Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής και Τεχνολογίας Υπολογιστών

Αποδοτικές τεχνικές μεταφοράς και διαχείρισης δεδομένων σε κατανεμημένα συστήματα μεγάλης κλίμακας

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΝΤΩΝΙΟΥ Α. ΖΗΣΙΜΟΥ

Αθήνα, Δεκέμβριος 2011

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής και Τεχνολογίας Υπολογιστών

Αποδοτικές τεχνικές μεταφοράς και διαχείρισης δεδομένων σε κατανεμημένα συστήματα μεγάλης κλίμακας

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΝΤΩΝΙΟΥ Α. ΖΗΣΙΜΟΥ

Συμβουλευτική Επιτροπή: Νεκτάριος Κοζύρης
Βασίλειος Μάγκλαρης
Παναγιώτης Τσανάκας

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 30η Δεκεμβρίου 2011.

. . . . .	. . . . .	. . . . .
Νεκτάριος Κοζύρης	Βασίλειος Μάγκλαρης	Παναγιώτης Τσανάκας
Αναπ. Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ
. . . . .	. . . . .	. . . . .
Τίμος Σελλής	Αντώνιος Δεληγιαννάκης	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής ΕΜΠ	Επικ. Καθηγητής Πολ.Κρήτης	Αναπ. Καθηγητής ΕΜΠ
	. . . . .	
	Νικόλαος Παπασπύρου	
	Επικ. Καθηγητής ΕΜΠ	

Αθήνα, Δεκέμβριος 2011

. . . . . .

ΑΝΤΩΝΙΟΣ Α. ΖΗΣΙΜΟΣ
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

Η εκπόνηση της διατριβής χρηματοδοτήθηκε από το έργο ΠΕΝΕΔ 03ΕΔ267 με τίτλο “Ανάπτυξη και βελτιστοποίηση υπηρεσιών υπολογιστικού πλέγματος”. Το έργο υλοποιήθηκε στο πλαίσιο του Μέτρου 8.3 του Ε.Π. “Ανταγωνιστικότητα” του Γ’ Κοινοτικού Πλαισίου Στήριξης και συγχρηματοδοτήθηκε κατά 80% της δημόσιας δαπάνης από την Ευρωπαϊκή Ένωση – Ευρωπαϊκό Κοινωνικό Ταμείο και 20% της δημόσιας δαπάνης από το Ελληνικό Δημόσιο – Υπουργείο Ανάπτυξης – Γενική Γραμματεία Έρευνας και Τεχνολογίας.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Συντμήσεις

API Application Programming Interface – Προγραμματιστική Διεπαφή Εφαρμογής

CERN European Organization for Nuclear Research – Conseil Européen pour la Recherche Nucléaire – Ευρωπαϊκός Οργανισμός Πυρηνικών Ερευνών

CHK Content Hash Key – Κλειδί Κατακερματισμένου Περιεχομένου

CMS Compact Muon Solenoid

DHT Distributed Hash Table – Κατανεμημένος Πίνακας Κατακερματισμού

DNS Domain Name Service – Υπηρεσία Ονοματοδοσίας

GDMP Grid Data Management Pilot

Giggle GIGa-scale Global Location Engine

GLP Grid Lookup Protocol

LCG Large Handron Collider – Μεγάλος Επιταχυντής Αδρονίων

LDAP Lightweight Directory Access Protocol

LFN Logical File Name – Λογικό όνομα αρχείου

LRC Local Replica Catalogs

P2P Peer to peer – Δίκτυα Ομότιμων Κόμβων ΔΟΚ

PFN Physical File Name – Φυσικό όνομα αρχείου

RLI Replica Location Index – Ευρετήριο Τοποθεσίας Αντιγράφων

RLS Replica Location Service – Υπηρεσία Εύρεσης Αντιγράφων

RPC Remote Procedure Calls – Απομακρυσμένη Κλήση Συνάρτησης

SPOF Single Point Of Failure – Κεντρικό σημείο αστοχίας/βλάβης

SSK Signed Subspace Key – Κλειδί Υπογεγραμμένης Υποπεριοχής

URL Uniform resource locator

UUID Universally Unique Identifiers

VO Virtual Organization – Εικονικός Οργανισμός

Ευχαριστίες

Η παρούσα διδακτορική διατριβή εκπονήθηκε στον Τομέα Τεχνολογίας Πληροφορικής & Υπολογιστών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Περιλαμβάνει την έρευνα και τα αποτελέσματα που προέκυψαν κατά την διάρκεια των μεταπτυχιακών μου σπουδών στο Εργαστήριο Υπολογιστικών Συστημάτων της εν λόγω σχολής. Η ολοκλήρωση της διατριβής ήταν μια πορεία δύσκολη αλλά και συναρπαστική, η οποία δεν θα ήταν δυνατή χωρίς την συμβολή κάποιων ανθρώπων τους οποίους θα ήθελα να ευχαριστήσω.

Αρχικά, θα ήθελα να ευχαριστήσω τον Καθηγητή Νεκτάριο Κοζύρη, επιβλέποντα της διατριβής, για την επιστημονική καθοδήγηση και την ουσιαστική υποστήριξη που μου προσέφερε καθ όλη τη διάρκεια εκπόνησής της, αλλά και για την εμπιστοσύνη και την πίστη που έδειξε στο πρόσωπό μου. Επιπρόσθετα, θα ήθελα να ευχαριστήσω τα υπόλοιπα μέλη της συμβουλευτικής επιτροπής, τους Καθηγητές Βασίλειο Μάγκλαρη και Παναγιώτη Τσανάκα για την σημαντική βοήθεια και τις πηγές έμπνευσης που μου προσέφεραν, οι οποίες ήταν καθοριστικές για την ολοκλήρωση της διατριβής.

Θα ήθελα επίσης, να ευχαριστήσω τον Δημήτρη Τσουμάκο για τις πολύτιμες συμβουλές του, την Κατερίνα Δόκα, την Νάσια Ασίκη και τον Αντώνη Χαζάπη για την δημιουργική συνεργασία μας, καθώς και τα υπόλοιπα μέλη του Εργαστηρίου Υπολογιστικών Συστημάτων, για τις εποικοδομητικές συζητήσεις μας σε ερευνητικά αλλά και φιλοσοφικά θέματα και το φιλικό καθημερινό περιβάλλον που μου προσέφεραν.

Δεν θα μπορούσα να μην αναφερθώ στους στενούς φίλους Δημήτρη, Γιώργο, Γιώργο και Βασίλη με τους οποίους μας συνδέει μια κοινή πορεία από τα προπτυχιακά χρόνια και να τους ευχαριστήσω για τις αξέχαστες αναμνήσεις, τις συζητήσεις και τους προβληματισμούς που μοιραστήκαμε. Εύχομαι να συνεχίσουμε *Citius, Altius, Fortius*.

Για το τέλος άφησα τους σημαντικότερους ανθρώπους στη ζωή μου, την μητέρα μου Ελένη, τον αδερφό μου Γιάννη και τη σύζυγό μου Κάλλια, που μου έδωσαν όλα τα απαραίτητα εφόδια και με στήριξαν με την αγάπη τους σε όλες τις δύσκολες στιγμές.

Αντώνιος Ζήσιμος
Αθήνα, Δεκέμβριος 2011

Περίληψη

Τα τελευταία χρόνια, το Πλέγμα είναι από τα πλέον διαδεδομένα συστήματα τόσο στον ερευνητικό όσο και στο επιχειρηματικό τομέα. Το Πλέγμα είναι ένα κατανεμημένο σύστημα μεγάλης κλίμακας, στο οποίο μπορούν να συνυπάρξουν ένας τεράστιος αριθμός ανεξάρτητων και διαφορετικών υπολογιστικών και αποθηκευτικών πόρων, οι οποίοι ενοποιούνται σε μία υπηρεσιοστρεφή αρχιτεκτονική λογισμικού. Δεδομένου της γεωγραφικά κατανεμημένης φύσης και έκτασης του Πλέγματος, οι υπηρεσίες του θα πρέπει να είναι ικανές να αντεπεξέλθουν σε κλιμάκωση φορτίου αρκετά μεγάλη, ώστε το Πλέγμα να γίνει διαθέσιμο σε παγκόσμια κλίμακα και να απευθυνθεί σε κάθε χρήστη. Ένα από τα πιο κρίσιμα υποσυστήματα στο Πλέγμα, είναι το επίπεδο διαχείρισης δεδομένων. Για την αντιμετώπιση του προβλήματος του τεράστιου μεγέθους των δεδομένων, η κοινότητα ανάπτυξης του Πλέγματος, σχεδίασε την αρχιτεκτονική του Πλέγματος Δεδομένων με τρεις βασικές υπηρεσίες: (α) την Υπηρεσία Μεταφοράς Αρχείων (Data Transfer Service), επιφορτισμένη για την ανταλλαγή δεδομένων μεταξύ των κόμβων του Πλέγματος, (β) την Υπηρεσία Διαχείρισης Αντιγράφων (Replica Location Service), υπεύθυνη για την φύλαξη των φυσικών τοποθεσιών που είναι αποθηκευμένο κάθε αρχείο στο Πλέγμα, και (γ) της Υπηρεσίας Βελτιστοποίησης (Optimization Service), η οποία επιλέγει τις καλύτερες τοποθεσίες για κάθε ανταλλαγή δεδομένων και διαχειρίζεται τα αντίγραφα αρχείων βάσει το ιστορικό χρησιμοποίησης τους. Όμως, οι παραπάνω υπηρεσίες ακολουθούν κεντρικοποιημένη σχεδίαση, η οποία επιφέρει μειωμένες επιδόσεις και κεντρικά σημεία βλάβης. Οι κεντρικοποιημένες υπηρεσίες δεν μπορούν να κλιμακώσουν σε μεγάλο αριθμό ταυτόχρονων χρηστών, ούτε να διατηρήσουν ένα υψηλό αριθμό ανανέωσης σε ένα δυναμικό περιβάλλον όπως αυτό του Πλέγματος. Στην εργασία μας, παρουσιάζουμε μια καινοτόμο αρχιτεκτονική διαχείρισης δεδομένων, η οποία ενοποιεί την υπηρεσία αναζήτησης αντιγράφων και τους μηχανισμούς ανταλλαγής δεδομένων σε ένα πλήρως κατανεμημένο και προσαρμοστικό

σύστημα. Η νέα αυτή αρχιτεκτονική αποτελείται από δύο μέρη τα οποία συνεργάζονται για την αποδοτική διαχείριση δεδομένων: (α) την Κατανεμημένη Υπηρεσία Διαχείρισης Αντιγράφων (Distributed Replica Location Service - DRLS) υπεύθυνη για την φύλαξη των φυσικών τοποθεσιών αποθήκευσης κάθε αρχείου και (β) το GridTorrent επιφορτισμένο με την διαχείριση των ανταλλαγών δεδομένων με αυτόματους μηχανισμούς βελτιστοποίησης. Το DRLS οργανώνει τους κόμβους του συστήματος με ένα Κατανεμημένο Πίνακα Κατακερματισμού (Distributed Hash Table - DHT) και διανέμει την πληροφορία σε όλους τους κόμβους. Το μοναδικό χαρακτηριστικό του DRLS είναι ότι εκτός από την αποκεντρικοποίηση της υπηρεσίας και την κλιμακωσιμότητα που της προσφέρει, υποστηρίζει εγγενώς την ανανέωση της πληροφορίας σε κάθε κόμβο που συμμετέχει στο DHT. Δεδομένου, ότι σε πολλές δυναμικές εφαρμογές τα δεδομένα αλλάζουν συνεχώς, το πρωτόκολλο στο οποίο βασίζεται το DRLS παρουσιάζει ανοχή σε Βυζαντινές συνθήκες σφαλμάτων και εγγυάται συνέπεια. Το GridTorrent είναι ένα πρωτόκολλο εμπνευσμένο από το BitTorrent, που εστιάζει στην βελτιστοποίηση της μεταφοράς δεδομένων σε πραγματικό χρόνο, χωρίς να παραβιάζονται οι αρχές ασφάλειας του Πλέγματος. Η συνεργατική φύση του πρωτοκόλλου, επιτρέπει τη διατήρηση χαμηλής απόκρισης και υψηλής χρησιμοποίησης του δικτύου, ακόμα και σε συνθήκες υψηλού φορτίου. Επιτρέπει μεταφορές δεδομένων από πολλαπλούς αποστολείς σε πολλαπλούς παραλήπτες και μεγιστοποιεί την απόδοση με την ανταλλαγή κομματιών του αρχείου μεταξύ όλων των συμμετεχόντων. Πολύ σημαντικό χαρακτηριστικό της προτεινόμενης αρχιτεκτονικής είναι ότι έχει σχεδιαστεί, ώστε να εκμεταλλευτεί υφιστάμενα και ευρέως χρησιμοποιούμενα πρότυπα στο χώρο του Πλέγματος, ώστε να διατηρεί την συμβατότητα με την υφιστάμενη αρχιτεκτονική και τις αντίστοιχες υλοποιήσεις. Τέλος, για την επαλήθευση των αποτελεσμάτων της εργασίας μας, έχει υλοποιηθεί ένα πρωτότυπο της αρχιτεκτονικής και έχουν γίνει αναλυτικά πειράματα του συστήματος τόσο σε περιβάλλοντα τοπικού δικτύου, όσο και σε περιβάλλοντα μεγάλης κλίμακας και υψηλής δυναμικότητας.

Abstract

In recent years, Grid systems have gained popularity and have been widely utilized within both the research and the business domains. The Grid is a wide-area, large-scale distributed computing system, in which a vast number of remotely located, disjoint and diverse processing and data storage facilities are integrated under a common service-oriented software architecture. Given the geographic dispersion, Grid Services must be scalable enough to cope with extreme load conditions. One of the most critical components in Grid systems is the data management layer. Faced with the problem of managing extremely large scale datasets, the Grid community has proposed the Data Grid architecture, defining a set of basic services. The most fundamental of them are the Data Transfer service, responsible for moving files among grid nodes, the Replica Location service, which keeps track of the physical locations of files and the Optimization service, which selects the best data source for each transfer in terms of completion time and manages the dynamic replica creation/deletion according to file usage statistics. However, all of the aforementioned services heavily rely on centralized mechanisms, which constitute performance bottlenecks and single points of failure. The so far centralized services can neither scale to large numbers of concurrent requests nor keep pace with frequent updates performed in highly dynamic environments. In our work, we introduce a novel data management architecture which integrates the location service with data transfer under a fully distributed and adaptive philosophy. Our scheme comprises of two parts that cooperate to efficiently handle multiple concurrent requests and data transfer: The Distributed Replica Location Service (DRLS) that handles the locating of files and GridTorrent that manages the file transfer and related optimizations. DRLS utilizes a set of nodes that, organized in a DHT, equally share the replica location information. The unique characteristic of the DRLS is that, besides the decentralization and scalability that it offers, it fully supports updates on the

multiple sites of a file that exist in the system. Since in many dynamic applications data locations change rapidly with time, our Byzantine-tolerant protocol guarantees consistency and efficiently handles updates on the various data locations stored, unlike conventional DHT implementations. GridTorrent is a protocol that, inspired by BitTorrent, focuses on real-time optimization of data transfers on the Grid, fully supporting the induced security mechanisms. Based on collaborative sharing, GridTorrent allows for low latency and maximum bandwidth utilization, even under extreme load and flash crowd conditions. It allows transfers from multiple sites to multiple clients and maximizes performance by piece exchange among the participants. A very important characteristic of the proposed architecture is that it is designed to interface and exploit well-defined and deployed Data Grid components and protocols, thus being completely backwards compatible and readily deployable. This work includes an extensive experimental section that contains a real implementation of the system and results over both LAN and WAN environments with highly dynamic and adverse workloads.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Σε ένα κατανεμημένο σύστημα μεγάλης κλίμακας, όπως το Πλέγμα, είναι ιδιαίτερα κρίσιμη η ύπαρξη ιδιαίτερα αποδοτικών μηχανισμών διαχείρισης και μεταφοράς δεδομένων. Στη διατριβή αυτή έχει γίνει μελέτη στη διεθνή βιβλιογραφία για τις σύγχρονες μεθόδους διαχείρισης και μεταφοράς δεδομένων σε αντίστοιχα συστήματα. Πιο συγκεκριμένα έχουν μελετηθεί τεχνικές δομημένων Δικτύων Ομότιμων Κόμβων με Κατανεμημένους Πίνακες Κατακερματισμού, οι οποίες χρησιμοποιούνται για την διαχείριση των αρχείων και των αντιγράφων τους. Επίσης, έχουν μελετηθεί τεχνικές που χρησιμοποιούν κίνητρα συνεργασίας πάνω σε αδόμητα Δίκτυα Ομότιμων Κόμβων, με στόχο την αποδοτικότερη μεταφορά δεδομένων. Με βάση τα παραπάνω, η παρούσα διατριβή προτείνει αποδοτικές τεχνικές μεταφοράς και διαχείρισης δεδομένων για κατανεμημένα συστήματα μεγάλης κλίμακας όπως το Πλέγμα, οι οποίες προσφέρουν τα πλεονεκτήματα της κλιμακωσιμότητας, της αποφυγής του κεντρικού σημείου βλάβης και της ανοχής – βέλτιστης διαχείρισης των πόρων σε απότομες μεταβολές του φορτίου. Όλα τα προτεινόμενα συστήματα αξιολογούνται πειραματικά, ενώ παρουσιάζονται και εφαρμογές που έχουν υιοθετήσει τα συστήματα αυτά, αποδεικνύοντας τη βιωσιμότητά τους συγκρινόμενα με υπάρχουσες λύσεις.

Η συμβολή της διατριβής εντοπίζεται κυρίως στα εξής θέματα:

Αναδεικνύει τα σχεδιαστικά προβλήματα που συναντά κανείς στην αρχιτεκτονική των βασικών υπηρεσιών που απαρτίζουν το Πλέγμα

Προτείνει μια διαφορετική αρχιτεκτονική για την υπηρεσία διαχείρισης αντιγράφων, εμπνευσμένη από το χώρο Δικτύων Ομότιμων Κόμβων που εξαφανίζει το πρόβλημα του κεντρικού σημείου βλάβης και προσφέρει κλιμακωσιμότητα στο σύστημα

Παρουσιάζει νέους μηχανισμούς μεταφοράς δεδομένων, διατηρώντας την συμβατότητα με τις πρότυπες τεχνολογίες του Πλέγματος. Οι μηχανισμοί αυτοί προσφέρουν κλιμακωσιμότητα στο σύστημα, αλλά και τη βέλτιστη διαχείριση των πόρων όταν παρατηρούνται απότομες μεταβολές του φορτίου

Αξιολογεί πειραματικά τα προτεινόμενα συστήματα και παρουσιάζει σενάρια χρήσης για πραγματικές εφαρμογές.

Στο Κεφάλαιο 1 της διατριβής συνοψίζεται το αντικείμενο της διατριβής και ο τρόπος με τον οποίο οργανώνεται. Στο Κεφάλαιο 2 και στο Κεφάλαιο 3 παρουσιάζονται κάποιες βασικές έννοιες από την περιοχή του Πλέγματος και των Δικτύων Ομότιμων Κόμβων ώστε να οριστεί το πλαίσιο στο οποίο υλοποιούνται οι προτεινόμενες τεχνικές.

Στο Κεφάλαιο 4 παρουσιάζεται η υπηρεσία αναζήτησης αντιγράφων των Πλέγματος. Η ανάλυση ξεκινά με μια χρονική ανασκόπηση της υπηρεσίας και εστιάζει στις διαφορετικές ανάγκες που οδήγησαν σε διαφορετικές σχεδιαστικές αποφάσεις. Στην συνέχεια προτείνεται μια λύση κατανεμημένη, που βασίζεται στο παράδειγμα των Δικτύων Ομότιμων Κόμβων, η οποία αντιμετωπίζει με επιτυχία τα υφιστάμενα προβλήματα.

Στο Κεφάλαιο 5 περιγράφονται οι μηχανισμοί μεταφοράς αρχείων στο Πλέγμα. Αναλύεται η υφιστάμενη αρχιτεκτονική, η οποία περιέχει λύσεις κεντρικές, βασισμένες σε προϋπάρχοντα πρωτόκολλα μεταφοράς δεδομένων, με ορισμένες βελτιώσεις ώστε να είναι συμβατά με τις τεχνολογίες πλέγματος. Στην συνέχεια αναδεικνύονται τα αδύνατα σημεία της παραπάνω λύσης και παρουσιάζεται ένας κατανεμημένος μηχανισμός μεταφοράς αρχείων - που ονομάζεται GridTorrent - με χαρακτηριστικά κλιμακωσιμότητας ώστε να ανθίσταται σε φαινόμενα απότομης μεταβολής της κίνησης και εξάρτησης από κεντρικά σημεία βλάβης.

Στο Κεφάλαιο 6 γίνεται μια αναφορά σε εφαρμογές στις οποίες μπορούν να αξιοποιηθούν τα αποτελέσματα της παρούσας διατριβής, ενώ το Κεφάλαιο 7 κλείνει τη διατριβή με τα συμπεράσματα που προκύπτουν και κατευθύνσεις για μελλοντικές επεκτάσεις της.

ΚΕΦΑΛΑΙΟ 2

Πλέγμα

2.1 Εισαγωγή

Τα συστήματα πλέγματος (Grid) είναι από τα πιο διαδεδομένα κατανεμημένα συστήματα μεγάλης κλίμακας. Το πλέγμα σύμφωνα με τον ορισμό που δίνουν οι I. Foster και C. Kesselman [Foster 99, Foster 01], από τους πρωτοπόρους στο χώρο αυτό, αποτελείται από ένα δίκτυο ευρείας περιοχής και ένα μεγάλης κλίμακας κατανεμημένο σύστημα πληροφορικής, στο οποίο απομακρυσμένοι, διασκορπισμένοι και διαφορετικοί υπολογιστικοί και αποθηκευτικοί πόροι ενσωματώνονται σε μια κοινά αποδεκτή υπηρεσιοστραφής αρχιτεκτονική λογισμικού. Στο επίπεδο του υλικού (hardware), το πλέγμα μπορεί να αποτελείται από οποιοδήποτε συσκευή, η οποία μπορεί να συνδεθεί σε ένα δίκτυο και να προσφέρει υπηρεσίες λογισμικού για την χρήση και διαχείρισή της. Απλοί υπολογιστές, συστοιχίες εξυπηρετητών, φάρμες υπολογιστικών συστημάτων, δικτυακοί αποθηκευτικοί χώροι, βιβλιοθήκες αποθηκευτικών ταινιών ή ακόμα και ειδικοί αισθητήρες και επιστημονικά όργανα μπορούν να αποτελέσουν μέρος ενός πλέγματος. Η υποδομή λογισμικού που χρειάζεται – το μεσισμικό (middleware) – αναλαμβάνει τους μηχανισμούς δίκαιου και ασφαλούς διαμοιρασμού των πόρων μεταξύ των χρηστών του συστήματος. Επιπλέον, οι χρήστες οργανώνονται σε κοινότητες με κοινά ενδιαφέροντα και επιδιώξεις. Οι κοινότητες αυτές ονομάζονται εικονικοί οργανισμοί (Virtual Organizations) και είναι θεμελιώδης δομή του πλέγματος, με σκοπό να ενισχύσει τη συνεργασία μεταξύ χρηστών. Έτσι, ενθαρρύνει τους συμμετέχοντες με κοινά ενδιαφέροντα να συνεργαστούν μεταξύ τους, ανεξαρτήτως γεωγραφικής θέσεως, ώστε να επιτύχουν ένα κοινό στόχο. Τέτοια παραδείγματα υπάρχουν αρκετά, ειδικά στον ακαδημαϊκό και

ερευνητικό χώρο, όπου επιστήμονες από ιδρύματα γεωγραφικά κατανεμημένα σε όλο τον κόσμο, χρησιμοποιούν το Πλέγμα για να συνεργαστούν προς ένα κοινό στόχο, ο οποίος είναι συνήθως η μελέτη πειραματικών δεδομένων και η εκτέλεση υπολογιστικών μοντέλων.

Ένα από τα πιο κρίσιμα στοιχεία σε ένα σύστημα Πλέγματος είναι το επίπεδο διαχείρισης δεδομένων. Οι αρχικές ερευνητικές προσπάθειες στο τομέα αυτό [Hoschek 00], ήρθαν αντιμέτωπες με το πρόβλημα της διαχείρισης εξαιρετικά μεγάλων όγκων δεδομένων – της τάξης των πεντάκις εκατομμυρίων bytes (Peta-Bytes), τα οποία θα είναι μοιρασμένα ανάμεσα σε μια πολυπληθή και ετερογενή κοινότητα χρηστών. Απαραίτητο λοιπόν είναι στο σχεδιασμό του συστήματος να ληφθεί πρόνοια για να ικανοποιηθούν αυτές οι ανάγκες με τρόπο που να συμμορφώνονται με τις γενικότερες απαιτήσεις ενός Πλέγματος. Η προτεινόμενη αρχιτεκτονική προσανατολισμένη στη διαχείριση δεδομένων, ονομάστηκε Πλέγμα Δεδομένων [Chervenak 00] και επιτρέπει την πρόσβαση σε κατανεμημένο αποθηκευτικό χώρο και σε ένα μεγάλο όγκο πληροφορίας. Στηρίζεται σε μια ομάδα βασικών υπηρεσιών που διαλειτουργούν και προσφέρουν μια προγραμματιστική διεπαφή (ΠΔ), όμοια με τις κοινές προγραμματιστικές διεπαφές για χειρισμό αρχείων που υπάρχουν στα συνήθη λειτουργικά συστήματα. Σε αυτή την ΠΔ βασίζονται οι υπόλοιπες υπηρεσίες υψηλού επιπέδου, καθώς και οι εφαρμογές των χρηστών. Σε ένα Πλέγμα Δεδομένων, τα δεδομένα μπορεί να είναι οτιδήποτε. Από ένα μικρό αρχείο κειμένου μέχρι ένα μεγάλο βίντεο ή μια τεράστια ομάδα αρχείων που προήλθε από ένα δίκτυο αισθητήρων ή από την εκτέλεση ενός υπολογιστικού μοντέλου.

Βασική υπηρεσία στο επίπεδο διαχείρισης των δεδομένων είναι η ανάκτηση της τοποθεσίας των αντιγράφων (Replica Location Service). Το περιβάλλον του πλέγματος επιβάλλει τα δεδομένα να είναι διασκορπισμένα σε παγκόσμια κλίμακα λόγω των περιορισμών σε αποθηκευτικό χώρο των διάφορων αποθετηρίων δεδομένων, αλλά και λόγω της ίσης απόστασης σε αυτά – στο επίπεδο του δικτύου - από όλους τους χρήστες. Σε τέτοιες περιπτώσεις είναι σύνηθες να χρησιμοποιούνται τοπικά αντίγραφα για να ελαττωθεί ο χρόνος απόκρισης του δικτύου που προστίθεται κάθε φορά που υπάρχει μια λειτουργία απομακρυσμένης πρόσβασης σε δεδομένα. Στην ορολογία του πλέγματος ονομάζουμε αντίγραφο (replica) ένα τοπικό αρχείο το οποίο έχει ιδιότητες ανάγνωσης μόνο (read-only local copies) και αποτελεί αντιγραφή ενός απομακρυσμένου αρχείου που βρίσκεται σε κάποιο αποθετήριο δεδομένων. Αντίστοιχα υπάρχει η υπηρεσία διαχείρισης αντιγράφων στο πλέγμα (Replica Location Service - RLS), η οποία είναι υπεύθυνη για την διαχείριση αυτών των αρχείων. Για την επεξεργασία ενός αρχείου στο πλέγμα, μια εργασία θα πρέπει πρώτα να επικοινωνήσει με το Replica Location Service, ώστε να βρει όλες τις τοποθεσίες που υπάρχει το ζητούμενο αρχείο. Σε περίπτωση που το αρχείο υπάρχει τοπικά, τότε η εργασία μπορεί να χρησιμοποιήσει τις συνηθισμένες λειτουργίες αρχείων για να το επεξεργαστεί, όπως ένα κανονικό αρχείο. Σε περίπτωση που δεν υπάρχει τοπικά, πρέπει να ενεργοποιηθεί ο μηχανισμός μεταφοράς δεδομένων του πλέγματος για να δημιουργηθεί ένα τοπικό αντίγραφο, να ενημερωθεί για την νέα τοποθεσία το Replica Location Service και να προχωρήσει η εργασία με την πρόσβαση στο αρχείο.

Τα τοπικά αντίγραφα αρχείων βοηθούν στην βελτίωση της επίδοσης εφαρμογών που απαιτούν συχνές λειτουργίες πρόσβασης σε απομακρυσμένα δεδομένα. Με την τοποθέτηση αντιγράφων πιο κοντά στο κόμβο εκτέλεσης της εφαρμογής, μειώνεται η συνολική απόκριση της εφαρμογής, αφού μηδενίζεται η απόκριση του δικτύου, ενώ μειώνεται και η συνολική χρήση του. Επιπλέον, μέσω ειδικών αλγορίθμων οι μηχανισμοί μεταφοράς δεδομένων μπορούν να αξιοποιήσουν τα πολλαπλά αντίγραφα για την αύξηση του συνολικού ρυθμού μεταφοράς δεδομένων, αλλά και εργαλεία επανάκτησης δεδομένων μπορούν να χρησιμοποιήσουν τα αντίγραφα για την αποκατάσταση των χαμένων αρχείων.

Σύγχρονες διανομές λογισμικού πλέγματος όπως το Globus Toolkit [Foster 97, Glo] περιλαμβάνουν υπηρεσίες διαχείρισης αντιγράφων (όπως το Replica Location and Management Service), καθώς είναι ιδιαίτερα σημαντικές στην συνολική αρχιτεκτονική του πλέγματος. Οι μηχανισμοί που έχουν χρησιμοποιηθεί από την επιστημονική κοινότητα τα τελευταία χρόνια για τις υπηρεσίες αυτές έχουν εξελιχθεί σημαντικά. Το αρχικό σχέδιο μιας κεντρικής βάσης δεδομένων που είχε αποθηκευμένες πληροφορίες για τις τοποθεσίες όλων των αντιγράφων, παραμερίστηκε προς όφελος μιας κατανεμημένης προσέγγισης. Η πιο διαδεδομένη λύση που χρησιμοποιείται ευρέως στο πλέγμα, γνωστή και ως Giggle Framework [Chervenak 02], ακολουθεί μια πολύ-επίπεδη ιεραρχική δομή και κατανέμει τα δεδομένα και τα ερωτήματα αναζήτησης σε μια πληθώρα ευρετηρίων. Πιο συγκεκριμένα, συντάσσει ένα ενιαίο χώρο ονοματολογίας αρχείων με μοναδικό αναγνωριστικό ανά Εικονικό Οργανισμό γνωστό και ως λογικό όνομα αρχείου (Logical Filename – LFN), το οποίο συνδέει με την φυσική τοποθεσία όλων των αντιγράφων του αρχείου (Physical Filenames – PFN). Ωστόσο, όλες οι μέχρι τώρα υλοποιήσεις σε αυτή την κατεύθυνση βασίζονται στην ύπαρξη ειδικού εξοπλισμού υψηλής διαθεσιμότητας, όπου οι βλάβες υλικού και οι διακοπές στο δίκτυο είναι σπάνιες περιπτώσεις και μπορούν να αντιμετωπιστούν από υλικό που βρίσκεται σε εφεδρεία ή ειδικά συστήματα backup. Αντίθετα, στην πλειοψηφία τους οι υφιστάμενες υλοποιήσεις Πλέγματος χρησιμοποιούνται από εφαρμογές της Ερευνητικής κοινότητας και βασίζονται σε μη εξειδικευμένο εξοπλισμό πληροφορικής, είτε αυτό είναι εξυπηρετητές είτε αυτό είναι σύστημα αποθηκευτικού χώρου.

Πρόσφατες εμπειρίες, όπως η χρησιμοποίηση του Πλέγματος για την οργάνωση πειραμάτων , τα οποία διεξάγονται στο Μεγάλο Επιταχυντή Ανδρονίων (Large Handron Collider) του CERN [lcg], καθορίζουν την διαχείριση δεδομένων ως την μεγαλύτερη προτεραιότητα για τον σχεδιασμό του Πλέγματος Νέας Γενιάς. Το τεράστιο μέγεθος των πειραματικών αποτελεσμάτων, ήδη δημιούργησε την ανάγκη για την βελτίωση των δυνατοτήτων του Πλέγματος σε μηχανισμούς αποθήκευσης και εκμετάλλευσης της υπολογιστικής ισχύος. Για την περαιτέρω βελτίωση της επεκτασιμότητας του Πλέγματος, θα πρέπει να απαλειφθούν οι ιεραρχικές δομές των κεντρικών υπηρεσιών και να υιοθετηθούν κατανεμημένοι αλγόριθμοι που ήδη λειτουργούν σε συστήματα παραγωγής. Σε ένα δίκτυο εκατομμυρίων και δισεκατομμυρίων κόμβων, μια δημοφιλής υπηρεσία

κατανεμημένη σε δεκάδες ή εκατοντάδες κόμβους μπορεί και πάλι να παρουσιάζει τα μειονεκτήματα ενός κεντρικού πόρου, θέτοντας στενωπούς και υποβιβάζοντας την συνολική απόδοση του συστήματος. Επομένως, ο επανασχεδιασμός των υπηρεσιών του Πλέγματος και η εκμετάλλευση κατανεμημένων δομών και αλγορίθμων από το ερευνητικό πεδίο των Δικτύων Ομότιμων Κόμβων (ΔΟΚ) θα οδηγήσει σε βελτιστοποιημένες και επεκτάσιμες υπηρεσίες. Υπηρεσίες που βασίζονται σε μια υποδομή ΔΟΚ μπορούν να επεκταθούν χωρίς περαιτέρω ρυθμίσεις σε δισεκατομμύρια ταυτόχρονους χρήστες. Αυτό είναι και το πλεονέκτημα των ΔΟΚ, καθώς έχουν σχεδιαστεί με τέτοιο τρόπο, ώστε η δυναμική τους να αυξάνεται με τον αριθμό των συμμετεχόντων, σε αντίθεση με το παραδοσιακό μοντέλο πελάτη-εξυπηρετητή, όπου η συνολική απόδοση υποβαθμίζεται όσο ολοένα και περισσότεροι πελάτες ανταγωνίζονται μεταξύ τους για τους περιορισμένους πόρους ενός εξυπηρετητή. Ο στόχος του Πλέγματος είναι στο κοντινό μέλλον να μπορεί να έχει ανεξάρτητους διασυνδεδεμένους υπολογιστές της τάξης των εκατομμυρίων, οι οποίοι να συνεισφέρουν τον χρόνο που παραμένουν αχρησιμοποίητοι προς όφελος των επεξεργαστικών αναγκών της επιστημονικής κοινότητας. Απαραίτητο στοιχείο για να επιτευχθεί αυτός ο στόχος είναι η υποδομή Πλέγματος να παρέχει υπηρεσίες, οι οποίες θα μπορούν να επεκταθούν σε μεγέθη πολλαπλάσια των υφιστάμενων υλοποιήσεων.

2.1.1 Βασικά χαρακτηριστικά

Μια υποδομή πλέγματος οφείλει να παρέχει συγκεκριμένες τεχνικές δυνατότητες, μερικές από τις οποίες είναι [Foster 05]:

Μοντελοποίηση Θα πρέπει να υπάρχει ένα μοντέλο που να περιγράφει τους διαθέσιμους πόρους, τις λειτουργίες τους και τις σχέσεις μεταξύ τους, ώστε να διευκολυνθεί η εύρεση των πόρων, η ορθολογική κατανομή τους και η διαχείρισή της ποιότητας υπηρεσίας που προσφέρουν.

Παρακολούθηση/ειδοποιήσεις Θα πρέπει να υπάρχει πρόσβαση στους μηχανισμούς παρακολούθησης της κατάστασης των πόρων, ενώ θα υποστηρίζονται ειδοποιήσεις προς τις εφαρμογές και τις υπηρεσίες υποδομής για αλλαγές στην κατάσταση των πόρων. Έτσι γίνεται εφικτή η διαχείριση συμβολαίων ποιότητας υπηρεσίας, η καταγραφή των παραπάνω αλλαγών κρίνεται απαραίτητη για την υποστήριξη λογιστικών υπηρεσιών υποδομής και ελέγχου.

Διαθεσιμότητα πόρων Διασφάλιση της ποιότητας υπηρεσίας μιας ομάδας πόρων για όλη τη διάρκεια χρήσης τους από μια εφαρμογή. Σε αυτό βοηθάει η ύπαρξη μηχανισμών διαπραγμάτευσης των επιθυμητών επιπέδων ποιότητας υπηρεσίας. Η διαθεσιμότητα των πόρων αυτών διασφαλίζεται με κάποιας μορφής μηχανισμού κρατήσεων και με την δυναμική δημιουργία συμβολαίων ποιότητας υπηρεσίας.

Διαχείριση πόρων Τέτοιοι μηχανισμοί θα αναθέτουν πόρους σε εφαρμογές ανάλογα με την διαθεσιμότητά τους και τις κρατήσεις του συστήματος. Οι πόροι θα πρέπει να ρυθμίζονται

αυτόματα και κατάλληλα για την εφαρμογή, να παρακολουθούνται για την ορθή λειτουργία τους για όλη τη διάρκεια της δέσμευσης τους από την εφαρμογή και να ρυθμίζονται πάλι αυτόματα στην πρότερη κατάσταση αφού αποδεσμευτούν από αυτή.

Λογιστικές υπηρεσίες υποδομής και ελέγχου Η χρησιμοποίηση των κοινών πόρων από μια εφαρμογή ή ένα χρήστη θα πρέπει να ανιχνεύεται και να χρεώνεται. Το κόστος τέτοιων πόρων πρέπει να το επωμίζεται ο αντίστοιχος χρήστης ή κοινότητα χρηστών.

Ασφάλεια Θα πρέπει να παρέχονται μηχανισμοί ασφάλειας, όπως κρυπτογράφηση, ταυτοποίηση και έλεγχος παραποίησης των δεδομένων ή/και των καναλιών επικοινωνίας.

2.1.2 Κατηγορίες πλέγματος

Οι πιο διαδεδομένες κατηγορίες Πλεγμάτων, τις οι οποίες αναφέραμε και παραπάνω, είναι τα Υπολογιστικά Πλέγματα και τα Πλέγματα Δεδομένων. Όμως, υπάρχουν και άλλες κατηγορίες [Stockinger 07] που διαφέρουν από αυτές τις δυο δημοφιλείς, κυρίως ως προς τον τρόπο χρήσης και λειτουργίας τους.

Collaboration Grids (Πλέγματα Συνεργασίας) Σε αυτά συνυπάρχουν πολλές διαφορετικές και ανεξάρτητες οντότητες από οργανισμούς μέχρι φυσικά πρόσωπα, υπάρχουν διάφορα επίπεδα ασφάλειας, πρωτόκολλα και μηχανισμοί εύρεσης πόρων. Τα κύρια χαρακτηριστικά τους είναι:

- Ευρεία κατανομή με πολλούς εικονικούς οργανισμούς
- Συμβόλαια παροχής υπηρεσίας και εμπορικές συμφωνίες
- Απώτερο όφελος η μεγιστοποίηση των οικονομικών προσόδων

Enterprise Grids (Επιχειρηματικά Πλέγματα) Αυτά έχουν αρκετές ομοιότητες με την προηγούμενη κατηγορία, καθώς είναι το ίδιο τεχνικά πολύπλοκα. Η κεντρική διαφορά τους είναι πως οι πολλαπλές διαφορετικές οντότητες δεν υφίστανται, αλλά στην ουσία πρόκειται για λιγότερες και πολύ πιο στενά συνδεδεμένες. Κατά κύριο λόγο αυτά υποστηρίζουν υπηρεσίες παραγωγής σε μεγάλα μηχανογραφικά κέντρα. Τα κύρια χαρακτηριστικά τους είναι:

- Η εικονοποίηση (virtualization) των πόρων και των εφαρμογών
- Η συγκέντρωση της διαχείρισης σε κεντρικό σημείο
- Απώτερο όφελος η μείωση του συνολικού κόστους κατοχής και λειτουργίας

Cluster Grid (Πλέγματα Συστοιχιών) Αυτά στοχεύουν στην υψηλή υπολογιστική επίδοση και απόδοση και στην πλειονότητά τους είναι απλά περιβάλλοντα χρονοπρογραμματισμού. Τείνουν να έχουν στατικές ιδιότητες, σε αντίθεση με τις παραπάνω κατηγορίες που είναι αρκετά

δυναμικές από τη φύση τους. Οι υπηρεσίες που προσφέρουν είναι συνήθως αρκετά γενικές όπως ένα σύστημα χρονοδρομολόγησης εργασιών και δεν υποστηρίζουν όλες τις φάσεις μιας εφαρμογής, με αποτέλεσμα την απαίτηση για επιπλέον εργασία από τον χρήστη στις αντίστοιχες φάσεις.

2.1.3 Αρχιτεκτονική του επιπέδου δεδομένων

Το επίπεδο Δεδομένων σε ένα Πλέγμα επιτρέπει τον διαμοιρασμό των διάφορων αποθηκευτικών και υπολογιστικών πόρων από εγκαταστάσεις σε όλη την υφήλιο, χρησιμοποιώντας εξειδικευμένο λογισμικό, το οποίο δημιουργεί μια κοινή Προγραμματιστική Διεπαφή - ΠΔ (Application Programming Interface - API) ανεξάρτητη του υλικού. Οι εφαρμογές που μπορούν να ωφεληθούν από την αρχιτεκτονική του Πλέγματος κυμαίνονται από εφαρμογές με απαιτήσεις σε μεγάλη επεξεργαστική ισχύ μέχρι εφαρμογές που απαιτούν αρχειοθέτηση και διαχείριση τεραστίου όγκου δεδομένων. Για το λόγο αυτό δύο βασικές υπηρεσίες του Πλέγματος είναι το Υπολογιστικό Στοιχείο (Computing Element) και το Αποθηκευτικό Στοιχείο (Storage Element). Το μεν πρώτο προσφέρει πρόσβαση σε επεξεργαστική ισχύ και το δε δεύτερο προσφέρει πρόσβαση σε αποθηκευτικό χώρο. Οι χρήστες και οι διάφορες εφαρμογές του Πλέγματος χρησιμοποιούν τις παραπάνω υπηρεσίες για ικανοποιήσουν τις ανάγκες τους με τους κατάλληλους μηχανισμούς που προσφέρουν ασφάλεια και ποιότητα υπηρεσίας. Οι μηχανισμοί αυτοί εξασφαλίζουν ότι θα υπάρχει ταυτοποίηση, πιστοποίηση, εξουσιοδότηση και έλεγχος για την χρησιμοποίηση των πόρων ανά πάσα στιγμή, ενώ υπάρχουν και οι κατάλληλες πολιτικές που εξασφαλίζουν την προτεραιότητα του κάθε χρήστη για την χρήση των πόρων.

