ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Στρατηγικές για caching βασισμένες σε προτιμήσεις χρηστών σε περιβάλλοντα με blockchain

Γεώργιος Καραολάνης

Επιβλέπων: **Δημήτριος Κατσαρός**

2ο Μέλος Επιτροπής: **Εμμανουήλ Βάβαλης**

ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία υλοποιήθηκαν και αξιολογήθηκαν στρατηγικές για caching βασισμένες σε προτιμήσεις χρηστών σε περιβάλλοντα όπου χρησιμοποιούνται εφαρμογές blockchain. Για τον σκοπό αυτό προσομοιώθηκε ένα δίκτυο κόμβων που τρέχουν μια εφαρμογή blockchain και χρησιμοποιούν διάφορες στρατηγικές για την αξιολόγηση των block και για την διαχείριση του διαθέσιμου χώρου της τοπικής τους μνήμης όπου αποθηκεύονται αυτά. Σε αυτή τη προσομοίωση κάθε κόμβος έχει τα δικά του κριτήρια επιλογής για τα blocks που έχουν την μεγαλύτερη αξία για αυτόν. Υπάρχουν συνολικά τέσσερις διαφορετικές κατηγορίες κόμβων, εκ των οποίων κάποιοι αποθηκεύουν ολόκληρα τα blocks ενώ άλλοι κρατάνε απλά την χρήσιμη για αυτούς πληροφορία. Επίσης υλοποιήθηκε ένα πρωτόκολλο επικοινωνίας ώστε να είναι εφικτή η ανταλλαγή πληροφοριών και η συνεργασία μεταξύ των κόμβων. Στο παραπάνω δίκτυο αξιολογήθηκαν τα αποτελέσματα από τρεξίματα της εφαρμογής για τις διάφορες παραμέτρους της, τις στρατηγικές για caching καθώς και τις διαθέσιμες ρυθμίσεις για τους κόμβους. Τα αποτελέσματα αυτά περιλαμβάνουν πληροφορίες για τον αριθμό και το συνολικό μέγεθος των αποθηκευμένων blocks σε κάθε κόμβο καθώς και το ποσοστό της προσωπικά χρήσιμης πληροφορίας στην τοπική μνήμη για τον εκάστοτε κόμβο. Από αυτά μπορούμε να βγάλουμε συμπεράσματα για τις πιο κατάλληλες παραμέτρους της εφαρμογής και για τις πιο αποτελεσματικές στρατηγικές caching για κάθε κόμβο.

Λέξεις κλειδιά : Blockchain, Block, Caching

ABSTRACT

In this bachelor's thesis user interest driven caching strategies in environments where blockchain applications are used, were studied. For this purpose, a network of nodes running a blockchain application, where each node uses different caching strategies to evaluate blocks and to manage the available space in their local memory, was simulated. In this simulation each node has its own criteria, based on which it chooses the blocks which have the greatest personal value. There are four different types of nodes, where some of them store whole blocks in their local memory, while others just store the information from the blocks which is important to them. Furthermore, a communication protocol was designed, in order to allow for information exchange and cooperation between the nodes. In the above described network, results from running the application for its different parameters, caching strategies and node configurations were recorded and evaluated. These results include information about the number and the overall size of the saved blocks in each node as well as the percentage of the personally useful data each node has in its local memory in comparison to the overall data that it stores. From those results, conclusions about the most suitable application parameters and the most effective caching strategies can be made.

Keywords : Blockchain, Block, Caching

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1 Ορισμός του blockchain

Το blockchain είναι μια κατανεμημένη δομή αποθήκευσης πληροφοριών, η οποία διαμοιράζεται σε κόμβους ενός δικτύου ομότιμων χρηστών. Αποτελείται από μια αλυσίδα από εγγραφές, οι οποίες ονομάζονται blocks. Στο blockchain, κάθε block συνδέεται με το προηγούμενο και με το επόμενο block μέσω ενός συνδέσμου, δημιουργώντας έτσι την αλυσίδα. Όταν δημιουργηθεί ένα καινούριο block αυτό συνδέεται στο τέλος της αλυσίδας. Οι σύνδεσμοι αυτοί είναι κρυπτογραφικοί, που σημαίνει πως οι εγγραφές συνδέονται μεταξύ τους με έναν προγραμματιστικό μηχανισμό ασφαλείας ο οποίος εντοπίζει και αποτρέπει την οποιαδήποτε αλλαγή σε μια εγγραφή, από τη στιγμή που αυτή έχει καταχωρηθεί στην αλυσίδα.

Λόγω του παραπάνω χαρακτηριστικού, το blockchain προσφέρει υψηλό επίπεδο ασφαλείας όσον αφορά στην ακεραιότητα των εγγραφών που βρίσκονται σε αυτό. Με αυτόν τον τρόπο το blockchain μπορεί να χρησιμοποιηθεί σε ένα δίκτυο ομότιμων χρηστών, σαν μια βάση δεδομένων η οποία αποθηκεύει όλες τις ενέργειες οι οποίες έχουν πραγματοποιηθεί στο δίκτυο από τους χρήστες.

Το blockchain στην παραπάνω μορφή σχεδιάστηκε από τον Satoshi Nakamoto [1], ο οποίος το χρησιμοποίησε σαν βάση για το ψηφιακό νόμισμα Bitcoin, για να αποθηκεύονται σε αυτό οι συναλλαγές που γίνονται μεταξύ των χρηστών που ανήκουν στο δίκτυο.

1.2 Παραδείγματα χρήσης του blockchain

Η πιο ευρέως γνωστή εφαρμογή που κάνει χρήση του blockchain είναι το ψηφιακό νόμισμα Bitcoin. Ταυτόχρονα με αυτό, έχει εμφανιστεί ένα μεγάλο πλήθος άλλων ψηφιακών νομισμάτων, κάποια εκ των οποίων είναι τα νομίσματα Ethereum [2], Litecoin [3] και Dogecoin [4].

Ωστόσο το blockchain μπορεί να χρησιμοποιηθεί και σε άλλες εφαρμογές. Στο [5] παρουσιάζονται αναλυτικά περιπτώσεις εφαρμογών όπου χρησιμοποιείται το blockchain. Κάποια από αυτά τα παραδείγματα είναι:

- Η ολοκλήρωση και η καταγραφή συναλλαγών που αφορούν σε ιδιωτικές κινητές αξίες.
- Η καταγραφή και η διαχείριση μετα-εμπορικών εκδηλώσεων.
- Η καταγραφή, διαχείριση και η ανίχνευση πολύτιμων ιδιοκτησιών.

Τέλος στο [6] έχουν προταθεί περιπτώσεις χρήσεως όπως για παράδειγμα:

- Εφαρμογές για διαχείριση και καταγραφή ψηφοφοριών.
- Αποθήκευση πληροφοριών για ασθενείς σε νοσοκομεία.
- Αποθήκευση σταστιστικών στοιχείων για πληθυσμούς ανθρώπων.
- Εφαρμογές για ολοκλήρωση πληρωμών τοκομεριδίων.

1.3 Χαρακτηριστικά εφαρμογών blockchain

Σε μια εφαρμογή blockchain οι κόμβοι πρέπει σε κάθε χρονική στιγμή να συμφωνούν για την κατάσταση του blockchain που χρησιμοποιούν. Έτσι ορίζονται κάποια πρωτόκολλα επικοινωνίας ή αλγόριθμοι για να επιτευχθεί αυτή η συμφωνία μεταξύ των κόμβων, οι οποίοι διαφέρουν από εφαρμογή σε εφαρμογή. Στο Bitcoin για παράδειγμα χρησιμοποιείται ένας

αλγόριθμος σύμφωνα με τον οποίο ένας κόμβος πρέπει να λύσει ένα μαθηματικό πρόβλημα υψηλής δυσκολίας. Αν επιτύχει σε αυτό τότε του επιτρέπεται από τους άλλους κόμβους να προσθέσει ένα block στην blockchain. Με αυτόν τον τρόπο δεν μπορεί κάποιος κόμβος να προσθέσει αυθαίρετα εγγραφές και όλοι οι κόμβοι συμφωνούν ως προς τις καταγεγραμμένες εγγραφές στο blockchain.

Επιπρόσθετα, οι κόμβοι πρέπει να συμφωνούν ως προς την εγκυρότητα των περιεχομένων ενός block. Ο τρόπος επίτευξης αυτής της απαίτησης πάλι εξαρτάται από την εκάστοτε εφαρμογή. Στο bitcoin, οι κόμβοι χρησιμοποιούνε δημόσια κλειδιά, όπου κάθε κόμβος κατέχει ένα από αυτά και υπογράφει τα περιεχόμενα ενός block με αυτό, αν τα θεωρήσει αξιόπιστα.

1.4 Σκοπός της εργασίας

Η διπλωματική αυτή εργασία επικεντρώνεται στην ανάπτυξη και στην αξιολόγηση στρατηγικών για caching σε εφαρμογές που χρησιμοποιούν την δομή blockchain, σε ένα δίκτυο από ομότιμους χρήστες. Οι στρατηγικές αυτές έχουν ως στόχο ο κάθε χρήστης στο δίκτυο, που εκπροσωπείται από έναν κόμβο, να έχει την δυνατότητα να αποθηκεύσει στην τοπική μνήμη ένα ποσοστό από blocks του blockchain, με βάση τα προσωπικά του κριτήρια. Κάθε κόμβος έχει κάποια ενδιαφέροντα με βάση τα οποία καθορίζεται η αξία ενός block για τον κόμβο αυτόν. Σκοπός είναι να βρεθούν οι αποτελεσματικότερες στρατηγικές για caching, ώστε κάθε κόμβος να αποθηκεύει στην τοπική του μνήμη τα blocks με την μεγαλύτερη για αυτόν αξία.

Ταυτόχρονα περιγράφεται ένα υλοποιημένο πρωτόκολλο επικοινωνίας μεταξύ των κόμβων που ανήκουν σε αυτό το δίκτυο, το οποίο επιτρέπει την ανταλλαγή πληροφοριών και την μεταξύ τους συνεργασία.

1.5 Αναγκαιότητα εκτέλεσης του έργου

Η δομή blockchain έχει το χαρακτηριστικό πως δεν μπορεί να μεταβληθεί παρά μόνο να προστεθούν καινούρια blocks στο τέλος της αλυσίδας. Παράλληλα, όλα τα blocks ανεξάρτητα από την θέση τους στην αλυσίδα, έχουν την ίδια σημασία για την εφαρμογή που τη χρησιμοποιεί.

Με βάση τα παραπάνω γίνεται κατανοητό πως το μέγεθος της blockchain αυξάνεται με το χρόνο. Αυτό έχει ως αποτέλεσμα οι κόμβοι που τρέχουν μια τέτοια εφαρμογή, να χρειάζονται όλο και περισσότερο ελεύθερο χώρο στην τοπική τους μνήμη για την αποθήκευση της.

Σε σύγχρονες εφαρμογές blockchain, συχνά οι κόμβοι αποθηκεύουν ολόκληρη την δομή. Αυτό γίνεται συνήθως για λόγους ασφαλείας, ώστε να προστατεύεται η ακεραιότητα της δομής, αφού με αυτό τον τρόπο υπάρχουν περισσότερα αντίγραφα στο δίκτυο.

Στην παρούσα διπλωματική εργασία προτείνεται η προσέγγιση της αποθήκευσης μόνο ορισμένων block με βάση τα ενδιαφέροντα του κάθε χρήστη. Αυτή η προσέγγιση έχει το πλεονέκτημα πως ο κάθε χρήστης μπορεί να κρατάει τα σημαντικά για αυτόν blocks στην τοπική του μνήμη, με βάση κριτήρια τα οποία θέτει ο ίδιος. Ταυτόχρονα προτείνεται ένα πρωτόκολλο επικοινωνίας μεταξύ των κόμβων, με σκοπό ο καθένας από αυτούς να έχει την δυνατότητα να αποκτήσει οποιαδήποτε στιγμή επιθυμεί, τα άλλα blocks του blockchain.

Για τον λόγο αυτό υλοποιήθηκαν και αξιολογήθηκαν πειραματικά διάφοροι αλγόριθμοι για την διαχείριση της διαθέσιμης μνήμης κάθε κόμβου και την αξιολόγηση κάθε block που

εμφανίζεται σε μια εφαρμογή blockchain, ώστε ένας κόμβος να μπορεί να αποφασίσει αν επιθυμεί να το αποθηκεύσει στην μνήμη του.

1.6 Περιορισμοί εργασίας

Στην παρούσα εγρασία οι αλγόριθμοι που σχεδιάστηκαν και υλοποιήθηκαν, αξιολογούνται σε μια εφαρμογή blockchain που τρέχει σε ένα δίκτυο κόμβων. Ωστόσο παραλείπονται κάποια χαρακτηριστικά των πραγματικών εφαρμογών blockchain.

Αρχικά, γίνεται η υπόθεση πως οι κόμβοι συμφωνούν με τετριμμένο τρόπο για τα περιεχόμενα των εγγραφών που δημιουργούνται και προστίθενται στην αλυσίδα. Οπότε δεν υπάρχει κάποια λογική επαλήθευσης ή υπογραφής αυτών από τους κόμβους στην εφαρμογή, όπως γίνεται σε άλλες παρόμοιες εφαρμογές.

Ταυτόχρονα υποθέτουμε πως οι κόμβοι συμφωνούν με αυθαίρετο τρόπο για την κατάσταση του blockchain σε κάθε χρονική στιγμή, δίχως να χρειάζεται να συγχρονιστούν μεταξύ τους.

Οι παραπάνω περιορισμοί επιτρέπουν την επικέντρωση στους αλγορίθμους διαχείρισης τοπικής μνήμης και αξιολόγησης των block. Ο τρόπος λειτουργίας τους δεν επηρεάζεται από τους παράγοντες που αναφέρθηκαν παραπάνω, οπότε μπορούν να αφαιρεθούν δίχως να αλλοιώνονται τα αποτελέσματα των αλγορίθμων.

1.7 Δομή εργασίας

Στο κεφάλαιο 2 περιγράφονται η δομή των block της υλοποιημένης εφαρμογής και η δομή των συναλλαγών.

Στο κεφάλαιο 3 αναλύονται και επεξηγούνται οι αλγόριθμοι αξιολόγησης block καθώς και οι αλγόριθμοι διαχείρισης μνήμης.

Στο κεφάλαιο 4 περιγράφεται η δομή του συστήματος. Αυτή περιλαμβάνει τους κόμβους τους συστήματος καθώς και το πρωτόκολλο επικοινωνίας που χρησιμοποιείται.

Στο κεφάλαιο 5 αναφέρεται η μεθοδολογία της πειραματικής διαδικασία και παρουσιάζονται και αξιολογούνται τα αποτελέσματα της.

Στο κεφάλαιο 6 αναφέρονται τα συμπεράσματα που προέκυψαν από την διπλωματική εργασία και αναφέρονται προτάσεις για μελλοντική εργασία και βελτιώσεις.

