ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ & ΔΙΚΤΥΩΝ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

Απώτατη συνέπεια δεδομένων σε περιβάλλοντα υπολογιστικού νέφους

(Eventual data consistency in cloud computing environments)

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Τρυφωνόπουλος Τρύφωνας

Επιβλέπων: κ. Κατσαρός Δημήτριος

Μάρτιος 2013

Περιεχόμενα

Σύνοψη

Η τεχνολογία του σύννεφου είναι πλέον πραγματικότητα και εμπορικά διαθέσιμη για τους χρήστες. Στόχος της τεχνολογίας αυτής είναι να καλύψει τις ανάγκες των χρηστών για περισσότερο αποθηκευτικό χώρο και υπολογιστική ισχύ χωρίς την παρουσία του απαραίτητου υλικού στο χώρο του χρήστη.

Όπως κάθε νέα τεχνολογία, έτσι και το σύννεφο για να παρέχει αυτά που υπόσχετε στους πελάτες του πρέπει να αντιμετωπίσει διάφορα ζητήματα γύρω από τον τρόπο λειτουργίας του. Μια βασική πρόκληση είναι η αντιμετώπιση της συνέπειας των αντιγράφων των δεδομένων που διατηρούνται στους εξυπηρετητές(servers) που διατηρεί η εκάστοτε εταιρεία που παρέχει υπηρεσίες σύννεφου. Ο χρήστης ενδιαφέρεται να έχει πρόσβαση στα ενημερωμένα δεδομένα του ανά πάσα στιγμή. Ενδεχόμενη ασυνέπεια των δεδομένων του μπορεί να οδηγήσει σε αποτυχία την υπηρεσία του σύννεφου γιατί η εμπειρία του χρήστη είναι πιθανώς αρνητική.

Στην παρούσα εργασία θα ασχοληθούμε με το πρόβλημα της συνέπειας των δεδομένων. Θα μελετήσουμε αλγορίθμους που έχουν προταθεί για την αντιμετώπιση του προβλήματος αυτού και θα παρουσιάσουμε τη δική μας προσέγγιση στο πρόβλημα βασιζόμενοι σε αυτούς τους αλγορίθμους συνδοιάζοντας τις τεχνικές λειτουργίας τους.

Το υπόλοιπο της παρούσης εργασίας οργανώνεται ως εξής: στο Κεφάλαιο 1 κάνουμε μια εισαγωγή στην τεχνολογία του σύννεφου.Στο Κεφάλαιο 2 παρουσιάζουμε κάποια ζητήματα έρευνας γύρω από το σύννεφο, στο Κεφάλαιο 3 παρουσιάζουμε τη βιβλιογραφική έρευνα πάνω στο ζήτημα της συνέπειας των δεδομένων, στο Κεφάλαιο 4 παρουσιάζουμε τη δική μας προσέγγιση στο πρόβλημα της συνέπειας των δεδομένων, στο Κεφάλαιο 5 παρουσιάζουμε τα αποτελέσματα του δικού μας αλγορίθμου και τέλος στο Κεφάλαιο 6 παρουσιάζεται η σχετική βιβλιογραφία που χρησιμοποιήθηκε στην παρούσα εργασία.

Κεφάλαιο 1: Εισαγωγή

1.1 Τι είναι το υπολογιστικό σύννεφο(cloud computing)

Ως υπολογιστικό σύννεφο ορίζουμε τη χρήση υπολογιστικών πόρων (υλικού και λογισμικού), οι οποίοι χρησιμοποιούνται από τους χρήστες σαν υπηρεσία πάνω από το δίκτυο και όχι σαν τοπικοί εξυπηρετητές ή προσωπικές συσκευές. Η ονοματολογία αυτή προήλθε από το σχήμα σύννεφου που χρησημοποιείται για την αναπαράσταση της περίπλοκης δομής που περιέχει. Ουσιαστικά η λέξη σύννεφο που χρησημοποιούμε είναι μια μεταφορική έννοια του Διαδικτύου και η φράση υπολογιστικό σύννεφο σημαίνει υπολογισμός βασισμένος στο Διαδίκτυο, όπου διαφορετικές υπηρεσίες προσφέρονται στους χρήστες μέσω του Διαδικτύου.

1.1.1 Τύποι υπολογιστικού σύννεφου

Το υπολογιστικό σύννεφο χωρίζεται σε κατηγορίες ανάλογα με τις απαιτήσεις των χρηστών. Παρακάτω παρουσιάζουμε τις τέσσερις κατηγορίες υπολογιστικού σύννεφου:

Δημόσιο Σύννεφο(Public Cloud)

Διαφόρων ειδών δημόσιες εφαρμογές, αποθηκευτικός χώρος ή αλλοι πόροι διατίθενται στο ευρύ κοινό από έναν φορέα παροχής υπηρεσιών(service provider). Οι υπηρεσίες αυτές είτε είναι δωρεάν προς τους χρήστες είτε επί πληρωμή με βάση το μοντέλο πληρωμή ανά χρήση(pay per use model). Μεγάλοι δημόσιοι πάροχοι υπηρεσιών σύννεφου, όπως η Amazon, η Microsoft, η Google, διαθέτουν και λειτουργούν υποδομές σύννεφου προσφέροντας πρόσβαση σε αυτές μόνο μέσω διαδικτύου.

Σύννεφο Κοινότητας(Community Cloud)

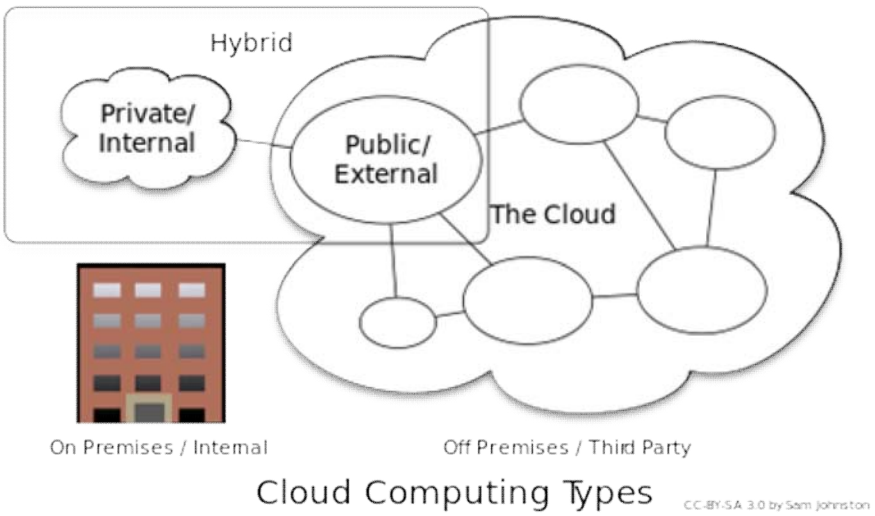
Το σύννεφο κοινότητας διαμοιράζει την υποδομή ανάμεσα σε αρκετούς οργανισμούς που έχουν κοινά συμφέροντα. Η διαχείρηση γίνεται εσωτερικά της κοινότητας είτε από τρίτους και η διατήρηση της υποδομής γίνεται εσωτερικά της κοινότητας ή εξωτερικά. Το κόστος χρήσης του σύννεφου κοινότητας μοιράζεται σε λιγότερους χρήστες από αυτούς του δημόσιου με αποτέλεσμα μερικές από τις δυνατότητες εξοικονόμησης κόστους του υπολογιστικού σύννεφου να πραγματοποιούνται.

Υβριδικό Σύννεφο(Hybrid Cloud)

Το υβριδικό σύννεφο είναι ένας συνδοιασμός δύο ή περισσοτέρων σύννεφων(δημόσιου, ιδιωτικού ή κοινότητας) τα οποία διατηρούν τις ιδιότητές τους αλλά ταυτόχρονα είναι συνδεδεμένα μεταξύ τους προσφέροντας τα πλεονεκτήματα όλων. Χρησιμοποιώντας την υβριδική αρχιτεκτονική, τόσο οι εταιρείες όσο και μεμονωμένοι χρήστες οφελούνται από την ανοχή σφαλμάτων συνδοιασμένη με την τοπική άμεση χρήση των δεδομένων τους χωρίς την ανάγκη σύνδεσης στο διαδίκτυο. Το υβριδικό σύννεφο απαιτεί τόσο την ύπαρξη τοπικών πόρων όσο και απομακρυσμένων.

