηση των αιτήσεων των χρηστών και είναι διασκορπισμένοι στο παγκόσμιο δίκτυο. Επίσης, για την εξυπηρέτηση της αίτησης του χρήστη είναι δυνατόν να καλούνται και διάφορα εξωτερικά προγράμματα. Η επικοινωνία μεταξύ των εξυπηρετητών γίνεται με την χρήση του πρωτοκόλλου του παγκόσμιου δικτυακού ιστού, του πρωτοκόλλου μεταφοράς υπερκειμένου (**Hypertext Transfer Protocol – HTTP**). Επομένως, ο επόπτης επίδοσης έχει ως κύριο στόχο την παρακολούθηση των διαφόρων εξαρτημάτων ως αυτούσιες οντότητες, μια και οι διαχειριστές των συστημάτων ψηφιακής βιβλιοθήκης δεν μπορούν να επηρεάσουν την εσωτερική λειτουργία κανενός ανεξάρτητου προγράμματος παρά μόνο των εξυπηρετητών. Ακόμη, βασικά του στοιχεία πρέπει να είναι η απλότητα, μια και το περιβάλλον λειτουργίας είναι εχθρικό (υπάρχουν πολλά προγράμματα που διεκδικούν τους ίδιους πόρους, ιδίως το δίκτυο) και η μεταφερσιμότητα, μια και τα συστήματα τα οποία χρησιμοποιεί είναι ιδιαίτερα ετερογενή.

Τέλος, στην παρούσα εργασία, η εποπτεία επίδοσης στα συστήματα ψηφιακών βιβλιοθηκών είναι ουσιαστικά το πρώτο βήμα για την υποστήριξη αναζήτησης και ανάκλησης σε διάφορα επίπεδα ποιότητας υπηρεσίας (**Quality of Service Levels**)[10, 11], τεχνικών δυναμικής αναπροσαρμογής των χρονικών περιθωρίων και στρατηγικών εξισορρόπησης φόρτου. Η δουλειά για την επίτευξη του παραπάνω στόχου, πέρα από το σημαντικό ενδιαφέρον που παρουσιάζει, απαιτεί και μια σε βάθος εξέταση του ρόλου της κλιμάκωσης (μεγέθους και κατανομής) στα θέματα αυτά που εισάγουν τα συστήματα βιβλιοθήκης. Αυτό συμβαίνει γιατί, παρόλο

που διάφορες προτάσεις για δρομολόγηση και εξισορρόπησης φόρτου σε καταναμημένα συστήματα διαχείρισης δοσοληψιών έχουν γίνει στο παρελθόν, ενδεικτικά είναι τα [14, 15, 16, 17], η ιδέα της διαχείρισης εκατοντάδων ή και χιλιάδων συστημάτων διασκορπισμένων στο παγκόσμιο δίκτυο τα οποία πρέπει να συντονιστούν για την εκτέλεση κάποιας εργασίας, όπως είναι και τα συστήματα ψηφιακής βιβλιοθήκης, εμφανίζεται ουσιαστικά μόνο στο σύστημα *Condor* [12, 13].

Κεφάλαιο 3. Ανάλυση Επίδοσης και Αρχιτεκτονική Επόπτη Επίδοσης.

Ο στόχος της εποπτείας επίδοσης ενός συστήματος είναι να δώσει στον διαχειριστή του μια εικόνα για τον χρόνο που καταναλώνεται στα διάφορα τμήματα της λειτουργίας του. Έτσι, ο διαχειριστής μπορεί να εντοπίσει τα σημεία συμφόρησης του συστήματος και να λάβει τα απαραίτητα μέτρα για την αποσυμφόρηση του. Συγχρόνως, ο επόπτης επίδοσης δίνει την δυνατότητα και στο ίδιο το σύστημα να γνωρίζει την κατάσταση του και να αλλάζει δυναμικά την συμπεριφορά του ώστε να προσαρμόζεται στις νέες συνθήκες.

Για την διαχείριση της επίδοσης στα συστήματα ψηφιακών βιβλιοθηκών προτείνουμε έναν επόπτη επίδοσης βασισμένο στο μοντέλο πελάτη-εξυπηρετητή (**Client-Server**). Το μοντέλο λειτουργίας του επόπτη βασίζεται σε ιδέες από την διαχείριση επίδοσης στα δίκτυα και στα καταναμημένα συστήματα. Τρία είναι τα βασικά κομμάτια της σχεδίασης:

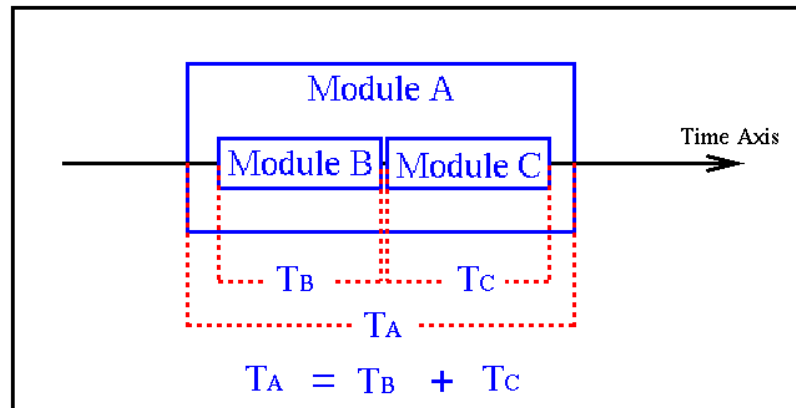
1. Ο ορισμός των παραμέτρων επίδοσης.
2. Η μέτρηση και αποθήκευση των παραμέτρων.
3. Το πρωτόκολλο για την ανάκληση των παραμέτρων.

Στην συνέχεια παρουσιάζεται η διαδικασία ορισμού των παραμέτρων επίδοσης, η οποία θα γίνει περισσότερο κατανοητή στα επόμενα κεφάλαια με την χρήση της στο σύστημα NCSTR-L-Dienst, και η αρχιτεκτονική του συστήματος μέτρησης και αποθήκευσης των παραμέτρων. Το πρωτόκολλο επικοινωνίας επόπτη - διαχειριστή/συστήματος παρουσιάζεται στο Παράρτημα 1.

3.1. Ο ορισμός των παραμέτρων επίδοσης.

Το πρώτο βήμα για την εποπτεία και την μέτρηση της επίδοσης ενός συστήματος είναι η ανάλυση της επίδοσης του. Σκοπός της ανάλυσης επίδοσης είναι η βαθύτερη κατανόηση του τρόπου λειτουργίας του συστήματος και ο προσδιορισμός των τμημάτων (**modules**) λειτουργίας ιδιαίτερου ενδιαφέροντος. Τέτοια τμήματα είναι, συνήθως, ιδιαίτερα χρονοβόρα τμήματα, ή τμήματα κρίσιμα για την ορθή λειτουργία του συστήματος, π.χ. η επικοινωνία μεταξύ των εξυπηρετητών που αποτελούν το σύστημα, ή η αναζήτηση στην αποθήκη αντικειμένων για την ικανοποίηση μιας αίτησης του χρήστη. Με την ολοκλήρωση της ανάλυσης επίδοσης έχουμε ένα πλήρες μοντέλο λειτουργίας του συστήματος, όπου οι σχέσεις και οι αλληλεξαρτήσεις μεταξύ των διάφορων τμημάτων αναπαρίστανται με μαθηματικές σχέσεις. Στις σχέσεις αυτές κάθε μεταβλητή αναπαριστά τον χρόνο που καταναλώνεται σε κάποιο τμήμα της λειτουργίας (βλέπε στο Σχήμα 1, όπου έχουμε ένα τμήμα, το τμήμα A που περιλαμβάνει δύο άλλα τμήματα, τα B και C, ο χρόνος στο τμήμα A είναι T_A , ο χρόνος για το B είναι T_B και ο χρόνος για το C είναι T_C , επομένως προκύπτει η εξίσωση $T_A \geq T_B + T_C$). Με επιλογή των τμημάτων που

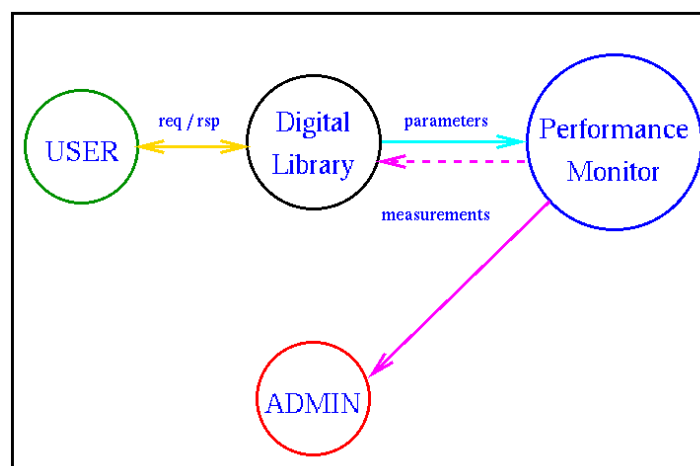
θεωρούμε σημαντικά ορίζουμε παραμέτρους επίδοσης που να απεικονίζουν τις αντίστοιχες μεταβλητές.



Σχήμα 1: Παράδειγμα Ανάλυσης Επίδοσης.

3.2. Η αρχιτεκτονική μέτρησης και αποθήκευσης των παραμέτρων επίδοσης.

Οι τιμές των διαφόρων παραμέτρων επίδοσης που ορίστηκαν από τον διαχειριστή του συστήματος μετά την ανάλυση επίδοσης, μετρούνται από τον επόπτη επίδοσης. Συγχρόνως, για κάθε παράμετρο επίδοσης υπολογίζονται και στοιχεία όπως η μέση τιμή της, η διασπορά της κ.α.. Ακόμα, ο διαχειριστής του συστήματος μπορεί να ζητήσει όλα ή κάποια από τα στοιχεία που υπολογίζει ο επόπτης επίδοσης για συγκεκριμένες ή όλες τις παραμέτρους επίδοσης με την χρήση του πρωτοκόλλου επικοινωνίας με τον επόπτη. Τέλος, το ίδιο το σύστημα μπορεί εάν το επιθυμεί να ενημερώνεται για την εξέλιξη των τιμών των παραμέτρων επίδοσης με την χρήση του ίδιου πρωτοκόλλου (βλέπε στο Σχήμα 2).

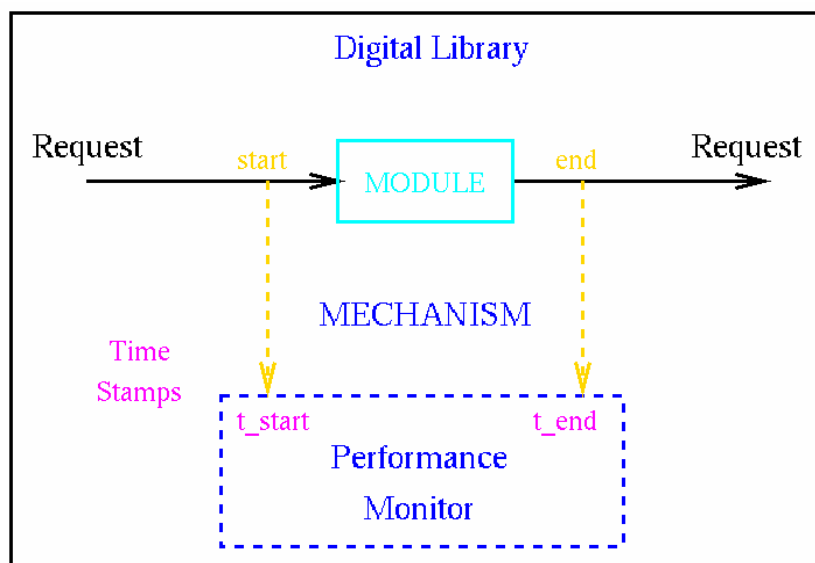


Σχήμα 2: Μοντέλο Λειτουργίας Επόπτη Επίδοσης.

3.2.1. Ο τρόπος μέτρησης των παραμέτρων επίδοσης.

Για να μετρήσουμε τον χρόνο που καταναλώνεται σε κάποιο τμήμα επεξεργασίας κατά την διάρκεια της λειτουργίας του συστήματος ψηφιακής βιβλιοθήκης πρέπει, να καταγραφεί, με την τοποθέτηση χρονοσφραγίδας

(**timestamp**), ο χρόνος εκκίνησης και τερματισμού του. Ο χρόνος που καταναλώθηκε προκύπτει ως η διαφορά των δύο χρονοσφραγίδων (βλέπε στο Σχήμα 3).



Σχήμα 3: Μηχανισμός Μέτρησης Παραμέτρων Επίδοσης.

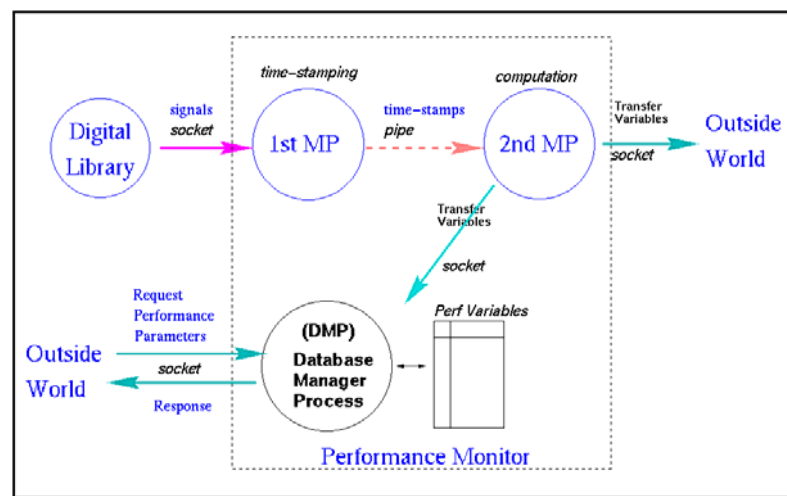
Πιο αναλυτικά, καθώς το σύστημα ψηφιακής βιβλιοθήκης εξυπηρετεί μία αίτηση του χρήστη, η εκτέλεση φτάνει στο προς μέτρηση τμήμα. Κατά την εκκίνηση του τμήματος το μήνυμα «αρχή τμήματος» στέλνεται στον επόπτη επίδοσης, ο οποίος και τοποθετεί χρονοσφραγίδα στο μήνυμα αυτό. Όταν η εκτέλεση του τμήματος ολοκληρωθεί το μήνυμα «τέλος τμήματος» στέλνεται στον επόπτη επίδοσης, και χρονοσφραγίζεται. Στην συνέχεια, η εκτέλεση προχωρά σε επόμενα τμήματα, ενώ ο επόπτης υπολογίζει την τιμή της αντίστοιχης παραμέτρου επίδοσης, δηλαδή την διαφορά των δύο χρονοσφραγίδων και ενημερώνει, με βάση την νέα τιμή, τα αλλά στοιχεία που σχετίζονται με την παράμετρο.

Τα ονόματα των παραμέτρων επίδοσης καθώς και η αντιστοίχιση τους με τα τμήματα λειτουργίας του συστήματος βιβλιοθήκης δεν είναι προκαθορισμένα, αλλά γίνονται γνωστά στον επόπτη επίδοσης μέσω των μηνυμάτων εκκίνησης τμήματος. Επομένως, για τον ορισμό μιας παραμέτρου αρκεί να σταλεί στον επόπτη ένα μήνυμα εκκίνησης με το όνομα της. Από την άλλη ο επόπτης έχει και ορισμένα προκαθορισμένα μηνύματα τα οποία είναι καλό να χρησιμοποιούνται για την πληρέστερη ενημέρωση και την καλύτερη μορφή των αρχείων καταγραφής των μετρήσεων, π.χ. τα μηνύματα εκκίνησης, τερματισμού και επανεκκίνησης (**reset**) των μετρήσεων του επόπτη επίδοσης. Η μορφή των μηνυμάτων εκκίνησης και τερματισμού τμήματος καθώς και των προκαθορισμένων μηνυμάτων περιγράφεται αναλυτικά στο Παράρτημα 1.

3.2.2. Η αρχιτεκτονική του επόπτη επίδοσης.

Ο επόπτης επίδοσης αποτελείται από δύο διαδικασίες μέτρησης (Measurement Process - MP) και μία διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης (Database Manager Process - DMP), οι οποίες ανταλλάσσουν μηνύματα μεταξύ τους παρακολουθώντας τις τιμές των παραμέτρων επίδοσης και προσφέρουν στον χρήστη, σε άλλα προγράμματα και στο ίδιο το σύστημα ψηφιακής βιβλιοθήκης πρόσβαση σε αυτές, εάν το επιθυμούν (βλέπε το Σχήμα 4). Ο επόπτης μπορεί να κρατά και κάποια αρχεία καταγραφής (log files) των μετρήσεων ώστε να

διατηρείται η ιστορία των μετρήσεων μεταξύ των διαδοχικών του ενεργοποιήσεων (για παράδειγμα βλέπε στο Παράρτημα 2). Η διαίρεση της λειτουργίας της μέτρησης σε δύο διαδικασίες μέτρησης έγινε για να μην επηρεάζεται η χρονοσφραγίδα των μηνυμάτων εκκίνησης/τερματισμού τμήματος από την επεξεργασία των μηνυμάτων. Τα μηνύματα του συστήματος ψηφιακής βιβλιοθήκης πρέπει να παραλαμβάνονται αμέσως και να μην καθυστερεί ιδιαίτερα το χρονοσφράγισμα τους, πράγμα που θα ήταν αδύνατο αν η ίδια διαδικασία έπρεπε και να τα επεξεργάζεται. Από την άλλη, η διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης κρίθηκε απαραίτητη πρώτον για να μπορεί το σύστημα ψηφιακής βιβλιοθήκης να ενημερώνεται αμέσως για τις τιμές των παραμέτρων επίδοσης όταν το επιθυμεί διότι στις τιμές αυτές μπορεί να βασίζονται αποφάσεις για την λειτουργία του, και δεύτερον για να μπορεί κάποιος στο μέλλον να προσθέσει νέες μεθόδους ανάκλησης ή περίπλοκους υπολογισμούς συναρτήσεων των τιμών των παραμέτρων χωρίς να χρειαστεί να προχωρήσει σε αλλαγές του συστήματος μετρήσεων.



Σχήμα 4: Αρχιτεκτονική Επόπτη Επίδοσης.

Η πρώτη διαδικασία μέτρησης είναι επιφορτισμένη με την συλλογή και το χρονοσφράγισμα των μηνυμάτων εκκίνησης/τερματισμού τμήματος του συστήματος ψηφιακής βιβλιοθήκης και την μεταφορά των χρονοσφραγισμένων μηνυμάτων στην δεύτερη διαδικασία μέτρησης. Η επικοινωνία με το σύστημα ψηφιακής βιβλιοθήκης γίνεται μέσω *sockets*. Συγκεκριμένα, κατά την εκκίνηση του επόπτη επίδοσης το σύστημα βιβλιοθήκης ανοίγει ένα *socket*, το οποίο παραμένει ανοικτό μέχρι την διακοπή της λειτουργίας των επόπτη. Μέσω του *socket* αυτού στέλνονται τα διάφορα μηνύματα εκκίνησης/τερματισμού τμημάτων. Το ίδιο *socket* χρησιμοποιείται και από πιθανές «διεργασίες παιδιά» της βασικής διεργασίας του συστήματος βιβλιοθήκης. Ταυτόχρονα, η πρώτη διαδικασία μέτρησης μπορεί να δεχτεί και συνδέσεις από άλλες πηγές. Τέτοιες πηγές θα μπορούσαν να είναι εξαρτήματα του συστήματος βιβλιοθήκης που λειτουργούν ως ανεξάρτητες διεργασίες. Τέλος, η μεταφορά των χρονοσφραγισμένων μηνυμάτων προς την δεύτερη διαδικασία μέτρησης γίνεται μέσω ενός *pipe*.

Η δεύτερη διαδικασία μέτρησης είναι επιφορτισμένη με την επεξεργασία των χρονοσφραγισμένων μηνυμάτων. Η επεξεργασία των μηνυμάτων περιλαμβάνει τον υπολογισμό των τιμών των παραμέτρων επίδοσης, δηλαδή τον προσδιορισμό της παραμέτρου επίδοσης στην οποία αναφέρονται τα μηνύματα και την συσχέτιση των αντίστοιχων μηνυμάτων εκκίνησης και τερματισμού τμήματος. Ακόμα, η

επεξεργασία περιλαμβάνει την ενημέρωση του μεγίστου, του ελαχίστου, της μέσης τιμής, της διασποράς και ενός σταθμισμένου μέσου όρου για κάθε παράμετρο επίδοσης, καθώς και την ενημέρωση των αρχείων καταγραφής των μετρήσεων. Τέλος, η δεύτερη διαδικασία μέτρησης είναι υπεύθυνη και για την μεταφορά των επεξεργασμένων στοιχείων στην διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης ή μιας οποιαδήποτε άλλης εφαρμογής, μέσω ενός *socket*, και την καταγραφή σε ένα αρχείο των μετρήσεων κατά τον τερματισμό του επόπτη καθώς και την φόρτωση του αρχείου αυτού κατά την εκκίνησή του.

Η διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης αποθηκεύει τις τιμές των παραμέτρων επίδοσης και τα στοιχεία που η δεύτερη διαδικασία μέτρησης έχει υπολογίσει για αυτές, και μπορεί να απαντήσει σε ερωτήσεις ανάκλησης των τιμών αυτών. Η υποβολή των ερωτήσεων μπορεί να γίνει είτε από το ίδιο το σύστημα ψηφιακής βιβλιοθήκης είτε από κάποια άλλη ανεξάρτητη εφαρμογή μέσω *socket*. Για την απάντηση κάθε ερώτησης δημιουργείται ένα αντίτυπο της διαδικασίας διαχείρισης, που θα την ικανοποιήσει. Οι ερωτήσεις μπορεί να αφορούν κάποιες ή όλες τις παραμέτρους επίδοσης και κάποια ή όλα τα στοιχεία που ο επόπτης υπολογίζει για τις παραμέτρους αυτές. Οι υποστηριζόμενες ερωτήσεις περιγράφονται στο Παράρτημα 1.

3.2.3. Θέματα Υλοποίησης.

Ο επόπτης επίδοσης υλοποιήθηκε σε *Perl*. Η επιλογή της *Perl* έγινε κυρίως για τις δυνατότητες μεταφοράς του προγράμματος (**portability**) που παρέχει. Ο κώδικας του επόπτη βρίσκεται σε πέντε διαφορετικά αρχεία *main.pl*, *procedures.pl*, *networking.pl*, *io_funcs.pl* και *socket_handler.pl*. Τα πρώτα τέσσερα αρχεία περιέχουν τον κώδικα για τις δύο διαδικασίες μέτρησης και το τελευταίο τον κώδικα για την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης.

Αναλυτικότερα, το *main.pl* περιέχει τους ορισμούς των δομών που χρησιμοποιούν οι διαδικασίες μέτρησης και τον κώδικα για την αρχικοποίηση των διαδικασιών αυτών. Κατά την αρχικοποίηση του ο επόπτης αναπαράγει τον εαυτό του (**fork**) και οι δύο διαδικασίες αναλαμβάνουν τον ρόλο της πρώτης και δεύτερης διαδικασίας μέτρησης. Ένα κανάλι επικοινωνίας (*pipe*) μεταξύ των δύο διαδικασιών κατασκευάζεται για την μεταφορά των χρονοσφραγισμένων μηνυμάτων.

Το *procedures.pl* περιέχει τον κώδικα για την διαχείριση των χρονοσφραγισμένων μηνυμάτων που παράγει η πρώτη διαδικασία μέτρησης. Για κάθε τύπο μηνύματος υπάρχει η αντίστοιχη ρουτίνα (**procedure**) για τον χειρισμό του (για τους τύπους των μηνυμάτων βλέπε στο Παράρτημα 1). Ο μηχανισμός της αντιστοίχισης των μηνυμάτων εκκίνησης/τερματισμού μέτρησης πάνω στον οποίο είναι κτισμένο το σύστημα παραγωγής μετρήσεων, βασίζεται σε μία δομή δεδομένων που αποθηκεύει τα μηνύματα εκκίνησης μετρήσεων, η αναζήτηση στην οποία εξασφαλίζει την ορθή αντιστοίχιση. Η δομή αυτή είναι ένας διδιάστατος πίνακας κατακερματισμού (**hash table**) στον οποίο το πρώτο κλειδί είναι τα ονόματα των αρχείων καταγραφής των μετρήσεων και το δεύτερο κλειδί είναι ένα από τα **FILE_HANDLER**, για την αποθήκευση του αναγνωριστικού για το αρχείο καταγραφής, **FORK_NUM**, για τον αριθμό των αναπαραγωγών (**fork**) του εξυπηρετητή και **SUB_STACK**, μία στοίβα για τα μηνύματα εκκίνησης μέτρησης τμήματος που αντιστοιχούν στο συγκεκριμένο αρχείο καταγραφής των μετρήσεων. Από τα παραπάνω γίνεται εμφανής η σημασία των διαφόρων πεδίων που αποτελούν τα μηνύματα επικοινωνίας με τις διαδικασίες μέτρησης. Πέρα από την παραπάνω δομή υπάρχει και μία δομή που κρατά τις τιμές των παραμέτρων επίδοσης. Η δομή είναι ουσιαστικά ένας διδιάστατος πίνακας κατακερματισμού όπου το πρώτο κλειδί

είναι το όνομα της παραμέτρου επίδοσης, το δεύτερο κλειδί είναι το όνομα του διακριτικού της παραμέτρου επίδοσης (π.χ. οι αρχές έκδοσης στο σύστημα NCSTRL-Dienst) ή *SimpleVar* για τις υπόλοιπες και οι τιμές του είναι ουσιαστικά μια λίστα με τις διάφορες τιμές τις παραμέτρου επίδοσης. Οι τιμές χωρίζονται μεταξύ τους με το ‘;’. Αυτή η δομή είναι που αποθηκεύεται σε ένα αρχείο όταν ο επόπτης σταματά την λειτουργία του και αυτή την δομή επανακατασκευάζει ο επόπτης με βάση το αυτό το αρχείο όταν ξαναξεκινά.

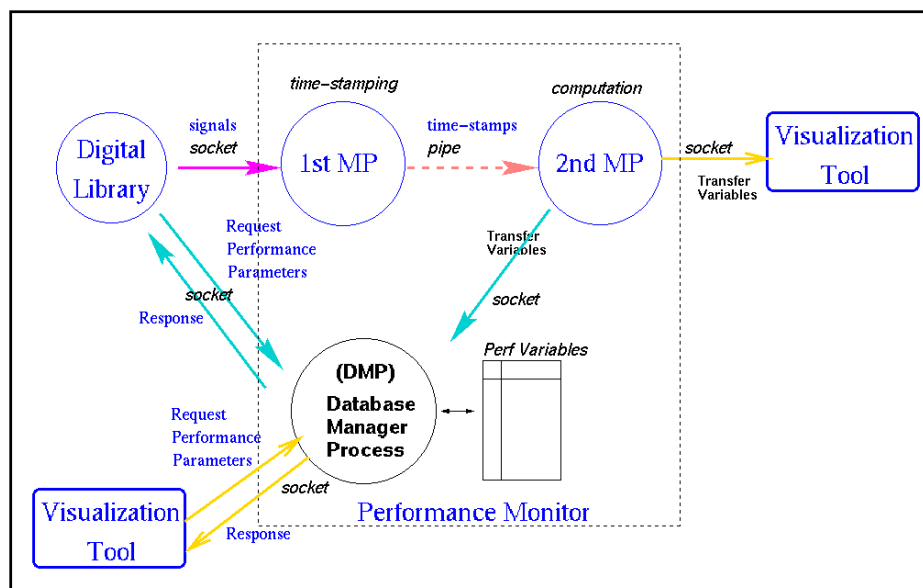
Το *io_funcs.pl* περιέχει τις ρουτίνες για την διαχείριση των αρχείων καταγραφής και το *networking.pl* τις ρουτίνες για την διαχείριση της επικοινωνίας (*sockets*). Τέλος, το αρχείο *socket_handler.pl* περιέχει κώδικα για την διαχείριση των μηνυμάτων για την επικοινωνία με την βάση των παραμέτρων επίδοσης και την εξυπηρέτηση των ερωτήσεων για αυτήν. Οι τιμές των παραμέτρων στην βάση των παραμέτρων αποθηκεύονται σε μία δομή παρόμοια με αυτήν που χρησιμοποιούν οι διαδικασίες μέτρησης με την διαφορά ότι οι τιμές δεν αποθηκεύονται σε λίστα αλλά σε ένα πίνακα.

Κεφάλαιο 4. Εφαρμογές της Εποπτείας Επίδοσης.

Η ανάλυση επίδοσης και η αρχιτεκτονική του επόπτη επίδοσης παρέχουν την δυνατότητα για την ανάπτυξη μιας σειράς νέων μηχανισμών που βασίζονται στις τιμές των παραμέτρων επίδοσης του συστήματος βιβλιοθήκης που παράγει ο επόπτης επίδοσης. Οι μηχανισμοί αυτοί δίνουν στον διαχειριστή του συστήματος βιβλιοθήκης την δυνατότητα να διαγνώσει τα προβλήματα στην λειτουργία του και επιτρέπουν στον χρήστη να καθορίζει δυναμικά τις απαιτήσεις του με την προσφορά διαφόρων επιπέδων ποιότητας υπηρεσίας. Τέλος, καθιστούν το σύστημα βιβλιοθήκης ικανό να παρακολουθεί την λειτουργία και να την αναπροσαρμόζει δυναμικά.

4.1. Οπτικοποίηση παραμέτρων επίδοσης.

Μία βασική χρήση του επόπτη επίδοσης είναι για την παρακολούθηση των παραμέτρων επίδοσης του συστήματος ψηφιακής βιβλιοθήκης από τους χρήστες ή τους διαχειριστές του. Η παρακολούθηση αυτή μπορεί να γίνει με την χρήση κάποιου εργαλείου οπτικοποίησης των παραμέτρων επίδοσης.



Σχήμα 5: Οπτικοποίηση Παραμέτρων Επίδοσης.

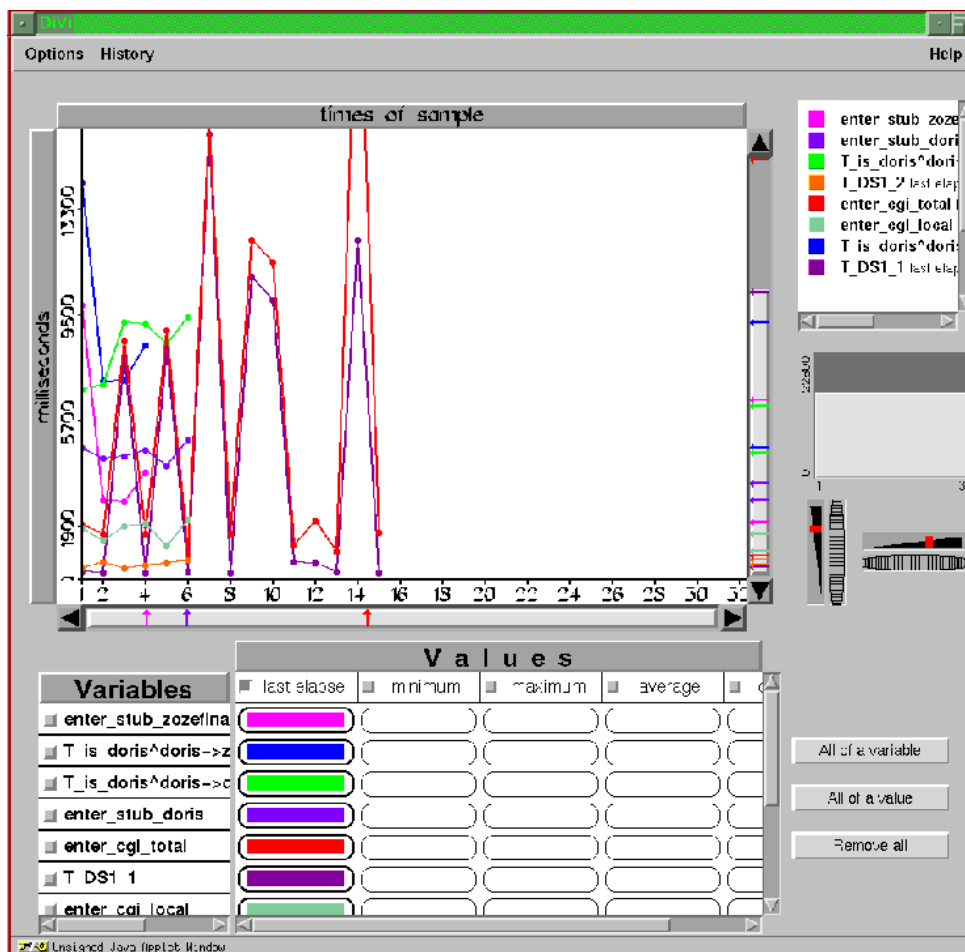
Η αρχιτεκτονική του επόπτη επίδοσης επιτρέπει την σύνδεση διαφόρων εργαλείων οπτικοποίησης με δύο διαφορετικούς τρόπους (βλέπε στο Σχήμα 5):

1. Σύνδεση στην δεύτερη διαδικασία μέτρησης. Με αυτό τον τρόπο το εργαλείο οπτικοποίησης παίρνει τα ίδια μηνύματα που παίρνει και η διαδικασία διαχείρισης

της βάσης των παραμέτρων επίδοσης, δηλαδή ενημερώνεται για κάθε νέα μέτρηση που παράγεται από τον επόπτη.