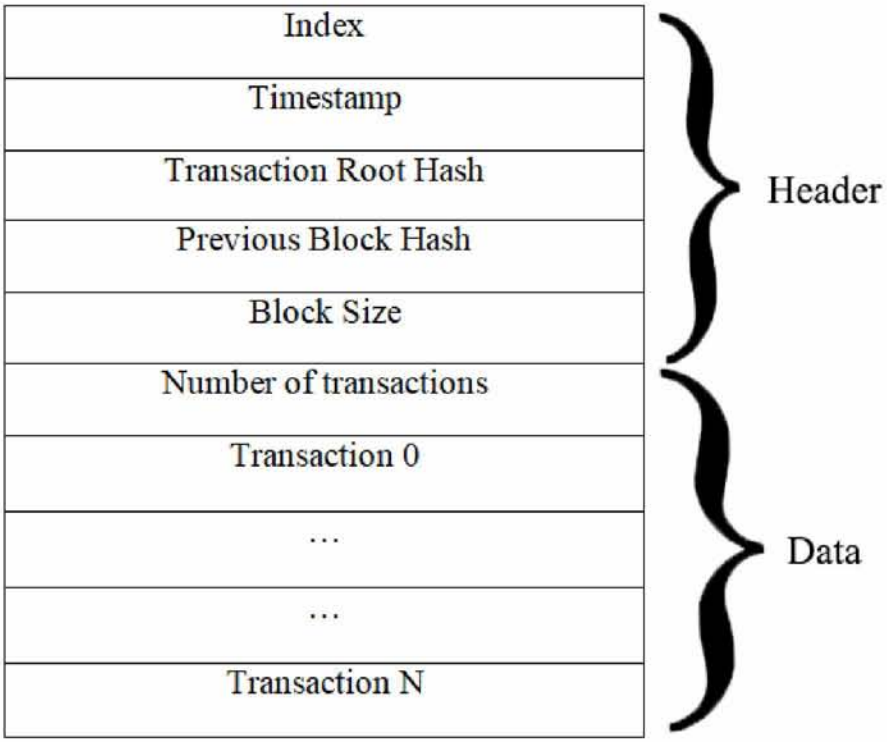
1.8 Κώδικας Εργασίας

Ο κώδικας που υλοποιήθηκε στα πλαίσιας της διπλωματικής εργασίας βρίσκεται στον παρακάτω σύνδεσμο https://github.com/Shalantor/Blockchain-Caching

2. ΠΕΡΙΓΡΑΦΗ ΔΟΜΩΝ BLOCKCHAIN

2.1 Ανάλυση δομής block εφαρμογής

Η δομή ενός block στην υλοποιημένη εφαρμογή φαίνεται στην Εικόνα 1:



Εικόνα 1. Δομή ενός block

Πιο αναλυτικά τα πεδία που αποτελούν την επικεφαλίδα του block είναι:

- **Index**: Αυτό το πεδίο αντιστοιχεί στην θέση του block στο blockchain. Η αρίθμηση ξεκινάει από την αρχή της αλυσίδας. Το block στην αρχή της αλυσίδας έχει index ίσο με 0, το δεύτερο 1 κ.ο.κ. Κάθε index είναι μοναδικό και με αυτό προσδιορίζονται τα blocks.
- **Timestamp**: Αντιστοιχεί στην χρονική στιγμή που δημιουργήθηκε το block.

- **Transaction Root Hash**: Αποτελεί την ρίζα δέντρου Merkle, όπως αυτό περιγράφεται στο [7]. Οι συναλλαγές (transactions) αποτελούν τα φύλλα του δέντρου Merkle και η διαδικασία δημιουργίας της ρίζας είναι η ίδια με την κρυπτογράφηση που πραγματοποιείται στην εφαρμογή Bitcoin, που περιγράφεται στο [1]. Κάθε κόμβος στο δέντρο συνενώνει τα αποτέλεσμα κατακερματισμού των δύο παιδιών του και τα κατακερματίζει ακόμη μία φορά με την ίδια συνάρτηση κατακερματισμού. Έπειτα αυτά αποθηκεύονται στον κόμβο. Η διαδικασία αυτή συνεχίζεται μέχρι να μείνει μόνο ένα αποτέλεσμα κατακερματισμού, που αποτελεί την ρίζα του δέντρου, η οποία τελικά αποθηκεύεται στην επικεφαλίδα του block. Η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η SHA-256 που αναλύεται στο [8].
- **Previous Block Hash**: Αντιστοιχεί στον κατακερματισμό της επικεφαλίδας του προηγούμενο block. Αξίζει να σημειωθεί πως στον κατακερματισμό αυτόν δεν χρειάζεται να συμπεριλαμβάνουμε τα δεδομένα του προηγούμενου block, καθώς ανήκει και η ρίζα του δέντρου Merkle στην επικεφαλίδα.
- **Block Size**: Είναι το συνολικό μέγεθος του block σε bytes. Στον υπολογισμό συμπεριλαμβάνονται τόσο η επικεφαλίδα όσο και τα δεδομένα.

Τα πεδία που αποτελούν τα δεδομένα του block είναι τα παρακάτω:

- **Number of transactions**: Ο αριθμός των συναλλαγών που περιέχονται στο block.
- **Transactions**: Είναι ένας πίνακας που περιέχει τις συναλλαγές που βρίσκονται στο block. Το μέγεθος του πίνακα ισούται με τον αριθμό που αναγράφεται στο πεδίο που αφορά στον αριθμό των transactions.

2.2 Ανάλυση δομής συναλλαγής

Μια συναλλαγή αποτελείται από ένα σύνολο λέξεων-κλειδιών που αντιστοιχίζονται σε τιμές. Οι τιμές αυτές μπορεί να είναι είτε αριθμοί είτε μια αλληλουχία αλφαριθμητικών χαρακτήρων. Τα κλειδιά ωστόσο έχουν πάντα την μορφή λέξεων, οι οποίες όμως μπορούν να περιέχουν και αριθμούς. Επίσης δεν μπορούν να υπάρχουν δύο ή παραπάνω κλειδιά με το ίδιο όνομα. Μια ενδεικτική μορφή συναλλαγής παρουσιάζεται παρακάτω στην Εικόνα 2:

Key	Value
sender	node100
receiver	node2
price	320.16
category	equipment
amount	15
fee	15.0
origin	greece

Εικόνα 2. Ενδεικτική μορφή συναλλαγής.

3. ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

3.1 Ανάλυση αλγορίθμων αξιολόγησης block

Οι κόμβοι στην εφαρμογή ενδιαφέρονται για συγκεκριμένες τιμές από ένα ή περισσότερα κλειδιά σε μια συναλλαγή. Για παράδειγμα ένας κόμβος μπορεί να ενδιαφέρεται για μια αριθμητική τιμή όταν αυτή η ανήκει ανήκει σε ένα συγκεκριμένο διάστημα τιμών. Επίσης ένας κόμβος μπορεί να ενδιαφέρεται για μια τιμή που είναι αλληλουχία αλφαριθμητικών χαρακτήρων, όταν αυτή ισούται με μια συγκεκριμένη λέξη. Αν βρεθεί μια συναλλαγή για την οποία ισχύει ένα από τα παραπάνω, ο κόμβος ενδιαφέρεται για την συναλλαγή αυτή.

Έτσι αν οι συναλλαγές έχουν για παράδειγμα ένα κλειδί με όνομα X και ο κόμβος ενδιαφέρεται για την τιμή Y που αντιστοιχεί στο X, τότε ο κόμβος ενδιαφέρεται για όλες τις συναλλαγές που έχουν την τιμή Y για το κλειδί X.

Με βάση τα παραπάνω, ακολουθάει η ανάλυση των αλγορίθμων αξιολόγησης των block από τους κόμβους.

3.1.1 Αλγόριθμος Interest Based (IB)

Στον IB αλγόριθμο ένας κόμβος ελέγχει τις τιμές για όλα τα κλειδιά που τον ενδιαφέρουν. Αυτή η διαδικασία πραγματοποιείται ξεχωριστά για κάθε συναλλαγή που βρίσκεται στο υπό έλεγχο block. Μόλις βρεθεί τουλάχιστον μία συναλλαγή για την οποία ενδιαφέρεται ο κόμβος, το block που την περιέχει επιχειρείται να αποθηκευθεί στην τοπική μνήμη του κόμβου.

3.1.2 Αλγόριθμος Threshold Based (TB)

Στον αλγόριθμο TB ο κόμβος πάλι ελέγχει τις τιμές για τα κλειδιά που τον ενδιαφέρουν. Ωστόσο θέτει ένα κάτω όριο συναλλαγών οι οποίες πρέπει να είναι ενδιαφέρουσες για τον κόμβο, ώστε να αποδεχθεί να αποθηκεύσει το block που τις περιέχει. Αν δεν υπάρχουν τουλάχιστον τόσες συναλλαγές όσες ορίζονται από αυτό το όριο, τότε ο κόμβος αγνοεί το block.

3.1.3 Αλγόριθμος Threshold Weight Based (TWB)

Στον αλγόριθμο TWB ο κόμβος πάλι θέτει ένα κάτω όριο για τις ενδιαφέρουσες συναλλαγές. Η σημαντική διαφορά όμως σε σχέση με τον TB ειναι ότι ο TWB θέτει βάρη για τα ενδιαφέροντα του, με αποτέλεσμα κάποια από αυτά να θεωρούνται πιο σηματικά από άλλα. Ο κόμβος ελέγχει όλες τις συναλλαγές και κρατάει ένα σκορ. Αν βρεθεί μια ενδιαφέρουσα συναλλαγή, προστίθεται στο σκορ το βάρος που έχει τεθεί για την τιμή που βρέθηκε. Αφού ελεγχθούν όλες οι συναλλαγές, ελέγχεται αν το σκορ είναι μεγαλύτερο ή ίσο από το κάτω όριο που έχει τεθεί. Στην περίπτωση αυτή, ο κόμβος επιχειρεί να αποθηκεύσει το block στην μνήμη του, διαφορετικά το αγνοεί. Ο αλγόριθμος TWB ισούται με τον TB όταν θέσουμε όλα τα βάρη να είναι ίσα με 1.

3.1.4 Ψευδοκώδικες αλγορίθμων

Παρακάτω παρουσιάζονται οι ψευδοκώδικες των αλγορίθμων που περιγράφηκαν παραπάνω:

```
1.  for each transaction in received block:
2.      for each key in transaction:
3.          if node is interested in value of key:
4.              exit and add block to cache
5.  ignore block
```

Εικόνα 3. Ο αλγόριθμος IB

```
1.      score = 0
2.      for each transaction in received block:
3.          for each key in transaction:
4.              if node is interested in value of key:
5.                  score = score + 1
6.      if score >= threshold :
7.          add block to cache
8.      else :
9.          ignore block
```

Εικόνα 4. Ο αλγόριθμος TB

```
1.  score = 0
2.  for each transaction in received block:
3.      for each key in transaction:
4.          if node is interested in value of key:
5.              score = score + weight of that interest
6.  if score >= threshold :
7.      add block to cache
8.  else :
9.      ignore block
```

Εικόνα 5. Ο αλγόριθμος TWB

3.2 Αλγόριθμοι διαχείρησης μνήμης

Μόλις ένα block θεωρηθεί σημαντικό για έναν κόμβο από τους αλγορίθμους αξιολόγησης, τότε επιχειρείται να αποθηκευθεί στην μνήμη. Για τον σκοπό αυτό σχεδιάστηκαν και υλοποιήθηκαν πολιτικές αντικατάστασης block στην μνήμη που θα παρουσιαστούν στις παρακάτω υποενότητες.

3.2.1 Recency Based αντικατάσταση

Στην Recency Based αντικατάσταση ο κόμβος απομακρύνει από την τοπική μνήμη το block με το μικρότερο index, σε περίπτωση που λάβει ένα καινούριο block και δεν επαρκεί ο διαθέσιμος κενός χώρος στην μνήμη για την αποθήκευση του. Το block με το μικρότερο index είναι το παλαιότερο block στην μνήμη. Αυτή η πολιτική αντικατάστασης θεωρεί πιο σημαντικές τις συναλλαγές που περιέχονται στα πιο πρόσφατα block που βρίσκονται στην μνήμη.

3.2.2 Block size based αντικατάσταση

Στην Block size based αντικατάσταση, απομακρύνεται από την τοπική μνήμη το block με το μεγαλύτερο μέγεθος σε bytes. Σε αυτή την πολιτική αντικατάστασης, ο κόμβος προσπαθεί να κρατήσει στην μνήμη του όσο το δυνατόν περισσότερα blocks που τον ενδιαφέρουν. Έτσι όταν απομακρύνεται το μεγαλύτερο block από την μνήμη, δημιουργείται χώρος για περισσότερα μικρότερα ενδιαφέροντα blocks.

3.2.3 Score based αντικατάσταση

Στην score based αντικατάσταση, απομακρύνεται από την τοπική μνήμη το block με το μικρότερο score. Το score υπολογίζεται με τον ίδιο τρόπο όπως στους αλγορίθμους αξιολόγησης block TB και TWB που παρουσιάστηκαν στην προηγούμενη ενότητα. Ο κόμβος μπορεί να επιλέξει αν θέλει να αντιστοιχίσει κάποια βάρη σε κάθε ενδιαφέρον του για τον υπολογισμό του

score ή αν θέλει απλά να μετράει το πλήθος των ενδιαφέροντων συναλλαγών. Με αυτό τον τρόπο ο κόμβος προσπαθεί να μεγιστοποιήσει την χρήσιμη για αυτόν πληροφορία που βρίσκεται στην τοπική του μνήμη, καθώς αποθηκεύει τα blocks με τις περισσότερες για αυτόν ενδιαφέρουσες συναλλαγές.

3.2.4 Cost based αντικατάσταση

Για την cost based αντικατάσταση χρησιμοποιήθηκε μια παραλλαγή του αλγορίθμου που περιγράφεται στο [9]. Σε αυτή την πολιτική αντικατάστασης ανατίθεται σε κάθε block ένα score την χρονική στιγμή *k* με βάση των παρακάτω τύπο:

$$S_i * \Delta T_k$$

Όπου S_i είναι το μέγεθος του block i , ενώ ΔT_k είναι ο αριθμός των προσβάσεων στη μνήμη που έχουν γίνει μετά από την τελευταία πρόσβαση στο block i την χρονική στιγμή k .

Ταυτόχρονα για να συνυπολογιστεί το κόστος μεταφοράς ενός block στο δίκτυο αξιοποιούμε δύο παράγοντες. Ο πρώτος είναι το μέγεθος του, καθώς ένα μεγάλο block χρειάζεται περισσότερους πόρους δικτύου και χρόνο για να μεταφερθεί από έναν κόμβο στον άλλον σε ένα δίκτυο. Ο δεύτερος παράγοντας αφορά στο θόρυβο που μπορεί να υπάρχει στο δίκτυο. Έτσι ο τελικός τύπος για το score ενός block i όταν επιθυμούμε να συμπεριληφθεί το κόστος είναι ο :

$$S_i * \Delta T_k (q * S_i + 2 * (1 - q) * A * \text{random}(0,1))$$

Όπου A είναι ο μέσος όρος των μεγεθών των block που βρίσκονται στην μνήμη. Το q είναι ένας παράγοντας που παίρνει τιμές από 0 έως 1. Αν το q έχει υψηλές τιμές τότε θεωρείται πιο σημαντικός παράγοντας για το κόστος του block το μέγεθος του. Το υπόλοιπο κόστος προκύπτει από τον θόρυβο στο δίκτυο και πολλαπλασιάζεται με τον παράγοντα 1 – q. Με αυτό τον τρόπο μπορούμε να ελέγξουμε την σημασία που δίνουμε στον θόρυβο του δικτύου και στο μέγεθος του block στον υπολογισμό του κόστους. Τέλος, επειδή δεν έχουμε πάντα θόρυβο στο δίκτυο μεταξύ δύο κόμβων, πολλαπλασιάζουμε τον παράγοντα του θορύβου με μια τυχαία μεταβλητή που παίρνει την τιμή 0 ή 1.

