Κατανεμημένα Συστήματα

Εξαμηνιαία Εργασία

NoobCash

Κατάκης Νικηφόρος Εμμανουήλ – 03115120

Κατσίρος Δημήτριος – 03115115

Σκοπός

Μέσω της εργασίας αυτής υλοποιήσαμε το σύστημα noobcash, το οποίο υλοποιεί και βασίζεται στην τεχνολογία του blockchain. Η εφαρμογή μας αποτελείται από ένα backend, μέσω του οποίου επικοινωνούν οι κόμβοι του δικτύου μεταξύ τους, και ενός cli εργαλείου μέσω του οποίου ο χρήστης μπορεί να συνδεθεί ως συμμετέχοντας του συστήματος και να δώσει διάφορες εντολές που θα προωθηθούν στο backend.

Αρχιτεκτονική

Επικοινωνία: Για να έχουν την δυνατότητα οι κόμβοι να επικοινωνήσουν μεταξύ τους, κατασκευάσαμε ένα API. Με αυτόν τον τρόπο, οι participants ανταλλάσσουν πληροφορίες μέσω get και post requests. Το framework που χρησιμοποιήσαμε για την υλοποίηση του API είναι το Django της Python, ενώ για το cli χρησιμοποιήσαμε απλή Python.

Όσον αφορά στον miner, κάθε φορά που κρίνεται απαραίτητη η διαδικασία του mining εκτελείται σαν ένα ξεχωριστό process παράλληλα με τον main server και τελικά τον ενημερώνει μέσω request, μόλις καταφέρει να βρει nonce και να λύσει το πρόβλημα. Επίσης ο main server, έχει την δυνατότητα οποιαδήποτε στιγμή να «σκοτώσει» τον miner, άμα κριθεί απαραίτητο.

Δομές Blockchain: Ο κάθε κόμβος διατηρεί ένα state, το οποίο περιλαμβάνει μεταξύ άλλων τις εξής ζωτικές πληροφορίες:

- Transactions
- Blockchain
- UTXOs
- Valid UTXOs

Συγκεκριμένα, τα transactions που κρατάμε στο state είναι αυτά που δεν έχουν μπει ακόμα σε κάποιο block αλλά έχουν γίνει validated. Αντίστοιχα τα Valid UTXOs είναι μία λίστα από utxos των transactions τα οποία έχουν μπει σε κάποιο block, ενώ τα απλά UTXOs είναι το set το οποίο ουσιαστικά προκύπτει εφαρμόζοντας τα Transactions πάνω στα Valid UTXOs.

Περαιτέρω Ορισμοί: Για την δημιουργία ή την επαλήθευση ενός transaction χρησιμοποιούμε το UTXOs set και ελέγχουμε τα εξής:

- Αν έχει valid signature
- Αν έχει valid id (προκύπτει από το hashing των δεδομένων του transaction)
- Αν ο sender, μέσω των utxo του, έχει αρκετά χρήματα
- Στην περίπτωση του validation ελέγχουμε αν τα inputs είναι όντως utxos του αποστολέα.

Όσον αφορά στα blocks με την δημιουργία και επαλήθευση τους ελέγχουμε τα παρακάτω:

- Αν έχει valid δομή, δηλαδή σωστό nonce και hash με βάση το difficulty που έχουμε ορίσει
- Αν έχει το σωστό μέγεθος transactions με βάση το Block capacity
- Αν όλα τα transactions είναι valid
- Αν έχει σωστό index και το previous hash του είναι το hash του τελευταίου block του blockchain

Μία ακόμα παραδοχή που κάναμε για την καλύτερη λειτουργία του συστήματος μας, είναι τα transactions τα οποία δεν μπαίνουν κατευθείαν σε κάποιο block δεν τις πετάμε, αλλά τις κρατάμε. Στην συνέχεια όταν θα ξανακαλεστεί ο miner τότε θα έχουν την δυνατότητα να εισαχθούν στο blockchain. Αυτό μπορεί να μην ταιριάζει ακριβώς με την λειτουργία του blockchain, αλλά λόγω του γεγονότος ότι στέλναμε 100+ transactions ταυτόχρονα οδηγούσε σε αρκετά να χάνονται, για αυτό και αποφασίσαμε μετά την δημιουργία ενός block να «ξαναπαίζουμε» τα transactions που δεν μπήκαν στο block ώστε να μην χαθούν.

