ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Π.Μ.Σ. ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση κατανεμημένου αλγορίθμου εξόρυξης συχνών προτύπων

Σταυροπούλου Δήμητρα
Α.Μ. 13009

Επιβλέπων: Κώστας Βασιλάκης

Τρίπολη, Απρίλιος 2016

Ευχαριστίες:

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Κώστα Βασιλάκη για την ευκαιρία που μου έδωσε να ασχοληθώ με μια τόσο ενδιαφέρουσα εργασία, για την καθοδήγηση και την κατανόησή του καθόλη τη διάρκεια εκπόνησης και που μου έκανε την τιμή να συνεργαστώ μαζί του διότι είναι ένας εξαίρετος επιστήμονας και καθηγητής και ένας σπάνιος άνθρωπος.

Επίσης οφείλω να ευχαριστήσω όλους τους καθηγητές του Μεταπτυχιακού διότι με τη διδασκαλία και τις εργασίες τους συνέβαλλαν στη διαμόρφωση ερευνητικής σκέψης, σύνθεσης και συγγραφής. Θέλω επίσης να ευχαριστήσω τα μέλη της τριμελούς εξεταστικής επιτροπής μου που με τίμησαν με τη συμμετοχή τους σε αυτήν.

Τέλος, θέλω να ευχαριστήσω θερμά το σύζυγό μου Τάσο για τη συμπαράσταση και την υπομονή του καθόλη τη διάρκεια των σπουδών μου, χωρίς αυτή τη συμβολή του δεν θα τα είχα καταφέρει.

“Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?”
T. S. Eliot (1888 - 1965), The Rock

Πίνακας Περιεχομένων

Ευρετήριο Εικόνων

Περίληψη

Στη σημερινή εποχή του υπερφόρτου των πληροφοριών (the Big Data era) ο εκθετικά αυξανόμενος όγκος της πληροφορίας στο διαδίκτυο κάνει εύκολη μεν την πρόσβαση αλλά δύσκολη την χρησιμότητα της ανακτημένης πληροφορίας. Έτσι δημιουργείται η ανάγκη για εξατομικευμένο περιεχόμενο, τόσο για τις εταιρείες παροχής υπηρεσιών όσο και για τον απλό χρήστη. Η εξατομίκευση στο επίπεδο αυτό μπορεί να υλοποιηθεί μέσω συστημάτων παροχής συστάσεων (recommender systems). Όμως, λόγω της αρχιτεκτονικής του διαδικτύου και του όγκου των δεδομένων, με το τεράστιο εύρος κατανεμημένης παγκόσμιας πληροφορίας (server farms, blog aggregators κ.α.), οι κυριότερες υπάρχουσες λύσεις είναι κεντρικοποιημένες (centralized) και παρουσιάζουν προβλήματα απόδοσης και επεκτασιμότητας. Στην παρούσα εργασία χρησιμοποιείται ένας αλγόριθμος συστάσεων που ανταπεξέρχεται σε αυτήν την αρχιτεκτονική, λαμβάνοντας υπόψη του την διεσπαρμένη πληροφορία και με διαμοιρασμένο το περιεχόμενο αλλά και τα αρχεία καταγραφής χρήσης (clickstream log files) συνεργατικά βρίσκει τα συχνά πρότυπα και εξάγει συστάσεις/προτάσεις προς τους χρήστες. Ο αλγόριθμος αυτός έχει υλοποιηθεί σε μελέτη περίπτωσης με διαμερισμό βάσει των μισάωρων χρήσης και προκειμένου να λυθούν προβλήματα που προκύπτουν από την κατανεμημένη λογική του, στην παρούσα υλοποίηση παρουσιάζονται δύο νέες προσεγγίσεις: 1) με λογική διαμέρισης βάσει του αντικειμένου που έχει καταγραφεί για τους χρήστες (την θεματολογία των κλικ που έχουν κάνει) και τις καταγεγραμμένες συνόδους (session) των χρηστών στα αρχεία καταγραφής του κάθε κόμβου και 2) με λογική διαμέρισης βάσει του αντικειμένου που έχει καταγραφεί για τους χρήστες (την θεματολογία των κλικ που έχουν κάνει) του κάθε κόμβου.

Abstract

In today's Big Data era the exponentially growth of information volume on the internet makes access of information easy but not always worthy. There is a need for personalized content for both companies and simple users. This personalization can derive from provision of recommendations. Majority of recommendation systems is centralized but because of internet architecture and vast distributed global information (server farms, blog aggregators etc.) this is neither efficient nor scalable. A recommendation algorithm taking into account the internet's architecture and scattered information, the distribution of content, the distribution of clickstream data and collaboratively discovers frequent patterns and proposes recommendations, is being used. This recommendation algorithm has been implemented using half hour partitioning distribution in the scattered log files. To solve problems arising from this previous implementation, in this paper two approaches have been implemented: 1) distribution based on the item's id (data users have clicked) and the sessions of users as recorded in the participating's peers log files and 2) distribution based on the item's id (data users have clicked) as recorded in the participating's peers log files.

1. Εισαγωγή

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών (Π.Μ.Σ.) στην "Επιστήμη και την Τεχνολογία Υπολογιστών" με θεματική κατεύθυνση "Τεχνολογίες Αιχμής και Ερευνητικά Θέματα Υπολογιστών" του τμήματος " Πληροφορικής και Τηλεπικοινωνιών" του Πανεπιστήμιου Πελοποννήσου. Σκοπός της είναι η υλοποίηση ενός κατανεμημένου αλγορίθμου εξόρυξης συχνών προτύπων με λογική διαμέρισης βάσει αντικειμένου όσον αφορά τα αρχεία καταγραφής της δραστηριότητας. Παραδοτέα είναι δύο έργα (Java NetBeans projects) και τα αντίστοιχα περιβάλλοντα εκτέλεσης (runtime folders) που αφορούν στις δύο προσεγγίσεις που υλοποιήθηκαν.

Στη συνέχεια της εισαγωγής θα αναφερθούμε στο πώς ο πληροφοριακός υπερφόρτος δημιουργεί την ανάγκη εξατομίκευσης μέσω της δημιουργίας συστημάτων συστάσεων που βασίζονται σε τεχνικές εξόρυξης γνώσης από δεδομένα ιστού (web mining). Στο δεύτερο κεφάλαιο θα παρουσιάσουμε το θεωρητικό υπόβαθρο της υλοποίησης. Στο τρίτο κεφάλαιο θα γίνει αναφορά στη σχετική δουλειά που έχει γίνει και στην οποία βασίστηκε η υλοποίησή μας. Στο τέταρτο κεφάλαιο θα γίνει η περιγραφή του συστήματος και της λειτουργικότητάς του. Στο πέμπτο κεφάλαιο θα παρουσιαστεί το περιβάλλον ανάπτυξης του συστήματος και θα γίνει αναφορά σε σημεία του κώδικα όπου χρειάζεται. Τέλος θα δείξουμε τα συμπεράσματα και τις πιθανές προεκτάσεις με σκοπό την περαιτέρω βελτίωση της εργασίας μας.

1.1. Πληροφοριακός υπερφόρτος και Εξατομίκευση

Η γρήγορη αύξηση του ρυθμού παραγωγής νέων πληροφοριών, προϊόντων και υπηρεσιών καθώς και οι τεχνολογικές διευκολύνσεις αποθήκευσης, μετάδοσης και διάδοσης των δεδομένων [1] έχουν οδηγήσει σε εκρηκτική αύξηση της προσβάσιμης ψηφιακής πληροφορίας. Σύμφωνα με τον Dennett [2] οι άνθρωποι μπορούν να χαρακτηριστούν σαν ένα είδος πληροφοριοβόρων (informavores) που "αναζητούν, συγκεντρώνουν, μοιράζονται και καταναλώνουν πληροφορίες σε βαθμό που δεν μπορεί να προσεγγιστεί από άλλους οργανισμούς". Μέσα σε όλη αυτή την υπερπροσφορά πληροφορίας, μπορεί κανείς να χαθεί και τελικά να μην ικανοποιήσει την ανάγκη που τον έκανε να ξεκινήσει την αναζήτηση πληροφορίας. Αυτή η αδυναμία διαχείρισης του μεγάλου όγκου πληροφοριών και εντοπισμού της χρήσιμης πληροφορίας αναφέρεται στη βιβλιογραφία [3] σαν πρόβλημα πληροφοριακού υπερφόρτου (information overload).

Παραφράζοντας τον γενικό κανόνα του 1% [4], μπορούμε να πούμε πως το 99% της παρεχόμενης πληροφορίας δεν έχει κανένα ενδιαφέρον για το 99% των ανθρώπων. Χαρακτηριστικά αναφέρουμε ότι το σύνολο των ιστοσελίδων έχει ξεπεράσει το ένα δισεκατομμύριο[1] ενώ οι συνολικοί χρήστες του διαδικτύου έχουν ξεπεράσει τα 3 δισεκατομμύρια δηλαδή σχεδόν το μισό πληθυσμό της γης 46.4 %, 3,366,261,156/7,259,902,243/[2].

Η εξατομίκευση μπορεί να αποτελέσει τη λύση στο πρόβλημα του πληροφοριακού υπερφόρτου (information overload) καθώς ο εξ' ορισμού στόχος της είναι το να παρέχεται στους χρήστες αυτό που θέλουν ή χρειάζονται, χωρίς να πρέπει να το ζητήσουν ρητά ή να το αναζητήσουν οι ίδιοι [5]. Πρόκειται για μια πολυσυλλεκτική ερευνητική περιοχή που χρησιμοποιεί τεχνικές από διάφορα επιστημονικά πεδία για να συγκεντρώσει τα απαραίτητα δεδομένα και να παράγει την εξατομικευμένη έξοδο για κάθε μεμονωμένο χρήστη ή ομάδα χρηστών. Τα πεδία από τα οποία δανείζεται τεχνικές περιλαμβάνουν την ανάκτηση πληροφοριών, τη μοντελοποίηση χρήστη, την τεχνητή νοημοσύνη, της βάσεις δεδομένων, και άλλα [3].

Η εξατομίκευση χρησιμοποιείται κατ' εξοχήν στο χώρο του ηλεκτρονικού εμπορίου με τη μορφή προτάσεων για προϊόντα, εξατομικευμένη προώθηση, τιμολόγηση, διαμορφώσεις της μορφής και των περιεχομένων των σελίδων των ηλεκτρονικών καταστημάτων κ.α. Παράλληλα, έχει σταδιακά εξαπλωθεί και σε άλλες κατηγορίες εφαρμογών στο διαδίκτυο, όπως για παράδειγμα σε πληροφοριακές πύλες (portals), σε περιβάλλοντα ηλεκτρονικής μάθησης και σε μηχανές αναζήτησης (όπου π.χ. τα αποτελέσματα φιλτράρονται ή/και ταξινομούνται σύμφωνα με το προφίλ κάθε χρήστη).

Μιλώντας πλέον για εξατομίκευση, εννοούμε περισσότερο συστήματα πρόβλεψης προτιμήσεων των χρηστών και λιγότερο ρητές παραμετροποιήσεις από τους χρήστες. Γενικότερα τα συστήματα πρόβλεψης προτιμήσεων ονομάζονται συστήματα συστάσεων (Recommender Systems) τα οποία είναι συστήματα που βοηθούν τους χρήστες να ανακαλύψουν πράγματα που ίσως τους ενδιαφέρουν [6].

Συνοψίζοντας, η εξατομίκευση μπορεί να βοηθήσει και το χρήστη να βρει την πληροφορία που τελικά αναζητούσε δίνοντάς του ικανοποίηση και προστιθέμενη αξία, αλλά και τις εταιρίες παροχής υπηρεσιών/προϊόντων που τελικά θα προωθήσουν την κατάλληλη υπηρεσία/προϊόν στο καταναλωτή που θα ανταποκριθεί δίνοντάς τους κέρδη. Η εξόρυξη γνώσης λοιπόν από όλο αυτό το μεγάλο όγκο πληροφοριών του διαδικτύου είναι ένας τομέας που προσελκύει το ενδιαφέρον πολλών πλευρών και αποτελεί την επιστημονική περιοχή του web mining.

[1] Έρευνα Μάρτιος 2016 http://news.netcraft.com/archives/2016/03/18/march-2016-web-server-survey.html#more-23062
[2] Έρευνα Νοέμβριος 2015, http://www.internetworldstats.com/stats.htm

1.2. Web mining

Ως web mining ορίζεται η χρήση τεχνικών ανάκτησης δεδομένων (data mining) για την ανακάλυψη και εξαγωγή γνώσης από έγγραφα και υπηρεσίες web [7]. Η πιο διαδεδομένη ταξινόμηση web mining ορίζει τρεις βασικές κατευθύνσεις έρευνας: εξόρυξη περιεχομένου (content mining), εξόρυξη δομής (structure mining) και εξόρυξη χρήσης (usage mining). Και οι τρεις κατευθύνσεις χρησιμοποιούν τεχνικές εξόρυξης γνώσης για να δημιουργήσουν μοντέλα της πληροφορίας αλλά βασίζονται σε διαφορετικά δεδομένα.

Η εξόρυξη περιεχομένου (content mining) χρησιμοποιεί δεδομένα από αντικείμενα του διαδικτύου όπως απλό κείμενο, ημιδομημένα έγγραφα (π.χ. HTML, XML), δομημένα έγγραφα (ψηφιακές βιβλιοθήκες), πολυμέσα κ.α. Η εξόρυξη δομής (structure mining) προσπαθεί να βρει την τοπολογία των διασυνδέσεων μεταξύ των αντικειμένων του διαδικτύου. Η εξόρυξη χρήσης (usage mining) χρησιμοποιείται για να ανακαλύψει πρότυπα από τα δεδομένα χρήσης. Τα δεδομένα αυτά αφορούν διάδραση του χρήστη με το διαδίκτυο και συνήθως αποτυπώνονται σε αρχεία καταγραφής των εξυπηρετητών [8] όπως web/proxy server logs, ερωτήματα – queries, κ.α.