Η δομή των υπηρεσιών του Πλέγματος που είναι αναγκαίες για την υλοποίηση ενός σχήματος διαχείρισης δεδομένων, το οποίο θα είναι επεκτάσιμο σε παγκόσμια κλίμακα, περιγράφεται από τους συγγραφείς της Αρχιτεκτονικής Πλέγματος Δεδομένων (Data Grid Architecture) [Chervenak 00]. Όπως έχουμε ήδη αναφέρει, ο όρος Πλέγμα Δεδομένων περιγράφει μια κατανεμημένη υποδομή μεγάλης κλίμακας που αποτελείται από ετερογενή αποθηκευτικά συστήματα και είναι ικανή να διαχειριστεί τεράστιο όγκο δεδομένων. Βασική προϋπόθεση της αρχιτεκτονικής του Πλέγματος Δεδομένων είναι η ουδετερότητα σε σχέση με εξειδικευμένα συστήματα αποθηκευτικού χώρου και μηχανισμούς ανάκτησης δεδομένων, καθώς και η δυνατότητα των χρηστών να ορίσουν οι ίδιοι πολιτικές ποιότητας υπηρεσίας χωρίς να χρειάζονται να προσαρμοστούν σε στατικές πολιτικές χρήσης καθορισμένες από την υποδομή. Στην πράξη αυτό σημαίνει ότι τα δεδομένα πρέπει να είναι προσβάσιμα με τρόπο ανεξάρτητο από το μέσο που χρησιμοποιείται και το υλικό στο οποίο βρίσκονται. Το Πλέγμα Δεδομένων πρέπει να επιτρέπει την αποθήκευση δεδομένων σε οποιοδήποτε συνδυασμό συστημάτων αρχείων, βάσεων δεδομένων και αποθηκευτικού συστήματος. Η αρχιτεκτονική της υπηρεσίας πρέπει να διαχειρίζεται τις εξειδικευμένες λειτουργίες κάθε συστήματος που χρησιμοποιείται και να προσφέρει μια κοινή διεπαφή για την πρόσβαση,

την αποθήκευση, την μεταφορά και την αναζήτηση στα δεδομένα. Επιπλέον, η αρχιτεκτονική του Πλέγματος Δεδομένων πρέπει να σχεδιαστεί με τρόπο τέτοιο, ώστε όλες οι διαθέσιμες παράμετροι προς διαμόρφωση που αναφέρονται στην απόδοση του συστήματος, να μπορούν να ρυθμιστούν από υπηρεσίες υψηλού επιπέδου ή τους ίδιους τους χρήστες, ώστε να είναι υπεύθυνοι για τις πολιτικές ποιότητας υπηρεσίας που θα εφαρμοστούν.

Η αρχιτεκτονική του Πλέγματος Δεδομένων βασίζεται σε δύο οριζόντια επίπεδα. Το χαμηλό επίπεδο περιλαμβάνει βασικά συστατικά της αρχιτεκτονικής, τα οποία προσφέρουν μηχανισμούς διαχείρισης δεδομένων και μετά-δεδομένων.

Υπηρεσίες Δεδομένων Εδώ περιλαμβάνονται όλοι οι μηχανισμοί που χρειάζονται για την ανάγνωση, εγγραφή δεδομένων, καθώς και γενικών πληροφοριών σχετικά με τα αρχεία που είναι αποθηκευμένα στο Πλέγμα. Το αρχείο θεωρείται σαν το βασικό στοιχείο πληροφορίας που μπορεί να αποθηκευθεί στο Πλέγμα. Τα αρχεία βρίσκονται σε διάφορα αποθηκευτικά συστήματα, γεωγραφικά διασκορπισμένα και κάθε αρχείο το συνοδεύει πληροφορία επιπλέον του ονόματος, όπως ημερομηνία δημιουργίας και τροποποίησης, μέγεθος και λίστες πρόσβασης. Τα αποθηκευτικά συστήματα μπορεί να είναι, συνηθισμένα συστήματα αρχείων, βάσεις δεδομένων σε σκληρούς δίσκους ή μαγνητικές ταινίες ή ακόμα και εξειδικευμένο λογισμικό κατανεμημένης αρχιτεκτονικής που διανέμει τα δεδομένα σε μια συστοιχία εξυπηρετητών. Οι Υπηρεσίες Δεδομένων του Πλέγματος προσφέρουν εργαλεία για την μεταφορά δεδομένων μεταξύ των αποθηκευτικών συστημάτων ή και μεταξύ αυτών και των χρηστών. Επίσης, παρακολουθούν τα διάφορα χαρακτηριστικά του κάθε αποθηκευτικού συστήματος και να τα προσφέρουν με ένα ενιαίο τρόπο προς τους χρήστες ή άλλες υπηρεσίες ανώτερου επιπέδου.

Υπηρεσίες Μεταδεδομένων Ανάλογες υπηρεσίες προσφέρονται για την διαχείριση των μετα-δεδομένων. Τα μέτα-δεδομένα είναι πληροφορίες για τα ίδια τα δεδομένα, που μπορούν να χρησιμοποιηθούν για να χαρακτηρίσουν τα δεδομένα από διάφορες οπτικές γωνίες. Πιο αναλυτικά πρόκειται για ιδιότητες των δεδομένων ανεξάρτητες από το πεδίο της εφαρμογής που τα χρησιμοποιεί όπως το ανεξαρτήτου αποθηκευτικού συστήματος όνομα αρχείου, η προέλευση των δεδομένων και ο τρόπος απόκτησής τους, η δομή των δεδομένων και ο τύπος τους, η πολιτική πρόσβασης στα δεδομένα, οι επιτρεπόμενες λειτουργίες (π.χ. ανάγνωση, εγγραφή, προσθήκη) κλπ. Επιπλέον, οι Εικονικοί Οργανισμοί μπορούν να προσθέσουν ειδικά μέτα-δεδομένα σχετικά με το πεδίο εφαρμογής τους για την καλύτερη περιγραφή των δεδομένων, βασισμένοι σε πρότυπα της ευρύτερης κοινότητας χρηστών. Οι επιπλέον αυτή πληροφορία βοηθάει στο χαρακτηρισμό των δεδομένων και τον σχηματισμό συνεργασιών

στο πλαίσιο ενός Εικονικού Οργανισμού. Ακόμα και ανεξάρτητοι χρήστες μπορούν να δημιουργήσουν την δικιά τους ομάδα μετά-δεδομένων για να διευκολυνθούν. Γενικά, τα μετά-δεδομένα μπορούν να χρησιμοποιηθούν για την εφαρμογή διαφορετικών οπτικών μιας υφιστάμενης πληροφορίας υποβοηθώντας την περαιτέρω ανάλυση και επεξεργασία μεγάλων συλλογών δεδομένων. Για παράδειγμα μια οπτική μπορεί να οριστεί ως η ομαδοποίηση των δεδομένων βάσει κοινών χαρακτηριστικών και σχέσεων.

2.1.4 Διαχείριση δεδομένων

Ένα από τα πιο κρίσιμα συστήματα του πλέγματος είναι η υπηρεσία ανάκτησης της τοποθεσίας των αντιγράφων (Replica Location Service) [Chazapis 06]. Το περιβάλλον του πλέγματος επιβάλλει τα δεδομένα να είναι διασκορπισμένα σε παγκόσμια κλίμακα λόγω των περιορισμών σε αποθηκευτικό χώρο των διαφόρων αποθετηρίων δεδομένων, αλλά και λόγω της υποχρέωσης να υπάρχει ίση απόσταση σε αυτά – στο επίπεδο του δικτύου - από όλους τους χρήστες. Σε τέτοιες περιπτώσεις είναι σύνηθες να χρησιμοποιούνται τοπικά αντίγραφα για να ελαττωθεί ο χρόνος απόκρισης του δικτύου που προστίθεται κάθε φορά που υπάρχει μια λειτουργία απομακρυσμένης πρόσβασης σε δεδομένα. Στην ορολογία του πλέγματος ονομάζουμε αντίγραφο (replica) ένα τοπικό αρχείο το οποίο έχει ιδιότητες ανάγνωσης μόνο (read-only local copies) και αποτελεί αντιγραφή ενός απομακρυσμένου αρχείου που βρίσκεται σε κάποιο αποθετήριο δεδομένων. Αντίστοιχα υπάρχει η υπηρεσία διαχείρισης αντιγράφων στο πλέγμα (Replica Location Service - RLS), η οποία είναι υπεύθυνη για την διαχείριση αυτών των αρχείων. Για την επεξεργασία ενός αρχείου στο πλέγμα, μια εργασία θα πρέπει πρώτα να επικοινωνήσει με το Replica Location Service, ώστε να βρει όλες τις τοποθεσίες που υπάρχει το ζητούμενο αρχείο. Σε περίπτωση που το αρχείο υπάρχει τοπικά, τότε η εργασία μπορεί να χρησιμοποιήσει τις συνηθισμένες λειτουργίες αρχείων για να το επεξεργαστεί, όπως ένα κανονικό αρχείο. Σε περίπτωση που δεν υπάρχει τοπικά, πρέπει να ενεργοποιηθεί ο μηχανισμός μεταφοράς δεδομένων του πλέγματος για να δημιουργηθεί ένα τοπικό αντίγραφο, να ενημερωθεί για την νέα τοποθεσία το Replica Location Service και να προχωρήσει η εργασία με την πρόσβαση στο αρχείο. Τα τοπικά αντίγραφα αρχείων βοηθούν στην βελτίωση της επίδοσης εφαρμογών που απαιτούν συχνές λειτουργίες πρόσβασης σε απομακρυσμένα δεδομένα. Με την τοποθέτηση αντιγράφων πιο κοντά στο κόμβο εκτέλεσης της εφαρμογής, μειώνεται η συνολική απόκριση της εφαρμογής, αφού μηδενίζεται η απόκριση του δικτύου, ενώ μειώνεται και η συνολική χρήση του. Επιπλέον, μέσω ειδικών αλγορίθμων οι μηχανισμοί μεταφοράς δεδομένων μπορούν να αξιοποιήσουν τα πολλαπλά αντίγραφα για την αύξηση του συνολικού ρυθμού μεταφοράς δεδομένων, αλλά και εργαλεία επανάκτησης δεδομένων μπορούν να χρησιμοποιήσουν τα αντίγραφα για την αποκατάσταση των χαμένων αρχείων.

Σύγχρονες διανομές λογισμικού πλέγματος όπως το Globus Toolkit περιλαμβάνουν υπηρεσίες διαχείρισης αντιγράφων (Replica Location and Management services), καθώς είναι ιδιαίτερα σημαντικές στην συνολική αρχιτεκτονική του πλέγματος. Οι μηχανισμοί που έχουν χρησιμοποιηθεί από την επιστημονική κοινότητα τα τελευταία χρόνια για τις υπηρεσίες αυτές έχουν εξελιχθεί σημαντικά. Το αρχικό σχέδιο μιας κεντρικής βάσης δεδομένων που είχε αποθηκευμένες πληροφορίες για τις τοποθεσίες όλων των αντιγράφων, παραμερίστηκε προς όφελος μιας κατανεμημένης προσέγγισης. Η πιο διαδεδομένη λύση που χρησιμοποιείται ευρέως στο πλέγμα, γνωστή και ως Giggle Framework, συντάσσει ένα ενιαίο χώρο ονοματολογίας αρχείων με μοναδικό αναγνωριστικό ανά Εικονικό Οργανισμό γνωστό και ως λογικό όνομα αρχείου (Logical Filename – LFN), το οποίο συνδέει με την φυσική τοποθεσία όλων των αντιγράφων του αρχείου (Physical Filenames – PFN).

Τα LFN χρησιμοποιούνται από τις εφαρμογές για την εύρεση δεδομένων, χωρίς να χρειάζεται ιδιαίτερη γνώση για την φυσική τοποθεσία των δεδομένων. Τα PFN χρησιμοποιούνται από τις υπηρεσίες του πλέγματος, όπως η υπηρεσία Replica Location και είναι δομημένα με τρόπο παρόμοιο με τις διευθύνσεις ιστοσελίδων (URL). Στο PFN αναφέρεται το πρωτόκολλο πρόσβασης στα δεδομένα που πρέπει να χρησιμοποιηθεί, η διεύθυνση του αποθετηρίου που είναι αποθηκευμένα καθώς και η ακριβής τοποθεσία (path) μέσα στο κόμβο. Για την κατανομή της πληροφορίας της υπηρεσίας Replica Location, το Giggle Framework ορίζει δύο επίπεδα:

- Το πρώτο επίπεδο υπάρχει ο τοπικός κατάλογος αντιγράφων (Local Replica Catalog – LRC). Το LRC διατηρεί πληροφορία σχετική με λογικά ονόματα αρχείων, όπως λίστες εξουσιοδοτημένων ατόμων ή ομάδων και το είδος πρόσβασης τους στα δεδομένα, χρόνος/ώρα δημιουργίας του αρχείου κλπ. Επίσης, διατηρεί μια λίστα για κάθε λογικό όνομα αρχείου που περιέχει τις τοποθεσίες όλων των αντιγράφων που υπάρχουν για αυτό και πιο συγκεκριμένα το URL τους. Επιγραμματικά, δοθέντος ενός LFN, το LRC επιστρέφει τη λίστα με τα PFN που αντιστοιχούν σε αυτό.

- Στο δεύτερο επίπεδο και πιο υψηλό στην ιεραρχία της υπηρεσίας Replica Location, υπάρχει το ευρετήριο τοποθεσίας των αντιγράφων (Replica Location Index – RLI). Στο ευρετήριο αυτό φυλάσσονται πληροφορίες για τους διαθέσιμους τοπικούς καταλόγους και τα αντίστοιχα λογικά ονόματα αρχείου. Επιγραμματικά, δοθέντος ενός LFN, το RLI γνωρίζει ποιος τοπικός κατάλογος διαθέτει πληροφορία σχετικά με τα αντίγραφα του συγκεκριμένου LFN.

Σε ένα συνηθισμένο σενάριο υλοποίησης, κάθε συμμετέχων σε ένα VO διαχειρίζεται ένα LRC, ενώ για την συνολική διαχείριση της υπηρεσίας υπάρχει ένα κεντρικό RLI για κάθε VO. Όταν οι προδιαγραφές το απαιτήσουν, μπορούν να συνυπάρξουν πολλαπλά RLI σε παράλληλη συνδεσμολογία, προσδίδοντας μηχανισμούς εξισορρόπησης φορτίου και ανάνηψης από βλάβη στην

συνολική υπηρεσία. Το Giggle Framework επιτρέπει, επίσης, ένα κατάλογο να συνδεθεί με πολλαπλά ευρετήρια και αντίστροφα. Τέλος, τα πολλαπλά ευρετήρια μπορούν να συνδυαστούν σε μια δενδρική ιεραρχία.

ΚΕΦΑΛΑΙΟ 3

Δίκτυα Ομότιμων Κόμβων

3.1 Εισαγωγή

Τα Δίκτυα Ομότιμων Κόμβων (ΔΟΚ) είναι συστήματα κατανεμημένα από τη φύση τους, χωρίς καμία ιεραρχική οργάνωση και κεντρικό έλεγχο. Οι κόμβοι αυτό-οργανώνονται και δημιουργούν ένα λογικό δίκτυο πάνω από το υφιστάμενο IP δίκτυο. Τέτοια δίκτυα προσφέρουν αρκετά πλεονεκτήματα όπως αξιόπιστη αρχιτεκτονική δρομολόγησης ευρείας κλίμακας, αποδοτική αναζήτηση δεδομένων, τοπικότητα και αξιοποίηση γειτονικών κόμβων, περίσσεια αποθηκευτικού χώρου, μονιμότητα της διάχυσης αντιγράφων, ιεραρχική ονοματοδοσία, εμπιστοσύνη, ταυτοποίηση ή ανωνυμία, κλιμακωσιμότητα και ανοχή σε σφάλματα. Τα ΔΟΚ διαφέρουν με τα κλασσικά συστήματα πελάτη-εξυπηρετητή, καθώς ο κάθε κόμβος μπορεί να έχει και τους δύο ρόλους σε ένα ΔΟΚ. Σε αντίθεση με τα σύστημα τύπου Grid, τα ΔΟΚ προέρχονται από την συνεργασία καθιερωμένων και δικτυωμένων ομάδων χρηστών χωρίς αξιόπιστους πόρους προς διάθεση.

Οι διάφοροι τύποι ΔΟΚ που υπάρχουν μπορούν να διακριθούν σε πολλές κατηγορίες ανάλογα με τα χαρακτηριστικά που έχουν, όμως η κυριότερη διαφοροποίηση βρίσκεται στη διαχείριση των κόμβων και ιδιαίτερα στους μηχανισμούς αναζήτησης κόμβων και στους αλγορίθμους δρομολόγησης των πακέτων. Με βάση αυτό μπορούμε να διακρίνουμε δύο μεγάλες κατηγορίες ΔΟΚ, τα δομημένα και τα αδόμητα ΔΟΚ (structured and unstructured peer to peer networks).

Ο όρος δομημένα Δίκτυα Ομότιμων Κόμβων προέρχεται από το γεγονός ότι η τοπολογία του δικτύου είναι ελεγχόμενη και τα δεδομένα τοποθετούνται σε συγκεκριμένες θέσεις και όχι σε τυχαίους κόμβους στο δίκτυο. Έτσι υπάρχει μεγάλο περιθώριο στην βελτιστοποίηση της απόδοσης

των ερωτημάτων (queries). Τέτοια δίκτυα χρησιμοποιούν ένα Κατανεμημένο Πίνακα Κατακερματισμού (ΚΠΚ) ως υπόστρωμα, με βάση το οποίο υπολογίζεται ντετερμινιστικά η τοποθεσία ενός αντικειμένου, η οποία προσδιορίζεται από τους κόμβους που αντιστοιχούν στο μοναδικό κλειδί του αντικειμένου. Τα συστήματα αυτά έχουν μια ιδιότητα η οποία αναθέτει ομοιόμορφα κατανεμημένους τυχαίους αριθμούς ως αναγνωριστικά σε όλους τους κόμβους του δικτύου με τρόπο συνεπή. Τα αντικείμενα που πρόκειται να αποθηκευθούν στο δίκτυο αποκτούν και αυτά ένα αντίστοιχο αναγνωριστικό το οποίο ονομάζεται κλειδί. Τα κλειδιά και τα αναγνωριστικά των κόμβων έχουν το ίδιο πεδίο τιμών, οπότε το δίκτυο μπορεί και αντιστοιχεί κάθε κλειδί σε ένα αναγνωριστικό, δηλαδή σε ένα ενεργό κόμβο του δικτύου, με βάση μια συνάρτηση εγγύτητας. Τα ΔΟΚ υποστηρίζουν την αποθήκευση και ανάκτηση ζευγαριών κλειδιού-τιμής (key-value pairs) σε μεγάλη κλίμακα. Έχοντας δεδομένο ένα κλειδί, μια λειτουργία αναζήτησης της τιμής ενός κλειδιού (`value=get(key)`) μπορεί να επιτευχθεί με μια σειρά αιτήσεων δρομολόγησης από το κόμβο που κάνει την αναζήτηση μέχρι να καταλήξει στον κόμβο που είναι υπεύθυνος για το συγκεκριμένο κλειδί.

Κάθε κόμβος διατηρεί ένα μικρό πίνακα δρομολόγησης το οποίο εμπεριέχει για κάθε γειτονικό κόμβο, το αναγνωριστικό του και την IP διεύθυνσή του. Τα ερωτήματα αναζήτησης ή τα μηνύματα δρομολόγησης προωθούνται σε άλλους κόμβους του δικτύου, με κριτήριο την εγγύτητα του αναγνωριστικού κάθε κόμβου στο κοινό πεδίο τιμών σε σχέση με το κλειδί του ερωτήματος/κόμβου προορισμού. Κάθε δίκτυο με ΚΠΚ έχει διαφορετική οργάνωση των πεδίων τιμών, καθώς και μηχανισμούς δρομολόγησης. Στην θεωρία, τα συστήματα βασισμένα σε ΚΠΚ μπορούν να εγγυηθούν πως κάθε τιμή μπορεί να βρεθεί με την μεσολάβηση ενός μικρού αριθμού κόμβων - O(logN) - όπου N είναι ο συνολικός αριθμός των κόμβων στο δίκτυο. Η διαδρομή μεταξύ δύο κόμβων στο φυσικό δίκτυο δεν είναι απαραίτητα όμοια με αυτή στο λογικό δίκτυο ενός ΔΟΚ, αλλά είναι πολύ πιθανό να διαφέρει κατά πολύ. Για αυτό, η καθυστέρηση κατά την αναζήτηση μιας τιμής στα ΔΟΚ ενδέχεται να είναι αρκετά σημαντική, σε βαθμό που να επηρεάζει την επίδοση των εφαρμογών που είναι βασισμένες σε αυτά. Το ΔΟΚ Plaxton [C. Plaxton] παρέχει ένα αρκετά ενδιαφέρον αλγόριθμο ο οποίος πετυχαίνει σχεδόν την ελάχιστη καθυστέρηση σε γράφους που διευρύνονται εκθετικά [L. Breslau], διατηρώντας ταυτόχρονα την κλιμακωσημότητα του αλγορίθμου δρομολόγησης ενός συστήματος ΚΠΚ. Όμως, ο αλγόριθμος αυτός απαιτεί συνεχή ερωτήματα μεταξύ των κόμβων για να διερευνήσουν τις μεταξύ τους καθυστερήσεις, μειώνοντας έτσι την πιθανότητα να είναι κλιμακώσιμος για όλους τους κόμβους του δικτύου. Τα συστήματα τύπου ΚΠΚ [D. R. Karger 97] [Rowstron] [Ratnasamy 01] [Stoica 03] [Zhao 04] είναι μια σημαντική κατηγορία στα ΔΟΚ. Υποστηρίζουν την ταχεία ανάπτυξη μιας ευρείας ποικιλίας εφαρμογών κλιμακώσιμων σε ολόκληρο το Internet, από κατανεμημένα συστήματα αρχείων μέχρι multicast στο επίπεδο της εφαρμογής. Επίσης, τα συστήματα αυτά προσέφεραν ένα εύκολο τρόπο για τον διαμοιρασμό δεδομένων ο οποίος είναι και κλιμακώσιμος σε παγκόσμια κλίμακα .

Από τους πρωτοπόρους στα ΔΟΚ ήταν το Napster [Nap], το οποίο έδωσε την δυνατότητα απευθείας ανταλλαγής αρχείων μεταξύ των απλών χρηστών. Οι τοποθεσίες των αρχείων ήταν σε ένα κεντρικό κατάλογο, ο οποίος προσέφερε δυνατότητες αναζήτησης. Η καινοτομία έγκειται στο γεγονός ότι ο δημιουργός του Napster συνειδητοποίησε πρώτος, ότι τα δημοφιλή δεδομένα δεν είναι ανάγκη να μεταφορτώνονται από ένα κεντρικό εξυπηρετητή, αλλά μπορούν να μεταφορτωθούν από άλλους κόμβους-πελάτες που έχουν ήδη αποκτήσει τα συγκεκριμένα δεδομένα. Τέτοια συστήματα έχουν την δυνατότητα να κλιμακώνονται αυτόματα, καθώς όσο περισσότεροι χρήστες/κόμβοι εισέλθουν στο δίκτυο, τόσο περισσότερο αυξάνεται το διαθέσιμο εύρος ζώνης για μεταφόρτωση. Το Napster χρησιμοποίησε μια κεντρική υποδομή αναζήτησης, βασισμένη σε λίστες αρχείων που προμήθευαν οι χρήστες και επομένως δεν χρησιμοποιεί δικτυακούς πόρους κατά την αναζήτηση αντικειμένων, αλλά έχει ένα κεντρικό σημείο βλάβης στο οποίο βασίζεται όλο το σύστημα. Το τέλος του Napster σηματοδότησε η αγωγή από την Recording Industry Association of America (RIAA), η οποία στρεφόταν κατά του μηχανισμού ανταλλαγής αρχείων μουσικής που ήταν και η καρδιά του συστήματος. Παρόλα αυτά, το παράδειγμα του Napster βοήθησε στην ανάπτυξη παρόμοιων υπηρεσιών, όπως το Gnutella [Gnu 01] [Gnu 02a] [Gnu 02b]. Το σύστημα Gnutella είναι ένα κατανεμημένο σύστημα που αποκεντρώνει και την αναζήτηση δεδομένων εκτός από την μεταφορά τους και είναι το πρώτο σύστημα το οποίο χρησιμοποιεί ένα αδόμητο ΔΟΚ. Στο σύστημα αυτό οι κόμβοι εντάσσονται στο δίκτυο με κάποιους χαλαρούς κανόνες και χωρίς γνώση της τοπολογίας εκ των προτέρων. Το δίκτυο χρησιμοποιεί την τεχνική της πλημμύρας με περιορισμένο εύρος για την αποστολή ερωτημάτων προς τους υπόλοιπους κόμβους. Με αυτήν την τεχνική όταν ένας κόμβος λαμβάνει ένα μήνυμα, στέλνει τα αποτελέσματα του ερωτήματος για τα τοπικά δεδομένα στον αρχικό αποστολέα. Η τεχνική αυτή είναι αρκετά αποδοτική όταν γίνεται αναζήτηση δημοφιλών αντικειμένων, αλλά δεν αποδίδει καλά όταν γίνεται αναζήτηση σπάνιων αντικειμένων. Επίσης, η τεχνική αυτή δεν είναι κλιμακώσιμη, όταν ο φόρτος σε κάθε κόμβο αυξάνει γραμμικά με το συνολικό πλήθος των ερωτημάτων ή το μέγεθος του δικτύου. Αυτό είναι ένα γενικότερο πρόβλημα των αδόμητων ΔΟΚ, ότι δηλαδή οι κόμβοι υπερφορτώνονται γρήγορα, οπότε το σύστημα δεν είναι κλιμακώσιμο όταν πρέπει να διαχειριστεί έναν υψηλό αριθμό ερωτημάτων ή μια απότομη αύξηση του μεγέθους του δικτύου. Τα δομημένα ΔΟΚ μπορούν να εντοπίσουν σπάνια αντικείμενα αρκετά αποδοτικά, λόγω της δρομολόγησης που είναι βασισμένη στο μοναδικό κλειδί του αντικειμένου, αλλά επιβαρύνονται με μεγαλύτερο φόρτο για την εξυπηρέτηση των δημοφιλών αντικειμένων σε σχέση με τα αδόμητα ΔΟΚ.

3.2 Αδόμητα ΔΟΚ

Σε αυτή τη κατηγορία οι κόμβοι του δικτύου οργανώνονται σε ένα τυχαίο γράφο σε ένα και μόνο επίπεδο ή εισάγοντας μια μικρή ιεραρχία. Οι μέθοδοι αναζήτησης που ακολουθούνται βασίζονται συνήθως σε τεχνικές πλημμύρας, διερεύνηση τυχαίων μονοπατιών (random walks) κλπ

Κάθε κόμβος που λαμβάνει ένα ερώτημα, το αποτιμά βάσει των δεδομένων που έχει αποθηκευμένα, ενώ υποστηρίζει και πολύπλοκα ερωτήματα. Βέβαια αυτό είναι μη αποδοτικό διότι τα ερωτήματα πρέπει να φτάσουν σε ένα μεγάλο ποσοστό των κόμβων του δικτύου για να προκύψουν αξιόπιστα αποτελέσματα, καθώς επίσης δεν υπάρχει καμία συσχέτιση των δεδομένων με την τοπολογία του δικτύου.

3.2.1 Gnutella

Το Gnutella [Ganesan 03] [Lv 02] [Chawathe 03] δημιουργήθηκε σαν ένα πρωτόκολλο κατανεμημένης αναζήτησης δεδομένων σε μια επίπεδη ιεραρχία κόμβων. Παρότι το πρωτόκολλο υποστηρίζει την παραδοσιακή επικοινωνία πελάτη-εξυπηρετητή, είναι μοναδικό διότι κάθε κόμβος είναι και πελάτης και εξυπηρετητής επιτρέποντας έτσι την κατανεμημένη αναζήτηση και ανάκτηση δεδομένων. Το δίκτυο δεν έχει κάποιο κεντρικό κατάλογο, δεν υπάρχει κάποια ιδιαίτερη δομή στην τοπολογία του και η εναπόθεση των αρχείων γίνεται χωρίς κάποια οργάνωση. Για την αναζήτηση ενός αντικειμένου ένας κόμβος θα επικοινωνήσει με τους γείτονές του και στην συνέχεια θα επεκτείνει το εύρος της αναζήτησης με τεχνικές πλημμύρας θέτοντας κάποια όρια στην ακτίνα δράσης της. Αυτός ο σχεδιασμός είναι εξαιρετικά ανθεκτικός σε αυξημένους ρυθμούς εισόδου/εξόδου κόμβων στο δίκτυο. Παρόλ’ αυτά οι μηχανισμοί αναζήτησης δεν είναι κλιμακώσιμοι και παράγουν σημαντικό φορτίο στους πόρους του δικτύου.

Οι κόμβοι του δικτύου είναι επιφορτισμένοι με εργασίες που αντιστοιχούν σε πελάτες αλλά και σε εξυπηρετητές σύμφωνα με το παραδοσιακό μοντέλο. Παρέχουν διεπαφές προς τους χρήστες για την αναζήτηση δεδομένων, ενώ ταυτόχρονα δέχονται ερωτήματα από άλλους κόμβους, ελέγχουν τα τοπικά δεδομένα τους και απαντούν στα σχετικά με αυτά ερωτήματα. Επίσης, είναι επιφορτισμένοι με την διαχείριση της κίνησης, δρομολογώντας τα διάφορα ερωτήματα προς άλλους κόμβους.

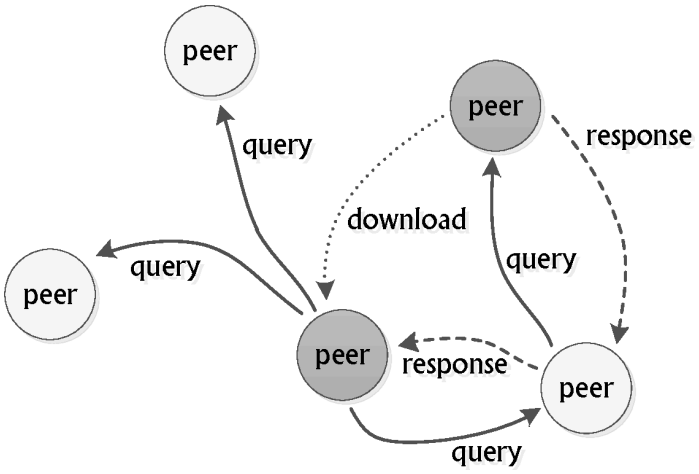
Για να εισέλθει ένας κόμβος στο δίκτυο, συνδέεται αρχικά με κάποιους γνωστούς κόμβους με αρκετή μεγάλη διαθεσιμότητα, τους οποίους μπορεί να μάθει από μια προκαθορισμένη τοποθεσία στο Internet. Αυτό δεν σημαίνει όμως ότι υπάρχει κεντρικό σημείο βλάβης (SPOF) καθώς η τοποθεσία συνήθως δίδεται μέσω ενός ονόματος DNS, το οποίο να αντιστοιχεί σε μια μεγάλη λίστα ανεξάρτητων εξυπηρετητών, οι οποίοι με τη σειρά τους κρατούν τη λίστα με τους κόμβους του Gnutella. Έτσι, στην ουσία στηριζόμαστε στην αξιοπιστία και διαθεσιμότητα του συστήματος ονοματοδοσίας του Internet, που έχει αποδειχθεί αρκετά καλή στην πράξη. Με το που θα συνδεθεί ο νέος κόμβος, λοιπόν, θα αρχίσει να στέλνει μηνύματα προς τους υπόλοιπους στη λίστα, ώστε να συνδεθεί με τους αρχικούς γείτονές του. Κάθε μήνυμα έχει ένα αναγνωριστικό που δημιουργείται με τυχαίο τρόπο, καθώς και ένα μετρητή που μειώνεται καθώς προωθείται το μήνυμα από κόμβο σε κόμβο. Επίσης, κάθε κόμβος διατηρεί μια μικρού μεγέθους μνήμη απεσταλμένων μηνυμάτων, ώστε να αποφεύγει την επαναδρομολόγηση ήδη απεσταλμένων μηνυμάτων και να μπορεί

να δρομολογήσει τυχόν απαντήσεις πίσω στον αρχικό αποστολέα. Τα μηνύματα που επιτρέπει το πρωτόκολλο είναι τα εξής:

- Group Membership (`PING, PONG`). Κάθε κόμβος κατά την είσοδό του στο δίκτυο στέλνει ένα μήνυμα τύπου PING για να ανακοινώσει την παρουσία του. Οι παραλήπτες απαντούν με ένα `PONG` που περιέχει πληροφορίες για το κόμβο, όπως την IP διευθυνσή του, το πλήθος και μέγεθος των αποθηκευμένων δεδομένων κλπ

- Search (`QUERY, QUERY RESPONSE`). Το μήνυμα αυτό περιέχει τις λέξεις αναζήτησης του χρήστη. Κάθε κόμβος αναζητά στην βάση με τα τοπικά ονόματα αρχείων, ενώ ταυτόχρονα προωθεί το μήνυμα και στους γείτονές του. Τα αποτελέσματα κάθε τοπικής αναζήτησης στέλνονται στον αποστολέα από την ίδια διαδρομή με ένα μήνυμα `QUERY RESPONSE`, καθώς και πληροφορίες για την πρόσβαση στα επιθυμητά αρχεία.

Επομένως, για να γίνει κάποιος μέλος του δικτύου αρκεί να συνδεθεί με ένα ή περισσότερους κόμβους που είναι ήδη μέλη του δικτύου. Σε αυτό το τόσο δυναμικό περιβάλλον, για να ξεπεραστούν τυχόν προβλήματα διαθεσιμότητας των κόμβων, κάθε κόμβος στέλνει περιοδικά μηνύματα `PING` προς τους γείτονές του για να μάθει νέους κόμβους, από τους οποίους θα διαλέξει ένα υποσύνολο για να συνδεθεί, βάση κάποιας δικής του μετρικής, η οποία μπορεί να αφορά το πλήθος και το είδος των δεδομένων των γειτόνων ή την απόστασή τους. Μια τυπική διαδικασία αναζήτησης ενός αντικειμένου στο Gnutella φαίνεται στο σχήμα 3.1.



Σχήμα 3.1: *Μια τυπική διαδικασία αναζήτησης στο Gnutella*

Πολλές μελέτες έχουν γίνει για να βελτιώσουν την κλιμακωσιμότητα του Gnutella και την απόδοση του μηχανισμού αναζήτησης. Μια από τις σημαντικότερες είναι η καθιέρωση μιας ιεραρχίας

στο δίκτυο με την εισαγωγή των υπέρ-κόμβων [Gnu 02b]. Τον ρόλο αυτό τον αναλαμβάνουν κόμβοι που έχουν διαθέσιμους πόρους - κυρίως δικτυακούς - αρκετά παραπάνω από το μέσο όρο του δικτύου. Οι υπέρ-κόμβοι βοηθούν στην βελτίωση της δρομολόγησης των μηνυμάτων στο δίκτυο, αλλά και πάλι περιορίζονται από την τεχνική της πλημμύρας για την μεταξύ τους επικοινωνία. Επιπλέον, ένας υπέρ-κόμβος θα πρέπει να πάρει μια απόφαση για τις δυνατότητες ενός γείτονα (ώστε να τον προάγει σε υπέρ-κόμβο ή όχι), ενώ δεν υπάρχει κάποιος μηχανισμός για την δυναμική προσαρμογή της απόφασης ανάλογα με τα χαρακτηριστικά του δικτύου. Οι υπέρ-κόμβοι διαχειρίζονται ερωτήματα εκ μέρους των απλών κόμβων. Κάθε κόμβος είναι συνδεδεμένος με μια ομάδα από υπέρ-κόμβους, οι οποίοι διαθέτουν την τοπική του λίστα με τα δεδομένα που φιλοξενεί. Με τον τρόπο αυτό, για να απαντηθεί ένα ερώτημα αρκεί να επικοινωνήσει με ένα υποσύνολο ή το σύνολο των υπέρ-κόμβων, μειώνοντας σημαντικά το πλήθος των εμπλεκόμενων κόμβων σε σχέση με την προηγούμενη κατάσταση. Μια επιπλέον βελτίωση του μηχανισμού αναζήτησης είναι η προοδευτική αύξηση της ακτίνας του κάθε μηνύματος (αύξηση της τιμής time-to-live του πακέτου) σε περίπτωση που το πλήθος των αποτελεσμάτων είναι κάτω από ένα όριο.

3.2.2 Freenet

Το Freenet [Clarke 01] [Clarke 02] είναι ένα ΔΟΚ με διαφορετικό προσανατολισμό από τα υπόλοιπα. Κύριοι σχεδιαστικοί στόχοι του είναι η ανωνυμία για τους παραγωγούς και τους καταναλωτές της πληροφορίας, η δυνατότητα άρνησης ενός κόμβου ότι είναι ενήμερος για την κατοχή κάποιας πληροφορίας, η προστασία από επιθέσεις για διακοπή της πρόσβασης στα δεδομένα, η αποδοτική δρομολόγηση των ερωτημάτων, ο δυναμικά αναπτυσσόμενος αποθηκευτικός χώρος και οι δικτυακά κατανεμημένες λειτουργίες του.

Στο Freenet κάθε αποθηκευμένο αντικείμενο αναγνωρίζεται από ένα κλειδί το οποίο είναι ανεξάρτητο με τη τοπολογία του δικτύου. Κάθε κόμβος διατηρεί ένα δυναμικό πίνακα δρομολόγησης που περιέχει τις διευθύνσεις των άλλων κόμβων καθώς και τα κλειδιά των αντικειμένων που αυτοί περιέχουν. Τα σημαντικότερα πλεονεκτήματα του Freenet είναι ότι έχει την δυνατότητα να διατηρεί τοπικά κάποια δεδομένα ανάλογα με το μέγιστο όριο αποθηκευτικού χώρου που είναι διαθέσιμο για το σκοπό αυτό, καθώς και το γεγονός ότι διαθέτει μηχανισμούς ασφαλείας για να αντιμετωπίσει κακόβουλους κόμβους. Το βασικό μοντέλο που ακολουθεί είναι η προώθηση των διάφορων ερωτημάτων από κόμβο σε κόμβο δημιουργώντας ενδιάμεσα ερωτήματα, τα οποία προωθούνται με βάση μια τοπική απόφαση του κάθε κόμβου. Σε αυτό το μονοπάτι δρομολόγησης που δημιουργείται, κάθε κόμβος ξέρει το προηγούμενο και τον επόμενο κόμβο μόνο, ώστε να διατηρείται η ανωνυμία αποστολέα και παραλήπτη. Για την αποφυγή ατέρμονων βρόχων κάθε μήνυμα φέρει και ένα μετρητή που μειώνεται κάθε φορά που αλλάζει κόμβο το μήνυμα, ενώ σε περίπτωση που μηδενίσει σταματά η περαιτέρω προώθηση του μηνύματος.

Τα κλειδιά στο Freenet προκύπτουν από την κρυπτογραφική συνάρτηση SHA-1 [SHA 95] και είναι μεγέθους 160 bit. Υπάρχουν δυο τύποι κλειδιών, τα content-hash που χρησιμεύουν στην αποθήκευση των αρχείων και τα signed-subspace που προορίζονται για ανθρώπινη χρήση. Αποτελούν το αντίστοιχο των inode και ονομάτων αρχείων σε ένα σύστημα αρχείων όπου το πρώτο χρησιμοποιείται από το λειτουργικό σύστημα και το δεύτερο από τον χρήστη.

Το κλειδί κατακερματισμένου περιεχομένου (content-hash key - CHK) παράγεται εφαρμόζοντας μια συνάρτηση κατακερματισμού στα περιεχόμενα του αντίστοιχου αρχείου. Χρησιμοποιώντας την συνάρτηση SHA-1 πετυχαίνουμε το κλειδί αυτό να είναι μοναδικό (καθώς οι συγκρούσεις στην συνάρτηση SHA-1 είναι σχεδόν απίθανες). Με το τρόπο αυτό έχοντας ένα κλειδί και ένα αρχείο μπορούμε να επιβεβαιώσουμε την σχέση τους, ενώ σε περίπτωση που κάποιος τρίτος εισάγει το ίδιο αρχείο, το σύστημα το αναγνωρίζει και εξοικονομεί πόρους.

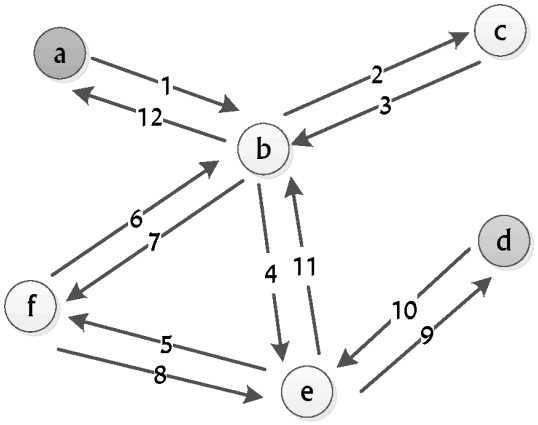
Το κλειδί υπογεγραμμένης υποπεριοχής (singed-subspace key SSK) δημιουργεί ένα προσωπικό χώρο ονομάτων που ο καθένας μπορεί να διαβάσει, αλλά μόνο ο ιδιοκτήτης μπορεί να γράψει. Ο χρήστης που έχει μια συλλογή αρχείων, δημιουργεί μια υποπεριοχή για αυτά, παράγοντας ένα ζεύγος δημόσιου-ιδιωτικού κλειδιού για να την αναγνωρίζει. Για την προσθήκη ενός αρχείου, χρησιμοποιείται ένα περιγραφικό όνομα (πχ Παγκόσμια-Οικονομία/Ευρωζώνη/Κρίση-Χρέους). Στην συνέχεια το SSK του αρχείου υπολογίζεται εφαρμόζοντας τη συνάρτηση κατακερματισμού ξεχωριστά στο δημόσιο κλειδί και στο περιγραφικό όνομα και στη συνέχεια και στα δύο μαζί. Τέλος, το αποτέλεσμα υπογράφεται ψηφιακά με το ιδιωτικό κλειδί. Έτσι κάθε κόμβος θα μπορεί να επιβεβαιώσει την αυθεντικότητα της συλλογής αρχείων, ενώ μόνο ο ιδιοκτήτης που έχει το ιδιωτικό κλειδί μπορεί να αλλάξει τα περιεχόμενα της συλλογής. Για να ανακτηθεί η συλλογή από ένα τρίτο χρήστη, θα πρέπει να είναι γνωστά σε αυτόν, το περιγραφικό όνομα της συλλογής καθώς και το δημόσιο κλειδί της, με τα οποία μπορεί να δημιουργηθεί ξανά το SSK. Προαιρετικά δύναται να χρησιμοποιηθεί και τρίτο κλειδί, το οποίο κρυπτογραφεί τα δεδομένα ενός αρχείου, ώστε να προστατευτούν από μη εξουσιοδοτημένους χρήστες, αλλά και να δοθεί η δυνατότητα σε κάποιο κόμβο του δικτύου της άρνησης ότι είναι ενήμερος για την κατοχή κάποιας ευαίσθητης πληροφορίας.

Ο αλγόριθμος δρομολόγησης για την αποθήκευση και την ανάκτηση δεδομένων έχει σχεδιαστεί για να προσαρμόζει δυναμικά τα μονοπάτια μεταξύ των κόμβων ανάλογα με την κατάσταση του δικτύου. Για την δρομολόγηση βγαίνουν αποφάσεις με βάση τα τοπικά στοιχεία του κάθε κόμβου, αφού δεν υπάρχει γενική εποπτεία του δικτύου, δεδομένου ότι κάθε κόμβος γνωρίζει μόνο τους γειτονικούς του. Επομένως, ο αλγόριθμος δρομολόγησης αποδίδει καλά για δημοφιλή δεδομένα, που έχουν ήδη γίνει γνωστά στην πλειοψηφία των κόμβων. Σε κάθε μήνυμα δίνεται μια τιμή Time-To-Live (TTL) η οποία ουσιαστικά αποτελεί το χρόνο ζωής του μηνύματος, καθώς θα μειώνεται σε κάθε κόμβο που θα προωθείται, ενώ θα σταματήσει η προώθησή του όταν η τιμή του μηδενιστεί. Επιπλέον, για να αποφευχθεί ατέρμων βρόχος σε κάθε μήνυμα δίνεται και ένα ψευδοτυχαίο, μοναδικό αναγνωριστικό. Σε περίπτωση που σε ένα κόμβο επιστρέψει μήνυμα από διαφορετικό γείτονα, τότε αυτό προωθείται από διαφορετικό γείτονα σε σχέση με τη προηγούμενη φορά.