Επίσης όσον αφορά τα public και τα private key του κάθε participant, προσπαθήσαμε να χρησιμοποιήσουμε κρυπτογράφηση που χρησιμοποιείται και πρακτικά από το blockchain. Για αυτό επιλέξαμε την βιβλιοθήκη pycryptodome της python, η οποία μας δίνει πρόσβαση στο SHA256 και στο RSA, που χρησιμοποιήθηκε για την δημιουργία των κλειδιών.

Το consensus χρησιμοποιείται σε περίπτωση που κάποιος participant κάνει receive ένα block του οποίου το previous hash δεν είναι κάποιου άλλου block του blockchain μας, που σημαίνει ότι σίγουρα θα υπάρχει κάποιο conflict. Για αυτό ο συγκεκριμένος participant θα ζητήσει από όλους τα άλλους τα blockchain τους και αφού τα κάνει validate, θα κρατήσει το μεγαλύτερο.

Σενάρια Χρήσης

CLI requests: Ο χρήστης έχει την δυνατότητα μέσω του cli να κάνει αίτημα για ένα νέο transaction, να δει το blockchain, όπως και επίσης το υπόλοιπο του. Εμείς για απλότητα επιλέξαμε να είναι ορατό το υπόλοιπο όλων των participant για τον ευκολότερο έλεγχο του συστήματος.

Νέο Transaction: Σε περίπτωση που κάποιος κόμβος «φτιάξει» ένα καινούριο transaction θα το προσθέσει στο set των transactions και θα ανανεώσει και τα utxos. Στην συνέχεια θα το κάνει broadcast και στους υπόλοιπους κόμβους, οι οποίοι αφού το πιάσουν θα το κάνουν validate και θα ανανεώσουν και τις δικές τους δομές αντίστοιχα.

Mining: Κάθε φορά που έχουμε την προσθήκη ενός transaction τότε ο κάθε κόμβος επιχειρεί να ξεκινήσει τον miner του. Συγκεκριμένα, ο miner θα ξεκινάει αν ο αριθμός των transaction έχει ξεπεράσει το ορισμένο block capacity. Το mining δεν

είναι απαραίτητο πως θα ολοκληρωθεί, καθώς σε περίπτωση που ο κόμβος λάβει κάποιο block θα τερματίσει τον δικό του miner. Αν όμως καταφέρει και ολοκληρώσει το mining τότε το subprocess θα ενημερώσει μέσω request το main process για την δημιουργία του νέου block και θα του στείλει τα στοιχεία του.

Νέο Block: Με την ολοκλήρωση του mining ο κόμβος θα προσπαθήσει να δημιουργήσει ένα νέο block και στην συνέχεια θα το κάνει broadcast και στους υπόλοιπους. Ανανεώνονται κατάλληλα και τα valid utxos όπως επίσης και τα transactions. Σε περίπτωση που κάποιος άλλος κόμβος έχει conflict, όπως αναφέραμε παραπάνω θα ενεργοποιηθεί η διαδικασία του consensus.