Έτσι όταν χρειάζεται να απομακρυνθεί ένα block από την cache, επιλέγεται το block με το χαμηλότερο score.

4. ΠΕΡΙΓΡΑΦΗ ΔΟΜΗΣ ΣΥΣΤΗΜΑΤΟΣ

4.1 Περιγραφή κόμβων δικτύου

Στο δίκτυο που υλοποιήθηκε υπάρχουν τέσσερα διαφορετικά είδη κόμβων. Κάθε είδος έχει έναν ξεχωριστό ρόλο στο δίκτυο. Όλοι οι κόμβοι ρυθμίζονται με τέτοιο τρόπο ώστε να γνωρίζουν τις τοπικές διευθύνσεις των άλλων κόμβων, δηλαδή την θύρα στην οποία αυτοί δέχονται αιτήματα. Στις παρακάτω υποενότητες αναλύονται οι λειτουργίες και τα χαρακτηριστικά καθενός από αυτά.

4.1.1 Full Node

Στο δίκτυο υπάρχει ένα κόμβος full node. Αυτός κρατάει όλο το blockchain που δημιουργείται στην εφαρμογή. Όταν δέχεται ένα block μέσω του δικτύου, το προσθέτει απλά στην αλυσίδα, δίχως να πραγματοποιεί κάποιον έλεγχο.

Ο full node μπορεί να απαντήσει σε αιτήματα από άλλους κόμβους που ζητάνε blocks με συγκεκριμένες τιμές στο πεδίο index της επικεφαλίδας του block. Ωστόσο μπορεί να απαντήσει και σε αιτήματα που αφοράνε οποιοδήποτε άλλο πεδίο ενός block, ακόμα και σε αιτήματα για συναλλαγές με συγκεκριμένες τιμές στις ετικέτες κάποιων κλειδιών. Αυτό επιτυγχάνεται με μια βάση δεδομένων στην οποία ο full node αποθηκεύει τα blocks που δέχεται. Έτσι μπορούν να αναζητηθούν συγκεκριμένες τιμές σε οποιοδήποτε πεδίο ζητάει κάποιος άλλος κόμβος

4.1.2 Miner node

Στο δίκτυο υπάρχει επίσης ένας miner node. Ο κόμβος αυτός είναι υπεύθυνος για την δημιουργία καινούριων block. Δέχεται συναλλαγές και τις αποθηκεύει προσωρινά στην μνήμη του. Μόλις αυτές είναι αρκετές για την δημιουργία ενός block, τις απομακρύνει από τη μνήμη και τις τοποθετεί στο καινούριο block. Έπειτα διοχετεύει το block στο δίκτυο, ώστε να μπορούν να το λάβουν και οι υπόλοιποι κόμβοι του δικτύου.

Ο miner node ρυθμίζεται με τρόπο ώστε να γνωρίζει το μικρότερο και το μεγαλύτερο επιτρεπτό μέγεθος για κάθε καινούριο block που δημιουργείται.

Επίσης υπάρχουν δύο διαθέσιμες ρυθμίσεις για τον τρόπο επιλογής των συναλλαγών που θα συμπεριληφθούν στο επόμενο block που θα δημιουργηθεί. Στην πρώτη, ο κόμβος δεν αλλάζει την σειρά των συναλλαγών που δέχεται. Μόλις το μέγεθος των συναλλαγών που έχει αποθηκεύσει ξεπεράσει το χαμηλότερο επιτρεπτό μέγεθος ενός block, δημιουργεί ένα καινούριο block και τις τοποθετεί σε αυτό.

Στην άλλη ρύθμιση, που ονομάζουμε ρύθμιση αλγόριθμου group, ο miner node επιλέγει ποιες από τις αποθηκευμένες συναλλαγές θα προσθέσει στο επόμενο block που θα δημιουργήσει. Ο τρόπος επιλογής των συναλλαγών είναι μια παραλλαγή του προβλήματος knapsack. Το κριτήριο επιλογής είναι η συχνότητα εμφάνισης ίδιων τιμών των κλειδιών. Τιμές που εμφανίζονται πιο συχνά από τις υπόλοιπες συμπεριλαμβάνονται στις επιλεγμένες συναλλαγές. Για να επιτευχθεί αυτό ο miner node αρχικά βρίσκει για κάθε κλειδί την τιμή που εμφανίζεται τις περισσότερες φορές στις αποθηκευμένες συναλλαγές. Μη αριθμητικές τιμές, συγκρίνονται μεταξύ τους απλά ως προς την συχνότητα εμφάνισης. Οσον αφορά στις αριθμητικές τιμές, ο miner node διαχωρίζει το εύρος των τιμών που έλαβε σε ίσα διαστήματα, ο αριθμός των οποίων μπορεί να προσδιοριστεί προγραμματιστικά ρυθμίζοντας το miner node

κατάλληλα κατά την εκκίνηση της εφαρμογής. Έπειτα συγκρίνονται οι εμφανίσεις σε κάθε διάστημα τιμών μεταξύ τους. Μόλις γίνει αυτό ο miner node συγκρίνει τις τιμες με τις περισσότερες εμφανίσεις για κάθε κλειδί μεταξύ τους. Σε αυτό το βήμα συγκρίνονται απλά οι συχνότητες εμφανίσεως μεταξύ τους πολλαπλασιασμένες με έναν παράγοντα. Ο παράγοντας αυτός υπολογίζεται ως το αποτέλεσμα της διαίρεσης του αριθμού των πιθανών τιμών που μπορεί να έχει η τιμή ενός κλειδιού δια των αριθμό τιμών που μπορεί να πάρει η άλλη τιμή που συγκρίνεται με αυτήν. Ο λόγος για τον οποίο συμπεριλαμβάνουμε τον παράγοντα αυτό στους υπολογισμούς είναι ότι αν μια τιμή ενός κλειδιού μπορεί να πάρει λίγες διαφορετικές τιμές τότε θα επιλέγεται πάντα αυτή ως η συχνότερη. Αν όμως ληφθούν υπόψην και όλες οι πιθανές τιμές, τότε δεν υπάρχει κάποια προτίμηση μεταξύ των τιμών, διότι με αυτό τον τρόπο είναι σαν να έχουν όλες οι τιμές κλειδιών τον ίδιο αριθμό πιθανών τιμών. Η παραπάνω διαδικασία συνεχίζεται μέχρι να βρεθούν αρκετές συναλλαγές για να δημιουργηθεί ένα block.

Ωστόσο στην παραπάνω διαδικασία υπάρχει η περίπτωση όταν προστεθούν οι συναλλαγές, το μέγεθος να ξεπερνάει το μεγαλύτερο επιτρεπτό μέγεθος ενός block. Τότε ο miner node επαναλαμβάνει την παραπάνω διαδικασία, δίχως όμως να συμπεριλαμβάνεται αυτή τη φορά η τιμή κλειδιού που οδήγησε στην περίπτωση αυτή. Έτσι στην χειρότερη περίπτωση δοκιμάζονται όλοι οι πιθανοί συνδυασμοί από συναλλαγές μέχρι να εξαντληθούν.

Σε αυτή τη ρύθμιση ο miner προσπαθεί να δημιουργήσει τα blocks με τέτοιο τρόπο ώστε να έχουν μέγεθος περίπου ίσο με το μέσο των τιμών για το μικρότερο και το μεγαλύτερο επιτρεπτό μέγεθος ενός block. Ο αλγόριθμος δεν ενεργοποιείται μέχρι το μέγεθος των αποθηκευμένων συναλλαγών να είναι **τουλάχιστον διπλάσιο από το όριο αυτό**. Αν δεν είναι εφικτό να δημιουργηθεί κάποιο block με αυτό το μέγεθος τότε θέτει σαν όριο το μικρότερο επιτρεπτό μέγεθος ενός block. Αυτό συμβαίνει όταν κανένας πιθανός συνδυασμός των

συχνότερων ετικετών είναι αρκετός για να έχει μέγεθος μεγαλύτερου του μέσου των μεγεθών για το block δίχως να είναι μικρότερο από το μεγαλύτερο επιτρεπτό μέγεθος ενός block.

Τέλος, ο miner ρυθμίζεται με ένα άνω όριο χρόνου για τον οποίο μπορεί να κρατήσει μια συναλλαγή δίχως να την αποθηκεύσει σε κάποιο block. Όταν μια συναλλαγή ξεπεράσει αυτό το όριο τότε ο miner υπό κάθε περίπτωση θα την τοποθετήσει στο επόμενο block. Αυτό σημαίνει πως στην ρύθμιση group η συγκεκριμένη συναλλαγή τοποθετείται στο επόμενο block πριν την εφαρμογή του αλγορίθμου block στις υπόλοιπες αποθηκευμένες συναλλαγές.

Στην Εικόνα 6 φαίνεται ο ψευδοκώδικας για την διαδικασία group:

```
1.  target_size = min_block_size + (max_block_size – min_block_size)/2
2.  highest_freq = []       // Empty array
3.  for each possible key:
4.      exclude values that the miner already tried to add
5.      Find the corresponding value with the highest frequency
6.      Add to highest_freq
7.  for i < length of highest_freq:
8.      mult = (num of values for key of highest_freq[i]) / (num of values for max)
9.      if mult * max < freq[i]:
10.         max = freq[i]
11. if size of transactions from max + current_size > max_block_size:
12.     goto 3
13. Add transactions from max to transactions of new block
14. current_size += size of transactions from max
15. if current_size > target_size:
16.     generate new block and exit
17. if already tried all combinations:
```

18.	*target_size = min_block_size*
19.	*clear excluded values*
20.	*goto 3*

Εικόνα 6. Ο αλγόριθμος group

4.1.3 Normal node

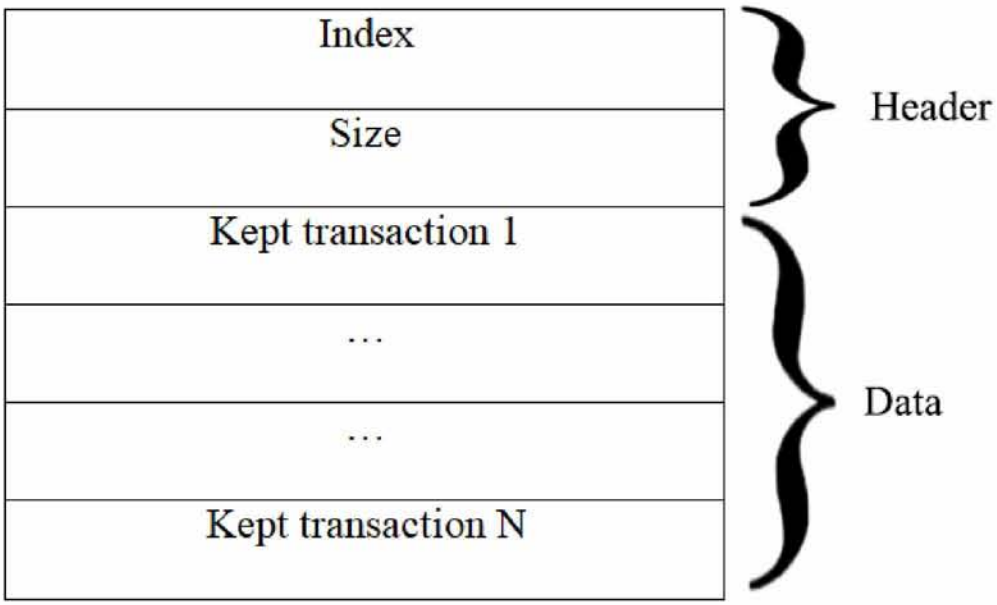
Στο δίκτυο υπάρχουν από 1 έως N normal nodes. Είναι ο πιο βασικός κόμβος του δικτύου. Στους κόμβους που ανήκουν σε αυτή τη κατηγορία τρέχουν οι αλγόριθμοι αξιολόγησης των block και η αλγόριθμοι διαχείρησης μνήμης.

Οι ρυθμίσεις για αυτόν τον κόμβο συμπεριλαμβάνουν το μέγεθος της τοπικής του μνήμης και τους αλγορίθμους αξιολόγησης και διαχείρισης μνήμης που θα χρησιμοποιηθούν κατά την λειτουργία του.

4.1.4 Light Node

Στο δίκτυο υπάρχουν από 1 έως N light nodes. Αυτοί οι κόμβοι εκτελούν ακριβώς τις ίδιες λειτουργίες με τους normal nodes. Χρησιμοποιούν τους ίδιους αλγορίθμους αξιολόγησης και διαχείρισης μνήμης και δέχονται τις ίδιες ρυθμίσεις.

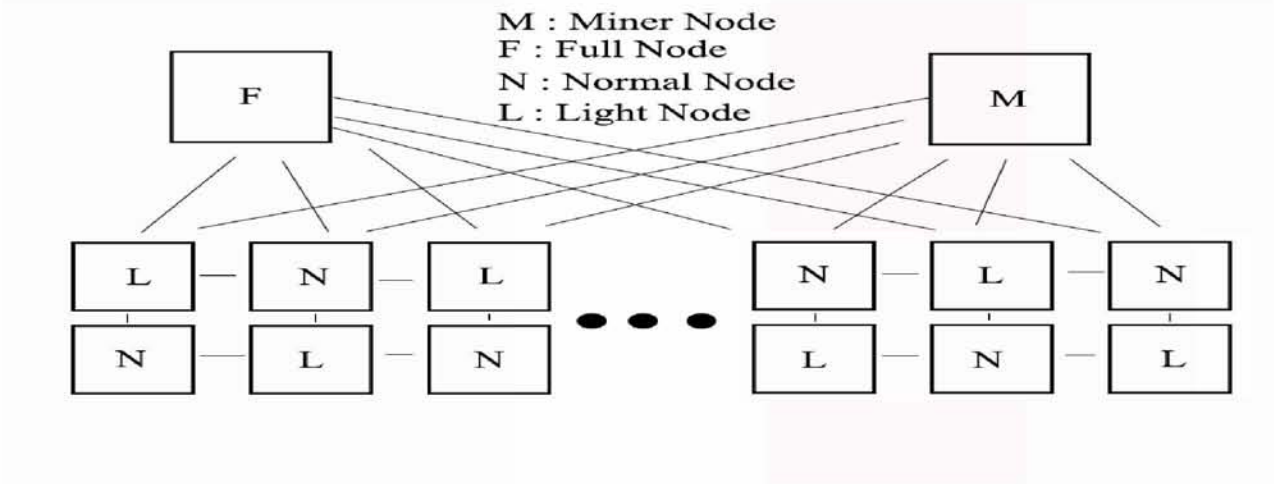
Ωστόσο η σημαντική διαφορά είναι ότι αυτοί οι κόμβοι δεν αποθηκεύουν ολόκληρα τα blocks στην μνήμη τους παρά μόνο τις σημαντικές για αυτούς συναλλαγές και κάποια από τα πεδία επικεφαλίδας των block. Η δομή που αποθηκεύεται στους κόμβους φαίνεται στο παρακάτω σχήμα:



Εικόνα 7. Δομή αποθηκευμένου block σε light nodes

4.1.5 Σχεδιάγραμμα κόμβων δικτύου

Παρακάτω στην Εικόνα 8 φαίνεται το σχεδιάγραμμα των κόμβων που συμμετέχουν στο δίκτυο:



Εικόνα 8. Δομή δικτύου εφαρμογής

4.2 Ανάλυση πρωτοκόλλου επικοινωνίας

Στα πλαίσια της εργασίας αναπτύχθηκε ένα πρωτόκολλο επικοινωνίας που επιτρέπει την ανταλλαγή πληροφοριών μεταξύ των κόμβων.

