

Ατομική Διπλωματική Εργασία

**ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ
ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ ΕΠΙΔΟΣΗΣ ΔΙΚΤΥΩΝ
ΠΑΡΑΔΟΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΜΕ ΠΟΛΛΑΠΛΟΥΣ
ΕΞΥΠΗΡΕΤΗΤΕΣ ΠΕΡΙΕΧΟΜΕΝΟΥ**

Νίκος Δημητρίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Προσομοίωση και Πειραματική Αξιολόγηση Αλγορίθμων για βελτίωση της
επίδοσης Δικτύων Παράδοσης Περιεχομένου με Πολλαπλούς Εξυπηρετητές
Περιεχομένου**

Νίκος Δημητρίου

Επιβλέποντες Καθηγητές
Γεωργίου Χρύσης και Πάλλης Γιώργος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2011

Ευχαριστίες

Θα ήθελα πρώτα από όλα να ευχαριστήσω τους επιβλέποντες καθηγητές μου, Επίκουρο Καθηγητή Χρύση Γεωργίου και Λέκτορα Γιώργο Πάλλη, για την μοναδική ευκαιρία που μου έδωσαν να εργαστώ σε αυτό το τομέα της Πληροφορικής και συγκεκριμένα στη εργασία αυτή.

Θα ήθελα να δώσω τις ευχαριστίες μου στην φίλη και συμφοιτήτρια μου Έλλη Ζαβού για την άψογη συνεργασία που είχαμε κατά την διάρκεια της διπλωματικής αυτής. Την ευχαριστώ για την πολύτιμη συνεισφορά της, καθώς δεν ήταν δυνατή η ολοκλήρωση της μελέτης αυτής χωρίς την συμμετοχή της και την βοήθεια της στο πειραματικό κομμάτι.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε σε όλη την διάρκεια των σπουδών μου.

Περίληψη

Τα Δίκτυα Παράδοσης Περιεχομένου (CDNs) αναπτύχθηκαν για να βελτιώσουν την επίδοση του Παγκόσμιου Ιστού προσφέροντας αξιόπιστες εφαρμογές και υπηρεσίες (εξ αποστάσεως εκπαίδευση, τηλεδιασκέψεις, μετάδοση πολυμεσικών δεδομένων κλπ) κατανέμοντας το περιεχόμενο στους CDN εξυπηρετητές, οι οποίοι βρίσκονται τοποθετημένοι κοντά στους χρήστες.

Η διπλωματική εργασία πραγματεύεται ένα σύνολο τεχνικών και μεθοδολογιών για την αποτελεσματικότερη διαχείριση των αντικειμένων στον Π.Ι., με κίνητρο να ελαττωθεί η καθυστέρηση πρόσβασης των πελατών στις προσφερόμενες υπηρεσίες. Συγκεκριμένα, η συνεισφορά της εργασίας επικεντρώνεται στην ανάπτυξη προγραμματιστικών εργαλείων για βελτίωση της επίδοσης Δικτύων Παράδοσης Περιεχομένου με πολλαπλούς εξυπηρετητές. Συγκεκριμένα ένας νέος αλγόριθμος υλοποιείται για να καθοριστεί το περιεχόμενο που θα αντιγραφεί στους CDN εξυπηρετητές, λαμβάνοντας υπόψη τα κοινά ενδιαφέροντα των χρηστών κατά την πλοήγησή τους σε πολλαπλούς εξυπηρετητές. Οι ιστοσελίδες ενός Διαδικτυακού τόπου ομαδοποιούνται με βάση τη συνδεσμολογία τους και οι ομάδες αυτές ονομάζονται κοινότητες του Π.Ι και είναι αυτές οι οποίες θα αντιγραφούν στους CDN εξυπηρετητές. Μέσω ενός αναλυτικού περιβάλλοντος προσομοίωσης, και χρησιμοποιώντας τόσο συνθετικά όσο και πραγματικά δεδομένα γίνεται η αξιολόγηση της προτεινόμενης μεθόδου.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Δίκτυα Παράδοσης Περιεχομένου	1
	1.2 Κοινότητες Ιστοσελίδων και Αλγόριθμος CiBC	4
	1.3 Κίνητρο και Συνεισφορά της Διπλωματικής Εργασίας	5
	1.4 Δομή διπλωματικής	6
Κεφάλαιο 2	Ορολογίες και Ορισμοί	7
	2.1 Γενικές Ορολογίες	7
	2.2 Ορισμός ΚΕΑ και ΚΕΚ	8
Κεφάλαιο 3	Περιγραφή Εργαλείων	12
	3.1 Αλγόριθμοι προς Υλοποίηση	12
	3.2 Υπάρχοντα Εργαλεία	13
	3.3 Εργαλεία που Υλοποιήθηκαν	16
	3.3.1 Εργαλεία Συνένωσης	17
	3.3.2 Εργαλεία για Αλγόριθμους	19
	3.3.3 Βοηθητικά Εργαλεία	25
	3.3.4 Προβλήματα κατά την υλοποίηση	25
Κεφάλαιο 4	Πειραματική Αξιολόγηση.....	27
	4.1 Γενική Περιγραφή	27
	4.2 Πειράματα Εύρεσης Ιδανικού Ποσοστού Χ	27
	4.3 Πειράματα Εύρεσης Ιδανικού Ποσοστού Υ	30
Κεφάλαιο 5	Συμπεράσματα	39
	5.1 Γενικά Συμπεράσματα	39
	5.2 Μελλοντικές Εργασίες	40
	Βιβλιογραφία	41
	Παράρτημα Α.....	A-1

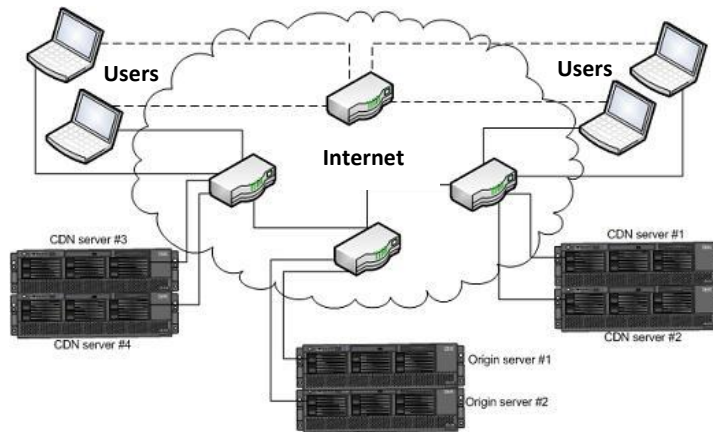
Κεφάλαιο 1

Εισαγωγή

1.1 Δίκτυα Παράδοσης Περιεχομένου	1
1.2 Κοινότητες Ιστοσελίδων και Αλγόριθμος CiBC	4
1.3 Κίνητρο και Συνεισφορά της Διπλωματικής Εργασίας	5
1.4 Δομή διπλωματικής	6

1.1 Δίκτυα Παράδοσης Περιεχομένου

Τα Δίκτυα Παράδοσης Περιεχομένου - ΔΠΠ (Content Delivery Network) [3] αναπτύχθηκαν για να βελτιώσουν την επίδοση του δικτύου προσφέροντας γρήγορες και αξιόπιστες εφαρμογές και υπηρεσίες κατανέμοντας το περιεχόμενο στους εξυπηρετητές εναποθήκευσης (cache servers) οι οποίοι βρίσκονται τοποθετημένοι κοντά στους χρήστες. Τα CDNs αποτελούνται από εξυπηρετητές αναπαραγωγής (surrogate servers) δηλαδή εξυπηρετητές οι οποίοι είναι τοποθετημένοι κατανεμημένα σε διάφορα σημεία στον κόσμο και συνεργάζονται με τους πηγαίους εξυπηρετητές (origin servers) για την μετάδοση ιστοσελίδων και περιεχομένου αποδοτικά στους χρήστες (users), βλέπε Εικόνα 1-1. Στους πηγαίους εξυπηρετητές τοποθετούνται ιστοσελίδες και στους εξυπηρετητές αναπαραγωγής αντίγραφα των ιστοσελίδων για να βρίσκονται πιο κοντά στους χρήστες. Με αυτό τον τρόπο μειώνεται η συμφόρηση και ο χρόνος διακίνησης πληροφοριών στο δίκτυο, εξυπηρετώντας γρηγορότερα τους χρήστες που τις ζητούν.



Εικόνα 1-1: Αναπαράσταση ενός ΔΠΠ [8]

Οι πληροφορίες διακινούνται από τους users στους εξυπηρετητές αναπαραγωγής και από τους εξυπηρετητές αναπαραγωγής στους πηγαίους εξυπηρετητές χωρίς να παρεμβάλλονται κάποια αιτήσεις χρηστών. Οι δύο αυτές ροές συμβάλλουν στην μείωση της συμφόρησης των πηγαίων εξυπηρετητών από τις πολλαπλές αιτήσεις προς αυτούς και επίσης συνεισφέρουν στην αύξηση της διαθεσιμότητας των πληροφοριών, αφού αυτές βρίσκονται πιο κοντά στους χρήστες. Οι πολλαπλοί εξυπηρετητές αναπαραγωγής που βρίσκονται κατανεμημένοι παρέχουν αντίγραφα των πληροφοριών και έτσι εξυπηρετούν ταυτόχρονα και γρηγορότερα χρήστες που κάνουν αιτήσεις προς αυτούς, χωρίς να επιβαρύνουν τους πηγαίους εξυπηρετητές. Επιπλέον, οι ιστοσελίδες που υπάρχουν στους εξυπηρετητές βρίσκονται σε κοινότητες, που προσφέρει καλύτερη διαχείριση των ιστοσελίδων. (Βλέπε Κεφάλαιο 1.2)

Τα Δίκτυα Παράδοσης Περιεχομένων βασίζονται σε παρακάτω τρεις παράγοντες που επηρεάζουν την αποδοτικότητά τους [3].

Πρώτος παράγοντας είναι η τοπολογία του δικτύου, δηλαδή η γεωγραφική τοποθεσία των εξυπηρετητών αναπαραγωγής. Οι θέσεις των εξυπηρετητών αναπαραγωγής επιλέγονται με τέτοιο τρόπο ούτως έτσι ώστε να επιτευχθεί μεγαλύτερη απόδοση του δικτύου και ταυτόχρονα μικρότερο κόστος υποδομής. Έχοντας όμως πολλούς εξυπηρετητές σε κοντινές τοποθεσίες προκαλεί σπατάλη χρόνου και υπολογιστικής δύναμης. Αυτό συμβαίνει για το λόγο ότι πρέπει πρώτα απ' όλα να αντιγραφούν οι πληροφορίες σε όλους τους εξυπηρετητές και επιπλέον θα χρειάζεται αρκετό κόστος ενημέρωσης και διαχείρισης τους. Πολλοί εξυπηρετητές δεν σημαίνει απαραίτητα

μεγαλύτερη αποδοτικότητα γι' αυτό και απαιτείτε σωστή τοποθέτηση των εξυπηρετητών.

Δεύτερος παράγοντας είναι η επιλογή του περιεχομένου που αντιγράφεται στους εξυπηρετητές αναπαραγωγής. Υπάρχουν διάφοροι τρόποι επιλογής των περιεχομένων που θα αντιγραφούν, με βάση τις προτιμήσεις των χρηστών και με βάση τη συνδεσμολογία των ιστοσελίδων μεταξύ τους μέσω υπερσυνδέσμων. Η αντιγραφή όλου του περιεχόμενου στους εξυπηρετητές είναι αδύνατο λόγω έλλειψης χώρου αλλά και για το ότι απαιτείτε αρκετό υπολογιστικό κόστος και χρόνος.

Τρίτος παράγοντας είναι η πολιτική που ακολουθείται για την αντιγραφή των περιεχομένων από τους πηγαίους εξυπηρετητές στους εξυπηρετητές αναπαραγωγής του δικτύου. Μια από τις πολιτικές αυτές είναι η **προανάκτηση και συνεργασία** (cooperative push-based), όπου τα περιεχόμενα ανακτούνται από πριν στους εξυπηρετητές αναπαραγωγής και μετά οι εξυπηρετητές συνεργάζονται για να κρατήσουν τα περιεχόμενά τους ανανεωμένα. Ανάλογα με την αίτηση του χρήστη επιλέγεται ο πιο κοντινότερος εξυπηρετητής που έχει το περιεχόμενο που ζητά ο χρήστης για να εξυπηρετήσει το αίτημα του. Αν δεν υπάρχει σε κανένα εξυπηρετητή αναπαραγωγής τότε η αίτηση μεταφέρεται στον πηγαίο εξυπηρετητή.

Η **ανάκτηση λόγω αίτησης χωρίς συνεργασία** (uncooperative pull-based), είναι η πολιτική στην οποία οι αιτήσεις των χρηστών μεταφέρονται στον κοντινότερο εξυπηρετητή αναπαραγωγής και αν το περιεχόμενο δεν υπάρχει στον εξυπηρετητή αυτό, η αίτηση μεταφέρεται είτε σε γειτονικό εξυπηρετητή αναπαραγωγής είτε στον πηγαίο εξυπηρετητή. Συγκεκριμένα αντιγράφεται το περιεχόμενο από τον πηγαίο εξυπηρετητή στον εξυπηρετητή αναπαραγωγής. Στην πολιτική αυτή οι εξυπηρετητές αναπαραγωγής δεν συνεργάζονται μεταξύ τους και έτσι δεν γνωρίζει ο ένας τι περιέχει ο γείτονάς του.

Τέλος, η **ανάκτηση λόγω αίτησης και συνεργασία** (cooperative pull-based), είναι η πολιτική στην οποία οι αιτήσεις μεταφέρονται στον κοντινότερο εξυπηρετητή αναπαραγωγής και αν το περιεχόμενο δεν υπάρχει στον εξυπηρετητή αυτό τότε αντιγράφεται από τον κοντινότερο εξυπηρετητή αναπαραγωγής που το περιέχει. Εδώ υπάρχει συνεργασία των εξυπηρετητών αφού γνωρίζουν από ποιόν να κάνουν την αίτηση αν δεν έχουν οι ίδιοι το περιεχόμενο που ψάχνουν.

1.2 Κοινότητες Ιστοσελίδων και Αλγόριθμος CiBC

Οι ιστοσελίδες ενός Διαδικτυακού τόπου ομαδοποιούνται με βάση τη συνδεσμολογία τους και οι ομάδες αυτές ονομάζονται κοινότητες του Παγκόσμιου Ιστού. Αν θεωρήσουμε τον ιστότοπο ένα γράφο, όπου οι κόμβοι είναι ιστοσελίδες και οι ακμές είναι οι υπερσυνδέσμοι μεταξύ τους, τότε οι κοινότητες, είναι ουσιαστικά υπο-γράφοι, οι οποίοι έχουν περισσότερες ακμές στο εσωτερικό τους (που συνδέουν δηλαδή τις ιστοσελίδες μεταξύ τους μέσα στην κοινότητα) και λιγότερες προς τα έξω της κοινότητας (που την συνδέουν με άλλες κοινότητες).

Μια τεχνική που προτάθηκε για την επιλογή των περιεχομένων που αντιγράφονται στους εξυπηρετητές αναπαραγωγής ενός ΔΠΠ, είναι με την εύρεση κοινοτήτων στις ιστοσελίδες του πηγαίου εξυπηρετητή [3]. Ο αλγόριθμος CiBC [3] προτάθηκε για να βρίσκει τις κοινότητες μεταξύ ιστοσελίδων σύμφωνα με την Μεταξύ Κεντρικότητα (Betweenness Centrality) του κάθε κόμβου, στο γράφο που αποτελούν τα περιεχόμενα ενός πηγαίου εξυπηρετητή. Η Μεταξύ Κεντρικότητα δείχνει πόσο κεντρικός είναι ένας κόμβος στο γράφο. Αν έχει μεγάλη Μεταξύ Κεντρικότητα σημαίνει πως τα περισσότερα μονοπάτια περνούν από τον κόμβο και επομένως ο χρήστης είναι πιο πιθανό να ζητήσει αυτή την ιστοσελίδα αν είναι σε κάποια γειτονική. Ο αλγόριθμος CiBC υπολογίζει αρχικά τη Μεταξύ Κεντρικότητα κάθε κόμβου και έπειτα δημιουργεί τις κοινότητες γύρω από τους κόμβους που έχουν την ελάχιστη τιμή. Σύμφωνα με το βαθμό συνδεσιμότητας μεταξύ των κοινοτήτων μπορεί να ενώσει και κάποιες από τις κοινότητες κάνοντας τις μεγαλύτερες. Μετά, ανάλογα με τη μνήμη του εξυπηρετητή, μπορεί να κάνει αλλαγές στους κόμβους των κοινοτήτων, ούτως ώστε να χωρούν στη μνήμη των εξυπηρετητών αναπαραγωγής που θα αντιγραφούν.

Ο αλγόριθμος αυτός είναι πολύ καλός γιατί ο χρόνος που χρειάζεται για να υπολογιστούν οι κοινότητες είναι ανάλογος με τον αριθμό των κόμβων (δηλαδή με τον αριθμό των ιστοσελίδων του εξυπηρετητή) ενώ η επίδοσή του επηρεάζεται μόνο γραμμικά από τον αριθμό των ακμών (δηλαδή των συνδέσεων μεταξύ των ιστοσελίδων του εξυπηρετητή).

