Quantum: Ένα Δίκτυο Ομότιμων Κόμβων για Κατανεμημένους Υπολογισμούς με Ενισχυμένη Ιδιωτικότητα

Γεώργιος Σταματελάτος, Γεώργιος Δροσάτος, Παύλος Εφραιμίδης

Τμήμα Ηλεκτρολόγων Μηχ. & Μηχ. Υπολογιστών, Δημοκρίτειο Πανεπιστήμιο Θράκης
{gs6646, gdrosato, pefraimi}@ee.duth.gr

Abstract

In this paper, a new platform, called Quantum, for distributed computations among independent agents, is presented. Quantum is capable to operate on infrastructures that are formed by massive numbers of agents communicating over the Internet, and is tolerant of adding/removing peers. Furthermore, the new platform is intended to support the privacy of the peers which participate in each distributed computation. In the proposed solution we have chosen a decentralized network architecture and exploited technologies of peer-to-peer networks.

Keywords: privacy, peer-to-peer, distributed computations.

1. Εισαγωγή

Στόχος μας είναι να υποστηρίξουμε κατανεμημένες εφαρμογές μεταξύ ομάδων χρηστών όπου κάθε χρήστης θα αντιπροσωπεύεται από κάποιο προσωπικό agent. Στις εφαρμογές αυτές οι προσωπικοί agent θα πρέπει να μπορούν να εκτελούν κατανεμημένους υπολογισμούς όπου θα χρησιμοποιούνται (χωρίς όμως να αποκαλύπτονται) προσωπικά δεδομένα που φυλάσσονται στους agents. Ένα σύνολο, για παράδειγμα, εκατοντάδων ή ακόμη και χιλιάδων προσωπικών agent θα μπορούν να εκτελούν έναν κατανεμημένο υπολογισμό ώστε να βρεθεί ποιος χρήστης βρίσκεται πιο κοντά σε κάποια τοποθεσία. Αφορμή για τη μελέτη τέτοιου είδους προβλημάτων αποτέλεσε το έργο Polis [Efraimidis et.al (2009)] στο οποίο αναπτύσσονται τεχνολογίες όπου κάθε χρήστης διαχειρίζεται ο ίδιος τα προσωπικά του δεδομένα και η πρόσβαση σε αυτά γίνεται αποκεντρωμένα με συμφωνίες μεταξύ των agents. Η ανάγκη για την εκτέλεση σύνθετων υπολογισμών με τη συμμετοχή μεγάλου πλήθους κόμβων του Polis οδήγησε στην ανάπτυξη του Quantum. Βασικό χαρακτηριστικό του Quantum είναι ότι η λειτουργία του δε βασίζεται σε κεντρικό

διακομιστή αλλά η επικοινωνία μεταξύ των χρηστών του είναι τελείως αποκεντροποιημένη, γεγονός που ενισχύει την ιδιωτικότητά τους.

Για την οργάνωση των κόμβων/agents σε τοπολογία επιλέγουμε την αποκεντρωμένη οργάνωση όπου όλοι οι κόμβοι συμμετέχουν ισότιμα. Ο λόγος είναι ότι έτσι εξασφαλίζεται η επεκτασιμότητα (scalability) του δικτύου ενώ μειώνουμε τους κινδύνους για την ιδιωτικότητα των κόμβων.

2. Σχετικές εργασίες

Αρκετές είναι οι υλοποιήσεις και τα πρωτόκολλα που έχουν προταθεί για δίκτυα ομότιμων κόμβων (peer-to-peer networks). Τα πρωτόκολλα Chord [Stoica et.al (2001)], Kademlia [Maymounkov et.al (2002)] και Freenet [Clarke et.al (1999)] που θα παρουσιαστούν παρακάτω αποτελούν κάποιες από τις υπάρχουσες ιδέες που εφαρμόζονται σήμερα.

2.1 Chord

Το *Chord* αποτελεί ένα πρωτόκολλο ομότιμου δικτύου που στοχεύει στην γρήγορη και αποτελεσματική αναζήτηση μεταξύ των κόμβων. Το Chord οργανώνει τους κόμβους του δικτύου γύρω από ένα νοητό δακτύλιο, βάση ενός μοναδικού ID που αντιστοιχεί σε κάθε κόμβο. Η γρήγορη αναζήτηση επιτυγχάνεται με τη χρήση αναφορών σε πολλαπλούς κόμβους σταθερής θέσης (Finger Tables), που αποτελούν σημεία με αποστάσεις τις δυνάμεις του δύο, έτσι ώστε να λογαριθμείται ο χρόνος που χρειάζεται να ταξιδέψει ένα αίτημα γύρω από τον κύκλο. Το Quantum υιοθετεί τις παραπάνω τεχνολογίες για την επίτευξη των στόχων του.