4.2.1 Αιτήματα προς το full node

Το full node μπορεί να απαντήσει σε μηνύματα από οποιουσδήποτε άλλους κόμβους, τα οποία ζητάνε blocks με συγκεκριμένα index, δηλαδή τις θέσεις τους στο blockchain. Για το σκοπό αυτό κάποιος κόμβος μπορεί να στείλει μια αίτηση που περιέχει έναν πίνακα με τις θέσεις των blocks που επιθυμεί. Επίσης μπορεί να στείλει διαστήματα τιμών για τις θέσεις που θέλει και το full node θα απαντήσει με τα blocks που βρίσκονται στα διαστήματα αυτά. Τέλος, το full node μπορεί να απαντήσει σε αιτήματα για συγκεκριμένες τιμές σε οποιοδήποτε πεδίο του, τόσο στην επικεφαλίδα όσο και στα δεδομένα του. Αυτό συμπεριλαμβάνει και τιμές για κάποια κλειδιά των συναλλαγών. Αυτή η δυνατότητα είναι διαθέσιμη επειδή το blockchain στο full node αποθηκεύεται σε μια βάση δεδομένων, η οποία μπορεί να ερωτηθεί για οποιαδήποτε τιμή σε πεδίο του block.

4.2.3 Αιτήματα προς normal και light nodes

Τα nodes αυτά μπορούνε να ανταλλάσουν αιτήσεις και τις αντίστοιχες απαντήσεις μεταξύ τους. Οι αιτήσεις μπορούν να αφορούν τα ενδιαφέροντα ή τα αποθηκευμένα blocks σε

έναν κόμβο. Όταν ένας από αυτούς τους κόμβους επιθυμεί να ενημερωθεί για τα ενδιαφέροντα ενός άλλου κόμβου, του στέλνει μια αίτηση και λαμβάνει ως απάντηση τα κλειδιά συναλλαγών και τις αντίστοιχες τιμές των ετικετών για τα οποία ενδιαφέρεται ο κόμβος αυτός. Οι αιτήσεις αυτές στέλνονται σε περισσότερους από έναν κόμβους, καθώς διοχετεύονται στο δίκτυο. Επίσης ένας κόμβος μπορεί να ζητήσει τα blocks ενός άλλου κόμβου ή ένα μέρος αυτών. Επειδή τα light nodes δεν αποθηκεύουν ολόκληρα τα blocks, απαντάνε με τις θέσεις των blocks που διαθέτουν. Έπειτα ο κόμβος που έλαβε αυτές τις θέσεις, μπορεί να ζητήσει τα blocks από το full node.

Οι αιτήσεις για τα ενδιαφέροντα στέλνονται όταν ένας κόμβος εισέρχεται στο δίκτυο για πρώτη φορά ή όταν αλλάζει ενδιαφέροντα. Με αυτόν τον τρόπο μαθαίνει ποιοι άλλοι κόμβοι έχουν τα ίδια ενδιαφέροντα με αυτόν και μπορεί να ζητήσει τα blocks τους.

4.2.3 Μηνύματα καινούριων block

Όλοι οι κόμβοι στο δίκτυο μπορούν να λάβουν και να αναμεταδώσουν μηνύματα καινούριων block που διοχετεύονται από τον miner node στο δίκτυο, ώστε αυτό τελικά να φτάσει σε όλους τους κόμβους του δικτύου.

5 ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

5.1 Υλοποίηση συστύματος

Η εφαρμογή blockchain για τον έλεγχο των αλγορίθμων και την καταγραφή των αποτελεσμάτων υλοποιήθηκε στην γλώσσα προγραμματισμού java, έκδοση 1.8.

Για τα μηνύματα που ανταλλάσονται στο δίκτυο χρησιμοποιήθηκε η δομή αρχείων JSON, χρησιμοποιώντας την βιβλιοθήκη json.org για την γλώσσα java.

Στους κόμβους τύπου full node χρησιμοποιήθηκε η βάση δεδομένων MongoDB, μέσω του MongoDB Java Driver. Η επιλογή αυτής της βάσης δεδομένων διευκόλυνε την μετατροπή των μηνυμάτων σε εγγραφές της βάσης δεδομένων, διότι έχουν ακριβώς την ίδια μορφή με αποτέλεσμα να μην χρειάζεται κάποια ενδιάμεση μετατροπή.

5.2 Πειραματική διαδικασία

Για την καταγραφή των αποτελεσμάτων χρησιμοποιήθηκε μια τοπική ρύθμιση, όπου οι κόμβοι δεν επικοινωνούν μεταξύ τους. Ωστόσο υπάρχει ένα πρόγραμμα-βοηθός το οποίο προσφέρει όλες τις απαραίτητες πληροφορίες σε όλους τους κόμβους. Η παραπάνω ρύθμιση επιλέχθηκε κυρίως για λόγους ταχύτητας εκτέλεσης των πειραμάτων.

Σε κάθε πείραμα οι κόμβοι ρυθμίζονται με αρχεία text, μέσω των οποίων διαβάζονται οι τιμές για όλες τις παραμέτρους που χρειάζονται για να λειτουργήσουν, όπως είναι το μέγεθος της cache, οι αλγόριθμοι αξιολόγησης block και οι αλγόριθμοι διαχείρισης μνήμης που θα χρησιμοποιήσουν. Επίσης με αυτόν τον τρόπο ορίζονται τα μεγέθη των block που δημιουργεί ο κόμβος miner node, καθώς και ο αλγόριθμος δημιουργίας block που θα χρησιμοποιηθεί.

Επίσης οι τιμές των κλειδιών των συναλλαγών δημιουργούνται με βάση μια κατανομή Zipf, η οποία υλοποιήθηκε μέσω της βιβλιοθήκης Apache Commons. Η κατανομή Zipf χρησιμοποιείται συχνά για να παρουσιαστούν τα ενδιαφέροντα χρηστών που ανήκουν σε κάποιο σύνολο, ώστε να προσομοιωθεί με τον καλύτερο τρόπο ένα σύνολο χρηστών υπό πραγματικές συνθήκες. Όταν δημιουργείται μια συναλλαγή, αυτή δίνεται στον miner node, ο οποίος αποφασίζει αν έχει αρκετές συναλλαγές για την δημιουργία ενός νέου block.

Επιπλέον, τα αρχεία για τα ενδιαφέροντα των κόμβων έχουν δύο μορφές. Αν ενδιαφέρονται για μη αριθμητικές τιμές, τότε ένας κόμβος μπορεί να ενδιαφέρεται για οποιαδήποτε τιμή-λεξη ενός κλειδιού. Οι τιμές αυτής της κατηγορίας είναι λέξεις που επιλέγονται από ένα διαθέσιμο σύνολο τιμών. Αν ενδιαφέρονται ωστόσο για αριθμητικές τιμές, τότε ένας κόμβος μπορεί να ενδιαφέρεται για ένα διάστημα τιμών. Για τον λόγο αυτό τίθεται μια μέγιστη και μια ελάχιστη τιμή που μπορεί να πάρει μια αριθμητική τιμή.

Από τα πειράματα που πραγματοποιήθηκαν, επιλέχθηκαν πληροφορίες που αφορούν:

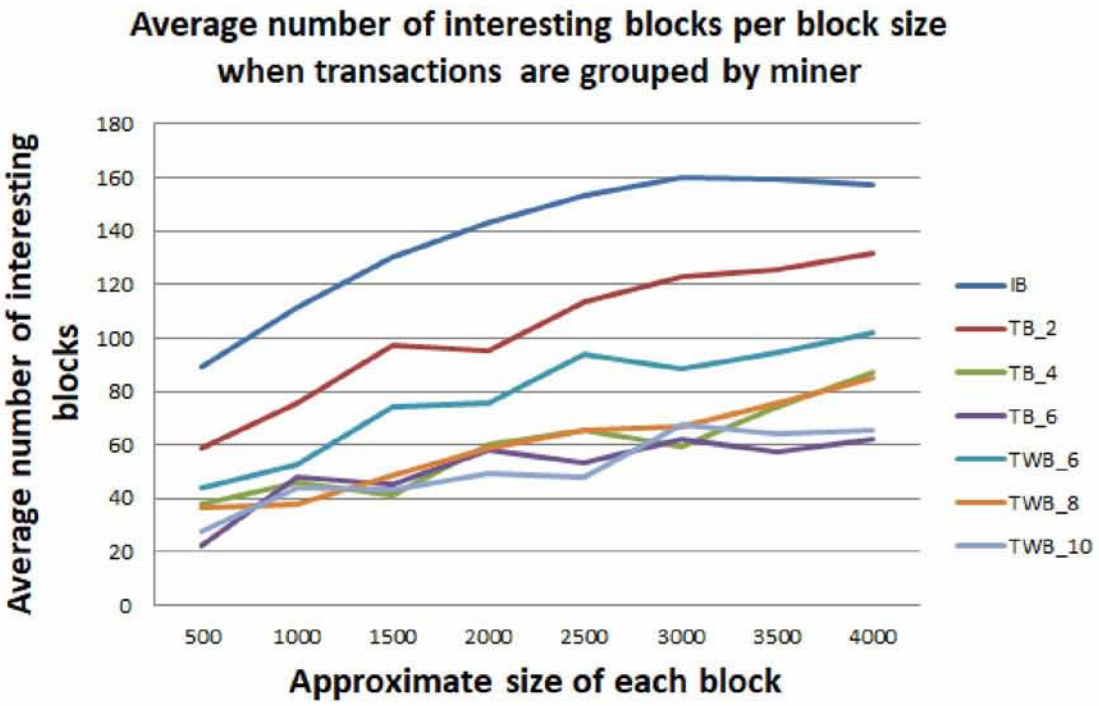
- Τον μέσο όρο του αριθμού των σημαντικών για κάθε κόμβο blocks.
- Τα μεγέθη των σημαντικών για κάθε κόμβο blocks.
- Το hit rate της τοπικής μνήμης όσον αφορά στα blocks που βρίσκονται στην cache. Το hit rate ορίζεται ως το πηλίκον του αριθμού των blocks που βρίσκονται στην τοπική μνήμη ενός κόμβου ως προς τον συνολικό αριθμό των blocks για τα οποία έχει ενδιαφερθεί ο κόμβος αυτός.
- Το ποσοστό των ενδιαφέροντων για τον κόμβο συναλλαγών ως προς τον συνολικό αριθμό των συναλλαγών που βρίσκονται στην τοπική μνήμη.

Στις επόμενες υποενότητες παρουσιάζονται τα αποτελέσματα των πειραμάτων:

5.2.1 Μέσος αριθμός ενδιαφέροντων blocks

Στις γραφικές παραστάσεις που ακολουθούν παρουσιάζονται οι πληροφορίες για τον μέσο αριθμό των ενδιαφέροντων blocks σε κάθε κόμβο. Σε κάθε πείραμα δημιουργήθηκαν **200 blocks** σε ένα δίκτυο **200 κόμβων** τύπου normal node. Δεν δημιουργήθηκαν κόμβοι τύπου light node διότι σε αυτά τα πειράματα εξετάζονται οι αλγόριθμοι αξιολόγησης block που δεν εξαρτώνται από τον κόμβο που τους χρησιμοποιεί.

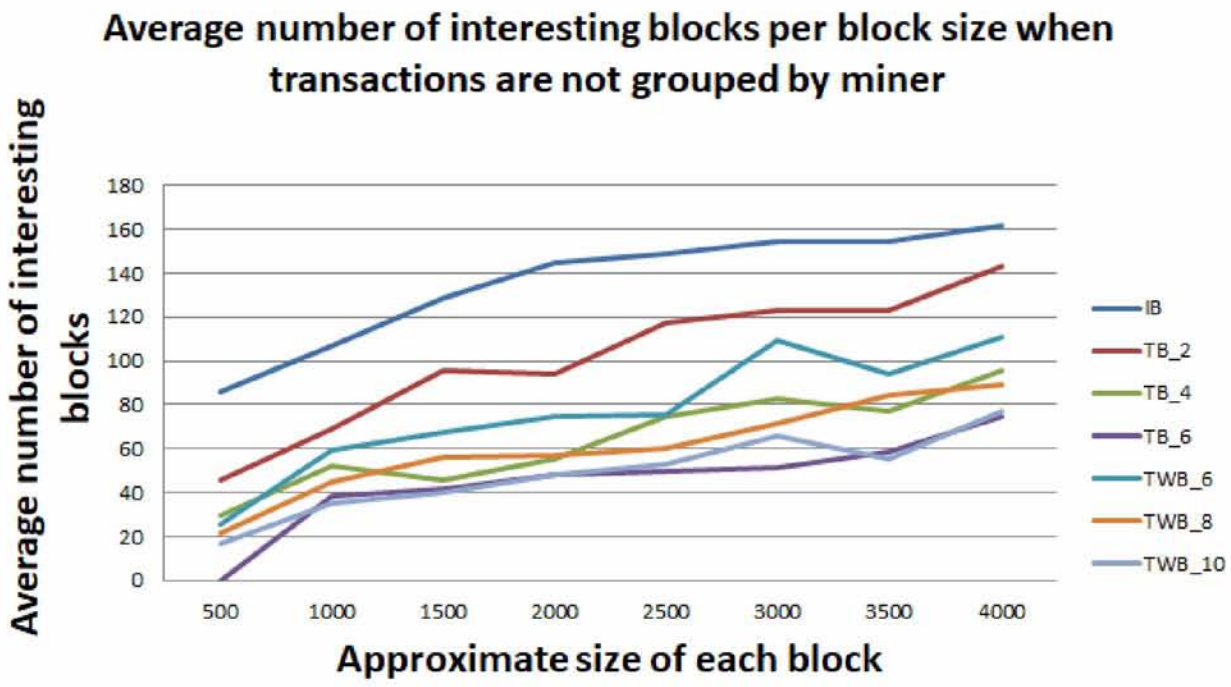
Στην Εικόνα 9 φαίνεται ο μέσος όρος ενδιαφέροντων blocks για κάθε κόμβο όταν τα blocks έχουν συγκεκριμένο μέγεθος, όταν οι συναλλαγές επεξεργάζονται από τον αλγόριθμο group από τον miner node. Κάθε γραμμή στην γραφική παράσταση αντιστοιχίζεται σε έναν αλγόριθμο αξιολόγησης block. Αν αυτός ο αλγόριθμος έχει κάποιο κάτω όριο ενδιαφέροντων συναλλαγών που πρέπει να υπάρχουν στο block, τότε αυτό αναγράφεται μετά το όνομα του αλγορίθμου, έπειτα από τον χαρακτήρα "_" . Επίσης το μέγεθος των blocks είναι το **μέγεθος σε bytes**.