1.3 Κίνητρο και Συνεισφορά της Διπλωματικής Εργασίας

Ένα από τα βασικά προβλήματα στα Δίκτυα Παράδοσης Περιεχομένων είναι η αυξημένη ζήτηση ιστοσελίδων που έχουν κοινό σημασιολογικό περιεχόμενο όπως για παράδειγμα το *eBay.com* μαζί με το *Amazon.com* που αφορούν αγορές προϊόντων μέσω διαδικτύου και το *BBC.com* μαζί με το *CNN.com* που αφορούν ιστοσελίδες για νέα της ξένης επικαιρότητας. Κάτι πιο ενδιαφέρον όμως είναι οι ιστοσελίδες που έχουν το ίδιο σημασιολογικό ενδιαφέρον για τους χρήστες βάσει την ζήτηση και τις προτιμήσεις τους. Δηλαδή παρατηρείτε κοινό ενδιαφέρον από τους χρήστες για συγκεκριμένες ιστοσελίδες, που αν αυτές βρίσκονταν μαζί σε ένα εξυπηρετητή θα ήταν αποδοτικότερο για την ανάκτηση τους. Ακόμα ένα άλλο πρόβλημα που προκύπτει είναι αφού εντοπιστούν οι κοινού ενδιαφέροντος ιστοσελίδες για τον χρήστη, πρέπει να αποφασιστεί σε ποιους εξυπηρετητές να τοποθετηθούν οι ιστοσελίδες αυτές. Τα προβλήματα οδεύουν στην αύξηση της συμφόρησης σε ΔΠΠ και των αποτυχημένων εμφανίσεων στη μνήμη (cache misses) που δημιουργούνται.

Σκοπός αυτής της διπλωματικής εργασίας είναι η εύρεση ιστοσελίδων κοινού σημασιολογικού ενδιαφέροντος για τον χρήστη, ούτως ώστε να μεταφερθούν σε κατάλληλους εξυπηρετητές αναπαραγωγής και η μελέτη της απόδοσης του δικτύου μετά από την μεταφορά των ιστοσελίδων σε αυτούς. Θα πρέπει να δούμε πότε αξίζει να γίνεται η συγχώνευση των κοινού ενδιαφέροντος κοινοτήτων στους ίδιους εξυπηρετητές και πόσο επηρεάζει την απόδοση του δικτύου.

Η συνεισφορά της διπλωματικής εργασίας συνοψίζεται ως ακολούθως:

- Υλοποιήθηκε ένα εργαλείο όπου βάσει ενός αλγόριθμου που αναπτύχθηκε στην εργασία [1] θα βρίσκει τις κοινού σημασιολογικού ενδιαφέροντος ιστοσελίδες για το χρήστη βάσει τις αιτήσεις από $n \geq 2$ πηγαίους εξυπηρετητές. Οι ιστοσελίδες αυτές που επιστρέφονται είναι βάσει κάποιου ποσοστού κοινού ενδιαφέροντος. Το ποσοστό αυτό δίνεται σαν είσοδος στο εργαλείο και στην έξοδό του επιστρέφονται οι ιστοσελίδες αυτές που το ικανοποιούν. Στη συνέχεια εντοπίζονται οι κοινότητες που θα αντιγραφούν, στις οποίες ανήκουν οι ιστοσελίδες αυτές.

- Πειραματική αξιολόγηση με χρήση του προσομοιωτή CDNSim [6]. Χρησιμοποιώντας τόσο συνθετικά όσο και πραγματικά δεδομένα γίνεται η αξιολόγηση της προτεινόμενης μεθόδου.
- Δίνονται κάποια αποτελέσματα για τις ιδανικές τιμές των ποσοστών κοινού ενδιαφέροντος αντικειμένων και κοινοτήτων όπως επίσης και κάποιες τεχνικές λεπτομέρειες του δικτύου (αριθμός εξυπηρετητών αναπαραγωγής) και σε ποιές περιπτώσεις έχουμε βελτίωση της απόδοσης.

1.4 Δομή διπλωματικής

Στο Κεφάλαιο 2 δίνονται κάποιες γενικές ορολογίες και συγκεκριμένοι ορισμοί για το θέμα και στο Κεφάλαιο 3 επεξηγούνται οι αλγόριθμοι που θα υλοποιηθούν και δίνεται μια περιγραφή των αλγορίθμων αυτών πιο βαθιά. Στο Κεφάλαιο 4 επεξηγούνται τα πειράματα που έχουν γίνει για την εύρεση των απαιτούμενων ποσοστών και τέλος στο Κεφάλαιο 5 δίνονται τα συμπεράσματα της διπλωματικής και κάποιες προτεινόμενες μελλοντικές εργασίες. Στο Παράρτημα Α δίνεται μέρος τους κώδικα κάποιων εργαλείων που υλοποιήθηκαν στα πλαίσια της διπλωματικής αυτής όπως επίσης και ένα εγχειρίδιο με την εκτέλεση των βασικών εργαλείων της εργασίας.

Κεφάλαιο 2

Ορολογίες και Ορισμοί

2.1	Γενικές Ορολογίες	7
2.2	Ορισμός ΚΕΑ και ΚΕΚ	8

2.1 Γενικές Ορολογίες

Στη διπλωματική εργασία επαναλαμβάνονται κάποιες ορολογίες οι οποίες επεξηγούνται εδώ:

Ιστότοπος είναι το σύνολο όλων των ιστοσελίδων ενός συγκεκριμένου θέματος. Οι ιστοσελίδες αυτές συνδέονται μεταξύ τους μέσω των υπερσυνδέσμων.

Πηγαίοι εξυπηρετητές είναι οι εξυπηρετητές του δικτύου, στους οποίους βρίσκονται αποθηκευμένοι ολόκληροι ιστότοποι.

Εξυπηρετητές αναπαραγωγής είναι εξυπηρετητές που βρίσκονται κατανεμημένοι στο δίκτυο και στους οποίους αποθηκεύονται αντίγραφα από το περιεχόμενο των πηγαίων εξυπηρετητών.

Αντικείμενο ή Αντικείμενο Ιστού (web-object) είναι ένα αντικείμενο που ανήκει σε κάποιο πηγαίο εξυπηρετητή ή αλλιώς μια ιστοσελίδα.

Σύνοδος (session) τα αντικείμενα τα οποία έχει ζητήσει ένας χρήστης από ένα εξυπηρετητή κάποια χρονική στιγμή.

Αρχείο Συνόδων (transactions file) είναι ένα αρχείο το οποίο περιέχει συνόδους και κάθε γραμμή αντιστοιχεί σε μια σύνοδο ενός χρήστη.

Αρχείο Αιτήσεων (trace file) είναι ένα αρχείο που περιέχει τις αιτήσεις που γίνονται σε κάποιο πηγαίο εξυπηρετητή για τα αντικείμενα που περιέχει στη μνήμη του.

Αρχείο Κοινοτήτων (community file) είναι ένα αρχείο που περιέχει σε κάθε γραμμή τα αντικείμενα που βρίσκονται στην ίδια κοινότητα. Βλέπε Εικόνα 3-15.

Μεταβλητή	Περιγραφή
X	Ποσοστό κοινού ενδιαφέροντος αντικειμένων
Y	Ποσοστό κοινού ενδιαφέροντος κοινοτήτων
Z	Ποσοστό ζευγών κοινού ενδιαφέροντος ιστοσελίδων από το σύνολο των ζευγών ιστοσελίδων δύο κοινοτήτων
C	Ποσοστό συγκρισιμότητας δύο κοινοτήτων

Πίνακας 2-1 Περιγραφή μεταβλητών που αναφέρονται στην εργασία αυτή

2.2 Ορισμός ΚΕΑ και ΚΕΚ

Δύο Αντικείμενα Ιστού θεωρούμε πως είναι ΚΕΚ (Κοινού Ενδιαφέροντος Αντικείμενα), αν το ποσοστό των υποκειμένων, από το σύνολο των όλων των υποκειμένων που κάνουν αιτήσεις στα σύνολα των αντικειμένων ιστού, που ζητούν και τα δυο αντικείμενα, είναι μεγαλύτερο ή ίσο με **X**, όπου το **X** ορίζεται ως όριο ποσοστού κοινού ενδιαφέροντος – ή αλλιώς βαθμός ομοιότητας – το οποίο θέτουμε εμείς.

Συγκεκριμένα, έχουμε δυο αντικείμενα ιστού, A και B. Το σύνολο των υποκειμένων που κάνουν αιτήσεις στα σύνολα των αντικείμενων ιστού είναι S. Αν το ποσοστό των υποκειμένων από το σύνολο S, που ζητούν και τα δυο αντικείμενα ιστού, είναι περισσότερο από το ποσοστό **X** που θέτουμε ως κατώτατο όριο, τότε τα αντικείμενα ιστού A και B είναι κοινού ενδιαφέροντος για τα υποκείμενα που τα ζητούν. Μας ενδιαφέρει όμως, τα αντικείμενα αυτά να είναι από διαφορετικά σύνολα αντικειμένων, ούτως ώστε η αντιγραφή τους σε κοινούς καταχωρητές αντικειμένων ιστού να επιφέρει αποδοτικότερα αποτελέσματα (να είναι από διαφορετικούς πηγαίους εξυπηρετητές).

$O_1, O_2 : \{\text{σύνολα αντικειμένων ιστού}\}$

$S : \{\text{υποκείμενα που ζητούν αντικείμενα ιστού}\}$

$A \in O_1$ (A αντικείμενο ιστού από το σύνολο O_1)

$B \in O_2$ (B αντικείμενο ιστού από το σύνολο O_2)

$S_A : \{\text{υποκείμενα που ζητούν το αντικείμενο ιστού A}\}$

$S_B : \{\text{υποκείμενα που ζητούν το αντικείμενο ιστού B}\}$

X: ποσοστό κατώτατου ορίου για τα ΚΕΑ

$$\text{Αν } \frac{|S_A \cap S_B|}{|S|} \geq X \text{ τότε}$$

A και B είναι ΚΕΚ

Με τον ίδιο τρόπο, για n αντικείμενα ιστού, θεωρούμε πως είναι ΚΕΑ για τα υποκείμενα, αν ο αριθμός των υποκειμένων που τα ζητούν δια το σύνολο όλων των υποκειμένων που κάνουν αιτήσεις, είναι ίσο ή περισσότερο από το κατώτατο όριο X . Είναι απαραίτητο τα αντικείμενα ιστού να προέρχονται από n διαφορετικά σύνολα αντικειμένων ιστού.

$$\text{Αν } \frac{|S_1 \cap S_2 \cap S_3 \cap \dots \cap S_n|}{|S|} \geq X \text{ τότε}$$

1, 2, 3, ... n είναι ΚΕΚ

Οι ΚΕΚ (Κοινού Ενδιαφέροντος Κοινότητες) είναι από διαφορετικούς πηγαίους εξυπηρετητές, ούτως ώστε να αντιγράφονται μαζί στους εξυπηρετητές αναπαραγωγής. Θα πρέπει να θέσουμε ένα δεύτερο κατώτατο όριο Y , το οποίο θα καθορίζει το ποσοστό κοινού ενδιαφέροντος δυο κοινοτήτων που μας ενδιαφέρει. Αν δυο κοινότητες έχουν βαθμό ομοιότητας μεγαλύτερο ή ίσο από αυτό το ποσοστό τότε θα θεωρούνται ΚΕΚ.

Ο αλγόριθμος **Similarity2wo** [1] βρίσκει το κοινό ενδιαφέροντος ποσοστό μεταξύ 2 αντικειμένων ιστού. Βρίσκει δηλαδή πόσα υποκείμενα, από αυτά που κάνουν αιτήσεις σε αντικείμενα ιστού, ζητούν και τα δυο αντικείμενα, το καθένα από διαφορετικό καταχωρητή αντικειμένων ιστού.

Για να βρούμε το ποσοστό κοινού ενδιαφέροντος κοινοτήτων από 2 πηγαίους εξυπηρετητές, ακολουθείται ο πιο κάτω αλγόριθμος [1]:

- 1) Αρχικά, χρησιμοποιώντας τον αλγόριθμο **CiBC** βρίσκουμε τις κοινότητες ιστοσελίδων κάθε σύνολο αντικειμένων ιστού – WC_1, WC_2 .
- 2) Έπειτα, για κάθε κοινότητα, των δυο συνόλων αντικειμένων ιστού, μετρούμε τον αριθμό μοναδικών χρηστών που κάνουν αιτήσεις για αντικείμενα ιστού της συγκεκριμένης κοινότητας – $u(wc)$.

- 3) Μετά, πρέπει να πάρουμε από τα σύνολα, ζεύγη από κοινότητες – έστω $wc_1 \in WC_1$, $wc_2 \in WC_2$ – και να ελέγξουμε κατά πόσο είναι συγκρίσιμες. Ουσιαστικά, για να είναι συγκρίσιμες πρέπει τα $u(wc_1)$ και $u(wc_2)$ να είναι της ίδιας τάξης.
- 4) Για τα ζεύγη κοινοτήτων που είναι συγκρίσιμες, θεωρούμε πως είναι υποψήφια για να θεωρηθούν κοινού ενδιαφέροντος. Έτσι, από κάθε κοινότητα παίρνουμε ένα αριθμό αντικειμένων, τα K πιο περιζήτητα – **topK(wc₁), topK(wc₂)** .
- 5) Από τα topK αντικείμενα κάθε κοινότητας, κάνουμε όλα τα πιθανά ζεύγη αντικειμένων για τα οποία βρίσκουμε, από τον **Similarity2wo**, τον βαθμό ομοιότητας μεταξύ τους. Αν το ποσοστό είναι μεγαλύτερο από **X** (το κατώτατο όριο που θέσαμε για κριτήριο ομοιότητας δυο αντικειμένων) τότε αυξάνουμε ένα μετρητή. Στο τέλος, υπολογίζουμε το ποσοστό ζευγών αντικειμένων (**z**), που είναι κοινού ενδιαφέροντος, από τα topK ζεύγη.
- 6) Αν το ποσοστό **z** είναι μεγαλύτερο ή ίσο από **Y**, τότε και οι δύο κοινότητες είναι ΚΕΚ.

X: ποσοστό κατώτατου όριου για ΚΕΑ

Y: ποσοστό κατώτατου όριου για ΚΕΚ

O₁, O₂ : {σύνολα αντικειμένων ιστού}

WC₁, WC₂ : {σύνολα κοινοτήτων}

WC₁: {wc: wc ∈ O₁}

WC₂: {wc: wc ∈ O₂}

u(wc) : αριθμός μοναδικών χρηστών που ζητούν αντικείμενα της wc

*Αν $u(wc_1) = c * u(wc_2)$ τότε οι wc_1 και wc_2 είναι συγκρίσιμες*

Γιακάθε ζεύγος αντικειμένων τα $\langle topK(wc_1), topK(wc_2) \rangle$

Αν $Similarity2WO(topK(wc_1), topK(wc_2)) \geq X$ τότε count ++

$$z = \frac{count}{\text{αριθμός ζευγών αντικειμένων από τις δύο κοινότητες}}$$

Αν $z \geq Y$ τότε οι wc_1 και wc_2 είναι ΚΕΚ

Είναι σημαντικό να αναφέρουμε πως το X θα πρέπει να είναι μεταξύ κάποιων ορίων, $X_{\min} \leq X \leq X_{\max}$.

Από κάτω ($X_{\min}$) φράσσεται από το ποσοστό ομοιότητας αντικειμένων ιστού του ίδιου συνόλου. Από πάνω ($X_{\max}$) φράσσεται από το ποσοστό X που βρέθηκε πειραματικά στα πλαίσια της διπλωματικής αυτής.

Κεφάλαιο 3

Περιγραφή Εργαλείων

3.1 Αλγόριθμοι προς Υλοποίηση	12
3.2 Υπάρχοντα Εργαλεία	13
3.3 Εργαλεία που Υλοποιήθηκαν	16
3.3.1 Εργαλεία Συνένωσης	17
3.3.2 Εργαλεία για Αλγόριθμους	19
3.3.3 Βοηθητικά Εργαλεία	25

3.1 Αλγόριθμοι προς Υλοποίηση

Τα εργαλεία που υλοποιήθηκαν στα πλαίσια της διπλωματικής βασίστηκαν στους αλγορίθμους που αναπτύχθηκαν στην εργασία [1]. Μετά από μελέτη των αλγορίθμων αυτών επιλέχτηκαν η αποδοτικότερες δομές δεδομένων που θα χρησιμοποιηθούν για αποθήκευση και επεξεργασία των δεδομένων. Οι δομές αυτές είναι Πίνακες και Λίστες για τον λόγο ότι είναι εύκολες να υλοποιηθούν σε προγραμματιστικό επίπεδο και μπορούμε εύκολα να ελέγξουμε την πολυπλοκότητα χώρου και χρόνου τους.