Ιδιωτικό Σύννεφο(Private Cloud)

Η ιδιωτική δομή σύννεφου προορίζεται για μεμονωμένη χρήση από οργανισμούς, διαχειρίζεται εσωτερικά του οργανισμού ή από τρίτους και διατηρείται εσωτερικά του οργανισμού ή εξωτερικά. Ο σχεδιασμός ενός ιδιωτικού σύννεφου απαιτεί σε μεγάλο βαθμό την κατανόηση των απαιτήσεων της εταιρείας και την επανεκτίμηση των αποφάσεων για τους υπαρχοντες πόρους της. Σε κάθε βήμα σχεδιασμού προκύπτουν ζητήματα ασφάλειας τα οποία πρέπει να λυθούν για να αποφευχθούν σοβαρότερες ευπάθειες του συστήματος. Η κριτική που δέχεται το ιδιωτικό σύννεφο είναι μεγάλη γιατί όπως λένε οι επικριτές του δεν αξιοποιεί τις δυνατότητες που προσφέρει το υπολογιστικό σύννεφο.



Εικόνα 1. Τύποι υπολογιστικού νέφους

1.1.2 Χαρακτηριστικά του υπολογιστικού σύννεφου

Ανεξάρτητα με τον τύπο του σύννεφου που χρησιμοποιείται ορισμένα χαρακτηριστικά διατηρούνται σε όλους τους τύπους.

Κλιμάκωση(Scalability)

Το σύννεφο πρέπει να είναι ελαστικό, δηλαδή πρέπει η δέσμευση των πόρων να γίνεται με βάση τις ανάγκες. Κλιμάκωση σημαίνει να μπορεί να το σύννεφο να δεσμεύσει περισσότερους πόρους οταν υπάρχει ανάγκη και να ελευθερώσει πόρους όταν η ζήτηση είναι χαμηλή. Επίσης, σημαντική είναι η κλιμάκωση που πετυχαίνει η εφαρμογή που χρησιμοποιεί το σύννεφο, όταν προστίθενται χρήστες που χρησιμοποιούν την εφαρμογή ή αλλάζουν οι απαιτήσεις τις εφαρμογής.

Πρόσβαση αυτοεξυπηρέτησης(Self-service access)

Οι πελάτες του σύννεφου πρέπει να μπορούν να λάβουν τις επιθυμητές υπηρεσίες χωρίς κάποια εκτενή διαδικασία. Με έναν πολύ απλό και γρήγορο τρόπο ζητούν από τον πάροχο τον αποθηκευτικό χώρο, την υπολογιστική δύναμη και το λογισμικό που χρειάζονται και εκτελλούν την εργασία τους. Μόλις τελειώσουν με τη χρήση των πόρων που δέσμευσαν, αποσυνδέονται από το διαδίκτυο και οι πόροι επιστρέφονται αυτόματα και γίνονται διαθέσιμοι στον πάροχο.

Γενική διεπαφή χρήστη(General User Interface (GUI))

Οι υπηρεσίες σύννεφου πρέπει να έχουν τυποποιημένες διεπαφές και να παρέχουν οδηγίες για το πως δύο εφαρμογές ή πηγές δεδομένων επικοινωνούν μεταξύ τους. Οι τυποποιημένες διεπαφές βοηθούν τους χρήστες να χρησιμοποιούν πιο εύκολα περισσότερες υπηρεσίες μαζί.

Δωρεάν ή μέτρηση με βάση τη χρήση(Free or service use metering)

Αυτή τη στιγμή υπάρχουν υπηρεσίες σύννεφου που παρέχονται δωρεάν στους χρήστες, όπως για παράδειγμα επεξεργασία κειμένου. Υπάρχουν βέβαια και υπηρεσίες που παρέχονται επί πληρωμή, μηνιαία ή ετήσια, ή βασισμένη στο μοντέλο πληρωμή ανά χρήση.

1.1.3 Πως το υπολογιστικό σύννεφο μειώνει το κόστος

Οι περισσότερες μικρού και μεσαίου μεγέθους επιχειρήσεις ενδιαφέρονται να μειώσουν το κόστος λειτουργίας τους περικόπτοντας δαπάνες από όπου είναι δυνατόν. Η λύση της τεχνολογίας του σύννεφου, ιδιωτικού ή δημόσιου, αξίζει να εξεταστεί. Το υπολογιστικό σύννεφο μπορεί να οφελήσει την επιχείρηση με τους ακόλουθους τρόπους:

1) Μείωση της εκ των προτέρων επένδυσης.
2) Δυναμική επεκτασιμότητα.
3) Ταχεία ανάπτυξη.
4) Κατανεμημένη ανάκτηση δεδομένων από ενδεχόμενη καταστροφή.
5) Τεχνολογία που παρέχεται από μεγάλους και με καλή φήμη παρόχους.

Η τεχνολογική λύση του σύννεφου προσφέρει χαμηλού κόστους λύσεις στις επιχειρήσεις, αφού πλέον δεν χρειάζεται η επιχείρηση να διατηρεί δικό της εξοπλισμό από υλικό, για παράδειγμα εξυπηρετητές, ώστε να ικανοποιήσει τις λειτουργικές της ανάγκες. Το μόνο που χρειάζεται είναι να ανεβάζει στο σύννεφο τις εφαρμογές που της είναι απαραίτητες και να τις χρησιμοποιεί από εκεί. Επίσης, η επιχείρηση πληρώνει στον πάροχο του σύννεφου για όσο κάνει χρήση της υπηρεσίας. Αποφεύγει, έτσι, το κόστος διατήρησης, συντήρησης και εποπτείας από εξειδικευμένο προσωπικό των εξυπηρετητών που ενδεχομένως θα χρειαζόταν να διατηρεί σε δικό της χώρο.

1.2 Το θεώρημα CAP

Η μέχρι τώρα αναφορά μας στο σύννεφο μας έχει ενδεχομένως πείσει ότι είναι μια από τις καλύτερες τεχνολογίες που δημιούργήθηκαν με πολλά πλεονεκτήματα. Όπως όλες οι τεχνολογικές λύσεις που παρουσιάζονται, βασίζουν τη λειτουργία τους σε κάποιους κανόνες που καθορίζουν τις συνθήκες κάτω από τις οποίες η συγκεκριμένη τεχνολογία λειτουργεί, έτσι και η τεχνολογία του σύννεφου διέπεται και περιορίζεται από το θεώρημα CAP.

Το θεώρημα CAP, γνωστό και ως θεώρημα του Brewer, αναφέρει ότι είναι αδύνατο ένα κατανεμημένο σύστημα να προσφέρει ταυτόχρονα τις τρεις ακόλουθες εγγυήσεις:

1) **Συνέπεια-Consistency**: Είναι η ιδιότητα σύμφωνα με την οποία κάθε εξυπηρετητής επιστρέφει το σωστό αποτέλεσμα σε κάθε αίτηση, όπου το αποτέλεσμα είναι το

κατάλληλο σύμφωνα με τις προδιαγραφές της υπηρεσίας που παρέχεται. Η ακριβής έννοια της συνέπειας, εξαρτάται από τον τύπο της υπηρεσίας.

2) **Διαθεσιμότητα-Availability**: Είναι η ιδιότητα σύμφωνα με την οποία κάθε αίτηση λαμβάνει τελικά και μια απάντηση. Μια γρήγορη απάντηση είναι σαφώς προτιμότερη από μια αργή απάντηση, αλλά στα πλαίσια του θεωρήματος ακόμα και μια απαιτούμενη απάντηση είναι ικανή να δημιουργήσει προβλήματα. Σε αρκετά πραγματικά συστήματα μια αργή απάντηση είναι τόσο άσχημη όσο η μη απάντηση.

3) **Ανοχή σε διαμοιρασμό-Partition tolerance**: Σε αντίθεση με τις προηγούμενες δύο ιδιότητες, αυτή η ιδιότητα δεν σχετίζεται τόσο με την υπηρεσία όσο με το κατανεμημένο σύστημα. Η επικοινωνία μεταξύ των εξυπηρετητών είναι αναξιόπιστη και οι εξυπηρετητές μπορούν να χωριστούν σε πολλές ομάδες που να είναι αδύνατη η επικοινωνία μεταξύ τους. Έτσι, τα μηνύματα που ενδεχομένως ανταλλάσονται μπορεί να καθυστερούν ή και να χάνονται. Πρακτικά αντιμετωπίζουμε τα καθυστερημένα μηνύματα ως μη απάντηση. Παρόλη αυτή την κατάσταση που μπορεί να περιέλθει ένα κατανεμημένο σύστημα, αυτό καταφέρνει να συνεχίσει τη λειτουργία του απρόσκοπτα.