2. Αποστολή ερωτήσεων στον επόπτη και λήψη των παραμέτρων επίδοσης. Με αυτό τον τρόπο το εργαλείο οπτικοποίησης μπορεί να χρησιμοποιήσει όλες τις ερωτήσεις που περιγράφονται στο Παράρτημα 1. Με τον ίδιο τρόπο ενημερώνεται και το σύστημα βιβλιοθήκης για τις τιμές των παραμέτρων επίδοσης.

Ένα εργαλείο για την οπτικοποίηση των παραμέτρων επίδοσης του συστήματος βιβλιοθήκης με βάση τον επόπτη επίδοσης έχει υλοποιηθεί από τον Α.Χατζησταματίου σε *Java* (βλέπε στο Σχήμα 6), το οποίο και παρουσιάστηκε στο Πρώτο Ευρωπαϊκό Συνέδριο για την Έρευνα και την Προηγμένη Τεχνολογία των Ψηφιακών Βιβλιοθηκών (**First European Conference on Research and Advance Technology for Digital Libraries**), Πίζα Σεπτέμβριος 1997. Περισσότερες πληροφορίες για αυτό μπορούν να βρεθούν στο <http://www.csi.forth.gr/~dienst>.



Σχήμα 6: Εργαλείο Οπτικοποίησης εν λειτουργία των Παραμέτρων Επίδοσης.

4.2. Εγγύηση ποιότητας υπηρεσίας.

Πολλές φορές τα δεδομένα που περιέχονται σε μια ψηφιακή βιβλιοθήκη είναι σε πολλαπλές μορφές (**format**). Κάθε μορφή έχει συνήθως διαφορετικές απαιτήσεις πόρων και διαφορετική ποιότητα παρουσίασης. Έτσι, ο χρήστης από την μια προτιμά να πάρει την μορφή με την καλύτερη ποιότητα παρουσίασης από την άλλη όμως, δεν είναι πάντα διατεθειμένος να καταναλώσει ή δεν διαθέτει τους αναγκαίους πόρους. Επομένως, είναι χρήσιμο για αυτόν να μπορεί να προσδιορίζει το επίπεδο της υπηρεσίας που θέλει, δηλαδή την ποσότητα των πόρων που μπορούν να

χρησιμοποιηθούν. Ειδικότερα στην περίπτωση της αναζήτησης πληροφοριών τα διάφορα επίπεδα υπηρεσίας καθορίζονται από την πληρότητα της απάντησης (εάν θα ερωτηθούν ή όχι όλοι οι εξυπηρετητές του συστήματος) και την ακρίβεια της (ποιοι τρόποι αναζήτησης θα χρησιμοποιηθούν). Διαφορετικές επιλογές, όμως, συνεπάγονται και διαφορετικούς χρόνους απόκρισης του συστήματος.

Σε ένα σύστημα ψηφιακής βιβλιοθήκης με την βοήθεια του επόπτη επίδοσης μπορούμε να παρέχουμε στο χρήστη εκτιμήσεις για τον χρόνο που χρειάζονται διάφορες λειτουργίες, καθώς και εκτιμήσεις για τους πόρους (π.χ. μνήμη, χρόνος επεξεργαστή) που καταναλώνουν. Οι εκτιμήσεις αυτές μπορούν να δοθούν στον χρήστη ώστε να επιλέξει το κατάλληλο για αυτόν επίπεδο υπηρεσίας και να χρησιμοποιηθούν από το σύστημα για την εξασφάλιση του επιπέδου αυτού. Έτσι, με βάση τις εκτιμήσεις του επόπτη κάθε εξυπηρετητής του συστήματος βιβλιοθήκης μπορεί να υπολογίσει την πιθανότητα να ικανοποιήσει μια αίτηση μέσα στα πλαίσια των περιορισμών του χρήστη. Εάν η πιθανότητα αυτή προκύψει αρκετά μικρή τότε ο εξυπηρετητής μπορεί να ειδοποιήσει το σύστημα βιβλιοθήκης ότι αδυνατεί να ικανοποιήσει την αίτηση ώστε αυτό είτε να προωθήσει την αίτηση σε άλλον εξυπηρετητή μέχρι να ικανοποιηθεί, είτε να ειδοποιήσει τον χρήστη για την αδυναμία εξυπηρέτησης. Σε περίπτωση που το σύστημα βιβλιοθήκης διαπιστώσει αδυναμία ικανοποίησης των απαιτήσεων του χρήστη μπορεί να του δίνει μια εκτίμηση για τους πόρους που χρειάζεται, δίνοντας την δυνατότητα να αποφασίσει εάν μπορεί να τους διαθέσει. Τέλος, εφόσον το σύστημα βιβλιοθήκης μπορεί να παρακολουθήσει την κατανάλωση των πόρων του από τους χρήστες μπορεί να προχωρήσει και στην υλοποίηση μηχανισμών χρέωσης για τους πόρους αυτούς.

4.3. Εξισορρόπηση φόρτου.

Με τον όρο Εξισορρόπηση φόρτου εννοούμε ουσιαστικά την δυναμική δρομολόγηση των αιτήσεων του χρήστη μεταξύ των εξυπηρετητών του συστήματος βιβλιοθήκης. Τα συστήματα ψηφιακής βιβλιοθήκης είναι φύσει δυναμικά συστήματα. Επομένως, μια στατική στρατηγική για την δρομολόγηση των αιτήσεων δεν μπορεί να είναι βέλτιστη αφού αγνοεί την δυναμική του συστήματος.

Ο επόπτης επίδοσης με τις τιμές των παραμέτρων επίδοσης που παράγει δίνει την δυνατότητα στο σύστημα βιβλιοθήκης να εκτιμά τον χρόνο απόκρισης των εξυπηρετητών που το αποτελούν. Έτσι, το σύστημα βιβλιοθήκης μπορεί να εφαρμόσει πολιτικές δυναμικής δρομολόγησης των αιτήσεων, αφού του παρέχεται ο κατάλληλος μηχανισμός (μπορεί να ενημερωθεί για τις τιμές των παραμέτρων επίδοσης και τις εκτιμήσεις για την εξέλιξη τους που παράγει ο επόπτης). Στόχος των πολιτικών αυτών είναι η ελαχιστοποίηση του χρόνου απόκρισης του συστήματος, η οποία επιτυγχάνεται με την δρομολόγηση της αίτησης προς τους εξυπηρετητές εκείνους που έχουν τον μικρότερο εκτιμώμενο χρόνο απόκρισης.

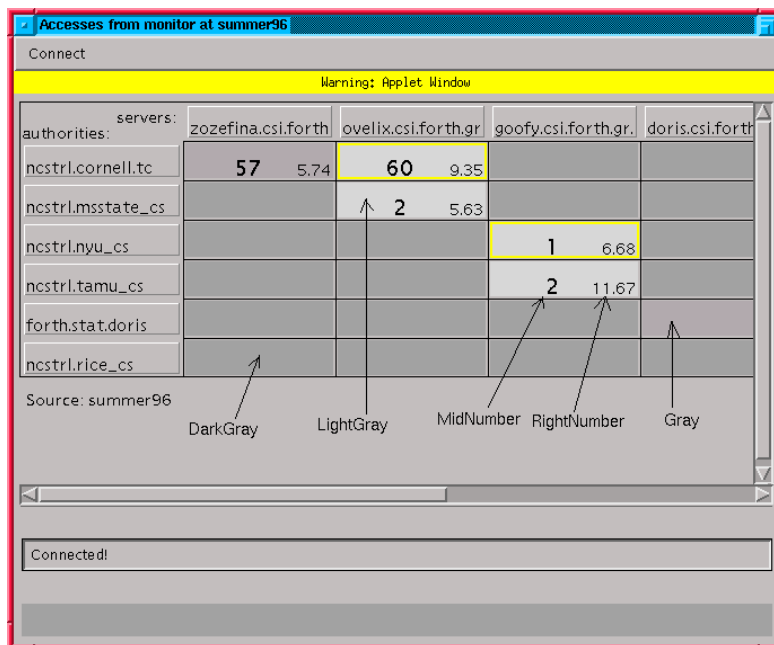
Οι μηχανισμοί για την απόκτηση των τιμών των παραμέτρων επίδοσης είναι οι παρακάτω:

- **Περιοδική Δειγματοληψία (Polling):** Το σύστημα βιβλιοθήκης μπορεί να στέλνει περιοδικά κάποιο μήνυμα σε όλους τους εξυπηρετητές που το αποτελούν για να μετρήσει την καθυστέρηση του δικτύου ή μπορεί να στέλνει και ψευδοαιτήσεις για να μετρά και τον χρόνο επεξεργασίας για κάθε εξυπηρετητή. Η περιοδική δειγματοληψία δεν χρησιμοποιείται πολύ γιατί είναι ιδιαίτερα δαπανηρή και δημιουργεί σημαντικό αριθμό μηνυμάτων.
- **Εξέταση (Probing):** Το σύστημα βιβλιοθήκης πριν από την εξυπηρέτηση κάθε αίτησης στέλνει μία **αίτηση-εξέταση** στους εξυπηρετητές και αυτοί

απαντούν με τον μέσο χρόνο επεξεργασίας των αιτήσεων και την τρέχουσα κατάσταση τους. Το μειονέκτημα αυτής της μεθόδου είναι ότι όλοι οι εξυπηρετητές πρέπει να ρωτηθούν γεγονός που αυξάνει σημαντικά τον χρόνο απόκρισης του συστήματος.

- *Υπηρεσίες Ονοματοδοσίας (Name Services)*: Κάθε εξυπηρετητής διαφημίζει τους χρόνους απόκρισης του στέλνοντας τους σε ειδικούς εξυπηρετητές ονοματοδοσίας. Στην περίπτωση αυτή έχουμε ουσιαστικά ένα συνδυασμό των παραπάνω μεθόδων, αφού οι εξυπηρετητές και στέλνουν περιοδικά στοιχεία και παίρνουν στοιχεία όταν χρειάζονται ενημέρωση, μόνο που μεσολαβούν οι εξυπηρετητές ονοματοδοσίας. Η λύση αυτή ενδείκνυται ιδιαίτερα για μεγάλης κλίμακας συστήματα. Μάλιστα, οι εξυπηρετητές ονοματοδοσίας μπορούν να οργανωθούν με καταναμημένο τρόπο, ουσιαστικά ένας εξυπηρετητής για κάθε ομάδα «κοντινών» εξυπηρετητών.

Για την οπτικοποίηση εν λειτουργία των επιλογών της δρομολόγησης που προκύπτει από την εξισορρόπηση φόρτου κατασκευάστηκε από τον Γιώργο Σαπουντζή σε *Java*, ένα γραφικό εργαλείο (βλέπε στο Σχήμα 7).



Σχήμα 7: Οπτικοποίηση εν λειτουργία των Επιλογών Δρομολόγησης του Αλγορίθμου Εξισορρόπησης Φόρτου.

4.4. Δυναμική αναπροσαρμογή των χρονικών περιθωρίων.

Η λειτουργία κάθε συστήματος ψηφιακής βιβλιοθήκης βασίζεται σε ορισμένα χρονικά περιθώρια. Αυτά τα χρονικά περιθώρια εξασφαλίζουν ότι προβλήματα στην λειτουργία του συστήματος δεν θα το μπλοκάρουν έπ' άπειρον, βασικά όταν χρησιμοποιείται το δίκτυο. Επίσης, η ανάλυση του συστήματος δείχνει ότι τα χρονικά αυτά περιθώρια επηρεάζουν σημαντικά την επίδοση του συστήματος. Αυτό συμβαίνει γιατί πολύ μικρά χρονικά περιθώρια κάνουν αδύνατη την ικανοποίηση πολύπλοκων ερωτήσεων ή την χρήση απομακρυσμένων εξυπηρετητών, ενώ από την άλλη πολύ μεγάλα χρονικά περιθώρια επιδεινώνουν τον χρόνο απόκρισης και μειώνουν τον χρήσιμο χρόνο του συστήματος (**utilization**), μια που αυτό είναι αναγκασμένο να περιμένει για μεγάλα χρονικά διαστήματα εξυπηρετητές που δεν είναι

προσπελάσιμοι. Σε μεγάλης κλίμακας συστήματα είναι συνηθισμένο να υπάρχουν εξυπηρετητές που δεν είναι προσπελάσιμοι κατά την διάρκεια επεξεργασίας μίας ερώτησης. Οι βέλτιστες τιμές των χρονικών αυτών περιθωρίων είναι δύσκολο να βρεθούν. Από την άλλη, το γεγονός ότι οι τιμές αυτές ορίζονται κατά την διαμόρφωση του συστήματος περιπλέκει το πρόβλημα μια που αυτό είναι ουσιαστικά ασυμβίβαστο με την δυναμική φύση των συστημάτων ψηφιακής βιβλιοθήκης, αφού λειτουργούν πάνω από ένα συνεχώς μεταβαλλόμενο δίκτυο.

Μια λύση του παραπάνω προβλήματος είναι η δυναμική αναπροσαρμογή των χρονικών περιθωρίων. Από την στιγμή που ο επόπτης επίδοσης κρατά την ιστορία του χρόνου που χρειάζονται τα διάφορα τμήματα του συστήματος βιβλιοθήκης και ιδιαίτερα τον χρόνο απόκρισης των διαφόρων εξυπηρετητών που το απαρτίζουν, μια λογική τιμή για τα χρονικά περιθώρια θα ήταν:

$$\min\{T_{O_{user}}, \max\{T_{est_i}\} + T_{SI}\}$$

(1) Δυναμικά Προσαρμοζόμενο Χρονικό Περιθώριο.

όπου $T_{O_{user}}$ είναι πάνω όριο για το χρονικό περιθώριο που θέτει ο χρήστης, T_{est_i} είναι η εκτίμηση του χρόνου για το τμήμα i και T_{SI} είναι ένα περιθώριο ασφαλείας. Ο σκοπός του περιθωρίου ασφαλείας είναι να διατηρεί την πιθανότητα να ολοκληρωθεί η εκτέλεση κάποιου τμήματος, πριν το πέρας του χρονικού περιθωρίου, πάνω από ένα όριο που ορίζει ο χρήστης. Έτσι, εάν είναι γνωστή η μέση τιμή και η διασπορά του χρόνου για την εκτέλεση ενός τμήματος, τότε η

$$P[|x - E(x)| > c] \leq \sigma^2(x)/c^2$$

(2) Ανισότητα Tchebisev.

όπου x είναι ο τρέχων χρόνος εκτέλεσης ενός τμήματος, $E(x)$ και $\sigma(x)$ είναι η τρέχουσα μέση τιμή και τυπική απόκλιση του χρόνου εκτέλεσης του τμήματος, c είναι μία σταθερά που δίνεται από τον χρήστη, μπορεί να μας δώσει το διάστημα ασφαλείας που εγγυάται ότι η πιθανότητα να ολοκληρωθεί η εκτέλεση του τμήματος μέσα στο δεδομένο χρονικό περιθώριο είναι μεγαλύτερη του ορίου του χρήστη.

Κεφάλαιο 5. Το Σύστημα NCSTRL – Dienst.

Το σύστημα NCSTRL – Dienst είναι ένας Κατανεμημένος, Αλληλεπιδραστικός, Επεκτάσιμος, Δικτυακός Εξυπηρετητής για Τεχνικές Αναφορές (**Distributed Interactive Extensible Network Server for Techreports**), που ξεκίνησε ως ερευνητική προσπάθεια στα πλαίσια του προγράμματος CS-TR από την ομάδα ψηφιακών βιβλιοθηκών του αμερικανικού πανεπιστημίου Cornell και ερευνητές της Xerox Corporation. Βασικός σκοπός του συστήματος ήταν να προσφέρει ένα πλαίσιο για την διαχείριση συλλογών ψηφιακών εγγράφων διαφορετικών τύπων, καθώς και ένα πρωτόκολλο για την πρόσβαση στις συλλογές αυτές μέσω του Παγκόσμιου Δικτυακού Ιστού (**World Wide Web**). Για τον χρήστη το σύστημα παρουσιάζεται ως μια ενοποιημένη συλλογή εγγράφων, τα οποία είναι διαθέσιμα σε μια πληθώρα μορφών (**formats**). Με την χρήση κοινόχρηστων πελατών του δικτυακού ιστού (**web clients**), οι χρήστες έχουν την δυνατότητα να ψάχνουν την συλλογή, να αναδιφούν (**browse**) και να διαβάζουν διάφορα έγγραφα σε όποια από τις διαθέσιμες μορφές, καθώς και να αποκτούν (**retrieve**) ή να τυπώνουν κάποιο έγγραφο. Από την άλλη, παρέχει στους διαχειριστές των διαφόρων συστημάτων εργαλεία για την διαχείριση της συλλογής τους.

Η διάδοση του Dienst οδήγησε στην υιοθέτηση του από το NCSTRL. Μετά από μια σειρά από αλλαγές στην αρχιτεκτονική και την υλοποίηση του συστήματος [8], το Dienst εναρμονίστηκε με τις βασικές αρχές της αρχιτεκτονικής του NCSTRL, και αποτελεί, σήμερα, την βασική πλατφόρμα έρευνας του NCSTRL.

5.1. Η βασική αρχιτεκτονική του συστήματος *Dienst*.

Η αρχιτεκτονική του συστήματος αποτελείται από τρία εξαρτήματα: ένα πρωτόκολλο για το ψάξιμο, την αναδίφηση και την παρουσίαση των εγγράφων, έναν εξυπηρετητή που υλοποιεί το πρωτόκολλο και συνεργάζεται με τους άλλους εξυπηρετητές, και ένα σύνολο από βοηθητικά εργαλεία για την διαχείριση της συλλογής και την διαχείριση του εξυπηρετητή. Η αρχιτεκτονική του συστήματος αποτελεί ένα επίπεδο λειτουργικότητας πάνω από αυτό του παγκόσμιου ιστού. Το επίπεδο αυτό προσφέρει μια σειρά από σημαντικές αφαιρέσεις, όπως ο ορισμός των συλλογών εγγράφων τα οποία μπορούμε να ψάξουμε και να δούμε, των μοναδικών κωδικών εγγράφων (**docids**) που επιτρέπουν την πρόσβαση στις διάφορες μορφές εγγράφων, και τις αποσυνθέσεις (**decompositions**) εγγράφων που επιτρέπουν την πρόσβαση στις σελίδες ή τα άλλα λογικά μέρη ενός εγγράφου.

5.1.1. Το μοντέλο εγγράφων.

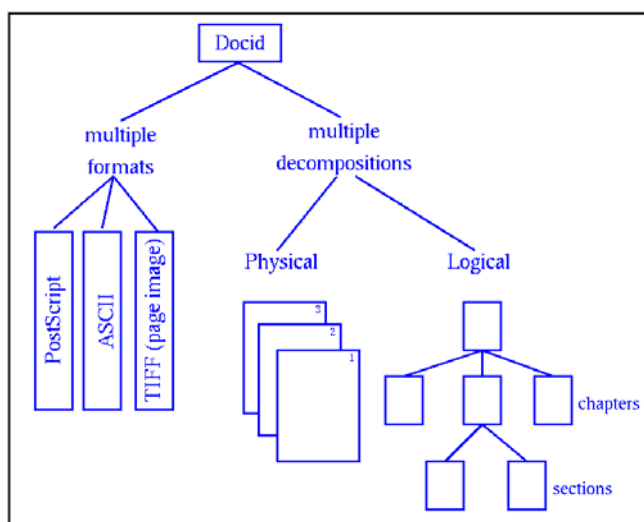
Το μοντέλο εγγράφων του Dienst παρέχει τρεις σημαντικές αφαιρέσεις: μοναδικούς κωδικούς εγγράφων (**docids**), πολλαπλές μορφές (**formats**) εγγράφων και αποσυνθέσεις (**decompositions**) εγγράφων (βλέπε το Σχήμα 8).

Κάθε έγγραφο μιας συλλογής αναγνωρίζεται από ένα μοναδικό, ανεξάρτητο της τοποθεσίας κωδικό εγγράφου, το docid. Το docid αποτελείται από δύο μέρη:

- ένα μοναδικό όνομα εκδότη,
- ένα μοναδικό για κάθε εκδότη κωδικό εγγράφου.

Η τελική μορφή του docid καθορίζεται από κάθε εκδότη. Η αντιστοίχιση του docid σε μια τοποθεσία για την δεικτοδότηση και την αποθήκευση γίνεται μέσω ενός κεντρικού καταλόγου δεικτοδότησης εγγραφών (**index**), που κάθε εξυπηρετητής αποκτά. Η αντιστοίχιση είναι διαφανής για τον χρήστη και η συλλογή παρουσιάζεται σε αυτόν ως ένα ενιαίο σύνολο εγγράφων.

Το docid αποτελεί ένα μέσο για την ομαδοποίηση των διαφόρων μορφών κάθε εγγράφου. Το docid επιτρέπει στον χρήστη του Dienst να βλέπει τις διάφορες μορφές αναπαράστασης των εγγράφων ως υποεξαρτήματα μίας μοναδικής οντότητας, παρόλο που αυτές οι μορφές μπορεί στην πραγματικότητα να είναι διαφορετικά αρχεία σε διάφορους καταλόγους.

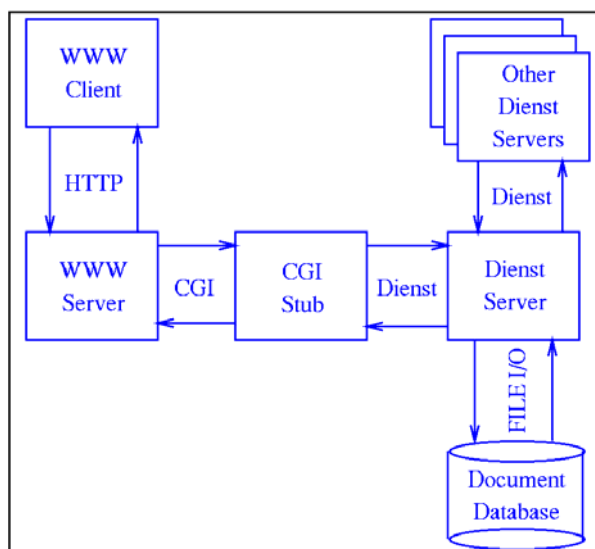


Σχήμα 8: Το Μοντέλο Εγγράφων του Dienst.

5.1.2. Η αρχιτεκτονική του εξυπηρετητή.

Κάθε εξυπηρετητής αποτελείται από τέσσερα εξαρτήματα: μια βάση εγγράφων, τον ίδιο τον εξυπηρετητή, τον εξυπηρετητή του παγκόσμιου ιστού και το Common Gateway Interface (CGI) stub του Dienst. Η σχέση των παραπάνω εξαρτημάτων παρουσιάζεται στο Σχήμα 9.

βάση εγγράφων - Ένα σύνολο από κείμενα που προσφέρει ένας εξυπηρετητής Dienst. Τα κείμενα είναι αποθηκευμένα ως αρχεία. Κάθε κείμενο μπορεί να είναι αποθηκευμένο σε πολλαπλές μορφές. Κάθε κείμενο πρέπει να περιγράφεται τουλάχιστον από μια βιβλιογραφική εγγραφή. Ο εξυπηρετητής γνωρίζει κανόνες για την αντιστοίχιση μιας πλειάδας του τύπου docid, μορφή εγγράφου και προαιρετικά αποσύνθεση, στο πραγματικό αρχείο της βάσης.



Σχήμα 9: Η Αλληλεπίδραση των Εξαρτημάτων του Συστήματος Dienst.

εξυπηρετητής Dienst - Ο εξυπηρετητής είναι μία πολυνηματική διεργασία, γραμμένη σε *Perl*. Ο εξυπηρετητής περιμένει συνδέσεις σε μια συγκεκριμένη πόρτα μηνυμάτων (**message port**). Οι έγκυρες αιτήσεις καθορίζονται από το πρωτόκολλο του Dienst. Οι αποκρίσεις είναι σε μορφή πρωτοκόλλου Μεταφοράς Υπερκειμένου (**Hyper Text Transfer Protocol - HTTP**) απαντήσεων και μεταφέρονται στον αιτούντα πελάτη χωρίς επεξεργασία από τον εξυπηρετητή του παγκόσμιου ιστού και το CGI stub πρόγραμμα. Η εξυπηρέτηση μιας αίτησης μπορεί να περιλαμβάνει την προώθηση της σε άλλους εξυπηρετητές και την συλλογή των απαντήσεων τους. Κάθε εξυπηρετητής Dienst παρέχει κυρίως τρεις υπηρεσίες.

- *υπηρεσία αποθήκευσης (repository service)* - παρέχει πρόσβαση στα κείμενα της βάσης εγγράφων. Όλα τα κείμενα ενός εκδότη πρέπει να βρίσκονται στην ίδια βάση κειμένων, ενώ μια βάση εγγράφων μπορεί να περιέχει κείμενα από διάφορους εκδότες.
- *υπηρεσία καταλόγου δεικτοδότησης (index service)* - επιτρέπει την αναζήτηση κειμένων της βάσης. Ο κατάλογος κατασκευάζεται off-line από τις βιβλιογραφικές εγγραφές (**bibliographic records**) των κειμένων. Ο κατάλογος φορτώνεται στην μνήμη του εξυπηρετητή κατά την εκκίνηση του. Η αναζήτηση γίνεται είτε με λογικές εκφράσεις των πεδίων αναζήτησης ή με αναζήτηση στο σύνολο του κειμένου (**full-text**).
- *υπηρεσία διεπιφάνειας χρήσης (user interface service)* - παρέχει μια φιλική διεπιφάνεια χρήσης σε Γλώσσα Σημείωσης Υπερκειμένου (**Hyper Text Markup Language - HTML**) για την πρόσβαση στις υπηρεσίες καταλόγου δεικτοδότησης και αποθήκευσης.

εξυπηρετητής παγκόσμιου ιστού - το σύστημα είναι σχεδιασμένο ώστε η πρόσβαση σε αυτό να γίνεται μέσω εξυπηρετητών παγκόσμιου ιστού. Η πρόσβαση είναι δυνατή μέσω οποιουδήποτε εξυπηρετητή υποστηρίζει το CGI. Οι αιτήσεις του Dienst περιέχονται στο path πεδίο της HTTP αίτησης προς τον εξυπηρετητή παγκόσμιου ιστού. Ο εξυπηρετητής πρέπει να διαμορφωθεί ώστε να καλεί το CGI stub του Dienst όταν λαμβάνει μια Dienst αίτηση.

Dienst CGI stub - Το stub λειτουργεί ως ενδιάμεσος για τον εξυπηρετητή Dienst. Είναι ένα μικρό και απλό πρόγραμμα που εξάγει την Dienst αίτηση

από την HTTP αίτηση για να την προωθήσει στο Dienst μαζί με κάποιες μεταβλητές του περιβάλλοντος. Η προώθηση γίνεται μέσω ενός socket.

Όπως αναφέραμε παραπάνω ο χρήστης του Dienst βλέπει το σύνολο των συλλογών εγγράφων σαν μία ενιαία συλλογή, παρόλο που αυτή είναι κατανομημένη σε διάφορους εξυπηρετητές. Αυτό επιτυγχάνεται με την επικοινωνία ανάμεσα στους εξυπηρετητές Dienst σε τρία επίπεδα λειτουργικότητας:

εγγραφή εξυπηρετητή - Όπως είπαμε προηγουμένως τα κείμενα αναγνωρίζονται με την χρήση των docids, που αποτελούνται από ένα όνομα εκδότη και ένα όνομα εγγράφου. Κάθε ένας από τους εξυπηρετητές διαθέτει έναν κατάλογο δεικτοδότησης (**index**) και λειτουργεί ως αποθήκη για έναν ή περισσότερους εκδότες. Επίσης, κάθε εξυπηρετητής μπορεί να βρει που βρίσκεται ο κατάλογος δεικτοδότησης και η αποθήκη (**repository**) για οποιοδήποτε κείμενο με βάση κάποιους πίνακες που τηρεί ένας ιδιαίτερος εξυπηρετητής, ο meta εξυπηρετητής, και τους οποίους παίρνει περιοδικά. Αυτό επιτρέπει κάθε εξυπηρετητή να παρέχει τις ακόλουθες δύο λειτουργίες.

κατανομημένη αναζήτηση - Κάθε εξυπηρετητής Dienst επιτρέπει μέσω της διεπιφάνειας χρήσης του την αναζήτηση εγγράφων στο σύνολο των συλλογών των διαφόρων εξυπηρετητών. Ο χρήστης έχει την δυνατότητα να προσδιορίσει τους εκδότες στους οποίους θα γίνει η αναζήτηση. Κάθε εξυπηρετητής, επομένως, προσδιορίζει τους εξυπηρετητές που πρέπει να επικοινωνήσει και τους προωθεί την αίτηση αναζήτησης. Στην συνέχεια συγκεντρώνει τα αποτελέσματα των διαφόρων εξυπηρετητών, τα μορφοποιεί και τα παρουσιάζει στον χρήστη σαν μια λίστα επιτυχιών.

κατανομημένη πρόσβαση κειμένων - Κάθε εξυπηρετητής Dienst μπορεί να ικανοποιήσει αιτήσεις για οποιοδήποτε έγγραφο της συλλογής. Αυτό επιτυγχάνεται με την ανάλυση του docid, τον προσδιορισμό της αποθήκης για τον εκδότη του εγγράφου μέσω των πινάκων του meta εξυπηρετητή και την προώθηση της αίτησης σε αυτόν. Η όλη διαδικασία είναι διαφανής για τον χρήστη.

5.1.3. Η διεπιφάνεια χρήσης του συστήματος.

Η διεπιφάνεια χρήσης του συστήματος Dienst παρέχει μια σειρά από λειτουργίες προς τον χρήστη (στο Παράρτημα 3 παρουσιάζεται ένα παράδειγμα χρήσης του συστήματος). Ορισμένες από αυτές τις λειτουργίες είναι προαιρετικές και δεν παρέχονται από όλους τους εξυπηρετητές, άλλες εξαρτώνται από τις μορφές στις οποίες είναι αποθηκευμένο κάθε έγγραφο. Οι διαθέσιμες λειτουργίες είναι οι εξής:

- Αναζήτηση με βάση HTML φόρμες
- Ενοποιημένη λίστα επιτυχιών με την μορφή υπερκειμένου
- Περιληπτική σελίδα κειμένου
- Παρουσίαση της δομής του κειμένου
- Ξεφύλλισμα κειμένου μέσω μικρών εικόνων (**thumbnail images**)
- Γραφική εικόνα των σελίδων κειμένου
- Εστίαση στις σελίδες του κειμένου
- Αναζήτηση στο σύνολο του κειμένου (**full-text**)
- Ανάκληση και Εκτύπωση κειμένου

5.2. Οι NCSTRL μετατροπές του συστήματος.

Όπως έχουμε ήδη αναφέρει, μια σειρά από μετατροπές στην αρχιτεκτονική και την υλοποίηση του συστήματος Dienst ήταν αναγκαίες για την συμμόρφωση του

με την αρχιτεκτονική του NCSTRL. Οι μετατροπές αυτές εστιάστηκαν σε δύο κυρίως θέματα, την οργάνωση μιας αρχιτεκτονικής εξυπηρετητών περιοχής (**regional server architecture**) και την αντικατάσταση των docids από ένα σύστημα αναγνωριστικών εγγράφων (**handle system**), όπως αυτό περιγράφεται στο [8].

Τέλος, εισήχθη στο σύστημα Dienst και η έννοια των εξυπηρετητών Dienst-Lite. Πρόκειται για κόμβους που διαθέτουν αποθήκη εγγράφων αλλά δεν διαθέτουν τον εξυπηρετητή Dienst. Οι αιτήσεις για έγγραφα των κόμβων αυτών εξυπηρετούνται από ένα κεντρικό εξυπηρετητή που βρίσκεται στο Reston της Virginia, έδρα της επιτροπής καθοδήγησης (**steering committee**) του NCSTRL. Οι κόμβοι αυτοί είναι οι παλιοί κόμβοι του συστήματος WATERS, που ενσωματώθηκε και αυτό στο σύστημα NCSTRL.

5.2.1. Η αρχιτεκτονική των εξυπηρετητών περιοχής.