2.2 Kademlia

Τα δίκτυα *Kademlia* διατηρούν την τυπική δομή δικτύων ομότιμων κόμβων που συναντάμε σε δίκτυα, όπως το Chord. Η κύρια διαφορά του Kademlia έγκειται στη χρήση δενδρικής δομής προκειμένου να οργανωθούν οι κόμβοι του.

2.3 Freenet

Το *Freenet* είναι ένα πρωτόκολλο αποκεντροποιημένης επικοινωνίας που επικεντρώνεται στην ιδιωτικότητα και την ανωνυμία των σταθμών του δικτύου. Κάθε κόμβος προστατεύει τα προσωπικά του στοιχεία με τη βοήθεια ενός ζεύγους δημοσίου-ιδιωτικού κλειδιού, διατηρώντας συνδέσεις ασφαλούς επικοινωνίας με τις αναφορές του.

3. Προσέγγιση - Περιγραφή

Το πρωτόκολλο Quantum που προτείνεται στην παρούσα εργασία βασίζεται σε ιδέες των προαναφερθέντων πρωτοκόλλων: Η βασική δομή του, όπως αναφέρθηκε, στηρίζεται στην αντίστοιχη του Chord. Για τις ανάγκες των πειραμάτων πάνω σε δίκτυα Quantum χρησιμοποιείται η δομή δέντρου των δικτύων Kademlia. Επιπρόσθετα, σε συνδυασμό με κατάλληλους αλγόριθμους κατανεμημένων υπολογισμών επιτυγχάνεται η προστασία της ιδιωτικότητας των κόμβων. Απαραίτητη προϋπόθεση για την διασφάλιση της ιδιωτικότητας, εκτός από τους ίδιους τους αλγορίθμους, είναι ότι οι κόμβοι είναι έντιμοι αλλά ταυτόχρονα μπορεί να είναι περίεργοι (Honest-But-Curious - HBC [Acquisti et.al (2008), pp.48-50]). Τα κύρια χαρακτηριστικά του πρωτοκόλλου παρουσιάζονται στην επόμενη υποενότητα.

3.1 Χαρακτηριστικά δικτυακής οργάνωσης

Η δικτυακή αρχιτεκτονική του Quantum περιλαμβάνει βασικά χαρακτηριστικά του Chord, όπως η ένταξη των κόμβων σε δακτύλιο, η διατήρηση fingers, η σταθεροποίηση των αναφορών καθώς επίσης και είσοδος και έξοδος κόμβων από το δίκτυο. Ενδιαφέρον παρουσιάζει ο αλγόριθμος αναζήτησης ενός κλειδιού. Ένας κόμβος που θα λάβει/εκκινήσει ένα αίτημα αναζήτησης θα προωθήσει το αίτημα στην μεγαλύτερη αναφορά του που είναι μικρότερη από το ζητούμενο κλειδί. Σε αντίθεση με πρωτόκολλα peer-to-peer δικτύων, το Quantum δεν ασχολείται με αποθήκευση δεδομένων και τα κλειδιά χρησιμοποιούνται μόνο για τη δρομολόγηση των πακέτων.

3.1.1 Δομή δακτυλίου

Το Quantum, όπως και το Chord, δομείται στο χαμηλότερο στάδιο με ένα νοητό δακτύλιο πάνω στον οποίο τοποθετούνται όλοι οι κόμβοι/agents του δικτύου. Η σειρά με την οποία θα τοποθετηθούν καθορίζεται από το κλειδί (ID) των κόμβων. Το ID ενός κόμβου είναι ένα μοναδικό χαρακτηριστικό του και στην περίπτωση του Quantum υπολογίζεται βάση του SHA-1 hash του συνδυασμού IP:Port (κάθε agent έχει ένα port ανοιχτό στο οποίο "δέχεται" αιτήματα από άλλους agents). Έτσι, με διαδικασίες που περιγράφονται πιο κάτω, οι κόμβοι τείνουν να τοποθετηθούν γύρω από το νοητό αυτό δακτύλιο με αύξουσα σειρά modulo n, όπου n το μέγιστο πλήθος κόμβων που μπορεί να υποστηρίξει το δίκτυο. Σύμφωνα με αυτή την αρχιτεκτονική, κάθε agent διατηρεί αναφορές στον αμέσως επόμενο του στον δακτύλιο (successor, αναφέρεται succ()), καθώς επίσης και για τον αμέσως προηγούμενό του (predecessor, αναφέρεται pred()). Η δικτυακή αυτή οργάνωση αποτελεί μια βασική δομή πάνω στην οποία μπορεί να θεμελιωθεί ένα ομότιμο δίκτυο.