Εικόνα 9. Μέσος αριθμός – group – σταθερά μεγέθη.

Όπως είναι αναμενόμενο ο αλγόριθμος IB θεωρεί περισσότερα block ενδιαφέροντα από τους άλλους αλγορίθμους, αφού σε αυτόν αρκεί να υπάρχει έστω και μία ενδιαφέρουσα συναλλαγή στο block. Στους αλγορίθμους TB και TWB βλέπουμε ότι όταν το κάτω όριο ενδιαφέρουσων συναλλαγών είναι αρκετά υψηλό, ο μέσος αριθμός των ενδιαφέροντων blocks φτάνει το 30% των blocks που δημιουργούνται. Ωστόσο το όριο αυτό μπορεί να αυξηθεί ακόμη περισσότερο από τον κόμβο που χρησιμοποιεί αυτούς τους αλγορίθμους, ελέγχοντας έτσι τον αριθμό blocks που θα θεωρηθούν ενδιαφέροντα.

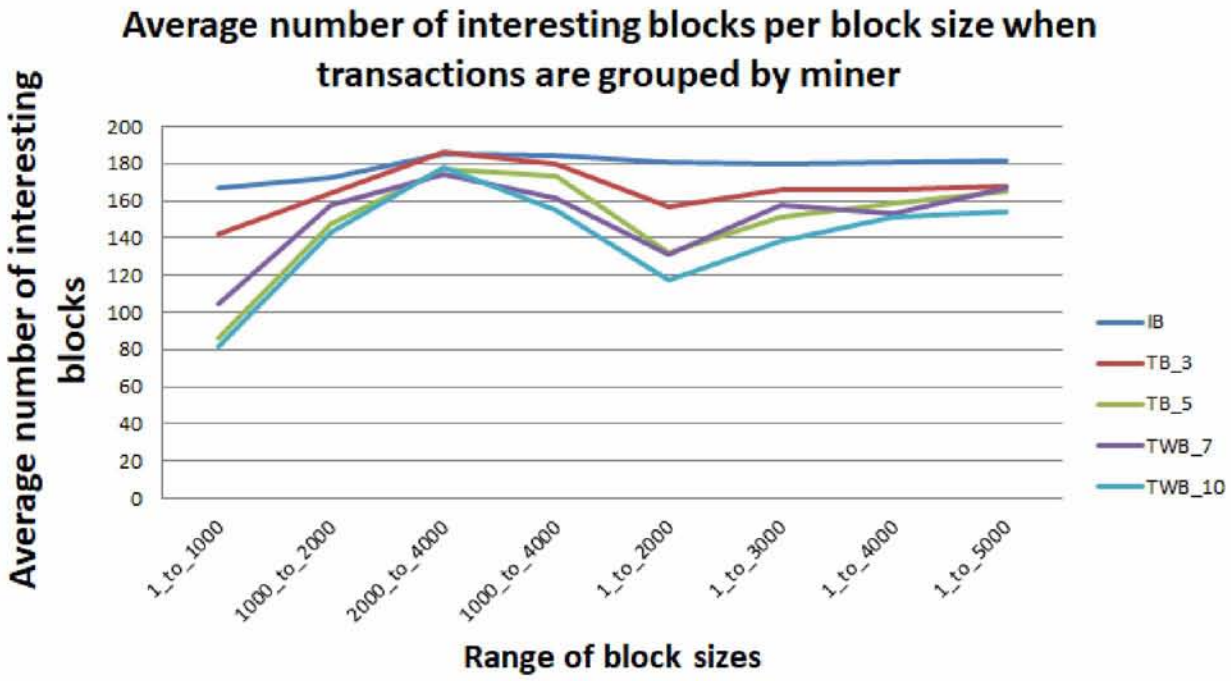
Στην Εικόνα 10 βλέπουμε τα αποτελέσματα της ίδιας διαδικασίας με την μόνη διαφορά να μην πραγματοποιείται ο αλγόριθμος group στον κόμβο miner node.



Εικόνα 10. Μέσος αριθμός – no group – σταθερά μεγέθη.

Από τα παραπάνω βλέπουμε ότι δεν αλλάζουν σημαντικά τα αποτελέσματα όταν δεν χρησιμοποιούμε τον αλγόριθμο group. Αυτό συμβαίνει τα οι τιμές των κλειδιών μπορούν να έχουν ένα αρκετά μεγάλο εύρος. Έτσι όταν ο miner node διαλέγει την δημοφιλέστερη τιμή σε κάθε βήμα, επιλέγει ολόκληρη την συναλλαγή που περιέχει την τιμή αυτή. Υπάρχουν και κόμβοι οι οποίοι ενδιαφέρονται για τις υπόλοιπες τιμές της συναλλαγής αυτής, με αποτέλεσμα από αυτό τον αλγόριθμο να επωφελούνται μόνο οι κόμβοι που ενδιαφέρονται για τις δημοφιλέστερες τιμές. Για τους υπόλοιπους κόμβους δεν υπάρχει κάποια διαφορά στα blocks που λαμβάνουν.

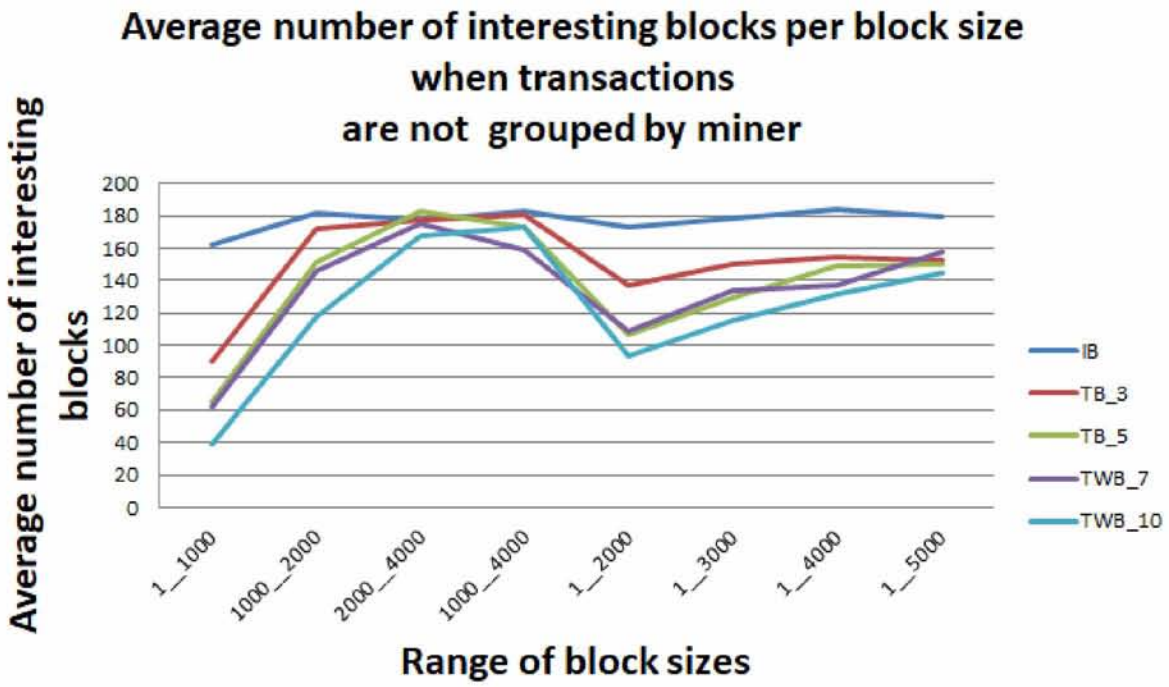
Στην Εικόνα 11 βρίσκονται τα αποτελέσματα για το ίδιο πείραμα στο οποίο όμως το μέγεθος των blocks αλλάζει κάθε φορά και παίρνει μια τυχαία τιμή σε ένα διαθέσιμο διάστημα τιμών. Σε αυτό το πείραμα χρησιμοποιείται ο αλγόριθμος group στον miner node.



Εικόνα 11. Μέσος αριθμός – group – μη σταθερά μεγέθη

Από τα παραπάνω βλέπουμε ότι ο μέσος αριθμός των block που θεωρούνται ενδιαφέροντα από τους κόμβους αυξάνεται δραματικά για τους αλγορίθμους TB και TWB, ειδικά όταν μέγεθος των blocks κυμαίνεται σε ένα μεγάλο διάστημα τιμών. Παρακάτω θα δούμε ότι αυτό συμβαίνει μόνο όταν χρησιμοποιείται ο αλγόριθμος group στον miner node.

Στην Εικόνα 12 φαίνονται τα αποτελέσματα όταν απενεργοποιηθεί ο αλγόριμος group για το ίδιο ακριβώς πείραμα:

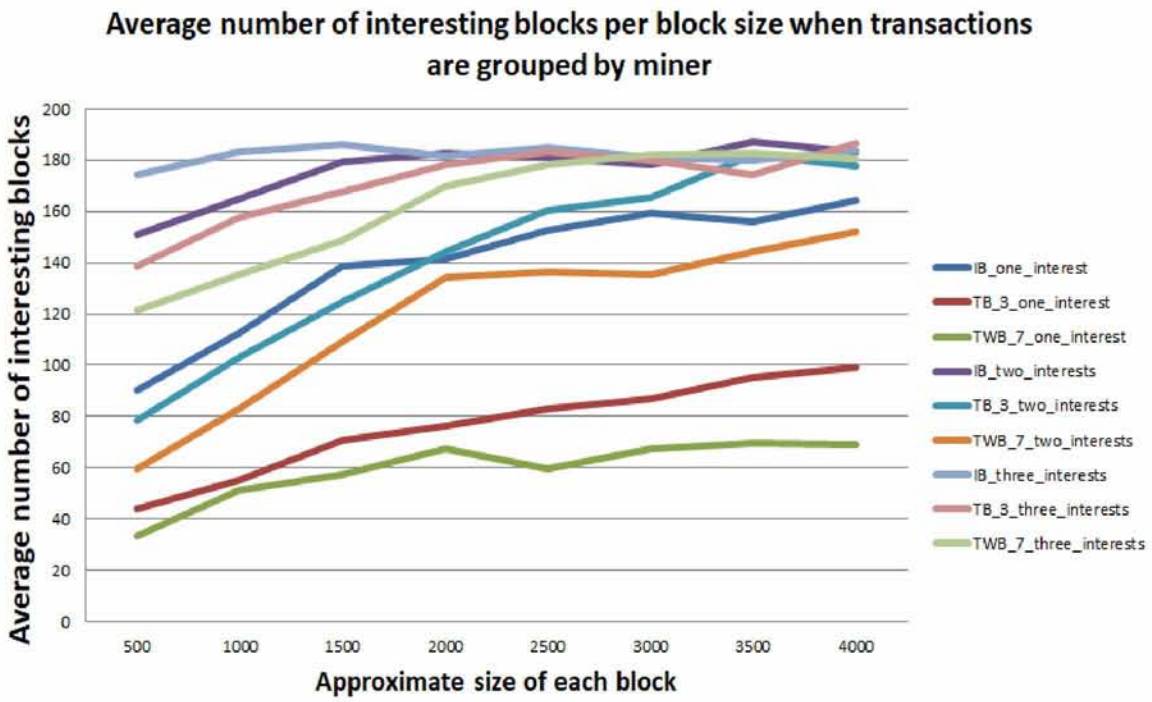


Εικόνα 12. Μέσος αριθμός – group – μη σταθερά μεγέθη

Πάλι παρατηρούμε το ίδιο φαινόμενο, ότι δηλαδή αυξάνεται ο αριθμός των ενδιαφέροντων blocks για τους κόβμους. Ωστόσο αυτό συμβαίνει σε μικρότερο βαθμό απότι στην περίπτωση που χρησιμοποιούμε τον αλγόριθμο group στον miner node.

Από τα παραπάνω συμπεραίνουμε πως δεν συνίσταται η χρήση του αλγόριθμου group όταν τα blocks δεν έχουν συγκεκριμένο μέγεθος. Επίσης συνίσταται τα blocks είτε να έχουν συγκεκριμένο μέγεθος είτε το μέγιστο μέγεθος να μην είναι πολύ μεγάλο (μεγαλύτερο από 2000 bytes στο παραπάνω πείραμα) όταν έχουμε τυχαία μεγέθη block.

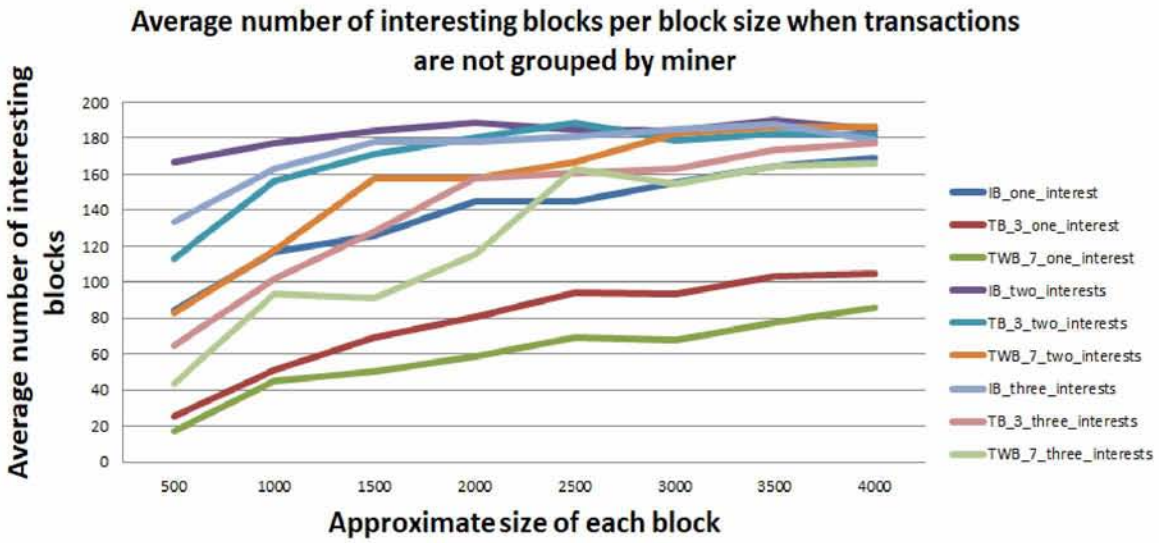
Τέλος στην Εικόνα 13 βλέπουμε τις διαφορές στον μέσο αριθμό των ενδιαφέροντων blocks όταν οι κόμβοι έχουν από 1 έως και 3 ενδιαφέροντα. Αυτό σημαίνει ότι ένας κόμβος ενδιαφέρεται για 3 τιμές τις ίδιας ή διαφορετικών ετικετών. Στην Εικόνα 13 χρησιμοποιήθηκε ο αλγόριθμος group στον miner node.