Η αρχιτεκτονική των εξυπηρετητών περιοχής διαμορφώθηκε σε δύο φάσεις. Στην πρώτη φάση εισήχθη στο σύστημα Dienst η έννοια του εξυπηρετητή υποστήριξης (**backup server**) με την υιοθέτηση του συστήματος από το NCSTRL. Στην συνέχεια, αναπτύχθηκε η πλήρης αρχιτεκτονική εξυπηρετητών περιοχής ως αποτέλεσμα της συμμετοχής του ERCIM στην ομάδα εργασίας του NCSTRL.

Οι εξυπηρετητές υποστήριξης είναι ουσιαστικά εξυπηρετητές Dienst οι οποίοι δεν διαθέτουν δική τους αποθήκη εγγράφων αλλά καταλόγους δεικτοδότησης εγγράφων άλλων κόμβων. Οι εξυπηρετητές αυτοί χρησιμοποιούνται όταν αποτύχουμε ένα προκαθορισμένο αριθμό φορές να επικοινωνήσουμε με κάποιο εξυπηρετητή Dienst. Ο εξυπηρετητής υποστήριξης χρησιμοποιείται μέχρι να περάσει κάποιο προκαθορισμένο χρονικό διάστημα, οπότε ξαναχρησιμοποιείται ο αρχικός εξυπηρετητής Dienst. Αρχικά το σύστημα είχε ένα μόνο εξυπηρετητή υποστήριξης στο Reston της Virginia.

Στην συνέχεια, μετά την υιοθέτηση του NCSTRL από το ERCIM, για την αντιμετώπιση των προβλημάτων της δικτύωσης με την Ευρώπη αναπτύχθηκε η αρχιτεκτονική των περιοχών. Σύμφωνα με αυτήν, ορίζονται περιοχές (**regions**) από γειτονικούς εξυπηρετητές (ο όρος «γειτονικός» ουσιαστικά υπονοεί καλής ποιότητας δικτύωση μεταξύ των εξυπηρετητών). Κάθε περιοχή περιλαμβάνει εκτός από τους κλασικούς εξυπηρετητές Dienst, ένα meta εξυπηρετητή για την περιοχή (**Regional Meta Server**) και ένα εξυπηρετητή συγχωνευμένου καταλόγου δεικτοδότησης (**Merged Index Server**).

Ο ρόλος του meta εξυπηρετητή της περιοχής είναι να διευκολύνει την ανανέωση των καταλόγων δεικτοδότησης των εγγράφων. Έτσι, κάθε εξυπηρετητής Dienst αντί να επικοινωνεί με ένα κεντρικό meta εξυπηρετητή, επικοινωνεί με τον meta εξυπηρετητή της περιοχής για την ανανέωση του καταλόγου δεικτοδότησης του. Ο meta εξυπηρετητής κάθε περιοχής επικοινωνεί περιοδικά με τον κύριο meta εξυπηρετητή (**Master Meta Server**) που βρίσκεται στο πανεπιστήμιο Cornell, για την ενημέρωση των πινάκων του.

Ο εξυπηρετητής συγχωνευμένου καταλόγου είναι ουσιαστικά ένας εξυπηρετητής Dienst χωρίς δική του αποθήκη εγγράφων. Ο εξυπηρετητής αυτός διαθέτει ουσιαστικά τους καταλόγους δεικτοδότησης (**indices**) όλων των εξυπηρετητών Dienst, που δεν ανήκουν στην περιοχή του. Έτσι, όταν ένας εξυπηρετητής Dienst χρειάζεται να προωθήσει μια υποερώτηση σε έναν εξυπηρετητή εκτός της περιοχής του, αντί να επικοινωνήσει απ' ευθείας μαζί του, προωθεί την υποερώτηση στον εξυπηρετητή συγχωνευμένου καταλόγου της περιοχής. Σε περίπτωση που ο εξυπηρετητής συγχωνευμένου καταλόγου μιας περιοχής δεν είναι διαθέσιμος κάποια στιγμή, τότε οι υποερωτήσεις που θα προορίζονταν σε αυτόν

προωθούνται στους κυρίως εξυπηρετητές Dienst (περισσότερες λεπτομέρειες για την διαδικασία δρομολόγησης των υποερωτήσεων θα παρουσιαστούν στο κεφάλαιο που αναφέρεται στις στρατηγικές εξισορρόπησης φόρτου).

Τέλος, κάθε περιοχή μπορεί να διαθέτει και ένα εξυπηρετητή υποστήριξης για όλους τους εξυπηρετητές της περιοχής.

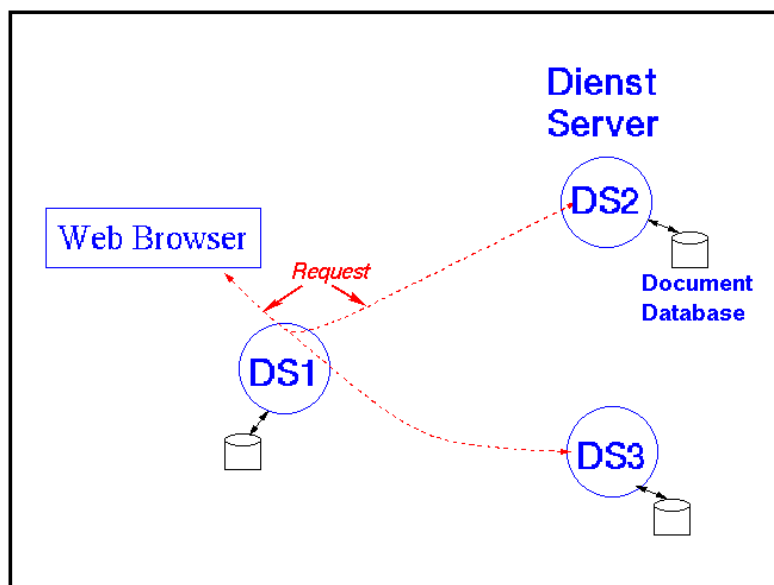
5.2.2. Το σύστημα των αναγνωριστικών εγγράφων.

Η μετατροπή των docids του συστήματος Dienst σε αναγνωριστικά εγγράφων του συστήματος NCSTRL γίνεται με άμεσο τρόπο. Τα ονόματα των εκδοτών του συστήματος Dienst θεωρούνται ονόματα αρχών έκδοσης (**authorities**) του συστήματος NCSTRL. Οι μοναδικοί κωδικοί των εγγράφων για κάθε εκδότη διατηρούνται ως αναγνωριστικά εγγράφων της αρχής έκδοσης. Το σύστημα των αναγνωριστικών εγγράφων έχει πρόβλεψη και για τις περιπτώσεις όπου υπάρχει μια ιεραρχία στις αρχές έκδοσης εγγράφων, π.χ. η αρχή έκδοσης του πανεπιστημίου Cornell και η αρχή έκδοσης του Κέντρου Θεωρίας του ίδιου πανεπιστημίου. Στην περίπτωση αυτή, το όνομα της αρχής έκδοσης είναι δυνατό να αποτελείται από επιμέρους τμήματα που διαχωρίζονται με τελείες.

Η μοναδικότητα των αναγνωριστικών εγγράφων εξασφαλίζεται μέσω ενός εξυπηρετητή αναγνωριστικών (**handle server**), που βρίσκεται στο Reston της Virginia. Κάθε αρχή έκδοσης όταν εισάγει ένα καινούργιο έγγραφο στην βιβλιοθήκη πρέπει να το αναφέρει στον εξυπηρετητή αναγνωριστικών για να πάρει το αναγνωριστικό του. Ο εξυπηρετητής αναγνωριστικών λειτουργεί ουσιαστικά ως εγγυητής για τα έγγραφα της βιβλιοθήκης. Από την άλλη, κάθε αρχή έκδοσης μπορεί να διαθέτει και έγγραφα χωρίς να έχουν εγκριθεί από τον εξυπηρετητή εγγράφων. Τα έγγραφα αυτά θεωρείται ότι είναι προσωρινά και το σύστημα της βιβλιοθήκης δεν παρέχει καμία εγγύηση για αυτά.

Κεφάλαιο 6. Ανάλυση Επίδοσης του Συστήματος *NCSTRL – Dienst.*

Στο κεφάλαιο αυτό ξεκινώντας από ένα απλό παράδειγμα της λειτουργίας του συστήματος NCSTRL – Dienst, προχωρούμε στην μοντελοποίηση του εξυπηρετητή Dienst για μια αίτηση. Με την χρήση του παραγόμενου μοντέλου αναλύεται ο χρόνος που χρειάζονται τα διάφορα στάδια της λειτουργίας του εξυπηρετητή κατά την εξυπηρέτηση μιας αίτησης. Σκοπός αυτής της ανάλυσης είναι ο προσδιορισμός των παραμέτρων που θέλουμε να παρακολουθεί ο επόπτης επίδοσης του συστήματος.



Σχήμα 10: Παράδειγμα Λειτουργίας του Συστήματος Dienst.

6.1. Μοντελοποίηση μιας αίτησης αναζήτησης.

Η μοντελοποίηση και η ανάλυση έχουν ως εξής: Στο Σχήμα 11 (παρακάτω) φαίνεται με λεπτομέρειες η ροή εξυπηρέτησης μιας αίτησης στο σύστημα στο Σχήμα 10. Υπάρχουν τρεις εξυπηρετητές Dienst: DS1, DS2 και DS3. Κάποιος χρήστης υποβάλλει μια αίτηση αναζήτησης στον εξυπηρετητή DS1, η οποία δρομολογείται στους μακρινούς εξυπηρετητές DS2 και DS3.

μετρηθούν. Κάθε τμήμα συνδέεται με ένα σύμβολο χρόνου, το οποίο δεν είναι τίποτε άλλο παρά μια μεταβλητή. Το μοντέλο αναλύεται βήμα προς βήμα ακολούθως.

Σημείωση: Η μεταβλητή T_X^Y , σημαίνει ότι T είναι ο χρόνος που χρειάζεται ένα τμήμα, ενώ το X προσδιορίζει το τμήμα και το Y μια δράση ή μια λειτουργία στα πλαίσιά του.

6.1.1. Τμήματα υποβολής αίτησης.

- T_{nds}^{total} : Ο χρόνος που χρειάζεται το CGI stub του Dienst (**nph-dienst-stub** – **NDS**) για να επεξεργαστεί μία αίτηση από τον πελάτη του παγκόσμιου ιστού (π.χ. **Web Browser**). Ο χρόνος αυτός περιλαμβάνει τον χρόνο μετάδοσης της αίτησης στο $DS1_0$, την βασική διεργασία του εξυπηρετητή Dienst, και τον χρόνο για την παρουσίαση της απάντησης σε μορφή γλώσσας υπερκειμένου στον πελάτη του παγκόσμιου ιστού. Σημείωση: Ο εξυπηρετητής του παγκόσμιου ιστού και ο εξυπηρετητής Dienst μπορεί να βρίσκονται σε διαφορετικά μηχανήματα.

$$\bullet T_{nds}^{total} = T_{nds \rightarrow DS1_0}^{socket} + T_{DS1_0}^{fork1} + T_{DS1_1}^{first} \left(\text{or } T_{DS1_1}^{last} \right)$$

ο $T_{nds \rightarrow DS1_0}^{socket}$: Ο χρόνος που χρειάζεται η μεταφορά της αίτησης από το nph-dienst-stub στο $DS1_0$ μέσω του socket.

ο $T_{DS1_0}^{fork1}$: Ο χρόνος που χρειάζεται το $DS1_0$ για να κάνει fork το $DS1_1$, που θα εξυπηρετήσει την αίτηση.

ο $T_{DS1_1}^{first}$: Ο χρόνος που χρειάζεται το $DS1_1$ για να υποβάλλει την αίτηση αναζήτησης και να στείλει τα πρώτα αποτελέσματα στο NDS. Σημείωση: Το $DS1_1$ στέλνει τα αποτελέσματα στο NDS καθώς φτάνουν από τις IS διεργασίες.

ο $T_{DS1_1}^{last}$: Ο χρόνος που χρειάζεται το $DS1_1$ για να υποβάλλει και να ολοκληρώσει την παράλληλη αναζήτηση και να στείλει όλη την σελίδα αποτελεσμάτων.

6.1.2. Εκκίνηση παράλληλης αναζήτησης.

$$\bullet T_{DS1_1}^{first} = T_{DS1_1}^{fork_{is}} + \min \left\{ T_{search}^{O}, \max \left\{ T_{is1}^{A \rightarrow B}, T_{is2}^{C \rightarrow D}, T_{is3}^{E \rightarrow F} \right\} \right\}$$

$\circ T_{DS1_1}^{fork_{is}}$: Ο χρόνος που χρειάζεται το DS1₁ για να κάνει fork όλα τα indexer_stubs (είναι stubs για την προώθηση των αιτήσεων αναζήτησης στους εξυπηρετητές δεικτοδότησης) : IS₁, IS₂ και IS₃.

$\circ TO^{search}$: Το συνολικό μέγιστο χρονικό περιθώριο αναζήτησης (**search_timeout**).

$\circ T_{is_1}^{A \rightarrow B}$: Ο χρόνος που χρειάζεται το IS₁ για να στείλει την ερώτηση στο DS1₀, να πάρει τα αποτελέσματα και να τα προωθήσει στο DS1₁. Αντίστοιχα για τις υπόλοιπες μεταβλητές. Σημείωση: στην περίπτωση της απευθείας απάντησης (**direct callback**) τα αποτελέσματα μεταδίδονται στο DS1₁ χωρίς την μεσολάβηση του IS₁.

Περίπτωση 0: Κανένας από τους εξυπηρετητές δεικτοδότησης δεν απαντά, τότε οι μεταβλητές $T_{is_1}^{A \rightarrow B}$, $T_{is_2}^{C \rightarrow D}$, $T_{is_3}^{E \rightarrow F}$ γίνονται άπειρο. Συνεπώς,

$$T_{DS1_1}^{first} = T_{DS1_1}^{fork_{is}} + TO^{search}.$$

Περίπτωση 1: Εάν κάποιο από τα $T_{is_i}^{X \rightarrow Y} > TO^{search}$ για $i = 1, 2, 3$, τότε

χρησιμοποιείται ο εξυπηρετητής υποστήριξης. Η περίπτωση αυτή αναλύεται σε επόμενη παράγραφο.

Περίπτωση 2: Εάν όλοι οι κόμβοι απαντήσουν εντός του χρονικού περιθωρίου, τότε $T_{DS1_1}^{first} = T_{DS1_1}^{fork_{is}} + \max \left\{ T_{is_1}^{A \rightarrow B}, T_{is_2}^{C \rightarrow D}, T_{is_3}^{E \rightarrow F} \right\}$.

6.1.3. Επεξεργασία αίτησης στον τοπικό κόμβο.

$$\bullet T_{is_1}^{A \rightarrow B} = \min \left\{ TO^{search}, \left(T_{is_1} + T_{is_1 \rightarrow DS1_1}^{socket} \right) \right\}$$

$\circ T_{is_1 \rightarrow DS1_1}^{socket}$: Ο χρόνος που χρειάζεται για την μεταφορά των αποτελεσμάτων από το IS₁ στο DS1₀ μέσω του socket.

$$\bullet T_{is_1} = T_{Http} + T_{nds}^{local}$$

$\circ T_{nds}^{local}$: Ο χρόνος που χρειάζεται το NDS (τοπικά) για να στείλει την ερώτηση στο DS1₀ και να πάρει την απάντηση σε μορφή υπερκειμένου.

$\circ T_{Http} = T_{Http}^{req} + T_{Http}^{rsp}$: Ο χρόνος της HTTP αίτησης και απάντησης.

- $T_{nds}^{local} = T_{nds \rightarrow DS1_0}^{socket} + T_{DS1_0}^{fork2} + T_{DS1_2} + T_{DS1_2 \rightarrow nds}^{socket}$

- $T_{DS1_2} = \min \{ T_{DS1_2}, TO^{remote} \}$

ο T_{DS1_2} : Ο χρόνος που χρειάζεται ο δεύτερος forked εξυπηρετητής Dienst για την επεξεργασία μίας τοπικής ερώτησης.

ο TO^{remote} : Το τοπικό ή μακρινό μέγιστο χρονικό περιθώριο για το ψάξιμο του καταλόγου δεικτοδότησης (**database_search_timeout**).

Η ανάλυση των $T_{is2}^{C \rightarrow D}$ και $T_{is3}^{E \rightarrow F}$, που ακολουθεί είναι παρόμοια με την

ανάλυση του $T_{is1}^{A \rightarrow B}$, παραπάνω.

6.1.4. Επεξεργασία αίτησης στον μακρινό κόμβο 1.

- $T_{is2}^{C \rightarrow D} = \min \left\{ TO^{search}, \left(T_{is2} + T_{is2 \rightarrow DS1_1}^{socket} \right) \right\}$

- $T_{is2} = T_{Http} + T_{nds}^{remote1}$

$$T_{Http} = T_{Http}^{req} + T_{Http}^{rsp}$$

- $T_{nds}^{remote1} = T_{nds \rightarrow DS2_0}^{socket} + T_{DS2_0}^{fork} + T_{DS2_1} + T_{DS2_1 \rightarrow nds}^{socket}$

- $T_{DS2_1} = \min \{ T_{DS2_1}, TO^{remote} \}$

6.1.5. Επεξεργασία αίτησης στον μακρινό κόμβο 2.

- $T_{is3}^{E \rightarrow F} = \min \left\{ TO^{search}, \left(T_{is3} + T_{is3 \rightarrow DS1_1}^{socket} \right) \right\}$

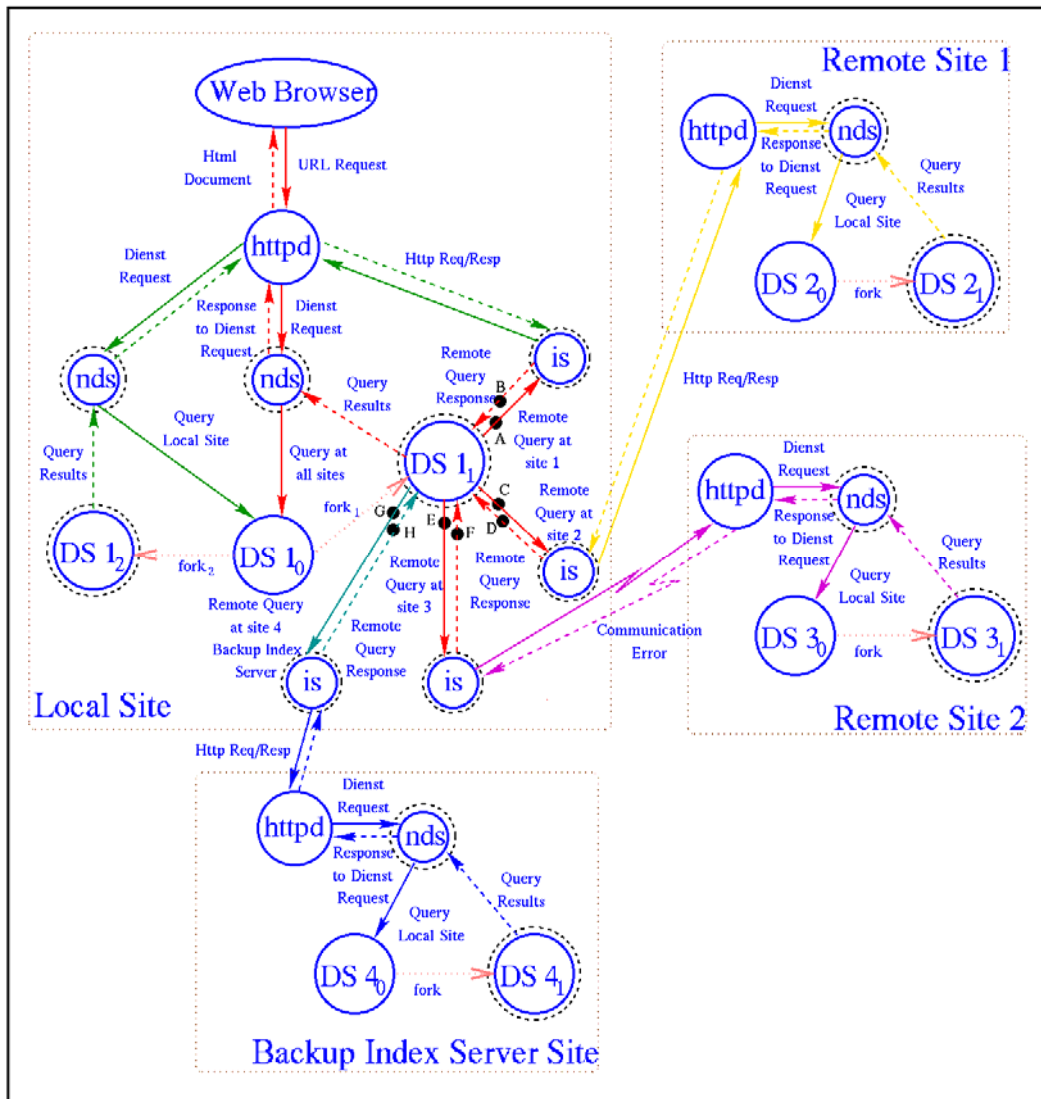
- $T_{is3} = T_{Http} + T_{nds}^{remote2}$

$$T_{Http} = T_{Http}^{req} + T_{Http}^{rsp}$$

- $T_{nds}^{remote2} = T_{nds \rightarrow DS3_0}^{socket} + T_{DS3_0}^{fork} + T_{DS3_1} + T_{DS3_1 \rightarrow nds}^{socket}$

- $T_{DS3_1} = \min \{ T_{DS3_1}, TO^{remote} \}$

6.2. Μοντέλο συστήματος με εξυπηρετητή υποστήριξης.



Σχήμα 12: Ροή Εξυπηρετήσης Αίτησης με Χρήση Εξυπηρετητή Υποστήριξης.

Στην περίπτωση που κάποιοι ή όλοι οι εξυπηρετητές Dienst αποτύχουν να εξυπηρετήσουν μια αίτηση, συνήθως λόγω κάποιου προβλήματος στην μεταξύ τους σύνδεση, τότε χρησιμοποιεί

ται ο εξυπηρετητής υποστήριξης (βλέπε στο Σχήμα 12). Η ροή εκτέλεσης στην περίπτωση αυτή μέχρι το σημείο που διαπιστώνεται το πρόβλημα είναι η ίδια με το Σχήμα 11. Στην συνέχεια, γίνεται η προώθηση της ερώτησης στον εξυπηρετητή υποστήριξης, που είναι ουσιαστικά ένας εξυπηρετητής Dienst, και η μεταφορά της απάντησής του στον χρήστη με τον ίδιο τρόπο που γίνεται και στο Σχήμα 11.

$$1. T_{DS1_1}^{first} = T_{DS1_1}^{fork_{is}} + TO^{search} + \min\left\{TO^{backup\ search}, T_{is4}^{G \rightarrow H}\right\}$$

$TO^{\text{backup search}}$: Το μέγιστο χρονικό περιθώριο για την αναζήτηση στον εξυπηρετητή υποστήριξης (**backup_search_timeout**).

$$2. T_{is3}^E \rightarrow F = TO^{\text{search}}$$

$$3. T_{is4}^G \rightarrow H = \min \left\{ TO^{\text{backup search}}, \left(T_{is4} + T_{is4 \rightarrow DS1_1}^{\text{socket}} \right) \right\}$$

$$4. T_{is4} = T_{\text{Http}} + T_{\text{nds}}^{\text{remote3}}$$

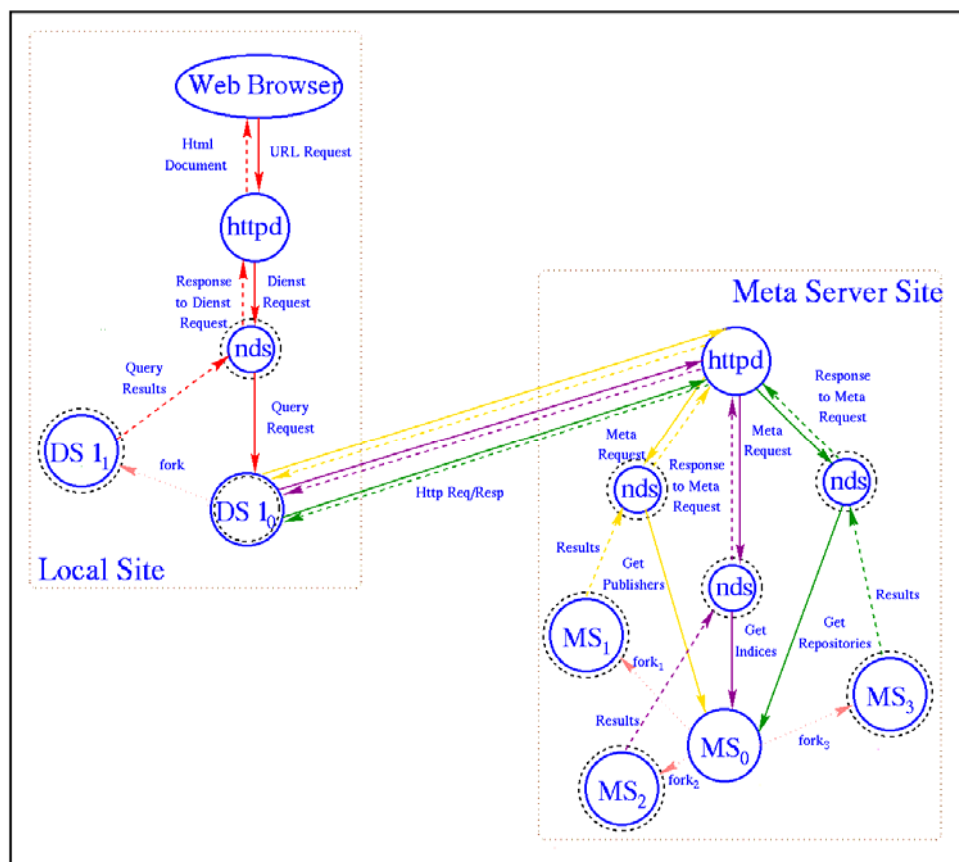
$$T_{\text{Http}} = T_{\text{Http}}^{\text{req}} + T_{\text{Http}}^{\text{rsp}}$$

$$5. T_{\text{nds}}^{\text{remote3}} = T_{\text{nds} \rightarrow DS4_0}^{\text{socket}} + T_{DS4_0}^{\text{fork}} + T_{DS4_1} + T_{DS4_1 \rightarrow \text{nds}}^{\text{socket}}$$

$$6. T_{DS4_1} = \min \left\{ T_{DS4_1}, TO^{\text{remote}} \right\}$$

Όλες οι παράμετροι επίδοσης στην περίπτωση της αναζήτησης στον εξυπηρετητή υποστήριξης, ορίζονται παρόμοια με τις αντίστοιχες μεταβλητές της προηγούμενης ανάλυσης.

6.3. Αίτηση με ενημέρωση των καταλόγων.



Σχήμα 13: Ροή Εξυπηρέτησης Αίτησης με Ενημέρωση των Καταλόγων.

Κάθε εξυπηρετητής Dienst για να αποφασίσει ποιους εξυπηρετητές αφορά κάθε ερώτηση πρέπει να ανατρέξει στους καταλόγους που διαθέτει. Πριν, όμως, χρησιμοποιήσει αυτούς τους καταλόγους πρέπει να εξετάσει αν είναι έγκυροι. Σε περίπτωση που διαπιστώσει ότι οι κατάλογοι του είναι άκυροι, δηλαδή παλιοί, τότε πρέπει να επικοινωνήσει με τον meta εξυπηρετητή για να πάρει ένα καινούργιο αντίτυπο των καταλόγων. Μετά την ενημέρωση των καταλόγων η εξυπηρέτηση της αίτησης γίνεται με τον τρόπο που φαίνεται στο Σχήμα 11.

$$1. T_{nds}^{total} = T_{nds \rightarrow DSI_0}^{socket} + T_{DSI_0}^{meta} + T_{DSI_0}^{fork} + T_{DSI_1}^{first} \left(\text{or } T_{DSI_1}^{last} \right)$$

ο $T_{DSI_0}^{meta}$: Ο συνολικός χρόνος που χρειάζεται για την ενημέρωση ή το φόρτωμα του καταλόγου δεικτοδότησης (ο χρόνος ρουτινών **update_meta_data** ή **load_meta_data**).

$$2. T_{DSI_0}^{meta} = \min \left\{ T_{MR}^{Publishers}, T_{O}^{meta} \right\} + \min \left\{ T_{MR}^{Indices}, T_{O}^{meta} \right\} + \min \left\{ T_{MR}^{Repositories}, T_{O}^{meta} \right\}$$

ο $T_{MR}^{Publishers}$: Ο χρόνος που χρειάζεται η αίτηση για τον κατάλογο των εκδοτών (**publishers meta request**).

ο $T_{MR}^{Indices}$: Ο χρόνος που χρειάζεται η αίτηση για τον κατάλογο των καταλόγων δεικτοδότησης (**indices meta request**).

ο $T_{MR}^{Repositories}$: Ο χρόνος που χρειάζεται η αίτηση για τον κατάλογο των αποθηκών (**repositories meta request**).

ο T_{O}^{meta} : Το μέγιστο χρονικό περιθώριο των αιτήσεων στον meta εξυπηρετητή Dienst (**meta_request_timeout**).

$$3. T_{MR}^{Publishers} = T_{Http} + T_{ndsPublishers}^{MS}$$

ο $T_{ndsPublishers}^{MS}$: Ο χρόνος που χρειάζεται το nph-dienst-stub για την αίτηση του καταλόγου των εκδοτών.

$$4. T_{ndsPublishers}^{MS} = T_{nds \rightarrow MS_0}^{socket} + T_{MS_0}^{fork1} + T_{MS_1}^{first} \left(\text{or } T_{MS_1}^{last} \right)$$

$$5. T_{MR}^{Indices} = T_{Http} + T_{ndsIndices}^{MS}$$

$\circ T_{ndsIndices}^{MS}$: Ο χρόνος που χρειάζεται το nph-dienst-stub για την αίτηση του καταλόγου των καταλόγων δεικτοδότησης.

$$6. T_{ndsIndices}^{MS} = T_{nds \rightarrow MS_0}^{socket} + T_{MS_0}^{fork2} + T_{MS_2}^{first} \left(\text{or } T_{MS_2}^{last} \right)$$

$$7. T_{MR}^{Repositories} = T_{Http} + T_{ndsRepositories}^{MS}$$

$\circ T_{ndsRepositories}^{MS}$: Ο χρόνος που χρειάζεται το nph-dienst-stub για την αίτηση του καταλόγου των αποθηκών.

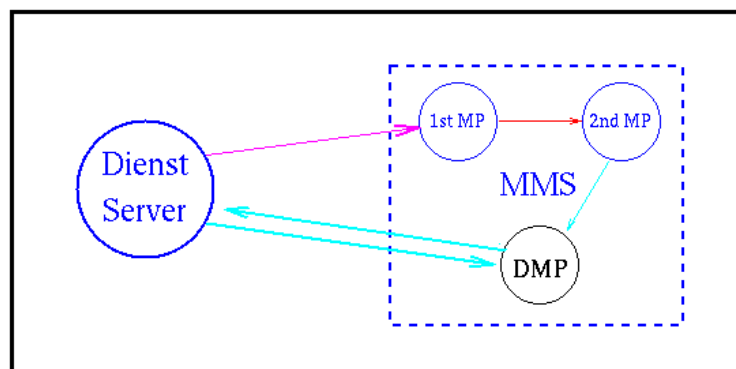
$$8. T_{ndsRepositories}^{MS} = T_{nds \rightarrow MS_0}^{socket} + T_{MS_0}^{fork3} + T_{MS_3}^{first} \left(\text{or } T_{MS_3}^{last} \right)$$

Όσες παράμετροι δεν υπάρχει ορισμός τους παραπάνω, ορίζονται κατά παρόμοιο τρόπο με τις αντίστοιχες μεταβλητές της αρχικής ανάλυσης.

Κεφάλαιο 7. Εφαρμογές της Εποπτείας Επίδοσης στο Σύστημα *NCSTRL-Dienst*.

Στο κεφάλαιο αυτό περιγράφουμε μερικά παραδείγματα της χρήσης του επόπτη επίδοσης στο σύστημα *NCSTRL-Dienst*. Με βάση την ανάλυση επίδοσης του συστήματος *NCSTRL-Dienst*, που παρουσιάστηκε στο Κεφάλαιο 6 επιλέχθηκαν τα ενδιαφέροντα τμήματα του συστήματος. Για τα τμήματα αυτά ορίστηκαν μια σειρά από παραμέτρους επίδοσης (που αναφέρονται στο Παράρτημα 5) και ο επόπτης επίδοσης συνδέθηκε με το σύστημα για την μέτρηση τους (βλέπε το Σχήμα 14).