Η προώθηση συνεχίζεται μέχρι να ικανοποιηθεί το ερώτημα ή να μηδενιστεί ο μετρητής TTL. Σε κάθε περίπτωση μια απάντηση επιτυχίας ή αποτυχίας θα σταλεί ακολουθώντας το ίδιο μονοπάτι αντίστροφα προς τον αποστολέα του αρχικού μηνύματος. Για την αύξηση της απόδοσης της αναζήτησης/ανάκτησης αρχείων γίνεται εκτενής χρησιμοποίηση τεχνικών προσωρινής αποθήκευσης (caching) σε κόμβους που βρίσκονται στο μονοπάτι μιας επιτυχούς απάντησης. Για την εισαγωγή ενός κόμβου στο δίκτυο, θα πρέπει είναι γνωστή η διεύθυνση τουλάχιστον ενός κόμβου του δικτύου. Όλοι οι κόμβοι είναι ισότιμοι, καθώς δεν υφίσταται κάποια ιεραρχία ή/και κεντρικό σημείο αναφοράς. Αυτό έχει σαν αποτέλεσμα την ανεκτικότητα στις βλάβες, ενώ ο ρυθμός εισόδου/εξόδου κόμβων σε αυτό, δεν επηρεάζει σημαντικά την δρομολόγηση μηνυμάτων. Για την αποθήκευση νέων αρχείων, ακολουθείται η ίδια τακτική με την αναζήτηση αρχείων. Δηλαδή γίνεται αναζήτηση για το κλειδί του αρχείου, οπότε αν υπάρχει τότε η αποθήκευση αποτυγχάνει, αλλιώς βρίσκουμε τον κόμβο με το κοντινότερο κλειδί σε σχέση με το κλειδί του αρχείου για να αποθηκευτεί το νέο αρχείο. Ένα ενδιαφέρον χαρακτηριστικό του Freenet, είναι πως με την πάροδο του χρόνου οι κόμβοι εκπαιδεύονται, καθώς κρατούν πληροφορία στους πίνακες δρομολόγησης σχετικά με τις επιτυχείς απαντήσεις, ώστε σε περίπτωση που λάβουν μήνυμα για ένα παρόμοιο κλειδί να γνωρίζουν σε πιο γείτονα να το προωθήσουν. Τέλος, ο αποθηκευτικός χώρος είναι περιορισμένος και για αυτό είναι πολύ πιθανόν να εξαντληθεί. Σε αυτή την περίπτωση χρησιμοποιούνται τεχνικές όπου τα λιγότερο χρησιμοποιημένα αρχεία (Least recently used – LRU) σβήνονται για να πάρουν τη θέση τους τα νέα αρχεία. Αντίστοιχες πολιτικές λαμβάνονται και στον πίνακα δρομολόγησης όταν αυτός γεμίσει.

Το σχήμα 3.2 απεικονίζει μια τυπική διαδικασία αναζήτησης. Ο χρήστης δημιουργεί ένα ερώτημα στο κόμβο Α και το προωθεί στο κόμβο Β, ο οποίος το προωθεί στον κόμβο Γ. Ο κόμβος Γ δεν μπορεί να επικοινωνήσει με άλλους και επιστρέφει μια αποτυχημένη απάντηση στο Β. Ο κόμβος Β τότε προσπαθεί να επικοινωνήσει με το κόμβο Ε, ο οποίος στην συνέχεια προωθεί το μήνυμα στο κόμβο F. Ο κόμβος F το προωθεί στο κόμβο Β, ο οποίος αντιλαμβάνεται ότι πρόκειται για βρόχο και απορρίπτει το μήνυμα. Ο κόμβος F επιστρέφει αποτυχημένη απάντηση στο κόμβο Ε, ο οποίος προωθεί το μήνυμα στο κόμβο D, ο οποίος και έχει το αρχείο. Το αρχείο θα ακολουθήσει το μονοπάτι επιστροφής E -> B -> A -> χρήστης. Ανάλογα με το διαθέσιμο αποθηκευτικό χώρο οι κόμβοι Ε,Β,Α θα αποθηκεύσουν το αρχείο για μελλοντική επανάκτηση.

Με αυτή τη προσέγγιση σε κάθε το ερώτημα ξεκινά μια εξαντλητική αναζήτηση από κόμβο σε κόμβο. Σε περίπτωση όμως που επαναληφθεί παρόμοιο ερώτημα οι πίνακες δρομολόγησης των ενδιάμεσων κόμβων θα έχουν αρκετή πληροφορία για παρακάμψουν τις λάθος διασταυρώσεις και να οδηγήσουν το ερώτημα στο σωστό κόμβο, ενώ αν πρόκειται για ακριβώς το ίδιο ερώτημα, τότε μπορεί να απαντηθεί και από τους ενδιάμεσους κόμβους που έχουν αποθηκεύσει την προηγούμενη απάντηση. Σταδιακά το δίκτυο θα εκπαιδεύεται, καθώς οι κόμβοι που απαντούν με επιτυχία στα ερωτήματα θα αποθηκεύονται στους πίνακες δρομολόγησης και επομένως θα αποκτούν καλύτερη



Σχήμα 3.2: *Μια τυπική διαδικασία αναζήτησης στο Freenet*

συνδεσιμότητα σε σύγκριση με κόμβους που έχουν λίγα δεδομένα και επιστρέφουν απαντήσεις με αποτυχία.

3.2.3 BitTorrent

Το πρωτόκολλο BitTorrent είναι ένα πρωτόκολλο ομότιμων κόμβων που επιτρέπει πολλούς κόμβους να λαμβάνουν ένα αρχείο δεδομένων μέσω δικτύου από πολλές πηγές ταυτόχρονα, ενώ συγχρόνως αποστέλλουν το μέρος του αρχείου που έχουν σε άλλους, μειώνοντας την συμφόρηση στους κόμβους-πηγές του αρχείου. Το πρωτόκολλο στοχεύει στην μείωση του χρόνου μεταφοράς μεγάλων και δημοφιλών αρχείων καθώς και της συμφόρησης των κεντρικών αποθετηρίων που περιέχουν αυτά τα αρχεία. Κάθε αρχείο, σύμφωνα με το πρωτόκολλο, διαιρείται σε κομμάτια (chunks), συνήθως των 256kB. Οι κόμβοι-πελάτες, leechers στην ορολογία του πρωτοκόλλου, μπορούν να λαμβάνουν πολλά κομμάτια ταυτόχρονα από διαφορετικές πηγές. Για λόγους ακεραιότητας των δεδομένων που επιβάλλεται από αυτή την μεγάλη κατάτμηση του αρχείου, για κάθε ομάδα κομματιών φυλάσσεται ένα άθροισμα ελέγχου, που προκύπτει από μια συνάρτηση κατακερματισμού. Με βάση αυτό το άθροισμα ελέγχου μπορεί ανά πάσα στιγμή να επιβεβαιωθεί ή όχι η ακεραιότητα των δεδομένων. Αυτή η πληροφορία, μαζί με το μέγεθος του αρχείου, αποθηκεύεται σε ένα αρχείο πληροφοριών (metainfo file), το οποίο χαρακτηρίζεται από τη κατάληξη .torrent. Κατά την αρχικοποίηση ενός κόμβου-πελάτη, γίνεται ανάγνωση του αρχείου .torrent ώστε να γίνει γνωστή η πολιτική κατάτμηση των δεδομένων, αλλά και η διεύθυνση του κόμβου-συντονιστή (tracker). Ο κόμβος-συντονιστής έχει την ευθύνη για την διατήρηση μιας λίστας με όλους τους ενεργούς κόμβους που συμμετέχουν στην μεταφορά του αρχείου. Μετά την επικοινωνία με το κόμβο-συντονιστή, ο κόμβος-πελάτη λαμβάνει μια λίστα με κόμβους που μπορεί να επικοινωνήσει

για την μεταφορά του αρχείου. Οι κόμβοι κατηγοριοποιούνται εκτός από τους κόμβους-πελάτες και σε κόμβους-πηγές (seeds) που έχουν ολόκληρο το αρχείο στη κατοχή τους. Από το σημείο εκείνο και έπειτα, ο κόμβος-πελάτης είναι υπεύθυνος για την επικοινωνία με τους υπόλοιπους και την επιλογή των κομματιών προς λήψη.

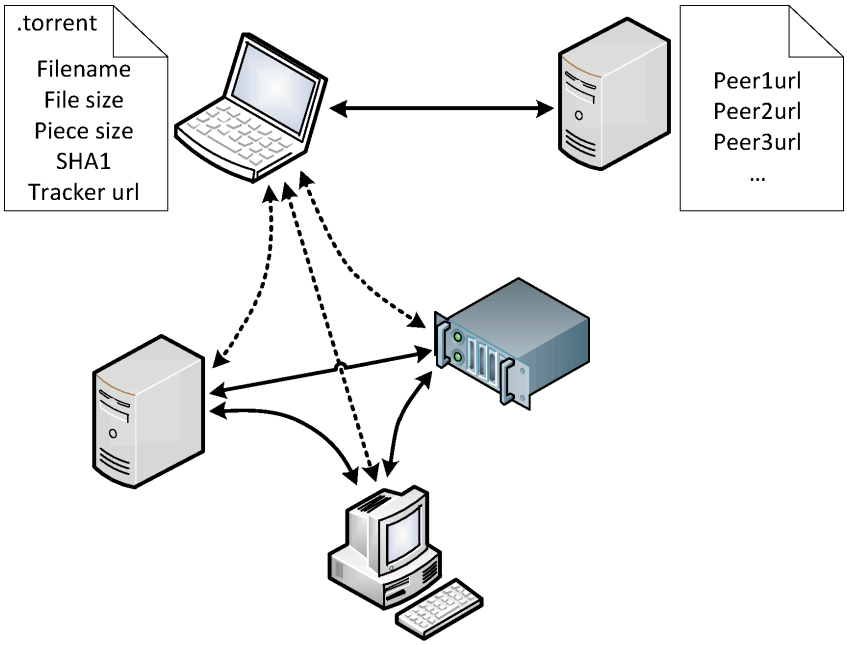
Για την βελτίωση της διαθεσιμότητας και διασφάλιση της ομοιόμορφης κατανομής των κομματιών στο δίκτυο, ένας κόμβος επιλέγει τα κομμάτια που θα πάρει βάσει μιας πολιτικής που ευνοεί τα σπανιότερα. Η αποτελεσματικότητα του BitTorrent στηρίζεται στον ενσωματωμένο μηχανισμό του για την δημιουργία κινήτρων, γνωστό και ως αλγόριθμο παρεμπόδισης (chocking algorithm). Ουσιαστικά, πρόκειται για ένα αλγόριθμο επιλογής κόμβων που θέτει όρια στον αριθμό των ταυτόχρονων συνδέσεων αποστολής δεδομένων, το οποίο είναι συνήθως τέσσερις συνδέσεις. Ο αλγόριθμος δίνει προτεραιότητα στις συνδέσεις με τις καλύτερες ταχύτητες μετάδοσης. Επίσης, υπάρχει και μια περίοδος επαναδιαπραγμάτευσης (rechoke period), όπου κάθε κόμβος υπολογίζει ξανά τις ταχύτητες μετάδοσης με τους γείτονές του και αποφασίζει ποιες πρέπει να κρατήσει (unchoke) και ποιες να παρεμποδίσει (choke). Επομένως, το πρωτόκολλο αποφεύγει φαινόμενα τύπου free-rider, όπου κάποιος κόμβος εισέρχεται στο δίκτυο χωρίς να προσφέρει πόρους στο δίκτυο. Ακόμα, ένας επιπρόσθετος κόμβος προστίθεται στο σύνολο αυτών στους οποίους αποστέλλονται δεδομένα, ώστε να εξεταστεί η δυνατότητα ανεύρεσης κάποιας καλύτερης σύνδεσης από άποψη ρυθμού μετάδοσης (optimistic unchoke). Ένα υποθετικό σενάριο λειτουργίας ενός δικτύου BitTorrent με βάση την παραπάνω περιγραφή φαίνεται στο σχήμα 3.3. Τέλος, η τελευταία έκδοση του πρωτοκόλλου αντικαθιστά το κόμβο-συντονιστή με ένα κατανεμημένο πίνακα κατακερματισμού για την δυναμική ανεύρεση των κόμβων που συμμετέχουν στην μεταφορά του αρχείου.

3.3 Δομημένα ΔΟΚ

Σε αυτή τη κατηγορία, το δίκτυο αναθέτει κλειδιά σε κάθε αντικείμενο και οργανώνει τους κόμβους σε ένα γράφο, έτσι ώστε κάθε αντικείμενο να έχει μια αντιστοίχηση με ένα κλειδί σε ένα κόμβο. Σε αυτό το δομημένο γράφο βασίζεται η αποδοτική ανάκτηση αντικειμένων χρησιμοποιώντας τα κλειδιά τους. Εν τούτοις, στην πιο απλή τους μορφή τα συστήματα αυτά δεν υποστηρίζουν πολύπλοκα ερωτήματα και πολλές φορές είναι αναγκαίο κάθε κόμβος να αποθηκεύει αντίγραφα των δεδομένων ή δείκτες στα δεδομένα τα οποία αντιστοιχούν στα κλειδιά για τα οποία είναι υπεύθυνος. Στην συνέχεια θα μελετήσουμε μερικά δομημένα ΔΟΚ.

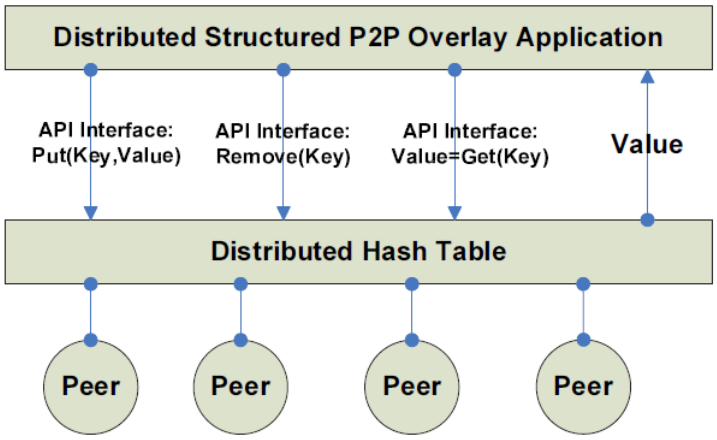
3.3.1 Content Addressable Network (CAN)

Το ΔΟΚ που ονομάζεται Content Addressable Network (CAN) [Ratnasamy 01] είναι ένα κατανεμημένο δίκτυο που προσφέρει την λειτουργικότητα του πίνακα κατακερματισμού σε παγκόσμια



Σχήμα 3.3: *Παράδειγμα ανάπτυξης BitTorrent*

κλίμακα. Το CAN έχει σχεδιαστεί να είναι κλιμακώσιμο, να έχει ανοχή σε βλάβες και να αυτό-οργανώνεται. Η αρχιτεκτονική του βασίζεται σε ένα εικονικό πολυδιάστατο Καρτεσιανό χώρο συντεταγμένων με διάταξη υπερκύβου (Multi-Torus). Αυτός ο ν-διάστατος χώρος συντεταγμένων είναι τελείως εικονικός. Ολόκληρος ο χώρος κατανέμεται δυναμικά σε όλους τους κόμβους του δικτύου, έτσι ώστε κάθε κόμβος να κατέχει την δικιά του, ανεξάρτητη και μοναδική περιοχή στο χώρο. Κάθε κόμβος διατηρεί ένα πίνακα δρομολόγησης με τις διευθύνσεις IP και τις περιοχές των γειτόνων του στο χώρο συντεταγμένων. Ένα μήνυμα στο δίκτυο έχει παραλήπτη μια διεύθυνση στο χώρο συντεταγμένων, με βάση την οποία κάθε κόμβος προωθεί το μήνυμα του χρησιμοποιώντας ένα άπληστο αλγόριθμό προώθησης προς τον πιο κοντινό γειτονικό κόμβο προς την διεύθυνση του παραλήπτη. Η απόδοση του αλγορίθμου δρομολόγησης είναι της τάξης $(dN^{1/d})$ και η κατάσταση δρομολόγησης του είναι οριοθετημένη στο $2d$. Όπως φαίνεται στο σχήματα 3.5 3.6 η αποθήκευση των ζευγαριών κλειδιού-τιμής ξεκινά πρώτα με το κλειδί να αντιστοιχείται με ντετερμινιστικό τρόπο σε ένα σημείο Π στο χώρο συντεταγμένων σύμφωνα με μια ομοιόμορφη συνάρτηση κατακερματισμού. Για την αναζήτηση ενός αντικειμένου που αντιστοιχεί σε ένα κλειδί Κ, ο οποιοσδήποτε κόμβος εφαρμόζει την ίδια ομοιόμορφη συνάρτηση κατακερματισμού στο κλειδί Κ, ώστε να προκύψει το σημείο Π, στο οποίο απευθύνει την ερώτηση για το περιεχόμενου του αντικειμένου. Αν ο ερωτών κόμβος ή οι γείτονές του δεν είναι υπεύθυνοι για το σημείο Π, το ερώτημα δρομολογείται μέσω του δικτύου CAN μέχρι να φτάσει στον κόμβο υπεύθυνο για το σημείο Π. Οι κόμβοι

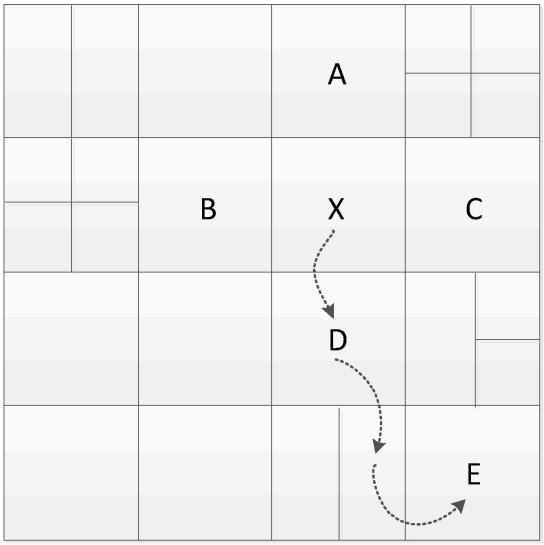


Σχήμα 3.4: *Διεπαφές μεταξύ διαφορετικών επιπέδων σε ένα δομημένο Δίκτυο Ομότιμων Κόμβων*

διατηρούν τις IP διευθύνσεις των γειτόνων τους στο χώρο συντεταγμένων στο πίνακα δρομολόγησης τους, ώστε να είναι δυνατή η υλοποίηση του παραπάνω αλγορίθμου δρομολόγησης.

Όταν εισάγεται ένας νέος κόμβος στο σύστημα τότε πρέπει να του αποδοθεί μια περιοχή ευθύνης από το χώρο συντεταγμένων. Αυτό μπορεί να επιτευχθεί με τον διαμοιρασμό των περιοχών ενός υφιστάμενου κόμβου, διατηρώντας κατά τον ήμισυ την περιοχή του και αποδίδοντας την υπόλοιπη στον νεοεισαχθέντα κόμβο. Το CAN διατηρεί ένα όνομα DNS, το οποίο αντιστοιχεί σε μια ή περισσότερες IP διευθύνσεις των αρχικών κόμβων (bootstrap peers). Οπότε, για την εισαγωγή του στο δίκτυο, ένας νέος κόμβος κάνει μια κλήση DNS και βρίσκει μια τουλάχιστον IP διεύθυνση ενός αρχικού κόμβου. Στην συνέχεια αφού επικοινωνήσει με τον αρχικό κόμβο λαμβάνει μια λίστα με διευθύνσεις τυχαίων κόμβων στο δίκτυο, διαλέγει ένα τυχαίο σημείο Π και στέλνει ένα αίτημα για ένταξη (JOIN request) προς το σημείο Π. Το αίτημα φθάνει στο τελικό παραλήπτη μέσω της δρομολόγησης του CAN και ο κόμβος χωρίζει την περιοχή για την οποία είναι υπεύθυνος σε δύο κομμάτια δίνοντας το ένα στον νέο κόμβο με την απάντηση στο αίτημα για ένταξη. Έπειτα μεταφέρονται και όλα τα ζευγάρια κλειδιού-τιμής τα οποία ανήκουν στην μεταφερόμενη περιοχή. Για να ολοκληρωθεί η ένταξη του νέου κόμβου συμπληρώνεται και ο πίνακας δρομολόγησης του νέου κόμβου με τους γειτονικούς του.

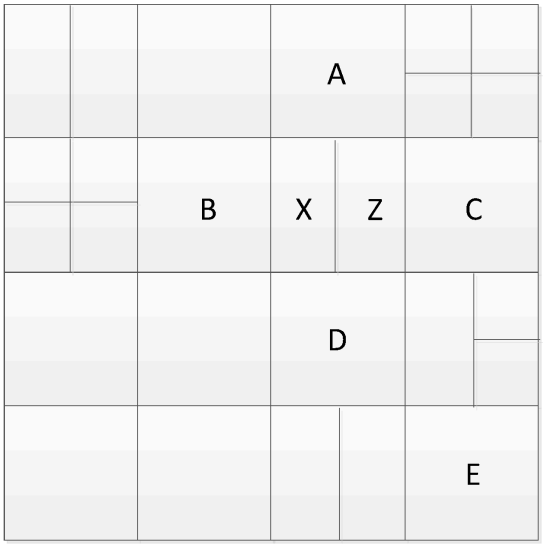
Όταν ένας κόμβος φύγει από το δίκτυο ενεργοποιείται άμεσα ένας αλγόριθμος επανάκτησης της περιοχής του κόμβου, την οποία αναλαμβάνει ένας γειτονικός κόμβος. Από τις πρώτες ενέργειες του κόμβου είναι η ενημέρωση του πίνακα δρομολόγησης του για την αποχώρηση του κόμβου, και στη συνέχεια η αποστολή ενημερώσεων στους υπόλοιπους γείτονες, ώστε να κάνουν την αντίστοιχη αλλαγή στους δικούς τους πίνακες. Το πλήθος των γειτόνων κάθε κόμβου είναι ευθέως ανάλογο με τις πλήθος των διαστάσεων του χώρου συντεταγμένων και δεν εξαρτάται από το συνολικό μέγεθος του δικτύου.



Σχήμα 3.5: *Ένα παράδειγμα ενός χώρου συντεταγμένων στο CAN, με δύο διαστάσεις: η δρομολόγηση του κόμβου X προς το σημείο E.*

Στα σχήματα 3.5 3.6 απεικονίζονται δύο παραδείγματα, η δρομολόγηση του κόμβου X προς το σημείο E και η εισαγωγή ενός νέου κόμβου Z στο δίκτυο. Για ένα d-διάστατο χώρο κατανεμημένο σε N διαφορετικές περιοχές (ή κόμβους), το μέσο μήκος μονοπατιού που θα ακολουθήσει η δρομολόγηση είναι $(d/4) * (N^{1/d})$ και ο κάθε κόμβος θα πρέπει να κρατά μια λίστα με τους $2.d$ γείτονες του. Από τα παραπάνω προκύπτει ότι η πληροφορία που χρειάζεται να έχει ένας κόμβος είναι ανεξάρτητη του συνολικού πλήθους των κόμβων, ενώ το ίδιο συμβαίνει και για το μέσο μήκος μονοπατιού μεταξύ δύο κόμβων, καθώς και τα δυο εξαρτώνται μόνο από το αριθμό των διαστάσεων του χώρου συντεταγμένων. Επιπρόσθετα, δεδομένου ότι υπάρχουν πολλά διαφορετικά μονοπάτια προς κάθε κόμβο στο χώρο, όταν κάποιος κόμβος αποτύχει, μπορεί να βρεθεί ένα εναλλακτικό μονοπάτι για να ολοκληρωθεί το ερώτημα.

Μια βελτίωση του αλγορίθμου που ακολουθεί το CAN μπορεί να επιτευχθεί διατηρώντας πολλαπλούς και ανεξάρτητους χώρους συντεταγμένων και κάθε κόμβους να εισάγεται σε όλους τους διαφορετικούς χώρους ή αλλιώς τις διαφορετικές πραγματικότητες. Για ένα δίκτυο CAN με r πραγματικότητες, ένα κόμβος θα έχει υπ' ευθύνη του r διαφορετικές περιοχές - μία σε κάθε πραγματικότητα - και κάθε κόμβος θα έχει r διαφορετικούς πίνακες δρομολόγησης με διευθύνσεις γειτόνων. Τα αντικείμενα του πίνακα κατακερματισμού διοχετεύονται σε κάθε πραγματικότητα, ώστε να



Σχήμα 3.6: *Ένα παράδειγμα ενός χώρου συντεταγμένων στο CAN, με δύο διαστάσεις: η εισαγωγή ενός νέου κόμβου Z στο δίκτυο.*

αυξηθεί η διαθεσιμότητα τους. Για επιπλέον αύξηση της διαθεσιμότητας το CAN μπορεί να χρησιμοποιήσει k διαφορετικές συναρτήσεις κατακερματισμού για να αντιστοιχήσει ένα κλειδί σε k σημεία στο χώρο συντεταγμένων. Αυτό έχει σαν αποτέλεσμα την αναπαραγωγή κάθε ζευγαριού κλειδιού-τιμής σε k κόμβους στο δίκτυο. Για να μην είναι διαθέσιμο ένα ζευγάρι κλειδιού-τιμής θα πρέπει να μην είναι διαθέσιμα κανένα από τα k αντίγραφα. Επομένως, τα ερωτήματα αναζήτησης για ένα κλειδί θα μπορούσαν να στέλνονται ταυτόχρονα προς όλους τους k κόμβους, ώστε να βελτιωθεί ο μέσος χρόνος απάντησης των ερωτημάτων, η αξιοπιστία και η ανοχή σε σφάλματα του συστήματος.

3.3.2 Chord

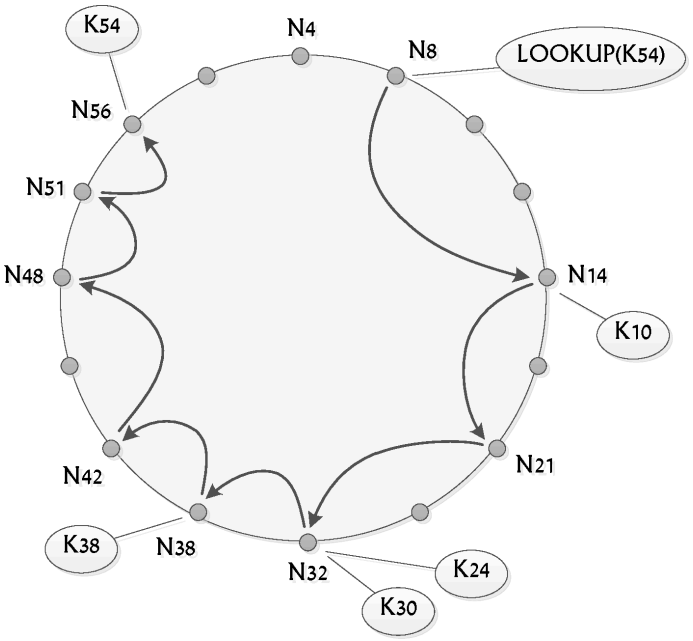
Το Chord [Stoica 03] χρησιμοποιεί συνεπή κατακερματισμό [D. R. Karger 97] για να αναθέσει κλειδιά σε κάθε κόμβο. Με το τρόπο αυτό οι κόμβοι εισέρχονται και αναχωρούν από το δίκτυο με την μικρότερη δυνατή παρεμβολή. Αυτό το κατανεμημένο σχήμα τείνει να ισορροπήσει το φόρτο του συστήματος, αφού κάθε κόμβος αναλαμβάνει περίπου το ίδιο πλήθος κλειδιών και υπάρχουν ελάχιστες μετακινήσεις κλειδιών από κόμβο σε κόμβο κατά τις εισόδους/εξόδους κόμβων στο/από

το δίκτυο. Σε μια σταθερή κατάσταση για N κόμβους στο δίκτυο, κάθε κόμβος διατηρεί πληροφορία για O(logN) άλλους κόμβους, όπου N το σύνολο των κόμβων στο δίκτυο. Αυτό μπορεί να είναι αποδοτικό, αλλά η επίδοση υποβαθμίζεται σταδιακά, όσο αυτή η πληροφορία δεν ενημερώνεται.

Η συνάρτηση κατακερματισμού αναθέτει σε κόμβους και αντικείμενα ένα κλειδί μεγέθους m bits χρησιμοποιώντας σαν βάση την κρυπτογραφική συνάρτηση SHA-1 [SHA 95]. Το κλειδί του κόμβου προκύπτει από την IP διεύθυνση μετά την εφαρμογή του SHA-1 , ενώ για τα αντικείμενα το κλειδί προκύπτει από την εφαρμογή του SHA-1 στα δεδομένα. Το μήκος (m bits) του κλειδιού ή αλλιώς αναγνωριστικού (identifier) είναι σταθερό για όλο το δίκτυο και πρέπει να είναι αρκετά μεγάλο, ώστε η πιθανότητα δύο αντικείμενα/κόμβοι να αντιστοιχιστούν στο ίδιο κλειδί να είναι αμελητέα. Τα κλειδιά όλου του δικτύου τοποθετούνται ταξινομημένα με βάση την συνάρτηση modulo 2^m σε ένα κύκλο ο οποίος ονομάζεται Chord ring. Κάθε κόμβος είναι υπεύθυνος για τα κλειδιά που βρίσκονται πριν από το κλειδί του στο κύκλο αυτό. Για ένα κλειδί K, λοιπόν, ο υπεύθυνος κόμβος είναι ο διάδοχος στο Chord ring ή αλλιώς ο successor(K). Για να διατηρηθεί η προηγούμενη συνθήκη, κατά την είσοδο ενός νέου κόμβου K στο δίκτυο, θα πρέπει να γίνει μια αναδιοργάνωση των κλειδιών που βρίσκονται στον successor(K), ενώ η αντίστροφη διαδικασία ακολουθείται κατά την έξοδο του κόμβου K από το δίκτυο. Στο παράδειγμα των σχημάτων 3.7 και 3.8 έχουμε ένα Chord ring με m=6. Το συγκεκριμένο έχει 10 κόμβους και αποθηκεύει πέντε κλειδιά. Ο διάδοχος (successor) του αναγνωριστικού 10 είναι ο κόμβος 14, οπότε το κλειδί 10 θα αποθηκευτεί σε αυτό το κόμβο. Όμοια, αν ένας κόμβος εισερχόταν με το αναγνωριστικό 26, θα γινόταν υπεύθυνος για το κλειδί 24 και θα έπαιρνε τα δεδομένα του από τον κόμβο με αναγνωριστικό 32.

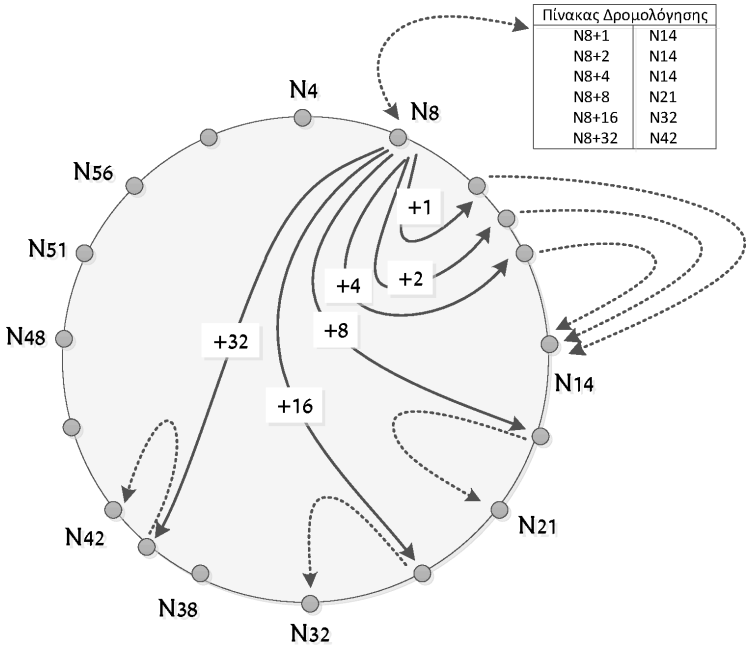
Από τα παραπάνω προκύπτει ότι κάθε κόμβος στο Chord ring αρκεί να γνωρίζει τον διάδοχο του ώστε να μπορεί να διασχίσει όλο το δίκτυο. Έτσι για την εύρεση ενός αναγνωριστικού, το ερώτημα θα διασχίσει ένα μονοπάτι από διαδοχικούς κόμβους (successor nodes), μέχρι να βρεθεί μεταξύ δύο διαδοχικών κόμβων των οποίων τα αναγνωριστικά τους περικλείουν το προς εύρεση αναγνωριστικό. Η απάντηση θα βρίσκεται στο δεύτερο στη σειρά κόμβο. Ένα παράδειγμα έχει σχεδιαστεί στο σχήμα 3.7, όπου ο κόμβος 8 κάνει μια αναζήτηση για το κλειδί 54, οπότε ενεργοποιεί την λειτουργία find_successor, η οποία επιστρέφει μετά το πέρας της αναζήτησης τον υπεύθυνο κόμβο για το κλειδί αυτό, τον κόμβο 56. Το ερώτημα πηγαίνει σε κάθε κόμβο στο κύκλο μεταξύ του κόμβου 8 και του κόμβου 56, και η απάντηση επιστρέφει από την αντίστροφη διαδρομή. Κάτι τέτοιο όμως είναι μη αποδοτικό, καθώς υπάρχει η πιθανότητα να χρειαστεί να διασχίσει όλους τους κόμβους του δικτύου για να βρει μια τιμή. Για το λόγο αυτό το Chord επιβάλλει την αποθήκευση επιπλέον πληροφορίας εκτός του διάδοχου κόμβου, δημιουργώντας ένα πίνακα δρομολόγησης ανά κόμβο.

Ειδικότερα, έστω m τα bits ενός κλειδιού στο Chord ring, τότε κάθε κόμβος (έστω ο κόμβος n) διατηρεί ένα πίνακα δρομολόγησης με μέγιστο m καταχωρήσεις, το οποίο αποκαλείται και ως finger table. Η i-οστή καταχώρηση ενός κόμβου στον πίνακα αυτό περιέχει τον κόμβο s, ο οποίος



Σχήμα 3.7: *Απεικόνιση ενός Chord ring με 10 κόμβους και 5 αντικείμενα. Απεικόνιση του μονοπατιού που ακολουθεί το ερώτημα για την αναζήτηση του κλειδιού 54 από τον κόμβο 8*

απέχει από τον κόμβο n κατ' ελάχιστο 2^{i-1} στο Chord ring, ή αλλιώς $s = successor(n + 2^{i-1})$ με $1 \leq i \leq m$ (πράξεις με βάση modulo 2^m). Η εγγραφή στο finger table περιέχει εκτός από το αναγνωριστικό του κόμβου στο Chord ring και την IP διεύθυνση. Στο σχήμα **??** φαίνεται το finger table του κόμβου 8 και η πρώτη εγγραφή του δείχνει προς το κόμβο 14, σύμφωνα με το αποτέλεσμα της παραπάνω σχέσης $(8 + 2^0) mod 2^6 = 9$, ενώ αντίστοιχα η τελευταία εγγραφή δείχνει προς το κόμβο $(8 + 2^5) mod 2^6 = 40$. Με τον τρόπο αυτό, οι κόμβοι αποθηκεύουν ένα μικρό μέρος του συνολικού πλήθους των κόμβων με την πλειοψηφία αυτών να βρίσκονται σε κοντινή απόσταση στο Chord ring. Βέβαια, αυτό έχει σαν αποτέλεσμα να μην περιέχεται όλη η απαιτούμενη πληροφορία στο finger table ενός κόμβου, ώστε να προκύπτει άμεσα ο υπεύθυνος κόμβος για ένα τυχαίο κλειδί k. Σε αυτή την περίπτωση, ο κόμβος στέλνει ερώτημα στο πιο κοντινό πιθανό προκάτοχο (predecessor) του κλειδιού k. Το ερώτημα αυτό προωθείται από κόμβο σε κόμβο με το ίδιο κριτήριο μέχρι να φτάσει στον προκάτοχο, δηλαδή τον κόμβο που θα έχει διάδοχο με αναγνωριστικό μεγαλύτερο του k, οπότε και επιστρέφεται ο διάδοχος αυτός σαν απάντηση στον ερωτώντα κόμβο. Το πλεονέκτημα του αλγορίθμου είναι πως με μεγάλη πιθανότητα, ο αριθμός των κόμβων που χρειάζεται να ερωτηθούν για ένα κλειδί k είναι O(logN).



Σχήμα 3.8: *Απεικόνιση του πίνακας δρομολόγησης ενός κόμβου σε ένα Chord ring με 10 κόμβους. Επίσης απεικονίζονται ενδεικτικά μονοπάτια.*

Κατά την είσοδο ενός κόμβου στο σύστημα θα πρέπει να μετακινηθούν κάποια δεδομένα, καθώς και να ενημερωθούν κάποιες εγγραφές σε μερικά finger table, έτσι ώστε να διασφαλιστεί η ορθή λειτουργία των αναζητήσεων. Ο μηχανισμός αυτός προβλέπεται από το Chord και ονομάζεται πρωτόκολλο σταθεροποίησης (stabilization protocol). Στο πλαίσιο αυτό του μηχανισμού γίνονται περιοδικά οι ακόλουθες λειτουργίες. Πρώτον, κάθε κόμβος n ελέγχει αν υπάρχει κάποιος ενδιάμεσος κόμβος στο διάστημα $(n, n.successor)$ και αναλόγως ενημερώνει τον successor. Δεύτερον, ενημερώνονται όλες οι εγγραφές στο finger table ώστε να ακολουθούν την σχέση $finger(i) = find\_successor(n + 2^{i-1})$. Τρίτον, γίνεται έλεγχος ότι ο predecessor είναι ενεργός.

Μια επιπλέον παράμετρος που επηρεάζει σημαντικά την αξιοπιστία του δικτύου είναι οι βλάβες των κόμβων, καθώς σε περίπτωση που αυτές προκύπτουν με αυξημένους ρυθμούς μπορεί να απομονώσουν κάποιο κόμβο και να τον οδηγήσουν σε λανθασμένες απαντήσεις στα διάφορα ερωτήματα που καλείται να απαντήσει. Η λύση στο πρόβλημα αυτό είναι η διατήρηση μιας λίστας μεγέθους r από successors αντί για έναν ανά κόμβο. Έτσι σε περίπτωση αποτυχίας του πρώτου, το ερώτημα μπορεί να προωθηθεί άμεσα στο δεύτερο στη λίστα. Για να απομονωθεί ο κόμβος θα

πρέπει να αποτύχουν ταυτόχρονα και όλοι οι r κόμβοι, κάτι που είναι αρκετά απίθανο. Οπότε αυξάνοντας το r μπορούμε να πετύχουμε διαφορετικά επίπεδα αξιοπιστίας ανάλογα με τις απαιτήσεις της εφαρμογής και τις δυνατότητες των πρωτοκόλλων επικοινωνίας/φυσικών μέσων.

3.3.3 Pastry

Το Pastry βασίζεται στον αλγόριθμο δρομολόγησης με βάση τα προθέματα του Plaxton [C. Plaxton], με τον οποίο δημιουργείται ένα αυτόδιαχειριζόμενο κατανεμημένο δίκτυο, όπου κάθε κόμβος δρομολογεί ερωτήματα και αλληλεπιδρά με τοπικές εφαρμογές. Κάθε κόμβος έχει ένα αναγνωριστικό 128-bit που αναφέρεται ως NodeID. Αυτό χρησιμοποιείται για την τοποθέτηση του κόμβου σε ένα κυκλικό χώρο στο οποίο βρίσκονται ταξινομημένα όλα τα αναγνωριστικά και έχει τιμές από $0 - 2^{128-1}$. Το αναγνωριστικό κάθε κόμβο επιλέγεται από με μια τυχαία συνάρτηση, η οποία έχει ομοιόμορφη κατανομή. Για ένα δίκτυο N κόμβων, το Pastry μπορεί να δρομολογήσει ένα κλειδί στο αριθμητικά εγγύτερο κόμβο σε λιγότερα από $log_B N$ βήματα υπό κανονική λειτουργία (Η παράμετρος $B = 2^b$ ρυθμίζεται κατά την κατασκευή του δικτύου και είναι συνήθως 4.). Τα αναγνωριστικά και τα κλειδιά θεωρούνται μια ακολουθία ψηφίων με βάση το B. Το Pastry δρομολογεί μηνύματα στους κόμβους με το εγγύτερο αναγνωριστικό στο δοθέν κλειδί.

Ένα μήνυμα προωθείται από ένα κόμβο προς ένα άλλο, ο οποίος διαθέτει ένα πρόθεμα του υπό αναζήτηση κλειδιού τουλάχιστον κατά ένα ψηφίο (ή b bits) μεγαλύτερο από τον προηγούμενο κόμβο. Όπως φαίνεται και στο σχήμα 3.9 κάθε κόμβος διατηρεί ένα πίνακα δρομολόγησης, ένα σύνολο γειτόνων και ένα σύνολο κόμβων φύλλων. Ο πίνακας δρομολόγησης έχει σχεδιαστεί με $log_B N$ σειρές, όπου κάθε σειρά δέχεται $B - 1$ εγγραφές. Οι $B - 1$ εγγραφές στην σειρά n του πίνακα, αναφέρονται σε κόμβους όπου το αναγνωριστικό τους μοιράζεται το ίδιο πρόθεμα μήκους n ψηφίων με το αναγνωριστικό του κόμβου, ενώ το n+1 ψηφίο του αναγνωριστικού τους έχει κάποια από τις $B-1$ διαφορετικές τιμές από αυτή του n+1 ψηφίου του κόμβου. Επίσης, κάθε εγγραφή στον πίνακα περιέχει και την ανάλογη διεύθυνση IP του κόμβου με το κατάλληλο πρόθεμα, ο οποίος επιλέγεται με βάση την μετρική της εγγύτητας. Η επιλογή της παραμέτρου b κατά την αρχικοποίηση του δικτύου πρέπει να γίνει προσεχτικά σταθμίζοντας μεταξύ του μεγέθους του κατειλημμένου κομματιού του πίνακα δρομολόγησης [περίπου $(log_B N) * (B - 1)$ εγγραφές] και του μέγιστου πλήθους των βημάτων που απαιτούνται για την δρομολόγηση μεταξύ δύο οποιοδήποτε κόμβων $(log_B N)$. Θέτοντας $b = 4$ και $N = 10^6$ κόμβους, ο πίνακας δρομολόγησης θα περιέχει κατά μέσο όρο 75 εγγραφές και ο εκτιμώμενος αριθμός βημάτων δρομολόγησης είναι 5. Το σύνολο των γειτονικών κόμβων M, περιέχει τα αναγνωριστικά και τις IP διευθύνσεις των |M| κόμβων που είναι πιο κοντά στον τοπικό κόμβο. Η μετρική για την απόσταση που χρησιμοποιείται βασίζεται στα βήματα δρομολόγησης της IP κίνησης μεταξύ των κόμβων. Το σύνολο κόμβων φύλλων L (leaf) περιέχει |L|/2 κόμβους με μεγαλύτερο αναγνωριστικό και |L|/2 με μικρότερο αναγνωριστικό σε σχέση με τον τοπικό κόμβο. Τυπικές τιμές για |L| και |M| είναι B και $2 * B$. Ακόμα

και σε περίπτωση που εμφανιστεί βλάβη σε δυο κόμβους ταυτόχρονα, η παράδοση των μηνυμάτων κάποια στιγμή θα ολοκληρωθεί, εγγυημένα με αξιοπιστία και ανάνηψη από σφάλματα, εκτός εάν |L|/2 κόμβοι παρουσιάσουν βλάβη ταυτόχρονα.