3.1.2 Διαδοχικοί κόμβοι

Πολλές φορές παρατηρούνται φαινόμενα αποχώρησης κόμβων από το δίκτυο, πράγμα το οποίο σημαίνει και διάσπαση του δακτυλίου που αναφέρθηκε πιο πάνω. Για το λόγο αυτό, και προς όφελος του αξιοπιστίας, δημιουργούμε μια προέκταση

του δακτυλίου κατά την οποία κάθε agent διατηρεί, εκτός από τους κοντινότερους του κόμβους, αναφορές και σε ένα ορισμένο αριθμό κόμβων ακριβώς μετά από αυτόν. Με τον τρόπο αυτό, όταν ένας agent αντιληφθεί ότι ο successor του δεν είναι προσβάσιμος, θα ορίσει ως successor τον επόμενο κόμβο που βρίσκεται στις διαδοχικές αναφορές του. Στην παράγραφο 3.1.6 περιγράφεται η διαδικασία δημιουργίας του πίνακα που κρατάει αυτές τις αναφορές.

3.1.3 Fingers

Συχνά ένα ομότιμο δίκτυο αποτελείται από ένα πολύ μεγάλο αριθμό κόμβων. Σε μια τέτοια περίπτωση η δικτυακή αρχιτεκτονική που αναφέρθηκε πιο πάνω, παρότι αξιόπιστη, δεν είναι επαρκής για την επεκτασιμότητα του δικτύου κάτω από τέτοιες συνθήκες και αυτό διότι ο χρόνος ταξιδιού ενός πακέτου γύρω από το δακτύλιο είναι ανάλογος με τον αριθμό των κόμβων. Τη λύση σε αυτό το πρόβλημα προσφέρει η δικτυακή οργάνωση των fingers του Chord, μέσω της οποίας επιτυγχάνουμε την επιθυμητή επεκτασιμότητα. Σύμφωνα με αυτή, κάθε agent διατηρεί έναν πίνακα ο οποίος περιέχει τις διευθύνσεις των κόμβων με εκθετικά αυξανόμενο ID από τον αρχικό. Πιο συγκεκριμένα, ο κόμβος με κλειδί S θα διατηρήσει αναφορές για τους κόμβους με κλειδιά $S+2^k$, με $k \geq 1$ και με όριο το μέγιστο αριθμό bits του κλειδιού (πχ στην περίπτωση του SHA-1 είναι 160). Αξίζει να σημειωθεί ότι για k=0 η αναφορά ισοδυναμεί με τον successor. Στην παράγραφο 3.1.6 περιγράφεται η διαδικασία δημιουργίας και ενημέρωσης των αναφορών αυτών.

3.1.4 Αίτημα αναζήτησης κλειδιού

Η αναζήτηση στο δίκτυο αφορά την εύρεση ενός κόμβου που χαρακτηρίζεται από ένα συγκεκριμένο κλειδί. Εάν δεν υπάρχει κόμβος με το εν λόγω κλειδί, το δίκτυο επιστρέφει τον κόμβο με το αμέσως επόμενο διαθέσιμο κλειδί. Το αίτημα αποτελείται από τα εξής πεδία: τον κόμβο S που ξεκίνησε το αίτημα, το κλειδί K, το οποίο ζητάμε καθώς επίσης και έναν ακέραιο ttl (time-to-live) ο οποίος περιορίζει τη διάρκεια ζωής του αιτήματος στο δίκτυο. Για την εύρεση του κλειδιού, το αίτημα διασχίζει το δίκτυο με τον ακόλουθο τρόπο:

Κάθε κόμβος M που λαμβάνει το αίτημα πραγματοποιεί τα παρακάτω βήματα:

- Εάν το ttl είναι μηδέν, αγνοεί πλήρως το αίτημα και τερματίζει, αλλιώς το μειώνει κατά ένα.
- Ο κόμβος M πράττει ανάλογα με τις εξής περιπτώσεις:
    - Εάν το K είναι το κλειδί του κόμβου M, τότε αποστέλλει στον S ένα πακέτο επιβεβαίωσης εύρεσης του κλειδιού.
    - Εάν το K βρίσκεται μεταξύ του pred(M) και M, τότε αποστέλλει και πάλι πακέτο επιβεβαίωσης στον S.
    - Εάν το K βρίσκεται μεταξύ του M και succ(M), προωθεί το αίτημα στον succ(M).