Η τεχνική εξόρυξης γνώσης από δεδομένα ιστού (web mining) που χρησιμοποιείται σε αυτή την εργασία προκειμένου να προτείνει συστάσεις, αφορά εξόρυξη χρήσης (usage mining) με χρήση δεδομένων log αρχείων εξυπηρετητών. Η πλειοψηφία όμως των κατανεμημένων αλγορίθμων συστάσεων προϋποθέτει την ύπαρξη μιας κεντρικής βάσης δεδομένων συναλλαγών που κατανέμεται μεταξύ των εμπλεκομένων επεξεργαστών. Πολύ λίγες είναι οι εφαρμογές που είναι κατανεμημένες αλλά δεν διαμοιράζονται τίποτα και η επεξεργασία γίνεται ανεξάρτητα στον κάθε κόμβο. Αυτό οδήγησε στο να φτιαχτεί μια υλοποίηση που να βασίζεται σε κατανεμημένα peer-to-peer (P2P) συστήματα και ταιριάζει με την αρχιτεκτονική των ψηφιακών βιβλιοθηκών (Federated Digital Library - FDL), των κατανεμημένων δικτύων περιεχομένου (Content Delivery Networks - CDN), των φαρμών εξυπηρετητών (server farms), των συλλογών προώθησης (blog aggregators) όπου οι παροχείς περιεχομένου είναι ανεξάρτητοι μεταξύ τους και δεν έχουν και κάποια σχέση ιεραρχίας [9].

Στην παρούσα εργασία έχει χρησιμοποιηθεί ένα κατανεμημένο σύστημα συστάσεων βασισμένο στον αλγόριθμο εξόρυξης γνώσης από γενικευμένα πρότυπα για την παροχή συστάσεων FGP (Frequent Generalized Pattern mining algorithm) [9] που συνδυάζει τον FP-Growth [10] αλγόριθμο αναζήτησης συχνών προτύπων με τον GP-close [11, 12] αλγόριθμο γενικευμένων κανόνων συσχέτισης (Generalized Association Rule Mining - GARM). Ακολουθεί μια θεωρητική επισκόπηση των βασικών ορολογιών και αλγορίθμων στα οποία βασίστηκε η εργασία.

2. Θεωρητικό υπόβαθρο

2.1. Εξόρυξη Δεδομένων (Data mining)

Ως Εξόρυξη Δεδομένων (data mining) ορίζεται η διαδικασία ανακάλυψης (discovery) προτύπων (patterns) από μεγάλα σύνολα δεδομένων (data sets) που πριν δεν ήταν γνωστά, ισχύουν, είναι πιθανών χρήσιμα, είναι κατανοητά και η ανάλυση τους μας παρουσιάζει μη αναμενόμενες σχέσεις ανάμεσα στα δεδομένα βοηθώντας μας να τα συνοψίσουμε με νέους τρόπους που είναι κατανοητοί και χρήσιμοι στους χρήστες [13]. Πρόκειται για μια τεχνολογία αιχμής για την αποτελεσματικότερη ανάλυση των δεδομένων και την αποκάλυψη νέων σχέσεων, την εξαγωγή ενδιαφέρουσας πληροφορίας ή προτύπων από μεγάλες βάσεις δεδομένων. Χρησιμοποιεί μεθόδους που συνδυάζουν την στατιστική, την τεχνητή νοημοσύνη, τη μηχανική μάθηση και τα συστήματα βάσεων δεδομένων.

Η μεθοδολογία εύρεσης γνώσης [14] αποτελείται από τα εξής στάδια:

1. Καθορισμός δεδομένων (data cleaning), δηλαδή, απομάκρυνση του θορύβου και των ακατάλληλων δεδομένων
2. Ενοποίηση δεδομένων (data integration), η οποία λαμβάνει χώρα όταν είναι απαραίτητος ο συνδυασμός διάφορων πηγών δεδομένων
3. Επιλογή δεδομένων (data selection), όπου ουσιαστικά γίνεται η επιλογή και η ανάκτηση των προς χρήση δεδομένων από τη βάση δεδομένων
4. Μετατροπή δεδομένων (data transformation), σε μία κατάλληλη προς επεξεργασία μορφή
5. Εξόρυξη δεδομένων (data mining), κατά την οποία γίνεται εξαγωγή μοτίβων – προτύπων από τα δεδομένα, με την εφαρμογή ευφυών μεθόδων
6. Αξιολόγηση μοτίβων (pattern evaluation), όπου γίνεται η αναγνώριση και η επιλογή των πραγματικά ενδιαφερόντων μοτίβων με την χρήση μετρικών ενδιαφέροντος (interestingness measures)
7. Αναπαράσταση γνώσης (knowledge presentation), κατά την οποία γίνεται παρουσίαση της εξαγόμενης γνώσης στους χρήστες με την χρήση των κατάλληλων τεχνικών οπτικοποίησης και αναπαράστασης

Οι κύριες τεχνικές εξόρυξης είναι:

1. Κατηγοριοποίηση (Classification) και αφορά στην ανάθεση ενός νέου αντικειμένου σε μία από τις προϋπάρχουσες κλάσεις (κατηγορίες) και απαιτεί πρότερη γνώση για τα δεδομένα

2. Συσταδοποίηση (Clustering) όπου γίνεται διαχωρισμός ενός συνόλου δεδομένων σε ένα σύνολο από δύο η περισσότερες συστάδες (clusters), με τέτοιο τρόπο ώστε τα αντικείμενα κάθε συστάδας να είναι κατά κάποιο τρόπο όμοια μεταξύ τους
3. Κανόνες συσχέτισης (Association Rules) που εμφανίστηκαν για τις ανάγκες ανάλυσης του «καλαθιού αγοράς» (market basket analysis) και αποκαλύπτουν ενδιαφέρουσες σχέσεις μεταξύ των δεδομένων

Στην παρούσα εργασία που αφορά εξόρυξη γνώσης από δεδομένα χρήσης ιστού (web usage mining), η μέθοδος ανακάλυψης προτύπων θα αφορά σε ανακάλυψη κανόνων συσχέτισης (association rule mining).

2.2. Εύρεση κανόνων συσχέτισης (Association rule mining)

Ως βάση αναζήτησης συχνών προτύπων θεωρείται η εύρεση κανόνων συσχέτισης (Association rule mining ARM) [15]. Ένας κανόνας συσχέτισης (association rule) είναι ένα μοντέλο που αναγνωρίζει ειδικούς τύπους συσχέτισης μεταξύ των δεδομένων [16, 17]. Είναι ουσιαστικά ένα συμπέρασμα της μορφής X → Y, όπου το X και το Y είναι μη κοινά μεταξύ τους συχνά στοιχειοσύνολα (frequent itemsets) σε μια βάση δεδομένων, και εμφανίζονται συχνότερα από μια ελάχιστη υποστήριξη και εμπιστοσύνη. Η υποστήριξη s (support) προκύπτει από τον αριθμό των εγγραφών της βάσης που περιέχουν και το X και το Y, ενώ η εμπιστοσύνη c (confidence) από τον αριθμό των εγγραφών της βάσης που περιέχουν το X εφόσον αυτές οι εγγραφές περιέχουν και το Y. Η εύρεση έγκυρων λοιπόν κανόνων συσχέτισης μπορεί να χρησιμοποιηθεί στην δημιουργία συστάσεων (recommendations) με το να προτείνει ένα σύνολο σελίδων Y σε ένα χρήστη που έχει ήδη δει ένα σύνολο σελίδων X, βασιζόμενη στο γεγονός ότι αυτή είναι μια κοινή συμπεριφορά περιήγησης σε πολλούς χρήστες [18].

Η πιο διαδεδομένη προσέγγιση για την εύρεση κανόνων συσχέτισης χρησιμοποιεί τα συχνά στοιχειοσύνολα (frequent itemsets). Αυτά ορίζονται ως τα στοιχειοσύνολα (itemsets) εκείνα των οποίων ο αριθμός των εμφανίσεων είναι πάνω από ένα κατώφλι s (support threshold). Η εύρεση των συχνών στοιχειο-συνόλων γίνεται εντοπίζοντας τα στοιχεία και στοιχειοσύνολα που ικανοποιούν τον περιορισμό του κατώτερου ορίου s. Έπειτα δημιουργούνται οι κανόνες από τα ανακαλυφθέντα συχνά στοιχειοσύνολα. Ο αλγόριθμος Apriori [19] είναι ο πιο γνωστός αλγόριθμος εύρεσης κανόνων συσχέτισης. Βασίζεται στην αρχή πως αν ένα σύνολο αντικειμένων είναι συχνό, πρέπει και όλα του τα υποσύνολα να είναι συχνά. Χρησιμοποιεί την προσέγγιση “Δημιούργησε και Δοκίμασε” (Generate and Test) για να βρει τα συχνά στοιχειοσύνολα και αφού δημιουργήσει σε κάθε στάδιο όλα τα υποψηφία στοιχειοσυνόλα, έπειτα σαρώνει τη βάση δεδομένων για να μετρηθούν και να βρεθεί αν αυτά είναι συχνά. Εξαιτίας αυτής της λογικής του είναι μη αποδοτικός και μη επεκτάσιμος.

Τέλος, στις μέρες μας χρησιμοποιείται και ο όρος κατανεμημένη εύρεση κανόνων συσχέτισης (Distributed Association Rule Mining - DARM) όπου οι συμμετέχοντες κόμβοι είναι ανεξάρτητοι μεταξύ τους με μόνη προϋπόθεση τη διασύνδεσή τους μέσω ενός δικτύου επικοινωνίας (shared-nothing architecture).

2.3 Frequent pattern mining

Η αναζήτηση συχνών προτύπων (frequent pattern mining) βασίζεται λοιπόν στην εύρεση κανόνων συσχέτισης. Έχουν γίνει πολλές επεκτάσεις και παραλλαγές από τον Apriori [19] προκειμένου να φτιαχτούν αποδοτικοί και επεκτάσιμοι αλγόριθμοι αναζήτησης συχνών προτύπων. Κάποιοι λαμβάνουν υπόψη την σειρά εμφάνισης των αντικειμένων, τη χρονική απόσταση εμφάνισης των αντικειμένων σε ένα στοιχειοσύνολο και άλλες παραμέτρους προκειμένου να λάβουν υπόψη τους ένα στοιχειοσύνολο ώστε να ελέγξουν τη συχνότητά του στη βάση και έτσι να εξάγουν τους κανόνες και τα συχνά πρότυπα. Ο αλγόριθμος που χρησιμοποιείται σε αυτήν την εργασία είναι ο αλγόριθμος εξόρυξης γνώσης από γενικευμένα πρότυπα για την παροχή συστάσεων FGP (Frequent Generalized Pattern mining algorithm) [9] που είναι συνδυασμός των FP-growth [10] και GP-close [11, 12] στους οποίους κάνουμε αναφορά στις επόμενες παραγράφους αυτού του κεφαλαίου.

2.3.1 FP-Growth

Ο αλγόριθμος FP-growth [10] ανήκει στην κατηγορία αλγορίθμων αναζήτησης συχνών προτύπων με προβολή της βάσης δεδομένων (Pattern Growth and Database Projection - PGDP) και χρησιμοποιεί δομή δέντρου μειώνοντας τόσο την απαιτούμενη μνήμη όσο και τα περάσματα από τη βάση. Βρίσκει τα συχνά σετ αντικειμένων χωρίς να δημιουργεί υποψηφίους. Διαβάζει τη βάση δύο φορές.Στο πρώτο πέρασμα βρίσκει τις συχνότητες των αντικειμένων, αφαιρεί όσα από αυτά δεν ικανοποιούν το κατώφλι s (support threshold) και ταξινομεί τα αντικείμενα κατά φθίνουσα σειρά συχνότητας. Στο δεύτερο πέρασμα δημιουργεί τη δομή FP-tree διαβάζοντας μία προς μία τις συναλλαγές στη βάση χρησιμοποιώντας τους ίδιους κόμβους στα μονοπάτια που μοιράζονται κοινά αντικείμενα. Τέλος εφαρμόζει τον αλγόριθμο FP-Growth στη δομή FP-tree και όχι στη βάση και με αναδρομή βρίσκει από τη βάση προς την κορυφή (bottom–up) όλα τα συχνά σετ αντικειμένων.

2.3.2 GP-close

Στην εξόρυξη γνώσης από χρήση δεδομένων του διαδικτύου υπάρχει η ιδιαιτερότητα πως τα αντικείμενα που επισκέπτεται ο χρήστης συνήθως έχουν μια ιεραρχία εννοιών (taxonomy), για παράδειγμα σε ένα ιστοχώρο ειδήσεων θα πρέπει κάποιος να επιλέξει τις αθλητικές ειδήσεις για να επιλέξει μετά άθλημα ποδόσφαιρο και να δει τα αποτελέσματα του ποδοσφαίρου κ.ο.κ.. Χρησιμοποιώντας αυτήν την ιεραρχία εννοιών, η οποία παρουσιάζει τη συσχέτιση του συνόλου μεταξύ διαφόρων στοιχείων, μπορούμε να εξάγουμε γενικευμένους κανόνες συσχετίσεων (Generalized Association Rule Mining - GARM) που επιτρέπουν δημιουργία κανόνων σε διάφορα επίπεδα και έτσι κανόνες συσχετίσεων θα μπορούσαν να δημιουργηθούν για οποιοδήποτε επίπεδο στην ιεραρχία. Ένας γενικευμένος κανόνας συσχέτισης (generalized association rule) $X \rightarrow Y$, ορίζεται όπως και ένας συνηθισμένος κανόνας συσχέτισης με τον περιορισμό ότι κανένα στοιχείο του Y δε μπορεί να είναι πάνω (στην ιεραρχία) από ένα στοιχείο του X. Όταν δημιουργούνται γενικευμένοι κανόνες συσχετίσεων όλοι οι πιθανοί κανόνες δημιουργούνται χρησιμοποιώντας μία ή περισσότερες δεδομένες ιεραρχίες.