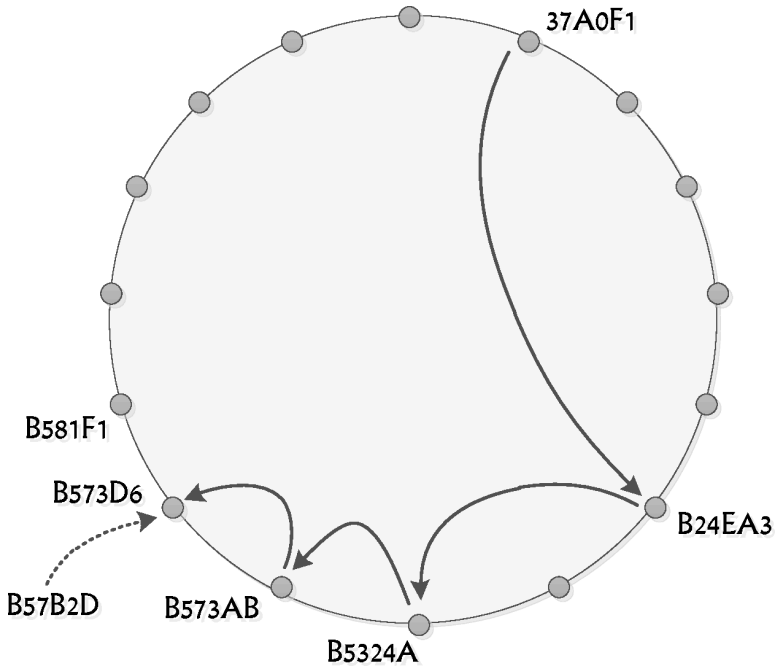
Πίνακας 3.1: *Πίνακας δρομολόγησης ενός κόμβου στο Pastry με αναγνωριστικό 37A0x, b=4. Τα ψηφία είναι στο δεκαεξαδικό σύστημα και το x είναι ένα οποιοδήποτε ψηφίο*

0x	1x	2x	**3x**	4x	...	Dx	Ex	Fx
30x	31x	32x	...	**37x**	38x	...	3Ex	3Fx
370x	371x	372x	...	**37Ax**	37Bx	...	37Ex	37Fx
37A0x	37A1x	37A2x	...	37ABx	37Cx	37ADx	37AEx	37AFx

Πίνακας 3.2: *Κατάσταση δρομολόγησης ενός κόμβου του Pastry με αναγνωριστικό 37A0F1, b=4, L=16, M=32*

Κόμβος με αναγνωριστικό 37A0F1			
Μικρό σύνολο κόμβων φύλλων (leaf set - smaller)			
37A001	37A011	37A022	37A033
37A044	37A055	37A066	37A077
Μεγάλο σύνολο κόμβων φύλλων (leaf set - larger)			
37A0F2	37A0F4	37A0F6	37A0F8
37A0FA	37A0FB	37A0FC	37A0FE
Σύνολο γειτονικών κόμβων (neighborhood set)			
1A223B	1B3467	245AD0	2670AB
3612AB	37890A	390AF0	3912CD
46710A	477810	4881AB	490CDE
279DE0	290A0B	510A0C	5213EA
11345B	122167	16228A	19902D
221145	267221	28989C	199ABC

Όταν ένας νέος κόμβος (με αναγνωριστικό X) εισέλθει στο δίκτυο, θα πρέπει να αρχικοποιήσει την κατάσταση του πίνακα δρομολόγησης και να παρουσιαστεί στους υπόλοιπους κόμβους. Για αυτό θα πρέπει να γνωρίζει την IP διεύθυνση ενός κόμβου που ανήκει ήδη στο δίκτυο. Μια μικρή λίστα κόμβων αρχικής επικοινωνίας, με βάση μια μετρική εγγύτητας (όπως το round trip time -RTT προς κάθε κόμβο) για την βελτίωση της επίδοσης, δίνεται στο νέο κόμβο, ώστε να επιλέξει ένα κοντινό κόμβο για την πρώτη του επικοινωνία (με αναγνωριστικό A). Στην συνέχεια, ο κόμβος X στέλνει μέσω του A ένα ειδικό μήνυμα τύπου join με κλειδί ίσο με το αναγνωριστικό X. Το Pastry δρομολογεί το μήνυμα στον υφιστάμενο κόμβο Z που έχει το πιο κοντινό αναγνωριστικό στο X. Στην συνέχεια οι κόμβοι A,Z καθώς και όλοι οι ενδιάμεσοι τους, στέλνουν τους πίνακες



Σχήμα 3.9: *Pastry: Τα σύνολα κόμβων γειτονικών και κόμβων φύλλων στο Pastry. Πίνακες δρομολόγησης. Παράδειγμα δρομολόγησης στο Pastry από τον κόμβο 37A0F1 για το κλειδί B57B2D.*

δρομολόγησης που έχουν στο X. Τελειώνοντας το X ενημερώνει τους κόμβους που πρέπει να είναι ενήμεροι για την άφιξή του στο δίκτυο. Επειδή, ο κόμβος A είναι κοντινός με τον κόμβο X, θα έχουν και όμοια σύνολα γειτόνων. Στην γενική περίπτωση όπου οι κόμβοι A και X δεν έχουν κανένα κοινό πρόθεμα, έστω A_i ο κόμβος στον πίνακα δρομολόγησης του A στη σειρά i. Το $_0$ περιέχει τις κατάλληλες τιμές για το X_0 καθώς οι εγγραφές στη σειρά 0 είναι ανεξάρτητες από το αναγνωριστικό του κάθε κόμβου, ενώ οι υπόλοιπες εγγραφές στις υπόλοιπες σειρές δεν χρησιμεύουν στον X. Οι τιμές για το X_1 θα προκύψουν από το B_1 όπου το B είναι ο πρώτος κόμβος στο μονοπάτι από τον A προς τον Z. Οι εγγραφές B_1 και X_1 μοιράζονται κοινό πρόθεμα διότι οι κόμβοι X και B έχουν το ίδιο πρώτο ψηφίο στο αναγνωριστικό τους. Με το πέρας της διαδικασίας, ο X μεταδίδει το πίνακα δρομολόγησης του, στους κόμβους γείτονες και κόμβους φύλλα ώστε να ενημερωθούν και αυτοί.

3.3.4 Tapestry

Η αρχιτεκτονική του Tapestry [Zhao 04], όπως και το Pastery, βασίζεται σε μια παραλλαγή του κατανεμημένου αλγόριθμου αναζήτησης του Plaxton [C. Plaxton], ενισχυμένη ώστε να προσθέτει

διαθεσιμότητα, κλιμακωσιμότητα και προσαρμοστικότητα σε βλάβες και κακόβουλες ενέργειες. Από το [C. Plaxton] προτείνεται μια κατανεμημένη δομή δεδομένων, βελτιστοποιημένη για την αναζήτηση ονοματισμένων αντικειμένων που συνδέονται με ένα κεντρικό κόμβο - ρίζα μιας δεντρικής δομής. Το Tapestry χρησιμοποιεί πολλαπλούς κόμβους - ρίζες για κάθε αποθηκευμένο αντικείμενο για την αποφυγή ενός κεντρικού σημείου βλάβης. Σε ένα πλέγμα τύπου Plaxton, οι κόμβοι μπορούν να έχουν το ρόλο του εξυπηρετητή (που αποθηκεύει δεδομένα), του δρομολογητή (που προωθεί μηνύματα) και του πελάτη (που στέλνει αιτήματα). Χρησιμοποιεί τοπικούς χάρτες δρομολόγησης σε κάθε κόμβο, για την σειριακή προώθηση των μηνυμάτων στον παραλήπτη με βάση το αναγνωριστικό του, πηγαίνοντας ψηφίο προς ψηφίο, όπως πχ $***7 \rightarrow **97 \rightarrow *297 \rightarrow 3297$. Η τεχνική αυτή είναι όμοια με την τεχνική που χρησιμοποιείται στην δρομολόγηση υποδικτύων IP διευθύνσεων τύπου CIDR [Rekhter 93]. Η ανάλυση των ψηφίων μπορεί να γίνει ξεκινώντας είτε από το πιο δεξιό ψηφίο είτε από το πιο αριστερό ψηφίο. Ο χάρτης δρομολόγησης κάθε κόμβου έχει πολλαπλά επίπεδα, κάθε ένα από τα οποία αναπαριστά ένα ταιριαστό επίθεμα (suffix) μέχρι κάποιο ψηφίο στο χώρο συντεταγμένων. Το αναγνωριστικό του ν-ιοστού κόμβου του μονοπατιού που φθάνει ένα μήνυμα έχει τα ίδια ν τελευταία ψηφία με αυτά του αναγνωριστικού του παραλήπτη. Για να βρεθεί ο επόμενος ενδιάμεσος κόμβος στο βήμα αυτό εξετάζεται ο χάρτης του ν+1 επιπέδου, ώστε να βρεθεί μια εγγραφή με ψηφίο ίδιο με το αντίστοιχο του παραλήπτη. Αυτή η μέθοδος δρομολόγησης, εγγυάται ότι κάθε κόμβος στο δίκτυο μπορεί να βρεθεί με το πολύ $log_B N$ βήματα σε ένα σύστημα με N κόμβους που χρησιμοποιούν αναγνωριστικά με βάση το B. Δεδομένου ότι ο χάρτης δρομολόγησης κάθε κόμβου υποθέτει ότι τα πρώτα ψηφία όλα ταιριάζουν με την κατάληξη του αναγνωριστικού του κόμβου, αρκεί η διατήρηση ενός μικρού σταθερού μεγέθους (B) εγγραφών σε κάθε επίπεδο, το οποίο σημαίνει ότι και ολόκληρος ο χάρτης θα έχει ένα σταθερό μέγεθος $M = (entries/map) * no.of.maps = B * log_B N$. Η αναζήτηση και δρομολόγηση στο Tapestry είναι όμοια με το Plaxton, που βασίζεται ταιριάζοντας τις καταλήξεις των αναγνωριστικών όπως περιγράφηκε. Οι χάρτες δρομολόγησης οργανώνονται σε επίπεδα, όπου κάθε ένα περιέχει εγγραφές που δείχνουν σε μια ομάδα κόμβων που είναι πιο κοντά στην λεξικογραφική απόσταση για την κατάληξη του επιπέδου. Επίσης, κάθε κόμβος κρατά μια λίστα με κόμβους τους οποίους θεωρεί γείτονες. Το Tapestry αποθηκεύει τις τοποθεσίες όλων των αντιγράφων των αντικειμένων για να αυξήσει την σημασιολογική ευελιξία, και να επιτρέψει σε εφαρμογές να επιλέγουν αντίγραφα βάση κάποιων κριτηρίων που θα ορίσουν αυτές, όπως η ημερομηνία. Για την ορθή λειτουργία του δικτύου θα πρέπει κάθε αναγνωριστικό ενός αντικειμένου να αντιστοιχίζεται σε ένα κόμβο. Όταν αυτή η αντιστοίχηση δεν είναι ένα-προς-ένα θα πρέπει να υπάρχει ένας μηχανισμός που εγγυάται ότι κάθε αναγνωριστικό μπορεί να αντιστοιχηθεί με ένα μοναδικό και υφιστάμενο κόμβο στο δίκτυο. Η τεχνική αυτή ονομάζεται surrogate routing (δρομολόγηση με αντικατάσταση) και με βάση αυτή επιλέγονται με αύξοντα τρόπο οι κόμβοι-ρίζα κατά την διαδικασία δημοσίευσης.

3.3.5 Kademlia

Το Kademlia βασίζεται και αυτό σε ένα Κατανεμημένο Πίνακα Κατακερματισμού. Κάθε κόμβος και κάθε αντικείμενο στο δίκτυο έχει ένα κλειδί στο χώρο συντεταγμένων που είναι μεγέθους 160-bit. Τα αντικείμενα αποθηκεύονται σε κόμβους με αναγνωριστικό κοντινό στο κλειδί. Για την εύρεση των κοντινών κόμβων χρησιμοποιείται ένας αλγόριθμος δρομολόγησης βασισμένος στα κλειδιά. Μια καινοτομία του Kademlia είναι η χρησιμοποίηση της μετρικής XOR για τον υπολογισμό της απόστασης μεταξύ δύο κλειδιών-αναγνωριστικών. Η μετρική αυτή είναι συμμετρική και έτσι οι κόμβοι δέχονται αριθμό μηνυμάτων ανάλογα με το πλήθος των κόμβων που έχουν στο πίνακα δρομολόγησης τους. Είναι επίσης δυνατόν να σταλεί ένα μήνυμα σε οποιοδήποτε κόμβο με αποτέλεσμα να εφαρμόζονται κριτήρια επιλογής όπως η καθυστέρηση σε επίπεδο δικτύου μεταξύ των κόμβων, ενώ επιτρέπεται και η παράλληλη αποστολή ασύγχρονων μηνυμάτων.

Όπως αναφέραμε, η αναζήτηση ενός κλειδιού στο Kademlia βασίζεται στην απόσταση μεταξύ δυο κλειδιών όπως αυτή ορίζεται από την μετρική XOR. Έστω ότι έχουμε δυο αναγνωριστικά a,b τότε η απόσταση τους θα είναι ίση με την δυαδική πράξη του αποκλειστικό Ή (exclusive OR – XOR), δηλαδή : $d(a,b) = a \oplus b = d(b,a)\ a = b \Leftrightarrow d(a,b) = 0\ d(a,b) > 0 \Leftrightarrow a \neq b$ $d(a,b) = d(b,a)$

Όπως και στο Chord ring έτσι και εδώ ο χώρος συντεταγμένων είναι μιας κατεύθυνσης. Δηλαδή, πολλαπλές αναζητήσεις για το ίδιο κλειδί θα καταλήξουν εγγυημένα στον ίδιο κόμβο, ανεξάρτητα το κόμβο που τις ξεκίνησε. Για αυτό η προσωρινή αποθήκευση ζευγαριών κλειδιού-τιμής είναι εφικτή και επιταχύνει τις αναζητήσεις, κατανέμοντας το φορτίο και σε άλλους κόμβους.

Κάθε κόμβος έχει οργανώσει τον πίνακα δρομολόγησης σε k-κάδους (k-buckets). Κάθε κάδος περιέχει μια λίστα με κόμβους με απόσταση 2^i έως 2^{i+1} από τον ίδιο. Για να είναι εφικτή η επικοινωνία, κάθε εγγραφή περιέχει την IP διεύθυνση του κόμβου, την πόρτα UDP/TCP και το αναγνωριστικό του.

Τα μηνύματα του δικτύου είναι τα εξής:

PING το μήνυμα αυτό χρησιμοποιείται για να διαπιστωθεί εάν κάποιος κόμβος είναι ενεργός και αποκρίσιμος

STORE το μήνυμα αυτό χρησιμοποιείται για την αποθήκευση ενός ζεύγους κλειδιού-τιμής, το οποίο έχει σαν παράμετρο

FIND_NODE το μήνυμα αυτό έχει σαν παράμετρο ένα κλειδί και επιστρέφει τους k-κοντινότερους κόμβους σε αυτό με όλα τα στοιχεία τους (IP διεύθυνση, πόρτα UDP/TCP, αναγνωριστικό)

FIND_VALUE το μήνυμα αυτό είναι πανομοιότυπο με το `FIND_NODE`, ενώ στην περίπτωση που βρεθεί το ζεύγος όπου ταιριάζει με το κλειδί του μηνύματος, τότε επιστρέφεται το ζεύγος κλειδιού-τιμής

Η πιο συχνή και συνάμα λειτουργία στο Kademlia είναι η εύρεση των k-κοντινότερων κόμβων σε κάποιο κλειδί. Αυτή η αναζήτηση ενεργοποιείται με την επιλογή Χ κόμβων από το κοντινότερο στο κλειδί μη-άδειο κάδο και με την αποστολή παράλληλων και ασύγχρονων μηνυμάτων `FIND_NODE`. Αν κανένα μήνυμα δεν επιστρέψει κάποιο κοντινότερο κόμβο, τότε έχουμε ήδη τους κοντινότερους κόμβους, αλλιώς συνεχίζει μέχρι να εξαντλήσει τους διαθέσιμους κόμβους. Όταν τελειώσει αυτή η διαδικασία, γίνεται αποστολή ενός μηνύματος `FIND_VALUE` στους k-κοντινότερους κόμβους και επιστρέφεται το ζεύγους κλειδιού-τιμής που αναζητείται. Για την εισαγωγή ενός κόμβου στο δίκτυο, αρκεί να υπάρχουν τα στοιχεία ενός τουλάχιστον κόμβου που ανήκει ήδη στο δίκτυο. Για την συμπλήρωση του πίνακα δρομολόγησης ο νέος κόμβος κάνει εικονικές αναζητήσεις, ώστε να αποκτήσει στοιχεία για άλλους κόμβους που θα τον βοηθήσουν στις μελλοντικές αναζητήσεις.

3.4 Δίκτυα Ομότιμων Κόμβων και Υπολογιστικό Πλέγμα

Η ιδέα να χρησιμοποιηθεί ένα δίκτυο ομότιμων κόμβων για την αναζήτηση αντιγράφων σε μια υποδομή υπολογιστικού πλέγματος έχει αναφερθεί ξανά στη διεθνή βιβλιογραφία. Πιο συγκεκριμένα, στην δημοσίευση του [Foster 03, Iamnitchi 02], επισημαίνεται ότι οι επιστημονικές κοινότητες του υπολογιστικού πλέγματος και των δικτύων ομότιμων κόμβων έχουν πολλά κοινά και μπορούν να μοιραστούν ακόμη περισσότερα. Υπηρεσίες που βασίζονται σε δίκτυα ομότιμων κόμβων μπορούν να υποστηρίξουν από μερικούς δεκάδες μέχρι εκατομμύρια συμμετέχοντες που χρησιμοποιούν ταυτόχρονα το δίκτυο, χωρίς αλλαγές στην εφαρμογή ή ειδικές ρυθμίσεις στο συνολικό σύστημα. Το δίκτυο έχει σχεδιαστεί με ένα τρόπο κλιμακώσιμο, με την ιδιότητά του αυτή να αυξάνεται ευθέως ανάλογα με τον αριθμό των συμμετεχόντων, σε αντίθεση με τις παραδοσιακές τεχνικές πελάτη-εξυπηρετητή, όπου η συνολική απόδοση του δικτύου μειώνεται όσο αυξάνονται οι πελάτες που προσπαθούν να προσπελάσουν το κεντρικό εξυπηρετητή. Οι ίδιοι οι συγγραφείς του Giggle Framework, ανατρέχουν στην διεθνή βιβλιογραφία στο χώρο των δικτύων ομότιμων κόμβων, για να οριοθετήσουν την σχετική βιβλιογραφία για την εργασία τους. Και στις δύο περιπτώσεις άλλωστε, το πρόβλημα είναι όμοιο: Δοθέντος ενός μοναδικού αναγνωριστικού, αναζήτησε με τρόπο κατανεμημένο και κλιμακώσιμο τον υπολογιστικό/αποθηκευτικό πόρο που χαρακτηρίζεται από το αναγνωριστικό αυτό. Πάνω από την υπηρεσία αναζήτησης μερικά συστήματα παρέχουν και προστιθέμενες υπηρεσίες στους συμμετέχοντες κόμβους, όπως μεταφορά αρχείων και παρακολούθηση οπτικοακουστικού περιεχομένου. Παρόλ' αυτά τα δίκτυα ομότιμων κόμβων χαρακτηρίζονται από την βασική τους υπηρεσία, αυτή της αναζήτησης, και για αυτό πολύ συχνά αποκαλούνται και ως συστήματα αναζήτησης (Lookup Systems). Οι αρχιτεκτονικές ομότιμων κόμβων μπορούν να διαχωριστούν σε δύο βασικές κατηγορίες, ανάλογα με την δομή του δικτύου. Τα δομημένα συστήματα ή αλλιώς τα συστήματα με κατανεμημένους πίνακες κατακερματισμού (Distributed Hash Tables – DHT), όπως το Kademlia, το Chord, το Tapestry, το CAN κ.ά. θέτουν

μια συγκεκριμένη εικονική δομή, η οποία διευθετεί τους κόμβους σε καθορισμένες θέσεις, καθώς εισέρχονται στο δίκτυο. Από την άλλη πλευρά, αδόμητα δίκτυα όπως το Gnutella αφήνουν τους κόμβους ελεύθερους να συμμετέχουν από οποιοδήποτε μέρος του δικτύου και ο γράφος διασύνδεσης όλων των κόμβων μεταξύ τους μοιάζει με αυτόν ενός δικτύου εκθετικού βαθμού (power-law network) [Ripeanu 02a]. Κάθε οικογένεια συστημάτων από τις παραπάνω, έχει τα δικά της πλεονεκτήματα και μειονεκτήματα: Σε δομημένα δίκτυα η διαδικασία αναζήτησης είναι ισχυρά ντετερμινιστική (σχεδόν πάντα θα επιστρέψει ένα αποτέλεσμα, εφόσον αυτό είναι διαθέσιμο) και κάθε διαδικασία θα επιτύχει ή θα αποτύχει σχεδόν σίγουρα σε ένα προκαθορισμένο αριθμό βημάτων. Αυτά τα βήματα είναι συνήθως της τάξεως του log(N), όπου N είναι ο αριθμός των συμμετεχόντων κόμβων.

Στα μη δομημένα συστήματα, οι αναζητήσεις γίνονται πλημμυρίζοντας το δίκτυο με μηνύματα, τεχνική γνωστή ως πλημμύρα (flooding). Με την τεχνική αυτή είναι πολύ πιθανό κάποιος κόμβος να απαντήσει για κάποιο ερώτημα, αλλά δεν είναι σίγουρο ότι η αναζήτηση θα επιτύχει. Εάν το αντικείμενο προς αναζήτηση δεν είναι δεν είναι δημοφιλές και είναι αποθηκευμένο σε κόμβο μακρινό, με βάση το πλήθος των κόμβων που απέχει από τον αρχικό, τότε η αναζήτηση δεν θα φτάσει ποτέ αυτό σε τον κόμβο. Επίσης, η τεχνική αυτή δημιουργεί πολύ περισσότερα μηνύματα σε σύγκριση με αυτές που χρησιμοποιούνται σε δομημένα δίκτυα, επομένως γίνεται μη βέλτιστη χρησιμοποίηση του δικτύου. Το μόνο πλεονέκτημα των μη δομημένων δικτύων είναι η δυνατότητα που έχουν να χειρίζονται αναζητήσεις του τύπου ελεύθερου κειμένου (free text queries) πιο αποδοτικά και σε πολύ λιγότερα βήματα, κάτι το οποίο είναι βασικό χαρακτηριστικό των δικτύων εκθετικού βαθμού (power-law networks) [Adamic 01].

Στο περιβάλλον του υπολογιστικού πλέγματος, η αναζήτηση δεδομένων αντιμετωπίζεται είτε σε επίπεδο εφαρμογής είτε από ανεξάρτητη υπηρεσία, την υπηρεσία μεταδεδομένων (Metadata Service) [Fitzgerald 01]. Η υπηρεσία αυτή διατηρεί αποθηκευμένα στοιχεία για το είδος και την ποιότητα των δεδομένων που βρίσκονται στο υπολογιστικό πλέγμα. Το σημαντικότερο πρόβλημα είναι πως θα βρεθούν όλα τα διαθέσιμα αντίγραφα, δοθέντος ενός μοναδικού αναγνωριστικού όπως το λογικό όνομα αρχείου. Είναι, επίσης, ισχυρά επιθυμητό ότι η διαδικασία αναζήτησης των αντιγράφων δεδομένων να πραγματοποιηθεί με τα ελάχιστα δυνατά βήματα, ενώ θα διατηρεί τις ιδιότητες της επεκτασιμότητας και της διαθεσιμότητας του επιπέδου αναζήτησης. Είναι προφανές πως ένα κεντρικοποιημένο σύστημα αποθήκευσης δυάδων λογικών και φυσικών ονομάτων αρχείων (LFN, PFN) θα έλυνε το πρόβλημα σε ένα βήμα, αλλά αυτό το σύστημα δεν είναι ούτε επεκτάσιμο ούτε διαθέτει μηχανισμούς υψηλής διαθεσιμότητας. Όσο αυξάνεται ο αριθμός των κόμβων που χρησιμοποιούνται στο επίπεδο αναζήτησης, ενώ κατανέμονται τα δεδομένα και τα ερωτήματα σε αυτά προς όλους αυτούς τους κόμβους, τόσο αυξάνονται και τα μηνύματα που πρέπει να διασχίσουν την ιεραρχία του συστήματος και να βρουν το κατάλληλο κόμβο που θα απαντήσει στο ερώτημά που φέρουν.

Τα δομημένα συστήματα ομότιμων κόμβων είναι σχεδιασμένα για να εξυπηρετούν αιτήσεις αποθήκευσης και αναζήτησης ζευγαριών κλειδιού-τιμής (key-value pairs). Τα κλειδιά είναι πάντα μοναδικά σε όλο το σύστημα και αποτελούν το μοναδικό αναγνωριστικό για την τιμή του ζευγαριού. Οι περισσότερες υλοποιήσεις κατανεμημένων πινάκων κατακερματισμού παράγουν κλειδιά χρησιμοποιώντας την τιμή του ζευγαριού, βάση κάποιας συνάρτησης κρυπτογράφησης όπως η SHA1 πάνω στα δεδομένα που αποθηκεύουν. Αυτή η μέθοδος παράγει ομοιόμορφες κατανομές κλειδιών σε ένα χώρο αναγνωριστικών μεγέθους 160bit. Ως αποτέλεσμα του παραπάνω, για να χρησιμοποιηθούν οι κατανεμημένοι πίνακες κατακερματισμού για την αναζήτηση αντιγράφων δεδομένων σε ένα περιβάλλον υπολογιστικού πλέγματος, πρέπει να κάνουμε τις κάτωθι υποθέσεις:

- Ένα ανεξάρτητο δίκτυο ομότιμων κόμβων θα χρησιμοποιείται για κάθε εικονικό οργανισμό
- Το κλειδί δεν θα παράγεται βάσει των δεδομένων/της τιμής του ζευγαριού του, αλλά βάσει του λογικού ονόματος αρχείου (LFN) που αντιπροσωπεύει. Θα είναι ένα μοναδικό αναγνωριστικό που θα χρησιμοποιείται σε όλες τις σχετικές υπηρεσίες και διαδικασίες.
- Η τιμή που θα αντιστοιχεί σε κάθε κλειδί και θα αποτελούν το ζευγάρι κλειδιού-τιμής, θα είναι ουσιαστικά μια λίστα που θα περιέχει όλες τις φυσικές τοποθεσίες των αντιγράφων δεδομένων (PFNs) για ένα λογικό όνομα αρχείου (LFN).

Επιπροσθέτως, στην ορολογία των δικτύων ομότιμων κόμβων, ο όρος δίκτυο αναφέρεται στις εικονικές συνδέσεις που δημιουργούνται σε επίπεδο εφαρμογής μεταξύ των φυσικών κόμβων ενός συστήματος. Αυτό πρακτικά σημαίνει ότι το δίκτυο μπορεί να δημιουργηθεί από μηχανήματα συνδεδεμένα στο υπολογιστικό πλέγμα και οι εφαρμογές των χρηστών να χρησιμοποιούν τα δεδομένα που αποθηκεύονται στο δίκτυο ομότιμων κόμβων μέσω καλά ορισμένων διεπαφών, όπως μια καλά τεκμηριωμένη προγραμματιζόμενη διεπαφή της εφαρμογής (Application Programming Interface).

Στην συνέχεια θα αναλύσουμε την υπηρεσία διαχείρισης αντιγράφων του πλεγμάτος και θα προτείνουμε βελτιώσεις σε αυτή με την εφαρμογή αλγορίθμων και τεχνικών από το ερευνητικό πεδίο των δικτύων ομότιμων κόμβων.

ΚΕΦΑΛΑΙΟ 4

Υπηρεσία Διαχείρισης Αντιγράφων

4.1 Μοντέλο συνδρομητή

Μία από τις πρώτες προσπάθειες αντιμετώπισης των αναγκών διαχείρισης δεδομένων στο Πλέγμα είναι το λογισμικό Grid Data Management Pilot (GDMP) [Samar 01]. Το λογισμικό GDMP αναπτύχθηκε από το ερευνητικό έργο European Data Grid με βάση τις ανάγκες του πειράματος Compact Muon Solenoid (CMS) [cms] το οποίο διεξάγεται στο ερευνητικό κέντρο CERN. Το συγκεκριμένο πείραμα δημιουργεί αρκετά TeraBytes δεδομένων σε ετήσια βάση, τα οποία αποθηκεύονται ως αντικείμενα σε μια αντικειμενοστραφή βάση δεδομένων, την Objectivity [Obj] . Τα δεδομένα αυτά αναλύονται, επεξεργάζονται και εικονοποιούνται από διάφορες ομάδες διεσπαρμένες γεωγραφικά σε όλο τον κόσμο, οπότε και δημιουργήθηκε η ανάγκη να υπάρχουν αντίγραφα τοπικά σε κάθε σύστημα επεξεργασίας των δεδομένων. Η αντιγραφή των δεδομένων (data replication) δεν είναι μια νέα έννοια, αλλά έχει χρησιμοποιηθεί σε μεγάλο βαθμό στο πεδίο των κατανεμημένων αρχιτεκτονικών σε βάσεις δεδομένων και συστήματα αρχείων. Για την ακρίβεια αντί για το νέο πρωτότυπο λογισμικό του GDMP θα μπορούσε να έχει χρησιμοποιηθεί ένα από τα ήδη υπάρχοντα συστήματα αντιγράφων των βάσεων δεδομένων, αλλά προτιμήθηκε μια νέα λύση που θα συνδεόταν σε μεγαλύτερο βαθμό με τα προηγμένα συστήματα ασφάλειας και μεταφοράς δεδομένων που προφέρει το Πλέγμα, ενώ θα έδινε την δυνατότητα να εφαρμοστούν πολιτικές πρόσβασης και διαχείρισης των δεδομένων προσαρμοσμένες στην λογική του Πλέγματος.

Το πρωτότυπο λογισμικό GDMP βασίζεται σε ένα μοντέλο συνδρομητή για την διαχείριση αντιγράφων δεδομένων. Κάθε κόμβος του Πλέγματος που θέλει να αποκτήσει αντίγραφα από την

κεντρική βάση δεδομένων (Objectivity), εγγράφεται στην κατάλληλη υπηρεσία και περιμένει για ειδοποιήσεις με λίστες αρχείων προς μεταφορά. Οι μεταφορές δεδομένων ξεκινούν από τους ενδιαφερόμενους πελάτες με ασύγχρονο τρόπο. Ένας πελάτης συμβουλεύεται την λίστα των νέων και των αλλαγμένων αρχείων της βάσης δεδομένων και προχωράει με την μεταφορά των χρήσιμων δεδομένων μέσω του πρωτοκόλλου GridFTP. Ειδικότερα, επειδή το GDMP είναι ουσιαστικά ένα σύνολο από εργαλεία της γραμμής εντολών, όλοι οι μηχανισμοί για ένα τυπικό σενάριο αντιγραφής δεδομένων ενεργοποιούνται με την παρέμβαση του χρήστη, καθώς δεν υπάρχει αυτόματη διαδικασία. Για παράδειγμα, στον εξυπηρετητή της υπηρεσίας για κάθε προσθήκη ή αλλαγή στην βάση δεδομένων θα πρέπει να ενημερωθούν όλοι οι συνδρομητές, ώστε να έχουν όλοι την ίδια εικόνα για την υπηρεσία (συνέπεια των δεδομένων). Η ενημέρωση αυτή προκαλείται με μη αυτόματο τρόπο από τον χρήστη με την εκτέλεση μια συγκεκριμένης εντολής στη γραμμή εντολών. Στην συνέχεια, στον κόμβο-πελάτη ο χρήστης θα πρέπει να τρέξει ένα αντίστοιχο εργαλείο στην γραμμή εντολών που λαμβάνει τις αλλαγές και συγχρονίζει τα δεδομένα με τον εξυπηρετητή της υπηρεσίας. Στις λίστες των αλλαγών μεταξύ πελάτη και εξυπηρετητή μπορούν να εφαρμοστούν και φίλτρα, επιτρέποντας σε κάθε κόμβο-πελάτη να εφαρμόσει πολιτικές αντιγραφής σε υποσύνολο δεδομένων. Τέλος, μέσα στην αρχιτεκτονική του GDMP έχει προβλεφτεί και η ανάνηψη από λάθη που μπορεί να συμβούν για λόγους πχ βλάβης στην δικτυακή σύνδεση μεταξύ πελάτη-εξυπηρετητή με επαναληπτικές προσπάθειες σύνδεσης.

Σχήμα 4.1: *Το μοντέλο του συνδρομητή για την διαχείριση αντιγράφων.*

Η συμβολή του GDMP στο σχεδιασμό της αρχιτεκτονικής Πλέγματος Δεδομένων είναι αρκετά σημαντική, καθώς η δομή του βοήθησε στην κατανόηση των απαραίτητων συστατικών που χρειάζονται για την εξέλιξή του Πλέγματος Δεδομένων. Τέλος, η οργάνωση των αντικείμενων σε αρχεία βάσεων δεδομένων οδήγησε μετέπειτα στην λογική της οργάνωσης των αρχείων σε συλλογές. Η ομαδοποίηση αυτή σχετικών δεδομένων σε δομές πιο υψηλού επιπέδου, έχει το πλεονέκτημα της διαχείρισης τάξεων μεγέθους λιγότερου όγκου μέτα-δεδομένων.

4.2 Κεντρικό μοντέλο

Μια από τις αδυναμίες του πρωτότυπου GDMP, είναι ότι δεν διατηρούσε πληροφορία που θα βοηθούσε στην καλύτερη διαχείριση και εκμετάλλευση των υφιστάμενων αντιγράφων. Σε αυτή τη βάση δημιουργήθηκε η υπηρεσία αναζήτησης αντιγράφων (Replica Location Service – RLS) [Stockinger 02b] από το Globus Toolkit.

Στην πρώτη της μορφή η υπηρεσία αυτή αποτελείται από ένα κατάλογο, ο οποίος διατηρεί την σχέση μεταξύ των λογικών ονομάτων των αρχείων και των διάφορων τοποθεσιών αποθήκευσής τους. Το λογικό όνομα αρχείου (Logical Filename – LFN) είναι μοναδικό αναγνωριστικό που χρησιμοποιείται από τους χρήστες και τις εφαρμογές για την αναφορά σε συγκεκριμένα αρχεία. Ο κατάλογος αντιγράφων διατηρεί επίσης και μια λίστα με φυσικά ονόματα αρχείων (Physical Filenames – PFNs) για κάθε λογικό όνομα. Τα φυσικά ονόματα είναι δομημένα όπως τα URL και περιγράφουν το πρωτόκολλο πρόσβασης, την διεύθυνση του κόμβου και της τοποθεσίας στο κόμβο αποθήκευσης για κάθε αντίγραφο. Με την απόκρυψη της λεπτομέρειας των φυσικών τοποθεσιών σε μια υπηρεσία αναζήτησης αντιγράφων, οι χρήστες και οι εφαρμογές μπορούν να χρησιμοποιήσουν το λογικό όνομα αρχείου, ώστε να ανεξαρτητοποιηθούν από τον κόμβο αποθήκευσης των δεδομένων και αλλά και το κόμβο επεξεργασίας τους. Τα λογικά ονόματα μπορούν επίσης να ομαδοποιηθούν σε συλλογές αρχείων, οι οποίες με αφαιρετική λογική αναπαριστούν συλλογές δεδομένων. Η διαμόρφωση αυτή προκύπτει από την ανάγκη πολλών εφαρμογών να επεξεργάζονται ομάδες αρχείων με διάφορα δεδομένα κατά την εκτέλεσή τους, οπότε και είναι απαραίτητη η ύπαρξη μηχανισμού ενιαίας διαχείρισης των αρχείων αυτών.

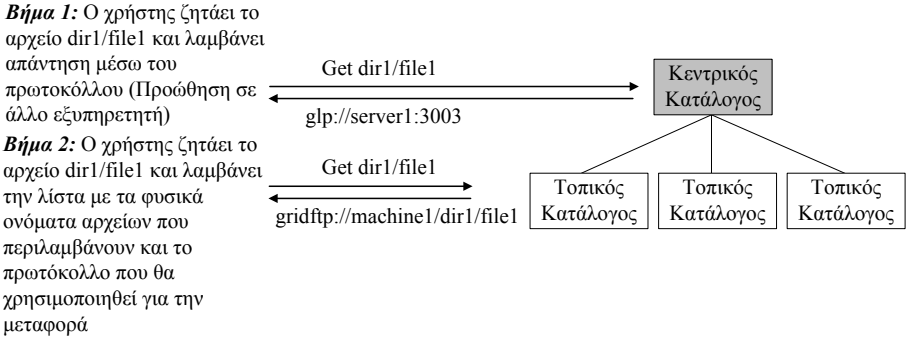
Επίσης, η υπηρεσία αναζήτησης αντιγράφων παρέχει την δυνατότητα να διαχειριστεί μεταδεδομένα για κάθε λογικό όνομα αρχείου, στη μορφή ζευγαριών παραμέτρου-τιμής (attribute-value pairs). Για τον άμεσο χειρισμό και διαμόρφωση των συλλογών αρχείων, τα λογικά ονόματα και τις διάφορες παραμέτρους τους, η υπηρεσία αναζήτησης αντιγράφων προσφέρει τις ανάλογες διεπαφές. Η υλοποίηση της υπηρεσίας αναζήτησης αντιγράφων αρχείων χρησιμοποιεί μια βάση LDAP για την αποθήκευση όλων των πληροφοριών. Μέσω της υπηρεσίας, δεδομένου του λογικού ονόματος ενός αρχείου, μπορεί να βρεθούν οι συλλογές που το περιέχουν και στην συνέχεια οι διάφορες τοποθεσίες που βρίσκεται αποθηκευμένο.

4.3 Γεωγραφική κατανομή σε τοπικούς καταλόγους

Η υπηρεσία αναζήτησης αντιγράφων για ένα κατανεμημένο σύστημα παγκόσμιας κλίμακας όπως το Πλέγμα Δεδομένων δεν μπορεί παρά να αντιμετωπίζει προβλήματα αξιοπιστίας και απόδοσης όταν αυτή βασίζεται σε ένα κεντρικό εξυπηρετητή και να αποτελεί στενωπό για ολόκληρο το σύστημα. Για το λόγο αυτό έγινε μια πρώτη προσπάθεια για την κατανομή του φορτίου σε περισσότερους από έναν εξυπηρετητή εισάγοντας μια υποτυπώδη ιεραρχία [Stockinger 02a]. Καταρχήν

σε κάθε κόμβο δημιουργήθηκε ένας τοπικός κατάλογος με λίστες λογικών ονομάτων αρχείων που είναι αποθηκευμένα τοπικά και την αντιστοίχηση του φυσικού ονόματός τους. Με τον τρόπο αυτό η κεντρική υπηρεσία αναζήτησης αντιγράφων δεν ήταν απαραίτητο να κρατάει λεπτομερή πληροφορία για κάθε λογικό όνομα αρχείου παρά μόνο την διεύθυνση του κόμβου, ο οποίος διατηρεί αυτή την πληροφορία για κάθε λογικό όνομα. Αυτό είχε σαν αποτέλεσμα μια διεπίπεδη ιεραρχία που κατένειμε τα δεδομένα και τις ερωτήσεις για αυτά. Στο πάνω επίπεδο βρίσκεται η κεντρική υπηρεσία που δεν καταργήθηκε, απλά πλέον ανακατεύθυνε τις ερωτήσεις στους κατάλληλους τοπικούς καταλόγους που βρίσκονται στο κάτω επίπεδο. Το φορτίο των ερωτημάτων αλλά και οι αντιστοιχήσεις λογικών και φυσικών ονομάτων κατανεμήθηκε στους τοπικούς καταλόγους και η κεντρική υπηρεσία απλά ενημερωνόταν μόνο για προσθήκες και διαγραφές λογικών ονομάτων.

Το παγκοσμίου κλίμακας κατανεμημένο σχήμα μπορεί να θεωρηθεί σαν ένα βήμα προς την φιλοσοφία του Πλέγματος, καθώς προσφέρει αυτονομία για κάθε κόμβο. Οι εργασίες που εκτελούνται σε κάθε κόμβο μπορούν να έχουν πρόσβαση στα τοπικά αντίγραφα, χωρίς να επικοινωνούν με την κεντρική υπηρεσία, καθώς η πληροφορία αυτή υπάρχει στους τοπικούς καταλόγους. Επιπλέον, με την εφαρμογή ενός απλού σχήματος caching στον τοπικό κατάλογο για λογικά ονόματα που πρέπει να ερωτηθεί η κεντρική υπηρεσία αυξάνεται ικανοποιητικά η απόδοση του συστήματος. Εκτός του προφανούς πλεονεκτήματος της ανοχής σε σφάλματα, υπάρχει και το πλεονέκτημα της εφαρμογής διαφορετικής πολιτικής πρόσβασης ανά κόμβο. Έτσι οι διαχειριστές κάθε κόμβο μπορούν να ορίζουν ποια αρχεία μπορούν να ανακοινωθούν και να αντιγραφούν στο υπόλοιπο Πλέγμα, χωρίς να χρειαστεί να μεσολαβήσουν με την κεντρική υπηρεσία αναζήτησης αντιγράφων.