- Δημιουργεί μια ταξινομημένη κατά ID λίστα που περιέχει όλους τους κόμβους που βρίσκονται τόσο στο Finger Table όσο και στον πίνακα των διαδοχικών κόμβων. Βρίσκει τον κόμβο P της παραπάνω λίστας που έχει το αμέσως μεγαλύτερο ID από το K και προωθεί το αίτημα στον κόμβο P.

Σύμφωνα με τα παραπάνω, το μέγιστο πλήθος αναπηδήσεων που μπορεί να κάνει ένα αίτημα μέσα από το δίκτυο είναι $\log_2 n$.

3.1.5 Διαδικασία εισόδου κόμβου στο δίκτυο

Στην παράγραφο αυτή παρουσιάζεται η διαδικασία που ακολουθείται ώστε να εισέλθει ένας agent στο ομότιμο δίκτυο του Quantum. Απαραίτητη προϋπόθεση είναι ο προς εισαγωγή agent S να γνωρίζει τα στοιχεία επικοινωνίας (διεύθυνση IP και Port) ενός οποιουδήποτε άλλου κόμβου M που ανήκει στο δίκτυο. Ο S στέλνει στον M ένα πακέτο αίτησης εισχώρησης στο δίκτυο μαζί με κάποια χαρακτηριστικά του agent, όπως η θύρα (port) στην οποία λαμβάνει μηνύματα. Εφόσον ο M λάβει το πακέτο, θα στείλει πίσω ένα πακέτο "δοκιμής" στη θύρα που ακούει ο S, στο οποίο ο S είναι υποχρεωμένος να απαντήσει, επιβεβαιώνοντας έτσι τη δυνατότητά του να λάβει μηνύματα. Ο S θέτει τον M ως successor του, δεδομένου ότι είναι ο μοναδικός agent που γνωρίζει και τον ενημερώνει σχετικά. Εάν ο M δεν έχει predecessor ή εάν το ID του pred(M) είναι μικρότερο από το ID του S, τότε τον δέχεται ως predecessor και η διαδικασία εισαγωγής του S έχει ολοκληρωθεί. Μπορεί να σημειωθεί ότι σε αυτό το σημείο δεν έχει επέλθει πλήρης σταθεροποίηση στο δίκτυο, μιας και κανένας άλλος κόμβος πέραν του M δε γνωρίζει την ύπαρξη του S. Στην επόμενη ενότητα αναλύεται ο τρόπος με τον οποίο το δίκτυο σταθεροποιείται.

3.1.6 Σταθεροποίηση δικτύου

Ανά τακτά χρονικά διαστήματα, των οποίων η διάρκεια ορίζεται από το χειριστή του agent (έστω S), εκτελούνται 3 διαφορετικά ήδη σταθεροποίησης (stabilization), τα οποία περιγράφονται στη συνέχεια.

Σταθεροποίηση Successor. Η σταθεροποίηση του successor γίνεται με δύο τρόπους.

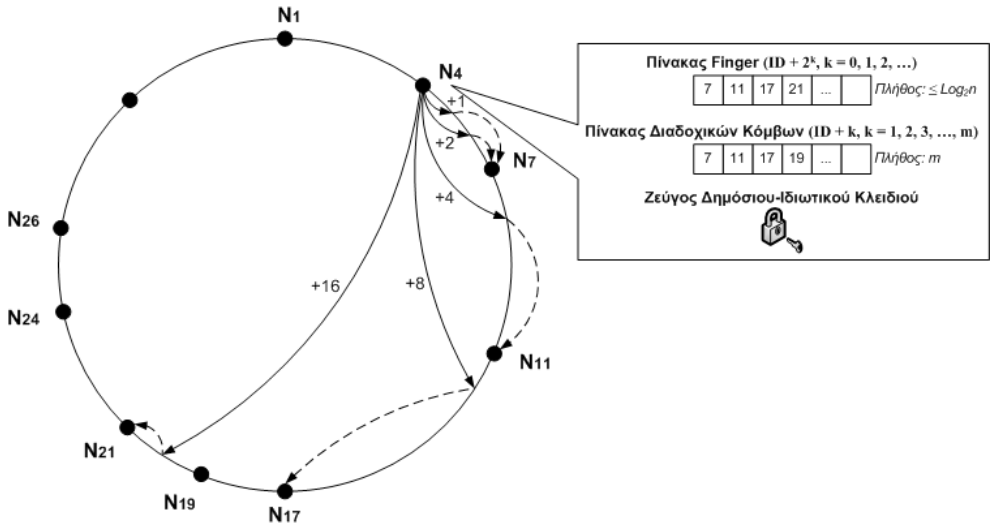
Σταθεροποίηση με έλεγχο predecessor. Ο S στέλνει ένα πακέτο στον succ(S), ζητώντας να του επιστρέψει τον Predecessor του (pred(succ(S) = K). Εάν ο K βρίσκεται μεταξύ του S και του succ(S), ο S θα θέσει ως νέο του Successor τον K και θα ενημερώσει τον K σχετικά. Ο K έχει τη δυνατότητα να δεχθεί τον S ως Predecessor του αναλόγως με τη σχετική του θέση ως προς τον τρέχων του Predecessor.