Η σύνδεση του εξυπηρετητή *Dienst* με τον επόπτη επίδοσης γίνεται με την χρήση *sockets*. Συγκεκριμένα, ο εξυπηρετητής, κατά την εκκίνηση του, ξεκινά τον επόπτη επίδοσης και συνδέεται μαζί του. Επίσης, του στέλνει το μήνυμα για την εκκίνηση των μετρήσεων. Κατά την διάρκεια της λειτουργίας του ο εξυπηρετητής στέλνει στον επόπτη μηνύματα για την μέτρηση των παραμέτρων επίδοσης, καθώς και μηνύματα για την αναπαραγωγή του εαυτού του (**fork**, βλέπε το Παράρτημα 1, για τα μηνύματα που βελτιώνουν την μορφή και την πληρότητα των αρχείων καταγραφής των μετρήσεων και εξασφαλίζουν την ορθή λειτουργία του). Όταν ο εξυπηρετητής *Dienst* δεκτεί σήμα για τον ομαλό τερματισμό του (**signal hungup**), ειδοποιεί τον επόπτη για τον επερχόμενο τερματισμό. Ο επόπτης, τότε, φροντίζει για το κλείσιμο των αρχείων καταγραφής των μετρήσεων, για την τοποθέτηση των τρεχόντων τιμών των παραμέτρων επίδοσης σε ένα αρχείο και για το κλείσιμο των *sockets*. Στην συνέχεια, ο επόπτης επίδοσης τερματίζει και τότε μπορεί ο εξυπηρετητής *Dienst* να τερματίσει κι αυτός την λειτουργία του. Για τις προσθήκες που ήταν απαραίτητες στο σύστημα *NCSTRL-Dienst* για τα παραπάνω λεπτομέρειες υπάρχουν στο Παράρτημα 6.



Σχήμα 14: Αρχιτεκτονική Επικοινωνίας *Dienst* – Επόπτη Επίδοσης.

Ακόμα, το πρωτόκολλο Dienst επεκτάθηκε με την εισαγωγή νέων αιτήσεων για την ανάκληση των τιμών των παραμέτρων επίδοσης. Οι παράμετροι επίδοσης παρουσιάζονται σε έναν πίνακα με την μορφή γλώσσας υπερκειμένου (**html**). Επίσης, ο εξυπηρετητής μπορεί να στέλνει ερωτήσεις στον επόπτη, σύμφωνα με το πρωτόκολλο που περιγράφεται στο Παράρτημα 1, για την ενημέρωσή του για τις τιμές των παραμέτρων επίδοσης. Η παραπάνω επικοινωνία είναι αναγκαία για τις περιπτώσεις της δυναμικής αναπροσαρμογής των χρονικών περιθωρίων και της εξισορρόπησης φόρτου, οι οποίες αναλύονται παρακάτω. Με την χρήση του επόπτη έγινε μια προκαταρκτική μελέτη της επίδοσης του συστήματος από την οποία προέκυψαν κάποιες προτάσεις για την βελτίωση της αρχιτεκτονικής και της υλοποίησης του. Τέλος, δοκιμάστηκαν ενδεικτικές στρατηγικές για την Εξισορρόπηση φόρτου και προτείνονται στρατηγικές για την δυναμική αναπροσαρμογή των χρονικών περιθωρίων.

7.1. Αποτελέσματα προκαταρκτικής μελέτης επίδοσης του συστήματος NCSTR-L-Dienst.

Ο επόπτης επίδοσης χρησιμοποιήθηκε για την μελέτη του προφίλ επίδοσης των αιτήσεων αναζήτησης (**queries**) και την μελέτη της επίδρασης μιας σειράς παραμέτρων στον χρόνο απόκρισης του συστήματος. Οι παράμετροι που μελετήθηκαν χωρίζονται σε δύο κατηγορίες: (α) αυτές που εξαρτώνται από την διαμόρφωση (**configuration**) του εξυπηρετητή Dienst (π.χ. το μέγεθος της βάσης δεικτοδότησης), (β) αυτές που εξαρτώνται από την μηχανή που φιλοξενεί των εξυπηρετητή (π.χ. το μέγεθος της μνήμης, η ταχύτητα του επεξεργαστή, κ.α.) και την καθυστέρηση δικτύου (**network latency**) στην επικοινωνία των εξυπηρετητών.

7.1.1. Πλατφόρμα πειραμάτων.

Η πειραματική πλατφόρμα αποτελείται από τρεις τοπικούς εξυπηρετητές τοποθετημένους στο ίδιο τοπικό δίκτυο και έναν απομακρυσμένο εξυπηρετητή σε ένα αρκετά γεωγραφικά μακρινό δίκτυο. Οι τοπικοί εξυπηρετητές βρίσκονται στα εξής μηχανήματα:

- 1) Έναν σταθμό εργασίας Sun4c με 48M μνήμη και SunOS 4.1.2.
- 2) Έναν σταθμό εργασίας Alpha με 32M μνήμη και Digital Unix V 3.2.
- 3) Έναν σταθμό εργασίας Alpha με 64M μνήμη και Digital Unix V 3.2.

Ο μακρινός εξυπηρετητής ήταν:

- 4) Έναν σταθμό εργασίας Sun4c με 16M μνήμη και SunOS 4.1.3.

Τα δύο Sun έχουν την ίδια βάση δεικτοδότησης (4644 λέξεις). Ο πρώτος Alpha σταθμός έχει βάση με 10979 λέξεις και ο δεύτερος με 20222 λέξεις. Ο φόρτος στο σύστημα ήταν τεχνητός και παραγόταν από ένα *script*, που υποβάλλει αναζητήσεις με βάση λέξεις κλειδιά στον έναν από τους εξυπηρετητές Dienst.

7.1.2. Πειραματικά αποτελέσματα.

Η πρώτη παρατήρηση στο σύστημα ήταν ότι καταναλώνει σημαντικό αριθμό πόρων και συνεπώς δεν μπορεί να εξυπηρετήσει υψηλούς ρυθμούς άφιξης αιτήσεων. Βασικές αιτίες τις υψηλής κατανάλωσης πόρων είναι (α) οι συνεχείς αναπαραγωγές (**fork**) του εξυπηρετητή για την εξυπηρέτηση των αιτήσεων, (β) το γεγονός ότι όλη η βάση δεικτοδότησης είναι στην κύρια μνήμη και (γ) το ότι το σύστημα είναι γραμμένο σε *Perl*.

Σημείωση: Όλα τα νούμερα που δίνονται παρακάτω έχουν προέλθει με σειρά πειραμάτων στην πλατφόρμα που περιγράφεται στην παραπάνω παράγραφο.

Συγκεκριμένα, η διαδικασία των πειραμάτων ήταν η εξής: η ίδια αίτηση αναζήτησης (*/Dienst/UI/2.0/Query?boolean=and&author=&title=&abstract=dat&authoriy=fort h.stat.panoramix*), δηλαδή δώσε όλα τα έγγραφα που έχουν στην περίληψη λέξεις που αρχίζουν με *dat*, π.χ data, database κ.α., υποβάλλονταν στο σύστημα πολλές φορές (150) από ένα script και συλλέγονταν τα αποτελέσματα με την χρήση των νέων ερωτήσεων που προστέθηκαν στο πρωτόκολλο Dienst (βλέπε στην επόμενη παράγραφο). Οι χρόνοι που δίνονται είναι η μέση τιμή των χρόνων που μετρήθηκαν στο σύστημα. Γενική παρατήρηση είναι ότι οι τιμές των παραμέτρων επίδοσης που περιλαμβάνουν και χρόνο δικτύου παρουσιάζουν σημαντική τυπική απόκλιση, γεγονός που αντικατοπτρίζει την αστάθεια και την ευμεταβλητότητα του δικτύου.

Έτσι αναλυτικότερα, είναι χαρακτηριστικός ο χρόνος που χρειάζεται μία αναζήτηση στην ίδια βάση δεικτοδότησης όταν η αίτηση υποβλήθηκε στον ίδιο ή σε άλλο εξυπηρετητή (όπως είδαμε στην ανάλυση επίδοσης του συστήματος για την εξυπηρέτηση μιας ερώτησης στον τοπικό κόμβο χρειάζεται ο εξυπηρετητής να αναπαράγει τον εαυτό δύο φορές ενώ διαφορετικά μόνο μία, βλέπε Σχήμα 11).

Μακρινή Αναζήτηση	1.16s
Τοπική Αναζήτηση	2.95s

(3) Χρόνος Τοπικής Έναντι Μακρινής Αναζήτησης.

Επομένως, ο χρόνος επεξεργασίας της ερώτησης στην τοπική βάση δεικτοδότησης, η παράμετρος επίδοσης $TDSI_2$ σχεδόν διπλασιάζεται.

Ο μέγιστος ρυθμός αιτήσεων που μπορεί να εξυπηρετήσει το σύστημα ανά μονάδα χρόνου εξαρτάται:

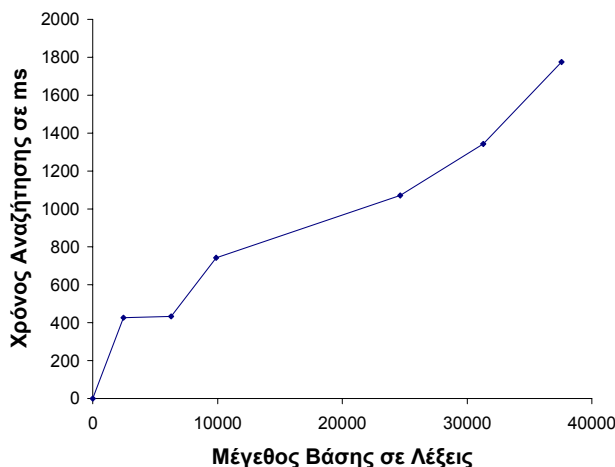
- Από το μέγεθος της μνήμης.

Ίδια μηχανή, μέγεθος μνήμης σε MB	16	48
Χρόνος Αναζήτησης	4.657s	0.638s

(4) Μέγεθος Μνήμης και Χρόνος Αναζήτησης.

Στο πείραμα αυτό, είχαμε έναν εξυπηρετητή να στέλνει τις ερωτήσεις σε δύο άλλους εξυπηρετητές που βρίσκονταν σε ίδια μηχανήματα (Sun4c) με 16 και 48 MB μνήμης και είχαν την ίδια βάση δεικτοδότησης. Έτσι, ο εξυπηρετητής με μέγεθος μνήμης 48M παρουσιάζει πάνω από **επτά** φορές μικρότερο χρόνο αναζήτησης στην τοπική του βάση (η παράμετρος επίδοσης $TDSI_2$) από τον εξυπηρετητή με τα 16M.

- Από το μέγεθος της βάσης δεικτοδότησης.



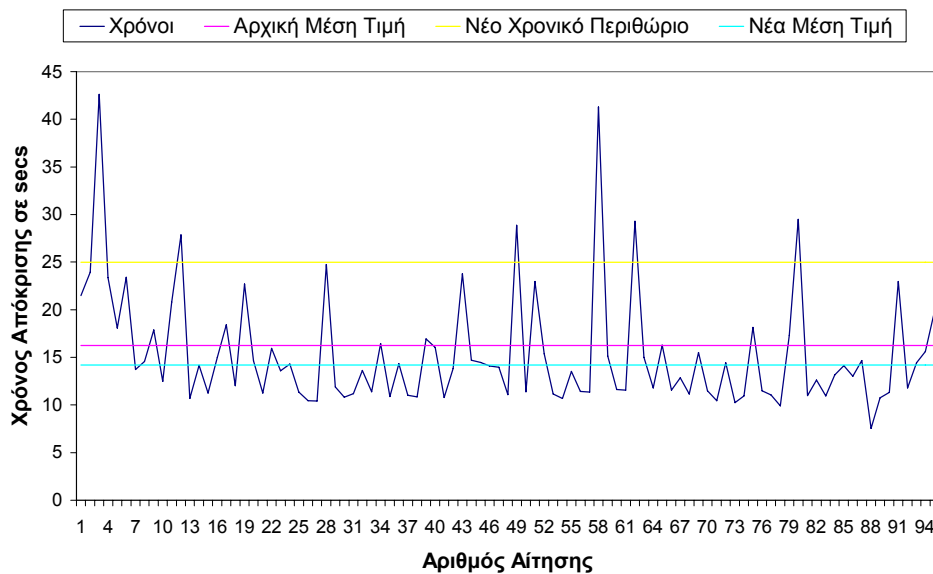
(5) Μέγεθος Βάσης και Χρόνος Αναζήτησης.

Επειδή όλη η βάση δεικτοδότησης φορτώνεται στην κύρια μνήμη, καθώς αυξάνει το μέγεθος της υπάρχει ένα μέγεθος πάνω από το οποίο ο χρόνος αναζήτησης αυξάνει εκθετικά. Αυτό οφείλεται στο ότι ο εξυπηρετητής δεν μπορεί να φέρει όλες τις σελίδες μνήμης που χρειάζεται ταυτόχρονα στην κύρια μνήμη (**swaping**).

□ Από την πολυπλοκότητα των ερωτήσεων.

Οι πολύπλοκες ερωτήσεις, δηλαδή οι ερωτήσεις που περιέχουν πολλές λέξεις κλειδιά και περίπλοκες λογικές σχέσεις, θέλουν πιο πολύ χρόνο για την εξυπηρέτησή τους. Αυτό συνεπάγεται ότι ο εξυπηρετητής παραμένει ενεργός για μεγαλύτερο χρονικό διάστημα αυξάνοντας τον φόρτο του μηχανήματος.

□ Από τις τιμές των μέγιστων χρονικών περιθωρίων (μεγάλες τιμές χρονικών περιθωρίων οδηγούν τον εξυπηρετητή στο να προσπαθεί να υπολογίσει απαντήσεις που δεν περιμένει κανένας).



(6) Χρονικά Περιθώρια και Χρόνος Εξυπηρέτησης Αίτησης.

Στο πείραμα αυτό μία αίτηση αναζήτησης στάλθηκε 100 φορές σε έναν εξυπηρετητή. Η αίτηση αφορούσε αρχές έκδοσης ενός άλλου μακρινού εξυπηρετητή. Το χρονικό περιθώριο TO^{search} στην περίπτωση αυτή ήταν 45s και 4 φορές η αίτηση το ξεπέρασε (στο γράφημα έχουμε 96 τιμές). Η μέση τιμή του χρόνου απόκρισης $T_X^{X \rightarrow Y}$ ήταν 16,223s και η τυπική απόκλιση ήταν 8,752. Ο συνολικός χρόνος απόκρισης είναι 1737,455s (το άθροισμα όλων των χρόνων που εμφανίζονται στο γράφημα συν $4 \cdot 45$, οι τέσσερις υπερβάσεις του χρονικού περιθωρίου). Θέτοντας, όμως, την τιμή του χρονικού περιθωρίου στα 24,975s (μέση τιμή συν τυπική απόκλιση), βλέπουμε ότι συνολικά 10 φορές η αίτηση θα ξεπερνούσε το νέο χρονικό περιθώριο. Ο συνολικός χρόνος απόκρισης στην περίπτωση αυτή θα ήταν 1607,786s. Επομένως, με μόνο 6% παραπάνω υπερβάσεις του χρονικού περιθωρίου θα είχαμε μειώσει τον συνολικό χρόνο απόκρισης κατά 129,669s, δηλαδή 7,5%. Το κέρδος γίνεται ακόμα σημαντικότερο στην περίπτωση μεγάλου αριθμού εξυπηρετητών. Τότε κατά την εξυπηρέτηση κάθε αίτησης η πιθανότητα κάποιος εξυπηρετητής να είναι αρκετά αργός, δηλαδή να βρεθεί μία ψηλή κορυφή που προσεγγίζει το χρονικό περιθώριο, όπως συνέβει στο παραπάνω γράφημα, είναι πολύ μεγάλη. Σε αυτές τις περιπτώσεις, όμως,

έχουμε και το μεγαλύτερο κέρδος με το νέο χρονικό περιθώριο. Συνεπώς, αφού αυτές οι περιπτώσεις είναι πιο συχνές είναι μεγαλύτερο και το συνολικό κέρδος χρόνου.

Το σημαντικότερο, όμως, ποσοστό του χρόνου απόκρισης είναι η καθυστέρηση στο δίκτυο. Η καθυστέρηση στο δίκτυο προκύπτει εάν από τον χρόνο απόκρισης ενός μακρινού εξυπηρετητή αφαιρέσουμε τον χρόνο αναζήτησης στην τοπική του βάση δεικτοδότησης, δηλαδή $T_{Net} = T_{X \rightarrow Y} - T_{DSn2}$. Όπως φαίνεται και στον πίνακα παρακάτω χρειάζεται ο επταπλασιασμός του μεγέθους της βάσης δεικτοδότησης για να μειώσουμε το ποσοστό της καθυστέρησης στο δίκτυο προς τον συνολικό χρόνο απόκρισης, δηλαδή $T_{Net} \div T_{DS1}$, στο μισό.

Αριθμός Λέξεων	10,979	77,105
καθυστέρηση Δικτύου ÷ Συνολικό Χρόνο Εκτέλεσης	70%	35%

(7) Μέγεθος Βάσης και καθυστέρηση στο Δίκτυο.

Το ποσοστό που αποτελεί η καθυστέρηση του δικτύου εξαρτάται φυσικά και από την απόσταση μεταξύ του κόμβου στον οποίο υποβλήθηκε η αίτηση και τον κόμβο που θα την εξυπηρετήσει. Στην περίπτωση της πειραματικής πλατφόρμας που περιγράφηκε παραπάνω η εξυπηρετήση μιας αίτησης στον μακρινό κόμβο αντί του τοπικού (οι τοπικοί εξυπηρετητές είναι πάνω στο ίδιο ethernet) αύξανε την καθυστέρηση του δικτύου κατά 12s περίπου.

7.2. Επικοινωνία μεταξύ των εξυπηρετητών και χρονικά περιθώρια.

Το σύστημα NCSTRL-Dienst αποτελείται από εξυπηρετητές δεικτοδότησης, εξυπηρετητές συγχωνευμένου καταλόγου, εξυπηρετητές υποστήριξης και meta εξυπηρετητές. Για την λειτουργία του συστήματος, όπως αναφέρθηκε και στο Κεφάλαιο 5, οι εξυπηρετητές αυτοί χρειάζεται να επικοινωνούν μεταξύ τους. Η επικοινωνία αυτή, όπως και κάθε μορφή επικοινωνίας μέσω δικτύου, μπορεί να αποτύχει για διάφορους λόγους. Μία αποτυχία στην επικοινωνία μπορεί να οδηγήσει σε μπλοκάρισμα της λειτουργίας του συστήματος. Για αυτό το λόγο το σύστημα χρησιμοποιεί κάποια χρονικά περιθώρια, τα οποία εγγυώνται την συνέχιση της λειτουργίας του μετά από κάποια αποτυχημένη επικοινωνία.

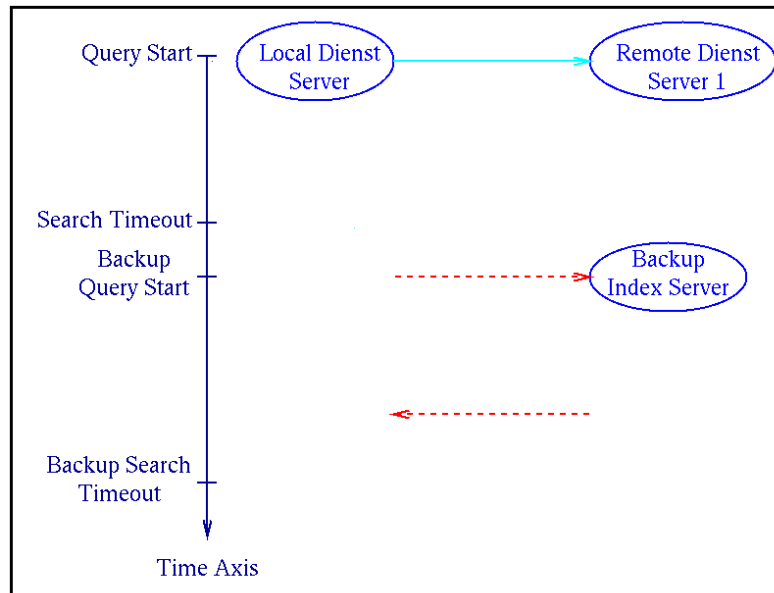
Η τρέχουσα υλοποίηση του συστήματος NCSTRL-Dienst χρησιμοποιεί τρία χρονικά περιθώρια: TO^{remote} (το χρονικό περιθώριο για μακρινούς ή τον τοπικό εξυπηρετητή για αναζήτηση στην βάση δεικτοδότησης), TO^{search} (το χρονικό περιθώριο για το σύνολο μιας αναζήτησης) και $TO^{backupsearch}$ (το χρονικό περιθώριο για την αναζήτηση στον εξυπηρετητή υποστήριξης), για τους τυπικούς ορισμούς των παραμέτρων βλέπε στο Κεφάλαιο 6. Για τους λόγους που αναφέρθηκαν στην παράγραφο 4.4 και τονίζονται από τα πειραματικά αποτελέσματα της παραπάνω παραγράφου, η εφαρμογή τεχνικών δυναμικής αναπροσαρμογής των χρονικών περιθωρίων μπορεί να εφαρμοστεί και να οδηγήσει σε βελτίωση της απόδοσης του συστήματος NCSTRL-Dienst. Από την άλλη, με μερικές τροποποιήσεις στον τρόπο επικοινωνίας μεταξύ των εξυπηρετητών του συστήματος μπορούμε να πετύχουμε ακόμα μεγαλύτερες βελτιώσεις.

7.2.1. Τροποποιημένο πρωτόκολλο επικοινωνίας του συστήματος NCSTRL-Dienst.

Το σύστημα NCSTRL-Dienst, όπως και κάθε σύστημα ψηφιακής βιβλιοθήκης, είναι φύσει δυναμικό. Οι προτεινόμενες αλλαγές στον τρόπο επικοινωνίας μεταξύ των εξυπηρετητών έχουν ως στόχο με την χρήση των παραμέτρων, που μετρά ο επόπτης επίδοσης, ώστε να μπορεί το σύστημα να προσαρμόζεται δυναμικά στο μεταβαλλόμενο περιβάλλον του.

- Προτείνεται η προσθήκη ενός νέου μηνύματος στην διαδικασία επικοινωνίας των εξυπηρετητών. Το μήνυμα αυτό θα στέλνεται από καθέναν από τους μακρινούς εξυπηρετητές, στους οποίους προώθησε ο τοπικός εξυπηρετητής την αίτηση του χρήστη, προς τον τοπικό εξυπηρετητή. Το περιεχόμενο του μηνύματος θα είναι «είμαι ζωντανός και μπορώ/δεν μπορώ να εξυπηρετήσω την ερώτηση σου». Με αυτό τον τρόπο κάθε εξυπηρετητής Dienst μπορεί να αρνηθεί την εξυπηρέτηση μιας αίτησης όταν κρίνει ότι κάτι τέτοιο είναι ουσιαστικά αδύνατο. Ταυτόχρονα, το σύστημα αποκτά την δυνατότητα να εγγυηθεί την ποιότητα των υπηρεσιών που παρέχει στον χρήστη (βλέπε την παράγραφο 4.2). Ακόμα, το μήνυμα αυτό μπορεί να περιέχει και μία εκτίμηση του εξυπηρετητή για τον χρόνο που θα χρειαστεί για την εξυπηρέτηση της αίτησης. Στην περίπτωση αυτή ο τοπικός εξυπηρετητής έχει την δυνατότητα να διορθώσει τις προβλέψεις που του παρέχει ο επόπτης επίδοσης. Επίσης, γίνεται αναγκαίο ο τοπικός εξυπηρετητής να στείλει ένα μήνυμα πίσω στον μακρινό που θα του λέει είτε να εγκαταλείψει την προσπάθεια είτε να συνεχίσει γνωρίζοντας ότι έχει περισσότερο χρόνο, αφού έχει αυξηθεί το χρονικό περιθώριο.
- Μία άλλη τροποποίηση είναι η κατάργηση του χρονικού περιθωρίου για την αναζήτηση στην τοπική ή Μακρινή βάση δεικτοδότησης (TO^{remote}) και η αντικατάστασή του στο μήνυμα του τοπικού εξυπηρετητή προς τον μακρινό από το TO^{search} , το χρονικό περιθώριο ολοκλήρωσης της αναζήτησης. Από την στιγμή που κάθε εξυπηρετητής μπορεί να εκτιμήσει τον χρόνο που χρειάζεται για την εξυπηρέτηση μιας αίτησης, τότε μπορεί, έχοντας το TO^{search} , να υπολογίσει με την ανισότητα Tchebysev την πιθανότητα να εξυπηρετήσει την αίτηση πριν το πέρας του χρονικού περιθωρίου. Εάν η πιθανότητα προκύψει μικρή τότε μπορεί ο μακρινός εξυπηρετητής να αρνηθεί την εξυπηρέτηση της αίτησης με την χρήση του μηνύματος που ορίστηκε παραπάνω.

Οι παραπάνω τροποποιήσεις στον τρόπο επικοινωνίας μεταξύ των εξυπηρετητών, οδηγούν σε ορισμένες αλλαγές στον τρόπο χρήσης των χρονικών περιθωρίων. Έτσι στο Σχήμα 15, ο τοπικός εξυπηρετητής Dienst στέλνει μία ερώτηση σε έναν μακρινό εξυπηρετητή, (*κυανό βέλος*), ο μακρινός εξυπηρετητής **δεν** απαντά και η αίτηση στέλνεται στο εξυπηρετητή υποστήριξης και παίρνει την απάντηση, (*διακεκομμένα βέλη*). Έτσι, με την χρήση του νέου μηνύματος ο τοπικός εξυπηρετητής θα γνωρίζει σχετικά γρήγορα εάν ο μακρινός κόμβος είναι διαθέσιμος ή όχι. Εάν ο μακρινός κόμβος δεν στείλει το μήνυμα «είμαι ζωντανός» τότε μπορεί να θεωρηθεί απροσπέλαστος, διαφορετικά γνωρίζουμε ότι είναι προσπελάσιμος και άρα μπορούμε να του δώσουμε περισσότερο χρόνο για να απαντήσει. Συνεπώς, για τους κόμβους που χρησιμοποιούν τις τροποποιήσεις μπορούμε να έχουμε επεκτεταμένο διάστημα ασφαλείας TESI. Ένα νέο χρονικό περιθώριο για τα νέα μηνύματα είναι απαραίτητο.



Σχήμα 15: Τρέχουσα Χρήση των Χρονικών Περιθωρίων στο Σύστημα NCSTRL-Dienst.

Οι παραπάνω αλλαγές καταργούν ουσιαστικά την έννοια του εξυπηρετητή υποστήριξης. Η διαδικασία «χρονικό περιθώριο για το μήνυμα «είμαι ζωντανός» - χρονικό περιθώριο για την συνολική αναζήτηση», μπορεί να επεκταθεί σε όλους του κατάλληλους εξυπηρετητές μέχρι είτε να πάρουμε την απάντηση είτε να εξαντληθεί το TO_{user} . Έτσι, στο Σχήμα 16, όπου ο τοπικός εξυπηρετητής στέλνει μια ερώτηση στον μακρινό εξυπηρετητή 1 (κυανό βέλος), αυτός απαντά «είμαι ζωντανός και δεν μπορώ να εξυπηρετήσω την ερώτηση (χρυσό βέλος), στην συνέχεια η ερώτηση στέλνεται στον μακρινό εξυπηρετητή 2 (το κόκκινο διακεκομμένο βέλος), αυτός απαντά «είμαι ζωντανός και μπορώ να εξυπηρετήσω την ερώτηση» (το χρυσό διακεκομμένο βέλος), το χρονικό περιθώριο εξαντλείται οπότε η ερώτηση στέλνεται στον μακρινό εξυπηρετητή 3, ο οποίος και απαντά «είμαι ζωντανός και μπορώ να εξυπηρετήσω την ερώτηση» και μετά από λίγο στέλνει την απάντηση (κόκκινα και χρυσά διακεκομμένα βέλη). Προσοχή: η ερώτηση στέλνεται στον μακρινό εξυπηρετητή 3 πριν εξαντληθεί το χρονικό περιθώριο για το σύνολο της αναζήτησης και δεν υπάρχει ειδικευμένος εξυπηρετητής υποστήριξης.

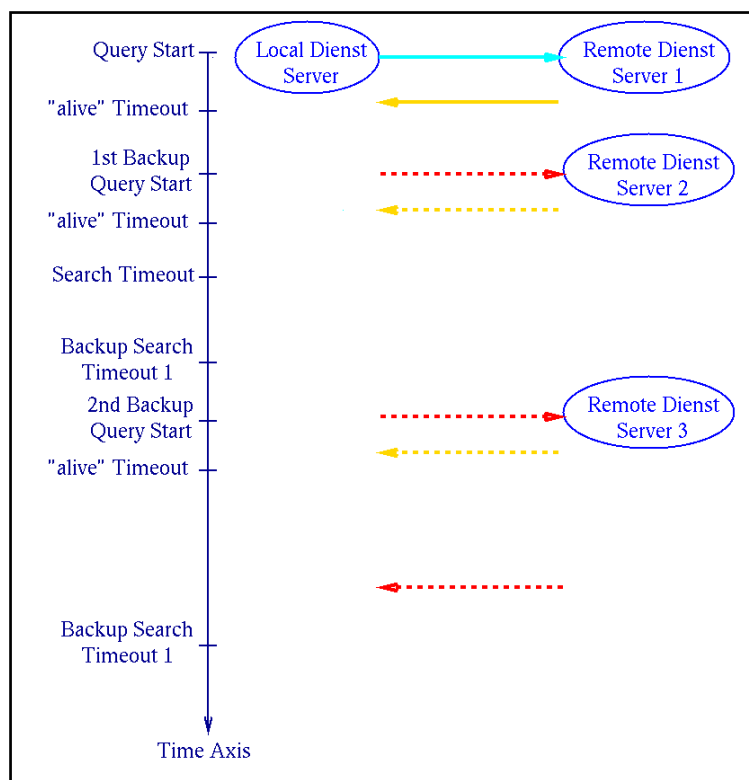
Η έννοια του χρονικού περιθωρίου για αναζήτηση στον εξυπηρετητή υποστήριξης ($TO^{backupsearch}$) μπορεί να διατηρηθεί και να οριστεί ως:

$$\min\{TO_{user}, T_{1st_est} + T_{BSI}\}$$

(8) Χρονικό Περιθώριο Αναζήτησης στον Εξυπηρετητή Υποστήριξης,

όπου $T_{1st_est} = \max\{T_{est_i}\}$ της σχέσης $\min\{TO_{user}, \max\{T_{est_i}\} + T_{SI}\}$

(1) και T_{BSI} είναι το διάστημα ασφαλείας υποστήριξης.



Σχήμα 16: Προτεινόμενη Χρήση των Χρονικών περιθωρίων στο Σύστημα NCSTRL-Dienst.

7.3. Επέκταση του πρωτοκόλλου Dienst.

Print Parameters

Performance Parameters

Parameter	Authorities	Calls	RSP	Min	Max	Total	Average	Std Dev	Weighted Average
$T_{is_B}^{A \rightarrow B}$									
	denver	166	18.136	17.514	146.408	4095.067	24.669	13.829	39.687
T_{DS1}^{req}									
	SimpleVar	4	0.13	0.13	0.329	0.898	0.224	0.081	0.202
T_{nds}^{total}									
	SimpleVar	171	52.026	0.628	195.275	7159.163	41.866	27.858	62.219
T_{DS1_1}									
	SimpleVar	166	90.317	20.11	193.721	6872.104	41.398	27.478	70.926
T_{DS1_2}									
	cornell.tc	163	1.158	0.911	1.488	194.351	1.192	0.124	1.222
$T_{is_A}^{A \rightarrow A}$									
	cornell.tc	163	7.143	5.325	17.492	1590.463	9.757	1.359	7.038
$T_{is_C}^{A \rightarrow C}$									
	icase	163	65.972	13.616	92.173	3961.152	24.301	12.212	44.869

Επιχειρησιακό Κέντρο: NCSTRL-ERCIM
 This server operates at Institute of Computer Science Technical Report Library at A.
 Send email to terzis@csi.forth.gr

Σχήμα 17: Σελίδα με τις Παραμέτρους Επίδοσης ενός Εξυπηρετητή.

Για την ανάκληση πληροφοριών που έχουν σχέση με την επίδοση του συστήματος, μέσα από το ίδιο το σύστημα, μία επέκταση του πρωτοκόλλου Dienst v.2.0 ήταν αναγκαία. Η νέα έκδοση ονομάστηκε 2.1. Οι νέες αιτήσεις περιλαμβάνονται σε μία νέα κλάση υπηρεσιών για το πρωτόκολλο, την κλάση *Stat* (στατιστικά). Σε πρώτη φάση δύο νέες αιτήσεις ορίστηκαν η *Print-Local-Parameter* και η *Print-Parameters*. Η πρώτη φέρνει όλες τις τιμές των παραμέτρων επίδοσης του συστήματος που διαθέτει ο τοπικός επόπτης επίδοσης (βλέπε στο Σχήμα 17), ενώ η δεύτερη προωθεί την αίτηση και στους υπόλοιπους εξυπηρετητές, δηλαδή στέλνει σε κάθε εξυπηρετητή που γνωρίζει μία αίτηση *Print-Local-Parameters*, και συγκεντρώνει όλες τις τιμές των παραμέτρων για όλους τους εξυπηρετητές Dienst του συστήματος.