Σύμφωνα με το θεώρημα αυτό, ένα κατανεμημένο σύστημα μπορεί να ικανοποιεί κάθε φορά μόνο δύο από τα τρία παραπάνω χαρακτηριστικά μια συγκεκριμένη χρονική στιγμή. Δεχόμενοι ότι το δίκτυο πανω από το οποίο λειτουργεί ένα κατανεμημένο σύστημα είναι αναξιόπιστο, θυσιάζουμε αμέσως τη συνέπεια ή τη διαθεσιμότητα του συστήματος μας. Βέβαια, έχουν γίνει προσπάθειες να υπάρξει μια ισορροπία ανάμεσα στη συνέπεια και τη διαθεσιμότητα που προσφέρει το κατανεμημένο σύστημα. Η τελική, όμως, απόφαση για το ποια από τις ιδιότητες θα θυσιαστεί, εξαρτάται από την υπηρεσία που θέλουμε να προσφέρει το σύστημά μας.

1.3 Εμπορικά συστήματα νέφους

Η τεχνολογία του σύννεφου είναι ήδη διαθέσιμη για τους χρήστες και παρέχεται από μεγάλους παρόχους. Ήδη μεγάλες εταιρείες που δραστηριοποιούνται στο χώρο της τεχνολογίας και κατέχουν σημαντική θέση, όπως η Amazon με το Dynamo, η Yahoo με το PNUTS, η Microsoft με το Azure, η Google έχουν εντάξει στις διαθέσιμες υπηρεσίες τους τόσο προς τις επιχειρήσεις όσο και προς απλούς χρήστες την τεχνολογία του σύννεφου. Κάθε εταιρεία έχει υιοθετήσει διαφορετική αρχιτεκτονική και τρόπο σχεδιασμού της δικής της πλατφόρμας σύννεφου, προσπαθώντας να βελτιστοποιήσει την εμπειρία των χρηστών και να ανταγωνιστεί με τον καλύτερο δυνατό τρόπο τις υπόλοιπες εταιρείες.

Κεφάλαιο 2: Ζητήματα έρευνας γύρω από το σύννεφο

Ως νεο-εμφανιζόμενη τεχνολογία το σύννεφο, μαζί με τις διευκολύνσεις που προσέφερε σε θέματα όπως η αποθήκευση δεδομένων, η μεγαλύτερη διαθέσιμη υπολογιστική δύναμη, άνοιξε και το δρόμο για νέες ερευνητικές δραστηριότητες για την αντιμετώπιση διαφόρων προβλημάτων.

2.1 Ασφάλεια

Σημαντικό πρόβλημα που χρίζει αντιμετώπισης στο σύννεφο είναι η ασφάλεια των δεδομένων. Οι ενδιαφερόμενοι πελάτες, απαιτούν από τις εταιρείες που παρέχουν λύσεις σύννεφου να τους εγγυηθούν την ασφάλεια των δεδομένων τους. Το πρόβλημα της ασφάλειας , μπορεί να χωριστεί σε δύο κατηγορίες: στην πρώτη κατηγορία ανήκουν τα θέματα ασφάλειας που αντιμετωπίζονται από τους παρόχους και στην δεύτερη κατηγορία αυτά που αντιμετωπίζονται από τους πελάτες.

Ο πάροχος οφείλει να διαβεβαιώσει τον πελάτη ότι οι υποδομές του είναι ασφαλείς, όπως επίσης τα δεδομένα του και οι εφαρμογές του. Ο πελάτης με τη σειρά του πρέπει να βεβαιωθεί ότι ο πάροχος έχει λάβει τα απαραίτητα μέτρα ασφαλείας, ώστε να προστατέψει τις πληροφορίες του.

Η εκτεταμένη χρήση της εικονικοποίησης στη δομή του σύννεφου, έφερε νέα ζητήματα ασφαλείας για τους πελάτες ή ενικοιαστές του σύννεφου. Η εικονικοποίηση άλλαξε τη σχέση μεταξύ υλικού και λογισμικού. Επομένως, έχουμε την εισαγωγή ενός επιπλέον επιπέδου, της εικονικοποίησης, που πρέπει είναι κατάλληλα ρυθμισμένο, διαχειρίσιμο και ασφαλές.

Οι κατηγορίες ελέγχου της ασφάλειας της αρχιτεκτονικής του σύννεφου είναι:

1) **Προληπτικός έλεγχος- Deterrent Control** : ο έλεγχος αυτός τίθεται σε εφαρμογή για να αποφευχθεί οποιαδήποτε σκόπιμη επίθεση στο σύστημα του σύννεφου. Μοιάζει περισσότερο με προειδοποιηση και δεν μειώνει την πραγματική ευπάθεια του συστήματος.
2) **Προληπτικός έλεγχος- Preventative Control** : ο έλεγχος αυτός αναβαθμίζει τη δυναμική του συστήματος με τη διαχείρηση των τρωτών σημείων και διασφαλίζει το σύστημα από αυτά. Εάν πραγματοποιηθεί μια επίθεση, ο έλεγχος αυτός είναι σε θέση

να καλύψει την επίθεση και να μειώσει τις ζημιές και την παραβίαση του συστήματος ασφαλείας.

3) **Διορθωτικός έλεγχος-Corrective Control** : Ο διορθωτικός έλεγχος χρησιμοποιείται για τη μείωση της επίδρασης μιας επίθεσης. Σε αντίθεση με τον προληπτικό έλεγχο(preventative control), ο διορθωτικός αναλαμβάνει δράση όταν συμβαίνει μια επίθεση.
4) **Ανιχνευτικός έλεγχος-Detective Control** : Ο έλεγχος αυτός χρησιμοποιείται για την ανίχνευση οποιασδήποτε επίθεσης μπορεί να συμβεί στο σύστημα. Στην περίπτωση επίθεσης, σηματοδοτεί τον προληπτικό(preventative) ή διορθωτικό(corrective) έλεγχο, ώστε να αντιμετωπίσουν το πρόβλημα.

2.2 Αποδοτικότητα στην παροχή υπηρεσιών

Η αποδοτικότητα στην παροχή των υπηρεσιών του σύννεφου εξαρτάται από διάφορες παραμέτρους. Για τη σωστή ανάπτυξη του συστήματος χρειάζεται η χρήση εξαρτημάτων και εργαλείων ανάπτυξης που θα βοηθήσουν στην αύξηση της παραγωγικότητας στην ανάπτυξη εφαρμογών, τη δυνατότητα ανάπτυξης παράλληλων εφαρμογών καθώς και βελτιστοποιημένων εφαρμογών μειώνοντας την περιπλοκότητα της διαδικασίας ανάπτυξης. Επίσης σημαντική είναι η δημιουργία αρχιτεκτονικών που κλιμακώνουν πολύ εύκολα. Είναι σημαντικό το σύστημά μας να μπορεί να προσαρμόζεται εύκολα στις απαιτήσεις που υπάρχουν κάθε φορα. Επιπλέον, η διαχείρηση των πόρων του συστήματος είναι ζωτικής σημασίας για την αποδοτικότητα στην παροχή των υπηρεσιών. Το σύννεφο, οφείλει να διαχειρίζεται με αποτελεσματικό τρόπο τους πόρους του, ώστε να μπορεί να ικανοποιεί αρκετούς πελάτες και οι ίδιοι να μπορούν να απολαμβάνουν την εμπειρία που τους προσφέρεται. Τέλος, η διαθεσιμότητα των υπηρεσιών συμβάλλει στην αποδοτική παροχή των υπηρεσιών.

2.3 Συνέπεια των δεδομένων

Η συνέπεια των δεδομένων είναι καθοριστικής σημασίας για την επιτυχία ενός κατανεμημένου συστήματος. Ο χρήστης ενδιαφέρεται να λαμβάνει τα πιο πρόσφατα δεδομένα του. Αδιαφορεί για τις δυσκολίες επίτευξης ενός τέτοιου στόχου σε ένα κατανεμημένο σύστημα, όπως είναι το σύννεφο. Οι ερευνητές έχουν καταφέρει να δημιουργήσουν αλγορίθμους που σε μεγάλο βαθμό επιτυγχάνουν τη συνέπεια των δεδομένων. Θα μελετήσουμε εκτενέστερα το πρόβλημα της συνέπειας των δεδομένων στο Κεφάλαιο 4.