Οι αλγόριθμοι που υλοποιήθηκαν είναι οι εξής όπως αυτοί αναφέρονται στην εργασία [1]:

- Ο αλγόριθμος **SimilarityKwo** βρίσκει το ποσοστό κοινού ενδιαφέροντος k αντικειμένων ιστού.
- Ο αλγόριθμος **CommIntKwo**, χρησιμοποιεί τον **SimilarityKwo**, και βρίσκει όλες τις k -άδες αντικειμένων (από διαφορετικούς πηγαίους εξυπηρετητές) που είναι κοινού ενδιαφέροντος μεγαλύτερου του κατώτατου ορίου το οποίο θέτουμε ως X .
- Ο αλγόριθμος **CommInt2wc** βρίσκει τα ζεύγη ΚΕΚ
- Ο αλγόριθμος **AssignWCtoSurr** αναθέτει τις ΚΕΚ σε εξυπηρετητές αναπαραγωγής.

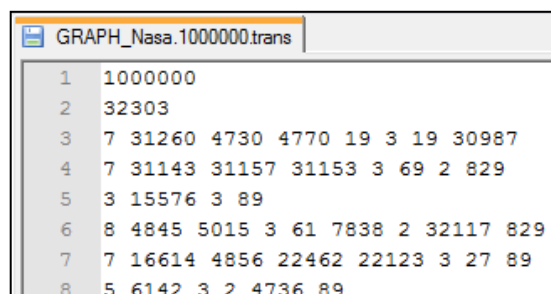
3.2 Υπάρχοντα Εργαλεία

Για την διεξαγωγή των πειραμάτων χρειάστηκαν συγκεκριμένα αρχεία με ειδική μορφή τα οποία δημιουργήθηκαν με κάποια έτοιμα εργαλεία τα οποία υπήρχαν διαθέσιμα [2].

Πρώτο διαθέσιμο εργαλείο είναι μια **Γεννήτρια Γράφων Ιστοσελίδων** (Web-graph generator) [2]. Το εργαλείο αυτό δημιουργεί ένα γράφο που αντιπροσωπεύει ένα ιστότοπο, χρησιμοποιώντας πραγματικούς ιστότοπους από το διαδίκτυο. Αρχικά κάνει ιχνηλασία (crawling) σε ένα ιστότοπο και στην συνέχεια βασισμένο σε αυτό δημιουργεί τον γράφο. Το αποτέλεσμα το παίρνουμε σε αρχείο και έχει την μορφή κόμβος-κόμβος, δηλαδή την σύνδεση μεταξύ δύο κόμβων από τον ιστιότοπο.

Δεύτερο εργαλείο είναι μια **Γεννήτρια Αιτήσεων** (Trace generator) [2] το οποίο λαμβάνει στην είσοδό του τον γράφο ενός ιστότοπου και τον αριθμό των συνόδων που επιθυμούνται και δημιουργεί εικονικές αιτήσεις χρηστών, χρησιμοποιώντας αλγόριθμους που βασίζονται σε πραγματικά δεδομένα. Στην έξοδο παίρνουμε δύο αρχεία, το αρχείο συνόδου και το αρχείο αιτήσεων. Το αρχείο συνόδου δημιουργείται πρώτο και περιέχει τόσες συνόδους όσες του δώσουμε στην είσοδο και από αυτό τελικά παράγεται το αρχείο αιτήσεων. Η μορφή των αρχείων που παράγονται παίζει σημαντικό ρόλο για τα εργαλεία που υλοποιήθηκαν.

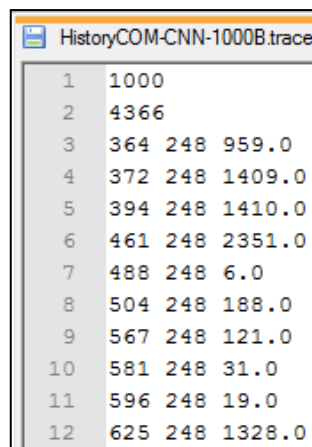
Ένα **αρχείο συνόδων** περιέχει σε κάθε γραμμή του μια σύνοδο εκτός από τις δύο πρώτες γραμμές που είναι κάποιες πληροφορίες για το αρχείο. Συγκεκριμένα ο αριθμός των συνόδων που περιέχει (γραμμή 1, εικόνα 3-1) και τα μοναδικά αντικείμενα που περιέχει το αρχείο (γραμμή 2, εικόνα 3-1). Έχει την κατάληξη “.trans”.



1	1000000
2	32303
3	7 31260 4730 4770 19 3 19 30987
4	7 31143 31157 31153 3 69 2 829
5	3 15576 3 89
6	8 4845 5015 3 61 7838 2 32117 829
7	7 16614 4856 22462 22123 3 27 89
8	5 6142 3 2 4736 89

Εικόνα 3-1 Ένα αρχείο συνόδων

Ένα **αρχείο αιτήσεων** περιέχει τις αιτήσεις των χρηστών σε τριάδες. Χρονο-σφραγίδα (timestamp) σε ακέραια μορφή ή δεκαδική, ταυτότητα χρήστη (user ID) και ταυτότητα αντικειμένου (object ID) εκτός από τις δύο πρώτες γραμμές που είναι κάποιες πληροφορίες για το αρχείο. Συγκεκριμένα ο αριθμός των χρηστών που περιέχει (γραμμή 1, εικόνα 3-2) και τα μοναδικά αντικείμενα που περιέχει το αρχείο (γραμμή 2, εικόνα 3-2). Οι ταυτότητες των αντικειμένων είναι μεταφρασμένες για να συμβαδίζουν με τα υπόλοιπα εργαλεία και για να γίνεται μια καθαρή αρίθμηση των αντικειμένων ξεκινώντας από το 0.



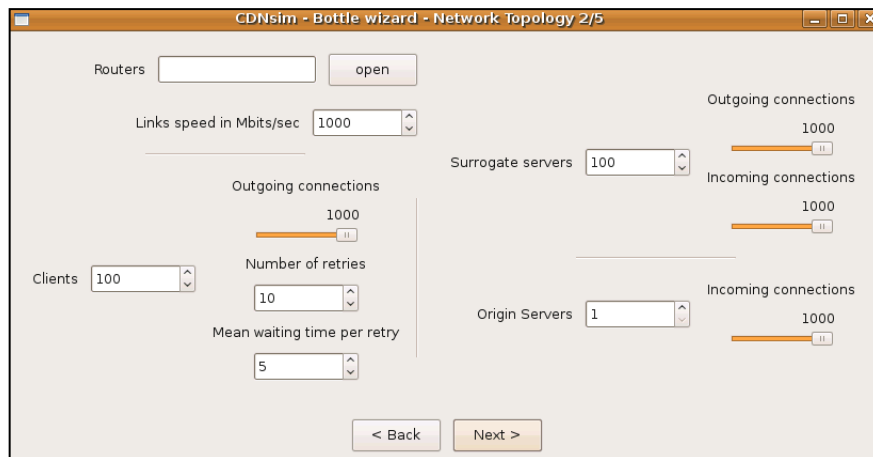
1	1000
2	4366
3	364 248 959.0
4	372 248 1409.0
5	394 248 1410.0
6	461 248 2351.0
7	488 248 6.0
8	504 248 188.0
9	567 248 121.0
10	581 248 31.0
11	596 248 19.0
12	625 248 1328.0

Εικόνα 3-2 Ένα αρχείο αιτήσεων

Η Γεννήτρια Αιτήσεων αποτελείται από τα script *make_trace.pl*, *trace_gen.pl* και το εκτελέσιμο αρχείο *trans_gen*. Το *make_trace.pl* είναι το γενικό script που χρησιμοποιεί τα υπόλοιπα και παράγει τα αρχεία συνόδων των γράφων, τα συνενώνει και έπειτα από το παραγόμενο δημιουργεί το τελικό αρχείο αιτήσεων. Για την δημιουργία των αρχείων συνόδων βάση του αρχείου γράφου που δόθηκε, υπεύθυνο είναι το *trans_gen*. Το πώς καλείται το εργαλείο αυτό και τα είδη των παραμέτρων του είναι γραμμένα στο *make_trace.pl*. Μετά το πέρας του αλγορίθμου θα έχει δημιουργηθεί ένα αρχείο συνόδων. Το αρχείο αυτό δίνεται στην είσοδο του τελευταίου script, *trace_gen.pl*, που εκτελείται και παράγει το τελικό αρχείο αιτήσεων. Το τι ακριβώς γίνεται είναι ότι για το κάθε αντικείμενο σε κάθε γραμμή ενός αρχείου συνόδων δημιουργείται μια τριάδα. Υπολογίζεται μια τυχαία χρόνο-σφραγίδα και για τον χρήστη, με αύξον αριθμό που ξεκινά από το 0, γράφεται στο παραγόμενο αρχείο αιτήσεων ότι έχει ζητήσει το αντικείμενο.

Το τρίτο εργαλείο που έχουμε στην διάθεσή μας υλοποιεί τον αλγόριθμο CiBC [2], ο οποίος βρίσκει τις κοινότητες κάποιου ιστότοπου. Παίρνει στην είσοδο του ένα αρχείο με τον γράφο του ιστότοπου και δίνει στην έξοδο του δύο αρχεία με τις κοινότητες που αντιστοιχούν στο γράφο. Οι κοινότητες αναπαριστούνται με τα αντικείμενα που ανήκουν σε μια κοινότητα να βρίσκονται στην ίδια γραμμή. Στο ένα από τα δύο αρχεία βρίσκονται οι ελαχιστοποιημένα ορφανές (minimized orphans) [2] κοινότητες, δηλαδή οι κοινότητες στις οποίες ο αλγόριθμος διαχειρίστηκε τα αντικείμενα που δεν ανήκουν σε καμία κοινότητα.

Τελευταίο εργαλείο που έχουμε στην διάθεση μας είναι ένας προσομοιωτής ΔΠΠ, ο CDNsim [6]. Δίνοντας κάποια αρχεία που περιγράφουν το δίκτυο μας καθώς και αρχεία αιτήσεων, ο προσομοιωτής παράγει κάποια στατιστικά για την απόδοση του δικτύου. Χρησιμοποιεί την ορολογία «μπουκάλι» (bottle), για να αναπαραστήσει την συλλογή των δεδομένων περιγραφής του δικτύου, δηλαδή το σενάριο του δικτύου. Ο προσομοιωτής αποτελείται από 5 βήματα για την εισαγωγή των δεδομένων για το δίκτυο μας και την δημιουργία ενός μπουκαλιού. Στο πρώτο βήμα επιλέγουμε την πολιτική με την οποία θα διαχειρίζονται τις ιστοσελίδες οι εξυπηρετητές. Έπειτα στο δεύτερο βήμα δημιουργούμε την τοπολογία μας, τοποθετώντας τους δρομολογητές (routers) και δηλώνοντας τον αριθμό των χρηστών, των πηγαίων εξυπηρετητών και των εξυπηρετητών αναπαραγωγής, όπως επίσης και κάποιες τεχνικές λεπτομέρειες για αυτούς, όπως η ταχύτητα των συνδέσεων στο δίκτυο. Στο τρίτο βήμα τοποθετούμε δύο αρχεία. Το πρώτο περιέχει όλες τις ιστοσελίδες των πηγαίων εξυπηρετητών και το δεύτερο τις αιτήσεις των χρηστών στο δίκτυο. Στο τέταρτο βήμα τοποθετούμε το αρχείο τοποθέτησης (placement file) (βλέπε τέλος υποκεφαλαίου 3.3.2) που δηλώνει τι περιέχει κάθε εξυπηρετητής αναπαραγωγής. Τέλος τοποθετούμε το όνομα του μπουκαλιού μας και έχοντας πλέον όλα τα δεδομένα είναι έτοιμο να «εκτελεστεί» (τρέξουν οι αιτήσεις) και θα δημιουργηθούν κάποια στατιστικά. Η δημιουργία στατιστικών γίνεται με κάποια επιπλέον εργαλεία του προσομοιωτή, τα οποία παράγουν τα στατιστικά σε μορφή HTML για καλύτερη ανάγνωση των αποτελεσμάτων. (Εικόνα 3-4).



Εικόνα 3-3 Βήμα 2 από τον προσομοιωτή CDNSim [7]

```

Bottle: test
CLIENT SIDE MEASUREMENTS
Completed requests: 500
Aborted requests: 0
Retries: 0
Not enough outgoing connections: 0
Mean response time: 0.01639287
SURROGATE SIDE MEASUREMENTS
Hit ratio percentage: 0.0
Byte hit ratio percentage: 0.0
Not enough outgoing connections: 0
Not enough incoming connections: 0
Failed pulls: 0
ORIGIN SIDE MEASUREMENTS
Not enough incoming connections: 0
SURROGATE SERVERS UTILITIES
Active surrogate servers: 34
Mean surrogate servers utility 0.516028628614

```

Εικόνα 3-4 Μορφή αρχείου στατιστικών του CDNSim

3.3 Εργαλεία που Υλοποιήθηκαν

Συνολικά για τους σκοπούς τις διπλωματικής, υλοποιήθηκαν πέντε εργαλεία. Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν είναι κυρίως C και Perl, όπως επίσης Java και Bash script. Τα εργαλεία υλοποιήθηκαν σε περιβάλλον Linux (Ubuntu 10.10). Σε κάποιες περιπτώσεις που εξηγούνται πιο κάτω έγιναν αλλαγές στα διαθέσιμα υπάρχοντα εργαλεία για να επιτευχθεί το επιθυμητό αποτέλεσμα και προστέθηκαν καινούρια κομμάτια σε αυτά.

3.3.1 Εργαλεία Συνένωσης

Το εργαλείο συνένωσης αιτήσεων χρηστών είναι η προεργασία για την ανάλυση των ζητήσεων των χρηστών. Δηλαδή πρέπει να δημιουργηθούν αιτήσεις χρηστών από διαφορετικούς πηγαίους εξυπηρετητές. Για τον λόγο ότι δεν μπορούμε εύκολα να βρούμε έτοιμα πραγματικά δεδομένα με πραγματικές αιτήσεις, όπως για παράδειγμα αρχεία κατάστασης (log files) με κίνηση (traffic) στους πηγαίους εξυπηρετητές, δημιουργούμε εικονικές αιτήσεις χρηστών βασιζόμενοι σε πραγματικά δεδομένα. Με τον τρόπο αυτό προσομοιώνονται οι επισκέψεις των χρηστών στις ιστοσελίδες, δηλαδή οι αιτήσεις που κάνει ο κάθε χρήστης από τις ιστοσελίδες διαφορετικών πηγαίων εξυπηρετητών. Όταν έχουμε πλέον στην διάθεση μας τις αιτήσεις των χρηστών δίνονται σαν είσοδο στο εργαλείο εύρεσης κοινού ενδιαφέροντος ιστοσελίδων. Σημειώνουμε ότι για το εργαλείο αυτό και γενικά για όλα τα υπόλοιπα εργαλεία υποθέτουμε ότι ένας ιστότοπος είναι ένας πηγαίος εξυπηρετητής.

Η συνένωση είναι ένα ενδιάμεσο κομμάτι ανάμεσα στην δημιουργία αρχείων συνόδων και αιτήσεων. Αφού έχουν δημιουργεί τα αρχεία συνόδων από η διαφορετικούς πηγαίους εξυπηρετητές, με την Γεννήτρια Αιτήσεων, τότε το εργαλείο συνένωσης παίρνει στην είσοδο τα αρχεία αυτά, τον αριθμό των χρηστών για τους οποίους θα δημιουργηθούν αιτήσεις, και διαλέγοντας μια γραμμή από τυχαίο αρχείο συνόδων κάθε φορά, την αντιγράφει σε μία κοινή γραμμή μέσα σε ένα καινούριο αρχείο συνόδων. Περισσότερα για τις συνόδους και τα αρχεία συνόδων στο Κεφάλαιο 3.2. Στη γραμμή αυτή ξεχωρίζει σε κάθε αντικείμενο σε ποιο πηγαίο εξυπηρετητή ανήκει προσθέτοντας την ταυτότητα (id) του εξυπηρετητή μετά την ταυτότητα του αντικειμένου. Για παράδειγμα, η εγγραφή 178.0: Το αντικείμενο 178 ανήκει στον πηγαίο εξυπηρετητή 0. Αξίζει να υπενθυμίσουμε ότι τα αντικείμενα αντιγράφονται από τα αρχεία συνόδων τα οποία παράγονται από τα αρχεία γράφων ιστότοπων του κάθε πηγαίου εξυπηρετητή. Όταν το καινούριο συνενωμένο αρχείο συνόδων έχει δημιουργηθεί, τότε συνεχίζεται η διαδικασία και το `trace_gen.pl` (Βλέπε Κεφάλαιο 3.2) αναλαμβάνει να παράξει τις τελικές αιτήσεις, οι οποίες και θα είναι συνενωμένες για διαφορετικούς πηγαίους εξυπηρετητές. Το όνομα του εργαλείου αυτού είναι `merge.pl` και έχει γραφτεί σε γλώσσα Perl. Παράλληλα υλοποιήθηκε και δεύτερο εργαλείο, παραλλαγή του προηγούμενου, που αντιγράφει για κάθε χρήστη μια γραμμή από όλα τα αρχεία

συνόδων που δόθηκαν. Ονομάζεται `merge2.pl`, και εκτελείται την θέση του `merge.pl` όπως περιγράφηκε πιο πάνω.