Η σελίδα στο Σχήμα 17 προήλθε από ένα σύστημα με τρεις εξυπηρετητές (*A*, *B*, *C*), οι οποίοι είναι διασκορπισμένοι στο παγκόσμιο δίκτυο και δεικτοδοτούν διαφορετικές αρχές έκδοσης (*icase*, *cornell.tc*, *denver*). Στο σύστημα στάλθηκαν ερωτήσεις στον εξυπηρετητή *A* που αφορούσαν και τις τρεις αρχές έκδοσης. Χαρακτηριστικό στον πίνακα είναι ότι οι χρόνοι απόκρισης των εξυπηρετητών *B* και *C* παρουσιάζουν σημαντική τυπική απόκλιση και επίσης είναι σημαντικά μεγαλύτεροι του χρόνου απόκρισης του εξυπηρετητή *A*, αυτό οφείλεται στο δίκτυο το οποίο χρησιμοποιείται και δείχνει τη ευμεταβλητότητα του.

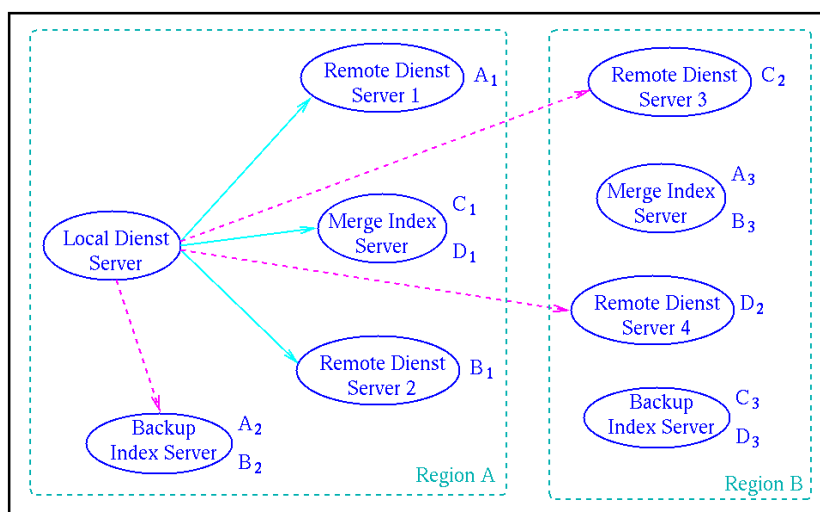
7.4. Εξισορρόπηση φόρτου.

Η αρχιτεκτονική των περιοχών (**regional architecture**, βλέπε την παράγραφο 5.2.1) με την εισαγωγή των εξυπηρετητών συγχωνευμένου καταλόγου έχει δώσει την δυνατότητα εναλλακτικών επιλογών δρομολόγησης των υποερωτήσεων στο σύστημα NCSTR-L-Dienst. Έτσι, με βάση την αρχιτεκτονική περιοχών κατά την εξυπηρέτηση μίας αίτησης, το σύστημα κατασκευάζει μία λίστα από εξυπηρετητές δεικτοδότησης στους οποίους πρέπει να απευθυνθεί για την κάλυψη όλων των αρχών έκδοσης που προσδιόρισε ο χρήστης στην αίτηση. Για κάθε μία από τις αρχές έκδοσης ο εξυπηρετητής μπορεί να απευθυνθεί είτε στον εξυπηρετητή δεικτοδότησης της αρχής έκδοσης, είτε σε κάποιο από τους εξυπηρετητές συγχωνευμένου καταλόγου κάθε περιοχής εκτός αυτής που βρίσκεται ο εξυπηρετητής της αρχής έκδοσης. Είναι επομένως, φανερό ότι όσο αυξάνουν οι περιοχές στο σύστημα τόσο αυξάνουν και οι εναλλακτικές επιλογές του εξυπηρετητή. Εάν μάλιστα εξεταστούν οι διαφορετικές επιλογές για το σύνολο των αρχών έκδοσης που περιλαμβάνει μία ερώτηση (συνήθως όλες οι διαθέσιμες), τότε το πλήθος των εναλλακτικών επιλογών αυξάνει εκθετικά. Από την άλλη, θα μπορούσε το σύστημα να έχει μια εντελώς δυναμική αρχιτεκτονική τοποθέτησης των αρχών έκδοσης στην οποία κάθε κόμβος ανάλογα με την ζήτηση που έχουν οι αρχές έκδοσης είτε τις φέρνει τοπικά (**copy**) είτε τις διώχνει μακριά (**drop**). Το πρόβλημα, λοιπόν, τίθεται στην γενική του περίπτωση ως εξής, ποιος συνδυασμός εξυπηρετητών είναι η καλύτερη επιλογή για την εξυπηρέτηση μίας αίτησης κάποια συγκεκριμένη χρονική στιγμή όπου το «καλύτερη» έχει την έννοια του ελαχίστου χρόνου απόκρισης για τον χρήστη.

Στην επίσημη υλοποίηση του συστήματος η επιλογή των εξυπηρετητών γίνεται στατικά. Έτσι, για κάθε αρχή έκδοσης υπάρχουν δύο περιπτώσεις. Πρώτη περίπτωση, ο εξυπηρετητής δεικτοδότησης της αρχής έκδοσης βρίσκεται στην ίδια περιοχή με τον εξυπηρετητή στον οποίο υποβλήθηκε η ερώτηση. Τότε επιλέγεται ο εξυπηρετητής δεικτοδότησης της. Σε περίπτωση που αυτός αποτύχει να εξυπηρετήσει την ερώτηση τότε η ερώτηση προωθείται στον εξυπηρετητή υποστήριξης της περιοχής. Διαφορετικά, εάν δηλαδή ο εξυπηρετητής στον οποίο υποβλήθηκε η ερώτηση και ο εξυπηρετητής δεικτοδότησης της αρχής έκδοσης βρίσκονται σε

διαφορετικές περιοχές, επιλέγεται ο εξυπηρετητής συγχωνευμένου καταλόγου της περιοχής του εξυπηρετητή που υποβλήθηκε η ερώτηση. Σε περίπτωση που ο εξυπηρετητής αυτός αποτύχει να εξυπηρετήσει την ερώτηση τότε η ερώτηση προωθείται στον εξυπηρετητή δεικτοδότησης της αρχής έκδοσης.

Έτσι για παράδειγμα, έστω ο τοπικός εξυπηρετητής που βρίσκεται στην περιοχή A δέχεται μία αίτηση για τις αρχές έκδοσης A, B, C και D (βλέπε στο Σχήμα 18). Τότε ο τοπικός εξυπηρετητής απευθύνεται στους εξυπηρετητές 1, 2 (τους εξυπηρετητές δεικτοδότησης για τις αρχές A και B αντίστοιχα, που βρίσκονται στην ίδια περιοχή με αυτόν, την περιοχή A) και στον εξυπηρετητή συγχωνευμένου καταλόγου της περιοχής A για τις αρχές έκδοσης C και D, των οποίων οι εξυπηρετητές δεικτοδότησης βρίσκονται στην περιοχή B (κυανά βέλη). Εάν κάποιος από τους εξυπηρετητές 1 και 2 αποτύχει να εξυπηρετήσει την ερώτηση τότε η ερώτηση προωθείται στον εξυπηρετητή υποστήριξης της περιοχής A (το κόκκινο βέλος). Εάν ο εξυπηρετητής συγχωνευμένου καταλόγου της περιοχής A αποτύχει να εξυπηρετήσει την ερώτηση για τις αρχές έκδοσης C και D, τότε η ερώτηση προωθείται στους εξυπηρετητές 3 και 4 αντίστοιχα, που είναι οι εξυπηρετητές δεικτοδότησης των αρχών έκδοσης C και D. Οι δείκτες στα ονόματα των αρχών έκδοσης δείχνουν την σειρά που έχει ο εξυπηρετητής που τις δεικτοδοτεί στην λίστα των εξυπηρετητών που κρατά για κάθε αρχή έκδοσης ο τοπικός εξυπηρετητής και είναι η ίδια για κάθε εξυπηρετητή της περιοχής A. Στην περιοχή B η σειρά των εξυπηρετητών στην λίστα είναι διαφορετική.



Σχήμα 18: Στατική Δρομολόγηση των Ερωτήσεων στο Σύστημα NCSTRL-Dienst.

Όπως προαναφέρθηκε, μία στατική δρομολόγηση των ερωτήσεων στο σύστημα NCSTRL-Dienst δεν μπορεί να είναι βέλτιστη γιατί αδυνατεί να παρακολουθήσει την δυναμική που έχει εγγενώς το σύστημα. Επομένως, σχεδιάστηκαν ενδεικτικές πολιτικές δυναμικής δρομολόγησης με βάση την στρατηγική δυναμικής δρομολόγησης των ερωτήσεων που προτείνεται στην παράγραφο 4.4, υλοποιήθηκαν και διεξήχθησαν μερικά πειράματα για την ανάδειξη της χρησιμότητάς τους.

7.4.1. Αλγόριθμοι δυναμικής δρομολόγησης ερωτήσεων.

Οι αλγόριθμοι δυναμικής δρομολόγησης ερωτήσεων βασίζονται σε εκτιμήσεις του χρόνου απόκρισης των εναλλακτικών επιλογών εξυπηρετητών, που παρέχει ο

επόπτης επίδοσης και στόχο έχουν την ελαχιστοποίηση του συνολικού χρόνου απόκρισης του συστήματος. Συγκεκριμένα, έστω ότι θέλουμε να δρομολογήσουμε μία ερώτηση για τις αρχές έκδοσης α_i , όπου $i = 1, \dots, N$, και έστω ότι $ER_{jk}^{(\alpha_1, \dots, \alpha_{n_k})}$ είναι ο εκτιμώμενος χρόνος απόκρισης του εξυπηρετητή S_k για μία ερώτηση από τον εξυπηρετητή S_j για τις αρχές έκδοσης α_i , $i = 1, \dots, n_k$ και M το πλήθος των εξυπηρετητών. Τότε το πρόβλημα επιλογής των εξυπηρετητών για την δρομολόγηση των ερωτήσεων για όλες τις αρχές έκδοσης εκφράζεται ως εξής:

Επέλεξε: $S_k, k = 1, \dots, M$

Για τις αρχές έκδοσης: $\alpha_i, i = 1, \dots, N$

$$\text{Τέτοιο ώστε: } \min_{\forall (S_1, S_2, \dots, S_M) \mid S_k \in (S_1, S_2, \dots, S_M)} \left\{ \max_{k=1, \dots, M} \left\{ ER_{jk}^{(\alpha_1, \dots, \alpha_{n_k})} \right\} \right\}$$

(9) Βέλτιστος Αλγόριθμος Δυναμικής Δρομολόγησης.

Το πρόβλημα είναι η επιλογή των εξυπηρετητών εκείνων που με κατάλληλη ανάθεση των υποερωτήσεων ελαχιστοποιούν τον συνολικό εκτιμώμενο χρόνο απόκρισης. Ο συνολικός εκτιμώμενος χρόνος απόκρισης είναι ο μέγιστος από όλους τους επιλεγμένους εξυπηρετητές εκτιμώμενος χρόνος απόκρισης.

Ο παραπάνω αλγόριθμος συνεπάγεται ουσιαστικά την εκτίμηση του συνολικού χρόνου για κάθε δυνατή ομάδα από εξυπηρετητές και η πολυπλοκότητα του αυξάνει εκθετικά με τον αριθμό των δυνατών επιλογών εξυπηρετητών, πράγμα που τον κάνει ουσιαστικά ασύμφορο. Έτσι, επιλέχθηκε μία τροποποιημένη έκδοση του αλγορίθμου η οποία είναι σημαντικά καλύτερης πολυπλοκότητας αλλά, όπως είναι αναμενόμενο, δεν δίνει βέλτιστη λύση. Η παραλλαγή έχει ως εξής:

Επέλεξε: $S_k, k = 1, \dots, M$

Για τις αρχές έκδοσης: $\alpha_i, i = 1, \dots, N$

$$\text{Τέτοιο ώστε: } \min_{\forall S_k} \left\{ ER_{jk}^{\alpha_i} \right\}$$

(10) Τροποποιημένος Αλγόριθμος Δυναμικής Δρομολόγησης.

Το πρόβλημα είναι να επιλεγεί για κάθε αρχή έκδοσης ο εξυπηρετητής που έχει τον ελάχιστο εκτιμώμενο χρόνο απόκρισης. Ο αλγόριθμος αυτός δεν είναι βέλτιστος διότι δεν λαμβάνει υπόψιν του την περίπτωση για δύο διαφορετικές αρχές έκδοσης ο εξυπηρετητής με τον ελάχιστο εκτιμώμενο χρόνο να είναι ο ίδιος. Στην περίπτωση αυτή στην εκτίμηση του συνολικού εκτιμώμενου χρόνου απόκρισης θα μπει ο εκτιμώμενος χρόνος απόκρισης για την εξυπηρέτηση των δύο αρχών έκδοσης από τον εξυπηρετητή, ο οποίος είναι φυσικά μεγαλύτερος του εκτιμώμενου χρόνου απόκρισης για μία αρχή έκδοσης. Είναι εμφανές ότι ο τροποποιημένος αλγόριθμος αποφασίζει σε πολύ μικρότερο χρόνο, πράγμα που είναι ιδιαίτερα σημαντικό γιατί η εκτέλεση στο σύστημα δεν μπορεί να προχωρήσει μέχρι να ληφθεί η απόφαση.

7.4.2. Θέματα υλοποίησης.

Τα ζητήματα που προέκυψαν στην υλοποίηση του παραπάνω αλγορίθμου ήταν δύο κατηγοριών. Πρώτον, υπήρξαν κάποιες αποφάσεις σε σχέση με τον ίδιο τον αλγόριθμο, όπως με ποιο τρόπο και κάθε πότε θα ενημερώνεται ο εξυπηρετητής για τις εκτιμήσεις του χρόνου απόκρισης των εξυπηρετητών δεικτοδότησης και με ποιο τρόπο θα υπολογίζονται οι εκτιμήσεις αυτές από τους χρόνους απόκρισης που καταγράφει ο επόπτης επίδοσης. Από την άλλη υπήρξαν και θέματα σχετικά με την ενσωμάτωση του αλγορίθμου στον κώδικα του συστήματος χωρίς να επηρεαστεί η λειτουργία του.

Όσον αφορά τα θέματα του αλγορίθμου, το σύστημα επιλέχθηκε να ενημερώνεται για τις εκτιμήσεις των χρόνων απόκρισης με την χρήση του πρωτοκόλλου επικοινωνίας με τον επόπτη επίδοσης (βλέπε στο Παράρτημα 1, για το πρωτόκολλο επικοινωνίας με την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης). Έτσι, το σύστημα ζητά τις εκτιμήσεις για τις παραμέτρους επίδοσης της μορφής $T_X^X \rightarrow Y$ (βλέπε στο Παράρτημα 5). Η ενημέρωση γίνεται στην φάση της επιλογής των εξυπηρετητών δεικτοδότησης για κάθε ερώτηση. Για την εκτίμηση του χρόνου απόκρισης χρησιμοποιείται ένας σταθμισμένος μέσος όρος των καταγεγραμμένων χρόνων απόκρισης, όπου όσο μία μέτρηση γίνεται παλιότερη το βάρος της στην εκτίμηση του χρόνου απόκρισης ελαττώνεται.

Η εισαγωγή του αλγορίθμου στο σύστημα έγινε με τρόπο τέτοιο που να μην μεταβάλλει το βασικό σκεπτικό της διαδικασίας επιλογής των εξυπηρετητών δεικτοδότησης. Σύμφωνα με το σκεπτικό αυτό κάθε αρχή έκδοσης έχει μία λίστα από εξυπηρετητές που την δεικτοδοτούν. Η σειρά των εξυπηρετητών στην λίστα καθορίζεται με βάση προκαθορισμένη προτεραιότητα (**priority**). Όλοι οι εξυπηρετητές κάθε στιγμή βρίσκονται σε μία από δύο δυνατές καταστάσεις ενήμεροι (**up to date**) ή όχι. Για κάθε, επομένως, αρχή έκδοσης επιλέγεται ο πρώτος ενήμερος εξυπηρετητής δεικτοδότησης από την αντίστοιχη λίστα των εξυπηρετητών. Με την εισαγωγή του αλγορίθμου αναδιατάσσεται η λίστα των εξυπηρετητών λίγο πριν την επιλογή, ώστε η σειρά των εξυπηρετητών να είναι με αύξοντα εκτιμώμενο χρόνο απόκρισης.

Τέλος, έχουν υλοποιηθεί δύο παραλλαγές του αλγορίθμου:

- 1) Ο εξυπηρετητής φροντίζει να στείλει μία ερώτηση σε κάθε εξυπηρετητή που παραμένει ανενεργός για αρκετό καιρό (δηλαδή έχει περάσει μεγάλο χρονικό διάστημα από την τελευταία φορά που επιλέχθηκε) για ενημερώσει την εκτίμηση για τον χρόνο απόκρισης του.
- 2) Ο εξυπηρετητής δεν επιλέγει με βάση τον εκτιμώμενο χρόνο απόκρισης αλλά με βάση κάποια πιθανότητα επιλογής που αποκτά κάθε εξυπηρετητής ανάλογα με τον εκτιμώμενο χρόνο απόκρισης του (οι γρήγοροι εξυπηρετητές έχουν μεγάλη πιθανότητα επιλογής).

7.4.3. Πειραματική μελέτη.

Για την πειραματική μελέτη των δύο παραπάνω παραλλαγών του τροποποιημένου αλγορίθμου στήθηκε μία πλατφόρμα πειραμάτων. Η πλατφόρμα αποτελείται από έναν εξυπηρετητή Dienst στον οποίο υποβάλλονται ερωτήσεις από ένα *script* και οκτώ εξυπηρετητές που εξυπηρετούν τις ερωτήσεις. Οι ερωτήσεις που στέλνει το *script* προέκυψαν μετά από ανάλυση των αρχείων καταγραφής του εξυπηρετητή που βρίσκεται στο πανεπιστήμιο Cornell.

Πλατφόρμα Πειραμάτων.

Η πλατφόρμα αποτελείται από εννέα εξυπηρετητές τοποθετημένους σε διαφορετικά μηχανήματα που είναι διασκορπισμένα στο παγκόσμιο δίκτυο. Συγκεκριμένα, ένας εξυπηρετητής λειτουργεί ως πηγή φόρτου για το σύστημα. Ο εξυπηρετητής αυτός βρίσκεται σε ένα μηχάνημα DEC/Alpha με 64M μνήμη και έχει μια πολύ μικρή βάση εγγράφων, η οποία δεν χρησιμοποιείται κατά την πειραματική μελέτη. Οι υπόλοιποι οκτώ εξυπηρετητές έχουν ανά δύο την ίδια βάση εγγράφων. Κάθε βάση εγγράφων έχει έγγραφα από τρεις διαφορετικές αρχές έκδοσης, συνολικά 10 αρχές έκδοσης. Οι οκτώ εξυπηρετητές είναι τοποθετημένοι ένας στην Γερμανία, ένας στην Ιταλία, ένας στις Η.Π.Α και οι υπόλοιποι στην Ελλάδα, ένας στο Πανεπιστήμιο Κρήτης και τέσσερις στο Ινστιτούτο Πληροφορικής (ΙΠ). Οι τέσσερις εξυπηρετητές που βρίσκονται στο ΙΠ είναι ψευδομακρινοί ως προς τον εξυπηρετητή παραγωγής φορτίου. Αυτό σημαίνει ότι κατά την εκτέλεση του *CGI stub* κάνουν μία τυχαία καθυστέρηση ώστε να έχουν χρόνους απόκρισης παρόμοιους με τους πραγματικά μακρινούς εξυπηρετητές. Η καθυστέρηση αυτή δίνεται από τον τύπο:

$$(M * (1 - W) + W * OD) + \alpha * (V - W * |OD - W|)$$

(11) Ψευδοκαθυστέρηση Δικτύου.

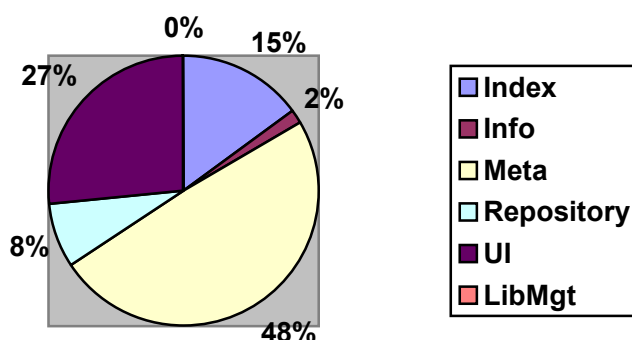
όπου M είναι η μέση τιμή της καθυστέρησης, V η τυπική απόκλιση της καθυστέρησης, OD η προηγούμενη καθυστέρησης, W ο ρυθμός με τον οποίο φθίνει ο ρόλος της προηγούμενης καθυστέρησης και α ένας τυχαίος αριθμός από 0 μέχρι 1. Ο τύπος αυτός προέκυψε από την ανάγκη να έχουμε καθυστερήσεις που σε διαδοχικές φορές υπολογισμού του τύπου, θα έχουν κοντινές τιμές, όπως περίπου συμβαίνει και στο πραγματικό δίκτυο. Τέλος, τα χαρακτηριστικά των οκτώ εξυπηρετητών αναλυτικά είναι τα εξής:

1. Εξυπηρετητής στην Ιταλία, SunOS 5.5.1, Sun4m με 32M και 504 έγγραφα, αρχές έκδοσης A, B, Γ.
2. Εξυπηρετητής στην Γερμανία, SunOS 5.5.1, Sun4m με 150M και 504 έγγραφα, αρχές έκδοσης A, B, Γ.
3. Εξυπηρετητής στο Πανεπιστήμιο Κρήτης, SunOS 4.1.4, Sun4m με 27M και 370 έγγραφα, αρχές έκδοσης Δ, E, Z.
4. Εξυπηρετητής 1 στο ΙΠ, Digital Unix 4.0B, DEC/Alpha με 64M και 370 έγγραφα, αρχές έκδοσης Δ, E, Z.
5. Εξυπηρετητής στις Η.Π.Α., SunOS 5.5, Sun4m με 120M και 315 έγγραφα, αρχές έκδοσης Z, H, Θ.
6. Εξυπηρετητής 2 στο ΙΠ, Digital Unix 4.0B, DEC/Alpha με 32M και 315 έγγραφα, αρχές έκδοσης Z, H, Θ.
7. Εξυπηρετητής 3 στο ΙΠ, Digital Unix 3.2, DEC/Alpha με 64M και 1251 έγγραφα, αρχές έκδοσης Γ, I, K.
8. Εξυπηρετητής 4 στο ΙΠ, Digital Unix 4.0B, DEC/Alpha με 96M και 1251 έγγραφα, αρχές έκδοσης Γ, I, K.

Ανάλυση των Αρχείων Καταγραφής του Cornell.

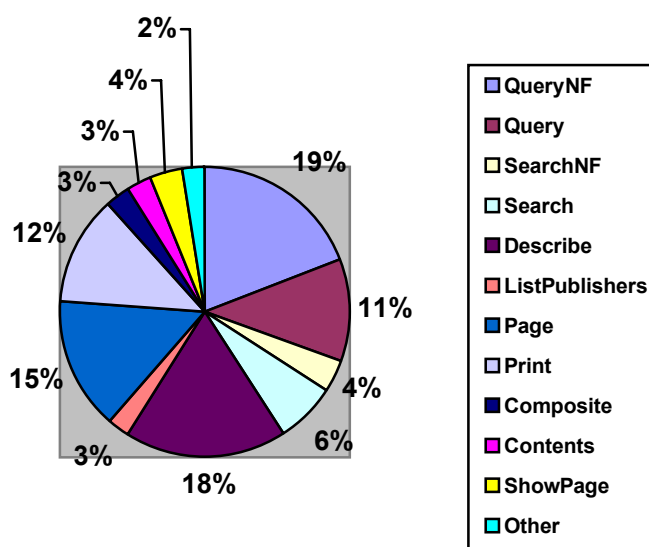
Η ανάλυση αφορά το αρχείο καταγραφής του εξυπηρετητή Dienst στο πανεπιστήμιο Cornell, το πανεπιστήμιο που υλοποίησε το σύστημα NCSTRL-Dienst. Τα αρχεία καταγραφής καλύπτουν ολόκληρο το 1996 και περιλαμβάνουν μονάχα τις δοσοληψίες (**transactions**) του συστήματος, δηλαδή εξαιρέθηκαν τα μηνύματα προειδοποίησης (**warnings**) και τα μηνύματα λάθους (**errors**). Συνολικά την περίοδο

αυτή ο εξυπηρετητής δέχτηκε 906,890 δοσοληψίες. Οι δοσοληψίες ανά κατηγορία υπηρεσίας χωρίζονται ως εξής:



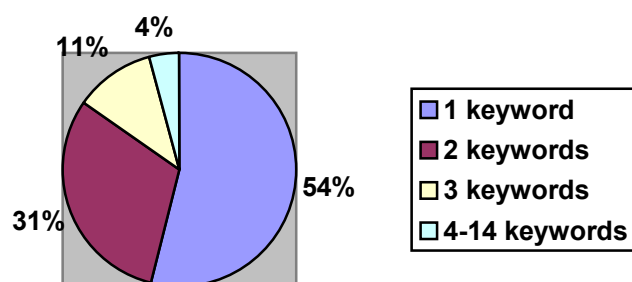
(12) Αιτήσεις ανά Κατηγορία Υπηρεσίας.

Η ενδιαφέρουσα υπηρεσία για μας είναι η υπηρεσία διεπιφάνειας χρήσης (UI) γιατί περιλαμβάνει τις αιτήσεις αναζήτησης των χρηστών (*Query* και *QueryNF*). Οι αιτήσεις ενδιαφέροντος αποτελούν το 40% των αιτήσεων διεπιφάνειας χρήσης και το 16% των συνολικών αιτήσεων.



(13) Είδη Αιτήσεων Διεπιφάνειας Χρήσης (UI).

Αναλύοντας παραπέρα τις αιτήσεις Query (αναζήτηση με βάση φόρμα πεδίων για ορισμένο αριθμό από αρχές έκδοσης) και QueryNF (αναζήτηση με απλή λέξη κλειδί σε όλες τις αρχές έκδοσης). Έχουμε ότι οι αιτήσεις Query αποτελούν το 37% των ενδιαφερόντων αιτήσεων και οι αιτήσεις QueryNF το 63%. Αναλύοντας, τις αιτήσεις QueryNF ως προς τον αριθμό των λέξεων κλειδίων έχουμε:



(14) Αριθμός Λέξεων Κλειδιών στις Αιτήσεις QueryNF.

Αναλύοντας τις αιτήσεις Query διαπιστώνουμε ότι απευθύνονται προς όλες τις αρχές έκδοσης κατά 62%.

Η γεννήτρια φόρτου θα πρέπει να παράγει κατά 70% αιτήσεις QueryNF και κατά 30% αιτήσεις Query. Οι αιτήσεις QueryNF κατά 50% έχουν μία λέξη κλειδί κατά 30% δύο και κατά 20% τρεις. Τέλος, οι αιτήσεις Query κατά 60% αφορούν όλες τις αρχές έκδοσης και κατά 40% κάποια από αυτές.

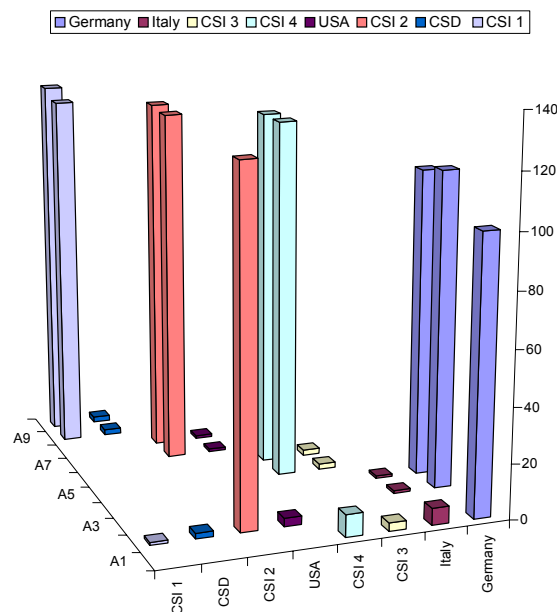
Πειραματικά Αποτελέσματα.

Στην παράγραφο αυτή παρουσιάζονται δύο πειράματα για κάθε παραλλαγή του αλγορίθμου εξισορρόπησης φόρτου, που παρουσιάστηκε παραπάνω. Τα δύο πειράματα κάθε παραλλαγής έγιναν σε διαφορετική ώρα και μέρα. Όλα τα πειράματα έγιναν με την αποστολή 50 αιτήσεων που έχουν τα χαρακτηριστικά που περιγράφηκαν παραπάνω στην ανάλυση του αρχείου καταγραφής του Cornell. Το σύνολο των αιτήσεων υποβλήθηκε τουλάχιστον δύο φορές. Οι αιτήσεις στάλθηκαν σε έναν εξυπηρετητή, ο οποίος έπαιξε τον ρόλο της πηγής φόρτου και αφορούσαν τις αρχές έκδοσης των υπολοίπων οκτώ εξυπηρετητών. Το μοντέλο αποστολής των αιτήσεων ήταν κλειστό, δηλαδή περιμένουμε πρώτα να έρθει η απάντηση μιας αίτησης πριν να στείλουμε την επόμενη. Μεταξύ των αιτήσεων μεσολαβεί ένα χρονικό διάστημα αναμονής (περίπου 15 secs). Επιλέχθηκε αυτό το μοντέλο υποβολής αιτήσεων λόγω των περιορισμών που έχει η παρούσα υλοποίηση του συστήματος NCSTRL-Dienst στον ρυθμό αιτήσεων που μπορεί να εξυπηρετήσει. Επίσης, παρά τον χαμηλό φόρτο του συστήματος υπάρχουν περιπτώσεις κατά την διάρκεια του πειράματος που ο εξυπηρετητής πηγή φόρτου αποτυγχάνει στην εξυπηρέτηση μίας αίτησης, ουσιαστικά κολλάει. Τα παραπάνω έχουν ως συνέπεια κάθε πείραμα να διαρκεί πάνω από δύο ώρες.

Παρακάτω, ακολουθούν ένα γράφημα για κάθε πείραμα, στα γραφήματα αυτά στον ένα άξονα έχουμε τους οκτώ εξυπηρετητές της πλατφόρμας που περιγράφηκε σε προηγούμενη παράγραφο, στον άλλον είναι οι δέκα αρχές έκδοσης που δεικτοδοτούνται από τους εξυπηρετητές αυτούς και τέλος οι τιμές στον τρίτο άξονα δείχνουν τον αριθμό των μετρήσεων που έγιναν για κάθε αρχή έκδοσης σε κάθε εξυπηρετητή. Κάθε γράφημα συνοδεύεται και από έναν πίνακα που περιέχει τα δεδομένα με βάση τα οποία έγινε το γράφημα. Με έντονα γράμματα σε αυτούς τους πίνακες έχουμε το πλήθος των μετρήσεων για κάθε αρχή έκδοσης σε κάθε εξυπηρετητή (ουσιαστικά το νούμερο που εμφανίζεται και στο γράφημα), ενώ με κανονικά γράμματα είναι ο αριθμός των φορών που επιλέχθηκε ο συγκεκριμένος

εξυπηρετητής για την συγκεκριμένη αρχή έκδοσης. Οι διαφορές στα δύο νούμερα οφείλονται συνήθως σε υπερβάσεις των χρονικών περιθωρίων του συστήματος. Τέλος, είναι σημαντικό να έχουμε υπόψιν ότι σε περίπτωση υπέρβασης του χρονικού περιθωρίου από κάποιο εξυπηρετητή, ο εξυπηρετητής αυτός θεωρείται ανενεργός για κάποιο χρονικό διάστημα (περίπου μία ώρα). Τέλος, μία γενική παρατήρηση για τα αποτελέσματα είναι ότι σε όλες τις περιπτώσεις οι χρόνοι απόκρισης γίνονταν χειρότερη καθώς περνούσε η ώρα λόγω της αύξησης της κίνησης στο δίκτυο με το πέρασμα της ώρας.