Σχήμα 4.2: *Παράδειγμα της κατανομής σε τοπικούς καταλόγους ανά κόμβο με την διατήρηση της κεντρικής υπηρεσίας. Δύο ερωτήματα είναι απαραίτητα για την εύρεση της λίστας με τα φυσικά ονόματα ενός λογικού ονόματος.*

Το πλεονέκτημα της προτεινόμενης λύσης είναι η συμβατότητα που προσφέρει με τις προηγούμενες προσεγγίσεις. Οι κατάλογοι είναι και πάλι βασισμένοι σε βάσεις δεδομένων τύπου LDAP και η ροή των δεδομένων για τα ερωτήματα αναζήτησης αντιγράφων παραμένουν ίδια με την προσθήκη ενός ενδιαμέσου βήματος προώθησης. Οι ανακατευθύνσεις των ερωτημάτων από την

κεντρική υπηρεσία στους τοπικούς καταλόγους υλοποιήθηκαν με την χρησιμοποίηση ενός νέου πρωτοκόλλου, του Grid Lookup Protocol (GLP). Με το πρωτόκολλο αυτό, ένα ερώτημα στην κεντρική υπηρεσία επιστρέφει ένα φυσικό όνομα αρχείου με το πρόθεμα glp:// αντί για το σύνηθες gridftp:// . Το νέο πρόθεμα το αντιλαμβάνεται το λογισμικό που κάνει το ερώτημα και καταλαβαίνει ότι η διεύθυνση που ακολουθεί είναι η διεύθυνση του καταλόγου που περιέχει τις ζητούμενες πληροφορίες, οπότε και επαναλαμβάνει το ερώτημα προς τον τοπικό αυτό κατάλογο και λαμβάνει τις φυσικές τοποθεσίες του αρχείου που επιθυμεί. Τέλος, το πρωτόκολλο GLP είχε υποστήριξη για τα αναγνωριστικά τύπου Universally Unique Identifiers (UUID), τα οποία όπως και τα πρωτεύοντα κλειδιά σε μια σχεσιακή βάση δεδομένων αναγνωρίζουν μοναδικά ένα αρχείο. Οπότε, πλέον τα λογικά ονόματα αρχείων μπορούσαν να επαναχρησιμοποιηθούν ή να αλλαχθούν χωρίς να απαιτούνται αλλαγές σε όλες τις υφιστάμενες αντιστοιχίσεις σε φυσικά ονόματα αρχείων.

4.4 Παραμετρικά σχήματα κατανομής - Giggle

Η προηγούμενη μέθοδος της κατανομής του καταλόγου σε τοπικές βάσεις δεδομένων ανά κόμβο λύνει μερικά από τα προβλήματα της μοναδικής κεντρικής υπηρεσίας, αλλά και πάλι στηρίζεται σε ένα κεντρικό κατάλογο, όσον αφορά πληροφορίες για λογικά ονόματα που βρίσκονται σε απομακρυσμένους κόμβους. Για το λόγο αυτό προτάθηκε το πλαίσιο Giggle (GIGa-scale Global Location Engine) [Chervenak 02, Chervenak 04] όπου ορίζεται μια πολύ-επίπεδη ιεραρχία για την υπηρεσία αναζήτησης αντιγράφων αρχείων με διάφορους γενικούς και τοπικούς καταλόγους συνδεδεμένους μεταξύ τους. Το Giggle εισαγάγει δύο νέες έννοιες σαν συστατικά της υπηρεσίας αναζήτησης αντιγράφων:

- Τοπικοί κατάλογοι αντιγράφων – local replica catalogs (LRC), οι οποίοι διατηρούν πληροφορίες για τα λογικά ονόματα αρχείων, όπως λίστες πρόσβασης, ημερομηνίες δημιουργίας κλπ. Επιπλέον, διατηρεί μια λίστα με όλα τα φυσικά ονόματα των αντιγράφων για κάθε λογικό όνομα. Δεδομένου ενός λογικού ονόματος από τον τοπικό κατάλογο μπορούμε να μάθουμε που βρίσκονται όλα τα αντίγραφα αυτού.

- Ευρετήρια αναζήτησης αντιγράφων – replica location indices (RLI), τα οποία διατηρούν πληροφορία σχετικά με το υπεύθυνο τοπικό κατάλογο ανά λογικό όνομα αρχείου. Δεδομένου ενός λογικού ονόματος από το ευρετήριο μπορούμε να μάθουμε που βρίσκονται οι πληροφορίες για αυτό το λογικό όνομα.

Επιπλέον, με την αρχιτεκτονική αυτή προσφέρεται η δυνατότητα διαφορετικού επιπέδου ασφάλειας και κατανομής του καταλόγου. Σε ένα κοινό σενάριο, όπου κάθε κόμβος ή εικονικός οργανισμός διαχειρίζεται ένα κατάλογο και τον γενικό συντονισμό τον έχει μια μοναδική οντότητα

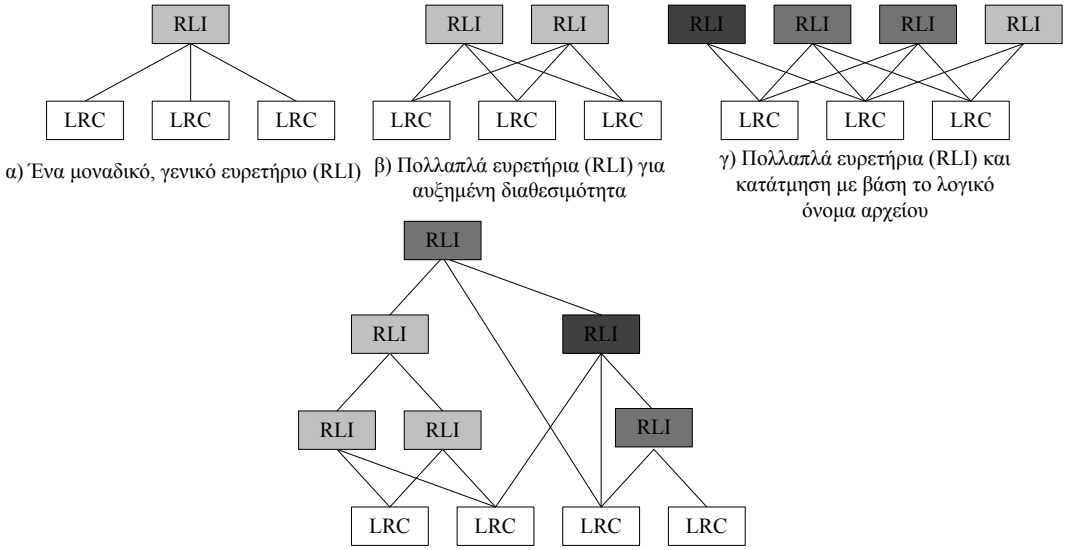
ευρετηρίου, τότε η δομή είναι όμοια με την προηγούμενη γεωγραφική κατανομή σε τοπικούς καταλόγους. Τα πλεονεκτήματα του Giggle γίνονται εμφανή όταν οι προδιαγραφές της υπηρεσίας γίνουν πιο αυστηρές. Πολλαπλά ευρετήρια μπορούν να αναπτυχθούν σε παραλληλία και να προσφέρουν ισοκατανομή του φορτίου των ερωτημάτων αλλά και την απαλοιφή του κεντρικού σημείο βλάβης στην υπηρεσία αναζήτησης αντιγράφων. Στην περίπτωση αυτή τα ευρετήρια και οι κατάλογοι διαμορφώνουν μια διεπίπεδη αρχιτεκτονική, όπου κάθε κατάλογος συνδέεται με πολλαπλά ευρετήρια και αντιστρόφως, ενώ υπάρχει και η επιλογή της δενδρικής δομής στην περίπτωση των πολλαπλών ευρετηρίων.

Το Giggle χρησιμοποιεί μια σχεσιακή βάση δεδομένων αντί για έναν κατάλογο LDAP, το οποίο επιτρέπει μεγαλύτερη ευελιξία για τον ορισμό του σχήματος της αποθηκευμένης πληροφορίας. Επιπλέον, η δομή της ιεραρχίας που διαμορφώνουν οι κατάλογοι και τα ευρετήρια μπορούν να ελεγχθούν από τον ορισμό διάφορων παραμέτρων ανάπτυξης. Για παράδειγμα μια παράμετρος ελέγχει τον αριθμό των ευρετηρίων, καθώς μπορεί να υπάρχουν ένα ή περισσότερα, ειδικά σε ένα σχήμα υψηλής διαθεσιμότητας, όπου ο αριθμός αυτός μπορεί να ξεπερνά και το πλήθος των καταλόγων. Σε αυτή την περίπτωση, ίσως είναι αναγκαίο να οργανωθούν τα ευρετήρια σε μια δενδρική ιεραρχία και ένα γενικό σχήμα κατάτμησης, για να αποφευχθούν τα μειονεκτήματα ενός απλοποιημένου μοντέλου. Τα δεδομένα μπορούν να κατανεμηθούν στα ευρετήρια με βάση την παράμετρο που ορίζει την επιθυμητή συνάρτηση κατάτμησης για τα λογικά ονόματα. Η πολιτική αυτή μπορεί να είναι μια από τις εξής:

- Όλα τα λογικά ονόματα θα αποθηκεύονται σε όλα τα ευρετήρια.
- Κατάτμηση με βάση κάποια μαθηματική συνάρτηση. Αυτό θα ισορροπήσει το φόρτο τους συστήματος, αλλά δεν δίνει εγγυήσεις για την τοπικότητα των δεδομένων.
- Κατάτμηση με βάση κάποια λογική παράμετρο, όπως η συλλογή που ανήκει το λογικό όνομα αρχείου. Αυτό θα εγγυηθεί την τοπικότητα των δεδομένων, αλλά θα δημιουργήσει ανισοκατανομή στο φόρτο του συστήματος.

Επιπλέον, το Giggle μπορεί να εκμεταλλευτεί την γεωγραφική θέση των γενικών ευρετηρίων με την κατάτμηση των λιστών λογικών – φυσικών ονομάτων με βάση τις διευθύνσεις DNS των τοπικών καταλόγων. Μια ακόμη σημαντική παράμετρος είναι ο βαθμός της διαθεσιμότητας στο επίπεδο των ευρετηρίων, καθώς μπορεί αυτή να κυμαίνεται από μηδενική ανάνηψη από βλάβη μέχρι πλήρη στοιχεία όλων των λογικών ονομάτων σε όλα τα ευρετήρια. Η ενδιάμεση κατάσταση συνεπάγεται στην ύπαρξη κοινών δεδομένων σε μικρότερα υποσύνολα των ευρετηρίων.

Το Giggle χρησιμοποιεί ένα πολύπλοκο σχήμα ενημέρωσης των αλλαγών μεταξύ των τοπικών καταλόγων και των ευρετηρίων. Οι κατάλογοι χρησιμοποιούν πρωτόκολλα αλλαγών τύπου soft-state για την εγγραφή νέων αρχείων ή την ανανέωση της παραμέτρου που ορίζει τον χρόνο ζωής των ήδη εγγεγραμμένων αρχείων στο ευρετήριο. Οι πλήρεις ενημερώσεις είναι αρκετά απαιτητικές



Σχήμα 4.3: *Διάφορα πιθανά σενάρια ανάπτυξης του Giggle. Διαφορετικά χρώματα ευρετηρίων αναπαριστούν ευρετήρια που διατηρούν διαφορετικά τμήματα του χώρου των λογικών ονομάτων*

σε ότι αφορά την δικτυακή χωρητικότητα, οπότε υπάρχει η επιλογή για περιοδικές ενημερώσεις μόνο των αλλαγών. Επιπλέον, η αποδοτικότητα αυτής της μεθόδου αυξάνεται αν η πληροφορία που ανταλλάσσεται είναι συμπιεσμένη. Η προτεινόμενη μέθοδος συμπίεσης είναι βασισμένη στα Bloom filters. Ο τύπος και η συχνότητα των αλλαγών που στέλνονται από τους καταλόγους στα ευρετήρια, καθώς και το επιθυμητό σχήμα συμπίεσης είναι από τις παραμέτρους που μπορεί κανείς να ορίσει κατά την ανάπτυξη του συστήματος Giggle.

Παρόλο που η επεκτασιμότητα σε δίκτυα παγκόσμιας κλίμακας ήταν από τις κύριες προδιαγραφές του Giggle κατά την φάση της σχεδίασης, παραμένει ανοιχτό ζήτημα το κατά πόσον η συγκεκριμένη αρχιτεκτονική θα συνεχίσει να λειτουργεί το ίδιο αποδοτικά όταν το πλήθος των αποθηκευμένων δεδομένων, των καταλόγων και των ευρετηρίων αυξηθούν κατά μερικές τάξεις μεγέθους. Το Giggle έχει σχεδιαστεί και δοκιμαστεί σε περιβάλλοντα πλέγματος υψηλής επίδοσης και δεν έχει δοκιμαστεί σε συνθήκες υποδομών παγκόσμιας κλίμακας όπου δεν υπάρχουν πόροι με χαρακτηριστικά υψηλής διαθεσιμότητας. Επίσης, οι πολύπλοκοι μηχανισμοί διάδοσης των αλλαγών που υλοποιεί το Giggle, θα αποτελέσουν πλήγμα για την αποδοτικότητα της υπηρεσίας αναζήτησης αντιγράφων αρχείων σε συνθήκες δικτύων μεγάλης κλίμακας, όπου υπάρχει αυξημένη καθυστέρηση και χαμηλότεροι ρυθμοί μετάδοσης δεδομένων. Ενώ δεν υπάρχουν δημοσιευμένα πειράματα για το Giggle σε συνθήκες παραγωγικής υπηρεσίας παγκόσμιας κλίμακας όπου υπάρχουν αποθηκευμένα εκατομμύρια και δισεκατομμύρια δεδομένα λογικών/φυσικών αρχείων και κάτι τέτοιο είναι πρακτικά αδύνατο να δοκιμαστεί, ώστε να βγουν χρήσιμα συμπεράσματα

απόδοσης, οι σχεδιαστικές επιλογές που ακολουθήθηκαν θέτουν όρια στην κλιμακωσιμότητά του και κατ' επέκταση στην επίδοση του.

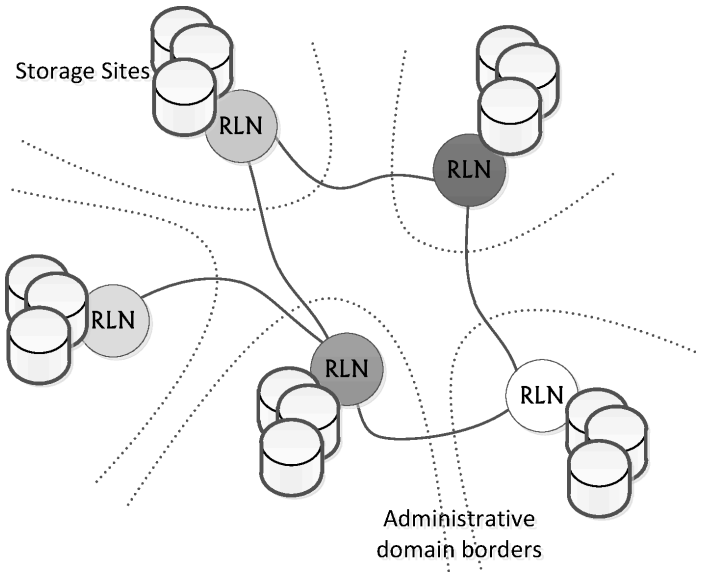
4.5 Κατανεμημένη υπηρεσία διαχείρισης αντιγράφων βασισμένη σε αδόμητο δίκτυο

Στην εργασία [Ripeanu 02b], αναλύεται μια διαφορετική προσέγγιση υπηρεσίας διαχείρισης αντιγράφων χρησιμοποιώντας ένα αδόμητο δίκτυο ομότιμων κόμβων. Οι έννοιες του λογικού και του φυσικού ονόματος αρχείου διατηρούνται, ενώ προστίθεται νέα ορολογία όπως ο κόμβος διαχείρισης αντιγράφων - ΚΔΑ (Replica Location Node - RLN) και ο κόμβος αποθήκευσης - ΚΑ (Storage Site - SS). Κάθε ΚΔΑ δέχεται λίστες με λογικά και φυσικά ονόματα από τους διάφορους ΚΑ με τα αποθηκευμένα τους αρχεία. Κάθε ΚΔΑ επίσης δέχεται ερωτήσεις για λογικά ονόματα αρχείων, τα οποία γνωρίζει λόγω της ενημέρωσης από τα διάφορα ΚΑ, ενώ επιτρέπονται και οι απομακρυσμένες ερωτήσεις σε άλλους ΚΔΑ. Με το τρόπο αυτό, οι ΚΔΑ οργανώνονται σε ένα αδόμητο δίκτυο. Οι πληροφορίες ανταλλάσσονται μεταξύ τους σε λίστες αντιστοίχισης ονομάτων, οι οποίες είναι σε μια μορφή συμπιεσμένη χρησιμοποιώντας την τεχνική των bloom filters [Bloom 70, Mitzenmacher 02].

Η οργάνωση της υπηρεσίας φαίνεται στο σχήμα 4.4. Οι ΚΑ δημοσιεύουν στους ΚΔΑ τις λίστες αντιστοίχησης λογικών-φυσικών ονομάτων για τα αρχεία που έχουν τοπικά. Οι ΚΔΑ αποθηκεύουν τις λίστες και τις επεξεργάζονται δημιουργώντας Bloom filters. Στην συνέχεια οι ΚΔΑ οργανώνονται σε ένα αδόμητο δίκτυο και ανταλλάσσουν μεταξύ τους τις λίστες σε μορφή bloom filters.

Τα bloom filters είναι δομές που επιτρέπουν την πιθανολογική αναπαράσταση ενός συνόλου στοιχείων, ώστε να υποστηρίζονται οι ερωτήσεις αν κάποιο στοιχείο ανήκει στο σύνολο αυτό. Το κόστος αυτής της δομής είναι ένα μικρό ποσοστό λανθασμένων θετικών απαντήσεων (false positives). Το μεγάλο κέρδος βρίσκεται στα μεγέθη συμπίεσης, καθώς ένα σύνολο N στοιχείων μπορεί να αναπαρασταθεί με 2N bytes, με ποσοστό λανθασμένων θετικών απαντήσεων κάτω από 0,1% και χρόνους απόκρισης περίπου 100μs. Τα φίλτρα διαδίδονται από ένα ΚΔΑ στους υπόλοιπους με τεχνικές multicast, ενώ περιοδικά στέλνονται και μεμονωμένες ενημερώσεις για την ελαχιστοποίηση της επιβάρυνσης του δικτύου. Μάλιστα τα bloom filters είναι αρκετά ευμετάβλητα, ώστε σε περίπτωση εξάντλησης πόρων όπως η μνήμη του κόμβου ή χωρητικότητα του δικτύου, ο αρχικός ή ένας ενδιάμεσος ΚΔΑ να ελαχιστοποιεί το μέγεθος του bloom filter, με κόστος βέβαια την αύξηση του ποσοστού λανθασμένων θετικών απαντήσεων. Το πρόβλημα με την προσέγγιση αυτή είναι ότι τα δεδομένα βρίσκονται σε ένα και μόνο ΚΑ και η κατανεμημένη διαδικασία σταματά την στιγμή που ανακαλύπτεται ο υπεύθυνος ΚΑ για ένα λογικό αρχείο. Η αδόμητη φύση του δικτύου και ο μηχανισμός ενημέρωσης με τα στοιχεία των κόμβων οδηγεί με τον χρόνο σε μια

κατάσταση όπου όλοι οι ΚΔΑ θα γνωρίζουν όλα τα στοιχεία των ΚΑ, χάνοντας έτσι τις ιδιότητες ενός δικτύου ομότιμων κόμβων όπως αυτή της κλιμακωσιμότητας. Τέλος, δεν προβλέπεται μηχανισμός αλλαγής της πληροφορίας, αλλά ο υπεύθυνος κόμβος για ένα λογικό όνομα είναι πάντα ο ίδιος, το οποίο αποτελεί ένα μεγάλο πρόβλημα για ένα δίκτυο παγκόσμιας κλίμακας όπως το πλέγμα.



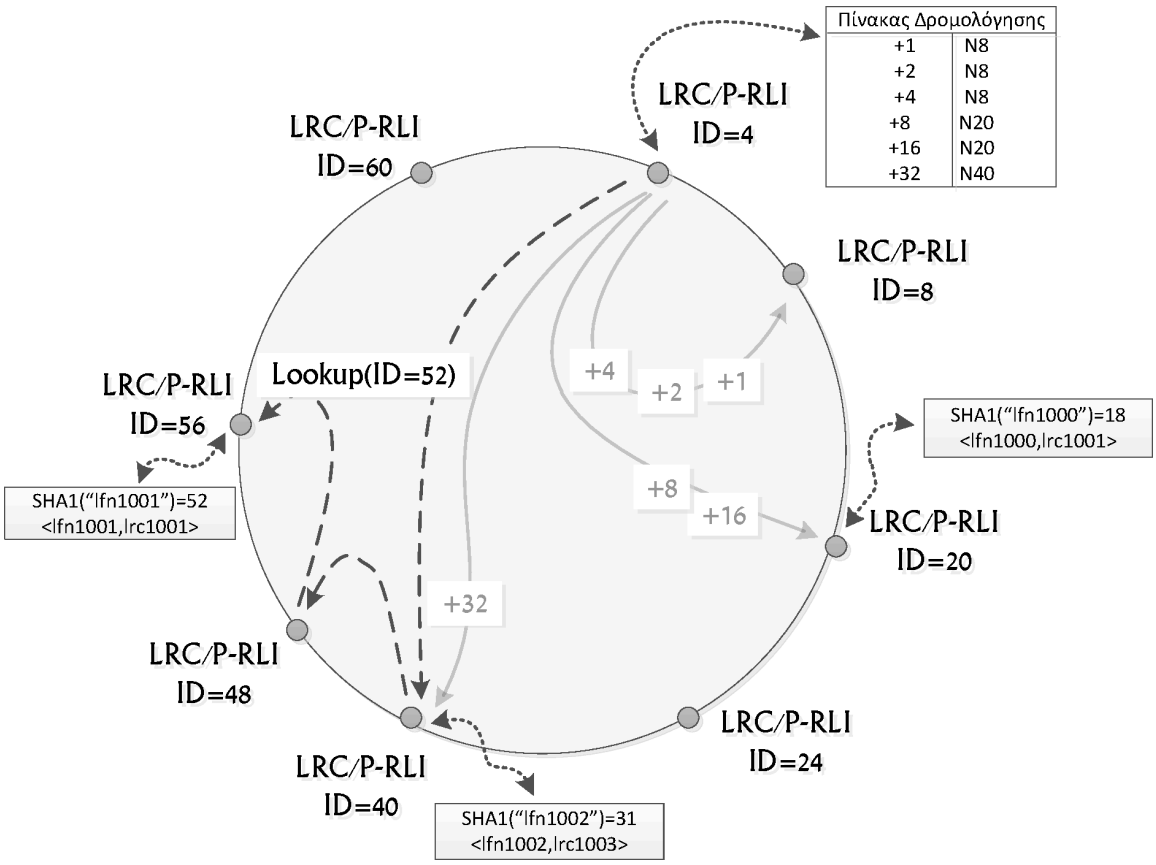
Σχήμα 4.4: *Οργάνωση της υπηρεσίας διαχείρισης αρχείων με τους κόμβους αποθήκευσης και τους κόμβους διαχείρισης αντιγράφων*

4.6 P-RLS - Κατανεμημένη υπηρεσία διαχείρισης αντιγράφων βασισμένη σε δομημένο δίκτυο

Στην εργασία [Cai 04], παρουσιάζεται η αρχιτεκτονική του P-RLS, μιας υπηρεσία αναζήτησης αντιγράφων βασισμένη στο Giggle, το οποίο και επεκτείνει με ένα δομημένο δίκτυο ομότιμων κόμβων όπως το Chord. Πιο συγκεκριμένα στο P-RLS έχει αντικατασταθεί η ιεραρχική δομή των ευρετηρίων με ένα δίκτυο Chord. Οι τοπικοί κατάλογοι αντιγράφων (LRC) έχουν παραμείνει με την ίδια λειτουργικότητα. Κάθε τέτοιος κατάλογος έχει ένα κόμβο, ο οποίος ονομάζεται P-RLI κόμβος και συμμετέχει στο δίκτυο Chord με ένα μοναδικό αναγνωριστικό μεγέθους m-bit.

Ένα τυπικό σενάριο χρήσης ξεκινά με τον χρήστη να καταχωρεί ή να διαγράφει αντιστοιχήσεις φυσικών αρχείων σε λογικά ονόματα σε ένα τοπικό κατάλογο. Οι κατάλογοι αυτοί στέλνουν

περιοδικά ενημερώσεις σχετικά με την κατάστασή τους στο P-RLS δίκτυο. Οι ενημερώσεις αυτές περιέχουν λίστες με λογικά ονόματα αρχείων και τοπικών καταλόγων οι οποίοι είναι επιφορτισμένοι με την αποθήκευση της σχετικής πληροφορίας για τα αντίγραφά τους *{logical name, LRC}*. Για κάθε λογικό όνομα, ο αντίστοιχος P-RLI κόμβος παράγει το m-bit αναγνωριστικό με βάση τη συνάρτηση SHA1 και αναζητεί τον successor κόμβο σύμφωνα με το πρωτόκολλο του Chord, ώστε να αποθηκεύσει εκεί την διεύθυνση του υπεύθυνου τοπικού καταλόγου. Ο κόμβος αυτός ονομάζεται και κόμβος-ρίζα για το συγκεκριμένο λογικό όνομα. Στο σχήμα 4.5 φαίνεται ένα στιγμιότυπο ενός τέτοιου δικτύου, με 8 κόμβους και 3 λογικά ονόματα αρχείων. Αντίστοιχα για να βρεθούν τα αντίγραφα που αντιστοιχούν σε ένα λογικό όνομα αρχείου, γίνεται μια αναζήτηση από οποιονδήποτε κόμβο στο P-RLS - πρώτα τοπικά και μετά στο δίκτυο - με βάση το αναγνωριστικό που παράγεται από το λογικό όνομα αρχείου. Η απάντηση θα επιστρέψει την διεύθυνση του υπεύθυνου καταλόγου, ο οποίος θα ερωτηθεί σε δεύτερη φάση και θα αντληθούν έτσι όλες οι διευθύνσεις για τις φυσικές τοποθεσίες των αντιγράφων.



Σχήμα 4.5: *Στιγμιότυπο ενός δικτύου με 8 κόμβους και 3 λογικά ονόματα αρχείων*

Ένα δίκτυο Chord μπορεί να ανεχτεί τυχαίες αναχωρήσεις και αφίξεις κόμβων στο δίκτυο, αλλά αυτό δεν εγγυάται ταυτόχρονα και την διατήρηση της αξιοπιστίας των δεδομένων της κάθε εφαρμογής, στην προκειμένη περίπτωση της υπηρεσίας διαχείρισης αντιγράφων. Στο P-RLS το συγκεκριμένο πρόβλημα αντιμετωπίζεται με την αναπαραγωγή της πληροφορίας - δηλαδή των αντιστοιχίσεων λογικών ονομάτων και τοπικών καταλόγων - στο σύνολο των successor κόμβων του κάθε κόμβου-ρίζας. Επίσης, για να διασφαλιστεί ότι τα δεδομένα αυτά δεν είναι ξεπερασμένα, υλοποιείται ένα πρωτόκολλο soft-state, όπου κάθε τιμή έχει μια ημερομηνία λήξης, με το πέρας της οποίας σβήνεται. Έτσι, κάθε κατάλογος θα πρέπει περιοδικά να ενημερώνει το P-RLS δίκτυο, ώστε να διατηρείται η πληροφορία του. Το πρόβλημα με τα παραπάνω είναι ότι οι προτεινόμενες αλλαγές, είναι εξωτερικές ως προς το πρωτόκολλο του Chord, οπότε θα πρέπει να εξεταστεί κατά πόσο μπορούν να διατηρηθούν τα χαρακτηριστικά του, ειδικά σε συνθήκες υψηλού φορτίου, όπου θα υπάρχει μεγάλη ανταλλαγή μηνυμάτων για την σταθεροποίηση του δικτύου και την αποθήκευση των ζευγαριών κλειδιού-τιμής στον σωστό κόμβο.

4.7 Κατανεμημένη διαχείριση δεδομένων με δίκτυα ομότιμων κόμβων

Τα δομημένα συστήματα ομότιμων κόμβων είναι σχεδιασμένα για να εξυπηρετούν αιτήσεις αποθήκευσης και αναζήτησης ζευγαριών κλειδιού-τιμής (key-value pairs). Τα κλειδιά είναι πάντα μοναδικά σε όλο το σύστημα και αποτελούν το μοναδικό αναγνωριστικό για την τιμή του ζευγαριού. Οι περισσότερες υλοποιήσεις κατανεμημένων πινάκων κατακερματισμού παράγουν κλειδιά χρησιμοποιώντας την τιμή του ζευγαριού, βάση κάποιας συνάρτησης κρυπτογράφησης όπως η SHA1 πάνω στα δεδομένα που αποθηκεύουν. Αυτή η μέθοδος παράγει ομοιόμορφες κατανομές κλειδιών σε ένα χώρο αναγνωριστικών μεγέθους 160bit. Ως αποτέλεσμα του παραπάνω, για να χρησιμοποιηθούν οι κατανεμημένοι πίνακες κατακερματισμού για την αναζήτηση αντιγράφων δεδομένων σε ένα περιβάλλον υπολογιστικού πλέγματος, πρέπει να κάνουμε τις κάτωθι υποθέσεις:

- Ένα ανεξάρτητο δίκτυο ομότιμων κόμβων θα χρησιμοποιείται για κάθε εικονικό οργανισμό

- Το κλειδί δεν θα παράγεται βάσει των δεδομένων/της τιμής του ζευγαριού του, αλλά βάσει του λογικού ονόματος αρχείου (LFN) που αντιπροσωπεύει. Θα είναι ένα μοναδικό αναγνωριστικό που θα χρησιμοποιείται σε όλες τις σχετικές υπηρεσίες και διαδικασίες.

- Η τιμή που θα αντιστοιχεί σε κάθε κλειδί και συνδυαστικά θα αποτελούν το ζευγάρι κλειδιού-τιμής, θα είναι μια λίστα που θα περιέχει όλες τις φυσικές τοποθεσίες των αντιγράφων δεδομένων (PFNs) για ένα λογικό όνομα αρχείου (LFN).

4.7.1 Προβλήματα

Το κυρίως πρόβλημα στην χρησιμοποίηση των ενός δικτύου ομότιμων κόμβων με κατανεμημένο πίνακα κατακερματισμού για την αποθήκευση των τοποθεσιών των αντιγράφων αρχείων, είναι η αδυναμία του να χειριστεί δεδομένα που αλλάζουν. Τα δίκτυα αυτά προσφέρουν συναρτήσεις αποθήκευσης και ανάκτησης ζευγαριών κλειδιού-τιμής, αλλά δεν προσφέρουν συναρτήσεις μεταβολής των ζευγαριών αυτών. Όταν ένα ζευγάρι κλειδιού-τιμής αποθηκευτεί στο δίκτυο, τότε θα μείνει αποθηκευμένο μέχρι την ημερομηνία λήξης του, η οποία μπορεί συνεχώς να ανανεώνεται. Αυτό το σχεδιαστικό μειονέκτημα εμφανίζεται σαν το αντάλλαγμα που πρέπει να πληρώσει κανείς για να έχει τα πλεονεκτήματα της επεκτασιμότητας και της διαθεσιμότητας. Όσο περισσότερο γίνονται αυτά τα συστήματα ανθεκτικά σε αφίξεις και αναχωρήσεις νέων κόμβων στο δίκτυο, τόσο πιο δύσκολο γίνεται να εντοπιστεί ο υπεύθυνος κόμβος για ένα ζευγάρι κλειδιού-τιμής. Αυτό είναι αναπόφευκτο. Σε ένα στατικό δίκτυο δεν θα υπήρχε καμία ανάγκη να υπάρχουν διπλά και προσωρινά δεδομένα. Τα ζευγάρια κλειδιού-τιμής θα είχαν συγκεκριμένες τοποθεσίες για την αποθήκευσή τους. Στα δίκτυα ομότιμων κόμβων με κατανεμημένο πίνακα κατακερματισμού κάθε κόμβος έχει και αυτός ένα μοναδικό αναγνωριστικό που παίρνει τυχαίες τιμές από τον ίδιο χώρο με αυτόν των ζευγαριών κλειδιού-τιμής. Οπότε τα ζευγάρια αντιγράφονται σε ένα αριθμό κόμβων που είναι κοντύτερα στο κλειδί τους και όχι σε ένα μόνο κόμβο. Δεν υπάρχει αλγόριθμος που επιστρέφει την ακριβή τοποθεσία ενός ζευγαριού κλειδιού-τιμής σε μια δεδομένη στιγμή, το οποίο είναι και προαπαιτούμενο για την ασφάλεια των δικτύων αυτών [Hazel 02].

Τα δίκτυα ομότιμων κόμβων με κατανεμημένο πίνακα κατακερματισμού δημιουργούνται δυναμικά και αποθηκεύουν μη μεταβλητά δεδομένα. Αυτό είναι κάτι αρκετό για την αποθήκευση δεδομένων ενός συστήματος διανομής αρχείων μόνο για ανάγνωση, αλλά δεν είναι αρκετό για να εξυπηρετήσει την υπηρεσία αναζήτησης αντιγράφων του υπολογιστικού πλέγματος. Η συνάρτηση ενημέρωσης των δεδομένων είναι απολύτως αναγκαία για την αποθήκευση των διάφορων τοποθεσιών των αντιγράφων, καθώς τα φυσικά ονόματα για κάποιο λογικό όνομα αρχείου μπορεί να αλλάζουν αρκετά συχνά και θα πρέπει να υπάρχει ένας μηχανισμός για την διάδοση των αλλαγών σε όλο το δίκτυο των ομότιμων κόμβων το συντομότερο δυνατό.

4.7.2 Προτεινόμενη λύση με Kademlia+

Μια λύση στα παραπάνω προβλήματα είναι η υιοθέτηση ημερομηνιών λήξης σε κάθε ζευγάρι κλειδιού-τιμής για την αλλαγή των τιμών στο δίκτυο. Τα δεδομένα σε ένα δίκτυο ομότιμων κόμβων με κατανεμημένο πίνακα κατακερματισμού λήγουν κάθε ένα, σε συγκεκριμένο διάστημα από την αρχική δημοσίευσή τους και η ανανέωση ή η οριστική διαγραφή τους επαφίεται σε εξωτερικά συστήματα εκτός των μηχανισμών του δικτύου ομότιμων κόμβων. Όμως αυτός ο τρόπος για την υποστήριξη μεταβλητών δεδομένων δεν είναι λύση. Η χρησιμοποίηση σύντομων ημερομηνιών λήξης και η προσθήκη ενός εξωτερικού συστήματος διαχείρισης των ημερομηνιών αυτών

θα δημιουργούσε προβλήματα επεκτασιμότητας, θα κατέστρεφε το μηχανισμό των προσωρινών αντιγράφων του δικτύου (cache) και θα επέφερε βαρύ φορτίο στο δίκτυο από συχνές αλλαγές δεδομένων. Επίσης, η ανά τακτά χρονικά διαστήματα ενημέρωση των κόμβων δεν εγγυάται την γρήγορη και άμεση διάδοση των αλλαγών στο δίκτυο. Οι αναζητήσεις θα επέστρεφαν αποτελέσματα χωρίς αντίκρισμα.

Η ιδανική λύση είναι η ενεργοποίηση μεταβλητών δεδομένων στο επίπεδο κάθε ανεξάρτητου ζευγαριού κλειδιού-τιμής, το οποίο είναι αποθηκευμένο στο δίκτυο. Αυτό θα μπορούσε να πραγματοποιηθεί με την προσθήκη μερικών αλλαγών στο βασικό πρωτόκολλο του κατανεμημένου πίνακα κατακερματισμού. Οι πίνακες αυτοί μπορούν να κατανέμουν δεδομένα σε πολλούς κόμβους του δικτύου, αλλά μόνο ένα υποσύνολο από αυτούς, οι πιο σημαντικοί, απαντούν σε ερωτήματα αναζήτησης για κάθε ζευγάρι κλειδιού-τιμής. Εάν αλλάξουμε την τιμή σε αυτούς τους κόμβους τότε με μεγάλη πιθανότητα, σε διαδοχικές αναζητήσεις για αυτό το κλειδί, τουλάχιστον ένας από τους ενημερωμένους κόμβους θα βρεθεί. Φυσικά, αυτό δεν είναι αρκετό καθώς το δίκτυο δεν είναι στατικό και οι υπεύθυνοι κόμβοι για κάθε ζευγάρι κλειδιού-τιμής αλλάζουν με την πάροδο του χρόνου. Τα δίκτυα ομότιμων κόμβων με κατανεμημένους πίνακες κατακερματισμού υποστηρίζουν δυναμικές αφίξεις και αναχωρήσεις κόμβων, οπότε οι σχέσεις μεταξύ υπεύθυνων κόμβων και ζευγαριών κλειδιού-τιμής μπορούν να αλλάξουν με απρόβλεπτο τρόπο.

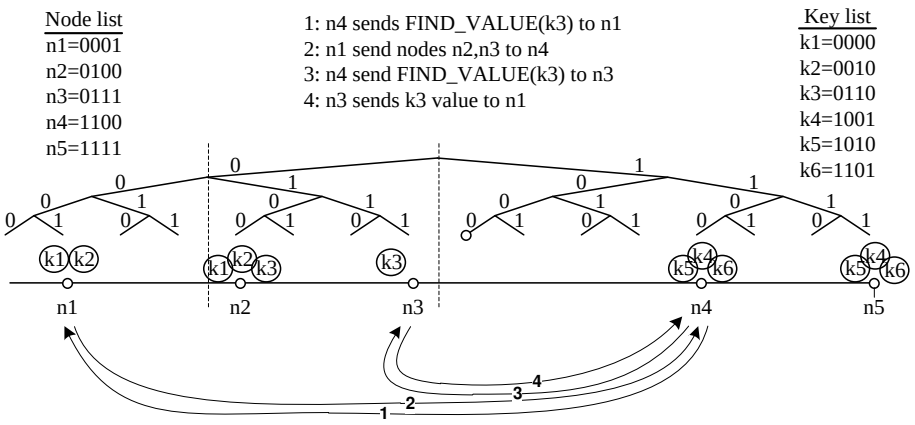
Αποτέλεσμα της ως άνω ιδιότητας είναι η διαδικασία της αναζήτησης να απευθύνεται σε όλους τους κόμβους που είναι υπεύθυνοι για ένα ζευγάρι κλειδιού-τιμής, συγκρίνοντας τα αποτελέσματα βάσει κάποιας προσυμφωνημένης μεθόδου ελέγχου διαφορετικών εκδόσεων. Σε περίπτωση που όλοι οι κόμβοι επιστρέφουν την ίδια έκδοση, τότε το σύστημα είναι ενημερωμένο. Σε περίπτωση που τα αποτελέσματα περιέχουν διαφορετικές εκδόσεις τότε επιλέγεται η πιο νέα και διαδίδεται και στους υπόλοιπους υπεύθυνους κόμβους. Το βήμα της διάδοσης απαιτεί αλλαγή στους υπάρχοντες αλγόριθμους αναζήτησης δεδομένων του κατανεμημένου πίνακα κερματισμού, ώστε να μην σταματά στο πρώτο αποτέλεσμα, αλλά να συνεχίζει μέχρις ότου έχει όλα τα αποτελέσματα για να ελέγξει πιθανές διαφορετικές εκδόσεις ενός ζευγαριού κλειδιού-τιμής. Ο κόμβος που έχει ξεκινήσει τη διαδικασία αναζήτησης έχοντας όλες τις διαφορετικές εκδόσεις ενός ζευγαριού κλειδιού-τιμής στην κατοχή του, θα αποφασίσει ποια είναι η σωστή και θα στείλει μηνύματα αποθήκευσης του σωστού ζευγαριού στους μη ενημερωμένους κόμβους. Οπότε η διαδικασία ενημέρωσης είναι εφικτό να γίνει μέσα από την ήδη υπάρχουσα διαδικασία αναζήτησης όπου θα γίνεται έλεγχος διαφορετικών εκδόσεων και θα στέλνονται οι νεότερες εκδόσεις όπου διαπιστωθεί πρόβλημα. Οι υπεύθυνοι κόμβοι με τη σειρά τους κάθε φορά που θα έχουν μια αίτηση αποθήκευσης θα ελέγχουν την περίπτωση να υπάρχει ήδη ένα ζευγάρι με αυτό το κλειδί αποθηκευμένο, οπότε θα διατηρούν αυτό με την νεώτερη έκδοση. Η υλοποίηση της μεθόδου ελέγχου διαφορετικών εκδόσεων μπορεί απλά να εφαρμοστεί με την εισαγωγή ενός πεδίου ημερομηνίας-ώρας (timestamp) στη τιμή κάθε ζευγαριού κλειδιού-τιμής.

Υλοποίηση

Με το παραπάνω σχεδιασμό υπόψη, τροποποιήσαμε το πρωτόκολλο του Kademlia για να υποστηρίζει μεταβλητά δεδομένα [Chazapis 05]. Οι αλλαγές αυτές θα μπορούσαν να προσαρτηθούν σε οποιοδήποτε άλλο παρεμφερές δίκτυο, όμως επιλέχτηκε το Kademlia διότι διαθέτει ένα πολύ απλό πίνακα δρομολόγησης για την επικοινωνία των κόμβων μεταξύ τους και χρησιμοποιεί ένα συνεπή αλγόριθμο καθ' όλη τη διαδικασία αναζήτησης. Το Kademlia βασίζεται στην συνάρτηση XOR μεταξύ των 160bit αναγνωστικών του – των κλειδιών δηλαδή – για την εύρεση των υπεύθυνων κόμβων για κάθε ζευγάρι κλειδιού-τιμής. Όπως σε κάθε σύστημα κατανεμημένου πίνακα κατακερματισμού, οι κόμβοι και τα ζευγάρια κλειδιού-τιμής στο Kademlia μοιράζονται αναγνωριστικά από τον ίδιο χώρο διευθύνσεων. Το XOR χρησιμοποιείται σαν μια συνάρτηση εύρεσης της απόστασης μεταξύ δύο αναγνωριστικών, ενός κλειδιού και των υπεύθυνων κόμβων του. Όταν ένας κόμβος στο Kademlia αναζητήσει ένα κλειδί μέσα στο δίκτυο, θα ξεκινήσει α παράλληλα ερωτήματα προς τους κ πλησιέστερους κόμβους στην τιμή που γνωρίζει. Η διαδικασία αυτή θα συνεχιστεί όσο δεν επιστρέφεται καμιά τιμή ή μαθαίνει συνεχώς καινούργιους κόμβους πιο κοντινούς στο ζητούμενο κλειδί. Η παράμετρος κ ισχύει για όλο το δίκτυο και ορίζει τα αντίγραφα των ζευγαριών κλειδιού-τιμής που αποθηκεύονται στο δίκτυο, καθώς και το μέγεθος των πινάκων δρομολόγησης σε κάθε κόμβο.

Σύμφωνα με το πρωτόκολλο Kademlia, ορίζονται τρεις συναρτήσεις (Remote Procedure Calls – RPC), οι οποίες χρησιμοποιούνται σε κάθε διαδικασία αποθήκευσης ή αναζήτησης`FIND_NODE`, `FIND_VALUE` και `STORE`. Για να αποθηκευτεί ένα ζευγάρι κλειδιού-τιμής, ο κόμβος που ξεκινά τη διαδικασία πρέπει πρώτα να βρει τους πιο κοντινούς κόμβους στο κλειδί. Αρχίζοντας με τη λίστα των κοντινότερων κόμβων που διατηρεί στο τοπικό πίνακα δρομολόγησης, στέλνει παράλληλα ασύγχρονα μηνύματα τύπου `FIND_NODE` στους πρώτους α κόμβους της λίστας. Οι κόμβοι που δέχονται ένα μήνυμα τύπου `FIND_NODE` απαντούν με μια λίστα των κ πλησιέστερων κόμβων στο κλειδί που περιέχει το μήνυμα. Ο αρχικός κόμβος θα παραλάβει το μήνυμα, θα συγχωνεύσει τις δύο λίστες και θα ταξινομήσει τους κόμβους με βάση την απόστασή τους από το ζητούμενο κλειδί. Στην πραγματικότητα ο αρχικός κόμβος δεν χρειάζεται να περιμένει κάθε φορά για όλες τις απαντήσεις από τα α παράλληλα ερωτήματα, αλλά στέλνει συνέχεια ώστε κάθε δεδομένη χρονική στιγμή να υπάρχουν α ερωτήματα εν εξελίξει. Όταν η λίστα με τους κοντινότερους κόμβους έχει οριστικοποιηθεί, το ζευγάρι κλειδιού-τιμής αντιγράφεται στους κόμβους αυτούς με μηνύματα τύπου `STORE`. Το αρχικό πρωτόκολλο του Kademlia ορίζει ότι όλα τα ζευγάρια κλειδιού-τιμής επαναδημοσιεύονται κάθε ώρα και λήγουν 24 ώρες μετά την αρχική τους δημοσίευση.