BBC	Ταυτότητα ισότοπου: 0	CNN	Ταυτότητα ισότοπου: 1
1	1000000	1	1000000
2	17031	2	42399
3	31 8542 6802 6591 108 8562 1771	3	33 3614 1 8 12 13938 7 41296 4 3 31
4	30 2810 2739 119 1036 11278	4	32 2743 9 99 3 2086 2096 5 100 39081

merge.pl

1	8542.0 6802.0 6591.0 108.0 8562.0 1771.0
2	3614.1 1.1 8.1 12.1 13938.1 7.1 41296.1 4.1 3.1 31.1
3	2810.0 2739.0 119.0 1036.0 11278.0 2743.1 9.1 99.1 3.1 2086.1 2096.1 5.1 100.1

Εικόνα 3 Error! Use the Home tab to apply 0 to the text that you want to appear here.-5 Παράδειγμα εκτέλεσης του εργαλείου `merge.pl` με τυχαία επιλογή αρχείων συνόδων

BBC	Ταυτότητα ισότοπου: 0	CNN	Ταυτότητα ισότοπου: 1
1	1000000	1	1000000
2	17031	2	42399
3	31 8542 6802 6591 108 8562 1771	3	33 3614 1 8 12 13938 7 41296 4 3 31
4	30 2810 2739 119 1036 11278	4	32 2743 9 99 3 2086 2096 5 100 39081

merge2.pl

1	8542.0 6802.0 6591.0 108.0 8562.0 1771.0 3614.1 1.1 8.1 12.1 13938.1 7.1 41296.1 4.1 3.1 31.1
2	2810.0 2739.0 119.0 1036.0 11278.0 2743.1 9.1 99.1 3.1 2086.1 2096.1 5.1 100.1

Εικόνα 3 Error! Use the Home tab to apply 0 to the text that you want to appear here.-6 Παράδειγμα εκτέλεσης του εργαλείου `merge2.pl` με επιλογή όλων των αρχείων συνόδων

Το γενικό εργαλείο που μπορεί να δημιουργήσει συνενωμένα αρχεία είναι το `make_trace.pl` (Βλέπε Κεφάλαιο 3.2) το οποίο έχει υποστεί κάποιες αλλαγές για να ενσωματωθεί και το `merge.pl`. Η δημιουργία όμως ενός αρχείου συνόδων, ειδικά αν θα δημιουργούν εκατομμύρια γραμμές, χρειάζεται αρκετή ώρα. Παραδείγματος χάριν ένα αρχείο συνόδων με 1.000.000 συνόδους χρειάστηκε σχεδόν δύο μέρες για να δημιουργηθεί. Όταν όμως δημιουργηθεί μια φορά τότε μπορεί να χρησιμοποιηθεί για αρκετά πειράματα, χωρίς να χρειαστεί να το δημιουργήσουμε ξανά. Για τον λόγο αυτό, έχει υλοποιηθεί ένα εργαλείο παραλλαγή του `make_trace.pl`, το οποίο προσπερνά το πρώτο βήμα κατασκευής των αρχείων συνόδων, τα οποία και δίνονται στην είσοδο του εργαλείου. Ονομάζεται `merge_traceGen.pl` και όταν εκτελείται έχουμε την επιλογή να διαλέξουμε ποιο είδος συνένωσης επιθυμούμε να χρησιμοποιήσουμε, δηλαδή επιλογή

γραμμής από τυχαίο αριθμό αρχείων συνόδων ή επιλογή γραμμής από όλα τα αρχεία που δίνονται.

3.3.2 Εργαλεία για Αλγόριθμους

Το πρώτο εργαλείο που υλοποιήθηκε, βρίσκει τις κοινού ενδιαφέροντος ιστοσελίδες μεταξύ n πηγαίους εξυπηρετητές, βάσει κάποιου αρχείου αιτήσεων το οποίο περιέχει αιτήσεις χρηστών σε αντικείμενα που ανήκουν στους εξυπηρετητές αυτούς και ενός ποσοστού ομοιότητας X . Το εργαλείο αυτό υλοποιεί τον αλγόριθμο για τη δημιουργία μιας δομής [1] που κάνει τη σύνδεση μεταξύ των χρηστών του δικτύου και των αντικειμένων ιστού που ανήκουν τους πηγαίους εξυπηρετητές. Στη συνέχεια υλοποιεί τον αλγόριθμο **CommIntKwo**. Στην είσοδο παίρνει το k που είναι ο αριθμός των αντικειμένων για τα οποία θέλουμε να υπολογίσουμε το ποσοστό ομοιότητας μεταξύ τους, όπως επίσης και το X . Το εργαλείο αυτό χρησιμοποιεί τόσο πίνακες όσο και λίστες για τις δομές δεδομένων των αλγορίθμων, ούτως ώστε να γίνει έλεγχος, μετά από πειραματισμό, για την αποδοτικότερη δομή, βλέπε Κεφάλαιο 4.2 για τα αποτελέσματα του ελέγχου αυτού.

Για την διεξαγωγή των πειραμάτων εύρεσης ιδανικού ποσοστού X για ΚΕΑ υλοποιήθηκε ένα διαδραστικό (interactive) εργαλείο με το όνομα “initial”. Το εργαλείο υλοποιεί τον αλγόριθμο εύρεση ΚΕΑ από δύο πηγαίους εξυπηρετητές και έχει γραφτεί εξολοκλήρου σε C. Στην είσοδο δίνεται το αρχείο αιτήσεων, το κατώτερο όριο X και ο τύπος δομής δεδομένων που θα χρησιμοποιηθεί για την εκτέλεση του πειράματος (πίνακας ή λίστα). Μετά την εκτέλεση του αλγορίθμου από το εργαλείο αποθηκεύονται τα ΚΕΑ σε ένα αρχείο.

```
Started: file Traces.10000 with type Array
Subjects:1000 Objects:3830

Structure created in 0ms

Users: 1000 No of Sets: 2 Type: Array
Options
--==--==--
1) Run Algorithm 2 with X=80 and K=2
2) Change parameters X, K
3) Toggle show/hide messages
4) Clear screen
5) Open output file in gedit
6) Exit
Enter option:
```

Εικόνα 3-7 Οθόνη εκτέλεσης του εργαλείου initial

```

1 Pairs found for X=60, K=2 and Subjects=10 : 5
2 17.0 36.1
3 17.0 39.1
4 21.0 34.1
5 21.0 36.1
6 21.0 45.1

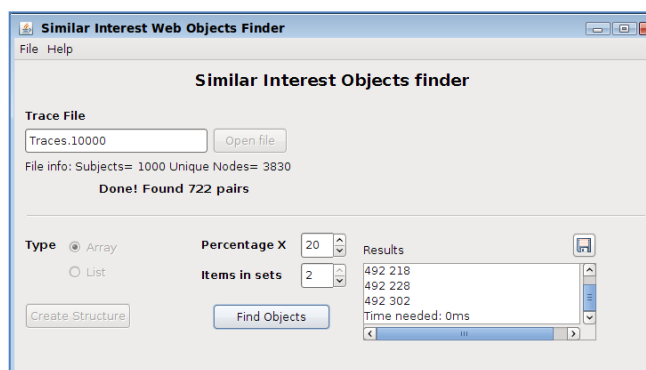
```

Εικόνα 3-8 Έξοδος εργαλείου initial – Αρχείο με τα ΚΕΑ

Στη συνέχεια, το εργαλείο αναβαθμίστηκε για να μπορεί να βρίσκει όλες τις κοινού ενδιαφέροντος k-άδες από αντικείμενα (επιλογή k πηγαίων εξυπηρετητών από τους συνολικούς n). Διαμέσου ενός μενού επιλογών, μπορεί κάποιος να επιλέξει να τρέξει τον αλγόριθμο CommIntKwo με το X και k που δόθηκαν στην είσοδο, να αλλάξει το X και k, να καθαρίσει την οθόνη, να ενεργοποιήσει κάποια επιπλέον μηνύματα και να βγει από το εργαλείο.

Το μέγεθος των αρχείων αιτήσεων διαφέρει ανάλογα με τον αριθμό των χρηστών για τους οποίους αφορά. Το initial είναι κατάλληλο για όλα τα μεγέθη αρχείων χωρίς περιορισμούς στη δέσμευση μνήμης.

Παράλληλα με το εργαλείο initial υλοποιήθηκε και μια έκδοση σε Java με γραφικό περιβάλλον για να ελεγχτούν και οι επιδόσεις του αλγορίθμου σε Java. Η έκδοση όμως αυτή είναι κατάλληλη μόνο για αρχεία με μικρό μέγεθος λόγω περιορισμούς στην μνήμη, τα οποία εκτελεί αρκετά γρήγορα. Ως αποτέλεσμα όμως οι επιδόσεις της Java ήταν πολύ χειρότερες από αυτές της C. Για παράδειγμα για ένα αρχείο αιτήσεων από τους ιστότοπους *Sciencedaily.com* και *Nasa.com*, και ποσοστό X ίσο με 10, στην έκδοση σε Java χρειάστηκε 2 χιλιοστά δευτερολέπτου για να εντοπίσει ΚΕΑ, ενώ στην έκδοση σε C χρειάστηκε μόλις 0,000000001 χιλιοστά δευτερολέπτων.



Εικόνα 3-9 Οθόνη εκτέλεσης του εργαλείου εύρεσης ΚΕΑ σε έκδοση Java

Το δεύτερο εργαλείο, βρίσκει τις ΚΕΚ από δύο διαφορετικούς πηγαίους εξυπηρετητές, ούτως ώστε να αντιγράφονται μαζί στους εξυπηρετητές αναπαραγωγής. Βασίζεται σε ένα αρχείο αιτήσεων, με αιτήσεις χρηστών από τους εξυπηρετητές αυτούς, δύο αρχεία με τις κοινότητες των ιστοσελίδων, ένα από τον κάθε πηγαίο εξυπηρετητή, το κατώτατο όριο για το ποσοστό κοινού ενδιαφέροντος (X) και τέλος το κατώτατο όριο (Y), το οποίο θα καθορίζει το ποσοστό κοινού ενδιαφέροντος των δυο κοινοτήτων που μας ενδιαφέρει. Αν δυο κοινότητες έχουν κοινό ενδιαφέρον μεγαλύτερο ή ίσο από αυτό το ποσοστό τότε θα θεωρούνται κοινού ενδιαφέροντος. Για τα πλαίσια της διπλωματικής μελετήσαμε το κοινό ενδιαφέρον μεταξύ δύο κοινοτήτων γιατί μας περιόριζε ο χρόνος, όμως πολύ εύκολα μπορεί να επεκταθεί και για k κοινότητες κάνοντας συνδυασμούς από ζεύγη των κοινοτήτων αυτών.

Για το εργαλείο αυτό, αρχικά υλοποιήθηκε ο αλγόριθμος κατασκευής μιας δομής που αποθηκεύει τις κοινότητες μαζί με τα αντικείμενα που ανήκουν στην κάθε μια, και έπειτα βάση του αρχείου αιτήσεων δημιουργεί την ίδια δομή που κάνει τη σύνδεση μεταξύ των χρηστών του δικτύου και των αντικειμένων που παρέχονται από τους πηγαίους εξυπηρετητές, όπως και στο προηγούμενο εργαλείο. Αφού κατασκευαστούν οι δομές αυτές τότε υπολογίζεται ο αριθμός των μοναδικών χρηστών που ζητούν αντικείμενα από τις δύο κοινότητες και βάση του αριθμού αυτού ταξινομούνται οι κοινότητες. Στη συνέχεια για την εύρεση κοινού ενδιαφέροντος κοινοτήτων βάση του κατώτατου ορίου Y χρησιμοποιήθηκε ο Similarity2wo. Ουσιαστικά θα βρει πόσοι χρήστες, από αυτούς που κάνουν αιτήσεις στο δίκτυο, ζητούν και τα δυο αντικείμενα ιστού (ένα από κάθε origin server). Όταν βρεθεί το συνολικό ποσοστό κοινού ενδιαφέροντος των αντικειμένων από όλες τις συγκρίσιμες κοινότητες, και είναι μεγαλύτερο ή ίσο με το Y οι κοινότητες θεωρούνται κοινού ενδιαφέροντος.

Για τα πειράματα εύρεσης ιδανικού ποσοστού Y , για τις ΚΕΚ, υλοποιήθηκε το δεύτερο εργαλείο με το όνομα “communities”. Το εργαλείο αυτό υλοποιεί τους αλγορίθμους **CommInt2wc** και **AssignWCtoSurr**. Είναι ένα εκτελέσιμο αρχείο γραμμένο σε C το οποίο παίρνει σαν είσοδο το αρχείο αιτήσεων με αιτήσεις από δύο πηγαίους εξυπηρετητές, τα αρχεία κοινοτήτων των ιστοσελίδων που ανήκουν στους δύο πηγαίους εξυπηρετητές, το ποσοστό X , το ποσοστό συγκρισιμότητας δύο κοινοτήτων C και το ποσοστό Y . Μετά την εκτέλεση του το εργαλείο βρίσκει τις ΚΕΚ βάση του Y και, αν

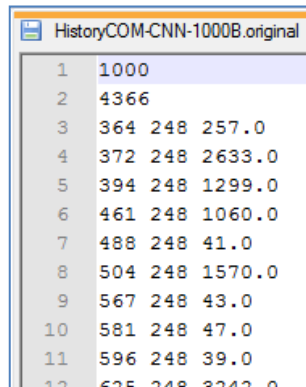
υπάρχουν, τις τοποθετεί μαζί στους ίδιους εξυπηρετητές αναπαραγωγής στα αρχεία τοποθέτησης. Αν δεν υπάρχουν απλά τοποθετεί τις κοινότητες τυχαία σε αρχεία τοποθέτησης. Χρησιμοποιήθηκε δομή πινάκων για την αποθήκευση των αντικειμένων και των χρηστών στο εργαλείο αυτό επειδή έκανε τον προγραμματισμό πιο εύκολο και επηρεάζει ελάχιστα τον χρόνο εκτέλεσης του εργαλείου αυτού. Για περισσότερα βλέπε κεφάλαιο 4.2.

Για την γρήγορη εξαγωγή αποτελεσμάτων από τα πειράματα, έχει υλοποιηθεί μια σειρά από τρία εργαλεία τα οποία παράγουν αποτελέσματα για συγκεκριμένα X, C και Y. Το πρώτο είναι το *experimentsC.pl* όπου δημιουργεί πειράματα με συγκεκριμένες τιμές του C (0.25, 0.50, 0.75), με την βοήθεια των επόμενων εργαλείων, για κάποιο φάκελο που περιέχει τα απαραίτητα αρχεία (αιτήσεων, κοινοτήτων, αντιστοίχισης, γράφου με μέγεθος αντικειμένων) και για συγκεκριμένο αριθμό εξυπηρετητών αναπαραγωγής. (Περισσότερες πληροφορίες για τα αρχεία στο τέλος αυτού του υποκεφαλαίου). Το εργαλείο αυτό χρησιμοποιεί το *experimentsY.pl*, το οποίο δημιουργεί πειράματα με συγκεκριμένες τιμές του Y (0.03, 0.05, 0.15, 0.25, 0.35, 0.45, 0.55), δηλαδή παίρνει στη είσοδο του τιμές C από το προηγούμενο εργαλείο και επιπλέον τον αριθμό των εξυπηρετητών αναπαραγωγής. Αυτό με την σειρά του εκτελεί το *experimentsX.pl* δίνοντας του στην είσοδο του τις τιμές Y και δημιουργεί πειράματα για συγκεκριμένες τιμές του X (1, 5, 10, 25, 40, 50, 75, 90). Όλα μαζί τρέχουν αλυσιδωτά και τέλος το *experimentsX.pl* τρέχει το εργαλείο *communities* για όλα τις τιμές X, C και Y.

Για κάθε πείραμα που ολοκληρώνεται αποθηκεύονται τα αποτελέσματα (*zmin*, *zmax*, *no of pairs*) σε ένα αρχείο και έπειτα εκτελείται το script *doAll.pl* για να ετοιμάσει τους φακέλους με τα πειράματα για την ευκολότερη εισαγωγή τους στον CDNsim. Πρώτα συνενώνει τα αρχεία γράφων με μεγέθη. Έπειτα μεταφράζει τα αρχεία τοποθέτησης στην τελική τους μορφή όπως επίσης και το αρχείο αιτήσεων. Τέλος τακτοποιεί τον φάκελο (αντιγράφοντας και αφαιρώντας τα αχρείαστα) και τον ετοιμάζει για την εισαγωγή των αρχείων στον CDNsim.

Τα επιπλέον αρχεία τα οποία περιέχουν δεδομένα για τα πειράματά μας είναι τα εξής:

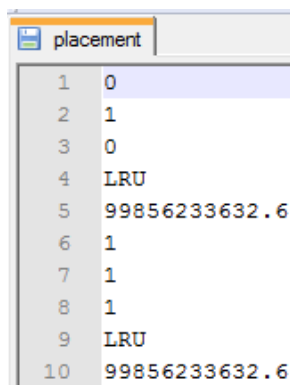
- Το **Αρχικό αρχείο αιτήσεων** (original trace file) περιέχει τις πραγματικές ταυτότητες των αντικειμένων και έχει ακριβώς την ίδια μορφή με το αρχείο αιτήσεων.