2.4 Φορητότητα μεταξύ των παρόχων

Ένα επιπλέον ζήτημα έρευνας στην τεχνολογία του σύννεφου είναι η φορητότητα μεταξύ των παρόχων. Κάθε πάροχος επιλέγει την αρχιτεκτονική, τις υπηρεσίες και τις εγγυήσεις που δίνει στους πελάτες του. Επιβεβλημένη, όμως, είναι η ανάγκη να ληφθεί υπόψη το γεγονός ότι οι πελάτες ανά πάσα στιγμή μπορεί να αλλάξουν πάροχο. Θέλουν, παρόλα αυτά, να έχουν πρόσβαση στα δεδομένα τους, όπως παλιά, και να συνεχίσουν να λαμβάνουν τις υπηρεσίες από τις εφαρμογές που χρησιμοποιούσαν και στον προηγούμενο πάροχο. Για να καταστεί εφικτό αυτό, πρέπει να υπάρχει η δυνατότητα της φορητότητας μεταξύ των παρόχων, τόσο για τα δεδομένα όσο και για τις υπηρεσίες. Η δυσκολία σε αυτή την προσέγγιση έγκειται στο γεγονός ότι κάθε πάροχος σχεδιάζει το σύστημά του με βάση τα δικά του κριτήρια και δεν είναι διατεθιμένος να κάνει γνωστές τις τεχνικές που χρησιμοποιεί.

Κεφάλαιο 3: Βιβλιογραφική έρευνα

Η πρόσφατη ανάπτυξη της τεχνολογίας του σύννεφου προσφέρει αρκετά πεδία έρευνας, όπως περιγράψαμε στο Κεφάλαιο 2. Παρακάτω θα παρουσιάσουμε κάποιες ενδιαφέρουσες εργασίες γύρω από το ζήτημα της συνέπειας των δεδομένων.

Οι συγγραφείς στο [Maintaining Strong Cache Consistency in the World Wide Web] μελετούν τη συνέπεια της προσωρινής μνήμης για τις ανάγκες του Διαδικτύου. Η λύση της επιβολής αυστηρής συνέπειας για τα δεδομένα της προσωρινής μνήμης είναι ακριβή για το Διαδίκτυο. Αντίθετα, μια πολιτική χαλαρής συνέπειας είναι περισσότερο κατάλληλη, όπως η τεχνική του TTL(Time To Live). Στην εργασία τους παρουσιάζουν τρεις πολιτικές συνέπειας. Την προσαρμοζόμενη TTL (Adaptive TTL), έλεγχος κάθε στιγμή(polling every time) και ακύρωση(invalidation). Σύμφωνα με τα πειράματά τους η μέθοδος της ακύρωσης(invalidation) έχει παρόμοια αποτελέσματα με την μέθοδο του προσαρμοζόμενου TTL(adaptive TTL) τόσο σε κίνηση δικτύου, όσο και σε μέσο χρόνο απόκρισης των χρηστών και του φόρτου των εξυπηρετητών. Ενώ, από την άλλη πλευρά, η τεχνική του ελέγχου κάθε στιγμή(polling every time) έχει σημαντικά μεγαλύτερη κίνηση δικτύου και υψηλότερους χρόνους απόκρισης από την προσαρμοζόμενη TTL(adaptive TTL). Καταλήγουν, τέλος, στο συμπέρασμα ότι με τη μέθοδο της ακύρωσης(invalidation) μπορούν να επιτύχουν αυστηρή συνέπεια.

Στο [Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency] οι συγγραφείς προτείνουν την τεχνική των μισθώσεων(leases) για την αντιμετώπιση του προβλήματος της συνέπειας. Πρόκειται για μια τεχνική που βασίζεται στο χρόνο. Ο κάτοχος ενός αρχείου δεδομένων διατηρεί και μια μίσθωση για το αρχείο αυτό, που έχει συγκεκριμένη διάρκεια και δηλώνει ότι το αρχείο αυτό δεν θα υποστεί αλλαγές από κανέναν άλλο χρήστη για όσο χρόνο διαρκεί η μίσθωση(lease).

Πιο συγκεκριμένα, η προσωρινή μνήμη που χρησιμοποιεί την τεχνική των μισθώσεων για τη συνέπεια των δεδομένων της, χρειάζεται μια έγκυρη μίσθωση για τα δεδομένα αυτά πριν τα επιστρέψει σαν απάντηση στο χρήστη που έχει κάνει αίτηση ανάγνωσης, ή τα τροποποιήσει σαν αποτέλεσμα αίτησης εγγραφής από το χρήστη. Ο εξυπηρετητής μαζί με τα δεδομένα που επιστρέφει στον πελάτη, επιστρέφει και τη μίσθωση για τα δεδομένα αυτά, εγγυώμενος, έτσι,

ότι τα δεδομένα δεν θα τροποποιηθούν από κανέναν άλλο χρήστη για όσο διαρκεί η μίσθωση, εκτός εάν ο εξυπηρετητής αποκτά τη δυνατότητα τροποποίησης από τον κάτοχο της μίσθωσης.

Εάν τα δεδομένα που είναι αποθηκευμένα στην προσωρινή μνήμη ζητηθούν εντός της διάρκειας της μίσθωσης, επιστρέφονται στον πελάτη χωρίς να χρειάζεται η επικοινωνία με τον εξυπηρετητή. Αντίθετα, όταν η μίσθωση λήξει, χρειάζεται επέκταση της μίσθωσης και επικοινωνία με τον εξυπηρετητή για να ικανοποιηθεί ένα αίτημα ανάγνωσης. Στην περίπτωση της εγγραφής, ο εξυπηρετητής μπορεί να αναβάλει την εκπλήρωση του αιτήματος μέχρις ώτου οι κάτοχοι των μισθώσεων επιτρέψουν την τροποποίηση των δεδομένων πριν τη λήξη της μίσθωσης ή λήξει η μίσθωση.

Στη συνέχεια παρουσιάζουμε μια παραλλαγή της προηγούμενης τεχνικής. Οι συγγραφείς στο [Volume Leases for Consistency in Large-Scale Systems] προτείνουν τη λύση των μισθώσεων τόμων(volume leases) για την αντιμετώπιση της συνέπειας των δεδομένων. Η τεχνική των απλών μισθώσεων είναι καλή όταν γίνονται συχνές αναφορές σε ένα αρχείο. Αν όμως οι αναφορές αυτές γίνονται εντός μια μεγαλύτερης χρονικής διάρκειας οι απλές μισθώσεις δεν επαρκούν. Έτσι, οι συγγραφείς με την τεχνική των μισθώσεων τόμων, κρατούν μισθώσεις τόσο για μεμωνομένα αρχεία όσο και για τόμους με συναφή αρχεία. Ένα αρχείο που διατηρείται στην προσωρινή μνήμη του πελάτη θεωρείται έγκυρο αν τόσο η μεμωνομένη μίσθωση του αρχείου όσο και η μίσθωση τόμου είναι έγκυρες.

Η ανάγνωση των δεδομένων από την προσωρινή μνήμη γίνεται αν οι πελάτες διατηρούν έγκυρη μίσθωση τόσο για το αντικείμενο όσο και για τον τόμο. Μισθώσεις που λήγουν ανανεώνονται κατόπιν επικοινωνίας με τον κατάλληλο εξυπηρετητή.

Όσον αφορά την εγγραφή των δεδομένων, πριν συμβεί οποιαδήποτε τροποποίηση, ο εξυπηρετητής στέλνει μήνυμα ακύρωσης σε όλους τους πελάτες που διατηρούν μια έγκυρη μίσθωση για το αντικείμενο. Ο εξυπηρετητής καθυστερεί την εγγραφή μέχρις ώτου να λάβει επιβεβαίωση από όλους τους πελάτες, σαν απάντηση στο μήνυμα ακύρωσης που απέστειλε, ή μίσθωση για το αντικείμενο λήξει. Έπειτα, ο εξυπηρετητής τροποποιεί το αντικείμενο και αυξάνει τον αριθμό έκδοσης του αντικειμένου.

Οι συγγραφείς της παραπάνω εργασίας, λαμβάνουν υπόψη και τις περιπτώσεις κάποιος πελάτης ή εξυπηρετητής να τεθεί εκτός κανονικής λειτουργίας. Για τη διαχείρηση των εκτός

λειτουργίας πελατών, διατηρείται σε κάθε εξυπηρετητή μια λίστα με τους πελάτες που είναι εκτός λειτουργίας. Όταν ο πελάτης επανέλθει σε κανονική λειτουργία και επικοινωνήσει με τον εξυπηρετητή, είτε για ανανέωση τη μίσθωσης τόμου είτε υποβάλοντας αίτημα ανάγνωσης ή εγγραφής, και ανήκει στην λίστα με τους πελάτες εκτός λειτουργίας, ενημερώνεται από τον εξυπηρετητή να ανανεώσει τις μισθώσεις όλων των αντικειμένων που ανήκουν στον τόμο.