Σταθεροποίηση successor με αίτημα. Ο S εκκινεί ένα αίτημα έρευσης του κλειδιού με αριθμό ID(S)+1 στο δίκτυο (η διαδικασία περιγράφεται σε επόμενη παράγραφο). Εάν η απάντηση M που θα δεχθεί από το δίκτυο αποτελεί έναν κόμβο πιο κοντά στον

succ(S), ο successor θα αντικατασταθεί με τον M. Η μέθοδος αυτή είναι αντικειμενικά πιο γρήγορη από τη μέθοδο με έλεγχο predecessor, αλλά εκτελείται με πιο αργούς ρυθμούς δεδομένης της ενδεχόμενης καταπόνησης του δικτύου.

Σταθεροποίηση Διαδοχικών Κόμβων. Μετά την ένταξή του στο δίκτυο, ο S διαθέτει έναν πίνακα διαδοχικών κόμβων, που στο ξεκίνημα της λειτουργίας του περιέχει μόνο τον successor του. Κατά τη διαδικασία της εν λόγω σταθεροποίησης, ο S επιλέγει τυχαία έναν κόμβο M από αυτό τον πίνακα και του αποστέλλει ένα ερώτημα, με το οποίο ζητάει να τον πληροφορήσει για τον succ(M). Στη συνέχεια, ο S θα προσθέσει τον succ(M) στον πίνακα των διαδοχικών κόμβων και θα αφαιρέσει (εάν το μέγεθός του υπερβαίνει αυτό που έχει ορίσει ο χειριστής του agent) τον κόμβο με τη μεγαλύτερη απόσταση από τον εαυτό του.

Σταθεροποίηση Fingers. Η λειτουργία αυτή επιτυγχάνεται αποκλειστικά με ερωτήματα στο δίκτυο. Ο N διατηρεί έναν πίνακα (όπως και στην περίπτωση των διαδοχικών) με τα fingers του. Ο πίνακας αυτός έχει σταθερό μέγεθος ίσο με τον αριθμό των bits των IDs του δικτύου. Κατά τη σταθεροποίηση αυτή, ο N θα επιλέξει τυχαία έναν ακέραιο k τέτοιος ώστε $0 \le k < n$ και ξεκινάει ένα ερώτημα εύρεσης του $S+2^k+1$. Ο N προσθέτει τον κόμβο - απάντηση (έστω M) στον πίνακα των fingers εάν στο διάστημα $(S+2^k , S+2^{(k+1)})$ δεν εμπεριέχεται άλλος κόμβος ή αν αυτός που εμπεριέχεται έχει μεγαλύτερο ID από τον M. Στην τελευταία περίπτωση ο ήδη υπάρχον αφαιρείται από τον πίνακα.



Εικόνα 1. *Παράδειγμα δικτύου Quantum με προβολή των δεδομένων του κόμβου N4*

3.2 Κατανεμημένοι υπολογισμοί

Η διαφοροποίηση του Quantum, από τις τεχνολογίες που αναφέρθηκαν, έγκειται στην δυνατότητα εκτέλεσης αποδοτικών κατανεμημένων υπολογισμών μεταξύ μεγάλου πλήθους κόμβων-χρηστών ενώ παράλληλα διευκολύνεται η προστασία της ιδιωτικότητας αυτών. Η συνεισφορά του Quantum στην προστασία της ιδιωτικότητας έγκειται στην παροχή στοιχείων που βοηθούν τους κατανεμημένους αλγορίθμους, όπως για παράδειγμα ζεύγη δημοσίου/ιδιωτικού κλειδιού για κάθε κόμβο. Ένα τέτοιο παράδειγμα κατανεμημένου υπολογισμού είναι το πρόβλημα του εκατομμυριούχου (Millionaire's Problem), όχι μόνο ανάμεσα σε δύο κόμβους, αλλά του συνόλου που λαμβάνουν μέρος στο δίκτυο. Αξίζει να σημειωθεί ότι κατά τη διάρκεια ενός κατανεμημένου υπολογισμού έχουμε θεωρήσει αξιόπιστους κόμβους οι οποίοι δεν αποτυγχάνουν.