Έτσι, αναλυτικότερα, στην περίπτωση της πρώτης παραλλαγής του αλγορίθμου, έχουμε θέσει το χρονικό διάστημα για την ενημέρωση ενός κόμβου στα 10 λεπτά. Στο πρώτο γράφημα το σύνολο των πενήντα αιτήσεων στάλθηκε 3 φορές, ενώ στο δεύτερο δύο. Βλέποντας τα δύο γραφήματα διαπιστώνουμε ότι σε γενικές γραμμές είναι παρόμοια, δηλαδή ο εξυπηρετητής μεταξύ των ομοίων που επιλέγεται στην πρώτη περίπτωση είναι ο ίδιος και στην δεύτερη. Το γεγονός αυτό οφείλεται στην ποιότητα της δικτυακής σύνδεσης των εξυπηρετητών. Συγκεκριμένα, οι εξυπηρετητές που βρίσκονται στο **CSI** εμφανίζονται επιθυμητοί έναντι των άλλων εξυπηρετητών διότι παρόλο που είναι ψευδομακρινοί και εμφανίζουν παρόμοιους χρόνους απόκρισης, δεν υπερβαίνουν ποτέ τα μέγιστα χρονικά περιθώρια, σε αντίθεση με τους πραγματικά μακρινούς εξυπηρετητές. Από τους πραγματικά μακρινούς εξυπηρετητές μόνο αυτός στην Γερμανία (**Germany**) εμφανίζει καλή συνδεσιμότητα. Για αυτό το λόγο ο εξυπηρετητής αυτός αποτελεί την επιλογή, και στις δύο περιπτώσεις, για την περίπτωση της αρχής έκδοσης **dartmouth**, που βρίσκεται σε τέσσερις εξυπηρετητές (**Germany, Italy, CSI 3, CSI 4**). Ειδικότερα στην δεύτερη περίπτωση το πείραμα έγινε κοντά στο μεσημέρι, που είναι ώρα μεγάλης κίνησης με αποτέλεσμα οι πραγματικά μακρινοί εξυπηρετητές να μην χρησιμοποιηθούν σχεδόν καθόλου αφού υπερέβησαν τα χρονικά περιθώρια επανηλειμένα. Αντίθετα το πρώτο πείραμα έγινε νωρίς το πρωί που η κίνηση στο δίκτυο είναι σαφώς χαμηλότερη. Καθώς, όμως, περνούσε η ώρα άρχισαν και πάλι να εμφανίζονται τα προβλήματα στην σύνδεση με τους πραγματικά μακρινούς εξυπηρετητές.

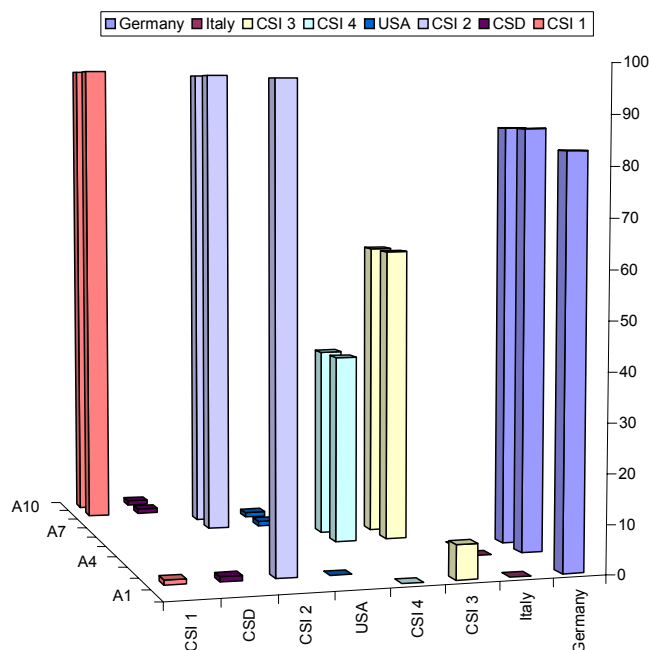


(15) Δρομολόγηση Ερωτήσεων Πρώτης Παραλλαγής 1.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Ger.	100 115		114 132	111 129						
Ital.	1 18		1 18	5 30						
CSI3	3 3				2 2	2 2				
CSI4	8 8				128 128	128 128				
USA		3 3					1 3	1 3		
CSI2		126 126					128 128	128 128		
CSD		2 2							2 3	2 2
CSI1		1 1							129 129	129 129

(16) Πίνακας Δρομολόγησης Πρώτης Παραλλαγής 1.

Μοναδική εξαίρεση στην ομοιομορφία των δύο γραφημάτων αποτελεί η περίπτωση των εξυπηρετητών **CSI 3** και **CSI 4** στους οποίους ενώ στο πρώτο πείραμα υπάρχει μία σαφής προτίμηση στον **CSI 4** στο δεύτερο πείραμα οι αιτήσεις είναι πιο μοιρασμένες και μάλιστα υπάρχει μια ελαφρά προτίμηση στον **CSI 3**. Το γεγονός αυτό οφείλεται στο ότι κατά την διάρκεια του πρώτου πειράματος, στο μηχάνημα που βρίσκεται ο εξυπηρετητής **CSI 3** δούλεψε και κάποιος χρήστης καταναλώντας πόρους τους μηχανήματος και ρίχνοντας των χρόνο απόκρισης του.

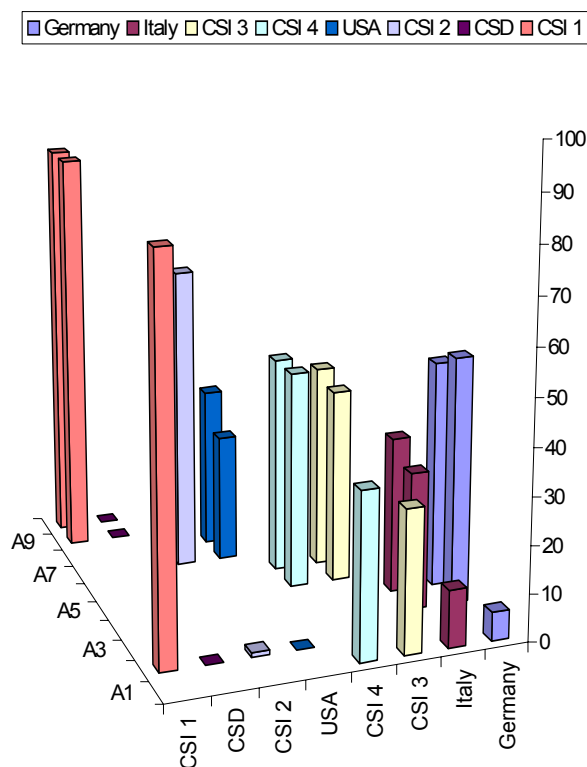


(17) Δρομολόγηση Ερωτήσεων Πρώτης Παραλλαγής 2.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Germ	83		87	87						
Ital	0		0	0						
CSI3	7				61	61				
CSI4	0				39	39				
USA		0					1	1		
CSI2		97					98	98		
CSD		1							1	1
CSI1		1							99	99

(18) Πίνακας Δρομολόγησης Πρώτης Παραλλαγής 2.

Στην περίπτωση της δεύτερης παραλλαγής του αλγορίθμου και στα δύο πειράματα το σύνολο των πενήντα αιτήσεων έχει σταλεί δύο φορές. Η πρώτη παρατήρηση είναι ότι στην περίπτωση αυτή, οι αιτήσεις μοιράζονται πιο πολύ μεταξύ των ομοίων εξυπηρετητών. Το γεγονός αυτό οφείλεται στον αλγόριθμο δυναμικής δρομολόγησης. Έτσι, επειδή οι διαφορές στον χρόνο απόκρισης μεταξύ των εξυπηρετητών είναι μικρές, η πιθανότητα επιλογής καθενός από τους όμοιους εξυπηρετητές είναι περίπου η ίδια. Γενικά και στα δύο πειράματα τα αποτελέσματα είναι παρόμοια, δηλαδή για κάθε αρχή έκδοσης υπάρχει μια προτίμηση προς τον ίδιο εξυπηρετητή και μάλιστα οι προτιμώμενοι εξυπηρετητές είναι οι ίδιοι και στις δύο παραλλαγές του αλγορίθμου δρομολόγησης.

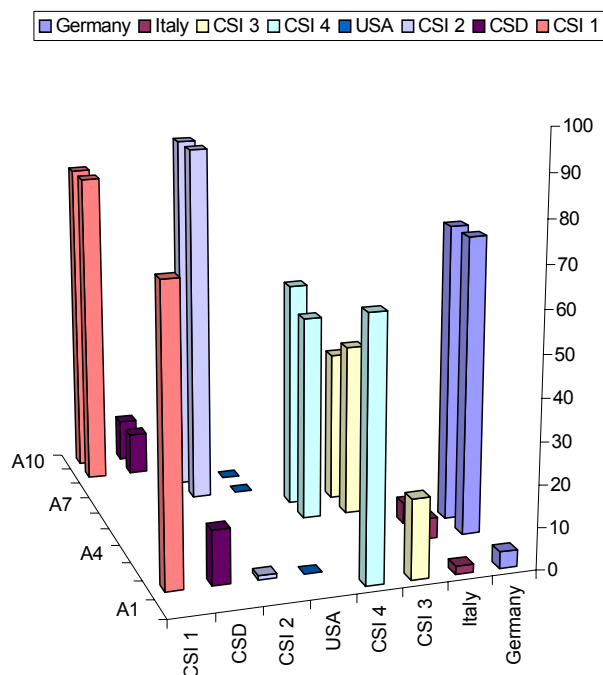


(19) Δρομολόγηση Ερωτήσεων Δεύτερης Παραλλαγής 1.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Germ	6 10		52 65	48 55						
Ital	12 13		29 35	33 45						
CSI3	30 32				41 43	43 45				
CSI4	35 35				46 46	46 46				
USA		0 1					27 33	34 40		
CSI2		1 1					64 66	57 59		
CSD		0 10							0 10	0 10
CSI1		83 86							85 87	85 87

(20) Πίνακας Δρομολόγησης Δεύτερης Παραλλαγής 1.

Η βασική διαφορά που παρατηρείται στα δύο γραφήματα είναι ότι στο δεύτερο έχουμε γενικά περισσότερες αιτήσεις προς τους πραγματικά μακρινούς εξυπηρετητές από το πρώτο. Το γεγονός αυτό οφείλεται στο ότι το δεύτερο πείραμα έγινε Σάββατο πρωί όπου η κίνηση στο δίκτυο είναι γενικά καλή, ενώ το πρώτο έγινε καθημερινή κοντά στο μεσημέρι. Ο μόνος πραγματικά μακρινός εξυπηρετητής που δεν ακολουθεί την παραπάνω παρατήρηση είναι ο **CSD**, αυτό γιατί έτυχε την ώρα του πρώτου πειράματος να έχει πέσει η γραμμή σύνδεσης.

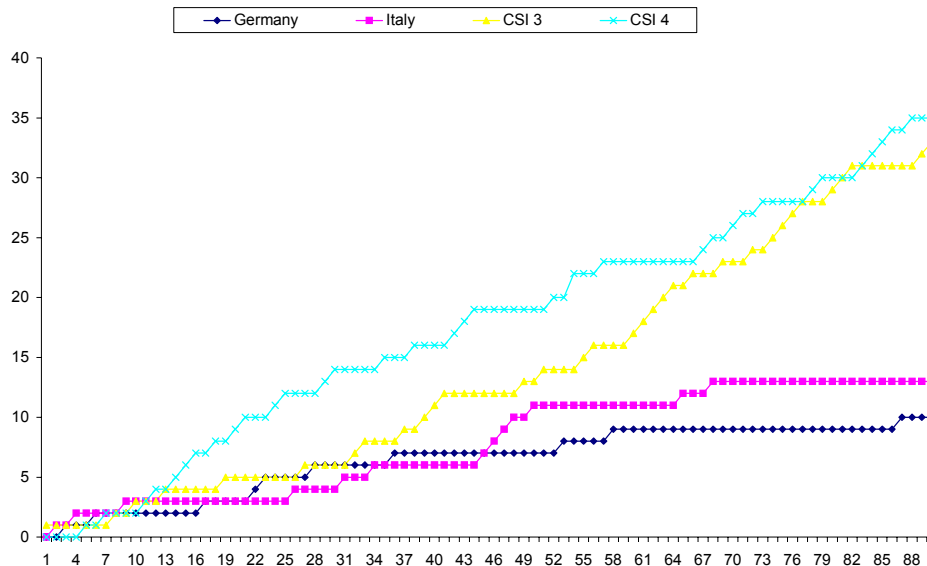


(21) Δρομολόγηση Ερωτήσεων Δεύτερης Παραλλαγής 2.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Germ	4 5		71 85	71 85						
Ital	2 14		5 24	5 24						
CSI3	19 20				39 41	35 38				
CSI4	60 62				49 49	54 54				
USA		0 13					0 15	0 15		
CSI2		1 1					86 91	86 91		
CSD		13 24							10 21	10 21
CSI1		68 70							74 76	74 76

(22) Πίνακας Δρομολόγησης Δεύτερης Παραλλαγής 2.

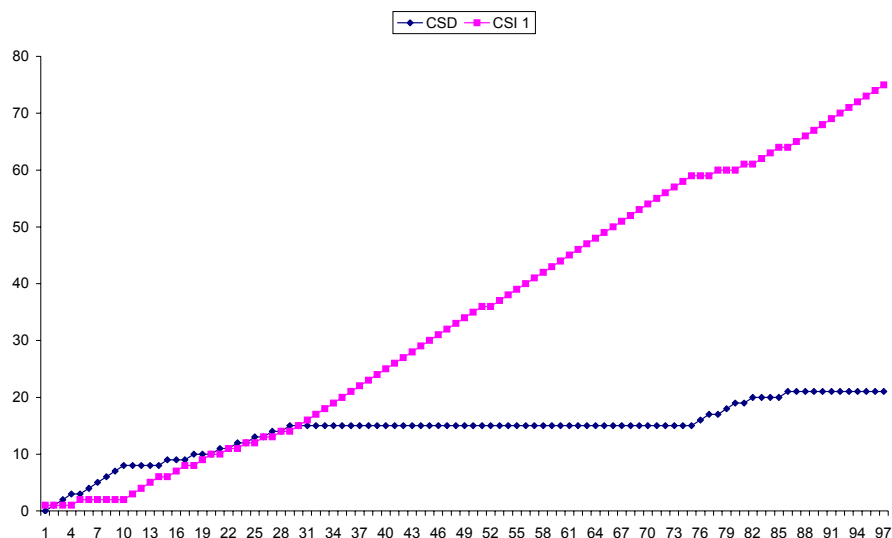
Παρακάτω ακολουθούν δύο γραφήματα που δείχνουν την εξέλιξη της επιλογής εξυπηρετητή για δύο ενδιαφέρουσες περιπτώσεις των παραπάνω πειραμάτων. Στα γραφήματα αυτά ο οριζόντιος άξονας δείχνει τον αριθμό αίτησης και ο κατακόρυφος άξονας δείχνει το πλήθος των φορών που επιλέχθηκε κάθε εξυπηρετητής μέχρι την συγκεκριμένη αίτηση. Σημαντική είναι η παρατήρηση ότι σταθερές τιμές για μεγάλο αριθμό αιτήσεων (π.χ. στο πρώτο γράφημα για την μπλε γραμμή από την αίτηση 58 μέχρι την αίτηση 88) δείχνουν περιόδους κατά τις οποίες ο συγκεκριμένος εξυπηρετητής δεν επιλέγεται από τον αλγόριθμο δρομολόγησης.



(23) Εξέλιξη των Επιλογών Δρομολόγησης στην Δεύτερη Παραλλαγή Πείραμα 1 για την Αρχή Έκδοσης A1.

Έτσι, στο πρώτο γράφημα που αφορά το πείραμα 1 της δεύτερης παραλλαγής του αλγορίθμου, έχουμε την εξέλιξη των επιλογών για την αρχή έκδοσης A1. Αυτό που βλέπουμε είναι ότι στην αρχή του πειράματος, οι εξυπηρετητές παρουσιάζουν

παρόμοιους χρόνους απόκρισης και επιλέγονται διαδοχικά. Καθώς, όμως, περνά ο χρόνος η κατάσταση για τους δύο εξυπηρετητές, **Germany** και **Italy**, χειροτερεύει και επομένως επιλέγονται όλο και αραιότερα. Από κάποια στιγμή και μετά (στην 58 αίτηση περίπου για τον **Germany** και την 64 για τον **Italy**) υπερβαίνουν το χρονικό περιθώριο και παύουν να επιλέγονται.



(24) Εξέλιξη των Επιλογών Δρομολόγησης στην Δεύτερη Παραλλαγή Πείραμα 2 για την Αρχή Έκδοσης A9.

Στο δεύτερο γράφημα, που αφορά το πείραμα 2 της δεύτερης παραλλαγής, έχουμε την εξέλιξη των επιλογών δρομολόγησης για την αρχή έκδοσης **A9**. Στην περίπτωση αυτή βλέπουμε στην αρχή και οι δύο εξυπηρετητές να παρουσιάζουν παρόμοιους χρόνους απόκρισης και να επιλέγονται διαδοχικά. Κάποια στιγμή, όμως, (περίπου στην αίτηση 30) ο εξυπηρετητής CSD υπερβαίνει το χρονικό περιθώριο οπότε για ένα μεγάλο χρονικό διάστημα (από την αίτηση 30 μέχρι την αίτηση 75) παύει να επιλέγεται. Μετά το διάστημα αυτό εξυπηρετητής επανέρχεται για λίγο (από 75 μέχρι 85 αίτηση) αλλά τελικά υπερβαίνει και πάλι το χρονικό περιθώριο και παύει να επιλέγεται μέχρι το τέλος του πειράματος.

Συμπερασματικά, βλέπουμε πως και οι δύο παραλλαγές του αλγορίθμου δυναμικής δρομολόγησης μπορούν με μικρότερη ή μεγαλύτερη επιτυχία να παρακολουθούν την δυναμική του συστήματος. Γίνεται, επομένως, εμφανές ότι θα μπορούσαν να κερδηθούν πολλά με την χρήση αλγορίθμων δυναμικής δρομολόγησης, χρειάζεται όμως μια σε βάθος μελέτη του θέματος πάνω σε μια πιο ρεαλιστική πλατφόρμα πειραμάτων για τον πλήρη καθορισμό των επιμέρους παραμέτρων.

Κεφάλαιο 8. Συμπεράσματα και Μελλοντικές Κατευθύνσεις.

Η αλματώδης ανάπτυξη του παγκόσμιου δικτυακού ιστού με τον τεράστιο όγκο της πληροφορίας που περιέχει καθώς και η πληθώρα των μορφών τις πληροφορίας αυτής κάνουν επιτακτική την ανάγκη για την δημιουργία συστημάτων που να διευκολύνουν την πρόσβαση και την διαχείριση της πληροφορίας αυτής. Για την κάλυψη της ανάγκης αυτής προτάθηκαν τα συστήματα ψηφιακών βιβλιοθηκών, ως συστήματα που συνδυάζουν, με την χρήση του παγκόσμιου δικτύου, τις διάφορες πηγές πληροφοριών και τις διάφορες διαθέσιμες υπηρεσίες για την εκμετάλλευση των πληροφοριών αυτών. Η ολοένα αυξανόμενη δημοτικότητα και η αλματώδης αύξηση του μεγέθους των συστημάτων ψηφιακής βιβλιοθήκης έφερε στο προσκήνιο τα θέματα επίδοσης τους. Συγχρόνως, οι ιδιαιτερότητες των συστημάτων αυτών κάνουν αναγκαία μια νέα προσέγγιση των θεμάτων επίδοσης, γενικότερα.

Στην εργασία αυτή προτείνεται μία πρώτη προσέγγιση στα θέματα εποπτείας επίδοσης των συστημάτων ψηφιακής βιβλιοθήκης καθώς και ένα πλαίσιο εργασίας για την υλοποίηση υπηρεσιών προστιθέμενης αξίας που δίνει σημαντικές νέες δυνατότητες στα συστήματα αυτά. Οι δυνατότητες αυτές είναι επιθυμητές και χρήσιμες τόσο από τους διαχειριστές όσο και από τους χρήστες των συστημάτων. Έτσι, ξεκινώντας από την ανάλυση επίδοσης του συστήματος προχωρούμε στην περιγραφή της αρχιτεκτονικής ενός επόπτη επίδοσης. Σκοπός του επόπτη είναι η παρατήρηση του τρόπου λειτουργίας των διαφόρων τμημάτων του συστήματος βιβλιοθήκης, ενόσω είναι ενεργό, καθώς και των αλληλεπιδράσεων (επικοινωνία) μεταξύ των τμημάτων αυτών. Με την υλοποίηση του επόπτη αποκτούμε την απαιτούμενη υποδομή για την υποστήριξη μηχανισμών όπως η παρακολούθηση της επίδοσης του συστήματος με την χρήση εργαλείων οπτικοποίησης, η εγγύηση ορισμένου επιπέδου ποιότητας υπηρεσιών, η δυναμική αναπροσαρμογή των χρονικών περιθωρίων και η Εξισορρόπηση φόρτου.

Επίσης, με την χρήση αυτού του πλαισίου εργασίας σε ένα υπάρχον και αρκετά διαδεδομένο σύστημα ψηφιακής βιβλιοθήκης, το σύστημα NCSTRL-Dienst, το οποίο μάλιστα αποτελεί και την πλατφόρμα για την δοκιμή νέων πολιτικών στα συστήματα ψηφιακής βιβλιοθήκης γενικότερα, οι δυνατότητες αυτού του πλαισίου εργασίας γίνονται περισσότερο εμφανείς. Έτσι, ξεκινώντας από μια περιγραφή του συστήματος και προχωρώντας σε μια ανάλυση της επίδοσης του, όχι μόνο έχουμε ένα ολοκληρωμένο παράδειγμα εφαρμογής των διαδικασιών αλλά είμαστε και σε θέση να προχωρήσουμε σε μια πρωταρχική μελέτη επίδοσης του συστήματος καθώς και σε παρουσίαση του τρόπου εφαρμογής κάποιων πρώτων υπηρεσιών προστιθέμενης αξίας. Αναλυτικότερα, η μελέτη επίδοσης του παραπάνω συστήματος οδηγεί σε μια σειρά από βελτιώσεις στην αρχιτεκτονική του, οι οποίες θα το κάνουν ικανό να ανταποκριθεί ικανοποιητικά στο αυξανόμενο μέγεθος και φόρτο λειτουργίας του.

Παράλληλα, δύο παραλλαγές μιας πολιτικής εξισορρόπησης φόρτου δίνουν μια πρώτη γεύση για τις νέες δυνατότητες του συστήματος.

Υπάρχουν διάφορες κατευθύνσεις στις οποίες θα μπορούσε να προχωρήσει η παρούσα εργασία στο μέλλον. Σε πρώτη φάση θα μπορούσε να επεκταθεί η αρχιτεκτονική του επόπτη επίδοσης ώστε πέρα από στοιχεία για την λειτουργία του συστήματος να παρέχει και στοιχεία για τους χρήστες του, π.χ. καταγραφή των ενεργειών κάθε χρήστη, προσδιορισμός συνηθισμένων προτύπων πρόσβασης (**access patterns**) κ.α.. Για να μπορεί, βέβαια, ο επόπτης επίδοσης να παρακολουθεί τέτοια στοιχεία θα πρέπει το σύστημα ψηφιακής βιβλιοθήκης να υποστηρίζει λειτουργίες όπως η έννοια της συνεδρίας (**session**) και μηχανισμούς διάκρισης των χρηστών όπως λογαριασμούς χρήσης (**accounts**) κ.τ.λ. Η εποπτεία στοιχείων που σχετίζονται με τον κάθε χρήστη θα δώσουν την δυνατότητα στα συστήματα ψηφιακής βιβλιοθήκης να υλοποιήσουν και πολιτικές δυναμικής χρέωσης των διαφόρων υπηρεσιών που παρέχουν. Σε μία επόμενη φάση θα μπορούσε ο επόπτης να υποστηρίξει το πρότυπο της παγκόσμιας αρχιτεκτονικής μετρήσεων [25] (**Universal Measurement Architecture – UMA**) ώστε να μπορεί να εκμεταλλευτεί τις δυνατότητες για την συλλογή μετρήσεων συστήματος που παρέχει. Τέλος, μία σημαντική κατεύθυνση για την συνέχιση αυτής της εργασίας είναι η χρήση του επόπτη επίδοσης που προτείνεται και της υποδομής που αυτός παρέχει για την μελέτη διαφόρων στρατηγικών δυναμικής αναπροσαρμογής των χρονικών περιθωρίων και εξισορρόπησης φόρτου στα συστήματα ψηφιακής βιβλιοθήκης.

Βιβλιογραφία

- [1] Patrick M. Ryan, “Introduction to Perl”, Νοέμβριος 1993.
- [2] Tom Christiansen, “The Perl Data Structures Cookbook: Lists of Lists”, Οκτώβριος 1995.
- [3] Carl Lagoze και Jim Davis, “Dienst: An Architecture for Distributed Documents Libraries”, *Communications of the ACM*, 38(4), Απρίλιος 1995, p. 47.
- [4] C. Lagoze, E. Shaw, J. R. Davis και D. B. Krafft, “Dienst: Implementation Reference Manual”, TR95-1514, Cornell University, Μάιος 1995.
- [5] Carl Lagoze και Jim Davis "Dienst protocol version 4.0", NCSTRL documentation, Οκτώβριος 1995.
- [6] Jim French, Ed Fox, Kurt Maly, και Alan Selman, "Wide Area Technical Report Service: Technical Reports Online", *Communications of the ACM*, 34(4), Απρίλιος 1995, p. 45
- [7] Digital Libraries, special issue, *Communications of the ACM*, 1995, 38 (4).
- [8] Robert Kahn, και Robert Wilensky, "A Framework for Distributed Digital Object Services", CNRI Digital Library, Μάιος 1995.
- [9] William Arms, "Key Concepts in the Architecture of the Digital Library", *D-Lib Magazine*, Ιούλιος 1995.
- [10] J. Sairamesh, S. Kapidakis και C. Nikolaou, “Architectures for QoS based Retrieval in Digital Libraries”, *Workshop on Networked Information Retrieval*, SIGIR’96.
- [11] J. Sairamesh, S. Kapidakis, S. Terzis και C. Nikolaou, “Performance Framework for QoS based Searching and Retrieval in Digital Libraries”, TR97-0204, Institute of Computer Science – FORTH, Αύγουστος 1997.
- [12] M. Litzkow, M. Livny και M. W. Mutka, “Condor – A Hunter of Idle Workstations”, *Proceedings of the 8th International Conference of Distributed Computing Systems*, p. 104-111, Ιούνιος 1988.
- [13] “A Worldwide Flock of Condors: Load Sharing among Workstation Clusters”, *Journal on Future of Computer Systems*, Volume 12, 1996.
- [14] D. Ferguson, L. Georgiadis και C. Nikolaou, “Satisfying Response Time goals in a Transaction Processing System”, *Proceedings of the Conference on Parallel and Distributed Information Systems*, 1993.
- [15] A. N. Tantawi και D. Towsley, “Optimal Static Load Balancing in Distributed Computer Systems”, *Journal of the ACM*, Απρίλιος 1985, p. 445-465.
- [16] D. Ferguson, J. Sairamesh και R. Cieslak, “Black Holes, Sacrificial Lambs, and a Robust Approach to Transaction Routing”, *Proceedings of the International Conference on Parallel and Distributed Computing*, 1995.
- [17] Phillip S. Yu, Avraham Leff και Yann-Hang Lee, “On Robust Transaction Routing and Load Sharing”, *ACM Transactions on Database Systems*, p. 476-512, Σεπτέμβριος 1991.
- [18] Beth A. Sroeder, “On-Line Monitoring: A Tutorial”, *IEEE Computer*, Ιούνιος 1995.
- [19] Weiming Gu, Jeffrey Vetter και Karsten Schwan, “An Annotated Bibliography of Interactive Program Steering”, *SIGPLAN Notices*, Σεπτέμβριος 1994.

- [20] Ziya Aral και Ilya Gertner, “Non-intrusive and interactive profiling in Parasight”, *Proceedings of the ACM/SIGPLAN PPEALS*, p. 21-30, Ιούλιος 1988.
- [21] D. A. Reed, R. D. Olson, R. A. Aydt, T. Madhyastha, T. Birkett, D. W. Jensen, B. A. A. Nazief και B. K. Totty, “Scalable performance environments for parallel systems”, In Proceedings of Sixth Distributed Memory Computing Conference, p. 562-569, *IEEE Computer Society Press*, 1991.
- [22] D. A. Reed, R. A. Aydt, R. J. Noe, K. A. Shields και B. W. Schwartz, “An Overview of the Pablo Performance Analysis Environment”, Department of Computer Science, University of Illinois, Urbana Illinois, Νοέμβριος 1992.
- [23] Frank Lange, Reinold Kroeger και Martin Gergeleit, “JEWEL: Design and Implementation of a Distributed Measurement System”, *IEEE Transactions on Parallel and Distributed Systems*, Volume 3, No 6, Νοέμβριος 1992.
- [24] Richard Snodgrass, “A Relational Approach to Monitoring Complex Systems”, *ACM Transactions on Computer Systems*, Volume 6, No 2, Μάιος 1988.
- [25] “UMA Technical Information”, Performance Management Working Group, Μάρτιος 1994.
- [26] W. Gu, G. Eisenhauer, E. Kraemer, K. Schwan, J. Stasko και J. Vetter, “Falcon: On-line Monitoring and Steering of Large-Scale Parallel Programs”, GIT-CC-94-21, College of Computing, Georgia Institute of Technology.
- [27] B. P. Miller, M. D. Callaghan, J. M. Cargille, J. H. Hollingsworth, R.B. Irvin, K. L. Karavanic, K. Kunchithapadam και T. Newhall, “The Paradyn Parallel Performance Measurement Tools”, *IEEE Computer*, 28 (11), Νοέμβριος 1995.
- [28] C. Kilpatrick και K. Schwan, “ChaosMON- Application-Specific Monitoring and Display of Performance Information for Parallel and Distributed Systems”, *Proceedings of the ACM/ONR Workshop on Parallel and Distributed Debugging*, p. 57-67, Μάιος 1991.
- [29] “Perfstat”, Product description, Instrumental Inc, 1996.
- [30] “Managing Application Performance with TME10”, Technical White Paper, Tivoli Systems, 1996.
- [31] “Resource Measurement Facility”, IBM Corp., Ιούνιος 1995.
- [32] “POLYCENTER Performance Advisor for Unix”, Product Description, Computer Associates, Digital Equipment Corp., 1996.
- [33] “Continuous Profiling Infrastructure”, Product Documentation, Digital Equipment Corp., Ιούνιος 1997.
- [34] A+UMA Performance Data Manager Technical Overview”, Amdahl Corp., Οκτώβριος 1996.
- [35] S. L. Harum, W.H. Mischo και B. R. Schatz, “Federating Repositories of Scientific Literature”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [36] D. E. Fenske και J. W. Dunn, “The VARIATIONS Project at Indiana University’s Music Library”, *D-Lib Magazine*, Ιούνιος 1996.
- [37] M. A. Hearst, “Research in Support of Digital Libraries at Xerox PARC”, *D-Lib Magazine*, Μάιος 1996.
- [38] J. L. Klavans, “New Center at Columbia University for Digital Library Research”, *D-Lib Magazine*, Μάρτιος 1996.
- [39] Chris Rusbridge, “The UK Electronic Libraries Programme”, *D-Lib Magazine*, Δεκέμβριος 1995.
- [40] E. A. Fox, J. L. Eaton, G. McMilan, N. A. Kipp, L. Weiss, E. Arce και S. Guyer, “National Digital Library of Theses and Dissertations”, *D-Lib Magazine*, Σεπτέμβριος 1996.

- [41] I. H. Witten, S. J. Cunningham και M. D. Apperley, “The New Zealand Digital Library Project”, *D-Lib Magazine*, Νοέμβριος 1996.
- [42] X. Νικολάου, Σ. Καπιδάκης και Γ. Γεωργιαννάκης, “Προς μία Πανευρωπαϊκή Επιστημονική Ψηφιακή Βιβλιοθήκη”, TR96-0167, Ινστιτούτο Πληροφορικής ΙΤΕ, Μάιος 1996.
- [43] W. Birmingham, “An Agent-Based Architecture for Digital Libraries”, *D-Lib Magazine*, Ιούλιος 1995.
- [44] Virginia Ogle και Robert Wilensky, “Testbed Development for the Berkeley Digital Library Project”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [45] James Few, Michael Freeston, Randall B. Kemp, Jason Simpson, Terence Smith, Alex Wells και Qi Zeng, “The Alexandria Digital Library Testbed”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [46] Howard D. Wactlar, “Informedia Digital Video Library: Technology Outreach”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [47] Daniel E. Atkins, “The University of Michigan Digital Library Project: The Testbed”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [48] Andreas Paepcke, “Summary of Stanford’s Digital Library Testbed Design and Status”, *D-Lib Magazine*, Ιούλιος/Αύγουστος 1996.
- [49] Dienst Server <http://www.csi.forth.gr/TR>, Institute of Computer Science – FORTH
- [50] The Informedia Digital Video Library, <http://informedia.cs.cmu.edu/>
- [51] UC Berkeley Digital Library Project, <http://elib.cs.berkeley.edu/>
- [52] University of Michigan Digital Library Project, <http://http2.sils.umich.edu/UMDL/HomePage.htm>
- [53] Alexandria Digital Library, <http://alexandria.sdc.ucsb.edu/>
- [54] Stanford Integrated Digital Library Project, <http://ic-www.arc.nasa/ic/projects/aim/Stanford-DL.html>
- [55] Illinois Digital Library Initiative Project, <http://www.grainger.uiuc.edu/dli/default.html>
- [56] The Electronic Libraries Programme (*eLib*), <http://ukoln.bath.ac.u/elib/intro.html>

Παράρτημα 1. Το Πρωτόκολλο Επικοινωνίας του Επόπτη Επίδοσης.