Εικόνα 13. Μέσος αριθμός – group – σταθερά μεγέθη – πολλαπλά ενδιαφέροντα

Από την Εικόνα 13 βγάζουμε το συμπέρασμα ότι ακόμη και με δύο ενδιαφέροντα ανά κόμβο, τα ενδιαφέροντα blocks αυξάνονται σημαντικά, για όλους του αλγορίθμους.

Στην Εικόνα 14 βλέπουμε την ίδια διαδικασία, χωρίς όμως τον αλγόριθμο group στον miner node.

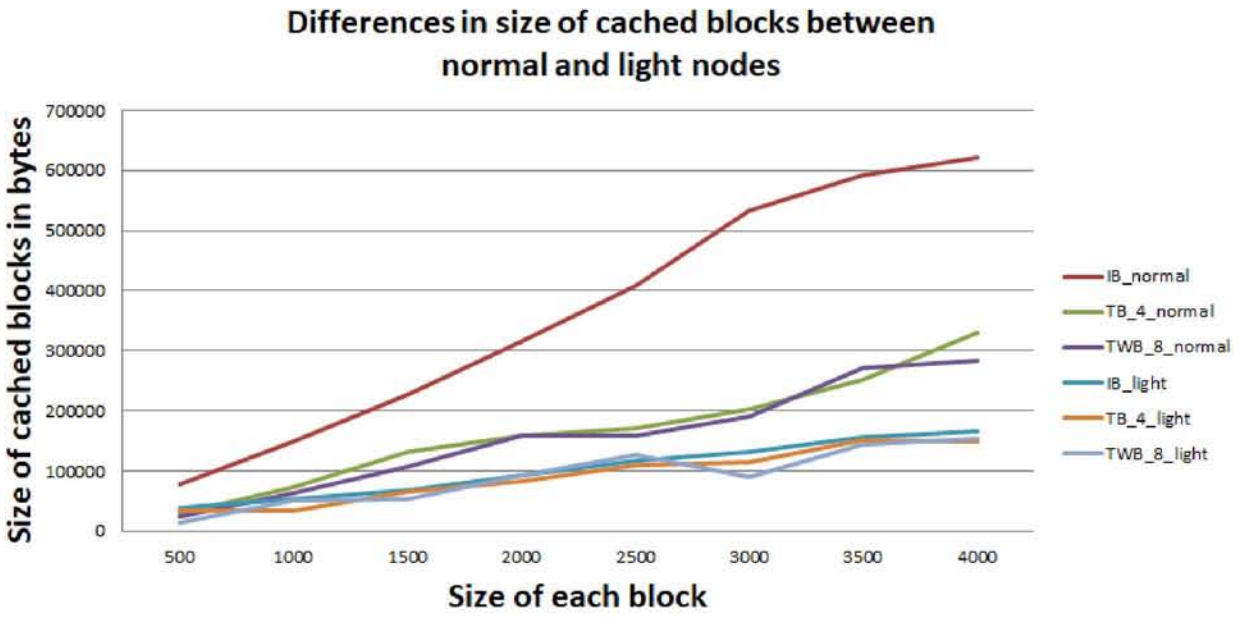


Εικόνα 14. Μέσος αριθμός – no group – σταθερά μεγέθη – πολλαπλά ενδιαφέροντα

Εδώ παρατηρούμε ότι για τους κόμβους με δύο ενδιαφέροντα μειώνεται λίγο ο μέσος αριθμός ενδιαφέροντων blocks για τους αλγορίθμους TB και TWB . Ωστόδο δεν υπάρχει σημαντική διαφορά για τους υπόλοιπους κόμβους, για οποιονδήποτε αλγόριθμο αξιολόγησης.

5.2.2 Μέσο μέγεθος αποθηκευμένων blocks

Σε αυτή την υποενότητα συγκρίνεται το συνολικό μέγεθος των αποθηκευμένων blocks στους κόμβους τύπου normal node σε σχέση με αυτούς στους κόμβους τύπου light node. Έτσι συγκρίνεται το συνολικό μέγεθος που θα είχαν τα block αν αποθηκεύονταν όλα στην τοπική μνήμη. Σε αυτά τα πειράματα θεωρούμε ότι **δεν υπάρχει περιορισμός χώρου στην τοπική μνήμη**. Πάλι σε κάθε πείραμα δημιουργούνται **200 blocks**. Επίσης υπάρχουν σε κάθε πείραμα **100 κόμβοι**, είτε τύπου normal node είτε τύπου light node, ανάλογα με το πείραμα.



Εικόνα 15. Μέσο μέγεθος – group – σταθερά μεγέθη

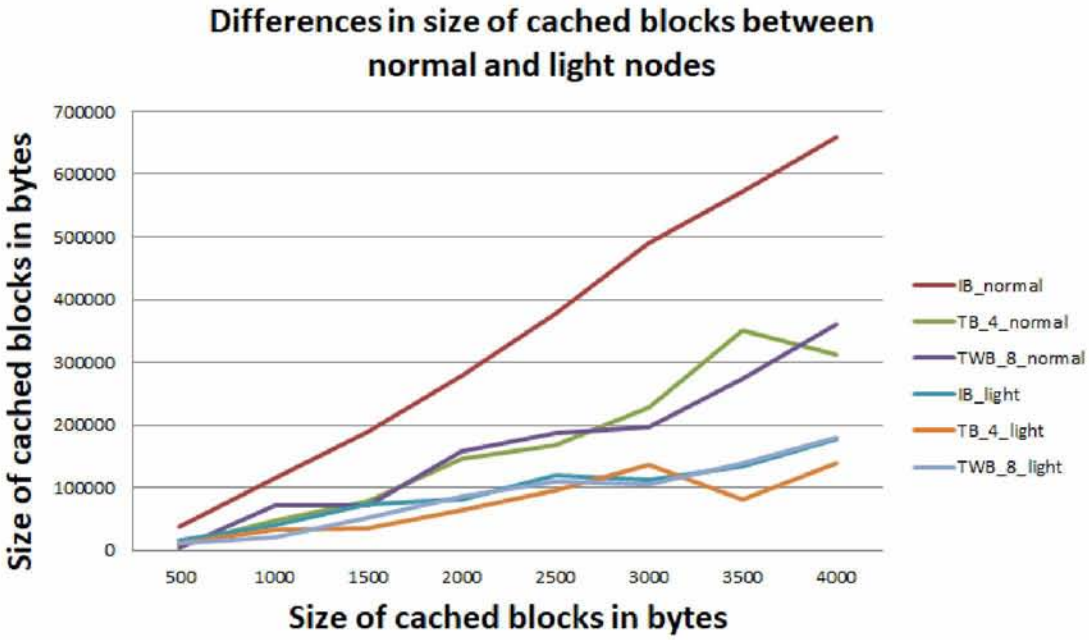
Από την Εικόνα 15 βλέπουμε ότι η διαφορά στο μέγεθος των αποθηκευμένων blocks είναι ιδιαίτερα αντιληπτή όταν χρησιμοποιούμε τον αλγόριθμο IB . Στους αλγορίθμους TB και TWB πάλι υπάρχει διαφορά, η οποία αυξάνεται με το block size, ωστόσο επειδή τα κάτω όρια για τις ενδιαφέρουσες συναλλαγές είναι αρκετά υψηλά, οι κόμβοι δεν αξιολογούν πολλά block ως ενδιαφέροντα για αυτούς. Ωστόσο πάλι, για μεγέθη block μεγαλύτερα των 2000 bytes γίνεται εύκολα αντιληπτή η διαφορά στο συνολικό μέγεθος των αποθηκευμένων block.

Από τα παραπάνω μπορούμε να συμπεράνουμε ότι σε κόμβους όπου χρησιμοποιείται ο αλγόριθμος IB συνίσταται η χρήση κόμβων τύπου light node που κρατάνε μόνο τις σημαντικές για αυτούς συναλλαγές.

Για μικρά μεγέθη block, δεν βλέπουμε μεγάλες διαφορές μεταξύ των συνολικών μεγεθών των αποθηκευμένων block, κυρίως όταν χρησιμοποιούνται οι αλγόριθμοι TB και TWB.

Επίσης φαίνεται να μην επηρεάζει σημαντικά τα αποτελέσματα ο αλγόριθμος που χρησιμοποιεί ο miner node για να παράγει τα blocks.

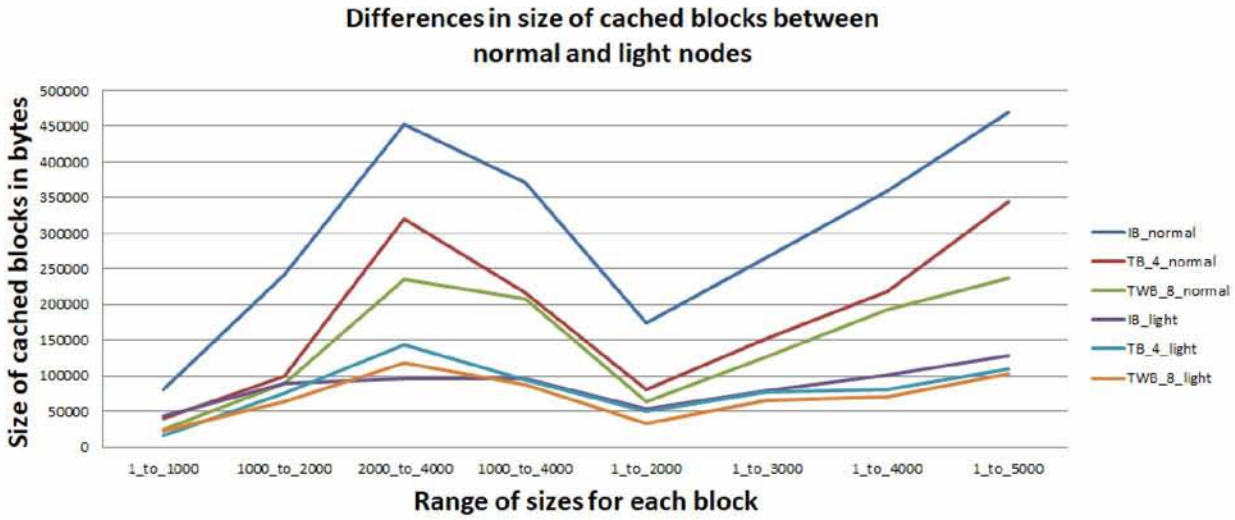
Παρακάτω βλέπουμε το ίδιο πείραμα όταν δεν χρησιμοποιείται ο αλγόριθμος group στον miner node.



Εικόνα 16. Μέσο μέγεθος – no group – σταθερά μεγέθη

Βλέπουμε πως δεν υπάρχει σημαντική διαφορά με τα αποτελέσματα για την έκδοση του πειράματος δίχως τον αλγόριθμο group.

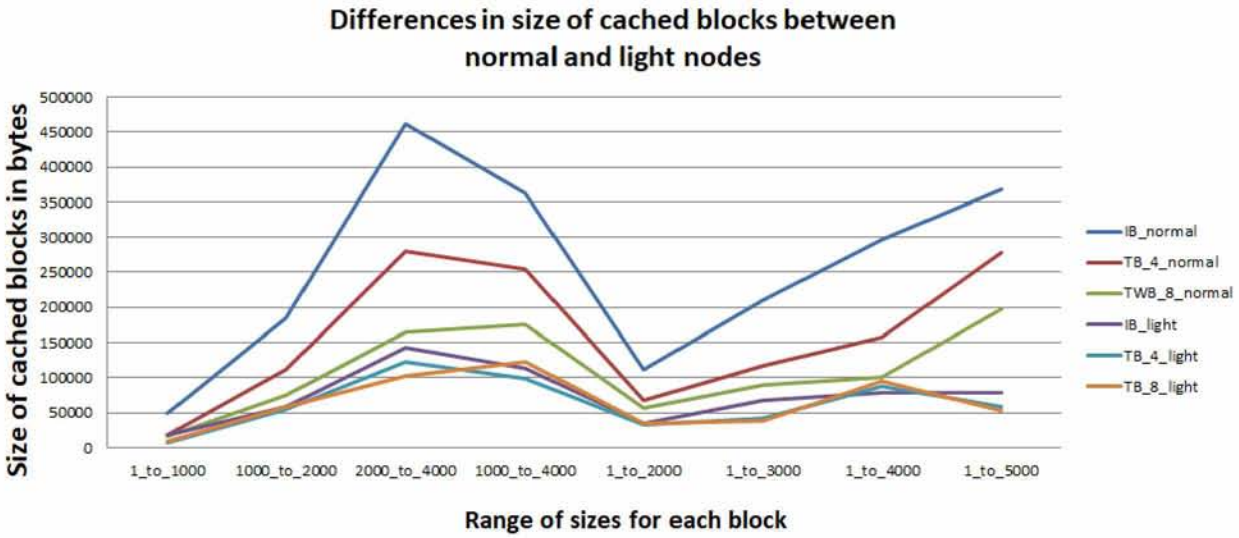
Στην Εικόνα 17 βλέπουμε το ίδιο πείραμα με τα μεγέθη block να επιλέγονται τυχαία από ένα διάστημα τιμών.



Εικόνα 17. Μέσο μέγεθος – group – μη σταθερά μεγέθη

Πάλι όπως ήταν αναμενόμενο βλέπουμε την μεγαλύτερη διαφορά στον αλγόριθμο IB μεταξύ των μέσων όρων για τα συνολικά μεγέθη των αποθηκευμένων block. Επίσης παρατηρούμε ότι η διαφορά μεταξύ των συνολικών μεγεθών των αποθηκευμένων block στους normal nodes και light nodes δεν είναι πλέον τόσο μεγάλη και σε μερικά σημεία μάλιστα είναι σχεδόν ίση με μηδέν.

Τέλος, στην Εικόνα 18 βλέπουμε το ίδιο πείραμα χωρίς τον αλόριθμο group στον miner node.



Εικόνα 18. Μέσο μέγεθος – no group – μη σταθερά μεγέθη

Στην Εικόνα 18 έχουμε ένα ενδιαφέρον φαινόμενο. Σε μερικά σημεία, το μέγεθος των αποθηκευμένων blocks στους κόμβους τύπου light node, είναι το ίδιο και για τους τρεις αλγορίθμους αξιολόγησης των block. Το ενδιαφέρον σε αυτό είναι πως συνήθως οι αλγόριθμοι TB και TWB έχουν καλύτερη απόδοση από τον αλγόριθμο IB, γεγονός το οποίο όμως δεν αντικατοπτρίζεται στην γραφική παράσταση αυτή. Επίσης βλέπουμε μεγάλη διαφορά για μεγάλα διαστήματα μεγεθών για τα blocks στους κόμβους τύπου normal node, κυρίως στα διαστήματα 1 έως 4000 bytes και 1 έως 5000 bytes.