Το ξεκίνημα μιας τέτοιας διαδικασίας μπορεί να κάνει οποιοσδήποτε κόμβος ανήκει στο δίκτυο, αποστέλλοντας ένα broadcast μήνυμα αναφέροντας στους κόμβους το πρωτόκολλο το οποίο θα τρέξουν. Κάθε κόμβος που λαμβάνει το συγκεκριμένο μήνυμα είναι έτοιμος να εκκινήσει την προσωπική του διαδικασία για την ολοκλήρωση του κατανεμημένου υπολογισμού.

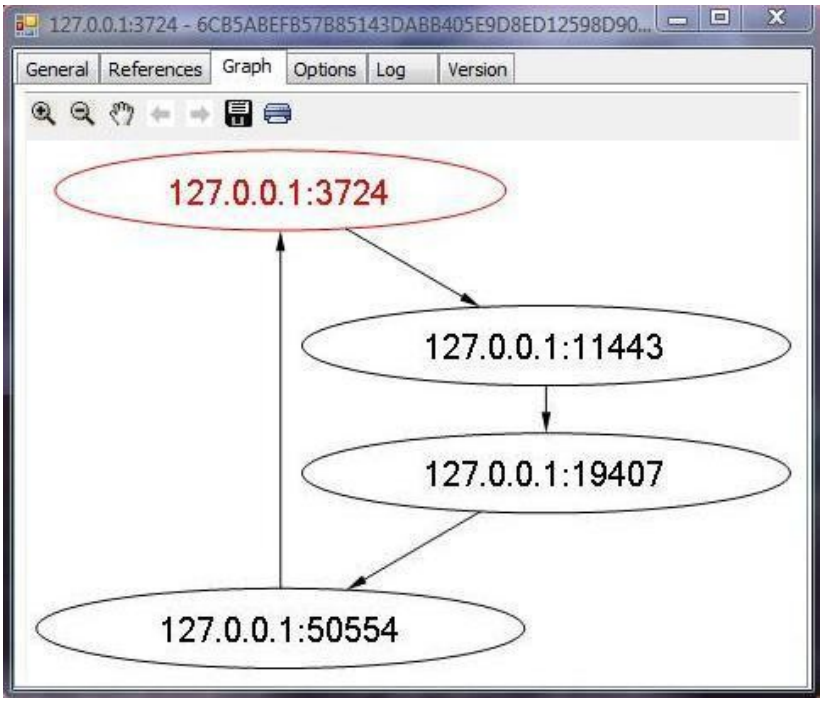
Ο μηχανισμός του κατανεμημένου υπολογισμού αποτελείται από ένα δυαδικό δένδρο, στα φύλλα του οποίου εκτελείται η πρώτη φάση του υπολογισμού κατά την οποία οι κόμβοι εκτελούν τον υπολογισμό ανά δύο (εάν βέβαια υπάρχουν δύο κόμβοι σε διαδοχικά κλειδιά του δέντρου) ενώ το ενδιάμεσο αυτό αποτέλεσμα προωθείται σε ανώτερο επίπεδο του δέντρου. Η διαδικασία συνεχίζεται έως ότου γίνουν όλοι οι απαραίτητοι υπολογισμοί ώστε να επικρατήσει ένα και μοναδικό αποτέλεσμα (στην κορυφή του δυαδικού δέντρου).

3.3 Υπηρεσίες μεσολάβησης

Το Quantum προϋποθέτει υπηρεσίες που θα βοηθούν έναν αυτόνομο agent να ενταχθεί στο δίκτυο, σε περίπτωση που αυτός δεν διαθέτει κάποιο τρόπο επικοινωνίας με τους άλλους κόμβους του ομότιμου δικτύου, καθώς επίσης και να μεσολαβεί για την εκτέλεση ενός κατανεμημένου υπολογισμού σε περιπτώσεις που ένας ανεξάρτητος agent δεν επιθυμεί την ένταξη στο δίκτυο για την εκκίνηση ενός υπολογισμού. Ο εν λόγω κόμβος, μετά από αίτημα, θα εκκινεί τη διαδικασία του κατανεμημένου υπολογισμού εκ μέρους του ανεξάρτητου κόμβου και θα προωθεί στον τελευταίο το τελικό αποτέλεσμα εφόσον η λειτουργία ολοκληρωθεί.