Από την άλλη πλευρά, όταν ένας εξυπηρετητής ανακάμψει από κάποιο πρόβλημα, ακυρώνει όλες τις μισθώσεις τόμων περιμένοντας να λήξουν. Αυτό μπορεί να συμβεί με δύο τρόπους. Πρώτον, ο εξυπηρετητής μπορεί να αποθηκεύει σε σταθερή μνήμη την πιο πρόσφατη χρονική στιγμή λήξης κάποιας μίσθωσης τόμου. Μετά την ανάκαμψη, ο εξυπηρετητής διαβάζει αυτήν τη χρονοσφραγίδα και καθυστερεί όλες τις εγγραφές μέχρι αυτή τη χρονική στιγμή. Δεύτερον, ο εξυπηρετητής μπορεί να αποθηκεύσει τη διάρκεια της μεγαλύτερης μίσθωσης τόμου και στη συνέχεια να καθυστερήσει τις εγγραφές μέχρι να παρέλθει η χρονική αυτή στιγμή.

Βέβαια, μπορούμε να αντιμετωπίσουμε το πρόβλημα της ανάκαμψης ενός εξυπηρετητή θεωρώντας ότι όλοι οι πελάτες βρίσκονται εκτός λειτουργίας, οπότε, ακολουθούμε τη διαδικασία που περιγράψαμε παραπάνω.

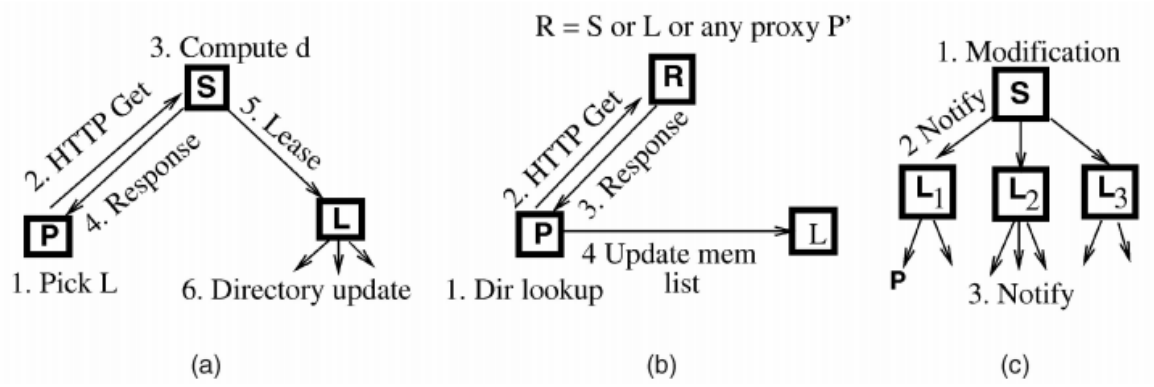
Οι συγγραφείς στο [Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web] εστιάζουν στη μελέτη της διάρκειας των μισθώσεων και μέσω μοντέλου που κατασκεύασαν προσπαθούν να καθορίσουν τη βέλτιστη διάρκεια της μίσθωσης ανάλογα με την επισκεψιμότητα που έχει ένα αρχείο αλλά και το πόσο γρήγορα αλλάζει.

Τέλος, οι συγγραφείς στο [Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases] περιγράφουν την τεχνική των συνεργατικών μισθώσεων(cooperative leases). Προτείνουν για κάθε αντικείμενο να εκλέγεται ένας αρχηγός(leader), ο οποιός κρατά τη μίσθωση για το συγκεκριμένο αντικείμενο σε όλους τους πληρεξούσιους εξυπηρετητές(proxies). Πιο συγκεκριμένα, όταν ένα αντικείμενο ζητείται για πρώτη φορά, χρειάζεται να γίνει εκλογή αρχηγού για αυτό το αντικείμενο. Αφού εκλεχθεί ο αρχηγός, ο κεντρικός εξυπηρετητής(server) επιστρέφει το ζητούμενο αντικείμενο στον πληρεξούσιο εξυπηρετητή που το ζήτησε και τη μίσθωση για το αντικείμενο αυτό στον αρχηγό. Στη συνέχεια ο αρχηγός κάνει γνωστό σε όλους τους εξυπηρετητές ότι ο ίδιος είναι αρχηγός

για το συγκεκριμένο αντικείμενο. Από αυτή τη στιγμή και έπειτα ο αρχηγός είναι υπεύθυνος για τη διαχείρηση της μίσθωσης για λογαριασμό των πληρεξούσιων εξυπηρετητών. Επίσης, ο αρχηγός διατηρεί μια λίστα με τους πληρεξούσιους εξυπηρετητές που διατηρούν το αντικείμενο.

Για κάθε επόμενη αίτηση του αντικειμένου, ο πληρεξούσιος εξυπηρετητής ελέγχει την προσωρινή του μνήμη και αν το αντικείμενο υπάρχει το επιστρέφει. Σε αντίθετη περίπτωση, ο πληρεξούσιος εξυπηρετητής μπορεί να λάβει το αντικείμενο είτε από τον κεντρικό εξυπηρετητή είτε από οποιονδήποτε άλλο πληρεξούσιο εξυπηρετητή το διαθέτει. Η μόνη απαίτηση που τίθεται από τη μέθοδο των συνεργατικών μισθώσεων είναι η ενημέρωση του αρχηγού από τον πληρεξούσιο εξυπηρετητή για το ενδιαφέρον για το συγκεκριμένο αντικείμενο. Ο αρχηγός στη συνέχεια, ενημερώνει τη λίστα που διατηρεί με τους πληρεξούσιους εξυπηρετητές που διατηρούν το αντικείμενο.

Τέλος, η τροποποίηση ενός αντικειμένου στον κεντρικό εξυπηρετητή πρέπει να γίνει γνωστή στους πληρεξούσιους εξυπηρετητές. Ο κεντρικός εξυπηρετητής ενημερώνει τον αρχηγό του αντικεμένου και αυτός με τη σειρά του όλους τους πληρεξούσιους εξυπηρετητές που βρίσκονται στη λίστα και διατηρούν το αντικείμενο.



Εικόνα 2. Αλληλεπίδραση μεταξύ εξυπηρετητών(S), αρχηγών(L), πληρεξούσιων εξυπηρετητών(P) στις συνεργατικές μισθώσεις. (α) Αίτημα πρώτης φοράς, (β) επακόλουθη αίτηση αντικειμένου και (γ) τροποποίηση αντικειμένου.

Κεφάλαιο 4: Μελέτη του προβλήματος της συνέπειας δεδομένων

Όπως διαπιστώσαμε στο προηγούμενο κεφάλαιο, αρκετοί ερευνητές έχουν ασχοληθεί με το πρόβλημα της συνέπειας των δεδομένων στο σύννεφο. Στόχος μας στην παρούσα εργασία είναι να παρουσιάσουμε τη δική μας προσέγγιση στο πρόβλημα αυτό.

Κατά τη μελέτη των ήδη δημοσιευμένων εργασιών εστιάσαμε στη χρήση των μισθώσεων ως εργαλείο για την επίλυση του προβλήματός μας. Πράγματι, πρόκειται για μια τεχνική που κατορθώνει να κρατήσει σε υψηλά επίπεδα τη συνέπεια των δεδομένων. Διάφορες παραλλαγές της τεχνικής των μισθώσεων έχουν προταθεί, μισθώσεις τόμων, συνεργατικές μισθώσει, με αξιόλογα αποτελέσματα.

Ιδιαίτερη προσοχή δώσαμε στη μελέτη των συνεργατικών μισθώσεων. Στο [Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases] οι συγγραφείς περιγράφουν με κάθε λεπτομέρεια την λειτουργία των συνεργατικών μισθώσεων. Από την έρευνά τους αυτή κρατήσαμε την ιδέα του αρχηγού(leader) που χρησιμοποιούν οι συγγραφείς.