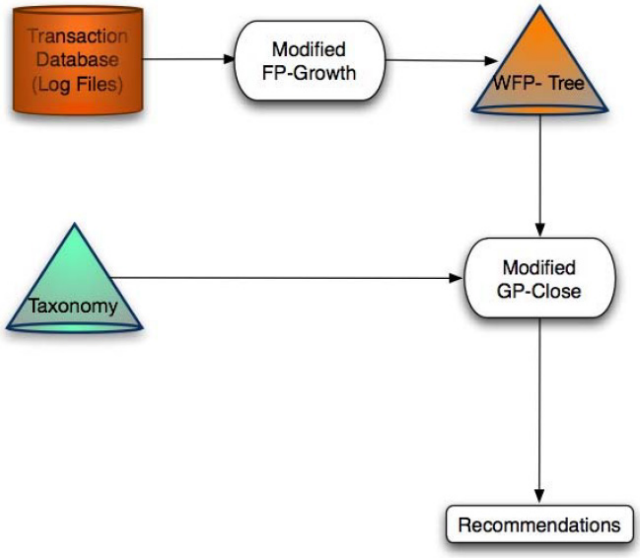
Στην παρούσα εργασία ο FGP χρησιμοποιεί τον GP-close αλγόριθμο [11] προκειμένου να κάνει αναζήτηση συχνών προτύπων λαμβάνοντας υπόψη του την ταξινομία των αντικειμένων και εξαλείφοντας τους περιττούς υπερ-γενικευμένους κανόνες. Ο GP-Close χρησιμοποιεί τις εγγραφές μιας βάσης δεδομένων και μια ταξινομία T που περιέχει όλα τα αντικείμενα της βάσης. Λαμβάνοντας υπόψη την ελάχιστη τιμή κατωφλιού s (minimum support threshold) δημιουργεί το δέντρο GT που περιέχει όλα τα γενικευμένα συχνά στοιχειοσύνολα. Κάνει ταξινόμηση κατά αύξουσα σειρά και επεκτείνει κάθε κόμβο παιδί κατά ένα αντικείμενο από ότι έχει το σετ αντικειμένων του γονέα. Ο GP-close χρησιμοποιεί δύο τεχνικές κλαδέματος στη δομή δέντρου που φτιάχνει: κλάδεμα κλειστότητας - παιδιού (child-closure pruning) και κλάδεμα υπο-δέντρου (subtree pruning). Αυτό ισοδυναμεί με εξόρυξη κλειστών προτύπων (closed pattern mining), δηλαδή εύρεση στοιχειοσύνολων που κανένα από τα άμεσα υπερσύνολα τους δεν έχει την ίδια υποστήριξη με αυτά (έχει μικρότερη υποστήριξη) [11, 12].

2.3.3 Αλγόριθμος εξόρυξης γνώσης από γενικευμένα πρότυπα FGP (Frequent Generalized Pattern mining algorithm)

Ο αλγόριθμος που χρησιμοποιείται στην παρούσα εργασία είναι ο αλγόριθμος εξόρυξης γνώσης από γενικευμένα πρότυπα FGP [20]. Συνδυάζει τις δυνατότητες εύρεσης κανόνων συσχέτισης του FP-Growth και την εξόρυξη γνώσης γενικευμένων προτύπων του GP-Close ώστε αποδοτικά να δημιουργεί κανόνες. Ο αλγόριθμος εύρεσης γενικευμένων συχνών προτύπων (FGP) θεωρεί ότι όλα τα αντικείμενα που εμφανίζονται σε ένα σύνολο συναλλαγών ανήκουν σε κατηγορίες που είναι οργανωμένες σε

ταξινομίες. Χρησιμοποιεί τις συναλλαγές μιας βάσης δεδομένων και μια ταξινομία των κατηγοριών και παράγει γενικευμένους κανόνες συσχέτισης που εμπεριέχουν αντικείμενα των συναλλαγών και/ή κατηγορίες αντικειμένων. Είναι ένας ιδιαίτερα χρήσιμος αλγόριθμος για εξατομίκευση ιστότοπων που ανανεώνουν διαρκώς το περιεχόμενό τους όπως είναι οι συλλογείς προώθησης (blog aggregators), οι πύλες ενημέρωσης (news portals) κ.α. Όσον αφορά τη χρήση της λογικής του FP-growth, εμπλουτίζει την δομή δέντρου FP-tree ώστε να περιλαμβάνει το βάρος της πληροφορίας κάθε αντικειμένου και παράγει το ζυγισμένο πλέον δέντρο FP-tree (weighted FPTree WFP-Tree) . Αντιμετωπίζει το πρόβλημα του συνεχώς ανανεώσιμου περιεχομένου των ιστότοπων με το να χρησιμοποιεί το WFP-Tree και την ταξινομία του περιεχομένου του ιστότοπου στον GP-Close και να παράγει γενικευμένες συστάσεις.

Ο αλγόριθμος ακολουθεί τα παρακάτω βήματα (Εικόνα 1):



Εικόνα 1. FGP (Frequent Generalized Pattern mining algorithm)

1. Διαβάζει την βάση και δημιουργεί το WFP-Tree.
2. Βρίσκει τα συχνά στοιχειοσύνολα με τη χρήση του WFP-Tree.
3. Δημιουργεί τα συχνά γενικευμένα στοιχειοσύνολα κάνοντας χρήση της ταξινομίας.
4. Ταξινομεί τα στοιχειοσύνολα βάσει αύξουσας σειράς μεγέθους υποστήριξης.
5. Κλαδεύει τα παιδιά: ενώ δημιουργεί το γενικευμένο δέντρο «κόβει» τις γενικεύσεις του αντικειμένου του οποίου η υποστήριξη s είναι ίση με την υποστήριξη συχνού στοιχειοσυνόλου που βρίσκεται ήδη στο δέντρο.
6. Συνδυάζει τα στοιχειοσύνολα ώστε να δημιουργήσει το πλήρες δέντρο γενικευμένων στοιχειοσυνόλων.

7. Κλαδεύει τα υποδέντρα: Εάν ένα n-στοιχειοσύνολο A μπορεί να υπαχθεί/απορροφηθεί από ένα αναγνωρισμένο k-στοιχειοσύνολο στο δέντρο, με n ⊂ k και υποστήριξη(A)<= υποστήριξη (B) τότε το αντίστοιχο υποδέντρο «κλαδεύεται».

Τα συχνά k-στοιχειοσύνολα χρησιμοποιούνται στη συνέχεια για να δημιουργήσουν συστάσεις για κάποιον χρήστη U. Δεδομένου ότι το ιστορικό πλοήγησης του χρήστη περιλαμβάνει (k-r) αντικείμενα από ένα συγκεκριμένο k-στοιχειοσύνολο, το σύστημα επιλέγει και συστήνει τα υπόλοιπα αντικείμενα. Το r είναι παράμετρος που ορίζεται από το σύστημα και παίρνει ακραίες τιμές από το 1 έως το k. Σε περίπτωση που το προβλεπόμενο στοιχείο είναι κατηγορία, το σύστημα μπορεί να παρέχει μια λίστα με τις n-κορυφαίες σελίδες που ανήκουν στην κατηγορία, επιλέγοντας τις πιο δημοφιλείς ή τις πιο πρόσφατες [21].

2.4 P2P systems

Τα συστήματα κόμβος με κόμβο (peer to peer - P2P) είναι συστήματα όπου κόμβοι σχηματίζουν μεταξύ τους ένα δίκτυο όπου διασυνδέονται, επικοινωνούν και ανταλλάσσουν πληροφορίες διατηρώντας την αυτονομία τους. Έχουν κατανεμημένη αρχιτεκτονική και οι συμμετέχοντες είναι εξίσου προνομιούχοι και ισοδύναμοι.

Ένα ευρύ φάσμα συστημάτων P2P έχει προταθεί στη βιβλιογραφία, που περιλαμβάνει από μη δομημένες προσεγγίσεις [22], χωρίς καμία συγκεκριμένη τοπολογία όσον αφορά τους συμμετέχοντες κόμβους, σε δομημένες προσεγγίσεις όπως είναι τα Chord [23], CAN [24] και P-Grid [25] που λαμβάνουν υπόψη τους τη διατήρηση της τοπολογίας δικτύου κατά τη διάρκεια ενημερώσεων και εγγυώνται σωστές απαντήσεις χωρίς υπερβολική κατανάλωση του bandwidth. Η πλειοψηφία των σύγχρονων P2P συστημάτων, συμπεριλαμβανομένων και των κατανεμημένων πινάκων κατακερματισμού (distributed hash tables - DHTs) εμπίπτουν σε αυτήν την κατηγορία. Επιπλέον, ιεραρχικές εφαρμογές όπως ο Edutella [26] και ο P2P-DIET [27], κατανέμουν τους κόμβους τους σε δύο σύνολα: σε κόμβους - πελάτες, που δημοσιεύουν έγγραφα ή υποβάλλουν ερωτήσεις και σε υπερ-κόμβους (super-peers) που αποθηκεύουν πληροφορίες σχετικά με τα αρχεία που έχουν οι κόμβοι-πελάτες. Τα P2P συστήματα μπορούν να ταξινομηθούν περαιτέρω σε κεντρικοποιημένα, (όπως π.χ. ο Napster) που διαχωρίζουν έναν κόμβο ο οποίος αποθηκεύει ένα ευρετήριο των υπολοίπων καθώς και τους πόρους που διατίθενται αυτοί να μοιραστούν, και σε αποκεντρωμένα όπως ο Gnutella [22] όπου δεν υπάρχει ανάγκη για έναν κύριο κόμβο.

3. Σχετική Εργασία

Υπάρχει πληθώρα προσεγγίσεων στο πρόβλημα της εξατομίκευσης στο διαδίκτυο. Μια εκτενής επισκόπηση μπορεί να δει κανείς στην εργασία των [28]. Στην υλοποίησή μας, μας ενδιαφέρουν τα γενικευμένα πρότυπα πρόβλεψης που κάνουν χρήση ιεραρχίας. Το πρόβλημα με ιστότοπους όπως τα ιστολόγια ή οι πύλες ενημέρωσης, είναι ότι το περιεχόμενό τους ανανεώνεται συνεχώς. Στην περίπτωση των συλλογών προώθησης (blog aggregators) υπάρχει λιγότερος έλεγχος στις ετικέτες που ανατίθενται στα αντικείμενα. Αφού δεν ανήκουν σε κάποια ιεραρχία χρειάζεται να κάνουμε παραπάνω προσπάθεια ώστε να τα αναθέσουμε σε κάποιο κόμβο ιεραρχικά [29]. Μερικές προσεγγίσεις βασίζονται σε πληροφορίες προτίμησης των χρηστών [14, 30] ωστόσο τα ενδιαφέροντα των χρηστών αλλάζουν με τον καιρό. Επειδή υπάρχει ακριβώς αυτό το ζήτημα της αλλαγής της προτίμησης, οι χρήστες είτε πρέπει να ενημερώνουν συνεχώς τις προτιμήσεις τους, είτε το σύστημα δεν θα μπορεί να παράσχει χρήσιμες, εξατομικευμένες προτάσεις. Αυτό μοιάζει με το πρόβλημα της ψυχρής εκκίνησης (cold-start problem) όπου ένα σύστημα πρέπει να κάνει προβλέψεις χωρίς να υπάρχει οποιοδήποτε ιστορικό συναλλαγών. Το πρόβλημα αυτό έχει απαντηθεί κυρίως για συστήματα που χρησιμοποιούν συνεργατικό φιλτράρισμα (collaborative filtering systems [31, 32]) με τη δημιουργία υβριδικών συστημάτων συστάσεων (hybrid recommender systems) που λαμβάνουν υπόψη και το περιεχόμενο του ιστότοπου και τις βαθμολογήσεις και το προφίλ του χρήστη. Όταν δεν υπάρχουν επαρκείς πληροφορίες για το χρήστη χρησιμοποιούνται ομοιότητες του περιεχομένου προκειμένου να γίνουν προβλέψεις.

Η ιδέα της ενσωμάτωσης του περιεχομένου στη διαδικασία συστάσεων έχει αντιμετωπισθεί γενικεύοντας τα πρότυπα πλοήγησης από επίπεδο σελίδας, σε ένα υψηλότερο, γενικό επίπεδο, με τη βοήθεια της ιεραρχίας των θεμάτων. Έχει προταθεί η αντιστοίχηση των συνόδων (sessions) των χρηστών με ιεραρχημένα θέματα [33] όπου οι γενικευμένες σύνοδοι έμπαιναν ως είσοδο στον αλγόριθμο Apriori algorithm [19] προκειμένου να παραχθούν συστάσεις βασισμένες στην κατηγοριοποίηση που είχε γίνει. Παρόμοια πρόταση έγινε από τους Oberle, Berendt [34] και για σημασιολογικούς ιστότοπους (semantic web sites) όπου το περιεχόμενο επισημαινόταν με τη χρήση οντολογίας περισσότερο όμως για εξόρυξη γνώσης και όχι για εξατομίκευση. Έπειτα από έρευνα σε συστήματα συστάσεων [28, 35] και συνδυάζοντας την αποδοτικότητα του FP-Growth με τον μηχανισμό γενικευμένων κανόνων συσχέτισης του GP-Close σχεδιάστηκε ο FGP προκειμένου να παράγει γενικευμένες συστάσεις (generalized recommendations) οι οποίες χρησιμοποιούν τόσο συνδυασμό σελίδων όσο και συνδυασμό κατηγοριών των σελίδων [21].

Έπειτα αναπτύχθηκε ο FGP+, μια προέκταση του FPG, που δέχεται ως δεδομένα εισόδου πιο περίπλοκες δομές από ότι είναι η ταξινομία. Η προσθήκη αυτή έγινε προκειμένου να ανταπεξέλθει ο FGP στα ιδιαίτερα χαρακτηριστικά που έχουν οι ιστότοποι του Web 2.0 όπως π.χ. οι συναθροιστές ροής (feed aggregators) όπου το θέμα ή η ετικέτα μιας σελίδας δύσκολα αποτελεί μέρος ταξινομία [21].

Οι ελλείψεις στην απόδοση της εφαρμογής του FGP σε κεντρικοποιημένη αρχιτεκτονική καθώς και η κυριαρχία των συστημάτων μηδενικής κοινοχρησίας (shared-nothing architectures) οδήγησαν σε νέα κατανεμημένη εφαρμογή του FGP και πάνω σε αυτήν βασίζεται η παρούσα εργασία. Εκεί παρουσιάζεται η χρήση του αλγόριθμου FGP στων τομέα των P2P συστημάτων και έδειξε αποδοτική εξόρυξη κατανεμημένων αρχείων καταγραφής χρήσης δικτύου (web logs) σε καθεστώς μηδενικής κοινοχρησίας (shared-nothing scheme). Για την συγκριτική αξιολόγηση είχαν χρησιμοποιεί διαφορετικού μεγέθους αρχεία καταγραφής με διαφορετικό αριθμό συμμετεχόντων κόμβων. Η απόδοση είχε μετρηθεί με διαφορετικές συνθήκες δικτύωσης και το αποτέλεσμα ήταν θετική αποτίμηση απόδοσης και καλή αξιολόγηση ποιότητας των συστάσεων που παρήγαγε η συγκεκριμένη υλοποίηση [9].

4. Περιγραφή συστήματος και λειτουργικότητας

Η υλοποίηση μας αφορά σε ένα κατανεμημένο σύστημα συστάσεων που αποτελείται από κόμβους που στον καθένα αποθηκεύεται ένα μέρος των αρχείων καταγραφής πλοήγησης του κάθε χρήστη. Ο κάθε κόμβος είναι πάροχος πληροφορίας και το κάθε αρχείο καταγραφής του αντιστοιχεί σε αιτήματα πληροφορίας που του ζήτησαν οι χρήστες. Κάποιος κόμβος μπορεί να εξειδικεύεται σε συγκεκριμένο θέμα ταξινομίας οπότε στην περίπτωσή του, τα αρχεία καταγραφής του θα περιέχουν αντικείμενα μόνο από το συγκεκριμένο θέμα ταξινομίας. Τέτοιο σενάριο είναι κοινό στις ψηφιακές βιβλιοθήκες, στις φάρμες εξυπηρετητών, στα δίκτυα διανομής περιεχομένου κ.α. Οι κόμβοι έχουν ήδη σχέση εμπιστοσύνης μεταξύ τους και έτσι διευκολύνεται και η ανταλλαγή δεδομένων μεταξύ τους για να επιτευχθεί ο στόχος της παραγωγής συστάσεων.