Για την ανάκτηση μιας τιμής από το δίκτυο Kademlia, ένας κόμβος θα αρχικοποιήσει μια παρόμοια διαδικασία χρησιμοποιώντας μηνύματα τύπου `FIND_VALUE` αντί `FIND_NODE`. Ο κόμβος που δέχεται μήνυμα `FIND_VALUE` επιστρέφει είτε την τιμή που υπάρχει στο τοπικό αποθηκευτικό χώρο του για το ζητούμενο κλειδί, είτε μια λίστα με τους κ κοντινότερους κόμβους στο κλειδί αυτό

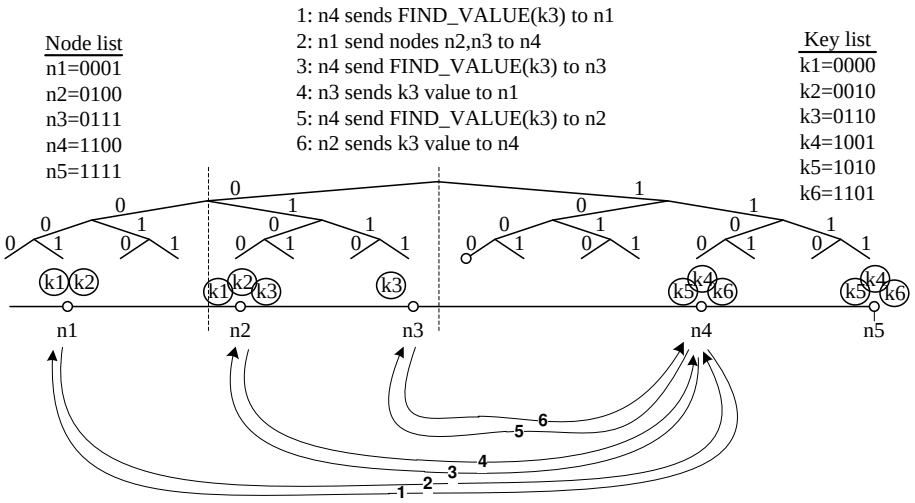


Σχήμα 4.6: *Μικρογραφία ενός δικτύου Kademlia*

σε περίπτωση που δεν έχει το κλειδί τοπικά. Στην τελευταία περίπτωση, αυτή η πληροφορία που επιστρέφει βοηθά τον αρχικό κόμβο να πλησιάσει περισσότερο στους κόμβους που είναι υπεύθυνοι για το ζητούμενο κλειδί στα επόμενα βήματα. Η διαδικασία σταματά όταν μια τιμή βρεθεί και επιστραφεί στον αρχικό κόμβο ή όταν δεν υπάρχουν πιο κοντινοί κόμβοι στο κλειδί και καμία τιμή δεν έχει βρεθεί. Σε μια επιτυχημένη αναζήτηση, ο αρχικός κόμβος θα αποθηκεύσει το ζευγάρι με ένα μήνυμα `STORE` στο κοντινότερο κόμβο ως προς το ζητούμενο κλειδί, ο οποίος δεν έχει το ζητούμενο ζευγάρι κλειδιού-τιμής. Επίσης, όταν ένας κόμβος δέχεται ένα μήνυμα από κάποιο και σε περιοδικά διαστήματα, ανταλλάσσει πληροφορία σχετικά με τα ζευγάρια κλειδιών-τιμής τα οποία είναι πιο κοντά στο αναγνωριστικό του άλλου κόμβου. Αυτό εγγυάται όταν θα υπάρχουν αντίγραφα τιμών σε όλους τους κοντινούς κόμβους και υποβοηθά τους κόμβους να ανακτήσουν τα δεδομένα που τους αντιστοιχούν όταν εισέρχονται στο δίκτυο.

Στο παράδειγμα του σχήματος 4.6 έχουμε μια μικρογραφία ενός δικτύου Kademlia, όπου οι κόμβοι και τα κλειδιά αντιστοιχίζονται στο ίδιο πεδίο τιμών μήκους 4bit για τα κλειδιά/αναγνωριστικά τους. Η τοπολογία λόγω της συνάρτησης αποστάσεως XOR που χρησιμοποιείται, μπορεί να παρασταθεί και σαν ένα δυαδικό δέντρο. Οι κόμβοι και τα κλειδιά παριστάνονται σαν τα φύλλα του δέντρου, ενώ κάθε κόμβος έχει πληροφορίες δρομολόγησης για τα κοντινά υποδέντρα και είναι υπεύθυνος για δεδομένα πιο κοντά στη δικιά του θέση. Για $\kappa=2$, ένα ζευγάρι κλειδιού-τιμής θα αποθηκευτεί τουλάχιστον σε δύο κοντινούς κόμβους (το *k3* αποθηκεύεται στους κόμβους *n2* και *n3*). Ένας τρίτος κόμβος μπορεί να αρχίσει να αναζητεί το ζευγάρι κλειδιού-τιμής ρωτώντας ένα κοντινό κόμβο. Εάν ο απομακρυσμένος κόμβος δεν έχει αποθηκευμένο το ζευγάρι, θα απαντήσει με μια λίστα κόμβων που είναι ακόμα πιο κοντά στο ζητούμενο κλειδί. Επαναλαμβάνοντας τη διαδικασία, ο αρχικός κόμβος θα καταλήξει τελικά σε ένα κόμβο υπεύθυνο για την αποθήκευση

ενός συγκεκριμένου κλειδιού (το *n4* αναζητεί το *k3*, που βρίσκεται στον *n3*, μέσω της λίστας των κόμβων που δέχεται από τον *n1*).

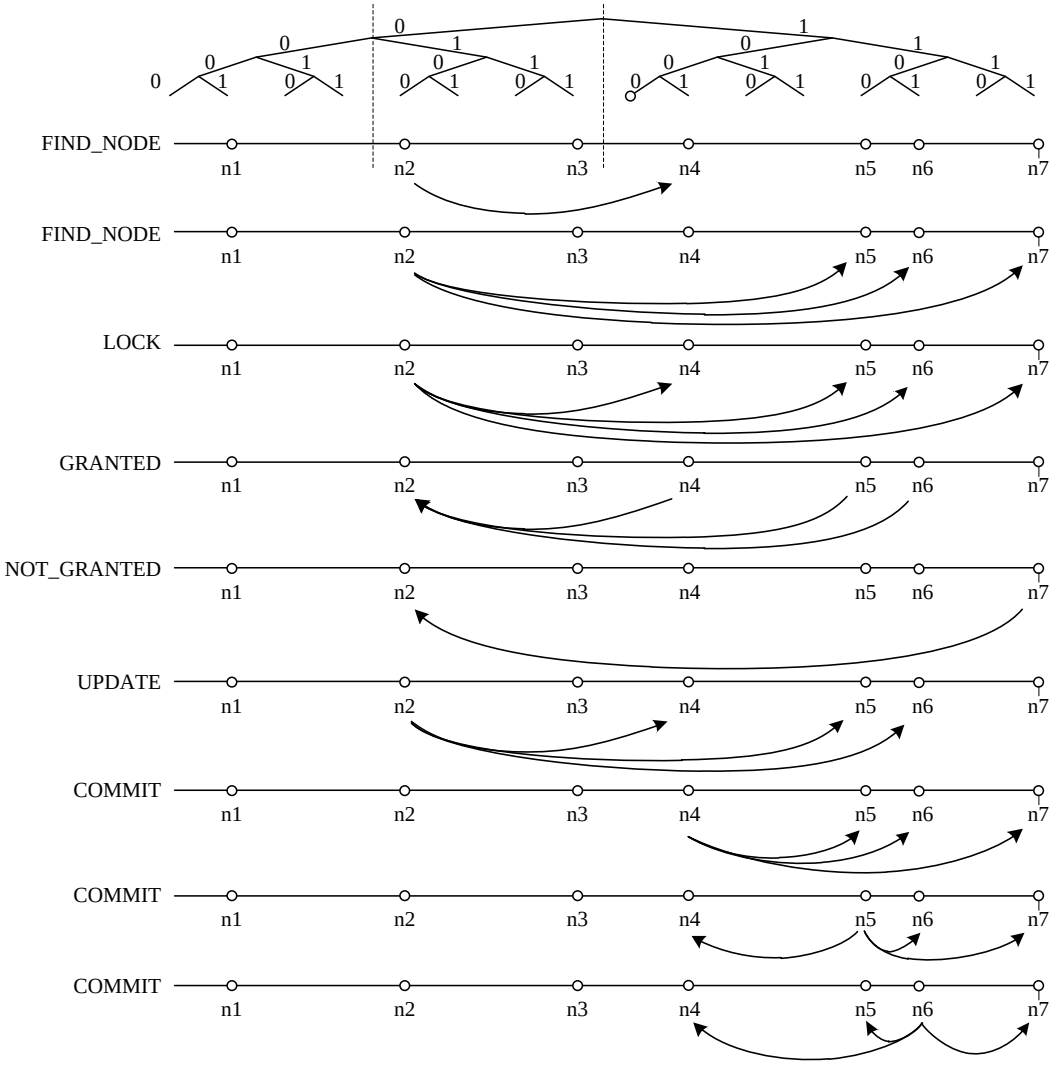


Σχήμα 4.7: *Μικρογραφία ενός δικτύου Kademlia με τον τροποποιημένο αλγόριθμο αναζήτησης. Μετά την εύρεση όλων των αντιγράφων γίνεται ο απαραίτητος έλεγχος της ημερομηνίας και ώρας, και αφού βρεθεί το πιο πρόσφατο ενημερώνονται και οι υπόλοιποι κόμβοι*

Ο τροποποιημένος αλγόριθμος αναζήτησης δουλεύει παρόμοια με την διαδικασία που ακολουθείται για το μήνυμα `FIND_NODE`, που αποστέλλεται όταν υπάρχει αίτημα για αποθήκευση δεδομένων στο δίκτυο. Πρώτα, ο κόμβος που ξεκινάει το ερώτημα, ανακαλύπτει τους κοντινούς κόμβους στο ζητούμενο κλειδί με μηνύματα `FIND_NODE` και στη συνέχεια στέλνει `FIND_VALUE` μηνύματα. Στη συνέχεια ο ίδιος κόμβος ελέγχει όλες τις τιμές που επιστρέφονται, επιλέγει την τιμή με το νεώτερο αριθμό έκδοσης και ειδοποιεί τους κόμβους που δεν είναι ενημερωμένοι με την σωστή τιμή για την αλλαγή αυτή. Φυσικά, εάν ένας κόμβος από τους πιο κοντινούς στο ζητούμενο κλειδί, απαντήσει με μια λίστα με τους κοντινότερους κόμβους, τότε θεωρείται μη ενημερωμένος. Όταν οι κ κοντινότεροι κόμβοι έχουν επιστρέψει κάποιο αποτέλεσμα, είτε αυτό είναι λίστα κόμβων, είτε είναι τιμή, τότε στέλνονται και τα σχετικά μηνύματα `STORE` στους μη ενημερωμένους κόμβους. Οι κόμβοι που δέχονται τέτοια μηνύματα αντικαθιστούν το δικό τους τοπικό αντίγραφο με την ενημερωμένη έκδοση. Η αποθήκευση ενός νέου κλειδιού στο σύστημα γίνεται με τον ίδιο ακριβώς τρόπο, με την μόνη διαφορά ότι η νεώτερη μόνο έκδοση επιστρέφεται στο χρήστη. Επιπλέον, η διαγραφή μιας τιμής ισοδυναμεί με την ενημέρωση της τιμής με μηδενική τιμή. Τα διαγραφέντα δεδομένα θα διαγραφούν τελικά από το σύστημα όταν λήξουν.

4.7.3 Προτεινόμενη λύση με XOROS

Μια διαφορετική λύση στο πρόβλημα παρουσιάζεται στο [Chazapis 07], όπου χρησιμοποιώντας σαν βάση το Kademlia προτείνονται διαφορετικές τροποποιήσεις, ώστε να δημιουργηθούν μηχανισμοί ενημέρωσης των αποθηκευμένων δεδομένων που να μπορούν να λειτουργήσουν και σε περιβάλλοντα με Βυζαντινές συνθήκες σφαλμάτων [Malkhi 99,Lamport 82]. Ο XOROS επιτυγχάνει την ενημέρωση δεδομένων ορίζοντας διακριτές σειριακές λειτουργίες που συνολικά ορίζουν μια συναλλαγή, με την έννοια της συναλλαγής των βάσεων δεδομένων, όπως φαίνεται στο σχήμα 4.8.



Σχήμα 4.8: *Απεικόνιση των βημάτων του μηχανισμού ενημέρωσης ενός δικτύου XOROS*

Όταν ένα κόμβος λαμβάνει ένα αίτημα για ενημέρωση δεδομένων, πρώτα ξεκινά μια αναζήτηση τύπου `FIND_NODE` ώστε να βρει τους κ κοντινότερους κόμβους που αποθηκεύουν το συγκεκριμένο αντικείμενο. Οι κόμβοι αυτοί είναι υπεύθυνοι για την έγκριση της ενημέρωσης του συγκεκριμένου αντικειμένου. Το επόμενο βήμα είναι να σταλεί ένα μήνυμα `LOCK` στους κόμβους αυτούς, που αποτελούν το *quorum* για το συγκεκριμένο αντικείμενο. Οι απαντήσεις θα είναι ένα μήνυμα τύπου `GRANTED` ή `NOT_GRANTED`, το οποίο εξαρτάται από το εάν προηγήθηκε αίτηση για ενημέρωση από άλλο κόμβο για το ίδιο αντικείμενο, και θα περιέχουν επίσης την υφιστάμενη τιμή του αντικειμένου. Για να προχωρήσει η ενημέρωση ο κόμβος θα πρέπει να έχει τουλάχιστον μ_{lock} θετικές ψήφους. Η τιμή του μ_{lock} θα πρέπει να είναι τέτοια ώστε να λειτουργεί ο αμοιβαίος αποκλεισμός αιτήσεων προς ενημέρωση στο ίδιο αντικείμενο. Τότε και μόνο τότε στέλνεται ένα μήνυμα `UPDATE` που περιέχει την ενημερωμένη τιμή του αντικειμένου. Για να προστατευθεί το δίκτυο από κακόβουλους κόμβους όμως, αυτό δεν αρκεί. Σαν τελευταίο βήμα, κάθε κόμβος που συμμετέχει στο *quorum* ανταλλάσσει με όλους τους υπόλοιπους, μηνύματα τύπου `COMMIT`, ώστε να ολοκληρωθεί η συναλλαγή, ακόμα και στους κόμβους που είχαν απαντήσει με μηνύματα `NOT_GRANTED`. Βέβαια, και σε αυτή την περίπτωση οι κόμβοι του *quorum* θα πρέπει να έχουν τουλάχιστον μ_{store} μηνύματα για την ίδια ενημέρωση, τόσα ώστε να λειτουργεί και πάλι ο αμοιβαίος αποκλεισμός αιτήσεων προς ενημέρωση στο ίδιο αντικείμενο. Σε περίπτωση που δεν υπάρχουν κακόβουλοι χρήστες και βλάβες στο δίκτυο, είναι προφανές ότι ισχύει $\mu_{lock} = \mu_{store} = \kappa/2+1$. Όμως, σε κάθε άλλη περίπτωση το μ_{store} έχει μια διαφορετική τιμή από το μ_{lock}. Πιο συγκεκριμένα το μ_{lock} είναι μεγαλύτερο από το μ_{store} κατά το μέγιστο αριθμό των κόμβων που ενδέχεται να έχουν απροσδιόριστη συμπεριφορά ή βλάβη.

Έστω, λοιπόν, ότι λ από τους κ κόμβους που διαμορφώνουν ένα *quorum* είναι κακόβουλοι, και επομένως έχουν απροσδιόριστη συμπεριφορά. Έστω σ οι κόμβοι που μπορούμε να εμπιστευτούμε ότι θα είναι συνεπείς στο πρωτόκολλο.

$$\kappa = \sigma + \lambda \tag{4.1}$$

Αν υποχρεώσουμε όλους τους κόμβους να επιστρέφουν την ίδια τιμή μεταξύ τους, κάθε κόμβος θα λάβει τουλάχιστον σ σωστές τιμές. Επιπλέον, κάθε κόμβος έχει αναγνωριστικό που προκύπτει από την διεύθυνση IP του βάση κάποιας κρυπτογραφικής συνάρτησης, ώστε να αποφευχθούν φαινόμενα πλαστοπροσωπίας. Σε κάθε φάση κλειδώματος όπου λαμβάνεται ένα μήνυμα `LOCK`, θα έχουμε σ κόμβους που θα κατανέμουν ορθά τα μηνύματα `GRANTED` και `NOT_GRANTED`, ενώ οι υπόλοιποι λ κόμβοι στη χειρότερη περίπτωση θα στείλουν μηνύματα `GRANTED` σε όλους. Το ίδιο μοτίβο ενδέχεται να συνεχιστεί και στις επόμενες φάσεις με τα μηνύματα `UPDATE` και `COMMIT`. Αν θεωρήσουμε λοιπόν ότι ένας κόμβος δέχεται μ μηνύματα, τότε από αυτά μπορεί να εμπιστευτεί ως ορθά τα $\mu - \lambda$, ενώ και $\sigma - (\mu - \lambda)$ κόμβοι για άλλη ανταγωνιστική ενημέρωση. Το μ_{store} πρέπει να

έχει την ελάχιστη ακέραια τιμή που ακόμα και με λ κακόβουλους κόμβους, κανένας ανταγωνιστής να μην μπορεί να παραβιάσει την συνάφεια του συστήματος έχοντας μ_{store} μηνύματα. Από τα παραπάνω προκύπτουν τα εξής:

$$\mu_{store} > \sigma - (\mu - \lambda) + \lambda \Rightarrow 2\mu_{store} > \sigma + \lambda + \lambda \Rightarrow \tag{4.2}$$

$$4.1, 4.2 \Rightarrow 2\mu_{store} > \kappa + \lambda \Rightarrow \mu_{store} > \frac{\kappa + \lambda}{2} + 1 \tag{4.3}$$

$$\mu_{lock} \geq \mu_{store} + \lambda \tag{4.4}$$

$$4.3, 4.4 \Rightarrow \mu_{lock} > \frac{\kappa + \lambda}{2} + 1 + \lambda \Rightarrow \mu_{lock} > \frac{\kappa + 3\lambda}{2} + 1 \tag{4.5}$$

$$\kappa \geq \mu_{lock} > \mu_{store} + \lambda \tag{4.6}$$

$$4.5, 4.6 \Rightarrow \mu_{lock} > \frac{\kappa + \lambda}{2} + 1 + \lambda \Rightarrow \mu_{lock} > \frac{\kappa + 3\lambda}{2} + 1 \tag{4.7}$$

Συμπερασματικά, το δίκτυο για να λειτουργεί χωρίς προβλήματα με λ κακόβουλους κόμβους, θα πρέπει να δημιουργούνται *quorum* με τουλάχιστον $3\lambda + 1$ κόμβους, το οποίο έχει αποδειχτεί βέλτιστο [Lamport 82]. Ακόμα και αν δεν υπάρχουν κακόβουλοι κόμβοι, αντίστοιχη συμπεριφορά μπορεί να σχηματιστεί από κόμβους με βλάβη ή σε στιγμές όπου υπάρχει ιδιαίτερα αυξημένος ρυθμός αφίξεων και αναχωρήσεων στο δίκτυο.

Πειραματική αξιολόγηση

Για την πειραματική αξιολόγηση του πρωτοτύπου δημιουργήσαμε ένα ρεαλιστικό σενάριο χρήσης όπου υπάρχουν 64 κόμβοι στο δίκτυο, με αποθηκευμένα 1000 κλειδιά και παράγουν τυχαίες αναζητήσεις και ενημερώσεις με αυξητικούς ρυθμούς. Μετρώντας το μέσο αριθμό μηνυμάτων και το χρόνο που χρειάζεται για κάθε λειτουργία του δικτύου, φαίνεται πως σε περίπτωση που δεν έχουμε μεταβολές στο αριθμό των κόμβων που απαρτίζουν το δίκτυο, τα αποτελέσματα παραμένουν σταθερά, ακόμα και όταν μεταβάλουμε συνεχώς το ρυθμό αιτήσεων από 1 λειτουργία κάθε 6 δευτερόλεπτα σε 10 λειτουργίες κάθε δευτερόλεπτο. Αυτό δείχνει πως το πρωτόκολλο XOROS, άρα και η υπηρεσία διαχείρισης αντιγράφων που υποστηρίζεται από αυτό, κλιμακώνεται σε περιπτώσεις flash crowd, όπου δηλαδή έχουμε μεγάλες και απότομες μεταβολές του φορτίου του δικτύου. Όταν οι κόμβοι που απαρτίζουν το δίκτυο αρχίζουν να αποχωρούν και νέοι εισέρχονται, κάποια μηνύματα χάνονται, οπότε μερικοί κόμβοι πρέπει να περιμένουν να λήξει η περίοδος του

timeout, ώστε να επεξεργαστούν μια εντολή. Παρόλα αυτά, όσο ο πληθυσμός των συμμετεχόντων κόμβων παραμένει σταθερός και οι πίνακες δρομολόγησης ενημερωθούν με την νέα πληροφορία, τα χαρακτηριστικά επίδοσης του δικτύου επιστρέφουν στα επιθυμητά επίπεδα. Ένα ενδιαφέρον εύρημα είναι πως σε περιόδους churn, ένας μεγαλύτερος ρυθμός αιτήσεων μπορεί να οδηγήσει σε παραγωγή περισσότερων μηνυμάτων, αλλά αυτό βοηθάει τους κόμβους να ανταποκρίνονται πιο γρήγορα σε αλλαγές του δικτύου και να ανανεώνουν τους πίνακες συμμόρφωσης σχετικά άμεσα.

Κατά την διάρκεια αυτής της σειράς δοκιμών, διερευνήθηκε και ο αντίκτυπος διάφορων παραμέτρων του δικτύου:

κ η παράμετρος *κ* ελέγχει το πλήθος των αντιγράφων για κάθε αντικείμενο και ταυτόχρονα ορίζει τον αριθμό της πλειοψηφίας για το πρωτόκολλο του αμοιβαίου αποκλεισμού

α η παράμετρος *α* ορίζει πόσα παράλληλα μηνύματα μπορούν να αποστέλλονται ταυτόχρονα κατά την διάρκεια μιας λειτουργίας του δικτύου

Q η παράμετρος *Q* ορίζει το πλήθος των κόμβων που θα παρουσιάσουν απροσδιόριστη συμπεριφορά ή θα αποτύχουν πριν την ολοκλήρωση μιας αιτήσεως

Όπως περιμέναμε, η παράμετρος αντιγραφής *κ* έχει τον πιο σημαντικό ρόλο στην διαμόρφωση του συνολικού όγκου μηνυμάτων και της απόκρισης του δικτύου. Ο πίνακας 4.1 συνοψίζει τα αποτελέσματα πολλαπλών δοκιμών, του παραπάνω σεναρίου χρήσης, για διαφορετικές τιμές του *κ*. Χαμηλώνοντας την παράμετρο *κ*, μειώνεται το πλήθος των κόμβων που πρέπει να συντονιστούν για κάθε λειτουργία, μειώνοντας με τον τρόπο αυτό την συνολική απόκριση του δικτύου. Όμως, η μέση απόκριση δεν είναι άμεσα ανάλογη στον αριθμό των μηνυμάτων, καθώς μεγάλο μέρος της επικοινωνίας γίνεται παράλληλα με πολλούς κόμβους. Διαιρώντας τα αποτελέσματα της απόκρισης με το μέσο κόστος ενός μηνύματος (80 sec) μας δίνει το μέσο πλήθος μηνυμάτων που πρέπει να σταλούν σειριακά, είτε λόγω του πρωτοκόλλου, είτε λόγω της παραμέτρου *α*. Όταν το δίκτυο είναι μικρό, όπως στην περίπτωση των 64 κόμβων, πιστεύουμε ότι η τιμή 5 για την παράμετρο *κ* είναι αρκετή. Από την άλλη μεριά, όταν το η υπηρεσία διαχείρισης αντιγράφων αναπτύσσεται μαζικά, σε ένα μεγάλο αριθμό κόμβων (π.χ. Desktop Grid), το να κρατήσουμε για την παράμετρο *κ* την τιμή 20, μας βοηθάει να αποφύγουμε απώλεια δεδομένων σε περίπτωση που έχουμε δικτυακές βλάβες ή άλλα απρογραμμάτιστα και μη ανακοινώσιμα προβλήματα κόμβων, ακόμα και αν το κόστος μηνυμάτων είναι αρκετά υψηλό.

Πίνακας 4.1: *Η επίδραση της παραμέτρου κ στην υπηρεσία διαχείρισης αντιγράφων*

κ	α	ϵ	Μέσος αριθμός μηνυμάτων	Μέση απόκριση (sec)
20	3	2	44	1.37
15	3	2	42	1.06
10	3	2	30	0.83
5	3	2	22	0.61

ΚΕΦΑΛΑΙΟ 5

Μεταφορά Δεδομένων

5.1 Το πρωτόκολλο GridFTP

Ένα βασικό κομμάτι στην συνολική αρχιτεκτονική του πλέγματος είναι η μεταφορά δεδομένων. Στο κομμάτι αυτό έχει επικρατήσει το δημοφιλές πρωτόκολλο μεταφοράς GridFTP [Allcock 02] [Allcock 05], το οποίο έχει οριστεί από το Open Grid Forum [Ope], και επικεντρώνεται στις μεταφορές αρχείων μεταξύ κόμβων του πλέγματος. Οι σύγχρονες διανομές ενδιάμεσου λογισμικού πλέγματος, όπως το Globus Toolkit [Glo] περιλαμβάνουν και μια υπηρεσία GridFTP, καθώς έχει πλέον κυριαρχήσει στο πλέγμα σαν de facto standard στις μεταφορές αρχείων. Πρόκειται για μια επέκταση του πρωτοκόλλου File Transfer Protocol (FTP) [Postel 85] με βελτιώσεις όπως η υποδομή ασφάλειας πλέγματος (Grid Security Infrastructure – GSI) [Foster 98] και η διαχείριση μεταφοράς δεδομένων από τρίτο συμμετέχοντα. Το τελευταίο παρέχει τη δυνατότητα ένας χρήστης ή μια εφαρμογή να ξεκινήσει, παρακολουθήσει και να έχει τον έλεγχο για μια μεταφορά αρχείων μεταξύ δύο άλλων μερών του πλέγματος, που θα έχουν το ρόλο της πηγής και του προορισμού του αρχείου. Αυτό επιτυγχάνεται με το διαχωρισμό του καναλιού ελέγχου από το κανάλι δεδομένων. Μια επιπρόσθετη επέκταση είναι η υποστήριξη για χειροκίνητη και αυτόματη ρύθμιση των μεγεθών του TCP buffer, ώστε να βελτιστοποιηθεί η μεταφορά μεγάλων αρχείων ή μεγάλων ομάδων από μικρότερα αρχεία. Τέλος, σημαντικές βελτιώσεις είναι και η υποστήριξη μεταφορών μερικών κομματιών από ένα αρχείο, μεταφορών από N σε M κόμβους και παράλληλα TCP streams μεταξύ δύο κόμβων ή και περισσότερων.

Παρόλ' αυτά οι τεχνικές που χρησιμοποιούνται στο GridFTP είναι βασισμένες στο μοντέλο του πελάτη-εξυπηρετητή επιφέροντας και τις παρενέργειες και τα ανεπιθύμητα χαρακτηριστικά του, όπως υπερφόρτωση του κεντρικού εξυπηρετητή, της ύπαρξης μοναδικού σημείου βλάβης και της αδυναμίας να αντεπεξέλθει σε φορτία όπου υπάρχει απότομη και δυσανάλογη αύξηση της ζήτησης για δημοφιλή δεδομένα και για περιορισμένο χρόνο (flash crowds). Για την βελτιστοποίηση των μεταφορών δεδομένων στο πλέγμα θα μπορούσε να χρησιμοποιηθεί η υπηρεσία αναζήτησης τοποθεσιών αντιγράφων (Replica Location Service). Μέσω ειδικών αλγορίθμων, που μπορούν να αντληθούν από τον χώρο των ομότιμων δικτύων, η μεταφορά δεδομένων μπορεί να εκμεταλλευτεί τα πολλαπλά αντίγραφα και να βελτιστοποιήσει τη ολική χρησιμοποίηση του δικτύου. Η λύση του κεντρικού αποθετηρίου που ακολουθείται, μπορεί να φτάσει τα όριά της όταν το πλήθος των συμμετεχόντων κόμβων και των διακινούμενων δεδομένων αυξηθεί μερικές τάξεις μεγέθους.

5.2 Σχετικές εργασίες

Η χρησιμοποίηση τεχνικών από το πεδίο των ομότιμων δικτύων για την αποδοτικότερη μεταφορά δεδομένων στο πλέγμα δεν είναι νέα. Υπάρχουν ήδη εργασίες που προσπάθησαν να εκμεταλλευτούν τέτοιες τεχνικές σε κάποιο βαθμό. Το GridTorrent Framework [Kaplan] χρησιμοποιεί το πρωτόκολλο BitTorrent και βασίζεται σε ένα κεντρικό κόμβο-συντονιστή (tracker), ο οποίος διατηρεί μια λίστα με όλες τις τοποθεσίες κάθε αρχείου και τους ενεργούς κόμβους που κατεβάζουν κάποιο αρχείο εκείνη την στιγμή. Αυτή η λίστα περιοδικά αποστέλλεται σε όλους του συμμετέχοντες και έτσι προωθείται μια συνεργασία για την αποδοτικότερη μεταφορά του κάθε αρχείου. Ταυτόχρονα, ο κόμβος-συντονιστής διατηρεί και λίστες πρόσβασης, ώστε να επιβάλει την επιθυμητή κάθε φορά πολιτική ασφάλειας για κάθε αρχείο. Αν ένας κόμβος δεν είναι εξουσιοδοτημένος για να έχει πρόσβαση σε κάποιο αρχείο, τότε δεν μπορεί να λάβει την λίστα με τις τοποθεσίες του αρχείου, άλλα και τους ενεργούς κόμβους εκείνη τη στιγμή. Η εργασία αυτή επεκτείνεται και προς την εκμετάλλευση παράλληλων καναλιών TCP, μεταξύ δύο κόμβων, με στόχο να υπερκεράσουν τους περιορισμούς του πρωτοκόλλου TCP και να καταφέρουν να φτάσουν στο μέγιστο όριο χρησιμοποίησης, υψηλής χωρητικότητας δικτυακές συνδέσεις. Παρόλα αυτά η συγκεκριμένη πρόταση εξακολουθεί να χρησιμοποιεί κεντρικές υπηρεσίες, με αποτέλεσμα να υποφέρει από όλα τα ανεπιθύμητα χαρακτηριστικά του μοντέλου του πελάτη-εξυπηρετητή, ενώ η μη χρησιμοποίηση καμιάς από τις υφιστάμενες βασικές υπηρεσίες του Πλέγματος παραμένει ένα σημαντικό μειονέκτημα.

Σε μια επίσης σχετική εργασία [Wei 05], οι συγγραφείς συγκρίνουν το πρωτόκολλο BitTorrent με το FTP για την μεταφορά δεδομένων σε περιβάλλον υπολογιστικού πλέγματος από σταθμούς εργασίας. Ένα τέτοιο περιβάλλον εφαρμογής, έχει το χαρακτηριστικό ότι οι διαθέσιμοι πόροι που συνήθως ειναι σταθμοί εργασίας, γίνονται διαθέσιμοι συγκεκριμένες μόνο ώρες κάθε μέρα, ενώ χρησιμοποιούνται ειδικοί αλγόριθμοι αναζήτησης πόρων λόγω και της εξαιρετικά ετερογενούς φύσης των πόρων. Οι εφαρμογές που εκτελούνται σε τέτοια συστήματα είναι εξαιρετικά απαιτητικές

σε υπολογιστική ισχύ, αλλά και εξαιρετικά απλό να παραλληλοποιηθούν με ένα απλουστευτικό διαμοιρασμό των δεδομένων εισόδου. Τέτοια παραδείγματα από την διεθνή βιβλιογραφία, βρίσκουμε στο SETI@Home [Anderson 97], το distributed.net [dis] και το BOINC [Anderson 04]. Η δουλειά τους καταλήγει στο συμπέρασμα ότι το πρωτόκολλο είναι εξαιρετικά αποδοτικό για μεγάλες μεταφορές αρχείων, όταν ο αριθμός των κόμβων αυξάνει. Παρόλα αυτά, το πρωτότυπο που χρησιμοποιούν βασίζεται σε μια κεντρική υπηρεσία καταλόγου και ένα κεντρικό αποθηκευτικό χώρο, δεν χρησιμοποιεί βασικές υπηρεσίες του Πλέγματος όπως το GridFTP πρωτόκολλο και η υπηρεσία αντιγράφων αρχείων, δεν χρησιμοποιεί τους μηχανισμούς της υποδομής ασφαλείας του Πλέγματος και δεν προσεγγίζει το πρόβλημα της αποδοτικής μεταφοράς αρχείων στον αποθηκευτικό χώρο του Πλέγματος.

Εκτός από τις προαναφερθέντες υπάρχουν και άλλες εργασίες στη γενικότερη περιοχή των κατανεμημένων συστημάτων που χρησιμοποιούν τεχνικές ομότιμων δικτύων για την μεταφορά αρχείων. Το Kangaroo [Thain 01] είναι ένα σύστημα μεταφοράς δεδομένων που στοχεύει στην βελτιστοποίηση της μεταφοράς κάνοντας μια καιροσκοπική χρησιμοποίηση μιας ομάδας εξυπηρετητών. Το Composite Endpoint Protocol [Weigle 05] συλλέγει δεδομένα μεταφοράς αρχείων από τον κάθε χρήστη και δημιουργεί ένα πρόγραμμα που βελτιστοποιεί τις μεταφορές αρχείων δημιουργώντας ένα ισοβαρώς κατανεμημένο κατευθυντικό γράφο. Παρόλα αυτά τα παραπάνω μοντέλο παραμένουν κεντρικά. Το Slurpie [Sherwood 04] ακολουθεί μια παρόμοια προσέγγιση με το BitTorrent, καθώς στοχεύει σε μαζικές μεταφορές αρχείων και κάνει τις ανάλογες υποθέσεις. Παρόλα αυτά δεν προωθεί την συνεργασία μεταξύ των κόμβων, όπως το πρωτόκολλο BitTorrent με τις ασφαλιστικές δικλείδες που διαθέτει.

5.3 Πρωτόκολλο GridTorrent

Η δική μας εργασία εστίασε στη χρησιμοποίηση τεχνικών από το πεδίο των ομότιμων δικτύων για την αποδοτικότερη μεταφορά δεδομένων στο πλέγμα, με κύριο στόχο την απαλοιφή κάθε κεντρικού σημείου βλάβης, αλλά και την υποστήριξη των ήδη υφιστάμενων και διαδεδομένων υπηρεσιών του Πλέγματος. Παράλληλα φροντίσαμε να αντιμετωπίσουμε και μερικές από τις αδυναμίες του υφιστάμενου μοντέλου, όπως της ανεπάρκειάς του να αντεπεξέλθει σε φορτία όπου υπάρχει απότομη και δυσανάλογη αύξηση της ζήτησης για δημοφιλή δεδομένα και για περιορισμένο χρόνο (flash crowds). Για την βελτιστοποίηση των μεταφορών δεδομένων στο πλέγμα χρησιμοποιούμε την υφιστάμενη υπηρεσία αναζήτησης τοποθεσιών αντιγράφων (Replica Location Service), αλλά και την προαναφερθείσα κατανεμημένη υπηρεσία αναζήτησης αντιγράφων (Distributed Replica Location Service) που προτείναμε στο πλαίσιο της εργασίας μας.

Σαν ένα πρώτο βήμα προς μια κατανεμημένη προσέγγιση, δημιουργήσαμε ένα πρωτότυπο GridTorrent [Zissimos 07], το οποίο εκμεταλλεύεται την πληροφορία που υπάρχει αποθηκευμένη στην υπηρεσία αναζήτησης αντιγράφων και χρησιμοποιώντας το πρωτόκολλο GridFTP, γινόταν

η μεταφορά του αρχείου τοπικά, ενώ σε περίπτωση που υπήρχαν και άλλοι GridTorrent κόμβοι ενεργοί την ίδια στιγμή, συνεργαζόντουσαν για την επίσπευση της μεταφοράς. Στην συνέχεια, αυτό το πρωτότυπο επεκτάθηκε και προστέθηκε υποστήριξη ασφάλειας με την υποδομή ασφαλείας πλέγματος (GSI), ενώ δημιουργήθηκε ένα νέο κανάλι επικοινωνίας με κάθε GridTorrent, στο οποίο μπορούσε να λάβει εντολές. Έτσι, ξεπεράστηκε το πρόβλημα που υπάρχει στις αρχιτεκτονικές τύπου torrent, όπου για ένα αρχείο δεν μπορεί να αρχίσει η μεταφορά δεδομένων, αν δεν ενδιαφερθεί πρώτα κάποιος για αυτό. Στη συνέχεια θα περιγράψουμε αναλυτικά το συγκεκριμένο πρωτόκολλο.

5.3.1 Γενικά

Το GridTorrent, όπως προαναφέραμε είναι μια τεχνική μεταφοράς αρχείων σε ένα περιβάλλον πλέγματος που πρωτοξεκίνησε σε δίκτυα ομότιμων κόμβων. Είναι βασισμένο στο BitTorrent, και επιτρέπει από κάθε κόμβο να κατεβάζει τοπικά ένα αρχείο από πολλές πηγές, ενώ ταυτόχρονα ανεβάζει το ίδιο αρχείο σε άλλους κόμβους που το ζητάνε. Χρησιμοποιώντας την ορολογία του BitTorrent, το GridTorrent δημιουργεί μια ομάδα κόμβων (swarm) όπου υπάρχουν οι κόμβοι ή οι χρήστες που ζητάνε το αρχείο ή το έχουν ημιτελές (leechers) και οι κόμβοι ή χρήστες (seeds) που έχουν ολόκληρο το αρχείο και το διαμοιράζουν στους πρώτους. Η συνεργατική φύση του αλγορίθμου που χρησιμοποιείται, διασφαλίζει την μέγιστη χρησιμοποίηση του διαθέσιμου εύρους της δικτυακής σύνδεσης, ενώ ταυτόχρονα ο μηχανισμός tit-for-tat προσφέρει επεκτασιμότητα σε συνθήκες υψηλού φορτίου (flash crowd). Ειδικότερα, το GridTorrent χρησιμοποιεί την υπάρχουσα υποδομή πλέγματος, καθώς οι υφιστάμενοι αποθηκευτικοί χώροι και εξυπηρετητές GridFTP μπορούν να χρησιμοποιηθούν σαν seeds και οι κόμβοι GridTorrent να μεταφέρουν από αυτούς αρχεία με την επιλογή του GridFTP για μερική μεταφορά αρχείου. Το αρχείο *.torrent* που χρησιμοποιείται στο BitTorrent αντικαταστάθηκε από την ήδη υπάρχουσα υπηρεσία στο Πλέγμα, υπηρεσία αναζήτησης αντιγράφων. Για να ξεκινήσει το κατέβασμα ενός αρχείου, πρέπει να υπάρχει το μοναδικό αναγνωριστικό του αρχείου (unique identifier - *UID*), το οποίο είναι συνήθως το αποτέλεσμα μιας κρυπτογραφικής συνάρτησης με είσοδο τα περιεχόμενα του αρχείου. Οι υπόλοιπες απαραίτητες πληροφορίες μπορούν να αποκτηθούν από την υπηρεσία αναζήτησης αντιγράφων χρησιμοποιώντας το UID. Επομένως, όλοι οι κόμβοι που συμμετέχουν σε ένα GridTorrent swarm είναι επίσης καταγεγραμμένοι στην υπηρεσία αναζήτησης αντιγράφων, ώστε να μπορούν να βρεθούν μεταξύ τους και να συνεργαστούν.

5.3.2 Ασφάλεια

Σε ένα περιβάλλον Πλέγμα, μόνο τακτοποιημένοι χρήστες είναι έμπιστοι για την μεταφορά κομματιών ή ενός ολόκληρου αρχείου. Επίσης, η κρυπτογράφηση προσφέρεται για την μεταφορά

ευαίσθητων δεδομένων. Για την εγγύηση των προδιαγραφών ασφαλείας, το GridTorrent, χρησιμοποιεί την υλοποίηση της Υποδομής Ασφάλειας Πλέγματος (GSI) από το Globus Toolkit. Πιο συγκεκριμένα, το GridTorrent αναπτύσσει όλους τους πρότυπους μηχανισμούς της υποδομής, δηλαδή ταυτοποίηση, ακεραιότητα και κρυπτογραφία. Κάθε φορά που χρησιμοποιείται ένα TCP socket, περιτυλίσσεται μαζί με τα πιστοποιητικά του χρήστη ή της υπηρεσίας σε ένα Grid Socket, σύμφωνα με τις μεθόδους που προσφέρει το GSS API. Έτσι, κάθε φορά που ένα κανάλι επικοινωνίας δημιουργείται, τηρούνται οι παράμετροι της κρυπτογραφίας, ακεραιότητας και ταυτοποίησης που έχουν τεθεί από τον χρήστη του κόμβου ή τον διαχειριστή.

5.3.3 Κανάλι Ελέγχου

Στο GridTorrent, οι κόμβοι επικοινωνούν μεταξύ τους και ανταλλάσσουν πληροφορίες σχετικά με την εξέλιξη της μεταφοράς δεδομένων. Ένα νέο χαρακτηριστικό του GridTorrent, είναι η δυνατότητα ένας κόμβος να στέλνει απομακρυσμένες εντολές σε ένα δεύτερο, μέσω ενός ξεχωριστού καναλιού επικοινωνίας, το κανάλι ελέγχου. Η δυνατότητα αυτή είναι παρόμοια με τη σχεδιαστική επιλογή του GridFTP, για χωριστά κανάλια μεταφοράς εντολών και δεδομένων και επιτρέπει εκτός των άλλων και οργάνωση μεταφοράς δεδομένων μεταξύ δύο κόμβων από ένα τρίτο (third party transfer). Με την δυνατότητα αυτή το BitTorrent ξεπερνά το μειονέκτημα που υφίσταται σε αρχιτεκτονικές τύπου torrent, όπου για να αρχίσει η μεταφορά δεδομένων για ένα αρχείο, πρέπει κάποιος να βρει αυτό το αρχείο και στην συνέχεια να ενδιαφερθεί, το οποίο είναι κοινή πρακτική σε δίκτυα ομότιμων κόμβων όπως το BitTorrent, αλλά διαφέρει αρκετά από το μοντέλο χρήσης που υπάρχει στο Πλέγμα. Αναλυτικότερα, το κανάλι ελέγχου υποστηρίζει τις ακόλουθες εντολές:

`Start` $[UID]$ $[RLS]$ Ξεκινά την μεταφορά του αρχείου με το μοναδικό αναγνωριστικό UID τοπικά, αφού πρώτα αντλήσει τις απαιτούμενες πληροφορίες από το RLS.