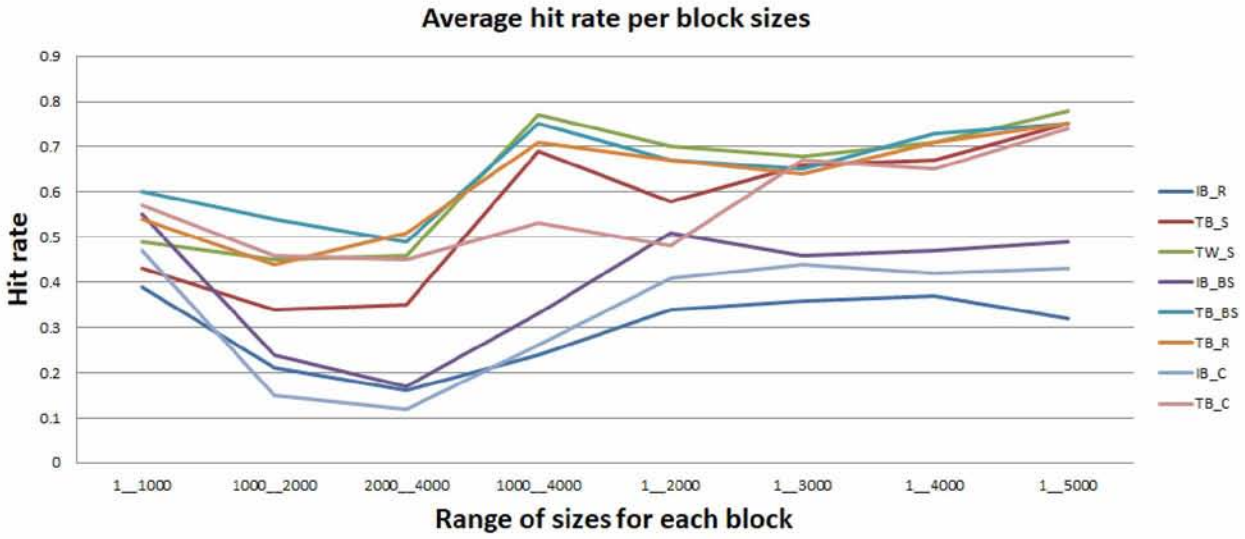
Από τα παραπάνω συμπεραίνουμε ότι οι κόμβοι τύπου light node δεν επηρεάζονται σημαντικά από τα μεγέθη των block όσον αφορά στο συνολικό μέγεθος των αποθηκευμένων blocks. Ωστόσο παρατηρούμε πως για τυχαία μεγέθη block, συχνά το συνολικό μέγεθος των αποθηκευμένων block σε κόμβους τύπου normal node, είναι μικρότερα απότι όταν έχουμε σταθερά μεγέθη για κάθε block.

5.2.3 Hit rate στην τοπική μνήμη

Για να υπολογιστεί το hit rate στην τοπική μνήμη, δημιουργήθηκαν συνολικά πάλι **200 blocks** σε ένα σύστημα **200 κόμβων** . Οι τοπική μνήμη έχει **μέγεθος ίσο με 25% του συνολικού μεγέθους** όλων των block που δημιουργήθηκαν. **Στον αλγόριθμο TB χρησιμοποιήθηκε για κάτω όριο το 4 ενώ για τον αλγόριθμο TWB χρησιμοποιήθηκε σαν κάτω όριο το 8**. **Τα βάρη για τον TWB είναι 1 ή 2.** Το hit rate για έναν κόμβο είναι, όπως αναφέρθηκε, το πηλίκο των blocks που υπάρχουν στην τοπική μνήμη ως προς τον συνολικό αριθμό των ενδιαφέροντων για αυτών blocks. Στα πειράματα συμπεριλήφθηκαν μόνο ρυθμίσεις όπου το μέγεθος των blocks επιλέγεται τυχαία από ένα διάστημα τιμών, καθώς για σταθερό μέγεθος block όλοι οι αλγόριθμοι έχουν το ίδιο αποτέλεσμα.

Στα παρακάτω διαγράμματα, κάθε γραμμή αντιστοιχεί σε έναν συνδυασμό από αλγόριθμο αξιολόγησης και έναν αλγόριθμο διαχείρισης μνήμης. Το πρόθεμα κάθε συντομογραφίας είναι ο αλγόριθμος αξιολόγησης (IB, TB, TWB) και ακολουθεί ο αλγόριθμος διαχείρισης μνήμης. Στο ενδιάμεσο υπάρχει ο χαρακτήρας "_".

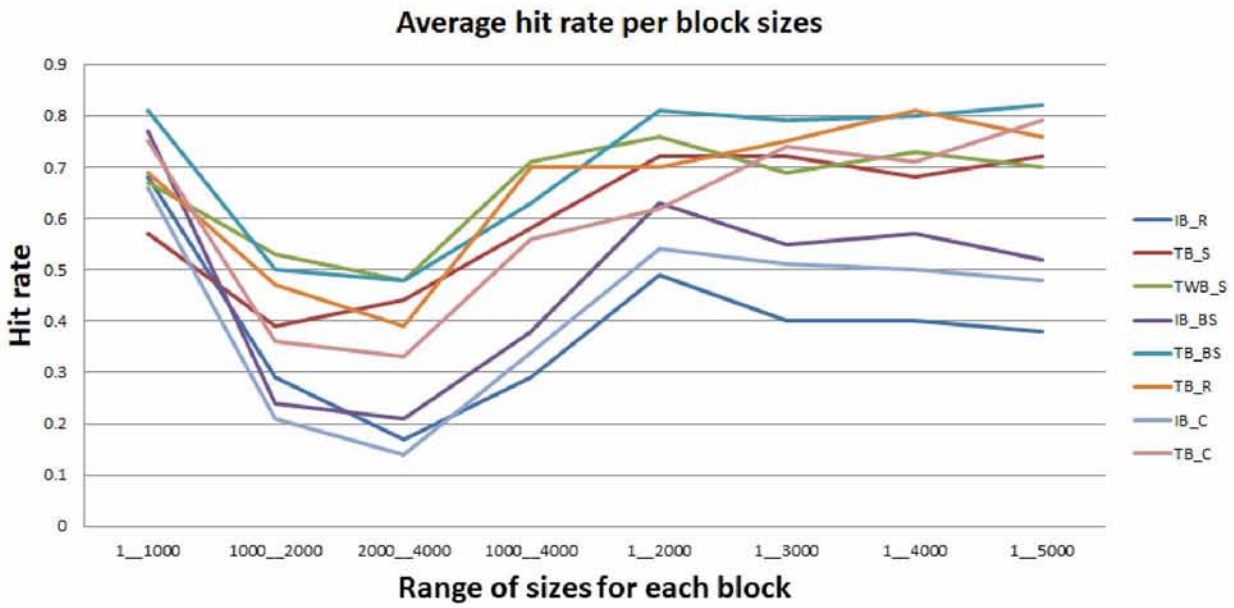
Στην Εικόνα 19 βλέπουμε τα αποτελέσματα για ένα πείραμα όπου χρησιμοποιείται ο αλγόριθμος group στον miner node.



Εικόνα 19. Μέσο hit rate – group – μη σταθερά μεγέθη

Από την Εικόνα 19 βλέπουμε ότι οι συνδυασμοί που χρησιμοποιούν σαν αλγόριθμο αξιολόγησης τον IB έχουν το χαμηλότερο hit rate στην τοπική τους μνήμη. Οι υπόλοιποι αλγόριθμοι έχουν όλοι περίπου το ίδιο hit rate. Η εξήγηση για αυτό είναι το γεγονός πως οι συνδυασμοί που στηρίζονται στους αλγορίθμους TB και TWB, δεν ενδιαφέρονται για ένα μεγάλο ποσοστό των blocks που ελέγχουν, διότι ο αριθμός των ενδιαφέροντων συναλλαγών που βρίσκονται σε αυτά δεν ξεπερνούν τα όρια που έχουν τεθεί.

Στην Εικόνα 20 βλέπουμε το ίδιο πείραμα όταν χρησιμοποιείται ο αλγόριθμος group στον miner node.



Εικόνα 20. Μέσο hit rate – no group – μη σταθερά μεγέθη

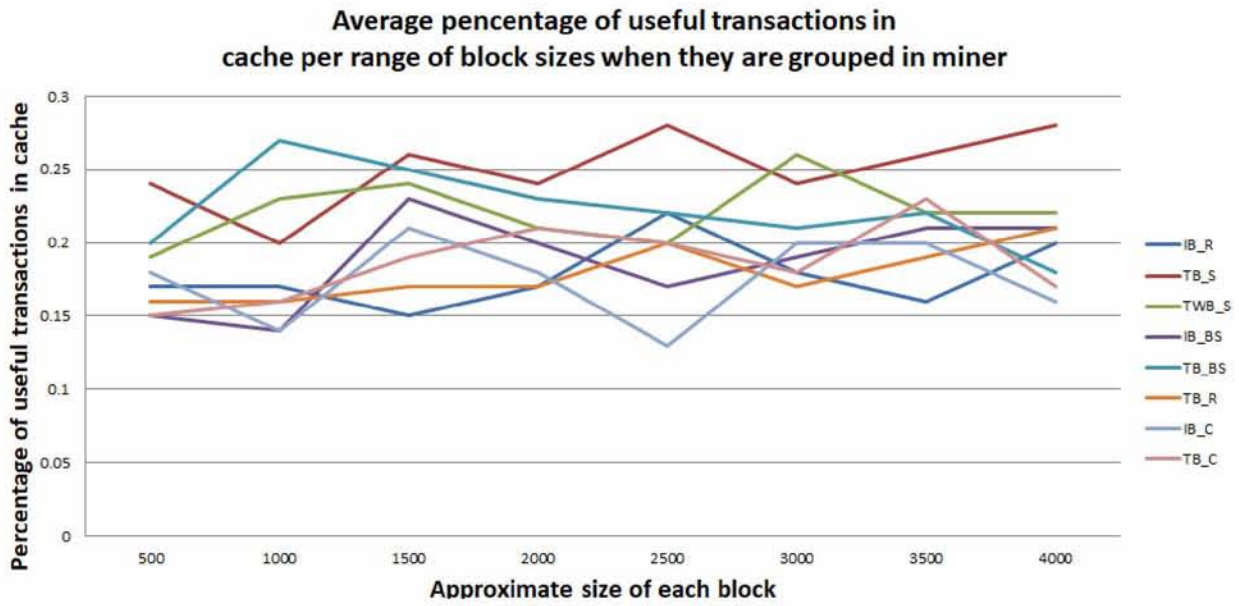
Πάλι βλέπουμε πως οι συνδυασμοί που χρησιμοποιούν σαν αλγόριθμο αξιολόγησης τον αλγόριθμο IB έχουν χαμηλότερο hit rate. Ωστόσο έχουμε μια μικρή αύξηση στο hit rate τους όταν χρησιμοποιείται ο αλγόριθμος group. Τα αποτελέσματα των άλλων συνδυασμών δεν επηρεάζονται σημαντικά από τον αλγόριθμο group.

Από τα παραπάνω κατανοούμε πως χρησιμοποιώντας τους συνδυασμούς με τους αλγορίθμους αξιολόγησης TB και TWB έχουμε ένα μεγαλύτερο hit rate, διότι δεχόμαστε τα blocks με πιο αυστηρά κριτήρια. Ωστόσο ο αλγόριθμος IB επιτρέπει στον κόμβο να μην χάσει κανένα block που έχει έστω και μια ενδιαφέρουσα συναλλαγή. Έτσι εξαρτάται από τον κόμβο αν επιθυμεί να αφιερώσει περισσότερο χώρο ώστε να κρατάει όλα τα ενδιαφέροντα blocks στην μνήμη ή να θέσει ένα υψηλό κάτω όριο ώστε να αγνοεί τα blocks με λίγες ενδιαφέρουσες συναλλαγές.

5.2.4 Ποσοστό ενδιαφέρουσων συναλλαγών στην τοπική μνήμη

Στα πειράματα σε αυτήν την υποενότητα χρησιμοποιήθηκαν πάλι όλοι οι πιθανοί συνδυασμοί από αλγορίθμους αξιολόγησης block και αλγορίθμους διαχείρισης μνήμης. Πάλι δημιουργήθηκαν **200 blocks** σε ένα σύστημα **200 κόμβων τύπου normal node**. Οι τοπική μνήμη έχει **μέγεθος ίσο με 25% του συνολικού μεγέθους** όλων των block που δημιουργήθηκαν. **Στον αλγόριθμο TB χρησιμοποιήθηκε για κάτω όριο το 15% των συναλλαγών ενώ για τον αλγόριθμο TWB χρησιμοποιήθηκε σαν κάτω όριο το πάλι περίπου το 15% των συναλλαγών. Τα βάρη για τον TWB έχουν τιμή 1 ή 2.** Δεν χρησιμοποιήθηκαν κόμβοι τύπου light node διότι αυτοί κρατάνε μόνο τις ενδιαφέρουσες για αυτούς συναλλαγές.

Στην Εικόνα 21 βλέπουμε το μέσο ποσοστό των ενδιαφέρουσων συναλλαγών στην τοπική μνήμη για συγκεκριμένα block sizes, όταν χρησιμοποιείται ο αλγόριθμος group στον miner node.

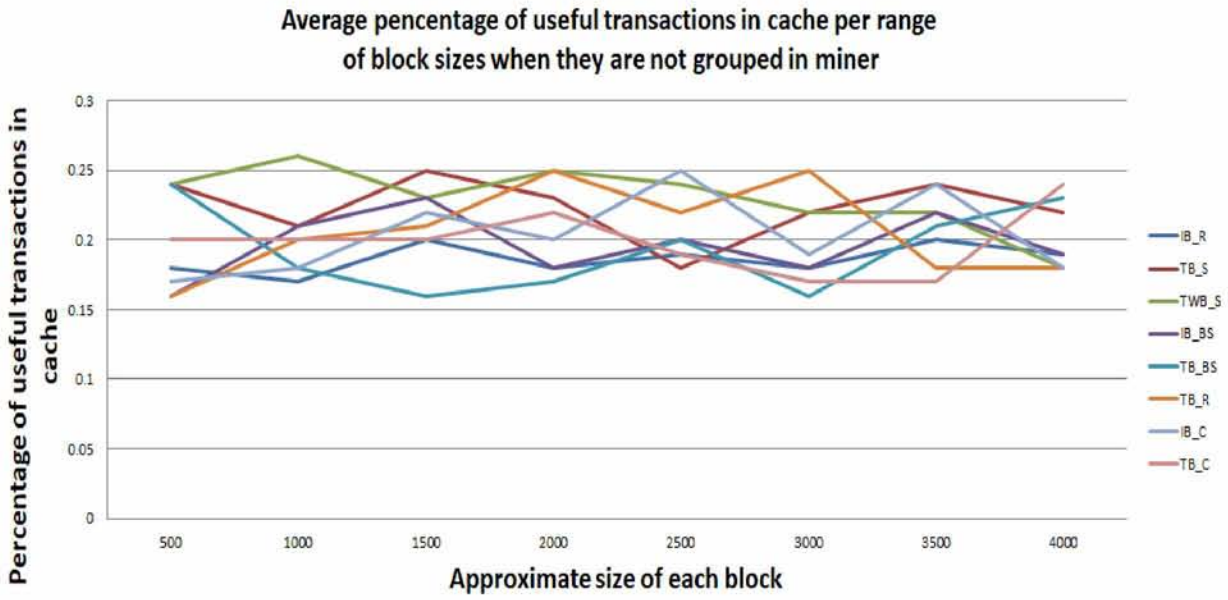


Εικόνα 21. Μέσο ποσοστό χρήσιμων συναλλαγών – group – σταθερά μεγέθη

Από την Εικόνα 21 βλέπουμε ότι οι αποδοτικότεροι συνδυασμοί είναι αυτοί που χρησιμοποιούν σαν αλγόριθμο διαχείρισης μνήμης τους Score based και τους block size based. Οι score based αλγόριθμοι αυτοί δίνουν περισσότερη βάση στον αριθμό των ενδιαφέρουσων συναλλαγών σαν κριτήριο αντικατάστασης των block, οπότε κρατάνε αυτά τα block με τις περισσότερες ενδιαφέρουσες συναλλαγές. Οι block size based αλγόριθμοι κρατάνε τα μικρότερα blocks στην μνήμη, που έχει σαν έμμεσο αποτέλεσμα να μην υπάρχουν πολλές μη ενδιαφέρουσες συναλλαγές στην μνήμη, καθώς εκδιώχνονται τα μεγάλα blocks που έχουν και περισσότερες ασήμαντες συναλλαγές για τον εκάστοτε κόμβο που τους χρησιμοποιεί.