Γενική Λειτουργία NoobCash

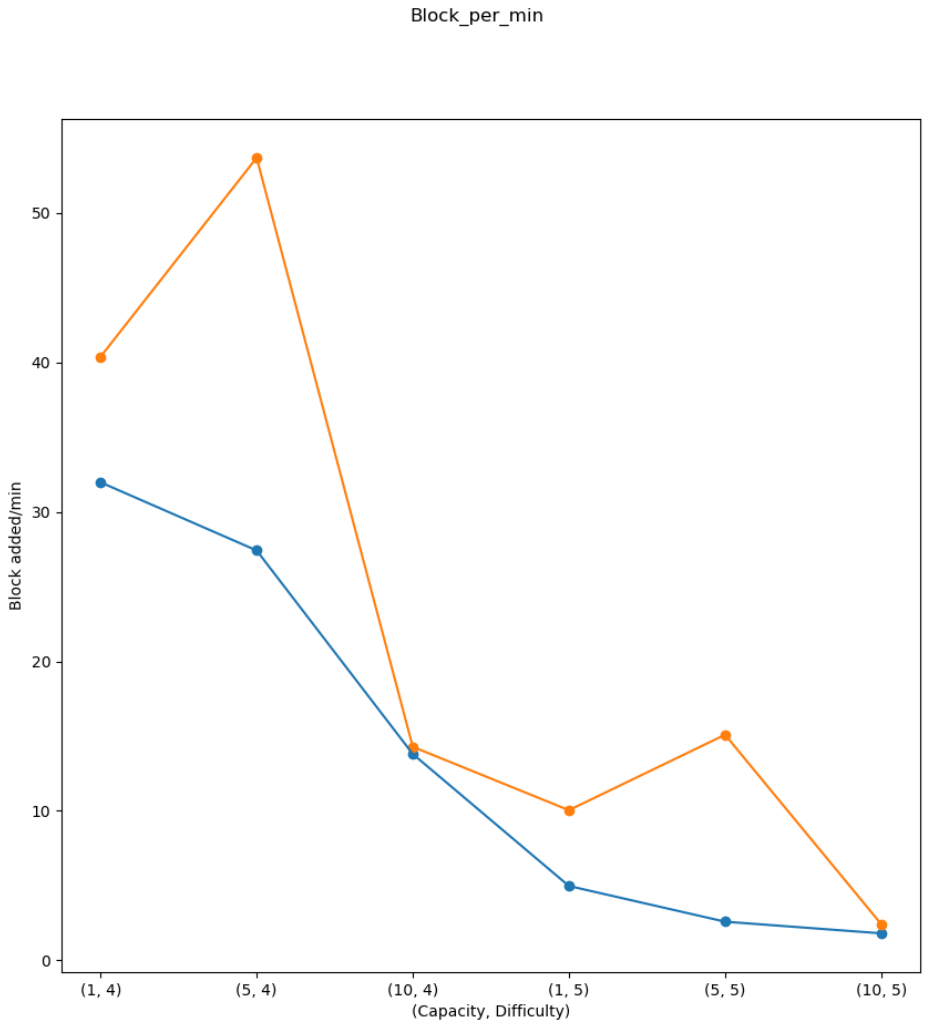
Για την έναρξη του noobcash κάθε participant ξεκινάει ένα σέρβερ στον host του και στο port που θέλει. Ως default coordinator έχουμε ορίσει εμείς το PC1 με διεύθυνση 192.168.1.5 και ως coordinator port έχουμε το 8000. Επίσης ο κάθε χρήστης συνδέεται και ως client μέσω του cli.py, και δηλώνει σε ποιον participant αντιστοιχεί. Στην περίπτωση του coordinator δηλώνει επίσης σαν όρισμα και ποιος θα είναι ο συνολικός αριθμός των participant. Με το που συνδεθεί ο coordinator δημιουργείται το genesis block και παίρνει και ο ίδιο 100*n nbc, όπου n ο αριθμός των participants. Στην συνέχεια, όταν συνδεθούν όλοι οι participants, τότε ο coordinator τους δίνει μέσω ενός transaction τα 100 nbc που αντιστοιχούν στον καθένα και το σύστημα είναι έτοιμο να λειτουργήσει.

Μετρήσεις

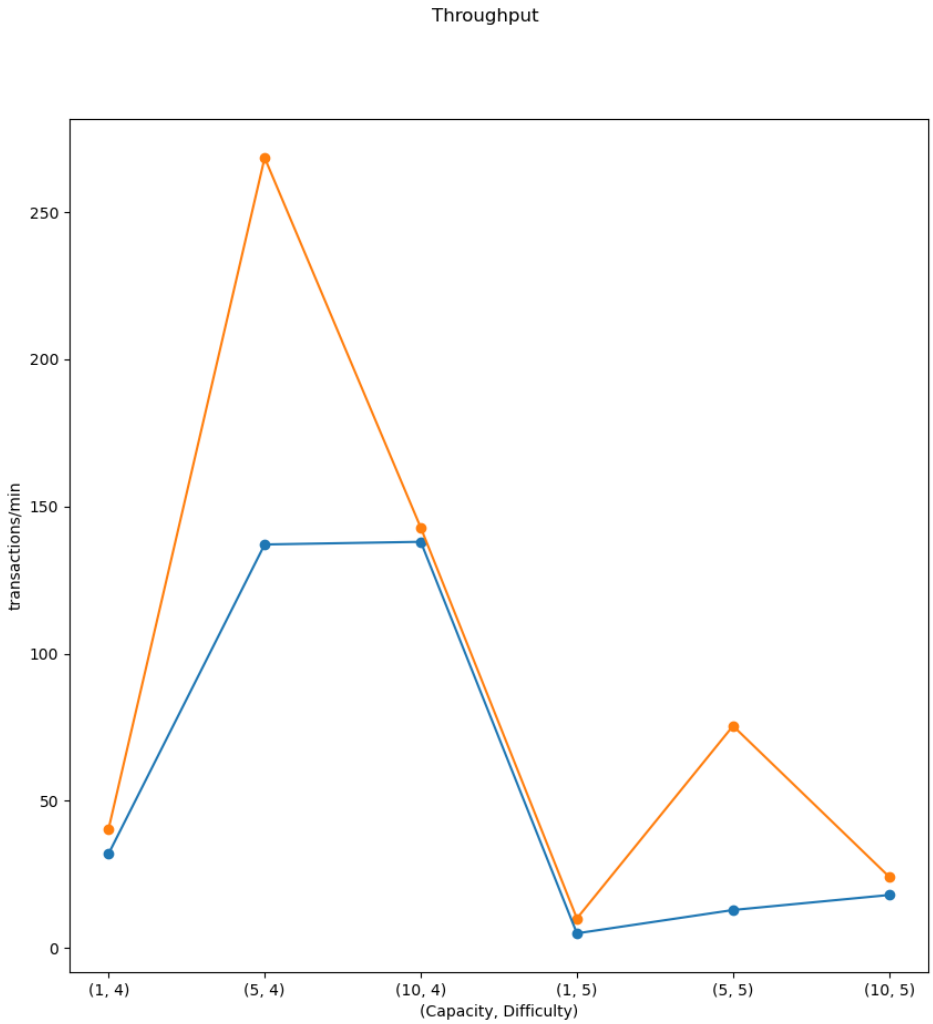
Τέλος για τις διάφορες τιμές block capacity, participators και PoW difficulty τρέξαμε τα backends και εκτελέσαμε για κάθε κόμβο ταυτόχρονα, τα σενάρια τα οποία μας δόθηκαν. Κρατήσαμε σε log files διάφορα αποτελέσματα ώστε τελικά να υπολογίσουμε τα δύο ζητούμενα και να τα παρουσιάσουμε σε γραφικές παραστάσεις, για να μελετηθούν.