`Start` $[filename]$ $[RLS]$ Ξεκινά τον διαμοιρασμό του αρχείου με το όνομα filename, που υπάρχει στο τοπικό αποθηκευτικό χώρο, αφού πρώτα δημοσιεύσει τις απαιτούμενες πληροφορίες στο RLS.

`Stop` $[UID]$ Σταματά μια ενεργή μεταφορά αρχείου, το οποίο έχει το μοναδικό αναγνωριστικό UID.

`Delete` $[filename]$ Σβήνει το αρχείο με όνομα filename από τον τοπικό αποθηκευτικό χώρο.

`List` Εμφανίζει μια λίστα με όλα τις ενεργές μεταφορές αρχείων που συμμετέχει ο κόμβος.

`Get` $[UID]$ Εμφανίζει στατιστικά χρήσης μιας ενεργής μεταφοράς αρχείου σχετικά με τα ανταλλαχθέντα μηνύματα και δεδομένα. Σαν παράμετρος δίνεται το μοναδικό αναγνωριστικό του αρχείου UID.

`Shutdown` Σταματά την λειτουργία του GridTorrent στο κόμβο.

5.3.4 Μεταδεδομένα GridTorrent

Για την λειτουργία του GridTorrent χρειάζονται μερικά μεταδεδομένα σχετικά με το κάθε αρχείο. Τα δεδομένα αυτά φυλάσσονται κάθε φορά στην υπηρεσία αναζήτησης αντιγράφων. Στην υφιστάμενη υλοποίηση αυτό είναι εφικτό καθώς όπως προαναφέραμε, υπάρχει δυνατότητα αποθήκευσης περαιτέρω παραμέτρων ανά λογικό όνομα αρχείο ή/και ανά φυσικό όνομα αρχείου. Επίσης, για μεγαλύτερη ασφάλεια από βλάβες και επεκτασιμότητα μπορει να χρησιμοποιηθεί και η κατανεμημένη υπηρεσία αναζήτησης αρχείων (DRLS) που βασίζεται σε τεχνολογίες δικτύων ομότιμων κόμβων. Τα μεταδεδομένα αυτά μπορεί να χωριστούν σε δυο κατηγορίες:

- στατική πληροφορία που αφορά μη μεταβλητά στοιχεία του αρχείου όπως το όνομα και το μέγεθός του
- και δυναμική πληροφορία που αφορά σε συνεχή μεταβλητά στοιχεία για ένα αρχείο, όπως οι τοποθεσίες αποθήκευσής του

Στην συνέχεια παραθέτουμε την στατική πληροφορία που χρησιμοποιείται για την αρχικοποίηση μιας μεταφοράς αρχείου μέσω GridTorrent:

Logical Filename (LFN): Το λογικό όνομα του αρχείου, το οποίο χρησιμοποιεί και θέτει ο χρήστης.

File size: Το συνολικό μέγεθος του αρχείου σε bytes.

File hash type: Ο τύπος της συνάρτησης κατακερματισμού που χρησιμοποιείται για την παραγωγή ενός κλειδιού που αναγνωρίζει μοναδικά το αρχείο με βάση τα δεδομένα που περιέχει. Η συνάρτηση κατακερματισμού χρησιμοποιείται για τον έλεγχο ως προς την ορθότητα των δεδομένων και την προστασία από αλλοιώσεις.

File hash: Το κλειδί που παράγεται από τα δεδομένα του αρχείου και την συνάρτηση κατακερματισμού. Χρησιμοποιείται επίσης και ως το μοναδικό αναγνωριστικό - UID για το αρχείο.

Piece length: Το μέγεθος κάθε κομματιού του αρχείου σε bytes. Τα κομμάτια αυτά χρησιμοποιούνται για έλεγχο και ανταλλαγή πληροφοριών μεταξύ των κόμβων GridTorrent. Αφού ένας κόμβος λάβει ένα ολόκληρο κομμάτι και το ελέγξει ως προς την ορθότητά του, τότε ενημερώνει όλους τους υπόλοιπους κόμβους.

Piece hash type: Ο τύπος της συνάρτησης κατακερματισμού που χρησιμοποιείται για την παραγωγή ενός κλειδιού που αναγνωρίζει μοναδικά κάθε κομμάτι του αρχείο με βάση τα δεδομένα που περιέχει. Η συνάρτηση κατακερματισμού χρησιμοποιείται για τον έλεγχο ως προς την ορθότητα των δεδομένων κάθε κομματιού και την προστασία από αλλοιώσεις.

Piece hash: Τα κλειδιά που παράγονται από τα δεδομένα των κομματιών του αρχείου και την συνάρτηση κατακερματισμού.

Όσον αφορά την δυναμική πληροφορία σχετικά με τις τοποθεσίες αποθήκευσης ενός αρχείου, αυτή ελέγχεται περιοδικά για αλλαγές κατά την μεταφορά ενός αρχείου. Ουσιαστικά, πρόκειται για μια λίστα με φυσικά ονόματα αρχείων, που έχουν την ακόλουθη δομή:

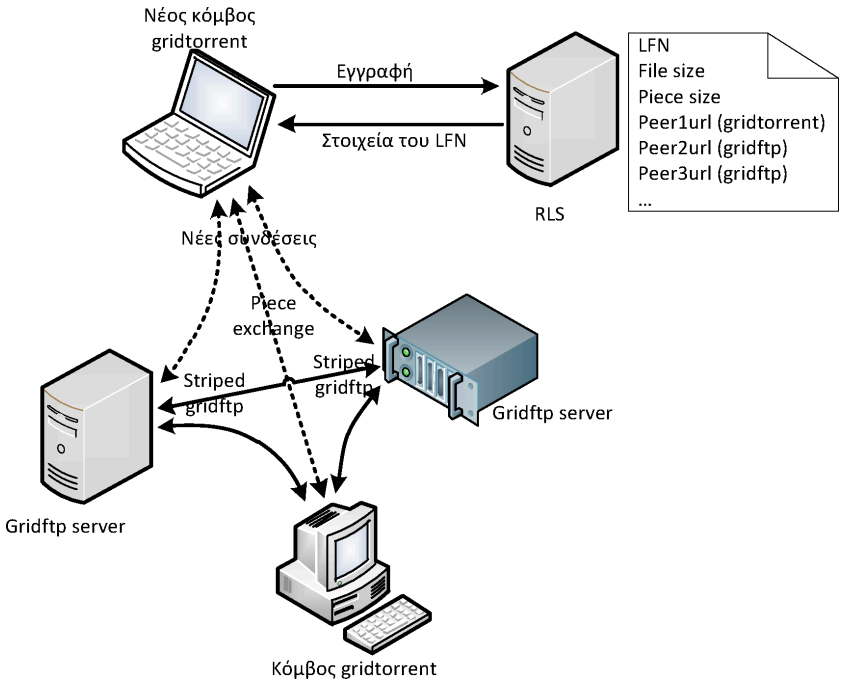
```
protocol://fqdn:port/path/to/file
```

όπου το `protocol` είναι το πρωτόκολλο που χρησιμοποιείται για την μεταφορά δεδομένων. Τη συγκεκριμένη στιγμή υποστηρίζονται τα πρωτόκολλα `gsiftp` (GridFTP) και `gtp` (GridTorrent). Το fully qualified domain name `fqdn` είναι το όνομα που ειναι καταχωρημένο στην υπηρεσία ονοματοδοσία του Internet (DNS) για τον κάθε κόμβο και ακολουθεί η πλήρης διαδρομή στο σημείο του τοπικού δίσκου που έχει αποθηκευτεί το αρχείο. Ένα πλεονέκτημα της προτεινόμενης προσθήκης είναι η χρήση ήδη υλοποιημένων χαρακτηριστικών για την μοντελοποίηση της λύσης μας, καθώς με αυτό τον τρόπο διατηρείται η συμβατότητα με τις υπάρχουσες τεχνολογίες και την αρχιτεκτονική του πλέγματος. Τα επιπλέον δεδομένα που θα αποθηκευτούν στην υπηρεσία αναζήτησης αντιγράφων δεν λαμβάνονται υπόψη από τις υπάρχουσες εφαρμογές. Εάν μια εφαρμογή δεν αναγνωρίσει το πρόθεμα του GridTorrent, απλά θα το αγνοήσει. Επομένως, οι προτεινόμενες αλλαγές στην αρχιτεκτονική του πλέγματος, όχι μόνο βελτιώνουν την απόδοση των μεταφορών δεδομένων, αλλά ενσωματώνονται στον υφιστάμενο περιβάλλον χωρίς επιπτώσεις στις υφιστάμενες υπηρεσίες και εφαρμογές.

5.3.5 Λεπτομέρειες υλοποίησης

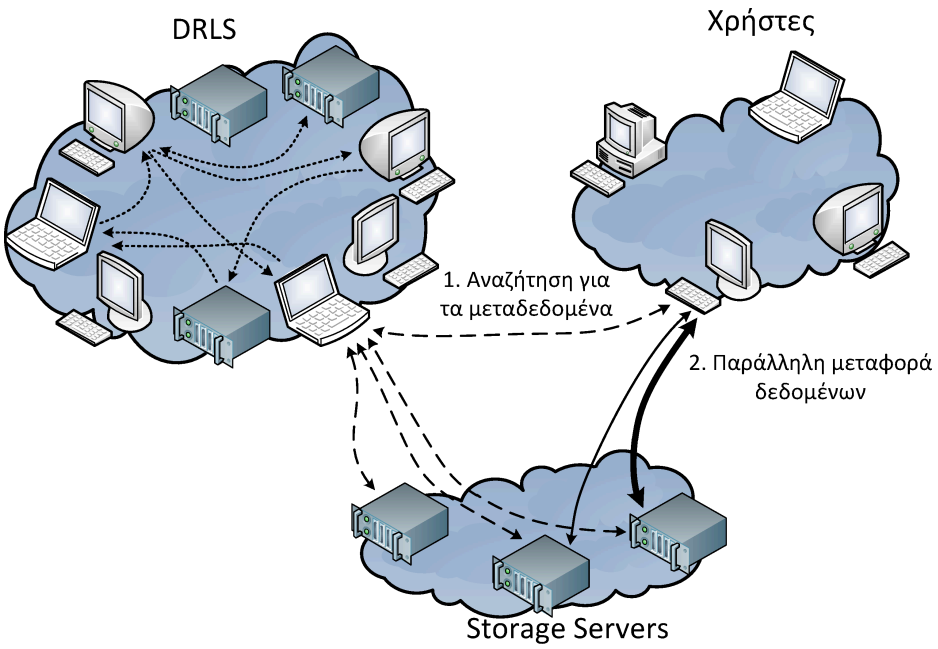
Το GridTorrent είναι μια μορφή του BitTorrent με ενεργοποιημένα χαρακτηριστικά της υποδομής πλέγματος. Χρησιμοποιεί υφιστάμενα πρωτόκολλα υποδομής πλέγματος για να υλοποιήσει μια αποδοτική υπηρεσία μεταφοράς αρχείων, καθώς έχει την δυνατότητα να επικοινωνεί με αποθετήρια αρχείων με πρωτόκολλο πρόσβασης GridFTP. Ο σχεδιασμός του GridTorrent περιλαμβάνει τα εξής υποσυστήματα σε επίπεδο λογισμικού:

- Τον RLSManager, ο οποίος χειρίζεται όλες τις επικοινωνίες με την υπηρεσία αναζήτησης αντιγράφων. Το υποσύστημα αυτό είναι υπεύθυνο για να βρίσκει τις πληροφορίες σχετικά με το αρχείο που μεταφέρεται (μέγεθος αρχείου, πλήθος κομματιών κλπ), να ενημερώνει την υπηρεσία για νέα αντίγραφα και να βρίσκει τα ήδη υπάρχοντα.
- Τον PeerManager, ο οποίος χειρίζεται όλες τις επικοινωνίες με τους υπόλοιπους κόμβους GridTorrent ή GridFTP.
- Τον DiskManager, ο οποίος χειρίζεται την επικοινωνία με το υποσύστημα αποθήκευσης (disk I/O). Εάν υπάρχουν αθροίσματα ελέγχου, τότε στο σημείο αυτό γίνονται και οι απαραίτητοι έλεγχοι για την ακεραιότητα των δεδομένων.



Σχήμα 5.1: *Παράδειγμα ανάπτυξης GridTorrent όπου ένας κόμβος χρησιμοποιεί τις υφιστάμενες υπηρεσίες Πλέγματος (Replica Location Service και GridFTP service)*

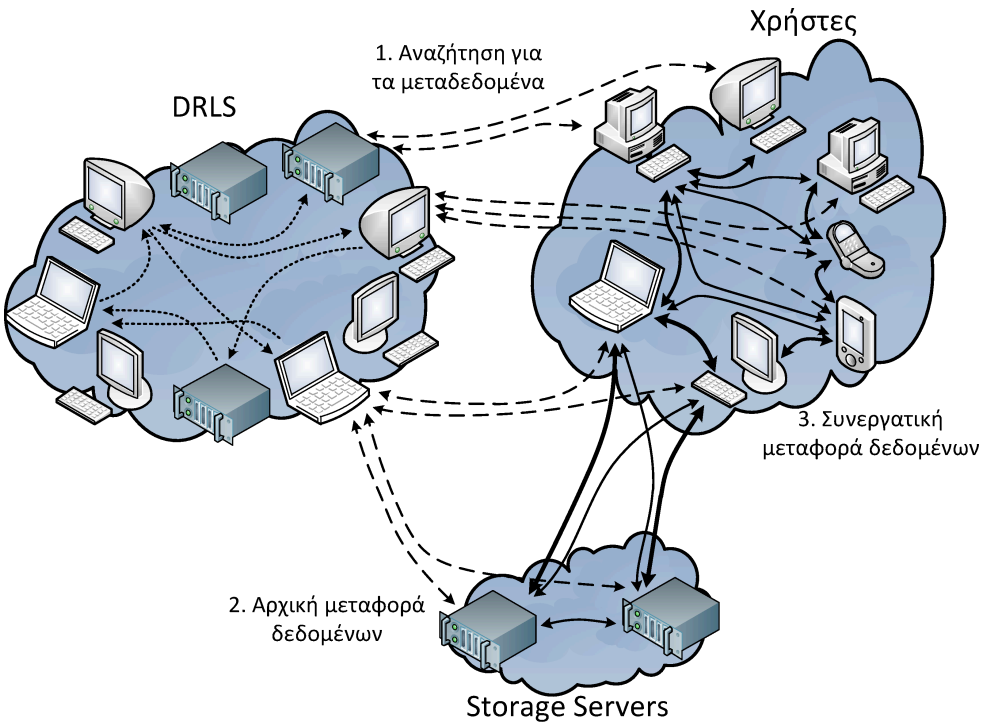
Το βασικό πλεονέκτημα του GridTorrent είναι η ύπαρξη του έμφυτου αλγόριθμου βελτιστοποίησης επιλογής διαθέσιμων αντιγράφων. Ένα αίτημα στο GridTorrent για ένα αρχείο θα πυροδοτήσει μια επικοινωνία με την υπηρεσία αναζήτησης αντιγράφων. Αυτή η επικοινωνία είναι δυνατόν να γίνεται σε περιοδικά διαστήματα, ώστε κάθε κόμβος να ενημερώνεται για τις νέες αφίξεις και αναχωρήσεις κόμβων από το πλέγμα, οι οποίοι μπορεί να έχουν δυνατότητες GridFTP ή GridTorrent. Μόλις ο νεοεισαχθείς κόμβος λάβει την λίστα με τους υπόλοιπους κόμβους που διαθέτουν μέρος ή ολόκληρο το αρχείο, το GridTorrent ενεργεί ανάλογα με το πρόθεμα του κάθε PFN που υπάρχει στη λίστα, δηλαδή του πρωτοκόλλου πρόσβασης. Εάν πρόκειται για κόμβο GridTorrent, τότε αρχικοποιείται η σύνδεση βάση το πρωτόκολλο του GridTorrent. Δηλαδή, ανταλλάσσεται πληροφορία σχετικά με τα διαθέσιμα κομμάτια εκατέρωθεν και στην συνέχεια αρχίζει η λήψη κομματιών κατά τυχαία σειρά. Επιπρόσθετα, κάθε φορά που ένας κόμβος λαμβάνει μια ολόκληρη ομάδα κομματιών, ειδοποιεί όλους τους γείτονες του για το νέο απόκτημα του με το μήνυμα have του πρωτοκόλλου BitTorrent. Για να πάρει ένα κομμάτι, ο κόμβος στέλνει ένα μήνυμα request σύμφωνα με το πρωτόκολλο BitTorrent, το οποίο περιέχει τον αριθμό του κομματιού και το μήκος του σύμφωνα με τις πληροφορίες που γνωρίζει από την υπηρεσία αναζήτησης αντιγράφων. Στην συνέχεια γνωρίζοντας τα διαθέσιμα κομμάτια αρχίζει να τα λαμβάνει επιλέγοντας



Σχήμα 5.2: *Παράδειγμα ανάπτυξης GridTorrent όπου ένας κόμβος προσπαθεί να αποκτήσει τα δεδομένα ενός αρχείου*

πρώτα τα λιγότερο δημοφιλή (rarest-first πολιτική). Σε περίπτωση που ο απομακρυσμένος κόμβος χρησιμοποιεί μόνο το GridFTP σαν πρωτόκολλο πρόσβασης, τότε θα περιέχει ολόκληρο το αρχείο και η λήψη κομματιών γίνεται με τη χρήση μερικής μεταφοράς του πρωτοκόλλου GridFTP (partial transfer). Η επιλογή του κομματιού γίνεται με τον ίδιο τρόπο. Ένα παράδειγμα υλοποίησης φαίνεται στο 5.2.

Η γνώση των διαθέσιμων κομματιών προς λήψη είναι αρκετή για να εφαρμοστούν πολλές πολιτικές μεταφοράς δεδομένων. Ένα παράδειγμα είναι η επιλογή του σπανιότερου για την αντιμετώπιση φόρτου από μεγάλα και δημοφιλή αρχεία, όπως αναφέρθηκε παραπάνω. Μια άλλη προσέγγιση είναι η βελτιστοποίηση των αλγορίθμου επιλογής, ώστε να δείχνει προτίμηση σε κόμβους που επιτυγχάνουν καλύτερους ρυθμούς μετάδοσης. Όσο τα δεδομένα λαμβάνονται από τους διάφορους γειτονικούς κόμβους, το GridTorrent, διατηρεί στατιστικά για το μέσο χρόνο λήψης ανά κόμβο. Κάθε φορά που ένα κομμάτι λαμβάνεται, ο μέσος χρόνος λήψης υπολογίζεται χρησιμοποιώντας τον τρέχοντα χρόνο μαζί με το ιστορικό του κόμβου. Το πρωτόκολλο GridTorrent χρησιμοποιεί το αλγόριθμο tit-for-tat, όπου όλοι οι κόμβοι συμβάλουν στην συνολική μεταφορά του αρχείου και αποθαρρύνονται κόμβοι που δεν προσφέρουν πόρους για το σκοπό αυτό. Κάθε κόμβος πρέπει να στείλει μερικά κομμάτια σε ένα αριθμό N κόμβων κάθε φορά, χρησιμοποιώντας τον αλγόριθμο optimistic unchoke που αναφέρθηκε.



Σχήμα 5.3: *Παράδειγμα ανάπτυξης GridTorrent όπου περισσότεροι από ένας κόμβοι προσπαθούν να αποκτήσουν τα δεδομένα ενός αρχείου*

5.4 Πειραματική Αξιολόγηση

Στο πλαίσιο της πειραματικής αξιολόγησης υλοποιήθηκε ένα πρωτότυπο, το GridTorrent. Το πρωτότυπο δεν βασίστηκε σε κάποια προϋπάρχουσα υλοποίηση αλλά έγινε ανάπτυξη λογισμικού από την αρχή στη γλώσσα προγραμματισμού JAVA. Η υλοποίηση αυτή χρησιμοποιεί το Replica Location Service API, το Grid Security Infrastructure API και το GridFTP client API. Έτσι είναι την διασύνδεσή του GridTorrent με τις υφιστάμενες υπηρεσίες του πλέγματος, όπως αποθετήρια δεδομένων που χρησιμοποιούν εξυπηρετητές GridFTP, μεταδεδομένα αποθηκευμένα στο Globus RLS καθώς και x509 πιστοποιητικά που έχουν εκδοθεί στους χρήστες για λόγους ταυτοποίησης, εξουσιοδότησης, ελέγχου ακεραιότητας και εμπιστευτικότητας. Στα πειράματα που παρουσιάζονται στην συνέχεια το GridTorrent έχει ενεργοποιηθεί σε ένα αριθμό φυσικών κόμβων, οι οποίοι ελέγχονται απομακρυσμένα από το κανάλι επικοινωνίας, το οποίο χρησιμοποιείται για την ενεργοποίηση και την παρακολούθηση μια μεταφοράς δεδομένων.

5.4.1 Χαρακτηριστικά ασφάλειας

Πρώτα δοκιμάστηκε η επίδραση που έχει η εφαρμογή των χαρακτηριστικών ασφαλείας στην συνολική επίδοση της μεταφοράς δεδομένων μέσω του GridTorrent παρακολουθώντας το χρόνο που χρειάζεται για την μεταφορά ενός αρχείου 128MB. Διακρίνουμε τρεις διαφορετικές περιπτώσεις ρυθμίσεων του Globus GSI:

Μόνο ταυτοποίηση Αυτή είναι η πιο απλή περίπτωση όπου κάθε κόμβος GridTorrent διαθέτει και επιδεικνύει για λόγους αμοιβαίας ταυτοποίησης ένα έγκυρο πιστοποιητικό x509 υπογεγραμμένο από μια κοινά αποδεκτή Αρχή Πιστοποίησης.

Έλεγχος ακεραιότητας Σε αυτή την περίπτωση πέραν από την αμοιβαία ταυτοποίηση, κάθε μήνυμα φέρει υπογραφή και ο αποδέκτης τα επιβεβαιώνει ώστε να αποτρέπονται οι επιθέσεις man-in-the-middle, όπου με την παρέμβαση κάποιου τρίτου το μήνυμα αλλοιώνεται.

Κρυπτογράφηση Αυτή είναι η πιο ασφαλής περίπτωση όπου ισχύουν τα παραπάνω και επιπλέον κάθε μήνυμα κρυπτογραφείται.

Οι δοκιμές έγιναν σε 100 επαναλήψεις μεταξύ δυο κόμβων (διαφορετικών κάθε φορά) οι οποίοι βρίσκονταν συνδεδεμένοι στο ίδιο τοπικό δίκτυο (LAN). Όπως φαίνεται στον Πίνακα 5.1 μόνο η τρίτη περίπτωση με ενεργοποιημένη την κρυπτογράφηση έχει αξιοσημείωτη επιβάρυνση (περίπου 30%) στο χρόνο που χρειάζεται για την μεταφορά του αρχείου. Αυτό το κόστος ειναι φυσικό, καθώς κατά την κρυπτογράφηση κάθε μήνυμα αντιγράφεται εις διπλούν στην μνήμη και αναλύεται από ένα υπολογιστικά απαιτητικό αλγόριθμο κρυπτανάλυσης.

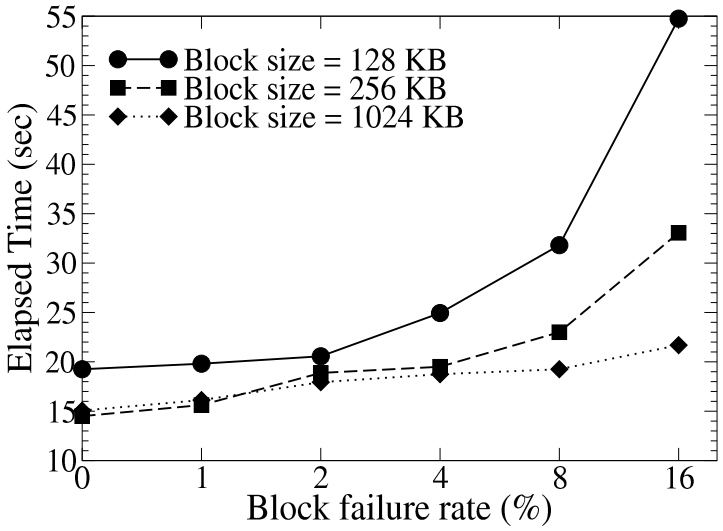
Πίνακας 5.1: *Επιβάρυνση των μηχανισμών ασφάλειας στον συνολικό χρόνο μεταφοράς του αρχείου*

Ρυθμίσεις ασφάλειας	μέσος χρόνος (sec)	επιβάρυνση
ταυτοποίηση	43,3	0%
ταυτοποίηση + έλεγχος ακεραιότητας	44,3	2%
ταυτοποίηση + κρυπτογράφηση	55,3	27%

5.4.2 Ανοχή σε σφάλματα

Στην συνέχεια δοκιμάστηκε το GridTorrent σε ένα δίκτυο με μεγάλο αριθμό σφαλμάτων στην μετάδοση πακέτων, ώστε να εξεταστεί η ανοχή του σε σφάλματα. Στην περίπτωση αυτή χρησιμοποιήθηκε ένας κόμβος ο οποίος είχε το ρόλο του seed και 16 κόμβοι οι οποίοι ενεργούσαν σαν leechers για ένα αρχείο 128MB. Για την καλύτερη αξιολόγηση του GridTorrent έπρεπε πρώτα γίνουν οι σωστές ρυθμίσεις όσον αφορά το μέγεθος κάθε block και κάθε piece. Μετά από εκτεταμένο πειραματισμό επιλέχθηκαν οι τιμές 1024KB για το ένα piece και 32KB για ένα block. Εδώ θα

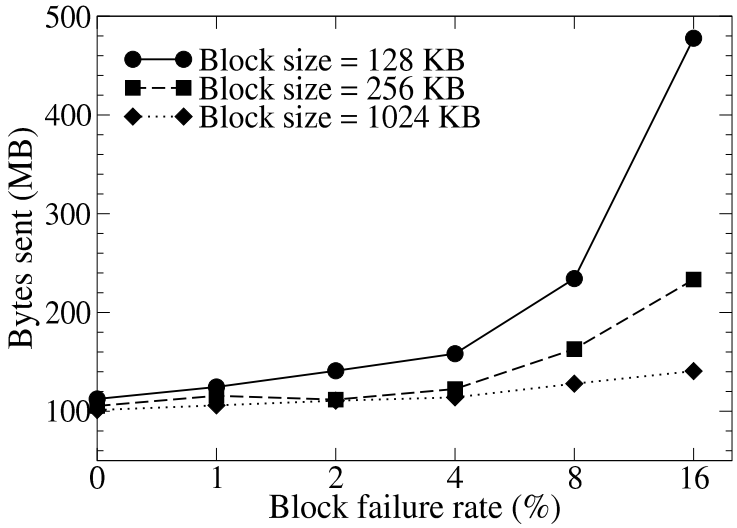
πρέπει να υπενθυμίσουμε ότι στο GridTorrent υπάρχει ένα hash για την επαλήθευση κάθε piece, ενώ οι κόμβοι μεταξύ τους αποθηκεύουν blocks. Για την εξομοίωση ενός δικτύου με σφάλματα στην μετάδοση των πακέτων κάθε κόμβος (leechers και seeds) παίρνει μια απόφαση βασισμένη σε μια ομοιόμορφα κατανεμημένη ψευδοτυχαία μεταβλητή κάθε φορά που στέλνει σε κάποιον ένα block ώστε να στείλει λανθασμένα δεδομένα. Τα αποτελέσματα απεικονίζονται στα σχήματα 5.4 και 5.5. Κατ' αρχήν σε όλες τις περιπτώσεις τα η μεταφορά του αρχείου ολοκληρώνεται με κάποια αποδεκτή καθυστέρηση, σε αντίθεση με την περίπτωση που θα είχαμε χρησιμοποιήσει GridFTP το οποίο δεν έχει μηχανισμό προστασίας από τέτοια λάθη. Επιπρόσθετα, παρατηρούμε ότι όσο ο ρυθμός σφαλμάτων αυξάνει οι μεταφορές αρχείων που βασίζονται σε μικρότερα μεγέθη block επηρεάζονται περισσότερο. Αυτό συμβαίνει διότι μια λήψη ενός λανθασμένου block έχει ως αποτέλεσμα την επαναμεταφορά και όλων των υπόλοιπων block που βρίσκονται σε αυτό το piece. Μικρότερο μέγεθος block σημαίνει περισσότερα blocks ανά piece, άρα όσο μικρότερο μέγεθος block τόσο περισσότερα blocks πρέπει να επαναμεταφερθούν, ενώ υπάρχει και η επιβάρυνση της κίνησης ελέγχου των συναλλαγών block μεταξύ των κόμβων. Σε περιπτώσεις λοιπόν, που το μέγεθος block είναι $\frac{1}{8}$ του μεγέθους piece, η καθυστέρηση είναι 3 με 4 φορές σε σχέση με την περίπτωση όπου τα μεγέθη block και piece είναι ίδια και για ρυθμούς σφαλμάτων μέχρι 16.



Σχήμα 5.4: *Μέσος χρόνος μεταφοράς του αρχείου σε διαφορετικά ποσοστά λανθασμένης μετάδοσης δεδομένων και μεγέθους block.*

5.4.3 Σύγκριση με GridFTP

Ακολούθως, έγινε μια συγκριτική δοκιμή της επίδοσης του GridTorrent και του GridFTP σε τοπικό δίκτυα αλλά και σε δίκτυα ευρείας περιοχής. Ειδικότερα, έχουμε ένα κόμβο ο οποίος έχει το ρόλο του seed για ένα αρχείο 128MB καθώς και διαρκώς αυξανόμενους κόμβους που λαμβάνουν



Σχήμα 5.5: *Μέσο μέγεθος όγκου δεδομένων που αποστέλλεται από τους κόμβους - leechers σε διαφορετικά ποσοστά λανθασμένης μετάδοσης δεδομένων και μεγέθους block.*

το ρόλο του leecher και προσπαθούν ταυτόχρονα να μεταφέρουν το αρχείο. Η ίδια σειρά δοκιμών έχει γίνει και μια διαφορετικά μεγέθη αρχείων μέχρι 512MB και τα αποτελέσματα είναι ποσοτικά όμοια. Οι μετρικές που καταγράφονται και απεικονίζονται είναι ο ελάχιστος, μέσος και μέγιστος χρόνος για την μεταφορά του αρχείου ανά κόμβο leecher. Στις δοκιμές μας χρησιμοποιούμε ένα φυσικό μηχάνημα για το κόμβο seed και μέχρι 32 φυσικά μηχανήματα για τους κόμβους leecher. Για τις δοκιμές τοπικού δικτύου χρησιμοποιήθηκε ο εργαστηριακός εξοπλισμός διάταξης cluster με διασύνδεση gigabit ethernet. Για τις δοκιμές δικτύων ευρείας περιοχής χρησιμοποιήθηκε ο ίδιος αριθμός κόμβων στο PlanetLab [Peterson 02, pla]. Σε αυτό το περιβάλλον δοκιμών, υπάρχουν αρκετοί ισχυρά φορτωμένοι κόμβοι, γεωγραφικά κατανεμημένοι με πολλούς διαφορετικούς περιορισμούς σε θέματα καθυστέρησης και ταχύτητας της δικτυακής τους σύνδεσης. Αυτό κάνει το PlanetLab ένα περιβάλλον με μεγάλη ομοιότητα σε σχέση με το περιβάλλον που υπάρχει στο πραγματικό κόσμο, όπου οι κόμβοι μπορεί να βρίσκονται οπουδήποτε στο κόσμο ενώ ο εξοπλισμός που χρησιμοποιούν να είναι εξίσου απρόβλεπτος. Οι πληροφορίες σχετικά με την τοποθεσία του αρχείου, η λίστα των κόμβων που συμμετέχουν στην μεταφορά του, καθώς και άλλα μεταδεδομένα που αφορούν το αρχείο βρίσκονται αποθηκευμένα στο DRLS, το οποίο λειτουργεί σε ένα δίκτυο 30 κόμβων που προσομοιώνεται και υποστηρίζεται από ένα φυσικό μηχάνημα.

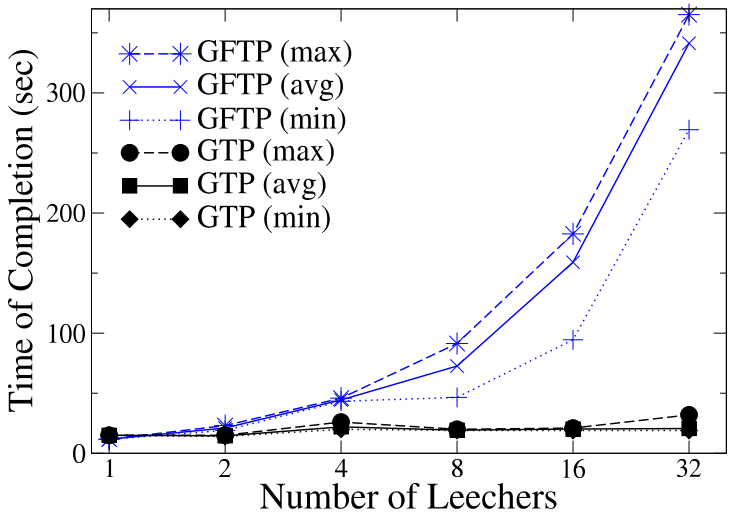
Στο 5.6 απεικονίζονται οι χρόνοι ολοκλήρωσης της μεταφοράς για τη δοκιμή του τοπικού δικτύου. Παρατηρούμε ότι το GridTorrent είναι έως και δέκα φορές πιο γρήγορο από το GridFTP στο σύνολο των μετρούμενων χρόνων. Αυτό συμβαίνει για τον μεγάλο αριθμό κόμβων leecher. Το GridFTP δεν μπορεί να επιβάλλει την συνεργασία μεταξύ των κόμβων, οπότε ένας εξυπηρετητής GridFTP πρέπει να εξυπηρετεί όλους τις αιτήσεις για μεταφορά αρχείων από τους πελάτες του, σειριακά. Θα περίμενε κανείς ότι το GridFTP δεν θα επηρεαζόταν από ένα φαινόμενο flash crowd,

όπου έχουμε απότομη αύξηση της ζήτησης, σε ένα περιβάλλον τοπικού δικτύου με υποδομή gigabit ethernet. Όμως τα αποτελέσματα των δοκιμών δείχνουν αντίθετη συμπεριφορά. Το GridTorrent παρουσιάζει αξιοσημείωτη διαφορά στην επίδοση και στις τρεις μετρικές καθώς μένουν σχεδόν ανεπηρέαστες από την αύξηση των αιτημάτων για εξυπηρέτηση. Το GridTorrent μπορεί εύκολα να διατήρηση τις επιδόσεις του κατά την επίδραση των φαινομένων flash crowd, καθώς προάγοντας την συνεργασία μεταξύ των κόμβων - leechers, μειώνει αποτελεσματικά το φορτίο στον κόμβο - seed, ο οποίος επιλέγει συγκεκριμένα κομμάτια του αρχείου για να μεταδώσει σε κάθε κόμβο - leecher. Στο 5.7 παρουσιάζεται η συνεργασία αυτή καταγράφοντας την κίνηση που ανταλλάσσουν αποκλειστικά μεταξύ τους οι κόμβοι leechers. Παρατηρούμε ότι όσο αυξάνει το πλήθος των κόμβων - leechers, η κίνηση αυξάνει, το οποίο σημαίνει ότι οι ίδιοι οι κόμβοι ανταλλάσσουν ολοένα και περισσότερα κομμάτια του αρχείου μεταξύ τους, λαμβάνοντας έτσι πιο ενεργό ρόλο στην μεταφορά του αρχείου και μειώνοντας το φορτίο του κόμβου - seed. Κατά μέσο όρο, κάθε κόμβος - leecher είναι υπεύθυνος για την αποστολή δεδομένων προς άλλους κόμβους τουλάχιστον ίσου μεγέθους με το αρχείο, και όχι παραπάνω από το διπλάσιο.

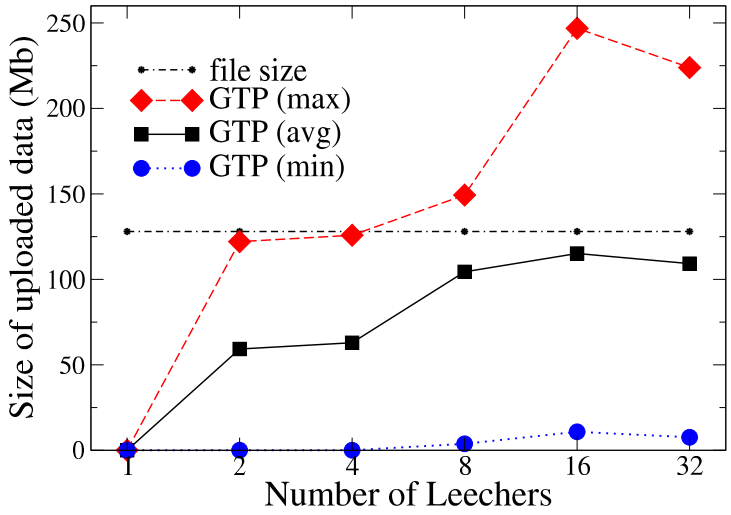
Το 5.8 απεικονίζει τα αποτελέσματα των δοκιμών σε περιβάλλον δικτύων ευρείας περιοχής. Από τα διαγράμματα γίνεται εύκολα αντιληπτό ότι το GridFTP δεν μπορεί να αντεπεξέλθει σε ολοένα και αυξανόμενα φορτία μεταφοράς δεδομένων σε ένα πραγματικό περιβάλλον. Οι ελάχιστοι χρόνοι ολοκλήρωσης του GridFTP παραμένουν σταθεροί και αρκετά χαμηλοί διότι πάντα υπάρχει ένας κόμβος - leecher που είναι αρκετά κοντά στον κόμβο - seed και μεταφέρει το αρχείο τοπικά σε αυτόν πολύ γρήγορα. Επιπλέον, παρατηρείται και μια μεγάλη διαφοροποίηση μεταξύ των ελάχιστων, μέγιστων και μέσων χρόνων ολοκλήρωσης της μεταφοράς ενός αρχείου με το GridFTP (πχ για 32 κόμβους - leechers ο τελευταίος λαμβάνει το αρχείο 30 φορές πιο αργά από τον πιο γρήγορο και περίπου 2 φορές πιο αργά από τον μέσο όρο. Αυτή η μεγάλη διακύμανση οφείλεται στο γεγονός ότι το πρωτόκολλο δεν μπορεί να αντεπεξέλθει με την ετερογενή φύση των πόρων - καθώς ένας μικρός αριθμός κοντινών κόμβων ολοκληρώνει την μεταφορά γρήγορα ενώ οι υπόλοιποι πιο απομακρυσμένοι κόμβοι επηρεάζονται πιο δραστικά.

Στο GridTorrent, οι πιο κοντινοί κόμβοι που ολοκληρώνουν την μεταφορά γρήγορα, βοηθούν και τους υπόλοιπους, αποστέλλοντας κομμάτια του αρχείου, λαμβάνοντας το ρόλο του κόμβου - seed. Έτσι μειώνεται ο συνολικός χρόνος ολοκλήρωσης της μεταφοράς του αρχείου, και μάλιστα μπορεί και κλιμακώνεται ομαλά σε σχέση με το πλήθος των κόμβων - leechers. Η μέθοδος αυτή είναι τρεις με δέκα φορές πιο γρήγορη κατά μέσο όρο και στην χειρότερη περίπτωση, ενώ παρουσιάζει πολύ μικρή διακύμανση μεταξύ των τριών μετρικών που καταγράφουμε. Στο 5.9 μπορούμε να δούμε το βαθμό συνεργασίας μεταξύ των κόμβων - leechers καθώς αυξάνεται ο αριθμός τους. Φαίνεται καθαρά μια μεγαλύτερη διακύμανση στα δεδομένα που ανταλλάχθηκαν μεταξύ των κόμβων - leechers σε σχέση με τις δοκιμές σε περιβάλλον τοπικού δικτύου. Αυτό δείχνει το μεγάλο βαθμό προσαρμοστικότητας που έχει το GridTorrent: Κοντινοί κόμβοι στο κόμβο - seed ολοκληρώνουν γρήγορα την μεταφορά και συνεισφέρουν άμεσα στους υπόλοιπους πιο πολύ από

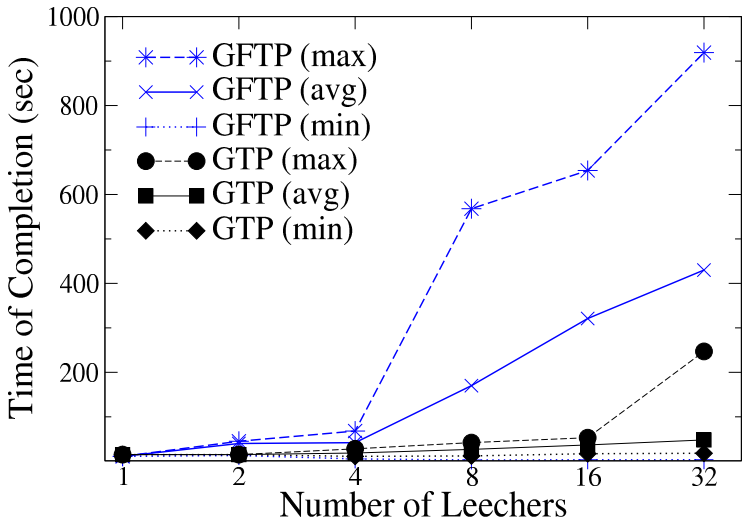
το μέσο όρο, ενώ οι πιο απομακρυσμένοι κόμβοι ολοκληρώνουν αργότερα τη μεταφορά και δεν έχουν απομείνει πολλοί κόμβοι για να βοηθήσουν. Οι δοκιμές με δίκτυα ευρείας περιοχής δείχνει με το καλύτερο τρόπο γιατί το πρωτόκολλο GridTorrent αποτελεί ένα αποδοτικό, γρήγορο και αποτελεσματικό τρόπο για την μεταφορά αρχείων που ξεπερνά κατά πολύ τις σημερινές τεχνικές.