Στην Εικόνα 22 βλέπουμε το ίδιο πείραμα όταν δεν χρησιμοποιείται ο αλγόριθμος group

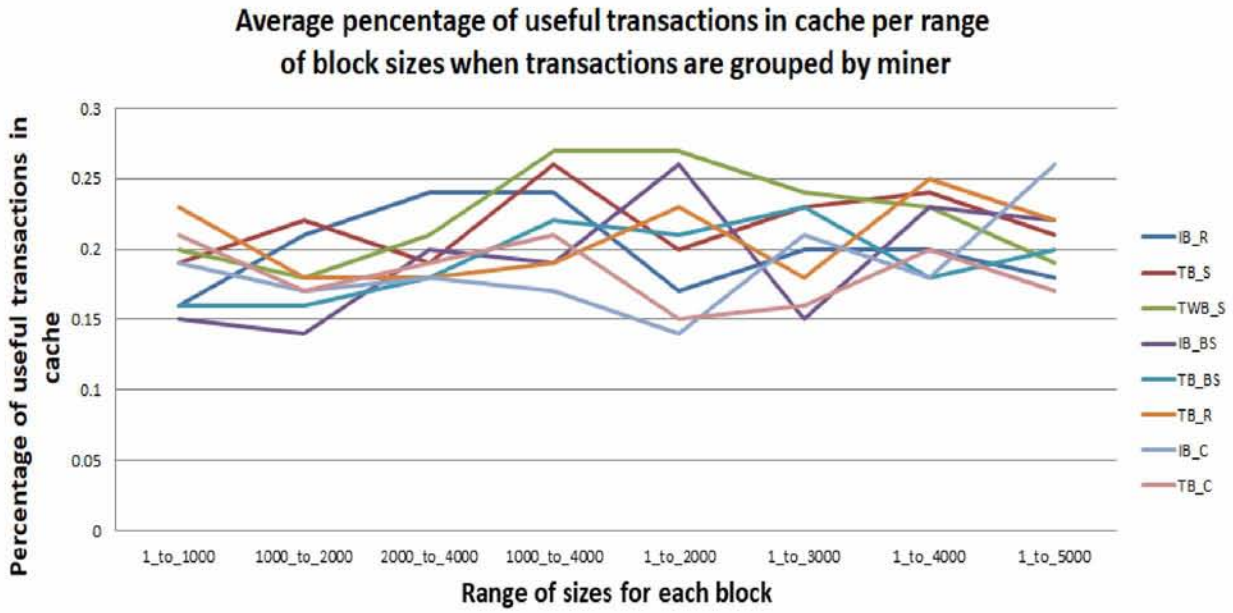
στον miner node.



Εικόνα 22. Μέσο ποσοστό χρήσιμων συναλλαγών – no group – σταθερά μεγέθη

Από την Εικόνα 22 καταλαβαίνουμε ότι δεν έχουμε σημαντική διαφορά στα αποτελέσματα. Μάλιστα μερικά από τα ποσοστά που απεικονίζονται, φαίνεται να μειώνονται όταν χρησιμοποιείται ο αλγόριθμος group. Οι αλγόριθμοι TB και TWB σε συνδυασμό με τον score based αλγόριθμο,έχουν καλύτερη απόδοση σε αρκετά σημεία, ωστόσο όχι σε τόσο μεγάλο βαθμό όπως στην Εικόνα 21 . Οι συνδυασμοί TB με το block size δεν έχουν καλή απόδοση με αυτές τις ρυθμίσεις.

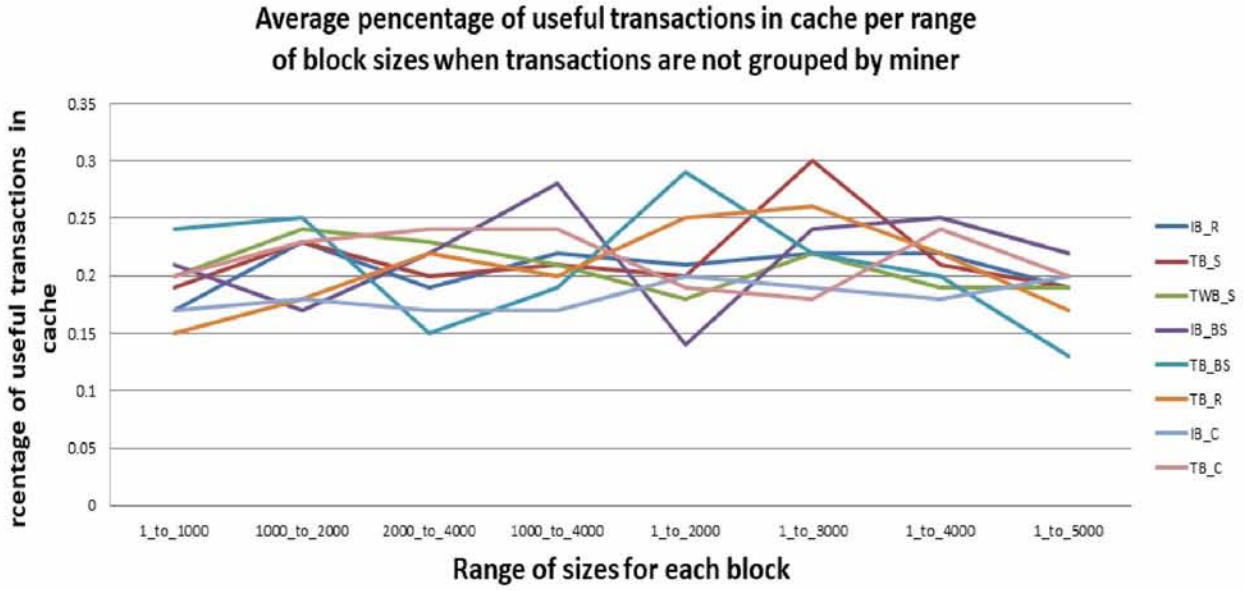
Στην Εικόνα 23 βλέπουμε το ίδο πείραμα όταν τα μεγέθη των block επιλέγονται τυχαία από ένα διάστημα μεγεθώνπου δίνεται. Επίσης χρησιμοποιείται ο αλγόριθμος group.



Εικόνα 23. Μέσο ποσοστό χρήσιμων συναλλαγών – group – μη σταθερά μεγέθη

Στην Εικόνα 23 βλέπουμε ότι η απόδοση των αλγορίθμων εξαρτάται σε μεγάλο βαθμό από τα μεγέθη των block που δημιουργούνται. Αλγόριθμοι που για κάποια διαστήματα τιμών για τα μεγέθη είναι οι αποδοτικότεροι, βρίσκονται στο τέλος της κατάταξης σε άλλα διαστήματα τιμών. Βλέπουμε ωστόσο ότι πάλι οι score based αλγόριθμοι σε συνδυασμό με τους TB και TWB έχουν καλή απόδοση. Επίσης καλή απόδοση στα περισσότερα σημεία έχουν οι block size based αλγόριθμοι για τους ίδιους αλγορίθμους αξιολόγησης block.

Τέλος στην Εικόνα 24 φαίνεται η ίδια διαδικασία όταν δεν χρησιμοποιείται ο αλγόριθμος group από τον miner node.



Εικόνα 24. Μέσο ποσοστό χρήσιμων συναλλαγών – no group – μη σταθερά μεγέθη

Πάλι δεν μπορούμε να βγάλουμε συμπέρασμα για τους αποδοτικότερους συνδυασμούς μεταξύ αλγορίθμων αξιολόγησης block και διαχείρισης μνήμης του κόμβου. Επίσης σε μερικά σημεία βλέπουμε τιμές να είναι πολύ κοντά στο 30 τις εκατό, ωστόσο υπάρχουν και σημεία κοντά στο ποσοστό των 15 τα εκατό.

Από τα παραπάνω μπορούμε να βγάλουμε το συμπέρασμα πως αν θέλουμε να έχουμε την δυνατότητα να προβλέψουμε την αποδοτικότητα των αλγορίθμων, τότε συνίσταται να χρησιμοποιηθούν οι αλγόριθμοι TB και TWB σε συνδυασμό με τους score based αλγορίθμους, όταν χρησιμοποιείται ο αλγόριθμος group στο σύστημα και τα blocks έχουν σταθερό μέγεθος.

6 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

6.1 Συμπεράσματα εργασίας

Είδαμε πως για κάθε πιθανή προτίμηση ενός κόμβου, υπάρχει ένας ανάλογος αλγόριθμος αξιολόγησης block.

Αν ο κόμβος επιθυμεί να λαμβάνει όλη την ενδιαφέρουσα για αυτόν πληροφορία τότε μπορεί να επιλέξει τον αλγόριθμο IB. Εάν ωστόσο επιθυμεί να κρατάει μόνο blocks με υψηλό ποσοστό ενδιαφέρουσας για αυτόν πληροφορίας τότε μπορεί να επιλέξει έναν από τους αλγορίθμους TB ή TWB.

Όσον αφορά στους αλγόριθμους διαχείρισης μνήμης, οι κόμβοι θέτουν με δικά τους κριτήρια τι θεωρούν σημαντικότερο να κρατήσουν.

- Κόμβοι που επιθυμούν να κρατάνε πρόσφατη πληροφορία μπορούν να χρησιμοποιήσουν τον Recency based αλγόριθμο.
- Κόμβοι που επιθυμούν να αποθηκεύουν όσο το δυνατόν περισσότερα blocks στην μνήμη, μπορούν να επιλέξουν τον block size based αλγόριθμο.
- Κόμβοι που επιθυμούν να αποθηκεύσουν όσο το δυνατόν περισσότερες ενδιαφέρουσες συναλλαγές, μπορούν να επιλέξουν τους score based αλγόριθμους.
- Κόμβοι που βρίσκονται σε ασύρματα δίκτυα, όπου είναι σημαντικοί οι περιορισμοί του διαθέσιμου εύρους ζώνης και η παρουσία θορύβου στο δίκτυο, μπορούν να επιλέξουν τον cost based αλγόριθμο.

Από τα παραπάνω συμπεραίνουμε πως δεν υπάρχει ένας αντικειμενικά καλύτερος συνδυασμός αλγορίθμων αξιολόγησης block και αλγορίθμων διαχείρισης μνήμης για όλες τις περιπτώσεις χρήσης. Ο κάθε κόμβος σε ένα δίκτυο που χρησιμοποιεί αυτήν την εφαρμογή ή μια παραλλαγή της μπορεί ξεχωριστά να επιλέξει όποιον συνδυασμό επιθυμεί. Αυτό συμβαδίζει με το γεγονός πως οι εφαρμογές blockchain έχουν έναν αποκεντρωμένο χαρακτήρα, με την έννοια ότι δεν υπάρχει ένα κεντρικό σημείο στο δίκτυο μέσω του οποίου ρέει όλη η πληροφορία του δικτύου.

6.2 Προτάσεις για μελλοντική εργασία

Όπως αναφέρθηκε στους περιορισμούς εργασίας, στην εφαρμογή blockchain που υλοποιήθηκε, δεν λήφθηκαν υπόψην δύο σημαντικά χαρακτηριστικά πραγματικών εφαρμογών blockchain.

Για παράδειγμα θεωρούμε πως οι κόμβοι συμφωνούν με τετριμμένο τρόπο για την κατάσταση του blockchain σε κάθε χρονική στιγμή. Ωστόσο αυτός ο παράγοντας είναι ίσος ο σημαντικότερος σε εφαρμογές blockchain. Δεν είναι εγγυημένο ότι οι κόμβοι λειτουργούν σύμφωνα με τους κανονισμούς που τίθονται σε μια πραγματική εφαρμογή blockchain. Σε πραγματικές εφαρμογές εφαρμόζονται μέτρα για την προστασία του blockchain από κακοπροαίρετους χρήστες. Έτσι είναι απαραίτητο να σχεδιαστούν κατάλληλοι αλγόριθμοι για να επιτευχθεί η συμφωνία μεταξύ των κόμβων ως προς την κατάσταση του blockchain σε μια εφαρμογή η οποία υποστηρίζει την μορφή caching που περιγράφεται στην παρούσα διπλωματική εργασία.

Επιπλέον, τα περιεχόμενα των συναλλαγών δεν ελέγχονται από τους κόμβους ως προς

την ορθότητα τους. Έτσι χρειάζεται και ένας αλγόριθμος ώστε να μπορούν οι κόμβοι να ελέγχουν την εγκυρότητα των περιεχομένων των συναλλαγών, σε μια εφαρμογή όπως αυτή που προτείνεται.

Τέλος, είναι απαραίτητο να προταθεί ένας τρόπος διοχέτευσης των μηνυμάτων-αιτήσεων σε μια εφαρμογή blockchain που υποστηρίζει τις λειτουργίες της υλοποιημένης εφαρμογής.

7. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system.

[2] E. Project Ethereum Homepage https://www.ethereum.org/

[3] L. Project Litecoin. https://litecoin.org/

[4] D. Project Dogecoin homepage. http://dogecoin.com/

[5] Sarah Underwood. Communications of the ACM, November 2016, Vol. 59 No. 11, Pages 15 - 17 DOI: 10.1145/299458

[6] Odini M Understanding Blockchain Slideshare presentation February 16, 2017, slideshare.net

[7] Merkle, R. C. (1988). "A Digital Signature Based on a Conventional Encryption Function". Advances in Cryptology — CRYPTO '87. Lecture Notes in Computer Science. 293. p. 369. Doi:10.1007/3-540-48184-2_32. ISBN 978-3-540-18796-7.

[8] *SHA-256*, October 2007, [online] Available:http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA256.pdf

[9] Aggarwal, Charu & Wolf, Joel & Yu, Philip. (2000). Caching on the World Wide Web. IEEE Transactions on Knowledge and Data Engineering. 11. 10.1109/69.755618.