Η επικοινωνία του επόπτη επίδοσης με το σύστημα ψηφιακής βιβλιοθήκης γίνεται με δύο διαφορετικούς τρόπους, όπως φαίνεται στο **Error! Reference source not found.** και περιγράφεται στην παράγραφο 3.2.2. “Η αρχιτεκτονική του επόπτη επίδοσης.”. Πρώτον, η επικοινωνία, που αφορά τον μηχανισμό μέτρησης των παραμέτρων επίδοσης και την καταγραφή τους, γίνεται με την αποστολή μιας σειράς μηνυμάτων εκκίνησης/τερματισμού τμήματος καθώς και κάποιων ειδικών μηνυμάτων για την πληρότητα και την ορθότητα των αρχείων καταγραφής μετρήσεων, από το σύστημα βιβλιοθήκης στην πρώτη διαδικασία μέτρησης. Δεύτερον, η επικοινωνία μεταξύ του συστήματος βιβλιοθήκης και της διαδικασίας διαχείρισης της βάσης των παραμέτρων επίδοσης γίνεται με την αποστολή ερωτήσεων (**queries**) και την λήψη απαντήσεων.

Επικοινωνία με τις διαδικασίες μετρήσεων.

Η επικοινωνία για την μέτρηση των παραμέτρων επίδοσης γίνεται με μια σειρά διαφορετικών μηνυμάτων. Τα μηνύματα αυτά συνήθως αποτελούνται από επιμέρους πεδία τα οποία διαχωρίζονται μεταξύ τους από τον χαρακτήρα ‘\|’ και έχουν ως τελευταίο χαρακτήρα το ‘\n’. Η σειρά των επιμέρους πεδίων είναι αυστηρά καθορισμένη, ενώ ο αριθμός των πεδίων και η σημασία τους προσδιορίζεται από το πρώτο πεδίο που είναι το όνομα του μηνύματος. Τα κύρια ονόματα μηνυμάτων είναι τα εξής:

- **‘OpenDebug’**: Το μήνυμα αυτό σηματοδοτεί την εκκίνηση των μετρήσεων. Πέρα από το όνομα του μηνύματος το μήνυμα περιέχει ‘το αναγνωριστικό διεργασίας (**process id**)’ του εξυπηρετητή που στέλνει το μήνυμα και ‘όνομα αρχείου’ του αρχείου καταγραφής των μετρήσεων. Η λήψη αυτού του μηνύματος αρχικοποιεί τον επόπτη επίδοσης και ανοίγει το αρχείο καταγραφής των μετρήσεων, εάν κατά την εκκίνηση του επόπτη ζητήθηκε η καταγραφή.
- **‘ShutDown’**: Το μήνυμα αυτό σηματοδοτεί τον ομαλό τερματισμό της λειτουργίας του εξυπηρετητή. Με την λήψη αυτού του μηνύματος ο επόπτης φροντίζει για την τοποθέτηση σε ένα αρχείο των τρεχόντων μετρήσεων, την μεταφορά του μηνύματος προς όλες τις διαδικασίες που τον αποτελούν και το κλείσιμο όλων των sockets. Κάθε διαδικασία που λαμβάνει το μήνυμα προετοιμάζει τον τερματισμό της.
- **‘Reset’**: Το μήνυμα αυτό σηματοδοτεί την επανεκκίνηση των μετρήσεων και μπορεί να σταλεί στον επόπτη οποτεδήποτε κρίνει το σύστημα βιβλιοθήκης. Μπορεί, ακόμη, να σταλεί και από τον διαχειριστή του συστήματος, όταν αυτός επιθυμεί. Με την λήψη αυτού του μηνύματος ο επόπτης καταγράφει την επανεκκίνηση στο αρχείο καταγραφής, μεταφέρει το μήνυμα αυτό στην διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης και αρχικοποιεί τις δομές που χρησιμοποιεί για την διαχείριση των μετρήσεων.

- **‘LogDebugEnter’**: Το μήνυμα αυτό σε συνδυασμό με το επόμενο είναι τα κύρια μηνύματα του μηχανισμού μετρήσεων. Συγκεκριμένα, το μήνυμα αυτό σηματοδοτεί την εκκίνηση της μέτρησης μιας παραμέτρου επίδοσης. Τα πεδία που απαρτίζουν το μήνυμα πέρα από τον τύπο του μηνύματος είναι τα εξής: ‘το αναγνωριστικό διεργασίας (**process id**)’ του εξυπηρετητή που στέλνει το μήνυμα, ‘όνομα αρχείου’ του αρχείου καταγραφής των μετρήσεων (όπως θα γίνει φανερό παρακάτω για κάθε ενεργοποίηση του εξυπηρετητή έχουμε διαφορετικό αρχείο καταγραφής των μετρήσεων, από τη στιγμή που τα αρχεία καταγραφής έχουν HTML μορφή μπορούμε να έχουμε συνδέσμους από το ένα στο άλλο για να δείξουμε την ροή εκτέλεσης), ‘το όνομα της παραμέτρου επίδοσης’ της οποίας η μέτρηση ξεκινά και ‘μία λίστα προαιρετικών ορισμάτων’ που χωρίζονται μεταξύ τους με κενά και καταγράφονται στο αρχείο καταγραφής μαζί με το μήνυμα (τα ορίσματα αυτά μπορούν να χρησιμοποιηθούν είτε για την καταγραφή περισσότερης πληροφορίας στα αρχεία καταγραφής που θα χρησιμοποιηθεί σε μια ανάλυση των αρχείων αυτών είτε για την διάκριση περιπτώσεων της παραμέτρου επίδοσης ιδιαίτερης σημασίας, π.χ. βλέπε τα ονόματα των αρχών έκδοσης στο Παράρτημα 5). Με την λήψη αυτού του μηνύματος ο επόπτης ενημερώνει τις δομές του για την εκκίνηση μιας νέας μέτρησης, εάν η παράμετρος επίδοσης που ορίζεται στο μήνυμα έχει ξαναεμφανιστεί, διαφορετικά ορίζει μια νέα παράμετρο επίδοσης.
- **‘LogDebugExit’**: Το μήνυμα αυτό σηματοδοτεί τον τερματισμό μίας μέτρησης. Τα πεδία του μηνύματος είναι τα ίδια με του μηνύματος εκκίνησης μέτρησης με την διαφορά ότι δεν υπάρχουν ορίσματα. Με την λήψη αυτού του μηνύματος ο επόπτης αναζητά στις δομές που διατηρεί για τον μηχανισμό μέτρησης, το αντίστοιχο μήνυμα εκκίνησης της μέτρησης. Εάν το μήνυμα εκκίνησης βρεθεί τότε ο επόπτης υπολογίζει την τιμή της παραμέτρου επίδοσης με βάση την χρονοσφραγίδα των δύο μηνυμάτων και ενημερώνει τις δομές του, την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης και όλους όσους είναι συνδεδεμένοι μαζί του (βλέπε παρακάτω για την επικοινωνία του επόπτη με τον «έξω κόσμο») και το αρχείο καταγραφής για την παραγωγή νέας μέτρησης.
- **‘in_cgi_total’**: Το μήνυμα αυτό μαζί με το επόμενο είναι ειδικά μηνύματα για την μέτρηση του χρόνου σε *CGI scripts*. Το μήνυμα αυτό σηματοδοτεί την εκκίνηση του *script*. Τα πεδία του μηνύματος πέρα από τον τύπο του είναι ‘το αναγνωριστικό διεργασίας (**process id**)’ του *script* και ένα προαιρετικό όρισμα (ο σκοπός του προαιρετικού ορίσματος είναι όμοιος με τα προαιρετικά ορίσματα των μηνυμάτων εκκίνησης μέτρησης). Με την λήψη αυτού του μηνύματος ο επόπτης ενημερώνει το βασικό αρχείο καταγραφής (αυτό που προσδιορίζεται από το μήνυμα εκκίνησης μετρήσεων) και τις δομές του για την εκκίνηση της μέτρησης του *CGI script*. Η βασική διαφορά μεταξύ της μέτρησης ενός οποιουδήποτε τμήματος και της μέτρησης του *CGI script* είναι ότι τα διάφορα τμήματα είναι ουσιαστικά διαδοχικά στην εκτέλεση και έτσι δεν υπάρχει περίπτωση σύγχυσης στην διαδικασία αντιστοίχισης των μηνυμάτων εκκίνησης και τερματισμού μέτρησης της αντίστοιχης παραμέτρου επίδοσης. Από την άλλη το *CGI script* μπορεί να ενεργοποιηθεί επανειλημμένα οδηγώντας στην ταυτόχρονη ενεργοποίηση περισσότερων του ενός αντιτύπων του. Στην περίπτωση αυτή και για την αποφυγή σύγχυσης στην αντιστοίχιση

μεταξύ μηνυμάτων εκκίνησης και τερματισμού για τις διάφορες εκφάνσεις του ίδιου script χρησιμοποιείται ως διακριτικό το αναγνωριστικό διεργασίας.

- **'out_cgi_total'**: Το μήνυμα αυτό σηματοδοτεί τον τερματισμό της μέτρησης για το *CGI script*. Τα πεδία του μηνύματος είναι τα ίδια με το παραπάνω μήνυμα με την διαφορά ότι δεν υπάρχει το προαιρετικό όρισμα. Κατά την λήψη αυτού του μηνύματος ο επόπτης βρίσκει το μήνυμα εκκίνησης *CGI script* με το ίδιο αναγνωριστικό διεργασίας και στην συνέχεια εκτελεί τις ίδιες πράξεις που εκτελεί και στην περίπτωση της ολοκλήρωσης της μέτρησης οποιουδήποτε άλλου τμήματος.
- **'in_stub_όνομα'/'out_stub_όνομα'**: Το μήνυμα αυτό και το επόμενο έχουν ρόλο παρόμοιο με τα δύο προηγούμενα μηνύματα, μόνο που στην περίπτωση αυτή το σύστημα βιβλιοθήκης μπορεί να προσδιορίσει αυτό το 'όνομα'. Τα πεδία του μηνύματος είναι τα ίδια με τα μηνύματα για το CGI script και ο μηχανισμός για την αντιστοίχιση των μηνυμάτων είναι ο ίδιος μόνο που εφαρμόζεται για την διάκριση μεταξύ μηνυμάτων με το ίδιο 'όνομα'.

Πέρα από τα παραπάνω μηνύματα υπάρχουν, όπως προαναφέρθηκε, και μια σειρά από μηνύματα που σκοπό έχουν την πληρέστερη παρουσίαση και την ορθότητα των αρχείων καταγραφής των μετρήσεων. Τα μηνύματα αυτά είναι τα εξής:

- **'in_cgi_local'/'out_cgi_local'**: Το μήνυμα αυτό αποτελεί ουσιαστικά ειδική περίπτωση του μηνύματος για εκκίνηση/τερματισμό μέτρησης *CGI script*. Η μορφή του μηνύματος είναι η ίδια καθώς και ο τρόπος που γίνεται η συσχέτιση. Ο λόγος που εισήχθηκε το μήνυμα αυτό είναι διότι για την περίπτωση του συστήματος NCSTR-L-Dienst ήταν επιθυμητό να διακρίνουμε σε δύο κατηγορίες τις ενεργοποιήσεις του CGI script. Η διάκριση γίνεται μεταξύ ενεργοποιήσεων που προκλήθηκαν από κάποιο πελάτη του παγκόσμιου ιστού (τότε χρησιμοποιείται ο πρώτος τύπος μηνύματος) ή από κάποιο άλλο πρόγραμμα (π.χ. έναν από τους εξυπηρετητές του συστήματος βιβλιοθήκης, οπότε χρησιμοποιείται η ειδική περίπτωση που ορίζουμε εδώ). Για τον διαχωρισμό αυτό χρησιμοποιείται το πεδίο **'user agent'** του *http* μηνύματος. Για περισσότερες λεπτομέρειες για την χρήση των μηνυμάτων αυτών βλέπε στο Παράρτημα 5.
- **'OpenChildDebug'**: Το μήνυμα αυτό σηματοδοτεί την αναπαραγωγή του εξυπηρετητή (**fork**). Τα πεδία του μηνύματος πέρα από τον τύπο του είναι 'το αναγνωριστικό διεργασίας (**process id**)' του εξυπηρετητή που στέλνει το μήνυμα και 'όνομα αρχείου' του νέου αρχείου καταγραφής των μετρήσεων. Με την λήψη αυτού του μηνύματος ο επόπτης δημιουργεί ένα νέο αρχείο καταγραφής των μετρήσεων ενώ δημιουργεί έναν σύνδεσμο από το παλιό προς το νέο αρχείο καταγραφής. Από την στιγμή αυτή και πέρα η καταγραφή των μετρήσεων γίνεται στο νέο αρχείο μέχρι την λήψη ενός μηνύματος κλεισίματος αρχείου.
- **'CloseDebug'**: Το μήνυμα αυτό σηματοδοτεί το κλείσιμο ενός αρχείου καταγραφής των μετρήσεων. Τα πεδία του μηνύματος είναι τα ίδια με του παραπάνω. Τα πεδία του μηνύματος είναι 'το αναγνωριστικό διεργασίας (**process id**)' του εξυπηρετητή που στέλνει το μήνυμα και 'όνομα αρχείου' ενός αρχείου καταγραφής των μετρήσεων. Το μήνυμα αυτό οδηγεί στο κλείσιμο του συγκεκριμένου αρχείου καταγραφής μετρήσεων, αφού πρώτα γίνει έλεγχος για ανεκπλήρωτες μετρήσεις (δηλαδή την

ύπαρξη μηνυμάτων εκκίνησης μέτρησης χωρίς τα αντίστοιχα μηνύματα τερματισμού) και η καταγραφή τους στο αρχείο καταγραφής.

- **‘ExecDebug’**: Το μήνυμα αυτό σηματοδοτεί την έναρξη της εκτέλεσης ενός ανεξάρτητου προγράμματος από τον εξυπηρετητή. Τα πεδία του μηνύματος πέρα από τον τύπο του είναι ‘το αναγνωριστικό διεργασίας (**process id**)’ του εξυπηρετητή που στέλνει το μήνυμα και ‘όνομα αρχείου’ ενός αρχείου καταγραφής των μετρήσεων. Με την λήψη του μηνύματος αυτού απλώς το μήνυμα καταγράφεται στο συγκεκριμένο αρχείο καταγραφής των μετρήσεων.

Παράδειγμα:

```
“LogDebugEnter^\3909^\dienst.debug_3900_4^\T_is_dredd.csi.forth.gr.80
80^dredd.csi.forth.gr->bagheera.gmd.de.8090^\dredd4050
forth.stat.panoramix”.
```

Επικοινωνία με την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης.

Η επικοινωνία με την διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης γίνεται με δύο διαφορετικούς τρόπους, πρώτον μέσω ερωτήσεων (**queries**) για την βάση των παραμέτρων επίδοσης και δεύτερον με την αποστολή δεδομένων για αποθήκευση και την ανάκληση τους. Στην πρώτη περίπτωση η διαδικασία διαχείρισης αναπαράγει τον εαυτό της (**fork**) προκειμένου να απαντήσει την ερώτηση, σε αντίθεση με την δεύτερη περίπτωση που κάτι τέτοιο δεν είναι δυνατό γιατί υπάρχουν τιμές που πρέπει να αποθηκευτούν. Εξάλλου, ο δεύτερος τρόπος επικοινωνίας έχει σκοπό να επιτρέψει στο σύστημα βιβλιοθήκης να διατηρήσει κάποιες σημαντικές πληροφορίες επίδοσης μεταξύ των διαδοχικών ενεργοποιήσεων του (**forks**).

Οι ερωτήσεις για την βάση των παραμέτρων επίδοσης που υποστηρίζει ο επόπτης είναι οι εξής:

- **‘ALL’**: Η ερώτηση αυτή επιστρέφει ολόκληρη την βάση των παραμέτρων επίδοσης. Η μορφή της απάντησης για κάθε παράμετρο επίδοσης είναι η εξής: **‘Variable:όνομα|n’** το όνομα της παραμέτρου, για κάθε διακριτικό της παραμέτρου (π.χ. τις αρχές έκδοσης στο Παράρτημα 5) **‘Pub:όνομα|n’** το όνομα διακριτικού, **‘Values:λίστα των τιμών|n’**, οι τιμές για την συγκεκριμένη παράμετρο και το συγκεκριμένο διακριτικό που χωρίζονται μεταξύ τους με το ‘;’.
- **‘Variable:όνομα;Authority:όνομα’**: Με την ερώτηση αυτή παίρνονται οι τιμές για την συγκεκριμένη παράμετρο επίδοσης και το συγκεκριμένο διακριτικό. Η μορφή της απάντησης είναι **‘Values:λίστα των τιμών|n’**, όπου οι τιμές χωρίζονται μεταξύ τους με το ‘;’.
- **‘Variable:όνομα;Parameter:όνομα’**: Με την ερώτηση αυτή παίρνονται για μια συγκεκριμένη παράμετρο επίδοσης κάποιες ή όλες οι τιμές για όλα τα διακριτικά των παραμέτρων. Στην περίπτωση όλων των τιμών το όνομα τιμής γίνεται **‘All’**. Στην περίπτωση περισσότερων της μίας τιμών τα ονόματα των τιμών χωρίζονται μεταξύ τους με το **‘&’**. Η μορφή της απάντησης για κάθε διακριτικό είναι **‘όνομα αρχής έκδοσης:λίστα των τιμών|n’**, οι τιμές στην λίστα χωρίζονται μεταξύ τους με το ‘;’.
- **‘Type:όνομα;Parameter:όνομα’**: Με την ερώτηση αυτή παίρνονται κάποιες ή όλες οι τιμές για κάθε διακριτικό παραμέτρου μιας λίστας παραμέτρων επίδοσης, οι οποίες έχουν παρόμοιο όνομα (θεωρείται ότι παράμετροι επίδοσης με παρόμοιο όνομα χρησιμοποιούνται για τμήματα

εκτέλεσης με κάποια λογική συσχέτιση). Στην περίπτωση αυτή το πρώτο όνομα είναι κάποιο πρότυπο (**pattern**) το οποίο αναζητείται στα ονόματα των παραμέτρων επίδοσης. Ο προσδιορισμός των επιθυμητών τιμών στο δεύτερο όνομα γίνεται με τον τρόπο που περιγράφηκε παραπάνω. Η μορφή της απάντησης για κάθε αρχή έκδοσης κάθε παραμέτρου επίδοσης είναι '*Rsp:όνομα παραμέτρου_όνομα διακριτικού παραμέτρου:λίστα επιθυμητών τιμών|n*', το *_* υποδηλώνει την ύπαρξη ενός '*'*', και οι τιμές στην λίστα χωρίζονται με το '*;*'.

Η αποθήκευση και ανάκληση παραμέτρων γίνεται με τους εξής τρόπους:

- '*InfoIndex|n*': Το μήνυμα αυτό δίνει στο σύστημα βιβλιοθήκης έναν μετρητή. Η απάντηση στο μήνυμα αυτό είναι ένας αριθμός. Με την αποστολή της απάντησης ο μετρητής αυξάνεται.
- '*InfoDelay|n*': Το μήνυμα αυτό δίνει στο σύστημα βιβλιοθήκης την δυνατότητα να αποθηκεύει κάποια τιμή η οποία υπολογίζεται σε διάφορα στάδια με βάση την προηγούμενη τιμή της. Έτσι, με την λήψη του μηνύματος αυτού ο επόπτης στέλνει πίσω την τιμή αυτής της μεταβλητής και περιμένει να πάρει τη νέα της τιμή.

Παράδειγμα: Στέλνουμε το "Type:T_is;Parameter:W_AVG&DATE" και η απάντηση είναι οι γραμμές:

```
"Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>bagheera.gmd.de.8090 ncstrl.icase:875802960 402345;21498217

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>bagheera.gmd.de.8090 forth.stat.panoramix:875883232
910443;12408634.0646062

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>frabala.csi.forth.gr.8080 ncstrl.cudenver_ccm:875803152
151960;11935764.784385

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>cacofonix.csi.forth.gr.8080 ncstrl.cudenver_ccm:875803220
458294;24529841.25

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>goofy.csi.forth.gr.8080 ncstrl.ifi_de:875803152 794219;11830082

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>goofy.csi.forth.gr.8080 ncstrl.nyu_cs:875809209
182385;13052818.4399372

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>scrootz.csi.forth.gr.8080 ncstrl.nyu_cs:875803270 441277;15379609.75

Rsp:T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>zozefina.csi.forth.gr.8080 ncstrl.cornell.tc:875809209
550172;10742211.1095537".
```

Επικοινωνία με τον «έξω κόσμο».

Η επικοινωνία του επόπτη με τον «έξω κόσμο» γενικότερα γίνεται και αυτή με δύο τρόπους, όπως φαίνεται στο Σχήμα 5 και περιγράφεται τόσο στην παράγραφο 3.2.2. "Η αρχιτεκτονική του επόπτη επίδοσης." όσο και στην παράγραφο 4.1."Οπτικοποίηση των μετρήσεων.". Ο πρώτος τρόπος είναι με την σύνδεση στην δεύτερη διαδικασία μέτρησης και την λήψη των παραγόμενων τιμών για τις παραμέτρους επίδοσης, όπως ενημερώνεται και η διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης. Ο δεύτερος τρόπος είναι με την αποστολή ερωτήσεων και

την λήψη απαντήσεων στην διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης που περιγράφηκε παραπάνω για την επικοινωνία με το σύστημα βιβλιοθήκης.

Τα μηνύματα που στέλνει η δεύτερη διαδικασία μέτρησης στην διαδικασία διαχείρισης της βάσης των παραμέτρων επίδοσης είναι τα εξής:

- “**HUNGUP**”*n*: Το μήνυμα για τον τερματισμό της λειτουργίας του επόπτη επίδοσης. Στην περίπτωση αυτή ο παραλήπτης του μηνύματος οφείλει να κλείσει το socket που χρησιμοποιεί για την επικοινωνία με την δεύτερη διαδικασία μέτρησης και να διακόψει την λειτουργία του.
- “**Reset**”*n*: Το μήνυμα για την επανεκκίνηση των μετρήσεων από τον επόπτη επίδοσης. Στην περίπτωση αυτή ο παραλήπτης του μηνύματος οφείλει να καθαρίσει το ιστορικό των μετρήσεων, εάν διατηρεί ιστορικό, ή μπορεί να αγνοήσει το συγκεκριμένο μήνυμα.
- “**Νέα Μέτρηση**”*n*: Το μήνυμα για την παραγωγή μίας νέας μέτρησης από τον επόπτη επίδοσης. Το μήνυμα περιλαμβάνει τα εξής: όνομα παραμέτρου επίδοσης, όνομα διακριτικού παραμέτρου (προαιρετικό, βλέπε και προηγούμενα), οι διάφορες τιμές για την παράμετρο επίδοσης. Το όνομα της παραμέτρου επίδοσης ακολουθείται πάντα από το ‘?’ , για να διαχωρίζεται από το διακριτικό της. Τα δύο αυτά ουσιαστικά προσδιορίζουν επακριβώς την παράμετρο επίδοσης (βλέπε στο Παράρτημα 5). Τα προσδιοριστικά της παραμέτρου επίδοσης μπορούν να είναι ένας οποιοσδήποτε ορθογραφικός χαρακτήρων (**string**) που δεν περιέχει τους χαρακτήρες διαχωριστικά και χωρίζονται από τις τιμές της με το ‘:’. Οι διάφορες τιμές της παραμέτρου επίδοσης διαχωρίζονται από το ‘;’. Οι τιμές είναι οι εξής: ο αύξων αριθμός μέτρησης, η νέα μέτρηση, το τρέχον ελάχιστο των τιμών, το τρέχον μέγιστο, το τρέχον άθροισμα, η τρέχουσα μέση τιμή (δηλαδή το τρέχον άθροισμα προς τον αύξων αριθμό μέτρησης), η τρέχουσα τυπική απόκλιση (δηλαδή $\sqrt{\frac{1}{n-1} \left(\sum x^2 - \frac{1}{n} (\sum x)^2 \right)}$,

όπου $\sum x^2$ το τρέχον άθροισμα των τετραγώνων των μετρήσεων, $\sum x$ το τρέχον άθροισμα των μετρήσεων και *n* ο τρέχων αριθμός μέτρησης), ο τρέχων σταθμισμένος μέσος όρος (δηλαδή ο προηγούμενος σταθμισμένος μέσος όρος συν η νέα μέτρηση δια δύο) και η χρονοσφραγίδα της μέτρησης (δηλαδή η χρονοσφραγίδα του μηνύματος τερματισμού της μέτρησης). Οι τιμές νέα μέτρηση, τρέχον ελάχιστο των τιμών, τρέχον μέγιστο, τρέχον άθροισμα και η χρονοσφραγίδα της μέτρησης αποτελούνται από δύο πεδία που χωρίζονται με κενό (δευτερόλεπτα, μικροδευτερόλεπτα). Οι υπόλοιπες τιμές αποτελούνται από έναν δεκαδικό αριθμό. Όλα τα πεδία του μηνύματος πρέπει να εμφανίζονται με την σειρά που αναφέρονται παραπάνω. Στην περίπτωση αυτή ο παραλήπτης μπορεί εάν τον ενδιαφέρει η παράμετρος επίδοσης στην οποία αναφέρεται το μήνυμα να το πάρει και να το επεξεργαστεί, διαφορετικά να το αγνοήσει.

Παράδειγμα: “T_DS1_1?SimpleVar:206;17 497;11 618320;105 337849;6742 980849;32732916.7427184;20573447.9532453;15190704.753962;875886685 533126”.

Παράρτημα 2. Αρχεία Καταγραφής Μετρήσεων.

Το κεντρικό αρχείο καταγραφής των μετρήσεων ακολουθεί:

Debug File

- [Fri Oct 3 11:17:58 1997] [pid=2235] **BEGIN variable=(enter_cgi_total)**
(args=dredd2235) (875866678 741635)
 - [Fri Oct 3 11:17:58 1997] [pid=2822] **NEW CHILD [fork child 1](#)**
- [Fri Oct 3 11:18:25 1997] [pid=2235] **END variables=(enter_cgi_total)**
(duration=27.109050) (875866705, 850685)
- [Fri Oct 3 11:18:43 1997] [pid=2842] **BEGIN variable=(enter_cgi_total)**
(args=dredd2842) (875866723 137249)
 - [Fri Oct 3 11:18:43 1997] [pid=2822] **NEW CHILD [fork child 2](#)**
- [Fri Oct 3 11:19:33 1997] [pid=2842] **END variables=(enter_cgi_total)**
(duration=49.944789) (875866773, 82038)
- [Fri Oct 3 11:19:50 1997] [pid=2340] **BEGIN variable=(enter_cgi_total)**
(args=dredd2340) (875866790 613320)
 - [Fri Oct 3 11:19:50 1997] [pid=2822] **NEW CHILD [fork child 3](#)**
- [Fri Oct 3 11:20:17 1997] [pid=2340] **END variables=(enter_cgi_total)**
(duration=26.542917) (875866817, 156237)
- [Fri Oct 3 11:20:34 1997] [pid=2828] **BEGIN variable=(enter_cgi_total)**
(args=dredd2828) (875866834 429925)
 - [Fri Oct 3 11:20:34 1997] [pid=2822] **NEW CHILD [fork child 4](#)**
- [Fri Oct 3 11:20:56 1997] [pid=2828] **END variables=(enter_cgi_total)**
(duration=22.526200) (875866856, 956125)
- [Fri Oct 3 11:21:14 1997] [pid=2905] **BEGIN variable=(enter_cgi_total)**
(args=dredd2905) (875866874 190403)
 - [Fri Oct 3 11:21:14 1997] [pid=2822] **NEW CHILD [fork child 5](#)**
- [Fri Oct 3 11:21:44 1997] [pid=2905] **END variables=(enter_cgi_total)**
(duration=30.577886) (875866904, 768289)
- [Fri Oct 3 11:22:01 1997] [pid=2892] **BEGIN variable=(enter_cgi_total)**
(args=dredd2892) (875866921 941367)
 - [Fri Oct 3 11:22:01 1997] [pid=2822] **NEW CHILD [fork child 6](#)**
- [Fri Oct 3 11:22:19 1997] [pid=2892] **END variables=(enter_cgi_total)**
(duration=17.940001) (875866939, 881368)
- [Fri Oct 3 11:22:37 1997] [pid=1815] **BEGIN variable=(enter_cgi_total)**
(args=dredd1815) (875866957 476304)
 - [Fri Oct 3 11:22:37 1997] [pid=2822] **NEW CHILD [fork child 7](#)**
- [Fri Oct 3 11:22:55 1997] [pid=1815] **END variables=(enter_cgi_total)**
(duration=18.79603) (875866975, 555907)
- [Fri Oct 3 11:23:13 1997] [pid=2850] **BEGIN variable=(enter_cgi_total)**
(args=dredd2850) (875866993 55367)
 - [Fri Oct 3 11:23:13 1997] [pid=2822] **NEW CHILD [fork child 8](#)**

- [Fri Oct 3 11:23:45 1997] [pid=2850] **END variables=(enter_cgi_total)** (duration=32.195480) (875867025, 250847)
- [Fri Oct 3 11:24:02 1997] [pid=2841] **BEGIN variable=(enter_cgi_total)** (args=dredd2841) (875867042 463458)
 - [Fri Oct 3 11:24:02 1997] [pid=2822] **NEW CHILD [fork child 9](#)**
- [Fri Oct 3 11:24:18 1997] [pid=2841] **END variables=(enter_cgi_total)** (duration=15.799720) (875867058, 263178)
- [Fri Oct 3 11:24:35 1997] [pid=2998] **BEGIN variable=(enter_cgi_total)** (args=dredd2998) (875867075 474488)
 - [Fri Oct 3 11:24:35 1997] [pid=2822] **NEW CHILD [fork child 10](#)**
- [Fri Oct 3 11:25:00 1997] [pid=2998] **END variables=(enter_cgi_total)** (duration=25.82756) (875867100, 557244)
- [Fri Oct 3 11:25:18 1997] [pid=2962] **BEGIN variable=(enter_cgi_total)** (args=dredd2962) (875867118 147656)
 - [Fri Oct 3 11:25:18 1997] [pid=2822] **NEW CHILD [fork child 11](#)**
- [Fri Oct 3 11:25:50 1997] [pid=2962] **END variables=(enter_cgi_total)** (duration=31.948840) (875867150, 96496)
- [Fri Oct 3 11:26:07 1997] [pid=1987] **BEGIN variable=(enter_cgi_total)** (args=dredd1987) (875867167 298149)
 - [Fri Oct 3 11:26:07 1997] [pid=2822] **NEW CHILD [fork child 12](#)**
- [Fri Oct 3 11:26:20 1997] [pid=1987] **END variables=(enter_cgi_total)** (duration=13.668654) (875867180, 966803)
- [Fri Oct 3 11:26:38 1997] [pid=2990] **BEGIN variable=(enter_cgi_total)** (args=dredd2990) (875867198 421743)
 - [Fri Oct 3 11:26:38 1997] [pid=2822] **NEW CHILD [fork child 13](#)**
- [Fri Oct 3 11:26:56 1997] [pid=2990] **END variables=(enter_cgi_total)** (duration=18.387403) (875867216, 809146)
- [Fri Oct 3 11:27:14 1997] [pid=1937] **BEGIN variable=(enter_cgi_total)** (args=dredd1937) (875867234 3617)
 - [Fri Oct 3 11:27:14 1997] [pid=2822] **NEW CHILD [fork child 14](#)**
- [Fri Oct 3 11:27:28 1997] [pid=1937] **END variables=(enter_cgi_total)** (duration=14.455893) (875867248, 459510)
- [Fri Oct 3 11:27:45 1997] [pid=2641] **BEGIN variable=(enter_cgi_total)** (args=dredd2641) (875867265 825633)
 - [Fri Oct 3 11:27:45 1997] [pid=2822] **NEW CHILD [fork child 15](#)**
- [Fri Oct 3 11:28:04 1997] [pid=2641] **END variables=(enter_cgi_total)** (duration=18.193543) (875867284, 19176)

Ένα από τα δευτερεύοντα αρχεία καταγραφής των μετρήσεων, ακολουθεί:

Debug File

- [Fri Oct 3 11:17:59 1997] [pid=2186] **BEGIN variable=(T_DS1_1)** (args=dredd2235) (875866679 918409)
 - [Fri Oct 3 11:18:00 1997] [pid=2186] **BEGIN variable=(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-bagheera.gmd.de.8090)** (args=dredd2235) (875866680 265388)
 - [Fri Oct 3 11:18:00 1997] [pid=2186] **EXEC CHILD 1**

- [Fri Oct 3 11:18:24 1997] [pid=2186] **END**
variables=(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
bagheera.gmd.de.8090) (duration=23.932431) (875866704, 197819)
- [Fri Oct 3 11:18:25 1997] [pid=2186] **END variables=(T_DS1_1)**
(duration=25.906911) (875866705, 825320)

Παράδειγμα αρχείου που αποθηκεύεται η βάση των παραμέτρων επίδοσης ακολουθεί:

```
(T_DS1^req,SimpleVar) 9 360958 262426 704513 3
914395 434932.777777778 1872039328793 398166.875
875871345 267403

(T_DS1_1,SimpleVar) 200 22688332 11618320 105337849
6657 745748 33288728.74 3.06257567254956e+17
21434216.2535677 875870650 988821

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>bagheera.gmd.de.8090,forth.stat.panoramix) 96 19901091
7521825 61043437 1557 455787 16223497.78125
3.25449995974797e+16 17456248.5168499 875870651 218078

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>bagheera.gmd.de.8090,ncstrl.icase) 1 21498217 21498217
21498217 21 498217 21498217 462173334179089
21498217 875802960 402345

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>cacofonix.csi.forth.gr.8080,ncstrl.cudenver_ccm) 3 19209366
19209366 38288524 78 909999 26303333
2.29348922404842e+15 24529841.25 875803220 458294

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>frabala.csi.forth.gr.8080,ncstrl.cudenver_ccm) 97 11284668
10801096 35284630 1245 563760 12840863.5051546
1.7395326640793e+16 11935764.784385 875803152 151960

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>goofy.csi.forth.gr.8080,ncstrl.ifi_de) 1 11830082 11830082
11830082 11 830082 11830082 139950840126724
11830082 875803152 794219

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>goofy.csi.forth.gr.8080,ncstrl.nyu_cs) 96 11056304 10049427
20947393 1138 360767 11857924.65625
1.37788311977368e+16 13052818.4399372 875809209 182385

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>scrootz.csi.forth.gr.8080,ncstrl.nyu_cs) 3 18450357 11507053
18450357 43 68082 14356027.3333333 644717662463842
15379609.75 875803270 441277

(T_is_dredd.csi.forth.gr.8080^dredd.csi.forth.gr-
>zozefina.csi.forth.gr.8080,ncstrl.cornell.tc) 100 10487139
5687115 29951178 1066 155996 10661559.96
1.19570788521481e+16 10742211.1095537 875809209 550172

(enter_cgi_total,SimpleVar) 209 1822377 1617225 106694394
6979 27323 33392475.2296651 3.26871533061276e+17
12228816.2005385 875871343 963050
```


Παράρτημα 3. Παράδειγμα Χρήσης του Συστήματος NCSTRL-Dienst.