Προτείνεται, λοιπόν, να εκλέγεται ένας αρχηγός για κάθε αντικείμενο που είναι διαθέσιμο προς χρήση. Η εκλογή αυτή μπορεί να γίνει με δύο τρόπους. Ο απλούστερος τρόπος είναι να εκλέγεται αρχηγός για το αντικείμενο, εκείνος ο πληρεξούσιος ο οποίος λαβάνει πρώτη φορά το αίτημα για το αντικείμενο. Αυτή η προσέγγιση έχει το πλεονέκτημα ότι ένας πληρεξούσιος μπορεί να γίνει αρχηγός για πολλά αντικείμενα και να είναι υπεύθυνος για τη συνέπεια αυτών, με αποτέλεσμα τη μικρότερη επιβάρυνση στην επικοινωνία που χρειάζεται για τη διατήρηση της συνέπειας. Παρόλα αυτά, όμως, το μειονέκτημα της προσέγγισης αυτής είναι ότι δεν γίνεται ισοκατανομή των αρχηγών, με αποτέλεσμα κάποιοι πληρεξούσιοι να είναι υπερφορτωμένοι με τις λειτουργίες που συνεπάγεται η ιδιότητα του αρχηγού και άλλοι όχι. Επίσης, είναι πιθανό αρκετοί πληρεξούσιοι να γίνουν αρχηγοί για ένα αντικείμενο αν λάβουν ταυτόχρονα αιτήσεις για αυτό το αντικείμενο. Επομένως, χρειάζονται επιπλέον μηχανισμοί για να αποτρέψουν ένα τέτοιο ενδεχόμενο παρουσίας περισσοτέρων του ενός αρχηγών για ένα αντικείμενο.

Μια δεύτερη προσέγγιση επιλογής αρχηγού είναι μέσω μιας συνάρτησης κατακερματισμού(hash function). Η εκλογή του αρχηγού με αυτόν τον τρόπο αποφεύγει τα

μειονεκτήματα της προηγούμενης προσέγγισης, έχει όμως και αυτή ένα μειονέκτημα. Αυξάνεται η επιβάρυνση στην επικοινωνία αφού λόγω της συνάρτησης κατακερματισμού, ο πληρεξούσιος που επιλέχθηκε για αρχηγός ενός αντικειμένου να μην έχει το αντικείμενο, οπότε είναι αναγκαία η επικοινωνία του με τους πληρεξούσιους που κατέχουν το αντικείμενο.

Στην εργασία μας υιοθετήσαμε την έννοια του αρχηγού και επιλέξαμε η εκλογή να γίνεται με βάση την απόσταση του χρήστη από τον εξυπηρετητή. Ως απόσταση εννοούμε τον αριθμό των δρομολογητών που χρειάζεται να περάσει ένα αίτημα του πελάτη για να φτάσει στον εξυπηρετητή. Έτσι, επιλέγουμε τον αρχηγό με βάση τη μικρότερη απόσταση. Η διαφορά είναι, ότι ο αρχηγός πλέον είναι για κάθε πελάτη και όχι για αντικείμενο. Κάθε πελάτης, δηλαδή, ζητά κάποιο αντικείμενο από τον αρχηγό του και μόνο.

Η δομή του δικτύου που χρησιμοποιούμε για τη μελέτη μας αποτελείται από ένα και μόνο επίπεδο εξυπηρετητών. Στόχος μας είναι να εφαρμόσουμε την έννοια του αρχηγού σε ένα επίπεδο εξυπηρετητών σε αντίθεση με τη δομή δικτύου που χρησιμοποιούν οι συγγραφείς στο [Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases] και να μελετήσουμε τη συνέπεια που επιτυγχάνουμε.

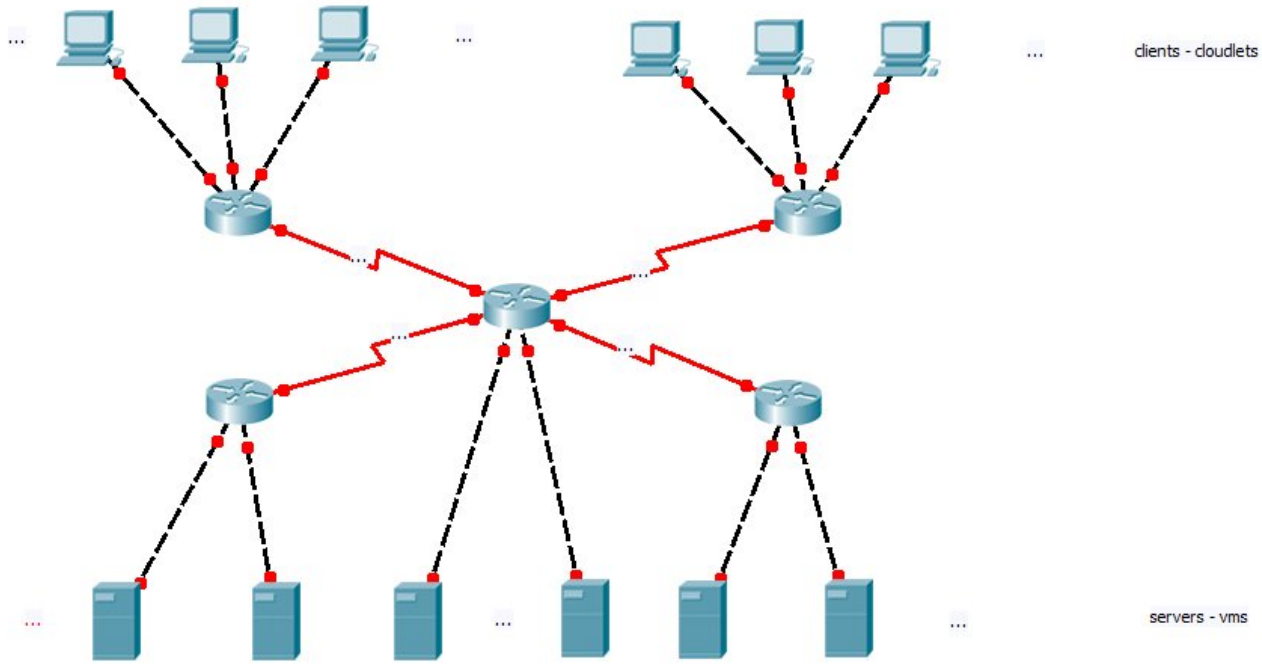
Στο επόμενο κεφάλαιο θα περιγράψουμε το περιβάλλον εκτέλεσης των πειραμάτων και τις τεχνικές λεπτομέρειες του αλγορίθμου που κατασκευάσαμε.

Κεφάλαιο 5: Αποτελέσματα πειραμάτων

Στην ενότητα αυτή θα παρουσιάσουμε τα αποτελέσματα της προσομοίωσης του αλγορίθμου που κατασκευάσαμε. Η προσομοίωση βασίστηκε στο εργαλείο CloudSim, που παρέχει το κατάλληλο πλαίσιο για την εκτέλεση και προσομοίωση πειραμάτων για ζητήματα γύρω από το σύννεφο. Δίνει τη δυνατότητα να επικεντρωθούμε σε συγκεκριμένα ζητήματα τα οποία μελετάμε χωρίς να χρειάζεται να μεριμνήσουμε για τις υπόλοιπες παραμέτρους που λαμβάνουν χώρα σε ένα περιβάλλον σύννεφου.

Για τη μελέτη της συνέπειας των δεδομένων που θέλουμε να εξετάσουμε, χρειαζόμαστε έναν αριθμό εξυπηρετητών, έναν αριθμό πελατών και ένα αρχείο το οποίο προσπελαύνουν οι πελάτες. Για να μοντελοποιήσουμε τους εξυπηρετητές χρησιμοποιούμε τις εικονικές μηχανές(virtual machines) που μας παρέχει το περιβάλλον του CloudSim. Η μοντελοποίηση των πελατών γίνεται από τα cloudlets που παρέχει το περιβάλλον προσομοίωσης.

Παρακάτω φαίνεται σχηματικά η τοπολογία του δικτύου που χρησιμοποιούμε. Αξίζει να τονίσουμε στο σημείο αυτό ότι έχουμε μόνο ένα επιπέδο απο εξυπηρετητές, οι οποίοι συνδέονται με τους πελάτες μέσω δρομολογητών, ενός ή περισσότερων.



Εικόνα 3. Τοπολογία του δικτύου.