4. Πλατφόρμα πειραμάτων - Συμπεράσματα

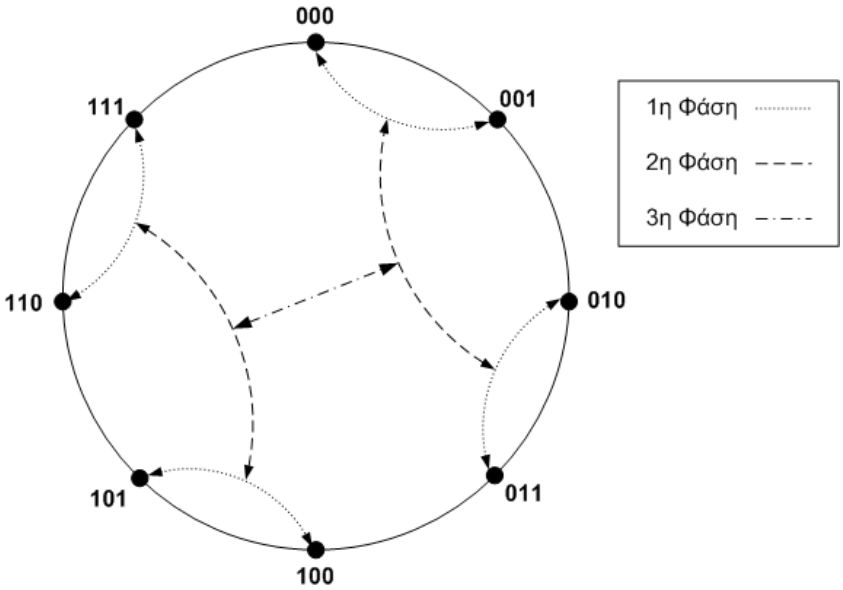
Για την παραπάνω δικτυακή αρχιτεκτονική, αναπτύχθηκε μια πιλοτική εφαρμογή σε προγραμματιστικό περιβάλλον .NET, η οποία αντιπροσωπεύει έναν agent-κόμβο (Εικόνα 2). Δοκιμές πάνω στη λειτουργικότητα της εφαρμογής έχουν γίνει τοπικά, στο ίδιο υπολογιστή, με ταυτόχρονη λειτουργία αρκετών agents.



Εικόνα 2. *Γραφική απεικόνιση του δικτύου από έναν agent*

Επιπρόσθετα, πραγματοποιήθηκε ένας απλός κατανεμημένος υπολογισμός, ο οποίος αποτελεί γενίκευση (για περισσότερα από δύο άτομα) του κρυπτογραφικού πρωτοκόλλου του προβλήματος των εκατομμυριούχων (Millionaire's Problem) του Yao [Yao (1982)]. Με το συγκεκριμένο πρωτόκολλο δίνεται η δυνατότητα σε δύο εκατομμυριούχους να συγκρίνουν τις περιουσίες τους και να αποφανθούν για το ποιος είναι ο πλουσιότερος χωρίς όμως σε κανένα στάδιο της διαδικασίας να αποκαλυφθεί η πραγματική τους περιουσία. Μια στοιχειώδης λύση για δύο άτομα μπορεί να γενικευτεί έτσι ώστε να μπορεί να τρέξει μεταξύ των N κόμβων του δικτύου Quantum. Τα βασικά βήματα αυτού του αλγορίθμου είναι τα εξής:

- Το δίκτυο Quantum με broadcast προωθεί το αίτημα για την έναρξη του πρωτοκόλλου σε όλους τους agents (N_1,N_2,…,N_N).
- Οι agents με βάση την δυαδική μορφή της αρίθμησης (κλειδιών) μέσα στο Quantum μπορούν να οργανωθούν σε $\log_2 n$ φάσεις και να τρέξουν με την μορφή δέντρου το πρωτόκολλο των εκατομμυριούχων, όπως φαίνεται ενδεικτικά στην Εικόνα 3.
- Σε κάθε φάση οι νικητές τρέχουν το πρωτόκολλο μεταξύ τους μέχρι να βρεθεί ο πιο πλούσιος που υπάρχει στο δίκτυο.



Εικόνα 3. *Δικτυακή τοπολογία για την οργάνωση N=8 κόμβων κατά τη εκτέλεση του προβλήματος των εκατομμυριούχων*

Τα πρώιμα συμπεράσματα δείχνουν ότι ο μηχανισμός συγκρότησης του δικτύου επιτυγχάνεται, συμπεριλαμβανομένου του δακτυλίου, των fingers και των διαδοχικών κόμβων. Ο αλγόριθμος δρομολόγησης των αιτημάτων που υλοποιήθηκε, ακολουθεί τα πρότυπα που αναφέρθηκαν στην ενότητα 3.1.6. Ωστόσο, η εκτέλεση του παραπάνω απλού κατανεμημένου υπολογισμού έδειξε ότι για την διασφάλιση της ιδιωτικότητας των κόμβων σημαντικό ρόλο έπαιξε τόσο η τοπολογία του ίδιου του δικτύου όσο και το πρωτόκολλο του υπολογισμού.