Στο παράρτημα αυτό παρουσιάζεται ένα παράδειγμα χρήσης του συστήματος NCSTRL-Dienst. Ξεκινάμε με την αρχική σελίδα του συστήματος (βλέπε στο Σχήμα 19).




INSTITUTE OF COMPUTER SCIENCE
TECHNICAL REPORT LIBRARY

Welcome to the on-line technical report library at the Institute of Computer Science – FORTH. This is an entry point to a distributed technical report library developed by the ARPA-sponsored Computer Science Technical Report Project. This library contains technical reports from a number of [participating institutions](#). You may search for and access documents from the distributed collection.




This is Dienst V4.1.8 with greek search support.

 Search the NCSTRL-ERCIM collection

 The [Fielded Search Form](#) allows you to perform a search on several fields of the bibliographic data, and/or to limit the search to specific institutions,

Or simply enter [one or several words](#) into the box below to search author, title, and abstract, and press [Enter]:



-  Or browse reports at any of the [participating institutions](#).
-  Or See the [Statistics of the users' queries](#).
-  Or See the [Summary Stats of Some Dienst Servers Usage](#)



FORTH
FOundation for Research
& Technology – Hellas



ERCIM
European Research Consortium
for Informatics and
Mathematics



NCSTRL
National Computer
Science Technical Report
Library

Σχήμα 19: Αρχική Σελίδα του Συστήματος NCSTRL-Dienst.

Fielded Search of the Collection

Bibliographic keywords: (◆ AND keyword fields ◆ OR keyword fields)

Author:

Title

Abstract

Select one or more organizations from this list:

Foundation for Research and Technology – Hellas, Institute of Computer Science
Albert-Ludwigs-Universität Freiburg, Institut fuer Informatik
Auburn University
Boston University
Brown University – Department of Computer Science
CNR, Pisa, Italy

or search all organizations

Document Identifier:

Help: [Information about fielded searching](#).

[[Search fields](#) | [Search simple](#)]



ERCIM

European Research Consortium
for Informatics and Mathematics

NCSTRL-ERCIM

This server operates at Institute of Computer Science Technical Report Library.

Send email to asiditrop@csi.forth.gr

Σχήμα 20: Αναζήτηση με Βάση τα Πεδία της Βιβλιογραφικής Εγγραφής.

Στην σελίδα αυτή μπορούμε είτε να δώσουμε κάποια λέξη κλειδί στο κατάλληλο πεδίο για την οποία θα ψάξει το σύστημα είτε μπορούμε να επιλέξουμε την σελίδα για το ψάξιμο με λέξεις κλειδιά σε συγκεκριμένα πεδία της βιβλιογραφικής εγγραφής των εγγράφων (βλέπε στο Σχήμα 20). Στην δεύτερη περίπτωση, έχουμε την δυνατότητα να περιορίσουμε την αναζήτηση σε ορισμένες μόνο από τις αρχές έκδοσης του συστήματος.

[[Search fields](#) | [Search simple](#)]

Search Results

Search fields:

organization = *Foundation for Research and Technology – Hellas. Institute of Computer Science*
author = *sarant*

Search Summary:

Organizations you selected are listed below by number of titles found.

■ (3) [Foundation for Research and Technology – Hellas. Institute of Computer Science](#)

Search Results:

Foundation for Research and Technology – Hellas. Institute of Computer Science

▣ [Προς Μία Πανερωπαϊκή Επιστημονική Ψηφιακή Βιβλιοθήκη](#), Χρήστος Νικολάου, Σαράντος Κοπιδάκης and Γιώργος Γεωργιαννάκης. (TR96–0167)

▣ [Distributed, Low Contention Task Allocation](#), Sarantos Kapidakis and Marios Mavronicolas. (TR95–0133)

▣ [Description of the Adaptive Resource Management Problem, cost functions](#), George Georgiannakis, Catherine Houstis, Sarantos Kapidakis, Maria Karavassili, Christos Nikolaou, Alexandros Labrinidis, Manolis Marazakis, Evangelos Markatos, Marios Mavronicolas, S. Chabridon, E. Gelenbe, E. Born, L. Richter and R. Riedl. (TR95–0130)

[[Search fields](#) | [Search simple](#)]



NCSTRL–ERCIM

This server operates at Institute of Computer Science Technical Report Library.
Send email to asidirop@csi.forth.gr

Σχήμα 21: Σελίδα Αποτελεσμάτων Αναζήτησης.

Τα αποτελέσματα της αναζήτησης παρουσιάζονται σαν μία λίστα επιτυχιών για κάθε αρχή έκδοσης με τον τίτλο κάθε εγγράφου και σύνδεσμο προς την βιβλιογραφική εγγραφή του (βλέπε στο Σχήμα 21). Τέλος, από την σελίδα με την βιβλιογραφική εγγραφή κάθε εγγράφου έχουμε συνδέσμους προς τις διαθέσιμες μορφές και αποσυνθέσεις (βλέπε στο Σχήμα 22).

Description of the Adaptive Resource Management Problem, cost functions

George Georgiannakis, Catherine Houstis, Sarantos Kapidakis, Maria Karavassili, Christos Nikolaou, Alexandros Labrinidis, Manolis Marazakis, Evangelos Markatos, Marios Mavronicolas, S. Chabridon, E. Gelenbe, E. Born, L. Richter and R. Riedl
TR95–0130
April 1995

Copyright ICS–FORTH

Deliverable

How to view this document

- Display the **whole** document in one of the following formats.
 - [PostScript](#) 1027479 bytes.
- [Print or download all or selected pages.](#)

You are granted permission for the non–commercial reproduction, distribution, display, and performance of this technical report in any format, BUT this permission is only for a period of 45 (forty–five) days from the most recent time that you verified that this technical report is still available from the Institute of Computer Science of the Foundation for Research and Technology – Hellas (FORTH) under terms that include this permission. All other rights are reserved by the author(s).

[[Search fields](#) | [Search simple](#)]



NCSTRL–ERCIM

This server operates at Institute of Computer Science Technical Report Library.
Send email to asidirop@csi.forth.gr

Σχήμα 22: Σελίδα Βιβλιογραφικής Εγγραφής Εγγράφου.

Παράρτημα 4. Το Πρωτόκολλο *Dienst*.

Η επικοινωνία με τους εξυπηρετητές *Dienst* γίνεται με την χρήση του πρωτοκόλλου *Dienst*. Μια που η επικοινωνία αυτή γίνεται μέσω κάποιου stub από πελάτες του παγκόσμιου ιστού, το Πρωτόκολλο *Dienst* χρησιμοποιεί το Πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) σαν το επίπεδο μεταφοράς (**transport layer**) των αιτήσεων. Μια *Dienst* αίτηση καταλαμβάνει το πεδίο *path* του ομοιόμορφου προσδιοριστή πόρων (**Uniform Resource Locator – URL**) που χρησιμοποιεί ο πελάτης του παγκόσμιου ιστού για την κατασκευή της HTTP αίτησης για τον εξυπηρετητή του παγκόσμιου ιστού. Επομένως, οι αιτήσεις του πρωτοκόλλου *Dienst* είναι σύμφωνες με τους κανόνες του πεδίου *path* των URL. Για παράδειγμα, όλοι οι χαρακτήρες κενού γράφονται %20. Άλλοι ειδικοί χαρακτήρες είναι:

- / - διαχωρίζει τα πεδία του URL.
- ? - διαχωρίζει τα προαιρετικά ορίσματα από το υπόλοιπο URL.
- # - δείχνει την αναφορά σε ένα σημείο αναφοράς (**named anchor**) μέσα σε ένα κείμενο.
- = - διαχωρίζει το όνομα από την τιμή σε μια λίστα ορισμάτων.
- & - διαχωρίζει τα ορίσματα μετά το ?.

Εάν κάποιος από αυτούς τους χαρακτήρες χρησιμοποιείται κατά τρόπο διαφορετικό από τον παραπάνω, τότε θα πρέπει να αντικατασταθεί από ένα % και την δεκαεξαδική τιμή του.

Όλες οι απαντήσεις ενός εξυπηρετητή *Dienst* έχουν την μορφή απαντήσεων του πρωτοκόλλου μεταφοράς υπερκειμένου και περιλαμβάνουν πεδία όπως η έκδοση του πρωτοκόλλου (**http version**), κατάσταση (**status**) και επικεφαλίδα περιεχομένου (**content header**) τύπου MIME.

Η μορφή των URL με τα οποία μεταφέρονται τα μηνύματα του πρωτοκόλλου *Dienst* είναι η εξής:

Dienst

Η λέξη αυτή πρέπει να εμφανίζεται πάντα στο URL.

Όνομα Υπηρεσίας

Το όνομα της υπηρεσίας που θα εξυπηρετήσει το μήνυμα. Οι διαθέσιμες υπηρεσίες είναι *Repository* (αποθήκης), *Index* (δεικτοδότησης), *UI* (διεπιφάνειας χρήσης), *Meta* (meta εξυπηρετητή). Μια άλλη ειδική υπηρεσία είναι η *Info* (πληροφορίες) που επιστρέφει διάφορα στοιχεία για τον εξυπηρετητή.

Έκδοση

Η έκδοση είναι ουσιαστικά δύο αριθμοί που χωρίζονται με μία τελεία. Η έκδοση είναι η έκδοση του μηνύματος και όχι του συστήματος *Dienst*. Σκοπός της έκδοσης είναι να διατηρεί την προς τα πίσω συμβατότητα.

Όνομα Αίτησης

Είναι το όνομα της αίτησης. Κάθε όνομα αίτησης είναι μοναδικό για κάθε υπηρεσία (βλέπε τον πίνακα παρακάτω).

Υποχρεωτικά Ορίσματα

Κάθε όνομα αίτησης συνοδεύεται από ένα αριθμό υποχρεωτικών ορισμάτων. Τα ορίσματα αυτά όχι μόνο πρέπει να παρέχονται πάντα αλλά να δίνονται και με προκαθορισμένη σειρά.

Προαιρετικά Ορίσματα

Υπάρχουν δύο είδη σύνταξης για τα προαιρετικά ορίσματα στο Πρωτόκολλο: λέξης κλειδί (**keywords**) και θέσης (**positional**).

- Στην περίπτωση της σύνταξης λέξης κλειδί τα ορίσματα έχουν την μορφή κλειδί = τιμή. Εάν υπάρχουν περισσότερα από ένα ορίσματα χωρίζονται μεταξύ τους με &. Η σειρά των ορισμάτων δεν παίζει ρόλο.
- Στην περίπτωση της σύνταξης θέσης τα ορίσματα εμφανίζονται με συγκεκριμένη σειρά, δεν έχουν ιδιαίτερο όνομα και όλα τα ορίσματα που χρειάζονται πρέπει να είναι παρόντα.

Το διαχωριστικό μεταξύ των τμημάτων του URL είναι το '/', μόνο τα προαιρετικά ορίσματα χωρίζονται με '?'.

<i>Repository</i>	<i>List-Contents</i>
	<i>Body</i>
	<i>Page</i>
	<i>Npages</i>
	<i>Formats</i>
<i>Index</i>	<i>List-Contents</i>
	<i>Bibliography</i>
	<i>SearchBoolean</i>
<i>UI</i>	<i>Search</i>
	<i>QueryNF</i>
	<i>Describe</i>
	<i>BrowseYears</i>
	<i>ListYears</i>
	<i>BrowseAuthors</i>
	<i>ListAuthors</i>
<i>Meta</i>	<i>Publishers</i>
	<i>Repositories</i>
	<i>Indices</i>
	<i>Lite</i>
<i>Info</i>	<i>Version</i>
	<i>Log</i>
	<i>List-Services</i>
	<i>List-Verbs</i>
	<i>Describe-Verb</i>

(25) Υπηρεσίες
και Ονόματα Αιτήσεων
τους.

Περισσότερες λεπτομέρειες για το Πρωτόκολλο καθώς και περιγραφή των λειτουργιών κάθε μιας από τις αιτήσεις στα [4] και [5].

Παράρτημα 5. Οι Παράμετροι Επίδοσης του Συστήματος *NCSTRL-Dienst*.

Οι παράμετροι επίδοσης του συστήματος *NCSTRL-Dienst* ορίστηκαν με βάση την ανάλυση επίδοσης που παρουσιάζεται στο Κεφάλαιο 6. Συγκεκριμένα, στο «Σχήμα 11: Ροή Εξυπηρέτησης μιας Ερώτησης στο Σύστημα *Dienst*» με διακεκομμένους κύκλους παριστάνουμε τις διεργασίες του συστήματος στις οποίες αντιστοιχούν παράμετροι επίδοσης, ομοίως και οι μαύρες τελείες ορίζουν ένα τμήμα της λειτουργίας του συστήματος για το οποίο επίσης αντιστοιχεί μία παράμετρος επίδοσης. Σε σχέση με τα παραπάνω στην μελέτη επίδοσης του συστήματος έγιναν κάποιες αλλαγές. Πρώτον, η μελέτη επίδοσης επικεντρώθηκε στις αιτήσεις αναζήτησης, ήταν, επομένως, αναγκαίο να διαχωρίσουμε τις αιτήσεις αναζήτησης από τις υπόλοιπες αιτήσεις που εξυπηρετεί ένας εξυπηρετητής. Για τον σκοπό αυτό εισήχθη η παράμετρος επίδοσης T_{DS1}^{req} . Δεύτερον, διαπιστώθηκε ότι σε σχέση με τον πρώτο *forked* εξυπηρετητή μας ενδιαφέρει ο συνολικός χρόνος κατά τον οποίο παραμένει ενεργοποιημένος. Έτσι, δεν γίνεται πλέον διάκριση μεταξύ του σημείου που αρχίζει την αποστολή των αποτελεσμάτων στο *nph-dienst-stub* και την στιγμή που η αποστολή ολοκληρώνεται, επομένως, έχουμε μία μόνο παράμετρο επίδοσης για την διεργασία *nph-dienst-stub*, την T_{DS1} . Τρίτον, από τη στιγμή που το σύστημα *NCSTRL-Dienst* χρησιμοποιεί την απευθείας απάντηση στην επικοινωνία μεταξύ των εξυπηρετητών, η μέτρηση της διεργασίας *indexer_stub* δεν έχει πλέον νόημα, αφού ο ρόλος του *indexer_stub* περιορίζεται στην αποστολή του μηνύματος στον μακρινό εξυπηρετητή. Τέλος, για τις παραμέτρους επίδοσης $T_{isX}^{X \rightarrow Y}$ και T_{DS2} κρίθηκε αναγκαίο να γίνει διαχωρισμός των μετρήσεών τους με βάση τις αρχές έκδοσης (**authorities**) στις οποίες αναφέρονται. Οι παράμετροι επίδοσης αυτές χρησιμοποιήθηκαν για την λήψη των αποφάσεων δρομολόγησης στις τεχνικές εξισορρόπησης φόρτου (βλέπε στην παράγραφο 7.4). Ο διαχωρισμός αυτός είχε σκοπό την βελτίωση της ακρίβειας των εκτιμήσεων που παράγει ο επόπτης επίδοσης και επομένως, την βελτίωση της ακρίβειας των αποφάσεων δρομολόγησης.

Οι παράμετροι επίδοσης αναλυτικά είναι οι εξής:

- T_{DS1}^{req} : Ο χρόνος που χρειάζεται η *forked Dienst* διαδικασία για την εξυπηρέτηση μίας οποιασδήποτε αίτησης (**request**) εκτός των αιτήσεων αναζήτησης (**query**), ουσιαστικά ο χρόνος που η *forked* διαδικασία παραμένει ενεργή.
- T_{nds}^{total} : Ο χρόνος που χρειάζεται το *nph-dienst-stub* για την αποστολή μιας οποιασδήποτε αίτησης, εκτός των αιτήσεων αναζήτησης στην τοπική βάση δεικτοδότησης, στον εξυπηρετητή και για τη λήψη της απάντησης, ουσιαστικά ο χρόνος που το *nph-dienst-stub* παραμένει ενεργό.

- T_{nds}^{local} : Ο χρόνος που χρειάζεται το *nph-dienst-stub* για την αποστολή μίας αίτησης αναζήτησης στην τοπική βάση δεικτοδότησης στον εξυπηρετητή και για την λήψη της απάντησης.
- $T_{isX}^{X \rightarrow Y}$: Ο χρόνος που χρειάζεται από την εκκίνηση στον εξυπηρετητή X του *indexer_stub* για την προώθηση μίας αίτησης αναζήτησης στον εξυπηρετητή Y και την λήψη της απάντησης, ουσιαστικά ο χρόνος απόκρισης του εξυπηρετητή Y για αιτήσεις αναζήτησης στην τοπική του βάση δεικτοδότησης.
- T_{DS1_1} : Ο χρόνος που χρειάζεται το $DS1_0$ για να ενεργοποιήσει τον forked εξυπηρετητή Dienst, να του μεταφέρει την αίτηση αναζήτησης, να υπολογίσει αυτός την απάντηση, και να στείλει τα αποτελέσματα στο *nph-dienst-stub*.
- T_{DS1_2} : Ο χρόνος που χρειάζεται ο δεύτερος forked εξυπηρετητής Dienst για την επεξεργασία μίας ερώτησης για την τοπική βάση δεικτοδότησης.

Παράρτημα 6. Προσθήκες και Αλλαγές στον Κώδικα του Συστήματος NCSTR-L-Dienst.

Στο παράρτημα αυτό παρουσιάζουμε τις αλλαγές και τις προσθήκες στον κώδικα του συστήματος NCSTR-L-Dienst που ήταν απαραίτητες για την σύνδεση του με τον επόπτη επίδοσης. Οι προσθήκες και οι αλλαγές χωρίζονται σε μερικές γενικότερες κατηγορίες, πρώτον προσθήκες που αφορούν κώδικα για την αποστολή των κατάλληλων μηνυμάτων στον επόπτη επίδοσης, δεύτερον προσθήκες για την επέκταση του πρωτοκόλλου και την υποστήριξη της νέας κατηγορίας υπηρεσιών, τρίτον αλλαγές και προσθήκες για την υλοποίηση μηχανισμών εξισορρόπησης φόρτου στο σύστημα.

Έτσι, για την αποστολή των μηνυμάτων προστέθηκε στο αρχείο *Kernel/utilities.pl* η ρουτίνα **OpenDebugSocket**, η οποία καλείται κατά την εκκίνηση του εξυπηρετητή Dienst και σκοπός της είναι να ξεκινήσει τον επόπτη επίδοσης (τα προγράμματα *main.pl* και *socket_handler.pl*) και να δημιουργήσει τα βασικά κανάλια επικοινωνίας (sockets) με αυτόν. Σε αυτό το σημείο προσδιορίζονται και τα ορίσματα των προγραμμάτων του επόπτη. Παρόμοια με την παραπάνω ρουτίνα είναι και η **Debug_Socket_Open** που ορίζεται και καλείται στο *nph-dienst_stub.pl* και σκοπός της είναι η δημιουργία του καναλιού επικοινωνίας του CGI stub με τον επόπτη. Για την αποστολή των μηνυμάτων ορίζεται ουσιαστικά μία ρουτίνα για κάθε τύπο μηνύματος. Έχουμε, λοιπόν, τις **OpenDebug**, **ExecDebug**, **OpenChildDebug** και **CloseDebug** για τους αντίστοιχους τύπους μηνυμάτων και τις **Log_debug1** και **Log_debug2** για τα μηνύματα εκκίνησης και τερματισμού μέτρησης τμήματος. Όλες οι ρουτίνες αποστολής μηνύματος καταλήγουν σε μία κλήση της ρουτίνας **SendDebug_Message**, η οποία κάνει την τελική μορφοποίηση του μηνύματος, και βρίσκονται στο *Kernel/utilities.pl*. Τα σημεία στα οποία καλούνται οι ρουτίνες αποστολής μηνύματος είναι τα εξής:

1. Η **OpenDebug** καλείται στο τέλος της ρουτίνας **LoadCode** στο *Kernel/dienst.pl*, η οποία καλείται κατά την εκκίνηση του εξυπηρετητή Dienst.
2. Η **ExecDebug** καλείται στο σημείο που ξεκινάει τα *indexer_stubs* στην ρουτίνα **Parallel_Search** στο *UI_Server/remote.pl*.
3. Η **OpenChildDebug** καλείται στο σημείο που ο εξυπηρετητής Dienst αναπαράγει τον εαυτό του για την εξυπηρέτηση μίας αίτησης στο *Kernel/dienst.pl*.
4. Η **CloseDebug** καλείται σε δύο σημεία, πρώτον κατά την ολοκλήρωση της εξυπηρέτησης μιας αίτησης για το κλείσιμο του αρχείου καταγραφής που σχετίζεται με αυτήν και δεύτερον κατά τον ομαλό τερματισμό της λειτουργίας του εξυπηρετητή για το κλείσιμο του βασικού αρχείου καταγραφής (αυτό που άνοιξε με την **OpenDebug**). Οι δύο κλήσεις βρίσκονται στο *Kernel/dienst.pl*.
5. Οι **Log_debug1** και **Log_debug2** καλούνται κατά ζεύγη σε δύο διαφορετικά σημεία, πρώτον κατά την εκκίνηση και τον τερματισμό της εξυπηρέτησης μιας αίτησης (πριν και μετά από την ρουτίνα **do_Dispatch_URL**), όπου ανάλογα με το **Context{‘agent’}** πρόκειται για εξυπηρέτηση αίτησης αναζήτησης στην τοπική βάση ή όχι και ανάλογα **Context{‘url’}** πρόκειται για αίτηση αναζήτησης ή άλλη αίτηση οπότε και η κλήση αναφέρεται στην αντίστοιχη κατά περίπτωση παράμετρο επίδοσης (για τις παραμέτρους επίδοσης βλέπε στο Παράρτημα 5). Η

δεύτερη κλήση του ζεύγους ξεκινά στο σημείο εκκίνησης του *indexer_stub* για κάθε εξυπηρετητή δεικτοδότησης και τελειώνει στο σημείο που λαμβάνεται η απάντηση από τον εξυπηρετητή δεικτοδότησης στο *UI_Server/remote.pl*. Στην περίπτωση αυτή η αντίστοιχη για κάθε όνομα εξυπηρετητή παράμετρος ορίζεται (για περισσότερες λεπτομέρειες βλέπε στο Παράρτημα 5) και επίσης η λίστα με τις αρχές έκδοσης στις οποίες αναφέρεται η αίτηση αναζήτησης κατασκευάζεται και περνιέται ως όρισμα (για την αναλυτική περιγραφή της μορφής των μηνυμάτων βλέπε στο Παράρτημα 1).

Τέλος, τα μηνύματα εκκίνησης και τερματισμού του *CGI script* στέλνονται το μήνυμα εκκίνησης στο σημείο που έχουν διαβαστεί οι παράμετροι περιβάλλοντος στην ρουτίνα *send_DIENST_request*, γιατί ανάλογα με το *HTTP_USER_AGENT* καθορίζεται η παράμετρος επίδοσης και ανάλογα με το είδος της αίτησης καθορίζονται τα ορίσματα του μηνύματος. Το μήνυμα τερματισμού στέλνεται με την ολοκλήρωση της παραπάνω ρουτίνας, οπότε κλείνεται και το socket.

Για την εισαγωγή της νέας υπηρεσίας αιτήσεων στον Πρωτόκολλο Dienst (βλέπε στην παράγραφο 7.3 και στο Παράρτημα 4) προστέθηκε η νέα υπηρεσία και οι νέες αιτήσεις στην λίστα των αιτήσεων με την κλήση της ρουτίνας *Dispatch* ενώ ο κώδικας για την εξυπηρέτηση των αιτήσεων τοποθετήθηκε σε ένα νέο αρχείο το *Optional/statistics.pl*. Οι ρουτίνες για την εξυπηρέτηση των νέων αιτήσεων είναι οι *Print_Params*, η οποία συνδέεται με τον τοπικό επόπτη επίδοσης ζητώντας του όλη την βάση των παραμέτρων επίδοσης (για τον τρόπο που γίνεται η επικοινωνία βλέπε στο Παράρτημα 1), στέλνει μία αίτηση για τις τοπικές παραμέτρους επίδοσης σε κάθε εξυπηρετητή δεικτοδότησης που γνωρίζει και μορφοποιεί τα αποτελέσματα σε HTML πίνακες (βλέπε το Σχήμα 17), και *Print_Local_Params*, η οποία συνδέεται με τον τοπικό επόπτη ζητώντας του όλη την βάση των παραμέτρων επίδοσης και μορφοποιεί και παρουσιάζει τις παραμέτρους.

Τέλος, οι ρουτίνες για την εξισορρόπηση φόρτου βρίσκονται σε ένα νέο αρχείο το *Optional/load_balancing.pl*. Η εξισορρόπηση φόρτου βασίζεται στην αναδιάταξη του πίνακα με τους εξυπηρετητές δεικτοδότησης (για περισσότερες λεπτομέρειες βλέπε την παράγραφο 7.4), επομένως, η βασική ρουτίνα είναι η *index_server_sort*, η οποία ανάλογα με το όρισμα της επιλέγει κάποιον από τους αλγορίθμους και βασική της λειτουργία είναι η επιλογή της κατάλληλης ρουτίνας για την επικοινωνία με τον επόπτη για την λήψη των αναγκαίων στοιχείων επίδοσης και η αναδιάταξη του πίνακα κατακερματισμού (**hash table**) *authority_to_index_server*. Η κλήση της ρουτίνας γίνεται στο σημείο που πρόκειται να γίνει η επιλογή των εξυπηρετητών δεικτοδότησης για κάθε αρχή έκδοσης στην ρουτίνα *authorities_get_index_servers* στο *Kernel/distributed.pl*.

Παράρτημα 7. Λεξικό Τεχνικών Όρων.

A

Αίτηση Αναζήτησης
Αλληλεπιδραστικός
Αναγνωριστικό Διεργασίας
Αναγνωριστικό Εγγράφου
Αναδιφώ
Ανάκληση
Αναπαραγωγή Διεργασίας
Ανοχή Λαθών
Αντικείμενο Περιεχομένου
Αξιοπιστία
Απευθείας Απάντηση
Αποσυνθέσεις Εγγράφων
Αποσφαλμάτωση
Αρχείο Καταγραφής
Αρχή Έκδοσης
Αρχιτεκτονική Περιοχών
Ασφάλεια

B

Βιβλιογραφική Εγγραφή

Γ

Γλώσσα Σημείωσης Υπερκειμένου

Δ

Διαδικασία Διαχείρισης της Βάσης των
Παραμέτρων Επίδοσης
Διαδικασία Μέτρησης
Διεπιφάνεια Χρήσης
Δοσοληψία

Ε

Εν Λειτουργία Επόπτης Επίδοσης
Εν Λειτουργία Καθοδήγηση
Ενήμερος
Εξέταση
εξισορρόπηση Φόρτου
Εξυπηρετητής
Εξυπηρετητής Αναγνωριστικών
Εξυπηρετητής Δεικτοδότησης
Εξυπηρετητής Περιοχής
Εξυπηρετητής Συγχωνευμένου
Καταλόγου Δεικτοδότησης
Εξυπηρετητής Υποστήριξης
Επανεκκίνηση

Query
Interactive
Process id
Document Handle
Browse
Retrieval
Fork
Fault-tolerance
Container Object
Dependability
Direct Callback
Document Decompositions
Debugging
Log File
[Publishing] Authority
Regional Architecture
Security/Safety

Bibliographic Record

Hypertext Markup Language-
HTML

Database Management Process

Measurement Process
User Interface
Transaction

On-line Performance Monitor
On-line Steering
Up to date
Probing
Load Balancing
Server
Handle Server
Index Server
Regional Server
Merged Index Server

Backup Server
Reset

Επίπεδο Μεταφοράς	Transport Layer
Z	
H	
Θ	
I	
K	
Καθυστέρηση Δικτύου	Network Latency
Κατάλογος Δεικτοδότησης	Index
Κατασκευή Προφίλ Προγράμματος	Program Profiling
Κωδικός Εγγράφου	Docid
Λ	
Λέξη Κλειδί	Keyword
Λογαριασμός Χρήσης	Account
M	
Μεταφερσιμότητα	Portability
Μηχανή Αναζήτησης	Search Engine
Μοντέλο Πελάτη-Εξυπηρετητή	Client-Server Model
N	
Ξ	
O	
Ομοιόμορφος Προσδιοριστής Πόρων	Uniform Resource Locator–URL
Ορμαθός Χαρακτήρων	String
Π	
Παγκόσμιο Δίκτυο	Internet
Παγκόσμιος [Δικτυακός] Ιστός	World Wide Web
Πελάτης Παγκόσμιου Ιστού	Web Client
Περιοδική Δειγματοληψία	Polling
Πίνακας Κατακερματισμού	Hash Table
Ποιότητα Υπηρεσίας	Quality of Service (QoS)
Πράκτορας	Agent
Πρότυπο Πρόσβασης	Access Pattern
P	
Ρουτίνα	Procedure
Σ	
Σημείο Αναφοράς	Named Anchor
Συμβάν	Event
Συνεδρία	Session
Σύστημα Αναγνωριστικών Εγγράφων	Handle System
T	
Τμήμα Εκτέλεσης	Module
Τυπικό Μοντέλο	Formal Model
Υ	
Υπερκείμενο	Hypertext
Υπηρεσία Αποθήκευσης	Repository Service
Υπηρεσία Δεικτοδότησης	Index Service
Υπηρεσία Ονοματοδοσίας	Name Service
Φ	

X

Χρήσιμος Χρόνος Συστήματος

[Μέγιστο] Χρονικό Περιθώριο

Χρονοσφραγίδα

Χώρος Αποθήκευσης

Ψ

Ψηφιακές Βιβλιοθήκες

Ω

Utilization

Timeout

Timestamp

Repository

Digital Libraries