Επιπλέον, χρειαζόμαστε ένα αρχείο το οποίο προσπελαύνουν οι πελάτες(cloudlets) και το οποίο θα αλλάζει με κάποιο τρόπο εκδόσεις ώστε να καταφέρουμε να μετρήσουμε τη συνέπεια μεταξύ της έκδοσης του αρχείου που έχει ο πελάτης και της έκδοσης που έχει ο εξυπηρετητής. Για να μοντελοποιήσουμε την αλλαγή της έκδοσης του αρχείου τόσο στον πελάτη όσο και στον εξυπηρετητή χρησιμοποιήσαμε την εκθετική κατανομή.

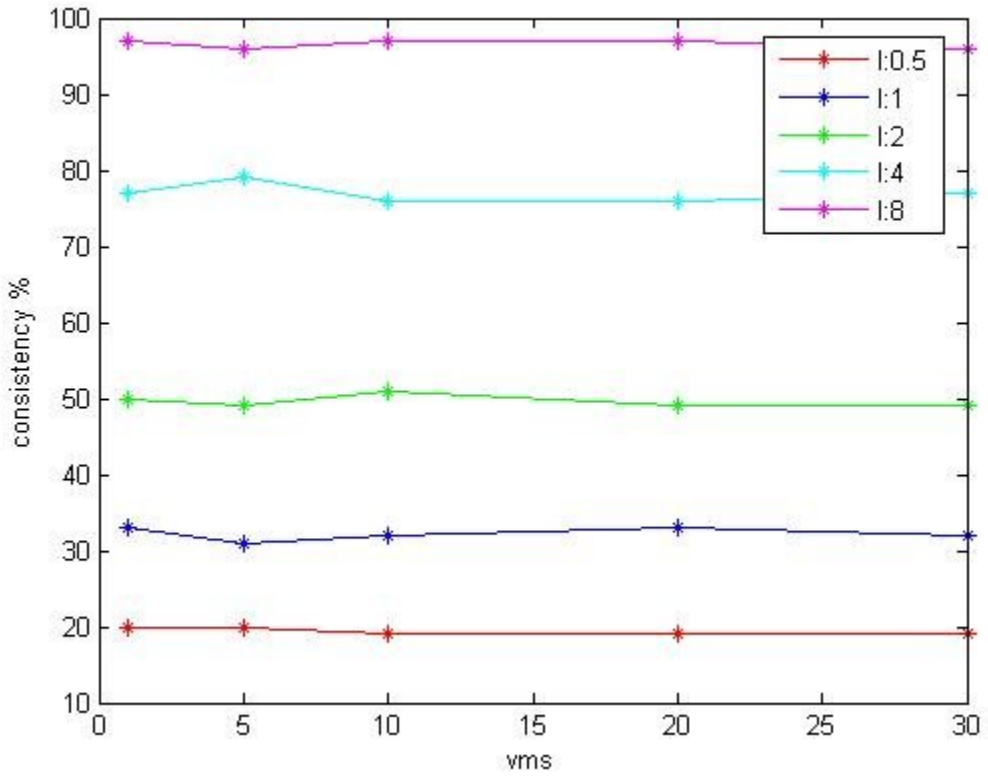
Η μελέτη του αλγορίθμου μας, υποθέτει ότι έχουμε ένα επίπεδο εξυπηρετητών και τους πελάτες, όπως αναφέραμε και παραπάνω. Αρχικά, με τυχαίο τρόπο, παράγονται οι αποστάσεις των πελατών από τους εξυπηρετητές. Οι αποστάσεις αυτές αναπαριστούν τους δρομολογητές που πρέπει να περάσει ένα αίτημα ενός πελάτη για να φτάσει σε έναν εξυπηρετητή. Έχοντας τις αποστάσεις των πελατών από τους εξυπηρετητές εκλέγουμε τον αρχηγό εξυπηρετητή για κάθε πελάτη με βάση τη μικρότερη απόσταση του πελάτη από τους εξυπηρετητές. Αφού γίνει η εκλογή των αρχηγών για όλους τους πελάτες γίνεται η ανάθεση των πελατών στον αρχηγό τους. Δηλαδή, κάθε πελάτης σχετίζεται πλεόν με τον αρχηγό του εξυπηρετητή και από αυτόν ζητά ό,τι δεδομένα χρειάζεται, στην περίπτωσή μας το αρχείο που προσομοιώνουμε.

Λόγω της ύπαρξης της τυχαιότητας εκτελούμε τα πειράματά μας για χίλιες επαναλήψεις. Σε κάθε επανάληψη παράγεται μια έκδοση αρχείου τόσο για τους εξυπηρετητές όσο και για τους πελάτες. Όλοι οι εξυπηρετητές έχουν την ίδια έκδοση αρχείου. Σε κάθε επανάληψη ελέγχεται αν ο πελάτης έχει την ίδια έκδοση αρχείου με τον αρχηγό του εξυπηρετητή. Στην περίπτωση που οι εκδόσεις είναι ίδιες έχουμε συνέπεια μεταξύ των αντιγράφων των αρχείων, αλλιώς θεωρούμε το αρχείο μη συνεπές.

Η προσομοίωση εκτελέστηκε σε υπολογιστικό σύστημα με επεξεργαστή Intel Core 2 Duo T5450, συχνότητας 1,67GHz, μνήμης RAM 2GB και λειτουργικού συστήματος Windows 7 32-bit. Η ανάπτυξη του κώδικα έγινε στο NetBeans IDE 7.2.1. Οι εξυπηρετητές(vms) που χρησιμοποιήθηκαν στην προσομοίωση αποτελούνται από έναν επεξεργαστή και έχουν μνήμη RAM 256MB. Επιπλέον, ο μέγιστος αριθμός δρομολογητών που χρησιμοποιούμε για την εκτέλεση των πειραμάτων είναι είκοσι(20). Αυτό σημαίνει ότι, για να φτάσει ένα αίτημα ενός πελάτη σε έναν εξυπηρετητή στη χειρότερη περίπτωση θα διέλθει από είκοσι δρομολογητές.

Στο σημείο αυτό θα παρουσιάσουμε τα αποτελέσματα της προσομοίωσης που εκτελέσαμε. Παρακάτω φαίνεται πως επηρεάζετε η συνέπεια από τον διαφορετικό αριθμό των

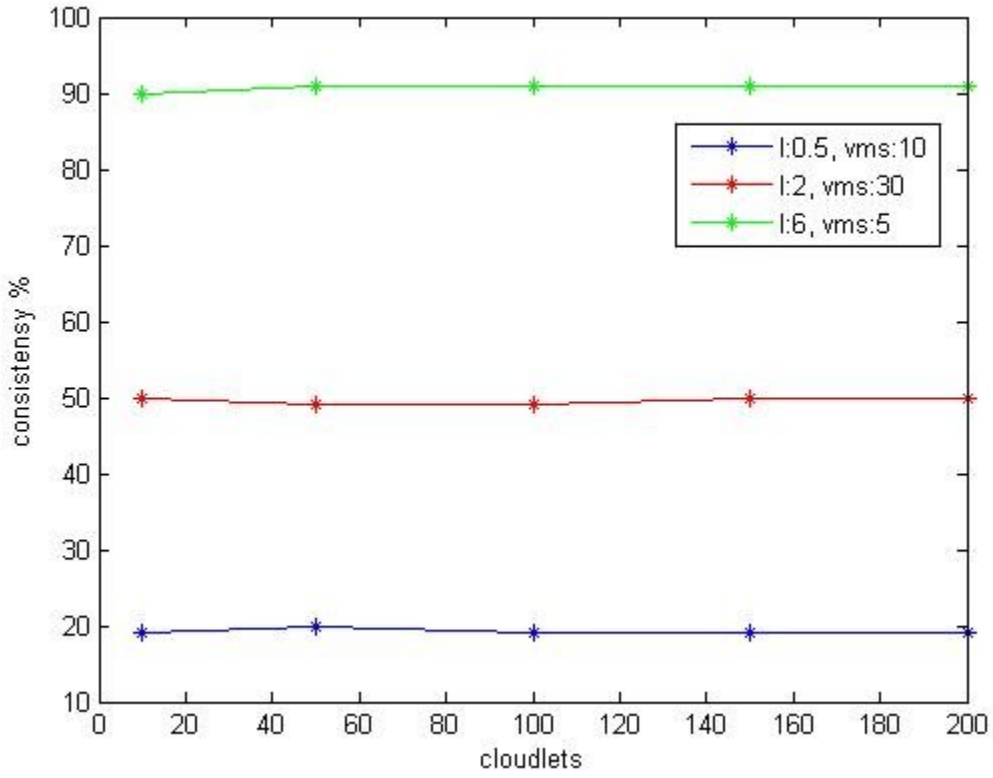
εξυπηρετητών(virtual machines) που χρησιμοποιούμε. Το πείραμα εκτελέστηκε για αριθμό πελατών(cloudlets) ίσο με 150. Για κάθε $\boldsymbol{\lambda}$ παρουσίαζεται η μεταβολή της συνέπειας.