1	1000
2	4366
3	364 248 257.0
4	372 248 2633.0
5	394 248 1299.0
6	461 248 1060.0
7	488 248 41.0
8	504 248 1570.0
9	567 248 43.0
10	581 248 47.0
11	596 248 39.0
12	625 248 3242.0

Εικόνα 3-10 Ένα αρχικό αρχείο αιτήσεων

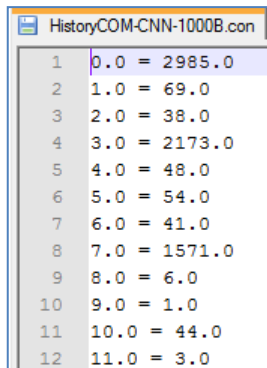
- Το **Αρχείο Τοποθέτησης** (placement file) περιέχει πληροφορίες για το προσομοιωτή CDNsim για το που θα τοποθετούνται οι κοινότητες κάποιων ιστότοπων.



1	0
2	1
3	0
4	LRU
5	99856233632.6
6	1
7	1
8	1
9	LRU
10	99856233632.6

Εικόνα 3-11 Ένα αρχείο τοποθέτησης

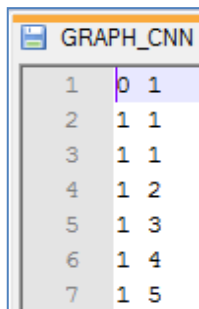
- Το **Αρχείο Αντιστοίχισης** (connection file) περιέχει γραμμές με την συνδεσιμότητα μεταξύ των αντικειμένων. Ουσιαστικά είναι σαν ένας πίνακας αντιστοίχισης που χρειάζεται για την μετάφραση των αντικειμένων των αρχείων αιτήσεων, τοποθέτησης και κοινότητας. Η μορφή τους είναι <Ταυτότητα αντικειμένου = Ταυτότητα μεταφρασμένου αντικειμένου>. Έχουν κατάληξη .con.



1	0.0 = 2985.0
2	1.0 = 69.0
3	2.0 = 38.0
4	3.0 = 2173.0
5	4.0 = 48.0
6	5.0 = 54.0
7	6.0 = 41.0
8	7.0 = 1571.0
9	8.0 = 6.0
10	9.0 = 1.0
11	10.0 = 44.0
12	11.0 = 3.0

Εικόνα 3-12 Ένα αρχείο αντιστοίχισης

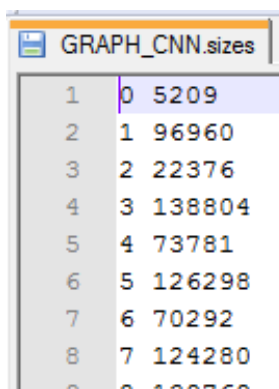
- Το **Αρχείο Γράφου** (graph file), περιέχει σε κάθε γραμμή την σύνδεση μεταξύ των αντικειμένων του γράφου. (Κόμβος <κενό> Κομβος)



1	0	1
2	1	1
3	1	1
4	1	2
5	1	3
6	1	4
7	1	5

Εικόνα 3-13 Ένα αρχείο γράφου

- Το **Αρχείο Γράφου με μέγεθος αντικειμένων** (.sizes file). Περιέχει σε κάθε γραμμή το αντικείμενο και το μέγεθος που έχει σε bytes.



1	0	5209
2	1	96960
3	2	22376
4	3	138804
5	4	73781
6	5	126298
7	6	70292
8	7	124280

Εικόνα 3-14 Ένα αρχείο γράφου με μέγεθος αντικειμένων

Comm1.size.clu.final.clu									
1	2								
2	584.0	197.0	191.0	433.0	4366.0	1217.0	193.0	653	
3	454.0	584.0	889.0	567.0	1417.0	1476.0	29.0	197.	

Εικόνα 3-15 Ένα αρχείο κοινότητας

3.3.3 Βοηθητικά Εργαλεία

Εκτός από τα εργαλεία που υλοποιούν τους αλγόριθμους, υλοποιήθηκαν και κάποια βοηθητικά εργαλεία για την ευκολότερη διεξαγωγή των πειραμάτων αλλά και για άλλους σκοπούς.

Ένα παράδειγμα είναι το εκτελέσιμο αρχείο “initial2”, το οποίο βρίσκει το μέγιστο ποσοστό X που μπορεί να υπολογιστεί από ένα αρχείο αιτήσεων από n πηγαίους εξυπηρετητές, είτε με επιλογή k πηγαίων εξυπηρετητών από τους n . Ουσιαστικά, λειτουργεί όπως ακριβώς το εργαλείο initial, χωρίς να είναι διαδραστικό και γράφει τα αποτελέσματα σε ένα αρχείο.

Το εργαλείο *TranslateSIWOs.pl* μεταφράζει τα αποτελέσματα του εργαλείου initial τα οποία είναι ένα αρχείο με KEA, σε αρχικές ταυτότητες γράφου (βλέπε Κεφάλαιο 3.1). Χρησιμοποιείται σε περιπτώσεις που επιθυμεί κάποιος να μελετήσει αν τα αποτελέσματα έχουν πράγματι κοινό ενδιαφέρον, από την πραγματική διεύθυνση των αντικειμένων στο διαδίκτυο.

3.3.4 Προβλήματα κατά την υλοποίηση

Κατά την διάρκεια της υλοποίησης αντιμετωπιστήκαν κάποια προβλήματα όσον αφορά την δημιουργία των διαφόρων αρχείων και την αντιστοιχία και σωστή μετάφραση των ταυτοτήτων των αντικειμένων μέσα στα αρχεία αιτήσεων και κοινοτήτων.

Κατά την δημιουργία των αρχείων αιτήσεων, παρουσιάστηκε το πρόβλημα ότι οι χρήστες που έκαναν αιτήσεις μέσα στο αρχείο αιτήσεων ήταν πολύ περισσότεροι από τα αντικείμενα. Αυτό είχε σαν συνέπεια να μην αντιστοιχούν αρκετά αντικείμενα στον κάθε χρήστη και ως επόμενο να δημιουργούνταν αρχεία αιτήσεων χωρίς πολλά KEA.

Για την επίλυση του προβλήματος εξετάζουμε τις περιπτώσεις που οι χρήστες ζητούν τα περισσότερα αντικείμενα κάποια χρονική στιγμή, δηλαδή σε μια σύνοδο. Συγκεκριμένα ταξινομήθηκαν τα αρχεία συνόδων βάση τον αριθμό αντικειμένων σε κάθε γραμμή πριν εισαχθούν στο εργαλείο συνένωσης. Επίσης αυτός είναι και ένας λόγος που δημιουργήθηκε το εργαλείο `merge2.pl` το οποίο κατά τη φάση της συνένωσης επιλέγει συνόδους από όλα τα αρχεία συνόδων που δόθηκαν στην είσοδο. Η απόφαση αυτή είχε καλά αποτελέσματα, αφού δημιουργούνται συνενωμένοι σύνοδοι στο τελικό αρχείο με πολύ περισσότερα αντικείμενα και έτσι επιλύθηκε το πρόβλημα αυτό.

Επίσης, κατά την δημιουργία των αρχείων αιτήσεων χρησιμοποιούνται ταυτότητες με αριθμούς σε αύξουσα σειρά ξεκινώντας από το 0. Για τον λόγο αυτό πρέπει να υπάρχει συνέπεια μεταξύ των αρχείων που συσχετίζονται με τα αρχεία αιτήσεων στα εργαλεία. Έτσι πρέπει να μεταφράζονται τα αρχεία ενδιάμεσα της εκτέλεσης των εργαλείων. Δηλαδή πριν τοποθετηθούν τα αρχεία κοινοτήτων στο εργαλείο εύρεσης ΚΕΚ πρέπει να μεταφραστούν από τις αρχικές ταυτότητες αντικειμένων σε αυτές των αρχείων αιτήσεων για να συμβαδίζουν. Για την επίλυση του προβλήματος αυτού, με την δημιουργία των αρχείων αιτήσεων παράγονται και τα αρχεία αντιστοιχίας (βλέπε Εικόνα 3–6). Τα αρχεία αυτά χρησιμοποιούνται για την μετάφραση περαιτέρω αρχείων. Αρχικά δημιουργήθηκαν κάποια εργαλεία τα οποία έκτιζαν μια δομή πίνακα με δύο στήλες από το αρχείο αντιστοιχίας. Στη πρώτη στήλη αποθηκευόταν η αρχική ταυτότητα του αντικειμένου και στην δεύτερη στήλη στην αντίστοιχη σειρά η μεταφρασμένη ταυτότητα. Το μειονέκτημα με την δομή αυτή ήταν ότι για την μετάφραση έπρεπε να γίνει αναζήτηση στο πίνακα για την εύρεση της αντίστοιχης ταυτότητας, κάνοντας την εκτέλεση του εργαλείου αρκετά αργή. Στη συνέχεια στα εργαλεία αυτά αλλάχτηκε η δομή του πίνακα σε πίνακα Hash. Χρησιμοποιήθηκε η αρχική ταυτότητα σαν το κλειδί του πίνακα και η τιμή που αποθηκεύεται στην θέση αυτή είναι η μεταφρασμένη ταυτότητα. Με την δομή αυτή η αναζήτηση για την ταυτότητα που θα χρησιμοποιηθεί για την μετάφραση γίνεται σε μοναδιαίο χρόνο γι' αυτό και αναμενόταν να γίνει το εργαλείο αρκετά πιο γρήγορο. Η πολιτική που χρησιμοποιήθηκε ήταν σωστή αφού όντως έκανε το εργαλείο πάρα πολύ γρήγορο, και εκτελείται σε σχεδόν μοναδιαίο χρόνο.

Κεφάλαιο 4

Πειραματική Αξιολόγηση

4.1	Γενική Περιγραφή	27
4.2	Πειράματα Εύρεσης Ιδανικού Ποσοστού X	27
4.3	Πειράματα Εύρεσης Ιδανικού Ποσοστού Y	31

4.1 Γενική Περιγραφή

Μετά από την υλοποίηση των εργαλείων πρέπει να γίνουν τα πειράματα εύρεσης των ιδανικών ποσοστών X και Y. Γενικά χρησιμοποιώντας την Γεννήτρια Γράφων Ιστοσελίδων ανακτήθηκαν οι γράφοι των ιστότοπων *BBC.com*, *CNN.com*, *Yahoonews.com*, που έχουν κοινό θέμα τα νέα της επικαιρότητας, *Nasa.com*, *Nationalgeographic.com*, *History.com*, *Sciencedaily.com*, που έχουν κοινό θέμα την επιστήμη και περιβάλλον και τέλος *MSNBC.com*, *Foxsports.com* και *Sportingnews.com* που έχουν κοινό θέμα τα νέα των σπορ. Από αυτούς δημιουργήθηκαν αιτήσεις, με την χρήση του εργαλείου συνένωσης, από ζεύγη ιστότοπων για να χρησιμοποιηθούν στα πειράματα. Τα ζεύγη που δημιουργήθηκαν περιγράφονται στα επόμενα κεφάλαια 4.2 και 4.3.

4.2 Πειράματα Εύρεσης Ιδανικού Ποσοστού X

Για την εύρεση ενός ιδανικού ποσοστού X μελετήθηκαν πειράματα για 1000 και 10000 χρήστες σε ζεύγη ιστότοπων τα οποία είχαν ίδιο σημασιολογικό περιεχόμενο και σε ζεύγη που δεν είχαν, ούτως ώστε να μελετηθεί και αυτή η περίπτωση. Χρησιμοποιήθηκε το εργαλείο *initial2* το οποίο βρίσκει το μεγαλύτερο ποσοστό σε ένα σενάριο όπως επίσης και τον αριθμό των ΚΕΑ που βρέθηκαν. Όταν εκτελέστηκαν τα πειράματα για όλα τα σενάρια, τότε βρέθηκε το ελάχιστο και μέγιστο X καθώς και μια μέση τιμή. Μελετήθηκε ξεχωριστά η συνένωση τόσο με επιλογή και από τα δύο αρχεία

συνόδων (Both transaction file selection) όσο και με τυχαία επιλογή αρχείων (Random transaction file selection) για να μελετηθεί και μια πιο ρεαλιστική περίπτωση. Πιο κάτω φαίνονται οι πίνακες των αποτελεσμάτων από κάποια πειράματα που έχουν γίνει. Πιο κάτω παρουσιάζονται 44 πειράματα και τα στοιχεία των αποτελεσμάτων τους. Τα στοιχεία αυτά είναι ο αριθμός των χρηστών, το μέγιστο ποσοστό, ο αριθμός των κοινού ενδιαφέροντος ζευγών που βρέθηκαν για το ποσοστό αυτό, ο αριθμός των μοναδικών αντικειμένων που συμπεριλήφθησαν στο πείραμα και τέλος ο τρόπος συνένωσης των συνόδων.

BBC - CNN

Users	Max percentage	Pairs	Number of Objects	Merging
1000	24%	1	5560	Both
10000	12%	2	16794	Both
1000	10%	2	4839	Random
10000	7%	1	14567	Random

CNN – Yahoo News

Users	Max percentage	Pairs	Number of Objects	Merging
1000	26%	1	6620	Both
10000	15%	1	25072	Both
1000	13%	1	5589	Random
10000	8%	1	21416	Random

National Geographic - History.com

Users	Max percentage	Pairs	Number of Objects	Merging
1000	23%	2	4694	Both
10000	17%	1	8636	Both
1000	11%	1	4279	Random
10000	9%	1	8154	Random

Fox Sport News – Sporting News

Users	Max percentage	Pairs	Number of Objects	Merging
1000	10%	1	6529	Both
10000	6%	2	26414	Both
1000	5%	2	5252	Random
10000	3%	5	22590	Random

BBC - MSNBC

Users	Max percentage	Pairs	Number of Objects	Merging
1000	35%	2	5389	Both
10000	24%	2	10436	Both
1000	18%	2	4851	Random
10000	12%	4	9882	Random

Science Daily - NASA

Users	Max percentage	Pairs	Number of Objects	Merging
1000	26%	1	5221	Both
10000	20%	1	11761	Both
1000	13%	1	4519	Random
10000	10%	1	11038	Random

Sporting News - MSNBC

Users	Max percentage	Pairs	Number of Objects	Merging
1000	36%	1	5636	Both
10000	28%	1	20150	Both
1000	19%	2	4747	Random
10000	14%	3	17525	Random

Yahoo News - BBC

Users	Max percentage	Pairs	Number of Objects	Merging
1000	20%	1	6880	Both
10000	11%	2	17958	Both
1000	10%	1	6036	Random
10000	6%	1	16140	Random

NASA - CNN

Users	Max percentage	Pairs	Number of Objects	Merging
1000	53%	1	4557	Both
10000	36%	1	16280	Both
1000	28%	1	3952	Random
10000	19%	1	13922	Random

Sporting News - NASA

Users	Max percentage	Pairs	Number of Objects	Merging
1000	41%	1	5064	Both
10000	33%	1	18880	Both
1000	22%	1	4220	Random
10000	17%	1	16369	Random

Yahoo News - NASA

Users	Max percentage	Pairs	Number of Objects	Merging
1000	45%	1	5877	Both
10000	35%	1	17444	Both
1000	22%	1	5054	Random
10000	17%	1	15516	Random

Από τα αποτελέσματα γενικά βρέθηκε ο μέσος όρος των Both (**26%**) και Random (**13%**) ξεχωριστά και έπειτα ο γενικός μέσος όρος (**20%**). Επίσης το μικρότερο ποσοστό που βρέθηκε είναι **3%** ενώ το μεγαλύτερο **53%**.