Ένα βασικό θέμα που επηρεάζει τον αλγόριθμο παραγωγής συστάσεων είναι ότι οι πληροφορίες της εκάστοτε συνόδου (session) μπορεί να είναι διασκορπισμένες σε διάφορα αρχεία καταγραφής μεταξύ των κόμβων οπότε το να τρέξει ο αλγόριθμος σε καθένα αρχείο σημαίνει ότι θα του λείπουν μοτίβα συμπεριφοράς που αφορούν άλλα αντικείμενα που ζητήθηκαν στην ίδια σύνοδο και άρα βρίσκονται σε αρχεία καταγραφής άλλου κόμβου (π.χ. κάποιος είδε πληροφορίες για αθλητικά και μετά ζήτησε σελίδες νέων επικαιρότητας). Ο επανασχηματισμός των συνόδων (session reconstruction - sessionize) αποτελεί το πρώτο πράγμα που πρέπει να λυθεί προκειμένου να παραχθούν έγκυρες συστάσεις. Το εύκολο θα ήταν να συγκεντρωθούν όλα τα αρχεία καταγραφής κάπου κεντρικοποιημένα αλλά θα υπήρχαν θέματα επίδοσης και επεκτασιμότητας.

Προκειμένου να λυθεί το πρόβλημα των συνόδων κατανεμημένα θα χρησιμοποιηθεί η λογική του κατανεμημένου συστήματος συστάσεων [9] που διαιρεί την επεξεργασία σε τέσσερις φάσεις: φάση ανάθεσης, φάση ανταλλαγής αρχείων καταγραφής, φάση τοπικής εξόρυξης και φάση ανταλλαγής κανόνων/μοτίβων. Η συμβολή αυτής της εργασίας αφορά στην διαφορετική υλοποίηση των φάσεων της ανάθεσης και της ανταλλαγής των αρχείων καταγραφής.

4.1 Φάση ανάθεσης

Στην υλοποίηση του κατανεμημένου συστήματος Giannikopoulos and Vassilakis [9] η χρονική περίοδος που κάλυπταν τα αρχεία καταγραφής χωριζόταν σε N ίσες υποπεριόδους (όπου N οι κόμβοι του P2P συστήματος) και κάθε κόμβος αναλάμβανε μια υποπερίοδο την οποία και επεξεργαζόταν. Ο καταμερισμός γινόταν διατηρώντας τα μισάωρα χρήσης προκειμένου να μην διασπώνται οι συνοδοί μεταξύ διαφορετικών κόμβων, κάτι που δεν αποκλειόταν αφού για καταμερισμό του 24ωρου σε 5

κόμβους χρησιμοποιούνταν η εξής διαμέριση {P1: [00:00–05:00), P2: [05:00–10:00), P3: [10:00–15:00), P4: [15:00–19:30), P5: [19:30–24:00 αντί για {P1: [00:00–04:48), P2: [04:48–09:36), P3: [09:36–14:24), P4: [14:24–19:12), P5: [19:12–24:00)} αλλά κάποιος που ξεκίνησε σύνοδο στις 19.05 και την έληξε στις 19.34 θα χωριζόταν τελικά σε δύο κόμβους. Ο διαχωρισμός και η ανάθεση γινόταν από έναν κόμβο «διαιτητή», τον Arbitrator, στον οποίο εγγράφονταν οι συμμετέχοντες κόμβοι και αυτός ανακοίνωνε σε όλους τις αναθέσεις των υποπεριόδων. Η επιλογή ενός κόμβου ως διαιτητή - Arbitrator δεν χρειάζεται την ύπαρξη υπερ-κόμβου αλλά μπορεί να συναποφασιστεί με οποιοδήποτε αλγόριθμο κατανεμημένης εκλογής.

Στην υλοποίησή μας παραμένει η λογική της ανάθεσης από ένα κόμβο διαιτητή μόνο που η διαμέριση δεν γίνεται πλέον βάσει υποπεριόδων χρόνου αλλά βάσει αντικειμένου που ζητήθηκε από τον χρήστη. Κάθε κόμβος στέλνει στον Arbitrator το πλήθος των διαφορετικών αντικειμένων που του ζητήθηκαν μαζί με το πλήθος αυτών (τα counts τους). Ο Arbitrator λαμβάνοντας αυτήν την πληροφορία από τον κάθε κόμβο, ισομερίζει τις εγγραφές βάση αντικειμένου και ενημερώνει όλους με τις αναθέσεις. Π.χ. ο κόμβος Α θα αναλάβει τις εγγραφές που αφορούν τα αντικείμενα a, b, z, ο Β τα αντικείμενα f, t, w, ο Γ τα c, g κ.ο.κ. Έχουν ληφθεί υπόψη δύο πιθανά σενάρια όσον αφορά τη λογική του συστήματος και τις συνόδους των χρηστών:

Σενάριο 1ο : η κάθε αίτηση του κάθε χρήστη τυπικά δρομολογείται σε έναν κόμβο για μία σύνοδό του ανεξάρτητα από τις προηγούμενες (π.χ. ανάλογα με το ποιος είναι ο λιγότερο φορτωμένος εκείνη τη στιγμή) και έτσι μπορούμε να φτιάξουμε τις συνόδους (sessionize) από πριν και να τις διανείμουμε

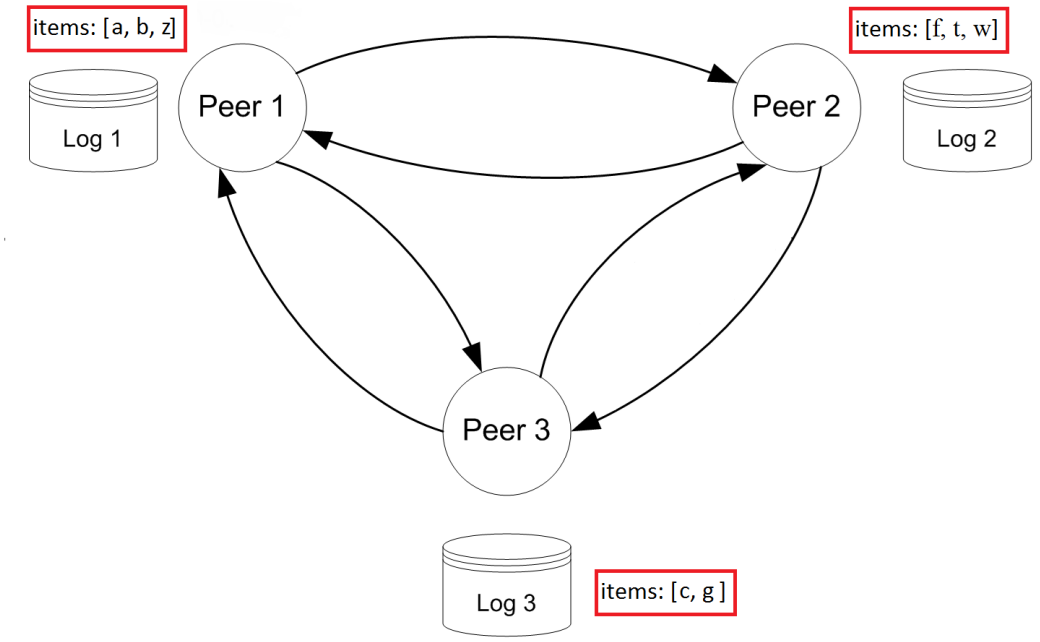
Σενάριο 2ο : οι κόμβοι παρουσιάζουν εξειδίκευση ανά θέμα, π.χ. ο κόμβος Α έχει τα items που αφορούν στη θεματολογία Α1, ο κόμβος Β αυτά που αφορούν στη θεματολογία Β1 και έτσι οι σύνοδοι φτιάχνονται μετά την ανταλλαγή

Στο 1ο σενάριο οι κόμβοι καταμετρούν πέραν των αντικειμένων και τις συνόδους που έχουν δημιουργηθεί στο αρχείο καταγραφής τους και έτσι η κατασκευή συνόδων (sessionize) καθορίζεται στο πρώτο βήμα και θα χρησιμοποιηθεί στην επόμενη φάση. Στο 2ο σενάριο δεν απαιτείται επιπλέον δουλειά από τους κόμβους σε αυτό το στάδιο πέραν της μέτρησης των αντικειμένων

4.2 Φάση ανταλλαγής αρχείων καταγραφής

Μόλις τελειώσει η φάση της ανάθεσης αντικειμένων κάθε κόμβος γνωρίζει πλέον τις δικές του αναθέσεις αλλά και των υπολοίπων κόμβων. Έτσι όλοι οι κόμβοι, διαβάζουν το αρχείο καταγραφής τους και επιλέγουν τι θα κρατήσουν για τον εαυτό τους και τι θα στείλουν στον κατάλληλο κόμβο. Παράλληλα

διαβάζουν τις εγγραφές που τους στέλνουν οι άλλοι κόμβοι. Όσον αφορά το 1ο σενάριο η διαμοίραση των εγγραφών ακολουθεί την τακτική όπου κάθε εγγραφή μιας συνόδου πηγαίνει σε κάθε κόμβο που έχει τουλάχιστον ένα αντικείμενο της συνόδου, άρα μια εγγραφή μπορεί να βρίσκεται σε πολλαπλούς κόμβους αφού άλλες εγγραφές της συνόδου μπορεί να έχουν αντικείμενα που ανήκουν σε άλλους κόμβους. Όσον αφορά το 2ο σενάριο, κάθε εγγραφή γράφεται μόνο σε έναν κόμβο και το ζήτημα των συνόδων αντιμετωπίζεται σε παρακάτω φάση. Όταν ολοκληρώνεται αυτή η φάση, κάθε κόμβος έχει ένα αρχείο με όλες τις εγγραφές που του έχουν ανατεθεί και πάνω στις οποίες θα κάνει εξόρυξη. Αυτό το αρχείο αποτελεί το αρχείο δεδομένων του FGP. Όπως είναι εμφανές, ο φόρτος του δικτύου σε αυτή τη φάση της ανταλλαγής εγγραφών είναι μεγαλύτερος στο 1ο σενάριο αφού μια εγγραφή μπορεί να χρειαστεί να σταλεί σε παραπάνω από έναν κόμβους.



Εικόνα 2. Απεικόνιση ανταλλαγής δεδομένων του συστήματος

4.3 Φάση τοπικής εξόρυξης

Οι κόμβοι πλέον έχουν το αρχείο με τις εγγραφές που τους έχουν ανατεθεί και μπορούν ο καθένας τοπικά να εφαρμόσει τον αλγόριθμο αναζήτησης συχνών προτύπων. Εφαρμόζουμε εδώ τον FGP όπως αυτός έχει παρουσιαστεί παραπάνω. Τα συχνά πρότυπα που παράγει κάθε κόμβος συνοδεύονται και από τον αριθμό των φορών που εμφανίζονται αυτά τα πρότυπα ώστε να προστεθούν συνολικά μαζί με τον αριθμό φορών που βρέθηκε το συχνό πρότυπο σε άλλους κόμβους. Αυτό πάλι γίνεται κατανεμημένα στην επόμενη φάση της ανταλλαγής κανόνων/μοτίβων.

4.4 Φάση ανταλλαγής κανόνων/μοτίβων

Επειδή κάποια συχνά πρότυπα που παρήχθησαν στην προηγούμενη φάση από το τοπικό αρχείο δεδομένων εισόδου του FGP μπορεί να παρήχθησαν και σε άλλους κόμβους, από το δικό τους αρχείο, για να βρεθεί η υποστήριξη (support) του προτύπου αυτού συνολικά, θα πρέπει να προστεθούν όλα τα τοπικά μεγέθη υποστήριξης του κάθε συχνού προτύπου. Για να γίνει αυτό κάθε κόμβος αντιστοιχεί μια τιμή ως αποτέλεσμα μιας συνάρτησης κατακερματισμού (hash function) που αντιστοιχεί κάθε συχνό πρότυπο σε ένα κόμβο στον οποίο και αποστέλλεται το συχνό πρότυπο. Ίδια συχνά πρότυπα θα αντιστοιχιστούν και θα σταλούν στον ίδιο κόμβο ο οποίος απλώς θα συναθροίσει τα τοπικά μεγέθη των ίδιων συχνών προτύπων για να βρει την υποστήριξη s του συγκεκριμένου προτύπου. Ένα συχνό στοιχειοσύνολο της μορφής $\{i1, i2, \dots, in\}$ μπορεί να θεωρηθεί ως κανόνας $[(i1, i2, \dots, in–1) \rightarrow in]$ εάν η σειρά των αντικειμένων έχει σημασία, δηλαδή εάν ένας χρήστης έχει δει τα αντικείμενα $i1, i2, \dots, in–1$ τότε του προτείνεται το αντικείμενο in. Εάν όμως η σειρά δεν έχει σημασία τότε το παραπάνω συχνό στοιχειοσύνολο μπορεί να θεωρηθεί ένα σύνολο κανόνων ως εξής: $\{[(i1, i2, \dots, in–1) \rightarrow in], [(i1, i3, \dots, in) \rightarrow i2], [(i2, \dots, in) \rightarrow i1]\}$. Για κάθε έναν τέτοιο κανόνα, υπολογίζεται μια τιμή (hash value), βάσει συνάρτησης κατακερματισμού, στο πρώτο του μέρος και έτσι να προσδιοριστεί σε ποιόν κόμβο θα προωθηθεί. Έτσι κάθε κόμβος αφού υπολογίσει τους κανόνες του, τους ομαδοποιεί βάσει αυτής της hash τιμής και τους στέλνει στον κατάλληλο κόμβο με ένα μήνυμα. Όταν στη συνέχεια χρειαστεί να γίνει σύσταση για κάποιον χρήστη που έχει δει τα αντικείμενα $\{i1, i2,\dots, in–1\}$, η hash τιμή του σετ υπολογίζεται και ερωτάται ο κατάλληλος κόμβος να παράγει την επιθυμητή σύσταση.