5. Μελλοντικές εργασίες

Το Quantum είναι ένα έργο σε εξέλιξη και έχει υλοποιηθεί μέρος των δυνατοτήτων που απαιτούνται. Παρακάτω αναφέρουμε ορισμένα χαρακτηριστικά που θα θέλαμε να προστεθούν στη λειτουργικότητα του Quantum.

5.1 Δυναμική αποστολή πρωτοκόλλου

Ένας κατανεμημένος υπολογισμός στο Quantum μπορεί να επιτευχθεί εάν όλοι οι κόμβοι του δικτύου έχουν γνώση του αλγορίθμου του υπολογισμού εκ των προτέρων. Αυτό θεωρείται περιορισμός του Quantum και μπορεί να λυθεί με σύγχρονες τεχνολογίες προγραμματισμού, όπως η ανταλλαγή του ίδιου του αλγόριθμου (π.χ. script ή binary κώδικα) μεταξύ των κόμβων κατά τη διάρκεια εκτέλεσης του υπολογισμού, ο οποίος θα υλοποιεί τον εν λόγω υπολογισμό.

Η παραπάνω λύση, όμως, δημιουργεί ένα σημαντικό μειονέκτημα. Η αποστολή του αλγορίθμου του υπολογισμού από κάποιον κόμβο στο υπόλοιπο δίκτυο θέτει σε κίνδυνο την ιδιωτικότητα των υπολοίπων agents, μιας και αυτοί ενδέχεται να εκτελέσουν αλγόριθμο που δεν τηρεί τα πρότυπα της ιδιωτικότητας.

5.2 Υπηρεσίες πιστοποίησης πρωτοκόλλων

Λύση στο παραπάνω πρόβλημα μπορούν να προσφέρουν αξιόπιστες υπηρεσίες, με σκοπό την πιστοποίηση των κατανεμημένων αλγορίθμων που εκτελούνται στο δίκτυο. Ένας κόμβος θα εκτελεί κάποιον αλγόριθμο μόνο εάν αυτός έχει προηγουμένως πιστοποιηθεί από την αξιόπιστη αυτή υπηρεσία.

5.3 Άρση παραδοχών

Εξετάζεται η δυνατότητα να προστεθούν μέθοδοι χειρισμού της συμπεριφοράς των κόμβων, έτσι ώστε να άρουμε τις διάφορες παραδοχές που έχουμε θεωρήσει. Έτσι, το Quantum αναμένεται να υποστηρίξει κόμβους οι οποίοι δεν λειτουργούν πάντα προς όφελος του δικτύου καθώς επίσης και μη αξιόπιστους κόμβους κατά τη διάρκεια του κατανεμημένου υπολογισμού. Πιο συγκεκριμένα, το Quantum αναμένεται να είναι σε θέση να εντοπίσει κακόβουλους agents που σαν σκοπό έχουν την παραπλάνηση άλλων και τη δυσλειτουργία της δικτυακής οργάνωσης. Επιπρόσθετα, θα πρέπει να βρεθεί λύση στο πρόβλημα της θεώρησης αξιόπιστων κόμβων κατά τη διάρκεια ενός υπολογισμού, δεδομένου ότι μία αποτυχία ενός κόμβου μπορεί να προκαλέσει εσφαλμένο αποτέλεσμα στον υπολογισμό, ακόμα και την αδυναμία τερματισμού του.

Αναφορές

1. Acquisti A., Gritzalis S., Lambrinoudakis C. and De Capitani di Vimercati S. (2008), *Digital privacy*, Auerbach Publications.
2. Clarke I., Sandberg O., Wiley B. and Hong T.W. (1999), *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, In ICSI Workshop on Design Issues in Anonymity and Unobservability.
3. Efraimidis P., Drosatos G., Nalbadis F. and Tasidou A. (2009), *Towards Privacy in Personal Data Management*, (accepted in) Information Management & Computer Security, 17(4).
4. Maymounkov P. And Mazieres D. (2002), *Kademlia: A peer-to-peer information system based on the XOR metric*, In Proc. of IPTPS, Cambridge, MA, pp. 53-65.
5. Stoica I., Morris R., Karger D., Kaashoek M. F. and Balakrishnan H. (2001), *Chord: A scalable peer-to-peer lookup service for Internet applications*, In Proc. ACM SIGCOMM'01, San Diego, CA.
6. Yao A. C. (1982), *Protocols for Secure Computations (extended abstract)*, Proceedings of the 21st Annual IEEE Symposium on the Foundations of Computer Science, pp. 160-164.