Επίσης κατά την διάρκεια των πειραμάτων ελέγχτηκε και ο χρόνος εκτέλεσης του εργαλείου χρησιμοποιώντας διαφορετική δομή, μεταξύ πίνακα και λίστας. Αναμέναμε ότι όταν η σχέση μεταξύ αντικειμένων και χρηστών είναι μεγάλη, δηλαδή πολλοί χρήστες ζητούν πολλά αντικείμενα, τότε ο πίνακας θα είναι πιο αποδοτικός από ότι οι λίστες. Αυτό οφείλεται στο ότι ο πίνακας μπορεί να προσπελαστεί σε μοναδιαίο χρόνο κατά την αναζήτηση σε αυτόν, αντίθετα με την λίστα στην οποία η αναζήτηση παίρνει χρόνο όσοι και οι χρήστες για κάποιο αντικείμενο. Επιπλέον, ο χώρος που χρειάζεται ένας πίνακας είναι πολύ περισσότερος από όσο η λίστα, έτσι δεν πάει χαμένος με μεγάλη σχέση αντικειμένων-χρηστών. Ανάλογα, αν η σχέση αυτή είναι μικρή, η

αναζήτηση γίνεται πιο γρήγορα και δεν γίνεται σπατάλη χώρου από την λίστα. Μετά από την διεξαγωγή των πειράματα τα αποτελέσματα δεν ήταν ακριβώς τα αναμενόμενα. Σε όλα τα αρχεία αιτήσεων η σχέση χρηστών και αντικειμένων ήταν μικρή. Η σχέση αυτή παρατηρήθηκε διαιρώντας τις συνολικές αιτήσεις του αρχείου με τις συνολικές δυνατές αιτήσεις που μπορούν να γίνουν, δηλαδή κάθε χρήστης να ζητά όλα τα αντικείμενα που υπάρχουν. Για καθένα από τα 44 πειράματα, εκτελέστηκε η εύρεση ΚΕΑ με ποσοστό X ίσο με 10. Σε κάποιες περιπτώσεις η δομή με πίνακες ήταν καλύτερη όσον αφορά τον χρόνο εκτέλεσης και σε κάποιες άλλες η λίστα. Δεν είναι ακριβώς ξεκάθαρο ποια από τις δύο δομές είναι η καλύτερη και η διαφορά στους χρόνους εκτέλεσης τους δεν είναι και πολύ μεγάλη. Για παράδειγμα, η σχέση μεταξύ του αρχείου αιτήσεων από το γράφο ιστότοπων YahooNews.com και NASA.com με 10000 χρήστες είναι μόλις 0.190% και ο χρόνος που χρειάστηκε για την εύρεση των ΚΕΑ με πίνακα ήταν 1.478 δευτερόλεπτα, ενώ με την λίστα 2.6128 δευτερόλεπτα. Επίσης για το αρχείο αιτήσεων από το γράφο ιστότοπων YahooNews.com και BBC.com με 10000 χρήστες, έχει σχέση χρηστών-αντικειμένων 0.149% και ο χρόνος εκτέλεσης με πίνακες ήταν 1.3209 δευτερόλεπτα και με λίστα ήταν 0.7016 δευτερόλεπτα.

4.3 Πειράματα Εύρεσης Ιδανικού Ποσοστού Y

Για τα πειράματα Εύρεσης Ιδανικού Ποσοστού Y χρησιμοποιήθηκε παρόμοια στρατηγική. Με τα εργαλεία `experimentC`, `experimentY` και `experimentX` δημιουργήθηκαν 216 πειράματα για κάθε αριθμό χρηστών και πάνω σε κάθε ζεύγος ιστότοπων. Επιλέχθηκαν τα αποτελέσματα με τα περισσότερα ζεύγη ΚΕΚ και για εκείνα ελέγχθηκαν οι επιδόσεις τους στο δίκτυο με τον προσομοιωτή. Δυστυχώς από όλους τους γράφους ιστότοπων μόνο για τους BBC.com, CNN.com, YahooNews.com, Nasa.com, Nationalgeographic.com και History.com είχαν δημιουργηθεί σωστά και ολοκληρωμένα αρχεία με κοινότητες, αρκετά όμως για τα πειράματα που διεξήχθησαν. Αυτό οφείλεται στο ότι κάποια από τα αρχεία γράφου ιστότοπων είναι μεγάλα (πάνω από 30MB) και το εργαλείο που υλοποιεί τον αλγόριθμο CiBC χρειάζεται πολύ χρόνο για να τερματίσει. Συγκεκριμένα για την δημιουργία αρχείου κοινότητας για το γράφο ιστότοπου του Sportingnews.com είχε περάσει σχεδόν ένας μήνας και δεν είχε τερματίσει η εκτέλεση. Για κάθε ζεύγος ιστότοπων έχουν δημιουργηθεί πειράματα που

αφορούν 1000 και 5000 χρήστες. Ωστόσο, τα ΔΠΠ είναι μεγάλης κλίμακας συστήματα (Large Scale Systems) και για να δούμε τις επιδόσεις του σε πραγματικό επίπεδο χρειάζονται 500000 χρήστες και περισσότεροι. Για τα πειράματα όμως, πάνω από 5000 χρήστες χρειαζόταν αρκετή υπολογιστική δύναμη για να δημιουργηθούν τα αποτελέσματα γιατί όσο αυξάνονται οι χρήστες αυξάνεται και ο αριθμός των αιτήσεων μέσα σε ένα αρχείο. Έγιναν δοκιμές για κάποια ζεύγη με 10000 χρήστες αλλά αντιμετωπίστηκαν προβλήματα στο ότι πάγωναν οι υπολογιστές, λόγω του ότι χρειάζεται υψηλή υπολογιστική δύναμη και αρκετή μνήμη και έτσι τα πειράματα περιορίστηκαν μέχρι τους 5000 χρήστες που και πάλι έδωσαν αποτελέσματα.

Τα πειράματα διεξήχθησαν με το εργαλείο communities το οποίο καταγράφει τα αποτελέσματα σε κάποιο αρχείο και έτσι ήταν εύκολη η διαχείριση τους. Συγκεκριμένα για κάθε πείραμα καταγράφηκε το ελάχιστο και μέγιστο Y και ο αριθμός των κοινού ενδιαφέροντος κοινοτήτων που βρέθηκαν για την τριάδα X , C και Y του πειράματος. Έπειτα, για κάποια πειράματα με τις περισσότερες ΚΕΚ δημιουργήθηκαν τα μπουκάλια (bottles) με το προσομοιωτή CDNSim και παράχθηκαν τα στατιστικά.

Οι ρυθμίσεις που τοποθετήθηκαν στο προσομοιωτή για το κάθε βήμα είναι οι εξής:

- 1) Στο πρώτο βήμα επιλέχθηκε η πολιτική ανακατεύθυνσης του ΔΠΠ ως η Surrogate Load Balance γιατί θεωρείται καλύτερη από τις άλλες πολιτικές και επίσης δεν επηρεάζει τα πειράματα μας, αφού επιθυμείται ο πειραματισμός για το ποσοστό Y . Η πολιτική αυτή ανακατευθύνει την αίτηση του χρήστη στον κοντινότερο εξυπηρετητή αναπαραγωγής. Αν αυτός είναι 95% φορτωμένος (loaded) τότε κατευθύνεται στον λιγότερα φορτωμένο. Αν δεν υπάρχει το αντικείμενο που αιτείται στον εξυπηρετητή αναπαραγωγής τότε αυτός το ανακτά από τον κοντινότερο εξυπηρετητή αναπαραγωγής. Αν και αυτός είναι 95% φορτωμένος η αίτηση κατευθύνεται στον λιγότερο φορτωμένο εξυπηρετητή αναπαραγωγής που το περιέχει. Αν το αντικείμενο δεν έχει αντιγραφεί σε κανένα εξυπηρετητή αναπαραγωγής τότε ο εξυπηρετητή αναπαραγωγής στον οποίο είχε γίνει αρχικά η αίτηση το αντιγράφει από τον κοντινότερο πηγαίο εξυπηρετητή που δεν είναι 95% φορτωμένος.
- 2) Στο δεύτερο βήμα επιλέγεται η τοπολογία του δικτύου, η οποία είναι σταθερή για όλα τα πειράματα. Η τοπολογία αυτή αποτελείται από 3037 κόμβους

τοποθετημένους σε ένα αυτόνομο σύστημα (Autonomous System). Οι υπόλοιπες ρυθμίσεις είναι και αυτές σταθερές για όλα τα πειράματα, όπως φαίνεται στον πιο κάτω πίνακα:

Περιγραφή ρύθμισης	Τιμή
Link speed	1000 Mbits/sec
Number of clients	Ανάλογα με το πείραμα - 1000 ή 5000
Clients' number of outgoing connections	1000
Clients' number of retries	10
Clients' mean waiting time per retry	5
Number of surrogate servers	Ανάλογα με το πείραμα – 50 ή 100
Surrogate servers' number of outgoing connections	1000
Surrogate servers' number of incoming connections	1000
Number of origin servers	2
Origin servers' number of incoming connections	1000

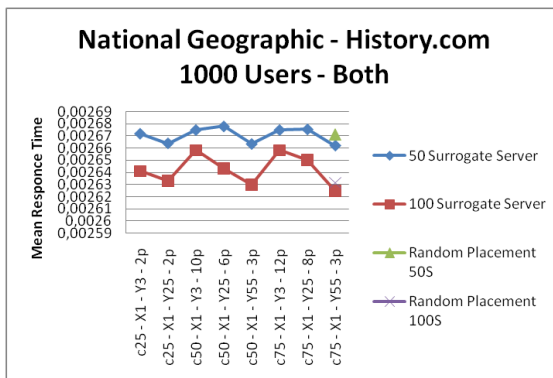
Πίνακας 4-1 – Ρυθμίσεις στο προσομοιωτή για το Βήμα 2 [7]

- 3) Στο τρίτο βήμα τοποθετούνται το αρχείο γράφου ιστότοπου και το αρχείο αιτήσεων ανάλογα με τα πείραμα. Το αρχείο γράφου ιστότοπου είναι ουσιαστικά αρχείο με μεγέθη των αντικειμένων του γράφου ιστότοπου. Επειδή χρησιμοποιείται ρύθμιση με 2 πηγαίων εξυπηρετητών στο βήμα 2, πρέπει να τοποθετηθεί ένα αρχείο που αφορά δύο ιστότοπους. Το τι έχει γίνει εδώ είναι η συνένωση 2 τέτοιων αρχείων και μετάφραση τους για να είναι συνεπείς με τα υπόλοιπα αρχεία που θα χρησιμοποιηθούν για το πείραμα.
- 4) Στο τέταρτο βήμα τοποθετείται το αρχείο τοποθέτησης των κοινοτήτων στους εξυπηρετητές αναπαραγωγής του σεναρίου. Και πάλι εδώ το αρχείο είναι ανάλογα με το πείραμα.
- 5) Στο πέμπτο βήμα τοποθετείται το μονοπάτι που πρόκειται να αποθηκευτεί το μπουκάλι που θα δημιουργηθεί και το όνομα που θα δοθεί στο μπουκάλι.

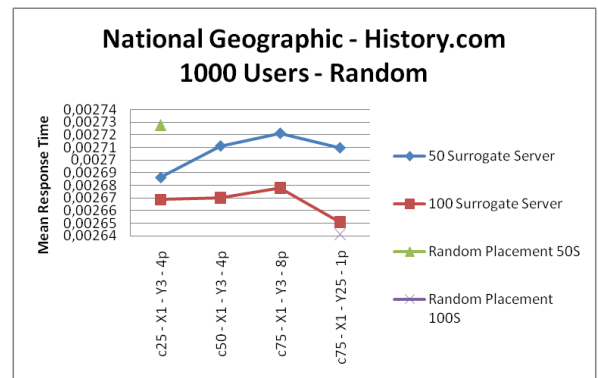
Τα στατιστικά που παράγονται μετά την δημιουργία των μπουκαλιών, αποτελούνται από τον αριθμό των επιτυχημένων αιτήσεων (Completed Requests), τον μέσο χρόνο ανταπόκρισης (Mean Response Time), το ποσοστό ύπαρξης κάποιου αντικειμένου στον εξυπηρετητή αναπαραγωγής (Hit Ratio Percentage), τον αριθμό των συνολικών ενεργών εξυπηρετητών αναπαραγωγής (Active Surrogate Servers) και τέλος την μέση χρησιμότητα εξυπηρετητών αναπαραγωγής (Mean Surrogate Servers Utility).

Ακολουθούν οι γραφικές παραστάσεις με τα αποτελέσματα από τα πειράματα. Μελετούνται τα αποτελέσματα στους 1000 και 5000 χρήστες για μέθοδο συνένωσης συνόδων με επιλογή τυχαίων αρχείων συνόδων και όλων των αρχείων ξεχωριστά και σε σενάριο με 50 και 100 εξυπηρετητές αναπαραγωγής. Σε κάθε γραφική παράσταση παρατηρούμε στον άξονα Y τον μέσο χρόνο ανταπόκρισης και στον άξονα X τις τριάδες X, C και Y που χρησιμοποιήθηκαν για το πείραμα, καθώς επίσης και για τον αριθμό των ζευγαριών που βρέθηκαν για τις τριάδες αυτές. Τέλος παρατηρούμε δυο επιπλέον τιμές για 50 και 100 εξυπηρετητές αναπαραγωγής, για το σενάριο που δεν έχουν τοποθετηθεί κάποια αρχεία τοποθέτησης και οι εξυπηρετητές αναπαραγωγής αναγκάζονται να αντιγράψουν από μόνοι τους στην μνήμη τους ιστότοπους που χρειάζονται. Να σημειωθεί ότι τα αποτελέσματα τα οποία παρατηρούμε είναι αυτά για τα οποία βρέθηκαν ΚΕΚ.

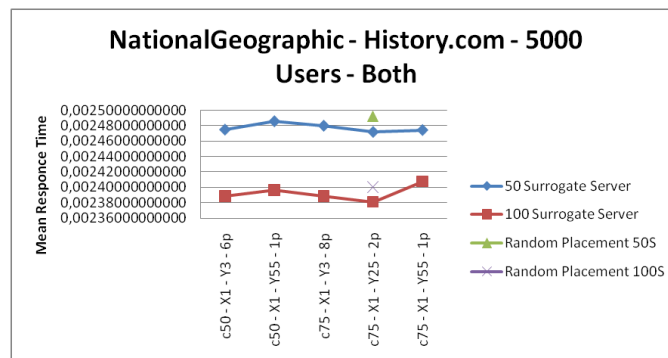
National Geographic – History.com



Σχήμα 4-1 Γραφική παράσταση National Geographic και History.com για 1000 χρήστες και Both μέθοδο συνένωσης



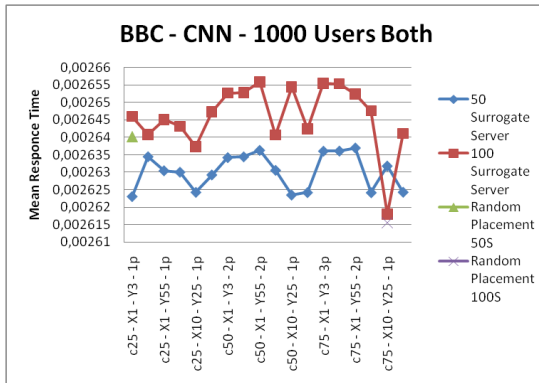
Σχήμα 4-2 Γραφική παράσταση National Geographic και History.com για 1000 χρήστες και Random μέθοδο συνένωσης



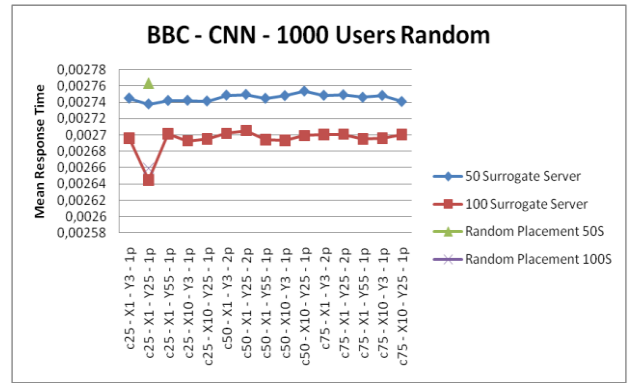
Σχήμα 4-3 Γραφική παράσταση National Geographic και History.com για 5000 χρήστες και Both μέθοδο συνένωσης

Στα πειράματα αυτά παρατηρούμε ότι με 100 εξυπηρετητές αναπαραγωγής ο χρόνος ανταπόκρισης ήταν καλύτερος από το χρόνο με 50 εξυπηρετητές, αν και η διαφορά μεταξύ τους είναι μηδαμινή αφού επηρεάζεται μόνο το πέμπτο δεκαδικό ψηφίο. Ο χρόνος ανταπόκρισης μπορεί να επηρεάζεται μονό στο πέμπτο δεκαδικό ψηφίο αλλά για εκατομμύρια αιτήσεις θα κάνει την διαφορά στο χρόνο. Σε όλες τις περιπτώσεις εκτός του σχήματος 4-2, η τυχαία τοποθέτηση αντικειμένων στους εξυπηρετητές αναπαραγωγής ήταν χειρότερη από την τοποθέτηση ΚΕΚ στους εξυπηρετητές και πάλι στην περίπτωση αυτή η διαφορά δεν είναι ούτε στο πέμπτο δεκαδικό ψηφίο. Στο σχήμα 4-1 παρατηρούνται κάποιες κορυφές για τους 100 εξυπηρετητές αναπαραγωγής στο χρόνο 0,00266 περίπου. Στα σημεία αυτά παρατηρούμε το X και Y είναι πολύ χαμηλά και έτσι βρέθηκαν πολλά ζεύγη ΚΕΚ (10 και 12), γεγονός που τελικά δεν δείχνει να έχει καλύτερα αποτελέσματα, γιατί οι κοινότητες αυτές έχουν πολύ χαμηλό ποσοστό κοινού ενδιαφέροντος.