5. Υλοποίηση και Πειραματική αξιολόγηση

5.1 Βασικά περιβάλλοντα υλοποίησης

Η ανάπτυξη του κώδικα αφορά εργασία σε δύο βασικά σύνολα αρχείων:

1. το σύνολο αρχείων με τον κώδικα που τρέχει ο κάθε συμμετέχων για να εκτελέσει την εξόρυξη στο κομμάτι που του ανατίθεται και να ανταλλάξει κανόνες με τους άλλους συμμετέχοντες
2. το σύνολο των αρχείων που αφορούν το περιβάλλον εκτέλεσης των πειραμάτων (runtime environment).

Το (1) αφορά σε NetBeans Java project και περιέχει τις βασικές κλάσεις java που αντιστοιχούν στη λειτουργικότητα των συμμετεχόντων κόμβων για επικοινωνία με τον Arbitrator, για επικοινωνία μεταξύ τους καθώς και ό,τι χρειάζεται για την εφαρμογή του FGP στο αρχείο τους.

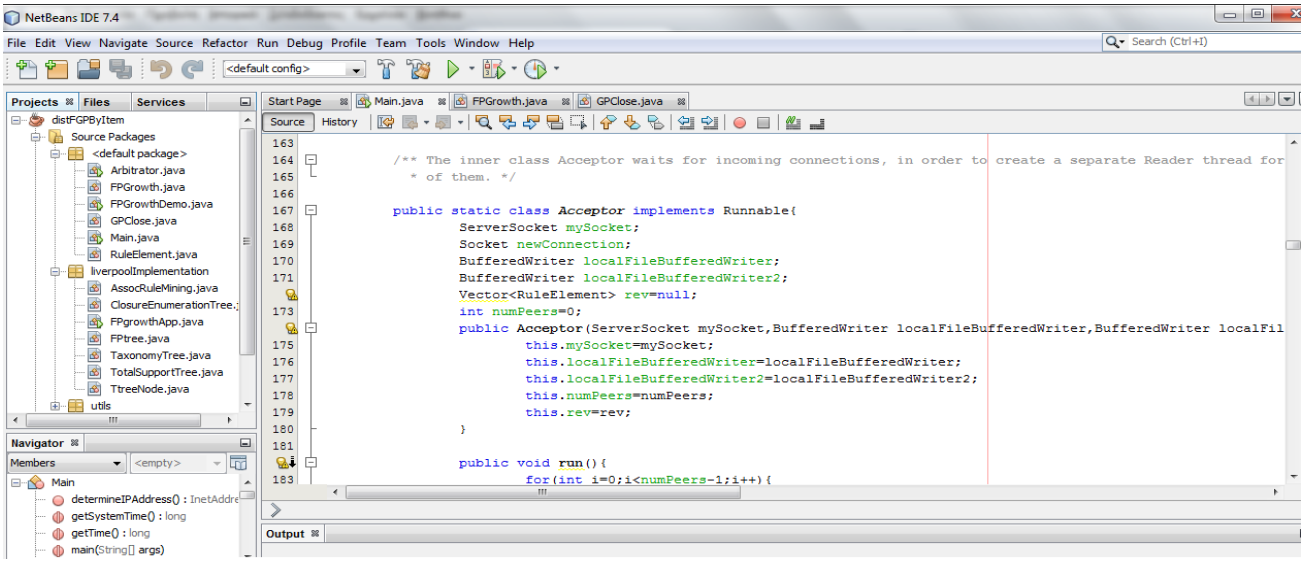
Το (2) αφορά σε ένα σύνολο αρχείων απαραίτητων ώστε να προσομοιωθεί το τρέξιμο του java project (1) και περιέχει την οντότητα του Arbitrator, κώδικα shell scripts για το τρέξιμο των πειραμάτων, αρχεία κώδικα awk για προεπεξεργασία του αρχείου καταγραφής και εξαγωγής της πληροφορίας που θα χρησιμοποιηθεί για τη διαμέριση και αρχεία που θα βοηθήσουν τον κάθε κόμβο να κάνει σωστά την εξόρυξη (π.χ. αρχείο χρησιμοποιούμενης ταξινομίας taxonomy.txt κ.α.). Χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java (Jdk 1.7.0_51), τεχνικές Java networking programming (Socket, Input/Output streams κ.α.) καθώς και τεχνικές νημάτωσης (Java threading programming) για παράλληλη επεξεργασία. Για την εξόρυξη δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη WEKA [3] (Waikato Environment for Knowledge Analysis).

5.1.1 περιβάλλον ανάπτυξης της εφαρμογής - NetBeans

Το NetBeans [4] IDE (Integrated Development Environment) είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών που βασίζεται στην πλατφόρμα NetBeans και ενώ αρχικά χρησιμοποιήθηκε για την ανάπτυξη εφαρμογών Java, πλέον χρησιμοποιείται και για άλλες γλώσσες όπως PHP, C/C++, και HTML5. Διανέμεται δωρεάν και είναι λογισμικό ανοιχτού κώδικα. Υποστηρίζει την ανάπτυξη όλων των ειδών εφαρμογών Java (lava SE, web, EJB και mobile εφαρμογές). Όλες οι λειτουργίες του IDE παρέχονται από συστατικές μονάδες λογισμικού τα modules. Κάθε module παρέχει μία απαιτούμενη λειτουργία όπως υποστήριξη ανάπτυξης εφαρμογών Java, επεξεργασία κώδικα, υποστήριξη

[3] http://www.cs.waikato.ac.nz/ml/weka
[4] https://netbeans.org/

συστημάτων διαχείρισης εκδόσεων CVS και SVN. Το NetBeans IDE παρέχει όλα τα απαιτούμενα modules για την ανάπτυξη εφαρμογών Java στο αρχικό πακέτο που κατεβάζει κανείς, το οποίο επιτρέπει στον χρήστη να αρχίσει άμεσα την ανάπτυξη μιας εφαρμογής. Τα modules επιτρέπουν την επέκταση του IDE. Νέες λειτουργίες, όπως η υποστήριξη επιπλέον γλωσσών προγραμματισμού μπορούν να προστεθούν μετά την αρχική εγκατάσταση. Η έκδοση που χρησιμοποιήθηκε είναι η 7.4 με χρήση Java jdk: 1.7.0_51.

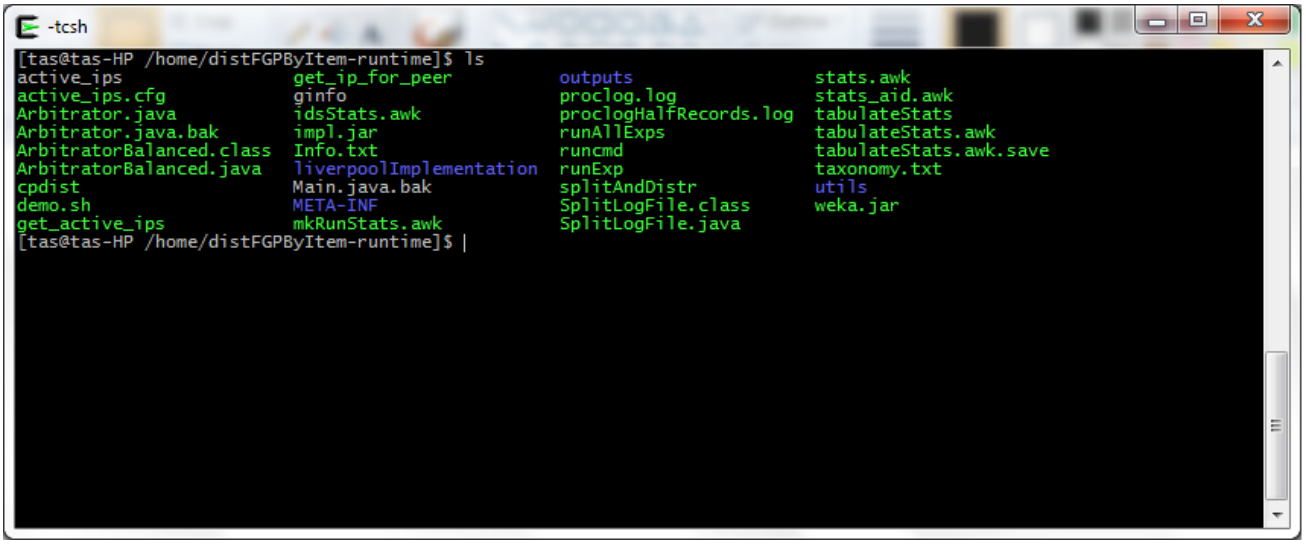


Εικόνα 3. Περιβάλλον ανάπτυξης NetBeans

5.1.2 περιβάλλον εκτέλεσης των πειραμάτων – Cygwin

Το Cygwin[5] είναι ένα περιβάλλον προσομοίωσης του UNIX/Linux στα Windows. Παρέχει στον χρήστη την δυνατότητα να δημιουργεί προγράμματα κάνοντας χρήση των βιβλιοθηκών και των υπηρεσιών του UNIX ενώ το ίδιο το εκτελέσιμο του προγράμματος τρέχει σε Windows. Εκτελώντας το cygwin ανοίγει ένα παράθυρο στο οποίο τρέχει ο κλασικός φλοιός bash του unix στον οποίο μπορούν να εκτελεστούν κανονικά εντολές του unix bash shell, καθώς και μια πληθώρα προγραμμάτων που συναντούμε σε UNIX/Linux. Σαν κεντρικός (root) κατάλογος στο περιβάλλον αυτό (κατάλογος /) ορίζεται ο κατάλογος στον οποίο έχει εγκατασταθεί το cygwin και τα περιεχόμενα των υπόλοιπων καταλόγων και δίσκων μπορούν να προσπελαστούν μέσω των ειδικών καταλόγων: /cygdrive/<γράμα_δίσκου>/<κατάλογος>. Η έκδοση που χρησιμοποιήθηκε είναι η 1.7.28 και Java jdk: 1.7.0_51 και αντί για bash shell χρησιμοποιήθηκε tcsh shell.

[5] https://www.cygwin.com/

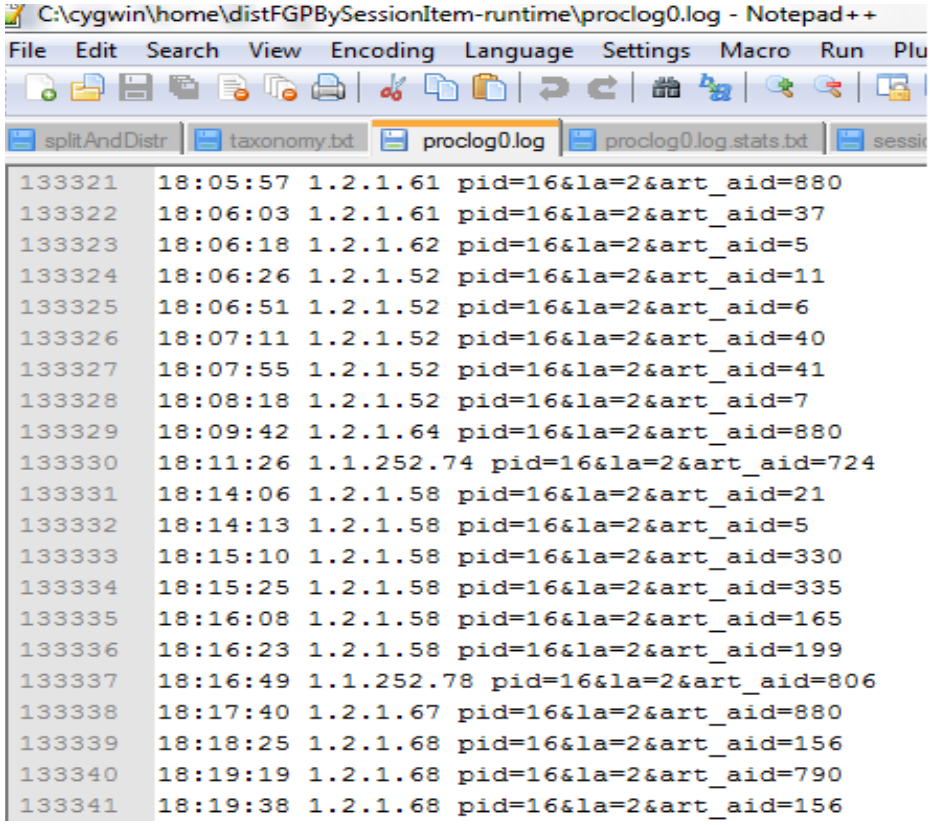


Εικόνα 4. Περιβάλλον εκτέλεσης Cygwin

5.2 Θέματα υλοποίησης και κώδικα

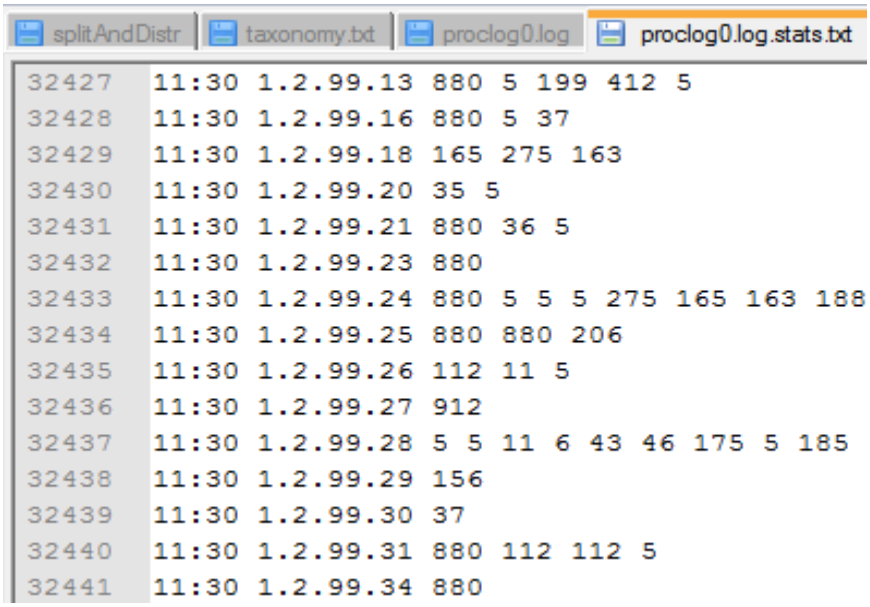
Το σενάριο είναι ότι έχουμε πολλούς αυτόνομους κόμβους - web servers, κάθε έναν με το δικό του αρχείο καταγραφής (log file) που αναλαμβάνουν από κοινού να κάνουν εξόρυξη συχνών προτύπων σο σύνολο της πληροφορίας. Τα παραδοτέα αφορούν σε ένα java NetBeans project και το αντίστοιχο περιβάλλον εκτέλεσης για το 1ο σενάριο και ένα δεύτερο java NetBeans NetBeans project και το αντίστοιχο περιβάλλον εκτέλεσης για το 2ο σενάριο (βλ. παράγραφο 4.1). Τα θέματα υλοποίησης θα παρουσιαστούν παράλληλα για κάθε σενάριο.