Εικόνα 4. Η συνέπεια που επιτυγχάνεται για διαφορετικό αριθμό εξυπηρετητών(vms)

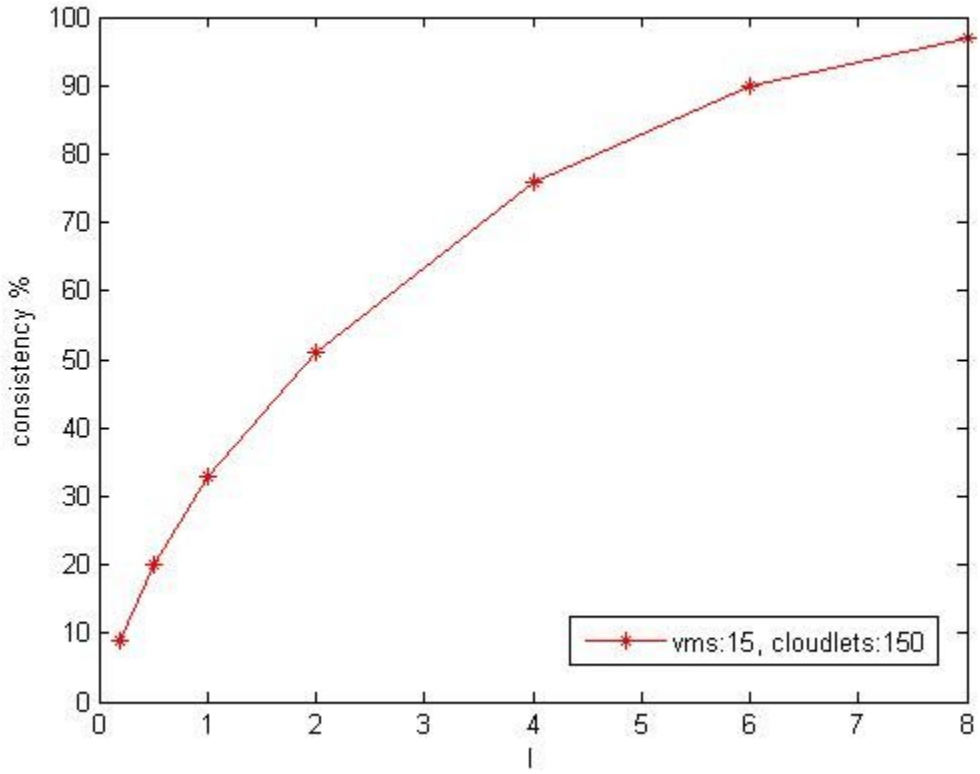
Από το διάγραμμα φαίνεται πως δεν υπάρχει μεταβολή στη συνέπεια που επιτυγχάνεται καθώς αλλάζει ο αριθμός των εξυπηρετητών(vms).

Στη συνέχεια εξετάζουμε τη μεταβολή της συνέπειας όταν αλλάζει ο αριθμός των πελατών(cloudlets). Στο διάγραμμα παρουσίαζονται τα στοιχεία για κάθε πείραμα που εκτελέσαμε. Όπως γίνεται αντιληπτό, η συνέπεια που επιτυγχάνεται καθώς αλλάζει ο αριθμός των πελατών παραμένει σταθερή.



Εικόνα 5. Η συνέπεια που επιτυγχάνεται για διαφορετικό αριθμό πελατών(cloudlets)

Τέλος, παραθέτουμε τα αποτελέσματα των πειραμάτων για τη συνέπεια που επιτυγχάνουμε μεταβάλλοντας τον ρυθμό αλλαγής της έκδοσης του αρχείου. Παρακάτω φαίνεται πως η παράμετρος $\boldsymbol{\lambda}$ της εκθετικής κατανομής, σύμφωνα με την οποία παράγονται οι εκδόσεις των αρχείων, επηρεάζει τη συνέπεια. Το συγκεκριμένο πείραμα εκτελέστηκε για 15 εξυπηρετητές(vms) και 150 πελάτες(cloudlets).



Εικόνα 6. Η συνέπεια που επιτυγχάνεται καθώς μεταβάλεται η παράμετρος λ της εκθετικής κατανομής που καθορίζει το πόσο αργά ή γρήγορα μεταβάλεται το αρχείο.

Παρατηρούμε ότι, καθώς το ***λ*** αυξάνεται η συνέπεια που επιτυγχάνουμε αυξάνεται. Αυτό δικαιολογείται από το γεγονός ότι, καθώς το ***λ*** αυξάνεται οι αλλαγές που συμβαίνουν στο αρχείο είναι μικρότερες, αφού η μέση τιμή της εκθετικής κατανομής που ακολουθούν οι αλλαγές στις εκδόσεις του αρχείου είναι ***1/λ***.

Όπως είναι φανερό από τα αποτελέσματα που παρουσιάσαμε, η συνέπεια που καταφέρνουμε και επιτυγχάνουμε σύμφωνα με τον αλγόριθμο που κατασκευάσαμε εξαρτάται από το πόσο συχνά γίνονται αλλαγές στα δεδομένα. Ο αριθμός των πελατών και των εξυπηρετητών, δεν φαίνεται να επηρεάζει σημαντικά το αποτέλεσμα της συνέπειας.

Κεφάλαιο 6: Βιβλιογραφία

[1] *Raghu Ramakrishnan,* Cap and Cloud Data Management, Computer Magazine, pg. 43-49, February 2012

[2] *Simon S.Y. Shim*, The CAP Theorem's Growing Impact, Computer Magazine, pg. 21-22, February 2012

[3] *Seth Gilbert*, *Nancy A. Lynch*, Perspectives on the CAP Theorem, Computer Magazine, pg. 30-36, February 2012

[4] *Pei Cao*,*Chengjie Liu*, Maintaining Strong Cache Consistency in the World Wide Web, IEEE Transactions on Computers Journal, pg. 445-457, April 1998

[5] *Jian Yin*, *Lorenzo Alvisi*, *Michael Dahlin*, *Calvin Lin*, Volume Leases for Consistency in Large-Scale Systems, IEEE Transactions on Knowledge and Data Engineering Journal, pg. 563-576, July/August 1999

[6] *Venkata Duvvuri*, *Prashant Shenoy*, *Renu Tewari*, Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web, IEEE Transactions on Knowledge and Data Engineering Journal, pg. 834-843, September/October 2003

[7] *Anoop George Ninan*, *Purushottam Kulkarni*, *Prashant Shenoy*, *Krithi Ramamritham*, *Renu Tewari*, Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases, IEEE Transactions on Knowledge and Data Engineering Journal, pg. 813-828, July/August 2003

[8] *Lei Gao*, *Mike Dahlin*, *Jiandan Zheng*, *Lorenzo Alvisi*, *Arun Iyengar*, Dual-Quorum: A Highly Available and Consistent Replication System for Edge Services, IEEE Transactions on Dependable and Secure Computing Journal, pg. 159-174, April-June 2010

[9] http://en.wikipedia.org/wiki/CAP_theorem

[10] *Carl Mazzanti*, *CEO*, *eMazzanti Technologies IT Support and Cloud Computing Services for Small Business Hoboken*, *NJ and NYC*, *201-360-4400,* How Does Cloud Computing Work?, http://www.emazzanti.net/wp-content/uploads/2010/11/How-Does-Cloud-Computing-Work.pdf

[11] http://en.wikipedia.org/wiki/Exponential_distribution

[12] http://www.cloudbus.org/cloudsim/

[13] http://code.google.com/p/cloudsim/

[14] http://blogs.sap.com/innovation/cloud-computing/top-9-challenges-in-cloud-computing-that-are-slowing-its-adoption-011918

[15] http://guide.couchdb.org/draft/consistency.html

[16] http://en.wikipedia.org/wiki/Eventual_consistency
[17] http://pbs.cs.berkeley.edu/

[18] http://en.wikipedia.org/wiki/Cloud_computing

[19] http://www.webopedia.com/TERM/C/cloud_computing.html

[20] http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html

[21] http://www.mpi-sws.org/~druschel/courses/ds/papers/cooper-pnuts.pdf

[22] http://en.wikipedia.org/wiki/Windows_Azure

[23] http://en.wikipedia.org/wiki/Google_Storage