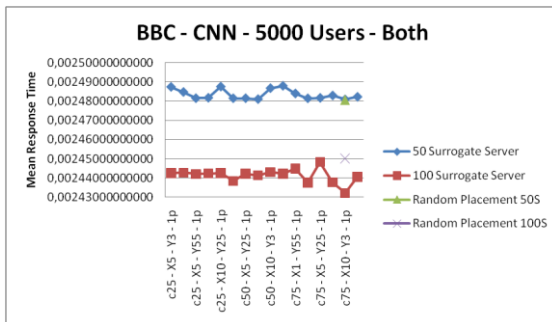
BBC - CNN



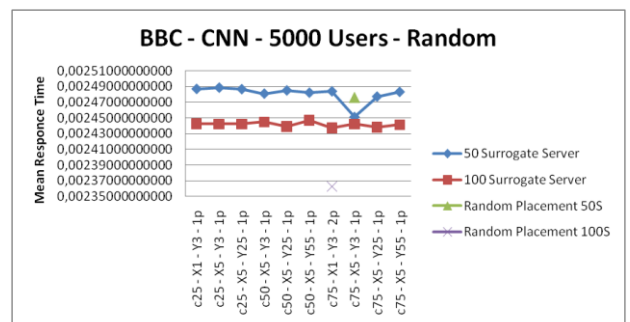
Σχήμα 4-4 Γραφική παράσταση BBC και CNN για 1000 χρήστες και Both μέθοδο συνένωσης



Σχήμα 4-5 Γραφική παράσταση BBC και CNN για 1000 χρήστες και Random μέθοδο συνένωσης



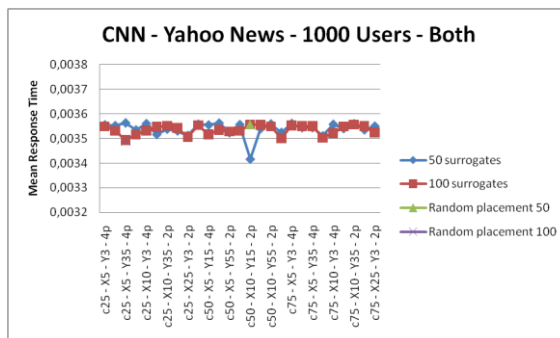
Σχήμα 4-6 Γραφική παράσταση BBC και CNN για 5000 χρήστες και Both μέθοδο συνένωσης



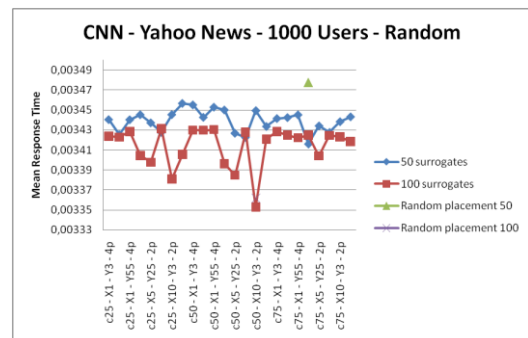
Σχήμα 4-7 Γραφική παράσταση BBC και CNN για 5000 χρήστες και Random μέθοδο συνένωσης

Στα πειράματα αυτά, παρατηρούμε τα ίδια φαινόμενα με τα πειράματα για National Geographic και History.com. Στο σχήμα όμως 4-4 με 50 εξυπηρετητές αναπαραγωγής ο χρόνος ανταπόκρισης ήταν καλύτερος από αυτόν με 100 εξυπηρετητές. Παρόλα αυτά τον καλύτερο χρόνο του σχήματος έχουν οι 100 εξυπηρετητές με $X=10\%$ και $Y=25\%$. Εκτός από τις κορυφές παρατηρούμε και κάποιες κοιλάδες αυτή τη φορά, όπως το σχήμα 4-5 στο χρόνο 0,00264 από τις οποίες δεν μπορούμε να βγάλουμε σίγουρα συμπεράσματα και μπορεί να είναι τυχαίες.

CNN – Yahoo News



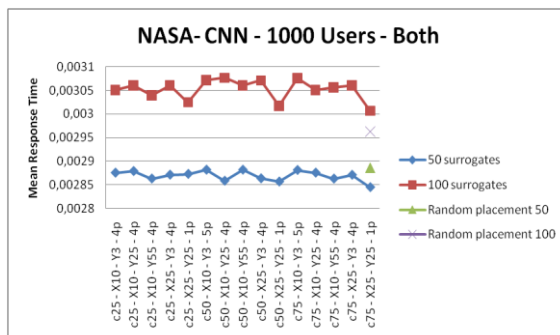
Σχήμα 4-8 Γραφική παράσταση CNN και Yahoo News για 1000 χρήστες και Both μέθοδο συνένωσης



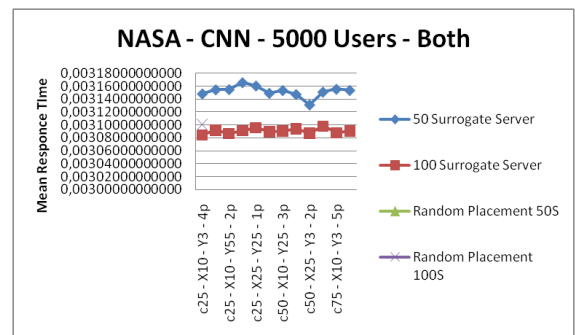
Σχήμα 4-9 Γραφική παράσταση CNN και Yahoo News για 1000 χρήστες και Random μέθοδο συνένωσης

Στις πιο πάνω γραφικές παραστάσεις παρατηρούμε ο μέσος όρος του χρόνου ανταπόκρισης είναι περίπου ο ίδιος είτε χρησιμοποιούμε 50 εξυπηρετητές είτε 100. Επίσης βλέπουμε πολύ έντονες κοιλάδες στα σημεία που το X είναι μεγάλο και το Y είναι μικρό.

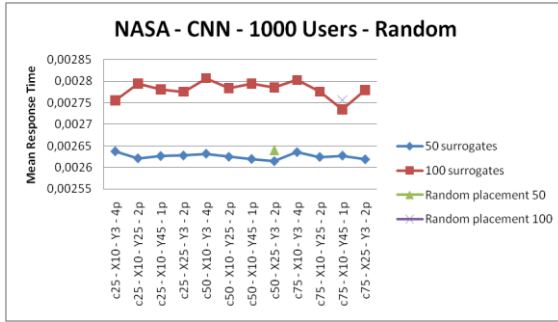
Nasa – CNN



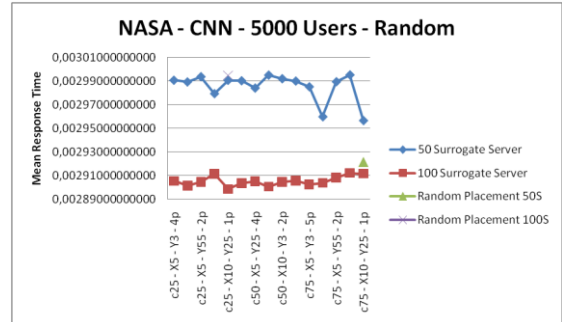
Σχήμα 4-10 Γραφική παράσταση Nasa και CNN για 1000 χρήστες και Both μέθοδο συνένωσης



Σχήμα 4-12 Γραφική παράσταση Nasa και CNN για 5000 χρήστες και Both μέθοδο συνένωσης



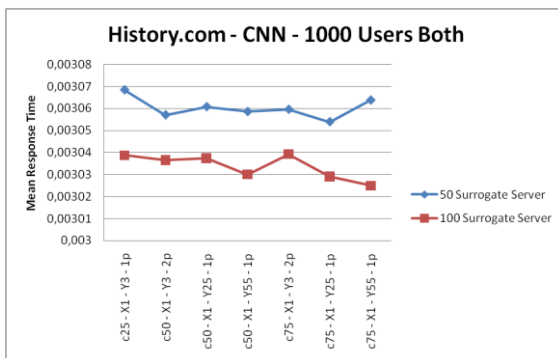
Σχήμα 4-11 Γραφική παράσταση Nasa και CNN για 1000 χρήστες και Random μέθοδο συνένωσης



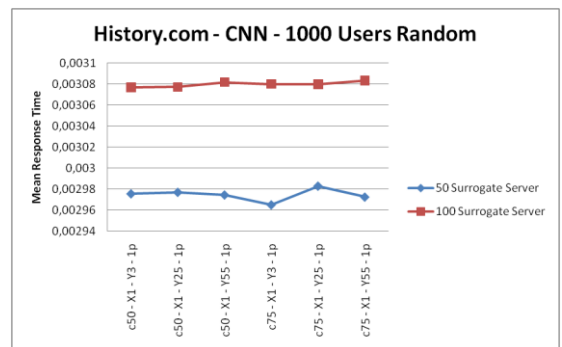
Σχήμα 4-13 Γραφική παράσταση Nasa και CNN για 5000 χρήστες και Random μέθοδο συνένωσης

Στα πιο πάνω πειράματα τα ζεύγη ιστότοπων δεν έχουν κοινό σημασιολογικό περιεχόμενο και στους 1000 χρήστες οι 50 εξυπηρετητές έχουν καλύτερο χρόνο ανταπόκρισης από ότι οι 100 εξυπηρετητές. Δεν υπάρχουν ιδιαίτερες κοιλάδες ή κορυφές και φαίνεται μια σταθερότητα στους χρόνους.

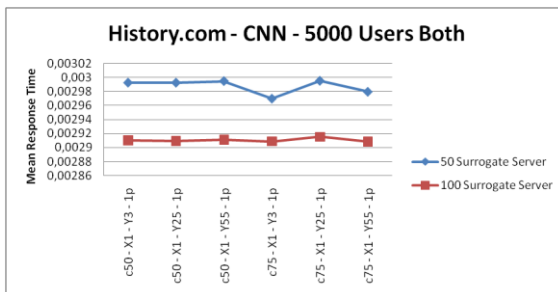
History.com - CNN



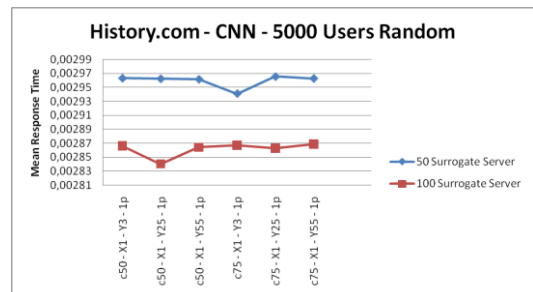
Σχήμα 4-14 Γραφική παράσταση History.com και CNN για 1000 χρήστες και Both μέθοδο συνένωσης



Σχήμα 4-15 Γραφική παράσταση History.com και CNN για 1000 χρήστες και Random μέθοδο συνένωσης



Σχήμα 4-16 Γραφική παράσταση History.com και CNN για 5000 χρήστες και Both μέθοδο συνένωσης



Σχήμα 4-17 Γραφική παράσταση History.com και CNN για 5000 χρήστες και Random μέθοδο συνένωσης

Στα πιο πάνω πειράματα τα ζεύγη ιστότοπων δεν έχουν κοινό σημασιολογικό περιεχόμενο και στους 1000 χρήστες οι 50 εξυπηρετητές με τυχαία επιλογή αρχείου συνόδων, έχουν καλύτερο χρόνο ανταπόκρισης από ότι οι 100 εξυπηρετητές. Παρόμοια και εδώ υπάρχει μια σταθερότητα στους χρόνους ανταπόκρισης.

Γενικά, με 100 εξυπηρετητές αναπαραγωγής τα αποτελέσματα σε χρόνο ανταπόκρισης είναι καλύτερα από αυτά με 50 εξυπηρετητές, αν και τα αποτελέσματα διαφέρουν στο πέμπτο ψηφίο του χρόνου. Στις περισσότερες περιπτώσεις η τυχαία τοποθέτηση κοινοτήτων στους εξυπηρετητές έχει χειρότερους χρόνους από την τοποθέτηση ΚΕΚ σε αυτούς αλλά η διαφορά ήταν και πάλι μηδαμινή.

Σε τελική ανάλυση και 50 εξυπηρετητές αντί 100 να είχε το δίκτυο, την ίδια απόδοση θα είχε. Η διαφορά όμως θα φανεί στις εκατομμύρια αιτήσεις που γίνονται πραγματικά σε ένα ΔΠΠ, όπου και οι χρόνοι θα είναι καλύτεροι για 100 εξυπηρετητές αναπαραγωγής. Επίσης περιμέναμε ότι όσο πιο πολλά ζεύγη ΚΕΚ είχε ένα πείραμα τόσο καλύτεροι χρόνοι θα προέκυπταν. Όμως τελικά δεν έπαιξε ρόλο πόσα ζεύγη έχει αλλά οι καλύτεροι χρόνοι παρατηρήθηκαν όταν το Y ήταν μεγάλο, το X μικρό και αντίστροφα, δηλαδή όταν ένα από τα δύο είναι αρκετά μεγάλο. Συγκεκριμένα, είχαμε βρει το ιδανικό X να είναι γύρω στο 20% από τα πρώτα πειράματα στο κεφάλαιο 4.2. Εδώ επιβεβαιώθηκε ότι όταν το X είναι γύρω στο 10-20% και το Y αρκετά μικρό, τότε παρατηρούνται καλύτεροι χρόνοι ανταπόκρισης. Το ίδιο συμβαίνει και όταν το X είναι πολύ πιο μικρό αλλά το Y κυμαίνεται μεταξύ 25-50%.

Αυτό που παρατηρήθηκε γενικά είναι ότι όσο πιο μικρό είναι το X, τόσο πιο μεγάλο μπορεί να γίνει το Y δίνοντας καλύτερα αποτελέσματα. Αντίθετα, αν βάλουμε πολύ μεγάλο το X, για να βρίσκει κοινού ενδιαφέροντος κοινότητες ο αλγόριθμος και να δίνουν καλά αποτελέσματα, πρέπει το Y να είναι αρκετά μικρό. Γενικότερα όμως το Y δεν φαίνεται να είχε καλά αποτελέσματα πάνω από 50%. Είναι όμως ένα ικανοποιητικό ποσοστό που έχει όντως βελτιστοποιήσεις στο δίκτυο.

Για τα ζεύγη ιστότοπων που δεν είχαν κοινό σημασιολογικό περιεχόμενο, όπως οι δύο τελευταίες ομάδες γραφικών παραστάσεων, παρατηρήθηκε πως σε αρκετές περιπτώσεις η χρήση 50 εξυπηρετητών αναπαραγωγής είχε καλύτερους χρόνους ανταπόκρισης από

την χρήση 100 εξυπηρετητών. Επίσης έχουν πιο σταθερούς χρόνους και οι τιμές των X και Y δεν διαφοροποιούν το τελικό αποτέλεσμα του χρόνου ανταπόκρισης. Έτσι για τις περιπτώσεις που οι ισότοποι δεν έχουν κοινό σημασιολογικό περιεχόμενο δεν μπορούμε να εξάγουμε ξεκάθαρα συμπεράσματα.

Κεφάλαιο 5

Συμπεράσματα

5.1	Γενικά Συμπεράσματα	40
5.2	Μελλοντικές Εργασίες	41

5.1 Γενικά Συμπεράσματα

Μέσα από την εργασία αυτή έχουν υλοποιηθεί κάποια εργαλεία βασισμένα στους αλγορίθμους της εργασίας [1]. Τα εργαλεία αυτά οποία βοήθησαν στο να παρθούν κάποια συμπεράσματα για ένα ιδανικό ποσοστό X και Y κοινού ενδιαφέροντος αντικειμένων και κοινοτήτων αντίστοιχα, ανάλογα με τις προτιμήσεις των χρηστών κατά την πλοήγηση και τις αιτήσεις τους σε αντικείμενα από πολλαπλούς πηγαίους εξυπηρετητές. Μετά την δημιουργία ενός συνόλου δεδομένων (data set) με πειράματα από διαφορετικούς ιστότοπους, είτε με κοινό σημασιολογικό περιεχόμενο είτε χωρίς, αξιολογήθηκε πειραματικά, το κατά πόσο βελτιώνεται η επίδοση όταν αντιγραφούν οι ΚΕΚ, από τους ιστότοπους αυτούς, στους ίδιους εξυπηρετητές αναπαραγωγής, με την χρήση του προσομοιωτή CDNsim. Αφού έγινε η πειραματική αξιολόγηση, εξήχθηκαν κάποια συμπεράσματα για τις ιδανικές τιμές που πρέπει να έχουν τα X και τα Y ούτως ώστε να επιφέρουν βελτιστοποίηση της απόδοσης του ΔΠΠ. Ο μέσος όρος για τις τιμές του X ήταν 20% και κυμαίνονται από 15-25%, ενώ οι τιμές τους Y κυμαίνονται στα 25-50% στις περιπτώσεις αυτές που ο χρόνος ανταπόκρισης ήταν καλύτερος. Παρατηρήθηκε επίσης ότι στις περιπτώσεις που χρησιμοποιήθηκαν 100 εξυπηρετητές αναπαραγωγής, αντί για 50, οι χρόνοι ήταν καλύτεροι, και τέλος, στα ζεύγη ιστότοπων με κοινό σημασιολογικό περιεχόμενο ήταν λιγότερο ξεκάθαρο για το πώς συμπεριφέρεται το ΔΠΠ.