Αρχικά κάθε κόμβος τρέχει ένα αρχείο κώδικα awk όπου διαβάζει το καθαρισμένο αρχείο καταγραφής του proclog$pid.log (όπου $pid αντιστοιχεί το αναγνωριστικό του εκάστοτε κόμβου π.χ. 0 για τον πρώτο, 1 για τον δεύτερο κ.οκ.) και δημιουργεί το αρχείο proclog$pid.log.stats.txt που κρατά συγκεντρωτικά πληροφορία όπως θα χρειαστεί να την αντλήσει ο κάθε κόμβος προκειμένου να ακολουθηθεί η λογική της διαμέρισης του κάθε σεναρίου. Παράδειγμα εγγραφών του proclog$pid.log βλέπουμε στην εικόνα 5.



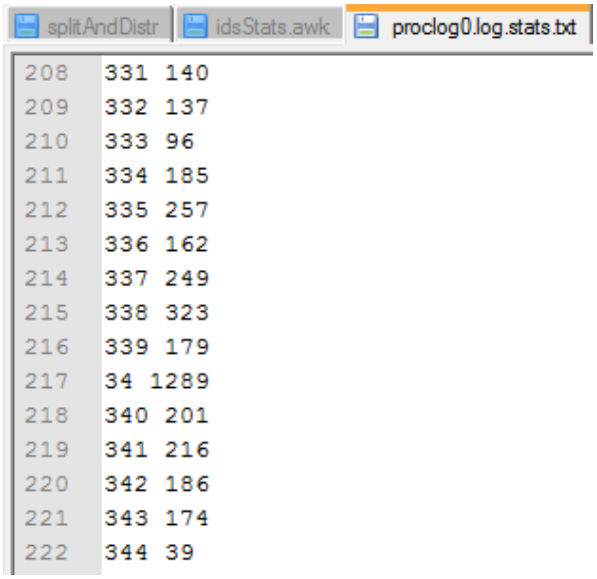
Εικόνα 5. Παράδειγμα εγγραφών αρχείου proclog$pid.log

Για το 1ο σενάριο φτιάχτηκε το αρχείο κώδικα awk sessionIdsStats.awk όπου διαβάζοντας το αρχείο καταγραφής του proclog$pid.log αποθηκεύει στο proclog$pid.log.stats.txt τις συνόδους των χρηστών (sessions) ανά μισάωρο ανά ip τα όπως φαίνεται στην Εικόνα 6.



Εικόνα 6. Αρχείο που θα αντλήσει τις πληροφορίες ο κόμβος για 1ο σενάριο

Για το 2ο σενάριο φτιάχτηκε το αρχείο κώδικα awk idsStats.awk όπου διαβάζοντας το αρχείο καταγραφής του proclog$pid.log αποθηκεύει στο proclog$pid.log.stats.txt τα αντικείμενα που ζητήθηκαν από τον κόμβο και το πόσες φορές ζητήθηκαν (τα counts τους) όπως φαίνεται στην εικόνα 7.



Εικόνα 7. Αρχείο που θα αντλήσει τις πληροφορίες ο κόμβος για 2ο σενάριο

Αφού το αρχείο από το οποίο θα αντλήσουν την πληροφορία είναι έτοιμο, ξεκινά η *Φάση της ανάθεσης* (βλ. παράγραφο 4.1). Οι κόμβοι γνωρίζουν ποιος είναι ο Arbitrator και πρέπει να του στείλουν τα αντικείμενα και το πλήθος αυτών ώστε αυτός να κάνει την ανάθεση.

Για το **1ο σενάριο** κατά την ανάγνωση του αρχείου proclog$pid.log.stats.txt δημιουργούνται δύο δομές: στη μία η πληροφορία των συνόδων και στη δεύτερη τα αντικείμενα με το αντίστοιχο πλήθος τους. Η πρώτη δομή θα χρησιμοποιηθεί αργότερα στη φάση ανταλλαγής των αρχείων. Η δεύτερη δομή είναι που στέλνεται τελικά στον Arbitrator.

Για το **2ο σενάριο** κατά την ανάγνωση του αρχείου proclog$pid.log.stats.txt δημιουργείται μία δομή με τα αντικείμενα και το αντίστοιχο πλήθος τους, η οποία και στέλνεται στον Arbitrator.

Η υλοποίηση του Arbitrator είναι κοινή και στα δύο σενάρια. Ο Arbitrator διαβάζει από κάθε κόμβο τα αντικείμενά του και το πλήθος τους, αρχικά τα αποθηκεύει, έπειτα φτιάχνει μια ενιαία δομή με όλα τα αντικείμενα και συνολικό πλήθος αυτών και έπειτα αναλαμβάνει να επιμερίσει τα αντικείμενα στους κόμβους. Για να το κάνει αυτό ακολουθεί τη στρατηγική να ξεκινά την ανάθεση από τα αντικείμενα με το μεγαλύτερο πλήθος και να τα αναθέτει στον κόμβο με το μεγαλύτερο ποσοστό αυτού του αντικειμένου μέχρι να ισομοιραστούν οι συνολικές εγγραφές που θα πάρει κάθε κόμβος. Μόλις

ολοκληρώσει στέλνει σε όλους τους κόμβους μια δομή με συνολικά όλα τα αντικείμενα και αντίστοιχο κόμβο που τα έχει αναλάβει ώστε όλοι να γνωρίζουν τις αναθέσεις, καθώς και τις διευθύνσεις (ip, port) των υπολοίπων.

Μόλις οι κόμβοι παραλάβουν την κοινή δομή με όλες τις αναθέσεις ξεκινά η *Φάση της ανταλλαγής αρχείων* (βλ. παράγραφο 4.2). Ο κάθε συμμετέχων πλέον διατρέχει το δικό του proclog$pid.log και 1)κρατάει αυτά που του αντιστοιχούν και 2) κατανέμει στους υπόλοιπους τα λοιπά σύμφωνα με τις αναθέσεις, ανοίγοντας σύνδεση μαζί τους. Παράλληλα και ο ίδιος υποδέχεται από τους λοιπούς συμμετέχοντες τα όσα του στέλνονται. Η αποστολή και η λήψη γίνεται παράλληλα χρησιμοποιώντας την ιδιότητα της πολυνημάτωσης της Java (Java multithreading)

Στο 1ο σενάριο κάθε σύνοδος (session) πηγαίνει σε κάθε κόμβο που έχει τουλάχιστον ένα στοιχείο της συνόδου. Έτσι ο κάθε κόμβος παίρνει τις αναθέσεις και χρησιμοποιεί τη δομή που έχει ήδη φτιάξει με τις συνόδους ώστε να τις αναθέσει αντίστοιχα. Κάθε σύνοδος επειδή μπορεί να περιέχει αντικείμενα που να έχουν ανατεθεί σε διαφορετικούς κόμβους μπορεί να γραφτεί σε παραπάνω από έναν κόμβους. Περιμένουμε λοιπόν μεγαλύτερο φόρτο δικτύου σε αυτήν την φάση καθώς και περισσότερες γραμμές επεξεργασίας στην εφαρμογή του FGP στη μετέπειτα φάση.

Στο 2ο σενάριο κάθε εγγραφή του log αρχείου περιέχει ένα αντικείμενο άρα θα γραφτεί σε έναν κόμβο. Έπειτα ξεκινά η *φάση της τοπικής εξόρυξης* (βλ. παράγραφο 4.3) που παράγει τα συχνά πρότυπα. Εδώ δεν έχει γίνει καμία προσθήκη/αλλαγή κώδικα.

Στην επόμενη φάση, την *φάση της ανταλλαγής κανόνων/μοτίβων* (βλ. παράγραφο 4.4), ο κάθε συμμετέχων στέλνει σε όλους τους άλλους τους κανόνες που έχει παράγει και λαμβάνει αυτούς που του στέλνουν. Η υλοποίηση είναι κοινή και για τα δύο σενάρια και έχει αλλάξει η μέθοδος που διαβάζει τις πληροφορίες σύνδεσης και εγγραφής μεταξύ των συμμετεχόντων (Socket/BufferedWriter info).

5.3 Μετρήσεις

Για την αξιολόγηση των δύο υλοποιήσεων (ανά σύνοδο και αντικείμενο, ανά αντικείμενο) χρησιμοποιήθηκε ένας μεγάλος υπολογιστής ο οποίος διαθέτει 6 μηχανήματα, κάθε ένα από τα οποία έχει 64 μονάδες εκτέλεσης (cores) σε 8 επεξεργαστές και 256 GB μνήμης. Έγιναν τρία διαφορετικά τρεξίματα για την κάθε υλοποίηση: 1)στο τρέξιμο Α’ χρησιμοποιείται μόνο ένα από τα μηχανήματα, 2)στο τρέξιμο Β’ αξιοποιούνται 5 από αυτά σε δίκτυο υπερυψηλής ταχύτητας (ονομαστική 40Gbit) 3)στο τρέξιμο Γ’ είναι τα 5 αυτά μηχανήματα με εναλλακτική δικτύωση διασύνδεσης και ταχύτητα 1 Gbit. Τα

σενάρια αναφέρονται ως 1ο σενάριο και 2ο σενάριο και αφορούν στις δύο υλοποιήσεις ανά σύνοδο και αντικείμενο και ανά αντικείμενο, αντίστοιχα.

Τα πειράματα έγιναν με πλήθος κόμβων μεταξύ 2 έως 128 για το τρέξιμο Α', και 2 έως 48 για τα τρεξίματα Β' και Γ'. Το πλήθος των πειραμάτων είναι 35 τρεξίματα Α' για κάθε ένα σενάριο, 30 τρεξίματα Β' για κάθε ένα σενάριο και 30 τρεξίματα Γ' για κάθε σενάριο (~5.000 γραμμές αριθμητικών δεδομένων). Τα πειράματα έγιναν με ένα μεγάλο αρχείο δεδομένων μεγέθους 1 GB. Από τα αριθμητικά δεδομένα υπολογίστηκαν οι μέσοι όροι, τα ελάχιστα και τα μέγιστα των εξής αποτελεσμάτων: αρχικές γραμμές κάθε κόμβου, γραμμές μετά την ανταλλαγή βάσει ανάθεσης, χρόνος ανταλλαγής δεδομένων(sec), χρόνος εξόρυξης (sec), χρόνος αναμονής(sec) και χρόνος εκτέλεσης (sec). Στην συνέχεια παρατίθενται πίνακες με τους υπολογισμένους Μέσους όρους, Ελάχιστα και Μέγιστα των αναφερόμενων παραμέτρων και συγκριτικά διαγράμματα.

5.3.1 Μετρήσεις σεναρίου 1ου ανά σύνοδο και αντικείμενο

5.3.1.1 Τρέξιμο Α' – συγκεντρωτικός πίνακας

πλήθος κόμβων		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	1,011,059	3.77	91.48	32.71	127.95
	MIN	606,542	992,178	2.40	58.38	0.00	120.92
	MAX	607,229	1,029,939	6.79	127.91	69.09	134.70
4	AVG	303,443	608,820	2.56	107.71	122.03	232.29
	MIN	302,650	584,517	1.76	51.48	0.00	204.80
	MAX	304,364	627,154	3.96	261.52	208.35	264.27
8	AVG	151,722	289,106	1.57	60.66	246.20	308.42
	MIN	151,180	220,908	1.06	13.35	0.00	267.19
	MAX	152,323	356,436	2.21	375.35	361.81	377.55
16	AVG	75,862	121,762	1.31	23.90	45.45	70.63
	MIN	75,443	83,670	0.69	8.03	0.01	60.89
	MAX	76,421	189,096	1.77	74.49	65.56	76.21
32	AVG	37,931	52,271	1.24	9.15	11.64	21.97
	MIN	37,578	18,659	0.39	4.18	0.02	17.18
	MAX	38,393	161,387	1.84	24.87	20.50	26.39
64	AVG	18,966	23,177	1.74	9.41	23.51	34.44
	MIN	18,724	2,769	0.34	3.59	0.03	31.57
	MAX	19,272	146,717	2.70	33.92	30.30	36.21
128	AVG	9,484	10,609	2.78	9.67	7.49	19.29
	MIN	9,251	1	0.37	3.35	0.10	15.90
	MAX	9,697	139,215	4.60	18.79	14.84	22.73

Εικόνα 8. Σενάριο 1ο τρέξιμο Α'

5.3.1.2 Τρέξιμο Β' – συγκεντρωτικός πίνακας

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	992,954	3.26	135.39	9.55	148.20
	MIN	606,518	990,130	2.22	121.02	0.00	145.59
	MAX	607,253	995,777	4.57	147.48	22.75	150.65
4	AVG	303,443	616,004	2.77	106.19	143.02	251.96
	MIN	303,110	609,911	1.88	33.87	0.00	233.94
	MAX	304,259	618,078	4.17	265.91	230.63	268.71
8	AVG	151,722	290,420	2.43	30.17	54.56	87.16
	MIN	151,301	262,035	1.25	6.59	0.00	73.49
	MAX	152,553	304,613	3.74	90.88	83.64	94.58
16	AVG	75,862	121,479	1.90	18.67	58.24	78.80
	MIN	75,510	68,483	0.85	3.10	0.00	69.83
	MAX	76,376	189,105	3.38	83.84	80.36	85.89
32	AVG	37,931	52,355	1.54	5.58	16.18	23.28
	MIN	37,504	10,381	0.39	2.00	0.00	20.58
	MAX	38,282	161,372	3.04	24.45	22.47	25.75
48	AVG	25,288	32,393	0.97	3.32	5.64	9.88
	MIN	24,909	6,576	0.30	1.80	0.01	7.17
	MAX	25,717	151,643	1.49	11.37	9.58	12.24