- Block time. Ως block time, υπολογίσαμε τον μέσο χρόνο που χρειάζεται για να προστεθεί ένα νέο block στην αλυσίδα, χρησιμοποιώντας το timestamp του τελευταίου block που προστέθηκε και τον συνολικό αριθμό των block του τελικού blockchain.
- Throughput. Ως throughput, υπολογίσαμε τον μέσο αριθμό transaction που μπαίνουν σε block ανά λεπτό.

Παρακάτω έχουμε λοιπόν τις γραφικές παραστάσεις των αποτελεσμάτων. Με πορτοκαλί χρώμα είναι η γραφική των 10 participators και με μπλε χρώμα είναι η γραφική των 5 participators.



Τα αποτελέσματα που έχουμε όσον αφορά τον ρυθμό με τον οποίο προστίθενται τα blocks είναι και αυτά που περιμέναμε. Δηλαδή, όπως παρατηρούμε, με μεγαλύτερο αριθμό participators η ταχύτητα δημιουργίας των block είναι μεγαλύτερη γεγονός που συμβαίνει λόγω του ότι έχουμε 10 αντί για 5 miners, οπότε και το Proof Of Work βρίσκεται γρηγορότερα. Επίσης, όσο αυξάνεται το difficulty αυξάνεται και η δυσκολία να γίνει mine ένα νέο block και αυτό παρατηρείται ξεκάθαρα και στην γραφική μας. Τέλος, η αύξηση του block capacity, μειώνει και αυτή τον ρυθμό με τον οποίον προστίθενται block στην αλυσίδα καθώς απαιτείται μεγαλύτερος αριθμός transaction, για να ξεκινήσει το mining.



Παρατηρώντας τώρα την γραφική παράσταση του throughput μπορούμε να εξάγουμε και εδώ χρήσιμα συμπεράσματα. Αρχικά έχει ενδιαφέρον, ότι παρόλο όπως είδαμε πριν, με μεγαλύτερο capacity αργούν περισσότερο να μπουν block στην αλυσίδα, όμως τελικά διεκπεραιώνεται μεγαλύτερος αριθμός από transactions, κάτι που πιθανώς οφείλεται στο γεγονός ότι με capacity 1 χαλιέται περισσότερος χρόνος στο mining, παρά στην διεκπεραίωση transactions. Από την άλλη έχουμε αντίστοιχα αποτελέσματα όπως και παραπάνω, όσον αφορά στον αριθμό των participants και το difficulty.

Κάποια τελευταία σχόλια για τα πειράματά μας είναι τα εξής. Το γεγονός ότι όλοι οι κόμβοι στέλνουν μαζί τα transactions τους και όχι σε τυχαίους χρόνους, μας εμποδίζει στο να έχουμε απολύτως σωστά δεδομένα για το σύστημά μας. Επίσης τα transactions στέλνονταν μαζικά, χωρίς να περιμένουν κάποια επιβεβαίωση, γεγονός που θα μπορούσε να οδηγήσει σε προβλήματα, αν κάποιο transaction γίνει dropped.