5.2 Μελλοντικές Εργασίες

Η εργασία αυτή μπορεί να συνεχιστεί και να αναπτυχθεί περαιτέρω. Είναι ένα θέμα αρκετά ενδιαφέρον για τον τομέα τον ΔΠΠ, στον οποίο η εργασία αυτή έδωσε τα βασικά θεμέλια. Μπορεί να αναπτυχθεί σε πολλά σημεία, όπως για παράδειγμα να γίνει επέκταση της εύρεσης ΚΕΚ για k κοινότητες, τόσο αλγοριθμικά όσο και προγραμματιστικά. Επίσης καλό θα ήταν τα πειράματα να είναι δοκιμαστούν χρησιμοποιώντας διαφορετικές πολιτικές και ρυθμίσεις για να μελετηθεί και η περίπτωση αύξησης της απόδοσης με κάποιες άλλες επιλογές. Κάτι επιπλέον θα ήταν αν χρησιμοποιούνταν περισσότεροι ισότοποι και να γίνονταν αρκετά ακόμα πειράματα, και τα στατιστικά να εξάγονταν παραπάνω από μια φορά για το κάθε μπουκάλι ούτος ώστε να παρθούν πιο ακριβείς αποτελέσματα και γραφικές παραστάσεις.

Όσον αφορά τον αριθμό των χρηστών στα πειράματα, εύλογο θα ήταν να ελεγχτούν και πειράματα με χρήστες περισσότεροι από 100000, αριθμοί ρεαλιστικοί για τα δεδομένα των ΔΠΠ. Για να γίνει όμως αυτό χρειάζεται μεγάλη υπολογιστική δύναμη και αρκετή μνήμη. Τέλος, αν ήταν εφικτό, τα εργαλεία αυτά να δοκιμαστούν και σε πραγματικά δεδομένα και σε περιβάλλον πραγματικού ΔΠΠ για την εξακρίβωση της αληθινής απόδοσης.

Βιβλιογραφία

- [1] Ε. Ζαβού, ΑΔΕ “Ανάπτυξη Αλγορίθμων για βελτίωση της επίδοσης Δικτύων Παράδοσης Περιεχομένου με Πολλαπλούς Εξυπηρετητές Περιεχομένου – Θεωρητική Ανάλυση”, Πανεπιστήμιο Κύπρου, 2011.
- [2] D. Katsaros, G. Pallis, K. Stamos, A. Vakali, A. Sidiropoulos and Y. Manolopoulos, “CDNs Content Outsourcing via Generalized Communities”, IEEE Computer Society, 2009.
- [3] G. Pallis and A. Vakali, “Insight and Perspectives for Content Delivery Networks”, Communications of the ACM, Vol.49, No.1, January 2006.
- [4] G. Pallis, A. Sidiropoulos, D. Katsaros, K. Stamos, A. Vakali, Y. Manolopoulos: Prefetching in Content Distribution Networks via Web Communities Identification and Outsourcing, Word Wide Web Journal, 11(1): 39-70, 2008.
- [5] K. Stamos, G. Pallis, A. Vakali, M.D. Dikaiakos, “Evaluating the Utility of Content Delivery Networks”, ACM, 2009.
- [6] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos and Y. Manolopoulos, “CDNsim: A Simulation Tool for Content Distribution Networks”, ACM, Vol.20, No. 2, April 2010.
- [7] CDNsim tutorial
<http://oswinds.csd.auth.gr/~cdn sim/docs/tut1/index.html>
- [8] Infrastructure architecture essentials, Part 5: Content delivery and distribution network design
<http://www.ibm.com/developerworks/webservices/library/ar-infraarch5/index.html>.

Π α ρ ά ρ τ η μ α Α

Κώδικας για το εργαλείο initial σε C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "algorithms.h"

/* ----- MAIN PROGRAM ----- */
//Global variables
time_t start,finish;
short show = 0;
short type;
int N = 0; //How many sets available

int main(int argc, char *argv[]){

    if(argc < 5){
        printf("Usage: ./<program> <Structure Type> <Trace file> <X percentage>
<K> [-d]\n");
        exit(1);
    }

    type = atoi(argv[1]);
    FILE *input = fopen(argv[2],"r");
    int X = atoi(argv[3]);
    int K = atoi(argv[4]);

    if((argc > 5) && (strcmp("-d",argv[5]) == 0)){
        show = 1;
    }

    char line[MAX];
    int subjects,objects,i;

    if(!input){
        printf("Could not open file %s\n",argv[2]);
        exit(1);
    }

    system("clear");
    printf("Started: file %s with type %s\n",argv[2],(type ? "List" : "Array"));

    fgets(line,MAX,input);
    subjects = atoi(line);
    fgets(line,MAX,input);
    objects = atoi(line);

    printf("Subjects:%d Objects:%d\n\n",subjects,objects);

    /*
    * T: T[i] is the amount of subjects requesting ith object
    * Q: Q[i] is the set in which the ith object belongs to
    */
    //Initialize arrays T,Q,U
    int *T = (int *)malloc(sizeof(int)*objects);
    int *Q = (int *)malloc(sizeof(int)*objects);
    for(i = 0; i < objects; i++){
        T[i] = 0;
        Q[i] = -1;
    }
}
```

```

int **Ua = NULL;
List *U1 = NULL;

//Useful variables
char *str;
int subj,obj,set;

time(&start);

/* Array option
*
* Ua: Ua[o][s] objects with corresponding subject requesting it
*/

if(!type){
    //Create and Initialize U
    Ua = (int **)malloc(sizeof(int *)*objects);
    for(i = 0; i < objects; i++){
        Ua[i] = (int *)malloc(sizeof(int)*subjects);
        int j;
        for(j = 0; j < subjects; j++){
            Ua[i][j] = 0;
        }
    }

    //Read from file and fill arrays
    while(fgets(line,MAX,input) != NULL){

        //get line's data
        str = strtok(line, " ");
        str = strtok(NULL, " ");
        subj = atoi(str);
        str = strtok(NULL, " .");
        obj = atoi(str);
        str = strtok(NULL, " .");
        set = atoi(str);

        if(N < set)
            N = set;

        //if object already in Ua
        if(Ua[obj][subj] == 1){
            continue;
        }
        //else add it to Ua,T,Q
        Ua[obj][subj] = 1;
        T[obj]++;
        if(Q[obj] < 0)
            Q[obj] = set;
    }

    /*printArrayU(Ua,objects,subjects);
    float sip = arrayFindSIP(Ua,A,B,subjects);
    printf("Similar interest Percentage: %.2f%%\nSets %d\n",sip,N);*/
}

/* List
*
* U1: U1[o] objects with corresponding subjects (in a list) requesting it
*/
else{
    //Initialize U1
    U1 = (List *)malloc(sizeof(List)*objects);
    for(i = 0; i < objects; i++){

```

```

        Ul[i].head = NULL;
        Ul[i].tail = NULL;
    }

//For all requests
while(fgets(line,MAX,input) != NULL){

    //get line's data
    str = strtok(line, " ");
    str = strtok(NULL, " ");
    subj = atoi(str);
    str = strtok(NULL, ".");
    obj = atoi(str);
    str = strtok(NULL, ".");
    set = atoi(str);

    if(N < set)
        N = set;

    Node *head = Ul[obj].head;
    Node *tail = Ul[obj].tail;
    Node *n = (Node *)malloc(sizeof(Node));
    n->subject = subj;
    n->next = NULL;

    //if list is empty
    if(head == NULL){
        Ul[obj].head = n;
        Ul[obj].tail = n;
        T[obj]++;
        Q[obj] = set;
        continue;
    }

    //insert object into the list in ascending sort
    if(subj < head->subject){
        n->next = head;
        Ul[obj].head = n;
        T[obj]++;
        Q[obj] = set;
    }
    else if(subj > tail->subject){
        Ul[obj].tail->next = n;
        Ul[obj].tail = n;
        T[obj]++;
        Q[obj] = set;
    }
    else{
        Node *cur = head;
        Node *pre = NULL;
        while(cur != NULL){
            if(cur->subject > subj){
                n->next = cur;
                pre->next = n;
                T[obj]++;
                Q[obj] = set;
                break;
            }
            else if(cur->subject == subj){
                free(n);
                break;
            }
            else{
                pre = cur;
                cur = cur->next;
            }
        }
    }
}

```

```

    }
    }
} //while all requests

/*printListU(Ul,objects);
float sip = listFindSIP(Ul,A,B,subjects);
printf("Similar interest Percentage: %.2f%%\n",sip);*/

//else list

time(&finish);
N++;

if(N < K){
    printf("\nNumber of sets (servers) given is greater than maximum
sets!\nExiting...\n");
    exit(-1);
}

printf("Structure created in %dms\n\n", (int)(finish-start));

/* ===== MENU =====*/

int choice = 0;
while(1){
    printf("Users: %d No of Sets: %d Type: %s\n",subjects, N, type ? "List" :
"Array");

    printf("Options\n-----\n");
    printf("1) Run Algorithm 2 with X=%d and K=%d\n",X,K);
    printf("2) Change parameters X, K\n");
    printf("3) Toggle show/hide messages\n");
    printf("4) Clear screen\n");
    printf("5) Open output file in gedit\n");
    printf("6) Exit\n");
    printf("Enter option: ");
    scanf("%d", &choice);

    switch(choice){
        case 1: break;
        case 2:
            printf("Enter X and K: ");
            int x,k;
            scanf("%d %d",&x, &k);
            system("clear");
            if(x>100 || k > N){
                printf("\nInvalid X or K!\n\n");
            }
            else{
                X = x;
                K = k;
                system("clear");
                printf("\nX and K changed!\n\n");
            }
            continue;

        case 3:
            system("clear");
            printf("\nMessages are ");
            if(show){
                show = 0;
                printf("off!\n\n");
            }
            else{
                show = 1;
            }
        }
    }
}

```

```

        printf("on!\n\n");
    }
    continue;

    case 4: system("reset"); continue;
    case 5: system("gedit output.txt"); system("clear");
printf("\nFile opened!\n\n"); continue;
    case 6: system("reset"); exit(1);

    default: system("clear"); printf("\nInvalid choice! Try
again!\n\n"); continue;
    }

    system("clear");
    printf("Algorithm 2 started!\n");

    //Structure to specify which origin servers to search in
    int *O = (int *)malloc(sizeof(int) * K);
    char outp[20];

    int size = 0;
    int **comb = nChoosek(N,K, &size); //create combinations
    printf("%d\n",size);

    for(i=0; i<size; i++){ //run alg. #2 for each combination
        int j;
        for(j=0; j<K; j++)
            O[j] = comb[i][j];

        sprintf(outp,"output-%d.txt",i);
        algorithm2(K,O,X,subjects,objects,T,Q,Ua,Ul,N,show,type,outp);
    }

    return 0;
}

/* ----- MAIN PROGRAM END -----*/

```

Εγχειρίδιο χρήσης των εργαλείων

Όλα τα εργαλεία δουλεύουν μόνο σε περιβάλλον Linux και εδώ παρουσιάζεται η εκτέλεση των εργαλείων από τον φάκελο που ανήκουν.

Εργαλείο Γεννήτριας Γράφου Ιστότοπου

Για την ιχνηλασία ενός ιστότοπου:

Μέσα στο φάκελο του εργαλείου websiteGraphGenerator εκτελούμε το αρχείο initial_step.sh δίνοντας του την διεύθυνση του ιστότοπου στο διαδίκτυο που θα ιχνηλατήσει. Στην έξοδο θα δώσει ένα φάκελο με τα αρχεία του ιστότοπου. Για το εργαλείο αυτό χρειάζεται αρκετός χώρος στο σκληρό δίσκο για να σώσει τα αρχεία ολόκληρου του ιστότοπου.

Τρόπος εκτέλεσης: `./initial_step <Διεύθυνση ιστότοπου>`

Για την μετατροπή του ιστότοπου σε γράφο:

Εκτελούμε το αρχείο run.sh δίνοντας του το όνομα του φακέλου στον οποίο βρίσκονται τα αρχεία του ιχνηλατημένου ιστότοπου. Στην έξοδο θα δώσει 3 αρχεία: το αρχείο του γράφου (GRAPH_), το αρχείο με την αντιστοιχία των ταυτοτήτων των αντικειμένων του γράφου με την πραγματική διεύθυνση στο διαδίκτυο (IDS_) και τέλος το αρχείο με τα μεγέθη που έχουν τα αντικείμενα σε bytes(SIZES_).

Τρόπος εκτέλεσης: `./run.sh <Όνομα φακέλου με αρχεία ιστότοπου>`

Εργαλείο Συνένωσης

Για την δημιουργία ενός αρχείου συνόδων εκτελούμε το εργαλείο make_trace.pl του φακέλου TraceGen με τις παραμέτρους εισόδου όπως φαίνονται πιο κάτω. Στην έξοδο θα δώσει το αρχείο συνόδου με κατάληξη “.trans”.

Τρόπος εκτέλεσης: `./make_trace <-d | Εμφάνιση Μηνύματα εξόδου> <Αριθμός χρηστών> <Αριθμός συνόδων> [Τύπος χρονοσφραγίδας -inttime (ακέραιος) ή -floattime (δεκαδικός)] <Αρχείο γράφου>`

Για την δημιουργία ενός συνενωμένου αρχείου αιτήσεων εκτελούμε το εργαλείο `merge_traceGen.pl` με τις παραμέτρους εισόδου όπως φαίνονται πιο κάτω. Στην έξοδο θα δώσει το αρχείο αιτήσεων (`.trace`), το αρχείο αντιστοίχισης (`.con`), το αρχείο με τις αρχικές ταυτότητες αντικειμένων (`.original`) και το αρχείο επεξήγησης των ταυτοτήτων των πηγαίων εξυπηρετητών (`.ids`).

Τρόπος εκτέλεσης: `./merge_traceGen.pl <-d | Εμφάνιση Μηνύματα εξόδου>`
`<Αριθμός χρηστών> <Αριθμός συνόδων> [Τύπος χρονοσφραγίδας -inttime`
`(ακέραιος) ή -floattime (δεκαδικός)] <Πολιτική συνένωσης - 0: Random`
`Surrogates | 1: All surrogates> <Αρχεία συνόδων>`

Εργαλείο εύρεσης KEA

Για την εκτέλεση του διαδραστικού εργαλείου εύρεσης KEA εντοπίζουμε το αρχείο `initial` μέσα στο φάκελο `CIWO`. Εκτελούμε το εργαλείο με τις παραμέτρους εισόδου όπως φαίνονται πιο κάτω. Στην έξοδο θα δώσει ένα αρχείο με τα KEA.

Τρόπος εκτέλεσης: `./initial <Τύπος δομής δεδομένων> <Αρχείο αιτήσεων>`
`<Ποσοστό X> <Αριθμός επιλογής k πηγαίων εξυπηρετητών> <-d | Εμφάνιση`
`Μηνύματα εξόδου>`

Εργαλείο εύρεσης KEK

Για την προετοιμασία των αρχείων για μπορούν να χρησιμοποιηθούν στο εργαλείο εύρεσης KEK εκτελούμε το `PrepareCommunities.pl` από τον φάκελο `CIWC`. Στην έξοδο παίρνουμε τα αρχεία κοινοτήτων μεταφρασμένα (`.clu`), το ανανεωμένο αρχείο αντιστοίχισης (`.con.new`), και τα αρχεία με τους μεγέθους των αντικειμένων των γράφων (`.sizes`).

Τρόπος εκτέλεσης: `./PrepareCommunities.pl <Φάκελος που περιέχει τα αρχεία`
`αντιστοίχισης> <Πρώτο αρχείο κοινότητας (.net.clusters)> <Πρώτο αρχείο`
`κοινότητας (.net.mo.clusters)> <Δεύτερο αρχείο κοινότητας`
`(.net.clusters)> <Δεύτερο αρχείο κοινότητας (.net.mo.clusters)> <Πρώτο`
`αρχείο (.sizes)> <Δεύτερο αρχείο γράφου (.sizes)>`

Για την δημιουργία πειραμάτων με αποτελέσματα διαφόρων X, Y και C εκτελούμε το αρχείο `experimentsC.pl`. Στην έξοδο δίνει 216 πειράματα με τιμές όπως αναφέρονται στο Κεφάλαιο 3.3.2.

Τρόπος εκτέλεσης: `./experimentsC.pl` <Φάκελος που περιέχει τα απαραίτητα αρχεία> <Αριθμός των εξυπηρετητών αναπαραγωγής>

Εξαγωγή στατιστικών από μπουκάλια του CDNsim

Μετά την δημιουργία των μπουκαλιών από τον CDNsim πρέπει να εξάγουμε τα στατιστικά. Μεταφέρουμε τα μπουκάλια μέσα σε ένα φάκελο με τα ακόλουθα αρχεία που βρίσκονται στον φάκελο `BottleScripts`: `batch.sh`, `extract.pl`, `runme.sh`, `stats.py`, `stats.sh` και `util.py`. Το κυρίως εργαλείο εξαγωγής στατιστικών είναι το `run.sh` το οποίο και εκτελείται χωρίς παραμέτρους και μας δίνει στην έξοδο το αρχείο `report.html` με τα στατιστικά σε μορφή υπερκειμένου και το αρχείο `stats.txt` που περιέχει συγκεκριμένα στατιστικά εξαγμένα από το `report.html`.