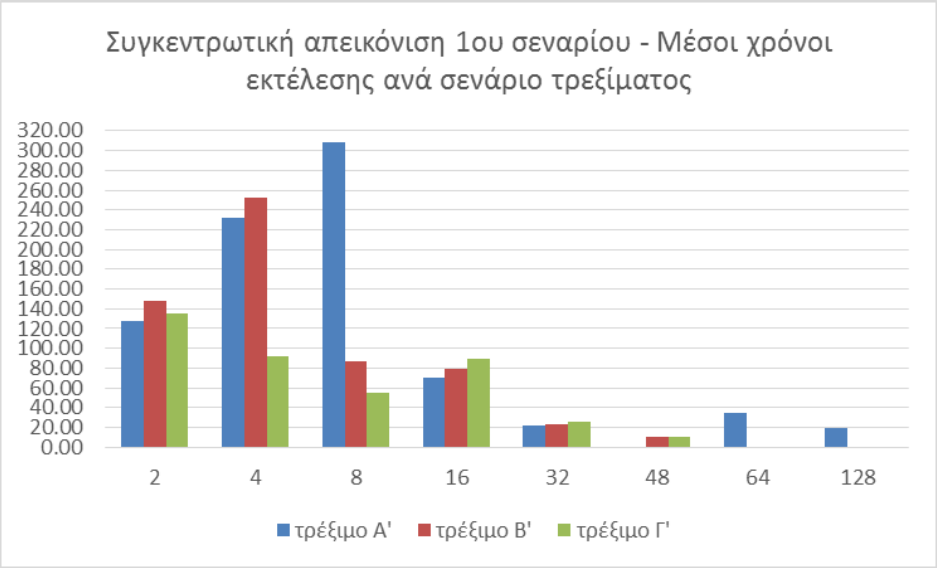
Εικόνα 9. Σενάριο 1ο τρέξιμο Β'

5.3.1.3 Τρέξιμο Γ' – συγκεντρωτικός πίνακας

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	999,216	3.29	129.35	2.82	135.45
	MIN	606,850	981,799	2.39	121.30	0.00	130.90
	MAX	606,921	1,016,633	3.75	134.36	8.68	138.06
4	AVG	303,443	623,353	2.74	63.37	25.47	91.57
	MIN	303,228	603,202	1.93	45.81	0.00	82.72
	MAX	303,653	648,315	4.01	93.36	47.56	97.37
8	AVG	151,722	289,788	2.66	28.41	23.55	54.61
	MIN	151,363	237,152	1.39	8.33	0.00	48.15
	MAX	151,971	336,821	4.71	59.50	51.18	64.20
16	AVG	75,862	121,420	1.71	17.99	69.51	89.19
	MIN	75,229	84,188	0.73	3.56	0.00	84.15
	MAX	76,479	189,375	2.52	94.63	90.71	96.88
32	AVG	37,931	52,294	1.22	4.73	19.33	25.26
	MIN	37,533	13,747	0.30	1.72	0.00	16.82
	MAX	38,432	161,307	2.24	34.25	32.33	35.86
48	AVG	25,288	32,371	1.08	3.25	5.80	10.07
	MIN	24,917	6,596	0.23	1.39	0.00	7.23
	MAX	25,600	151,468	1.68	9.97	8.37	11.19

Εικόνα 10. Σενάριο 1ο τρέξιμο Γ'

5.3.1.4 Συγκεντρωτικό διάγραμμα Μέσων Χρόνων εκτέλεσης



Εικόνα 11. Συγκεντρωτική απεικόνιση Μέσων χρόνων εκτέλεσης 1ου σεναρίου

5.3.2 Μετρήσεις σεναρίου 2ου ανά αντικείμενο

5.3.2.1 Τρέξιμο Α' – Συγκεντρωτικός πίνακας

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	606,886	2.34	20.36	1.53	24.22
	MIN	606,207	595,560	2.21	18.21	0.00	22.63
	MAX	607,564	618,211	2.52	22.80	4.05	25.15
4	AVG	303,443	303,443	2.07	11.20	7.35	20.62
	MIN	303,054	295,271	1.91	7.33	0.00	18.10
	MAX	304,019	309,953	2.24	21.94	14.09	24.18
8	AVG	151,722	151,722	1.53	6.39	3.15	11.06
	MIN	151,093	98,808	1.45	4.25	0.00	10.57
	MAX	152,354	192,329	1.62	10.51	5.84	12.10
16	AVG	75,862	75,862	1.05	4.41	12.35	17.80
	MIN	75,171	49,166	1.01	1.96	0.02	17.26
	MAX	76,291	131,365	1.10	17.49	15.15	18.61
32	AVG	37,931	37,931	1.29	4.37	8.97	14.51
	MIN	37,547	9,210	1.08	2.38	0.02	11.21
	MAX	38,430	131,365	1.53	16.50	13.82	17.82
64	AVG	18,966	18,966	1.69	5.17	14.19	20.80
	MIN	18,597	2,594	1.40	2.54	0.04	19.08
	MAX	19,305	131,365	1.88	20.37	17.51	22.10
128	AVG	9,484	9,484	3.26	5.78	4.95	13.46
	MIN	9,230	55	2.51	1.35	0.12	11.54
	MAX	9,728	131,365	4.00	10.98	9.66	14.62

Εικόνα 12. Σενάριο 2ο Τρέξιμο Α'

5.3.2.2 Τρέξιμο Β' – Συγκεντρωτικός πίνακας

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	606,886	2.44	30.53	15.74	48.70
	MIN	605,967	606,617	2.01	13.29	0.00	46.84
	MAX	607,804	607,154	2.70	51.24	36.27	53.25
4	AVG	303,443	303,443	2.05	9.32	2.77	14.13
	MIN	302,247	294,218	1.86	6.85	0.00	13.09
	MAX	304,125	312,104	2.16	12.69	5.63	14.79
8	AVG	151,722	151,722	1.45	11.64	48.09	61.16
	MIN	151,025	99,513	1.42	2.99	0.00	56.78
	MAX	152,422	182,683	1.48	65.54	61.90	66.97
16	AVG	75,862	75,862	1.03	5.77	26.56	33.33
	MIN	75,318	26,641	0.96	1.83	0.00	29.48
	MAX	76,373	131,365	1.09	34.94	33.10	35.97
32	AVG	37,931	37,931	0.99	6.08	65.79	72.74
	MIN	37,573	7,468	0.87	1.13	0.00	61.84
	MAX	38,325	131,365	1.23	104.50	103.13	105.36
48	AVG	25,288	25,288	1.06	6.53	104.70	112.08
	MIN	24,862	3,364	0.82	1.00	0.00	89.07
	MAX	25,558	131,365	1.23	176.86	175.75	177.92

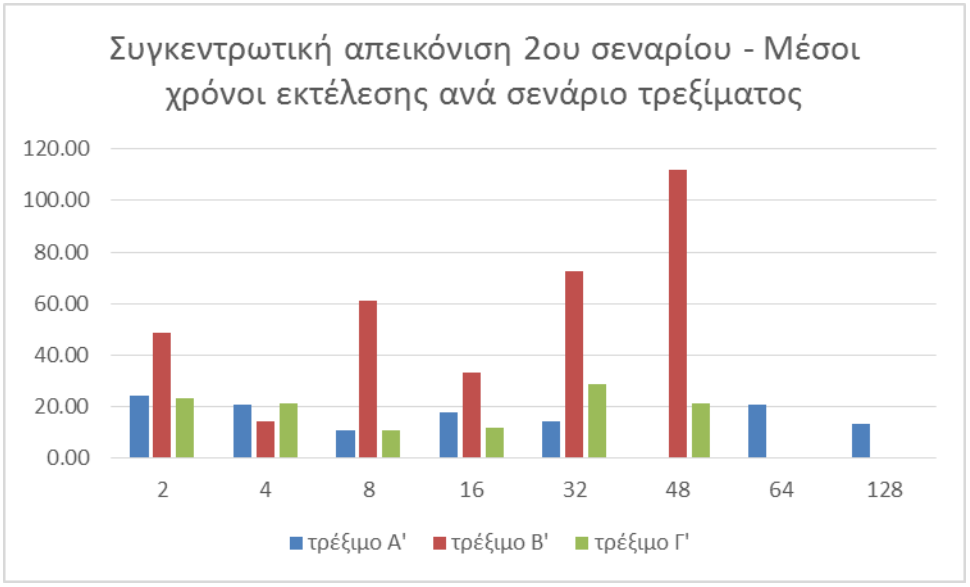
Εικόνα 13. Σενάριο 2ο Τρέξιμο Β'

5.3.2.3 Τρέξιμο Γ' – Συγκεντρωτικός πίνακας

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	606,886	2.78	17.12	3.52	23.40
	MIN	606,190	606,195	2.73	12.61	0.00	19.89
	MAX	607,581	607,576	2.83	25.14	12.13	27.96
4	AVG	303,443	303,443	2.01	10.59	8.93	21.51
	MIN	302,970	288,722	1.74	5.89	0.00	17.13
	MAX	304,383	317,270	2.16	23.01	16.04	24.75
8	AVG	151,722	151,722	1.45	5.68	3.91	11.02
	MIN	151,253	131,680	1.41	3.10	0.00	9.09
	MAX	152,399	164,439	1.47	11.47	7.98	12.92
16	AVG	75,862	75,862	1.07	3.68	7.21	11.92
	MIN	75,393	35,693	0.99	1.75	0.00	8.70
	MAX	76,188	131,365	1.14	14.44	12.66	15.53
32	AVG	37,931	37,931	1.09	4.50	23.49	28.95
	MIN	37,456	7,790	0.78	1.14	0.01	22.69
	MAX	38,331	131,365	1.40	34.95	33.65	36.10
48	AVG	25,288	25,288	1.07	3.09	17.47	21.42
	MIN	24,979	3,653	0.70	1.09	0.00	17.94
	MAX	25,813	131,365	1.51	23.38	22.24	24.26

Εικόνα 14. Σενάριο 2ο Τρέξιμο Γ'

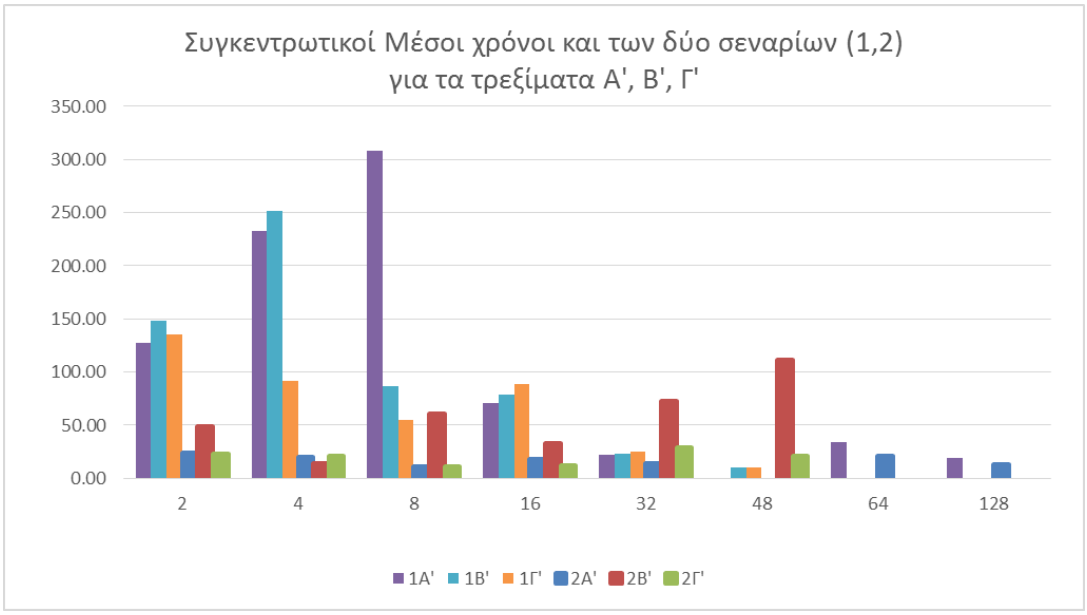
5. 3.2.4 Συγκεντρωτικό διάγραμμα Μέσων Χρόνων εκτέλεσης



Εικόνα 15. Συγκεντρωτική απεικόνιση Μέσων χρόνων εκτέλεσης 2ου σεναρίου

5.3.3 Συγκριτικά διαγράμματα δύο σεναρίων

5.3.3.1 Συγκριτικό διάγραμμα Μέσων χρόνων εκτέλεσης



Εικόνα 16. Συγκριτικό διάγραμμα Μέσων χρόνων εκτέλεσης

5.3.4 Παρατηρήσεις

Όσον αφορά την απόδοση μέσων χρόνων εκτέλεσης παρατηρούμε πως όσο αυξάνονται οι κόμβοι οι Μέσοι χρόνοι μειώνονται και για τα δύο σενάρια. Για λίγους κόμβους (<16) είναι αισθητή η κακή

απόδοση του 1ου σεναρίου (ανάθεση ανά σύνοδο και αντικείμενο) που όμως υπερτερεί εάν οι κόμβοι είναι περισσότεροι από 16.

Όσον αφορά τη δικτύωση, παρόλο που τα τρεξίματα Β' έχουν υψηλή ονομαστική ταχύτητα (40Gb) δεν δείχνει να υπερτερούν στους χρόνους σε σχέση με τα τρεξίματα Γ' μιας τυπικής δικτύωσης 1Gb. Αυτό που φαίνεται είναι πως η ονομαστική ταχύτητα είναι καλή, αλλά η υλοποίηση δεν αξιοποιεί τις δυνατότητες του υλικού.