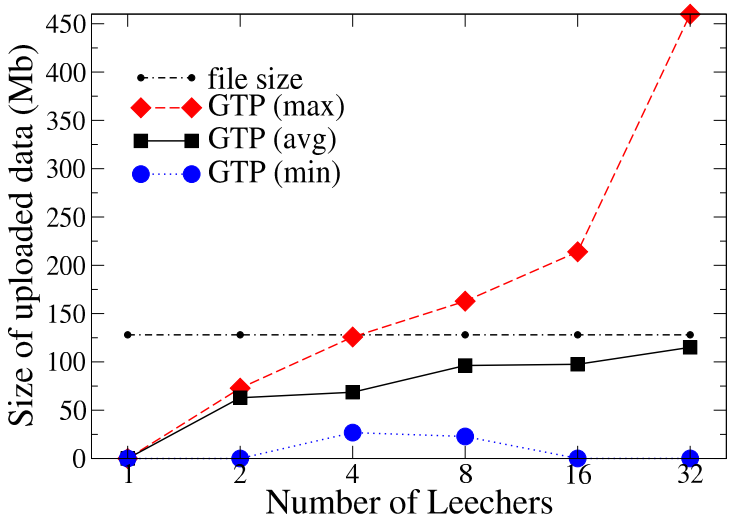
Σχήμα 5.6: *Ελάχιστος, μέγιστος και μέσος χρόνος μεταφορας αρχείου για GridFTP και GridTorrent για διαφορετικό πλήθος κόμβος - leecher σε περιβάλλον τοπικού δικτύου.*



Σχήμα 5.7: *Ελάχιστο, μέγιστο και μέσο μέγεθος όγκου δεδομένων που μεταδόθηκαν μεταξύ κόμβων - leecher για GridFTP και GridTorrent για διαφορετικό πλήθος κόμβος - leecher σε περιβάλλον τοπικού δικτύου.*



Σχήμα 5.8: *Ελάχιστος, μέγιστος και μέσος χρόνος μεταφορας αρχείου για GridFTP και GridTorrent για διαφορετικό πλήθος κόμβος - leecher σε περιβάλλον δικτύου ευρείας περιοχής.*



Σχήμα 5.9: *Ελάχιστο, μέγιστο και μέσο μέγεθος όγκου δεδομένων που μεταδόθηκαν μεταξύ κόμβων - leecher για GridFTP και GridTorrent για διαφορετικό πλήθος κόμβος - leecher σε περιβάλλον δικτύου ευρείας περιοχής.*

ΚΕΦΑΛΑΙΟ 6

Εφαρμογές

Τα αποτελέσματα της παρούσας εργασίας μπορούν να εφαρμοστούν σε ποικίλες περιπτώσεις όπου οι συνθήκες το επιτρέπουν, επιτυγχάνοντας μεγάλες βελτιώσεις στην λειτουργία των συστημάτων. Στην συνέχεια θα αναφερθούμε σε δύο αντιπροσωπευτικές περιπτώσεις, από τον επιστημονικό και τον επιχειρηματικό τομέα.

6.1 Large Hadron Collider (LHC)

Ένα από τα μεγαλύτερα επιστημονικά πειράματα του σύγχρονου κόσμου είναι ο μεγάλος επιταχυντής αδρονίων, που δημιουργεί συγκρούσεις σωματιδίων εξαιρετικά μεγάλης ενέργειας, οι οποίες στην συνέχεια δημιουργούν νέα σωματίδια που εξασθενούν με πολύπλοκο τρόπο καθώς περνούν από τα συστήματα των ανιχνευτών. Οι ανιχνευτές καταχωρούν το πέρασμα των σωματιδίων με ένα πολύ μεγάλο αριθμό σωματιδίων και τελικά μια ψηφιοποιημένη περίληψη της καταγραφής αποθηκεύεται σαν ένα γεγονός. Ο όγκος των δεδομένων που παράγονται από τον επιταχυντή είναι εκατοντάδες φορές μεγαλύτερος από προηγούμενα αντίστοιχα συστήματα. Αυτό θα συμβεί διότι σε αυτό το σύστημα υπάρχουν πολύ περισσότεροι ανιχνευτές και τα γεγονότα που ανιχνεύονται είναι πιο πολύπλοκα, ενώ ο ρυθμός ανίχνευσης είναι και πάλι πολύ μεγαλύτερος. Εκτός από τον όγκο των δεδομένων, μεγάλη είναι και η υπολογιστική ισχύς που χρειάζεται, δεδομένου ότι απασχολούνται 8000 ερευνητές σε 4 πειράματα (ALICE, ATLAS, CMS και LHCb),

οι οποίοι εκτελούν εκτεταμένα προγράμματα ανάλυσης δεδομένων. Για την εξυπηρέτηση των παραπάνω αναγκών, δηλαδή για την αποθήκευση και επεξεργασία των παραπάνω δεδομένων δημιουργήθηκε εδώ και μια δεκαετία το Υπολογιστικό και Αποθηκευτικό Πλέγμα του επιταχυντή αδρονίων (Worldwide Large Hardon Collider Computing Grid) [Bird 11]. Έχοντας περάσει πολυπληθείς αλλαγές σε τεχνολογίες τόσο στο υλικό όσο και στο λογισμικό, πλέον χρησιμοποιείται από χιλιάδες επιστήμονες σε όλο το κόσμο, οι οποίοι απολαμβάνουν άμεση πρόσβαση στα δεδομένα του επιταχυντή αλλά και σε υπολογιστική ισχύ για την επεξεργασία τους.

Η δομή του Πλέγματος όμως δεν είναι μονό-επίπεδη, αλλά έχει μια ιεραρχία τριών επιπέδων. Στο πρώτο επίπεδο βρίσκεται το Tier-0, στο οποίο ανήκει ένα υπολογιστικό κέντρο, του CERN. Όλα τα δεδομένα από τον επιταχυντή περνούν από το κεντρικό αυτό σημείο, ενώ διαθέτει υπολογιστική ισχύ λιγότερη από 20% της συνολικής. Συνδέεται με άλλα μεγάλα υπολογιστικά κέντρα, με δεσμευμένα τηλεπικοινωνιακά κυκλώματα ή απευθείας οπτικές ίνες. Στο δεύτερο επίπεδο βρίσκονται μερικές δεκάδες υπολογιστικά κέντρα, τα οποία βρίσκονται στο Καναδά, την Γαλλία, την Γερμανία, την Ιταλία, την Ολλανδία, τις χώρες της Βαλτικής, την Ισπανία, την Ταϊβάν, το Ηνωμένο Βασίλειο και τις ΗΠΑ. Τα κέντρα αυτά προσφέρουν δίκτυα διανομής της πληροφορίας, αλγόριθμους ανάλυσης, υπολογιστική ισχύ και αποθηκευτικό χώρο. Στο τρίτο και τελευταίο επίπεδο βρίσκονται περίπου μερικές εκατοντάδες κέντρα, διασκορπισμένα σε όλο τον κόσμο. Όλα μαζί τα κέντρα αυτά προσφέρουν το 50% της υπολογιστικής ισχύος που χρειάζεται για την επεξεργασία των δεδομένων. Περιληπτικά, οι συνολικοί πόροι ανέρχονται σε εκατοντάδες χιλιάδες επεξεργαστές μεταξύ 170 κέντρων σε 34 χώρες.

Τα παραπάνω στοιχεία περιγράφουν ένα κατανεμημένο σύστημα μεγάλης κλίμακας, στο οποίο μπορούν να συνυπάρξουν ένας τεράστιος αριθμός ανεξάρτητων και διαφορετικών υπολογιστικών και αποθηκευτικών πόρων, οι οποίοι ενοποιούνται σε μία υπηρεσιοστρεφή αρχιτεκτονική λογισμικού. Το σύστημα αυτό αναπτύσσεται σε παγκόσμια κλίμακα και αποτελεί ιδανικό παράδειγμα εφαρμογής της προτεινόμενης κατανεμημένης αρχιτεκτονικής για την μεταφορά και διαχείριση δεδομένων. Η υιοθέτηση μιας κεντρικοποιημένης σχεδίασης, θα είχε ως αποτέλεσμα μειωμένες επιδόσεις και κεντρικά σημεία βλάβης με σημαντικές επιπτώσεις στην διαθεσιμότητα του συστήματος. Οι κεντρικοποιημένες υπηρεσίες δεν μπορούν να κλιμακώσουν σε μεγάλο αριθμό ταυτόχρονων χρηστών, ούτε να διατηρήσουν ένα υψηλό αριθμό ανανέωσης σε ένα δυναμικό περιβάλλον όπως αυτό του Πλέγματος.

6.2 Grid enabled access to rich media content - (GREDIA)

6.2.1 Γενικά

Το GREDIA είναι ένα ευρωπαϊκό ερευνητικό πρόγραμμα, μέρος του IST 6th Framework Programme (FP6-34363). Το έργο στόχευσε στην δημιουργία μιας πλατφόρμας ανάπτυξης εφαρμογών Πλέγματος, που προσφέρει υποστήριξη για την υλοποίηση εμπορικών εφαρμογών μέσω γραφικού περιβάλλοντος. Η πλατφόρμα συνδυάζει παλαιές και νέες τεχνολογίες για την άμεση ενεργοποίηση επιχειρηματικών υπηρεσιών για την πρόσβαση και διαμοιρασμό μεγάλου όγκου σχολιασμένου πολυμεσικού υλικού. Επίσης, το έργο διευκολύνει κινητές συσκευές να εκμεταλλευτούν τις τεχνολογίες πλέγματος με διάφανο τρόπο, προσφέροντας πρόσβαση σε αυτό το κατανεμημένο και ογκωδέστατο πολυμεσικό υλικό. Τα αποτελέσματα της πλατφόρμας επιδεικνύονται με δύο πιλοτικές εφαρμογές, μια στο τραπεζικό τομέα και μια στο δημοσιογραφικό τομέα.

Ο σχεδιασμός και η υλοποίηση ενός τέτοιου συστήματος αντιμετωπίζει πολλά προβλήματα. Το πιο κρίσιμο είναι η ανάπτυξη μιας κατανεμημένης αρχιτεκτονικής για την διαχείριση μεγάλου όγκου δεδομένων σε γεωγραφικά διεσπαρμένους πόρους. Η απόδοση ενός τέτοιου συστήματος δεν θα πρέπει να φθίνει, καθώς ο όγκος των αποθηκευμένων δεδομένων και το πλήθος των εκτελούμενων λειτουργιών αυξάνει. Το σύστημα πρέπει να είναι κλιμακώσιμο και να προσφέρει αξιόπιστες υπηρεσίες. Σχετικά με την δημοσίευση και την αναζήτηση του περιεχομένου, οι χρήστες του συστήματος θα πρέπει να μπορούν να αναζητήσουν με βάση μια ακριβή τιμή, αλλά και με βάση κάποιο διάστημα τιμών. Η απάντηση στην αναζήτηση κάποιου διαστήματος τιμών θα αφορά ένα σύνολο δεδομένων που έχουν ένα κοινό χαρακτηριστικό. Για παράδειγμα, ένας χρήστης μπορεί να αναζητήσει κάποιο περιεχόμενο με βάση συγγραφέα, τίτλο, ενώ η ημερομηνία δημιουργίας του να είναι μέσα σε κάποιο χρονικό διάστημα. Όμως, η απάντηση σε τέτοιου είδους ερωτήματα είναι ιδιαίτερα δύσκολη, ειδικά όταν δεν υπάρχει μια κεντρική δομή ευρετηρίου. Σε μια κατανεμημένη αρχιτεκτονική, πρέπει να προβλεφθούν μηχανισμοί, ώστε τέτοιες αναζητήσεις να μην διατρέχουν όλο το σύστημα, αλλά μόνο ένα μικρό αριθμό κόμβων. Τέλος, η ανάπτυξη ενός ενιαίου μηχανισμού για την πρόσβαση σε ετερογενείς πόρους και την μεταφορά μεγάλων ποσοτήτων δεδομένων είναι μια μεγάλη πρόκληση. Θα πρέπει να ληφθούν υπόψη οι περιορισμοί στο εύρος ζώνης των καναλιών επικοινωνίας και ειδικά όταν ένας κόμβος πολλαπλές ταυτόχρονες αιτήσεις πρόσβασης στο ίδιο περιεχόμενο.

6.2.2 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος ασχολείται με τη διαχείριση δεδομένων σε ένα κατανεμημένο περιβάλλον το οποίο αποτελείται από πόρους διαφορετικών εικονικών οργανισμών (Virtual Organizations). Ένα κύριο χαρακτηριστικό αυτού του περιβάλλοντος είναι οι ετερογενείς πόροι όσον αφορά την υπολογιστική ισχύ, τον αποθηκευτικό χώρο και το εύρος ζώνης του καναλιού

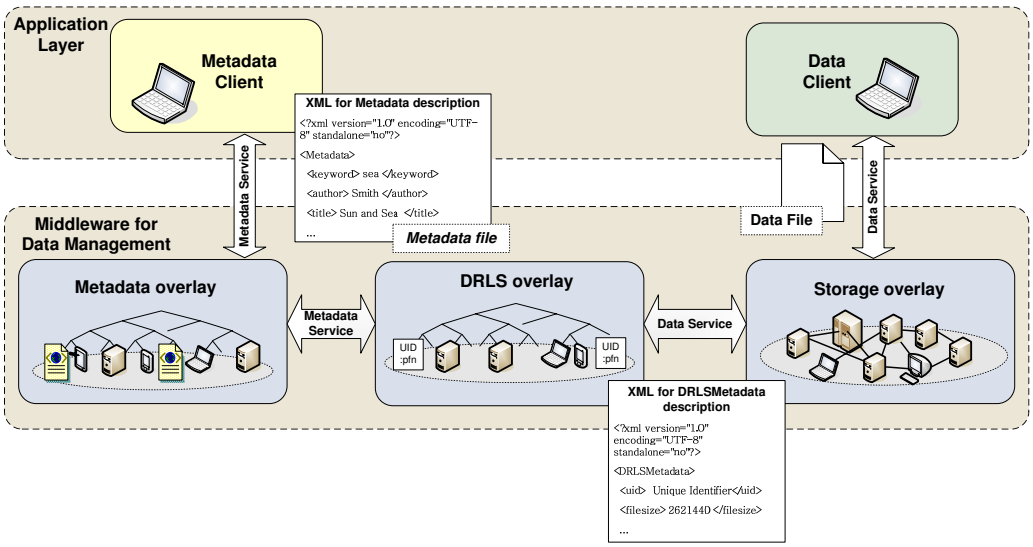
επικοινωνιών. Πόροι με διαφορετικές δυνατότητες, για παράδειγμα φορητοί υπολογιστές, υπολογιστές γραφείου, εξυπηρετητές ή ακόμα και κινητά τηλέφωνα μπορούν να ενταχθούν στο σύστημα. Επιπλέον, θεωρούμε ότι μερικές συσκευές μπορούν να παραμείνουν εκτός συστήματος για μεγάλα χρονικά διαστήματα. Γι' αυτό ο σχεδιασμός του συστήματος πρέπει να επιτρέπει την άφιξη και την αναχώρηση των κόμβων μέχρι ενός ορίου.

Η αρχιτεκτονική αποτελείται από τα ακόλουθα επίπεδα:

- Το *επίπεδο Αποθήκευσης*, όπου το πολυμεσικό υλικό αποθηκεύεται και υλοποιούνται οι μηχανισμοί μεταφοράς δεδομένων. Οι κόμβοι στο επίπεδο αυτό δρουν σαν εξυπηρετητές αρχείων και παρέχουν την υπηρεσία αυτή συνεχώς και αξιόπιστα.

- Το *επίπεδο Μεταδεδομένων*, όπου τα μεταδεδομένα κάθε ξεχωριστού υλικού αποθηκεύεται και ανακτάται. Το επίπεδο αυτό παρέχει ένα ισχυρό μηχανισμό αναζήτησης, ο οποίος υποστηρίζει αναζήτηση ακριβούς τιμής αλλά και πεδίου τιμών.

- Το *επίπεδο Υπηρεσίας Αναζήτησης Κατανεμημένων Αντιγράφων - YAKA* (Distributed Replica Location Service - DRLS), όπου διατηρείται μια λίστα με απαραίτητα στοιχεία για την υπηρεσία μεταφοράς δεδομένων. Το επίπεδο αυτό συσχετίζει επίσης, τα μεταδεδομένα με τα κυρίως δεδομένα κάθε πολυμεσικού υλικού. Υλοποιεί ένα κατανεμημένο κατάλογο, ο οποίος περιέχει αντιστοιχήσεις μεταξύ ενός μοναδικού αναγνωριστικού του υλικού και τις τοποθεσίες των πραγματικών αρχείων.



Σχήμα 6.1: *Η αρχιτεκτονική πλέγματος του GREDIA*

Κάθε κόμβος του συστήματος δύναται να υποστηρίξει παραπάνω από ένα επίπεδο, το οποίο καθορίζεται από τους διαθέσιμους πόρους του σε αποθηκευτικό χώρο, υπολογιστική ισχύ και χωρητικότητα του καναλιού επικοινωνίας. Η υλοποίηση των επιπέδων Μεταδεδομένων και YAKA γίνεται σύμφωνα με το πρωτόκολλο Kademlia. Οι κόμβοι με κοντινά αναγνωριστικά στο χώρο διευθύνσεων, θεωρούνται γείτονες χωρίς κατ' ανάγκην να συμβαίνει το ίδιο και στο φυσικό χώρο. Όπως φαίνεται στο 6.1 κάθε επίπεδο αλληλεπιδρά με τους αντίστοιχους πελάτες/χρήστες/υπηρεσίες μέσω προτυποποιημένων υπηρεσιών Πλέγματος. Τέτοιες είναι η Υπηρεσία Δεδομένων και η Υπηρεσία Μεταδεδομένων που αναπτύσσονται στους κόμβους των επιπέδων Αποθήκευσης και Μεταδεδομένων αντίστοιχα.

Πολλαπλά επίπεδα εισάγουν επεκτασιμότητα και δυναμική στο σύστημα. Οι υλοποιημένες υπηρεσίες κάθε επιπέδου είναι αυτόνομες οντότητες. Ο στόχος της σχεδίασης αυτής είναι η δημιουργία φορητών και ευέλικτων υπηρεσιών, οι οποίες μπορούν να χρησιμοποιηθούν για την επίτευξη ένα συγκεκριμένο σκοπό η να ενσωματωθούν σε ένα γενικότερο σύστημα. Η αλληλεπίδραση με άλλες υπηρεσίες γίνεται μέσω σαφώς προδιαγεγραμμένες διεπαφές. Η εκτέλεση αυτών των υπηρεσιών είναι διαφανής στο χρήστη, ο οποίος το μόνο που χρειάζεται να γνωρίζει είναι η διεύθυνση της υπηρεσίας και όχι τις λεπτομέρειες υλοποίησής της.

Τα δεδομένα που χειρίζεται το σύστημα μπορούν να διαχωριστούν σε δύο κατηγορίες, τα αρχεία πολυμεσικού υλικού και τα αρχεία μεταδεδομένων τους. Ένα αρχείο πολυμεσικού υλικού περιέχει την κυρίως πληροφορία για το υλικό και αποθηκεύεται και αντιγράφεται στο επίπεδο Αποθήκευσης. Κάθε τέτοιο αρχείο περιγράφεται με παραμέτρους ενός προσδιορισμένου σχήματος μεταδεδομένων και οι αντίστοιχες τιμές των παραμέτρων βρίσκονται στο αρχείο μεταδεδομένων. Αυτά βρίσκονται αποθηκευμένα στο επίπεδο Μεταδεδομένων. Η σύνδεση μεταξύ των δύο επιπέδων γίνεται με ένα μοναδικό αναγνωριστικό (UID) για κάθε αρχείο. Η τιμή του αναγνωριστικού προκύπτει από την εφαρμογή μιας συνάρτηση κατακερματισμού στο αρχείο πολυμεσικού υλικού, ενώ ταυτόχρονα φυλάσσεται και στο αρχείο μεταδεδομένων. Το επίπεδο YAKA κατέχει ένα εξίσου σημαντικό ρόλο στο σύστημα, καθώς γνωρίζει όλες τις φυσικές τοποθεσίες, τις διευθύνσεις που βρίσκονται τα δεδομένα για κάθε αναγνωριστικό (UID). Με δεδομένο ένα αρχείο μεταδεδομένων, θα χρειαστεί μια αναζήτηση για το αντίστοιχο UID στην YAKA ώστε να βρεθούν τα αρχείο δεδομένων. Στην YAKA αποθηκεύονται ζεύγη κλειδιού-τιμής, όπου το κλειδί αντιπροσωπεύει το UID του κάθε αρχείου και η τιμή είναι ένα αρχείο XML το οποίο περιέχει τις διευθύνσεις για τις τοποθεσίες που βρίσκεται αποθηκευμένο το αρχείο και τα αντίγραφά του.

ΚΕΦΑΛΑΙΟ 7

Επίλογος

Ένα από τα πιο διαδεδομένα κατανεμημένα συστήματα μεγάλης κλίμακας είναι το Πλέγμα, όπου μπορούν να συνυπάρξουν αρκετός μεγάλος αριθμός υπολογιστικών και αποθηκευτικών πόρων, της τάξεως των εκατομμυρίων. Οι υπηρεσίες που προσφέρονται από την υποδομή Πλέγματος θα πρέπει να είναι ικανές να αντεπεξέλθουν σε κλιμάκωση φορτίου αρκετά μεγάλη, ώστε το Πλέγμα να γίνει διαθέσιμο σε παγκόσμια κλίμακα και να απευθυνθεί στον απλό χρήστη. Μια κρίσιμη υπηρεσία στο επίπεδο δεδομένων του Πλέγματος είναι η διαχείριση των δεδομένων και πιο συγκεκριμένα η υπηρεσία αναζήτησης αντιγράφων (Replica Location Service). Η συγκεκριμένη υπηρεσία διατηρεί αποθηκευμένη την πληροφορία των διάφορων τοποθεσιών που φυλάσσεται ένα αρχείο αλλά και τις τοποθεσίες των αντιγράφων του, ώστε να μπορέσει να γίνει η ανάκτησή του από ένα απλό χρήστη. Στις υφιστάμενες αρχιτεκτονικές και υλοποιήσεις υπολογιστικού πλέγματος η δομή της συγκεκριμένης υπηρεσίας είναι κεντρική. Η συγκεκριμένη δομή έχει βασικά μειονεκτήματα, όπως αυτό του μοναδικού σημείου βλάβης (Single point of failure – SPOF) και της μη κλιμάκωσης σε περίπτωση αύξησης του φόρτου εργασίας (scalability), με αποτέλεσμα να αποτελεί τροχοπέδη στην περαιτέρω ανάπτυξη του Πλέγματος. Στην παρούσα διατριβή προτείνεται μια νέα δομή της υπηρεσίας αυτής χρησιμοποιώντας τις δημοφιλείς τεχνικές των δικτύων ομότιμων κόμβων (peer-to-peer), οι οποίες προσφέρουν λύσεις για τα παραπάνω προβλήματα, δηλαδή του κεντρικού σημείου βλάβης και της κλιμάκωσης του φόρτου εργασίας. Εκτός από την διαχείριση των δεδομένων, ιδιαίτερη κρίσιμη στη λειτουργία του Πλέγματος είναι και η μεταφορά των δεδομένων από κόμβο σε κόμβο. Η υφιστάμενη αρχιτεκτονική του προτείνει λύσεις κεντρικές και βασισμένες σε προϋπάρχοντα πρωτόκολλα μεταφοράς δεδομένων, όπως το File Transfer

Protocol. Πάνω σε αυτά προσθέτονται επιπλέον χαρακτηριστικά αρκετά σημαντικά για τους χρήστες του Πλέγματος όπως η πιστοποίηση χρηστών και η διαχείριση μεταφορών δεδομένων από τρίτους, εκτός δηλαδή των αποστολέα και παραλήπτη. Η προσέγγιση αυτή παρουσιάζει αρκετά μειονεκτήματα. Ένα βασικό είναι ότι ένα αποθετήριο δεδομένων αποτελεί κεντρικό σημείο βλάβης για την μεταφορά ενός αρχείου αλλά στενωπός όσον αφορά την μεταφορά σε πολλούς άλλους κόμβους σε περίπτωση που είναι αρκετά δημοφιλές.

Τα παραπάνω σχεδιαστικά προβλήματα που συναντά κανείς στην κεντρικοποιημένη αρχιτεκτονική των βασικών υπηρεσιών που απαρτίζουν τέτοια συστήματα, αποτελούν τροχοπέδη στην ομαλή λειτουργία και ανάπτυξή τους. Μετά από μελέτη στη διεθνή βιβλιογραφία για τις σύγχρονες μεθόδους διαχείρισης και μεταφοράς δεδομένων και πιο συγκεκριμένα στα πεδία των συστημάτων δομημένων Δικτύων Ομότιμων Κόμβων με Κατανεμημένους Πίνακες Κατακερματισμού, καθώς και των συστημάτων που χρησιμοποιούν κίνητρα συνεργασίας πάνω σε αδόμητα Δίκτυα Ομότιμων Κόμβων, καταλήξαμε σε μια καινοτόμο αρχιτεκτονική με αποδοτικές τεχνικές μεταφοράς και διαχείρισης δεδομένων, οι οποίες προσφέρουν τα πλεονεκτήματα της κλιμακωσιμότητας, της αποφυγής του κεντρικού σημείου βλάβης και της ανοχής – βέλτιστης διαχείρισης των πόρων σε απότομες μεταβολές του φορτίου. Η νέα αυτή αρχιτεκτονική αποτελείται από δύο μέρη τα οποία συνεργάζονται για την αποδοτική διαχείριση δεδομένων: (α) την Κατανεμημένη Υπηρεσία Διαχείρισης Αντιγράφων (Distributed Replica Location Service - DRLS) υπεύθυνη για την φύλαξη των φυσικών τοποθεσιών αποθήκευσης κάθε αρχείου και (β) το GridTorrent επιφορτισμένο με την διαχείριση των ανταλλαγών δεδομένων με αυτόματους μηχανισμούς βελτιστοποίησης. Το DRLS οργανώνει τους κόμβους του συστήματος με ένα Κατανεμημένο Πίνακα Κατακερματισμού (Distributed Hash Table - DHT) και διανέμει την πληροφορία σε όλους τους κόμβους. Το μοναδικό χαρακτηριστικό του DRLS είναι ότι εκτός από την αποκεντρικοποίηση της υπηρεσίας και την κλιμακωσιμότητα που της προσφέρει, υποστηρίζει εγγενώς την ανανέωση της πληροφορίας σε κάθε κόμβο που συμμετέχει στο DHT. Δεδομένου, ότι σε πολλές δυναμικές εφαρμογές τα δεδομένα αλλάζουν συνεχώς, το πρωτόκολλο στο οποίο βασίζεται το DRLS παρουσιάζει ανοχή σε Βυζαντινές συνθήκες σφαλμάτων και εγγυάται συνέπεια. Το GridTorrent είναι ένα πρωτόκολλο εμπνευσμένο από το BitTorrent, που εστιάζει στην βελτιστοποίηση της μεταφοράς δεδομένων σε πραγματικό χρόνο, χωρίς να παραβιάζονται οι αρχές ασφάλειας του Πλέγματος. Η συνεργατική φύση του πρωτοκόλλου, επιτρέπει τη διατήρηση χαμηλής απόκρισης και υψηλής χρησιμοποίησης του δικτύου, ακόμα και σε συνθήκες υψηλού φορτίου. Επιτρέπει μεταφορές δεδομένων από πολλαπλούς αποστολείς σε πολλαπλούς παραλήπτες και μεγιστοποιεί την απόδοση με την ανταλλαγή κομματιών του αρχείου μεταξύ όλων των συμμετεχόντων. Πολύ σημαντικό χαρακτηριστικό της προτεινόμενης αρχιτεκτονικής είναι ότι έχει σχεδιαστεί, ώστε να εκμεταλλευτεί υφιστάμενα και ευρέως χρησιμοποιούμενα πρότυπα στο χώρο του Πλέγματος, ώστε να διατηρεί την συμβατότητα με την υφιστάμενη αρχιτεκτονική και τις αντίστοιχες υλοποιήσεις. Τέλος, για την επαλήθευση των αποτελεσμάτων της εργασίας μας, έχει υλοποιηθεί ένα πρωτότυπο της αρχιτεκτονικής και έχουν

γίνει αναλυτικά πειράματα του συστήματος τόσο σε περιβάλλοντα τοπικού δικτύου, όσο και σε περιβάλλοντα μεγάλης κλίμακας και υψηλής δυναμικότητας.

ΚΕΦΑΛΑΙΟ 8

Δημοσιεύσεις

Κεφάλαια Βιβλίων

- *Replica management services on the Grid: Evolving from a centralized design to a fully distributed, scalable and fault-tolerant peer-to-peer infrastructure.* από τους Αντώνιο Χαζαπη, Αντώνιο Ζήσιμο, Νεκτάριο Κοζύρη και Παναγιώτη Τσανάκα στο βιβλίο Grid Technologies: Emerging from Distributed Architectures to Virtual Organizations, Series: Advances in Management Information, από τον εκδοτικό οίκο WIT Press, Τόμος 5, ISBN: 1-84564-055-1, Έκδοση: 2006, Σελίδες: 107-138, .

Διεθνή Περιοδικά

- *A Grid middleware for data management exploiting Peer-to-Peer techniques* από τους Αθανασία Ασίκη, Αικατερίνη Δόκα, Ιωάννη Κωνσταντίνου, Αντώνιο Ζήσιμο, Δημήτριο Τσουμάκο, Νεκτάριο Κοζύρη και Παναγιώτη Τσανάκα στο περιοδικό «FUTURE GENERATION COMPUTER SYSTEMS», από τον εκδοτικό οίκο Springer, 25 (2009) 426-435.

Διεθνή Συνέδρια

- *Optimizing Data Management in Grid Environments* από τους Αντώνιο Ζήσιμο, Αικατερίνη Δόκα, Αντώνιο Χαζάπη, Δημήτριο Τσουμάκο, Νεκτάριο Κοζύρη, στα πρακτικά του

συνεδρίου «11th International Symposium on Distributed Objects, Middleware, and Applications (DOA'09)», που πραγματοποιήθηκε στη Vilamoura, Πορτογαλία, 01-03 Νοέμβρη 2009

- *Gredia Middleware Architecture* από τους Ιωάννη Κωνσταντίνου, Αικατερίνη Δόκα, Αθανασία Ασίκη, Αντώνιο Ζήσιμο, Νεκτάριο Κοζύρη, στα πρακτικά του «Cracow 2007 Grid Workshop (CGW'07).», που πραγματοποιήθηκε στην Κρακοβία, Πολωνία, 16-17 Οκτώβρη 2007

- *A Distributed Architecture for Multi-Dimensional Indexing and Data Retrieval in Grid Environments* από τους Αθανασία Ασίκη, Αικατερίνη Δόκα, Ιωάννη Κωνσταντίνου, Αντώνιο Ζήσιμο, Νεκτάριο Κοζύρη, στα πρακτικά του «Cracow 2007 Grid Workshop (CGW'07)», που πραγματοποιήθηκε στην Κρακοβία, Πολωνία, 16-17 Οκτώβρη 2007

- *GridTorrent: Optimizing data transfers in the Grid with collaborative sharing.* από τους Αντώνιο Ζήσιμο, Αικατερίνη Δόκα, Αντώνιο Χαζάπη, Νεκτάριο Κοζύρη, στα πρακτικά του συνεδρίου «11th Panhellenic Conference on Informatics (PCI'07)», που πραγματοποιήθηκε στην Πάτρα, τον Μάιο του 2007

- *A peer-to-peer replica management service for high-throughput Grids* από τους Αντώνιο Χαζάπη, Αντώνιο Ζήσιμο, Νεκτάριο Κοζύρη στα πρακτικά του συνεδρίου «34th International Conference on Parallel Processing (ICPP05)», που πραγματοποιήθηκε στο Όσλο, Νορβηγία, τον Ιούνιο του 2005

Βιβλιογραφία

[Adamic 01] L. A. Adamic, R. M. Lukose, A. R. Puniyani & B. A. Huberman. *Search in power law networks*. Physical Review E64, vol. 64, pages 46135–46143, 2001.

[Allcock 02] Bill Allcock, Joe Bester, John Bresnahan, Ann L. Chervenak, Ian Foster, Carl Kesselman, Sam Meder, Veronika Nefedova, Darcy Quesnel & Steven Tuecke. *Data management and transfer in high-performance computational grid environments*. Parallel Computing, vol. 28, no. 5, pages 749–771, 2002.

[Allcock 05] William Allcock, John Bresnahan, Rajkumar Kettimithu, Michael Link, Catalin Dumitresku, Ioan Raicu & Ian Foster. *The Globus Striped GridFTP Framework and Server*. In In Proceedings of the ACM/IEEE Conference on Supercomputing, SC'05, 2005.

[Anderson 97] D. Anderson, S. Bowyer, J. Cobb, D. Gedye, W. T. Sullivan & D. Werthimer. *New Major SETI Project Based on Project Serendip Data and 100,000 Personal Computers*. In Astronomical and Biochemical Origins and the Search for Life in the Universe, Proc. of the Fifth Intl. Conf. on Bioastronomy, 1997.

[Anderson 04] D. Anderson. *BOINC: A System for Public-Resource Computing and Storage*. In In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, 2004.

[Bird 11] Ian Bird. *Computing for the Large Hadron Collider*. Annual Review of Nuclear and Particle Science, vol. 61, no. 1, pages 99–118, 2011.

[Bloom 70] Burton H. Bloom. *Space/time trade-offs in hash coding with allowable errors*. Commun. ACM, vol. 13, no. 7, pages 422–426, July 1970.

[C. Plaxton] R. Rajaraman C. Plaxton & A. Richa.

[Cai 04] Min Cai, Ann Chervenak & Martin Frank. *A Peer-to-Peer Replica Location Service Based on a Distributed Hash Table*. In Proceedings of the 2004 ACM/IEEE conference on Supercomputing, Pittsburgh, PA, Nov 2004.

[Chawathe 03] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham & Scott Shenker. *Making gnutella-like P2P systems scalable*. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, pages 407–418, New York, NY, USA, 2003. ACM.

[Chazapis 05] Antony Chazapis, Antonis Zissimos & Nectarios Koziris. *A peer-to-peer replica management service for high-throughput Grids*. In Proceedings of the 2005 International Conference on Parallel Processing (ICPP05), 2005.

[Chazapis 06] Antony Chazapis, Antonis Zissimos, Nectarios Koziris & Panayiotis Tsanakas. Grid technologies: Emerging from distributed architectures to virtual organizations, chapitre Replica management services on the Grid: Evolving from a centralized design to a fully distributed, scalable and fault-tolerant peer-to-peer infrastructure., pages 107–138. Advances in Management Information. WIT Press, 2006.

[Chazapis 07] Antony Chazapis & Nectarios Koziris. *XOROS: A mutable Distributed Hash Table*. In Proceedings of the 5th International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2007), 2007.

[Chervenak 00] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury & S. Tuecke. *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*. Journal of Network and Computer Applications, 2000.

[Chervenak 02] Ann Chervenak, Ewa Deelman, Ian Foster, Leanne Guy, Wolfgang Hoschek, Adriana Iamnitchi, Carl Kesselman, Peter Kunszt, Matei Ripeanu, Bob Schwartzkopf, Heinz Stockinger, Kurt Stockinger & Brian Tierney. *Giggle: a*

framework for constructing scalable replica location services. In Proceedings of the 2002 ACM/IEEE conference on Supercomputing, pages 1–17. IEEE Computer Society Press, 2002.

[Chervenak 04] A.L. Chervenak, N. Palavalli, S. Bharathi, C. Kesselman & R. Schwartzkopf. *Performance and Scalability of a Replica Location Service*. In Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13'04), Honolulu, HI, Jun 2004.

[Clarke 01] Ian Clarke, Oskar Sandberg, Brandon Wiley & Theodore W. Hong. *Freenet: A Distributed Anonymous Information Storage and Retrieval System*. In International Workshop on Designing Privacy Enhancing Technologies: Design issues in anonimity and unobservability, pages 46–66. Springer-Verlag New York, Inc., 2001.

[Clarke 02] Ian Clarke, Scott G. Miller, Theodore W. Hong, Oskar Sandberg & Brandon Wiley. *Protecting Free Expression Online with Freenet*. IEEE Internet Computing, vol. 6, pages 40–49, January 2002.

[cms] *Compact Muon Solenoid (CMS)*. http://cms.web.cern.ch/.

[D. R. Karger 97] F. T. Leighton R. Panigrahy M. S. Levine D. R. Karger E. Lehman & D. Lewin. *Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web*. In Proceedings of the ACM Symposium on Theory of Computing, page 654 663, May 1997.

[dis] *Distributed.net, RSA Labs 64bit RC5 Encryption Challenge*. http://www.distributed.net.

[Fitzgerald 01] Steven Fitzgerald. *Grid Information Services for Distributed Resource Sharing*. In Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC'01). IEEE Computer Society, 2001.

[Foster 97] I. Foster & C. Kesselman. *Globus: A Metacomputing Infrastructure Toolkit*. International Journal of Supercomputer Applications, vol. 11, no. 2, pages 115–128, 1997.

[Foster 98] Ian Foster, Carl Kesselman, Gene Tsudik & Steven Tuecke. *A security architecture for computational grids*. In Proceedings of the 5th ACM conference on Computer and communications security, pages 83–92, New York, NY, USA, 1998. ACM Press.

[Foster 99] I. Foster & C. Kesselman. The grid: Blueprint for a new computing infrastructure. Morgan-Kaufmann, 1999.

[Foster 01] Ian Foster, Carl Kesselman & Steven Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. International Journal of Supercomputer Applications, 2001.

[Foster 03] Ian Foster & Adriana Iamnitchi. *On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing*. In Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), Berkeley, CA, Feb 2003.

[Foster 05] Ian Foster & Steven Tuecke. *Describing the Elephant: The Different Faces of IT as Service*. Queue, vol. 3, no. 6, pages 26–29, July 2005.

[Ganesan 03] Prasanna Ganesan, Q. Sun & H. Garcia-Molina. *YAPPERS: a peer-to-peer lookup service over arbitrary topology*. In INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, volume 2, pages 1250 – 1260 vol.2, march-3 april 2003.

[Glo] *The official site of Globus Toolkit*. http://globus.org/toolkit.

[Gnu 01] *Gnutella development forum, the gnutella v0.6 protocol*, 2001. http://groups.yahoo.com/group/the gdf/files/.

[Gnu 02a] *Gnucleus, the gnutella web caching system*, 2002. http://www.gnucleus.net/gwebcache/.

[Gnu 02b] *Gnutella ultrapeers*, 2002. http://rfc-gnutella.sourceforge.net/Proposals/Ultrapeer/Ultrapeers.htm/.

[Hazel 02] Steven Hazel & Brandon Wiley. *Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems*. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Cambridge, MA, Mar 2002.

[Hoschek 00] Wolfgang Hoschek, Javier Jaen-Martinez, Asad Samar, Heinz Stockinger & Kurt Stockinger. *Data Management in an International Data Grid Project*. In Proceedings of the 1st International Workshop on Grid Computing (Grid 2000), Bangalore, India, Dec 2000.

[Iamnitchi 02] Adriana Iamnitchi, Ian Foster & Daniel C. Nurmi. *A Peer-to-Peer Approach to Resource Location in Grid Environments*. In Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11'02), Edinburgh, UK, Jul 2002.

[Kaplan] Ali Kaplan, Geoffrey Fox & Gregor von Laszewski. *GridTorrent Framework: A High-performance Data Transfer and Data Sharing Framework for Scientific Computing*. In Proceedings of GCE07, Reno, Nevada, 2007.

[L. Breslau] L. Fan G. Phillips L. Breslau P. Cao & S. Shenker.

[Lamport 82] Leslie Lamport, Robert Shostak & Marshall Pease. *The Byzantine Generals Problem*. ACM Transactions on Programming Languages and Systems, vol. 4, pages 382–401, 1982.

[lcg] *The LHC Computing Grid Project (LCG)*. http://lcg.web.cern.ch/LCG/.

[Lv 02] Qin Lv, Sylvia Ratnasamy & Scott Shenker. *Can Heterogeneity Make Gnutella Scalable?* In In Proceedings of the first International Workshop on Peer-to-Peer Systems, pages 94–103, 2002.

[Malkhi 99] Dahlia Malkhi, Yishay Mansour & Michael K. Reiter. *On Diffusing Updates in a Byzantine Environment*. In In Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, pages 134–143. IEEE Computer Society, 1999.

[Mitzenmacher 02] Michael Mitzenmacher. *Compressed bloom filters*. IEEE/ACM Trans. Netw., vol. 10, no. 5, pages 604–612, October 2002.

[Nap] *Napster*. http://www.napster.com/.

[Obj] *Objectivity Inc.* http://www.objectivity.com.

[Ope] *Open Grid Forum*. http://www.gridforum.org/.

[Peterson 02] Larry Peterson, Tom Anderson, David Culler & Timothy Roscoe. *A Blueprint for Introducing Disruptive Technology into the Internet*. In Proceedings of HotNets–I, Princeton, NJ, Oct 2002.

[pla] *PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services*. http://www.planet-lab.org/.

[Postel 85] J. Postel & J. Reynolds. *RFC 959 - File Transfer Protocol*, 1985.

[Ratnasamy 01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp & Scott Shenker. *A scalable content-addressable network*. SIGCOMM Comput. Commun. Rev., vol. 31, pages 161–172, August 2001.

[Rekhter 93] Y. Rekhter & T. Li. *An architecture for ip address allocation with cidr, IETF Internet draft RFC1518*, 1993. http://tools.ietf.org/html/rfc1518/.

[Ripeanu 02a] M. Ripeanu & I. Foster. *Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems*. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Cambridge, MA, Mar 2002.

[Ripeanu 02b] Matei Ripeanu & Ian Foster. *A Decentralized, Adaptive, Replica Location Service*. In Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11'02), Edinburgh, UK, Jul 2002.

[Rowstron] A. Rowstron & P. Druschel.

[Samar 01] Asad Samar & Heinz Stockinger. *Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication*. In Proceedings of IASTED International Conference on Applied Informatics (AI2001), Innsbruck, Austria, Feb 2001.

[SHA 95] *NIST. Secure hash standard. Federal Information Processing Standard, FIPS-180-1*, April April 1995. http://www.itl.nist.gov/fipspubs/fip180-1.htm.

[Sherwood 04] Rob Sherwood, Ryan Braud & Bobby Bhattacharjee. *Slurpie: A Cooperative Bulk Data Transfer Protocol*. In Proceedings of IEEE INFOCOM, March 2004.

[Stockinger 02a] Heinz Stockinger & Andrew Hanushevsky. *HTTP Redirection for Replica Catalogue Lookups in Data Grids*. In Proceedings of the 17th ACM Symposium on Applied Computing (SAC2002), Madrid, Spain, Mar 2002.

[Stockinger 02b] Heinz Stockinger, Asad Samar, Koen Holtman, Bill Allcock, Ian Foster & Brian Tierney. *File and Object Replication in Data Grids*. Cluster Computing, vol. 5, no. 3, pages 305–314, 2002.

[Stockinger 07] Heinz Stockinger. *Defining the grid: a snapshot on the current view*. J. Supercomput., vol. 42, pages 3–17, October 2007.

[Stoica 03] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek & Hari Balakrishnan. *Chord: a scalable peer-to-peer lookup protocol for internet applications*. IEEE/ACM Trans. Netw., vol. 11, pages 17–32, February 2003.

[Thain 01] Douglas Thain, Jim Basney, Se-Chang Son & Miron Livny. *The Kangaroo Approach to Data Movement on the Grid*. In Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10), 2001.

[Wei 05] Baohua Wei, Gilles Fedak & Franck Cappello. *Collaborative Data Distribution with BitTorrent for Computational Desktop Grids.* In Proceedings of the 4th International Symposium on Parallel and Distributed Computing, ISPDC'05, 2005.

[Weigle 05] Eric Weigle & Andrew A. Chien. *The Composite Endpoint Protocol (CEP): Scalable Endpoints for Terabit Flows*. In Proceedings of the IEEE International Symposium on Cluster Computing and the Grid, CCGrid'05, 2005.

[Zhao 04] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph & John D. Kubiatowicz. *Tapestry: A Resilient Global-scale Overlay for Service Deployment*. IEEE Journal on Selected Areas in Communications, vol. 22, pages 41–53, 2004.

[Zissimos 07] An. Zissimos, K. Doka, A. Chazapis & N. Koziris. *GridTorrent: Optimizing data transfers in the Grid with collaborative sharing*. In Proceedings of the 11th Panhellenic Conference on Informatics, 2007.