Για τα τρεξίματα Β' και Γ' του 2ου σεναρίου (5 μηχανήματα δυο διαφορετικές δικτυώσεις), παρατηρούμε ότι σε κάποια τρεξίματα (παρακάτω εικόνα) ένας (τουλάχιστον) κόμβος κάνει πολύ χρόνο για να πραγματοποιήσει την εξόρυξη και αναγκάζει τους άλλους σε αναμονή οπότε και σε περισσότερο χρόνο επεξεργασίας. Το πλήθος των γραμμών του αρχείου του δεν δικαιολογεί το μεγάλο χρόνο επεξεργασίας που του παίρνει. Τα τρεξίματα Β' και Γ' χρησιμοποιούν πέντε μηχανήματα οπότε η καθυστέρηση δεν φαίνεται να οφείλεται στο ότι όλοι πάνε να διαβάσουν από τον ίδιο δίσκο.

peers		αρχικές γραμμές	γραμμές μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
2	AVG	606,886	606,886	2.44	30.53	15.74	48.70
	MIN	605,967	606,617	2.01	13.29	0.00	46.84
	MAX	607,804	607,154	2.70	51.24	36.27	53.25
4	AVG	303,443	303,443	2.05	9.32	2.77	14.13
	MIN	302,247	294,218	1.86	6.85	0.00	13.09
	MAX	304,125	312,104	2.16	12.69	5.63	14.79
8	AVG	151,722	151,722	1.45	11.64	48.09	61.16
	MIN	151,025	99,513	1.42	2.99	0.00	56.78
	MAX	152,422	182,683	1.48	65.54	61.90	66.97
16	AVG	75,862	75,862	1.03	5.77	26.56	33.33
	MIN	75,318	26,641	0.96	1.83	0.00	29.48
	MAX	76,373	131,365	1.09	34.94	33.10	35.97
32	AVG	37,931	37,931	0.99	6.08	65.79	72.74
	MIN	37,573	7,468	0.87	1.13	0.00	31.84
	MAX	38,325	131,365	1.23	104.50	103.13	105.36
48	AVG	25,288	25,288	1.06	6.53	104.70	112.08
	MIN	24,862	3,364	0.82	1.00	0.00	99.07
	MAX	25,558	131,365	1.33	176.86	175.75	177.92

1	peerId	αρχικές γραμμές	μετά τη φάση ανταλλαγής	χρόνος ανταλλαγής δεδομένων	χρόνος εξόρυξης συστάσεων	χρόνος αναμονής	χρόνος εκτέλεσης
476	0	25,291	25,312	1.16	1.65	175.22	177.76
477	1	25,161	14,954	1.16	176.86	0.01	177.78
478	2	25,455	24,302	1.16	2.43	174.44	177.80
479	3	25,536	28,593	1.16	1.35	175.51	177.77
480	4	25,352	28,015	1.22	1.70	175.09	177.90
481	5	25,342	6,377	1.16	4.12	172.76	177.78
482	6	25,522	131,365	1.23	2.94	173.86	177.76
483	7	25,183	16,008	1.16	7.43	169.43	177.80
484	8	25,185	5,686	1.16	2.87	174.00	177.80
485	9	25,087	24,177	1.19	2.22	174.59	177.89
486	10	25,196	8,021	1.16	3.01	173.85	177.75
487	11	25,174	28,371	1.17	1.61	175.25	177.77
488	12	25,234	28,269	1.16	1.46	175.39	177.77
489	13	25,321	21,858	1.17	26.48	150.37	177.79
490	14	25,262	9,169	1.20	15.23	161.57	177.90
491	15	25,237	5,818	1.17	5.72	171.14	177.79
492	16	25,338	31,767	1.17	1.62	175.24	177.83
493	17	25,383	51,318	1.17	1.83	175.03	177.76
494	18	25,185	18,026	1.17	2.34	174.50	177.77
495	19	25,101	4,249	1.22	2.40	174.38	177.90
496	20	25,352	10,355	1.17	7.59	169.25	177.74
497	21	25,075	81,342	1.18	2.36	174.48	177.77

Εικόνα 17. Τρέξιμο Β' 2ου σεναρίου – Συγκεντρωτικός πίνακας -Max τιμές

Οι ξαφνικές μέγιστες τιμές στους χρόνους εξόρυξης του 2ου σεναρίου (ανάθεση ανά αντικείμενο) για τα τρεξίματα Β' και Γ' παρουσιάζονται όσο αυξάνονται οι κόμβοι. Αυτό ίσως σημαίνει πως η παράλληλη επεξεργασία στην προσομοίωση οδηγεί κάποιο κόμβο στο να μένει «πίσω» στη χρήση πόρων του συστήματος και έτσι όταν τελικά έρθει η σειρά του, οι άλλοι έχουν ολοκληρώσει και απλά τον περιμένουν. Θα μπορούσαμε να μην κάνουμε χρήση των ακραία μέγιστων τιμών όμως αυτό επηρεάζει όλο το τρέξιμο αφού η ακραία μέγιστη τιμή σε μία εγγραφή χρόνου εξόρυξης ενός κόμβου, τελικά γίνεται ακραία τιμή στο χρόνο αναμονής όλων των υπολοίπων. Πιστεύουμε σε πραγματικές συνθήκες δεν θα

συνέβαινε τέτοια εμφάνιση ακραίων τιμών διότι βασίζονται σε μηδενική κοινοχρησία (shared-nothing). Άρα υπάρχει το θέμα μη αξιοποίησης πλήρως του υλικού.

Μέγιστες τιμές εμφανίζονται και στο 1ο σενάριο (ανάθεση ανά σύνοδο και αντικείμενο) και στα τρία τρεξίματα αλλά μόνο για λίγους κόμβους και δικαιολογούνται από το μεγάλο μέγεθος των αρχείων. Σε περισσότερους κόμβους (>16) η κατανομή των χρόνων εξόρυξης είναι ομαλή οπότε και εξομαλύνεται και ο χρόνος επεξεργασίας.

Όσον αφορά την κατανομή των εγγραφών στους συμμετέχοντες κόμβους, από τις τιμές των δύο πρώτων στηλών των πινάκων των συγκεντρωτικών μετρήσεων, βλέπουμε ότι η στρατηγική ανάθεσης (βλ. παράγραφο 4.1 και 5.2) δεν αποδίδει σε αυξημένο πλήθος κόμβων. Οι μετρήσεις δείχνουν ότι δεν μπορεί να ισοκατανέμει το φόρτο διότι βλέπουμε ότι όσο αυξάνεται ο αριθμός των κόμβων τόσο ανομοιόμορφη γίνεται η κατανομή. Αυτό μπορεί να εξηγηθεί αν σκεφτούμε ότι η λογική του να μην «σπάνε» εγγραφές ενός αντικειμένου σε κόμβους, αλλά ένας να αναλαμβάνει όλες τις αιτήσεις τότε τα πιο πολύ ζητούμενα αντικείμενα που έχουν πολλές εγγραφές θα γεμίσουν τους κόμβους και έπειτα θα μείνουν τα μη συχνά ζητούμενα αντικείμενα να επιμεριστούν.

6. Συμπεράσματα

Στην παρούσα διπλωματική εργασία βασιστήκαμε σε ένα υπάρχον κατανεμημένο σύστημα εξόρυξης συχνών προτύπων ώστε να υλοποιήσουμε δύο νέες προσεγγίσεις που αφορούν σε ποια δεδομένα θα κάνει εξόρυξη ο κάθε συμμετέχων κόμβος. Τα σενάρια ήταν δύο: Σενάριο 1ο) η κάθε αίτηση του κάθε χρήστη τυπικά δρομολογείται σε έναν κόμβο για μία σύνοδό του ανεξάρτητα από τις προηγούμενες (π.χ. ανάλογα με το ποιος είναι ο λιγότερο φορτωμένος εκείνη τη στιγμή) και έτσι μπορούμε να φτιάξουμε τις συνόδους (sessionize) από πριν και να τις διανείμουμε , Σενάριο 2ο) οι κόμβοι παρουσιάζουν εξειδίκευση ανά θέμα, π.χ. ο κόμβος Α έχει τα αντικείμενα (items) που αφορούν στη θεματολογία Α1, ο κόμβος Β αυτά που αφορούν στη θεματολογία Β1 και έτσι οι σύνοδοι φτιάχνονται μετά την ανταλλαγή. Η ποιότητα των κανόνων που εξάγονται, είτε στο ένα είτε στο άλλο σενάριο, είναι αποδεδειγμένη.

Από την πειραματική αξιολόγηση που κάναμε φάνηκε ότι για μεγάλο πλήθος κόμβων παίρνουμε πολύ καλά αποτελέσματα χρόνου επεξεργασίας. Ειδικά για το 1ο σενάριο που κάνει κατασκευή συνόδων πριν την εξόρυξη, σε μεγάλο πλήθος κόμβων δεν υπήρξε κανένα ξαφνικό μέγιστο κάτι που μας οδηγεί στο συμπέρασμα ότι η χρήση του FGP λειτούργησε πιο ομαλά σε αυτά τα αρχεία. Επειδή υπάρχει θέμα με την ισοκατανομή των εγγραφών σε μεγάλο πλήθος κόμβων θα πρέπει να ερευνηθεί η λογική της στρατηγικής ανάθεσης του Arbitrator – διαιτητή. Θα είχε ενδιαφέρον να τρέξουν τα πειράματα του 1ου σεναρίου και σε ένα αρχείο καταγραφής (log file) που να αντανακλά την λογική της συγκέντρωσης των συνόδων σε ένα κόμβο. Τέλος θα έβγαιναν χρήσιμα συμπεράσματα από μια συναξιολόγηση και των τριών υλοποιήσεων, όσον αφορά τόσο την απόδοση όσο και τους κανόνες που εξήχθησαν από την κάθε μια.

7. Βιβλιογραφία

1. Παυλίδης, Γ., *Εφαρμοσμένα Πληροφοριακά Συστήματα ΙΙ*. 2013, Πανεπιστήμιο Πατρών - Τμήμα Μηχανικών Η/Υ και Πληροφορικής: Πάτρα.

2. Dennett, D.C., *Real patterns.* The journal of Philosophy, 1991. **88**(1): p. 27-51.

3. Ρήγκου, Μ., *Αποδοτικοί Αλγόριθμοι Εξατομίκευσης βασισμένοι σε εξόρυξη γνώσης από δεδομένα χρήσης web*. 2005, Πανεπιστήμιο Πατρών - Πολυτεχνική Σχολή: Πάτρα.

4. Wikipedia. *1% rule (Internet culture)*. Available from: https://en.wikipedia.org/wiki/1%25_rule_%28Internet_culture%29. (14/04/2016)

5. Mulvenna, M.D., S.S. Anand, and A.G. Büchner, *Personalization on the Net using Web mining: introduction.* Communications of the ACM, 2000. **43**(8): p. 122-125.

6. Seroussi, Y. *The wonderful world of recommender systems*. 2015; Available from: https://yanirseroussi.com/tag/recommender-systems/. (10/04/2016)

7. Kosala, R. and H. Blockeel, *Web mining research: A survey.* ACM Sigkdd Explorations Newsletter, 2000. **2**(1): p. 1-15.

8. Baglioni, M., et al., *Preprocessing and mining web log data for web personalization*, in *AI* IA 2003: Advances in Artificial Intelligence*. 2003, Springer. p. 237-249.

9. Giannikopoulos, P. and C. Vassilakis, *A distributed recommender system architecture.* International Journal of Web Engineering and Technology, 2012. **7**(3): p. 203-227.

10. Han, J., et al., *Mining frequent patterns without candidate generation: A frequent-pattern tree approach.* Data mining and knowledge discovery, 2004. **8**(1): p. 53-87.

11. Jiang, T. and A.-H. Tan. *Mining rdf metadata for generalized association rules: knowledge discovery in the semantic web era*. in *Proceedings of the 15th international conference on World Wide Web*. 2006. ACM.

12. Jiang, T., A.-H. Tan, and K. Wang, *Mining generalized associations of semantic relations from textual web content.* Knowledge and Data Engineering, IEEE Transactions on, 2007. **19**(2): p. 164-179.

13. Chakrabarti, S., et al., *Data mining curriculum: A proposal (Version 1.0).* Intensive Working Group of ACM SIGKDD Curriculum Committee, 2006.

14. Gabrilovich, E., S. Dumais, and E. Horvitz. *Newsjunkie: providing personalized newsfeeds via analysis of information novelty*. in *Proceedings of the 13th international conference on World Wide Web*. 2004. ACM.

15. Agrawal, R., T. Imielinski, and A. Swami, *Database mining: A performance perspective.* Knowledge and Data Engineering, IEEE Transactions on, 1993. **5**(6): p. 914-925.

16. Ευθαλία, Κ.-Π., *Αλγόριθμοι Εξόρυξης Χωρικών Δεδομένων - Εφαρμογή σε Αλγόριθμους Συσταδοποίησης* in *Σχολή Αγρονόμων και Τοπογράφων Μηχανικών*. 2006, Εθνικό Μετσόβιο Πολυτεχνείο: Αθήνα.

17. Srikant, R. and R. Agrawal, *Mining generalized association rules*. 1995: IBM Research Division.

18. Giannikopoulos, P., *Recommender Systems Over Distributed Architectures*, in *Computer Science and Technology*. 2012, University of Peloponnese: Tripolis.

19. Agrawal, R. and R. Srikant. *Fast algorithms for mining association rules*. in *Proc. 20th int. conf. very large data bases, VLDB*. 1994.

20. Giannikopoulos, P., I. Varlamis, and M. Eirinaki. *Mining frequent generalized patterns for Web personalization*. in *Proceedings of the Workshop on Mining Social Data, 18th European Conference on Artificial Intelligence*. 2008.

21. Giannikopoulos, P., I. Varlamis, and M. Eirinaki, *Mining frequent generalized patterns for web personalization in the presence of taxonomies.* Exploring Advances in Interdisciplinary Data Mining and Analytics: New Trends: New Trends, 2011: p. 52.

22. Kirk, P. *Gnutella Protocol Development* 2003; Available from: http://rfc-gnutella.sourceforge.net/. (18/04/2016)

23. Stoica, I., et al., *Chord: A scalable peer-to-peer lookup service for internet applications.* ACM SIGCOMM Computer Communication Review, 2001. **31**(4): p. 149-160.

24. Ratnasamy, S., et al., *A scalable content-addressable network*. Vol. 31. 2001: ACM.

25. Aberer, K., et al. *Indexing data-oriented overlay networks*. in *Proceedings of the 31st international conference on Very large data bases*. 2005. VLDB Endowment.

26. Nejdl, W., et al. *EDUTELLA: a P2P networking infrastructure based on RDF*. in *Proceedings of the 11th international conference on World Wide Web*. 2002. ACM.

27. Idreos, S., et al. *Query processing in super-peer networks with languages based on information retrieval: the p2p-diet approach*. in *Current Trends in Database Technology-EDBT 2004 Workshops*. 2004. Springer.

28. Mobasher, B., *Data mining for web personalization*, in *The adaptive web*, B. Peter, K. Alfred, and N. Wolfgang, Editors. 2007, Springer-Verlag. p. 90-135.

29. Tsatsaronis, G., I. Varlamis, and M. Vazirgiannis. *Word sense disambiguation with semantic networks*. in *Text, Speech and Dialogue*. 2008. Springer.

30. Banos, E., et al., *PersoNews: a personalized news reader enhanced by machine learning and semantic filtering*, in *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*. 2006, Springer. p. 975-982.

31. Lam, X.N., et al. *Addressing cold-start problem in recommendation systems*. in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. 2008. ACM.

32. Schein, A.I., et al. *Methods and metrics for cold-start recommendations*. in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 2002. ACM.

33. Eirinaki, M., M. Vazirgiannis, and I. Varlamis. *Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process*. in *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'03), Washington DC*. 2003.

34. Oberle, D., et al., *Conceptual user tracking*, in *Advances in Web Intelligence*. 2003, Springer. p. 155-164.

35. Middleton, S.E., N.R. Shadbolt, and D.C. De Roure, *Ontological user profiling in recommender systems.* ACM Transactions on Information Systems (TOIS), 2004. **22**(1): p. 54-88.