Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ροή Υ- Εξάμηνο 9ο

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ 2021-2022

NOOBCASH

ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Μέλη ομάδας
Αθανασίου Ιωάννης 03117041
Μαντζούτας Ανδρέας 03117108

Σκοπός εργασίας

Σκοπός της εργασίας ήταν να χτίσουμε ένα σύστημα που θα ακολουθεί την αρχιτεκτονική blockchain, και το οποίο θα δίνει την δυνατότητα στους χρήστες του να πραγματοποιούν δοσοληψίες, τις οποίες θα αποθηκεύει, εξασφαλίζοντας το consensus με χρήση Proof-of-work.
Το σύστημά μας ονομάζεται Noobcash.

Εισαγωγή - Λίγα λόγια για το bitcoin

Το bitcoin είναι ένα κρυπτονόμισμα που δουλεύει χωρίς την ανάγκη ύπαρξης κάποιας κεντρικής αρχής (πχ τράπεζας). Στηρίζεται στην peer-to-peer επικοινωνία και στην κρυπτογραφία. Με τη χρήση του, εξασφαλίζεται ότι οι συναλλαγές είναι ασφαλείς και εχέμυθες και λύνονται προβλήματα όπως το double spending. Οι δοσοληψίες αποθηκεύονται στην δομή του Blockchain, την οποία και θα μιμηθούμε σε μία απλουστευμένη εκδοχή της.

Το σύστημά μας

Στα πλαίσια της εργασίας χτίσαμε 4 κύρια components.

1. Το πιο βασικό component όπου υλοποιούνται οι ζητούμενες προδιαγραφές
2. Ένα rest api, το οποίο χρησιμεύει για την επικοινωνία των nodes μεταξύ τους, αλλά και για την επικοινωνία του back-end με τα παρακάτω components
3. Ένα front-end
4. Ένα cli

Τεχνολογίες που χρησιμοποιήσαμε

- Για το back-end του συστήματος χρησιμοποιήσαμε την Python
- Για το rest api χρησιμοποιήσαμε Flask
- Το front-end γράφτηκε σε Javascript, με την βιβλιοθήκη Reactjs
- Γράψαμε το cli σε Python, χρησιμοποιώντας το pip package Click

Σχεδιασμός Συστήματος - συνοπτική παρουσίαση βασικών components

- Σε γενικές γραμμές, στηριχθήκαμε στο αρχικό template που μας δώθηκε.
- Υλοποιήσαμε όλες τις ζητούμενες λειτουργικότητες.
- Πιο συγκεκριμένα, **το back-end του συστήματος** αποτελείται από τα εξής βασικά συστατικά στοιχεία-κλάσεις:

    - Block
        - Τα blocks αποτελούν τις συστατικές μονάδες του blockchain
        - Ουσιαστικά είναι δομές δεδομένων, όπου αποθηκεύονται οι πληροφορίες για τα transactions.
        - Κάθε transcation μπαίνει αρχικά σε κάποιο block, μέχρι να γίνει validated.
        - Όταν τα transactions ενός μπλοκ φτάσουν την μέγιστη χωρητικότητά του (θεωρήσαμε ότι τα blocks έχουν μία σταθερή χωρητικότητα), τότε ξεκινάει η διαδικασία του mining και το block που επικυρώνεται πρώτο προστίθεται στο blockchain.

    - Miners
        - Η κλάση αυτή δεν δινόταν στο template και την προσθέσαμε εμείς.
        - Όπως εξηγούμε και σε σχόλια στον πηγαίο κώδικα, η επιλογή αυτή έγινε ώστε να είναι πιο εύκολη η ασύγχρονη εκτέλεση του mining, και να είναι πιο decoupled αυτή η λειτουργικότητα.
        - Με αυτήν την επιλογή, ταυτόχρονα, μας δίνεται η δυνατότητα να χειριστούμε πιο εύκολα την αποστολή σημάτων που χρειάζονται για να ρυθμίσουν από το εξωτερικό την κατάσταση του miner (όταν για παράδειγμα θέλουμε να διακόψει την διαδικασία του mining)
        - Τέλος, όταν ο miner βρει το κατάλληλο nonce και επιβεβαιώσει το block, το στέλνει στο api μέσω του endpoint /mined και έπειτα προστίθεται το block στο blockchain και γίνεται broadcast.

    - Το Blockchain, επομένως, είναι μία αλυσίδα από blocks
        - Μπορεί να προκύψουν διακλαδώσεις, αλλά επιλύονται με τον αλγόριθμο consensus, με το κριτήριο στο σύστημά μας να είναι το μήκος της διακλάδωσης)
        - Στο blockchain είναι αποθηκευμένα μόνο τα blocks που έχουν όλα τα transactions τους validated.
        - Μία αξιοσημείωτη προσθήκη που κάναμε στην κλάση του Blockchain, είναι η χρήση του **checkpoint.**

        - Το checkpoint, είναι στην ουσία το index του τελευταίου block που εισήλθε στο blockchain και στο οποίο έχουν συμφωνήσει όλοι οι nodes.
        - Η χρήση του αποτελεί μία εξαιρετικά σημαντική βελτιστοποίηση, αφού γλιτώνει το σύστημα από το broadcast ολόκληρου του chain κάθε φορά.

    - Transaction:
        - Σε κάθε transaction αποθηκεύονται τα βασικά στοιχεία του αποστολέα του ποσού, του παραλήπτη, το ποσό που μεταφέρεται και inputs (utxos) και outputs.
        - Για να μπορούμε να εγγυηθούμε ότι το transaction δεν προέρχεται από κάποιον κακόβουλο χρήστη, ο οποίος προσποιείται μία ψεύτικη ταυτότητα, χρησιμοποιούμε το ζεύγος private-public key του αποστολέα, για να υπογράψουμε και να ελέγξουμε την υπογραφή του transaction αντίστοιχα.

    - Node: Η κλάση node αποτελεί την εκτενέστερη κλάση της υλοποίησής μας
        - Κάθε αντικείμενο της κλάσης αντιστοιχεί σε έναν χρήστη, και περιέχει σημαντικά περισσότερα πεδία από τα αντικείμενα των κλάσεων που έχουμε δει έως τώρα.
        - Αυτά τα πεδία έχουν να κάνουν με πληροφορίες που αφορούν τον ίδιο τον χρήστη (πχ id, wallet, ring_node), την εικόνα που έχει ο χρήστης για τους υπόλοιπους κόμβους (chain), καθώς και το mining.
        - Για τις ίδιες λειτουργικότητες έχουμε κατασκευάσει και ειδικές μεθόδους, (για την κατασκευή νέων συναλλαγών, το validation υπάρχοντων συναλλαγών, την επέκταση του chain και την επίλυση conflicts πάνω του).
    - RingNodeList:
        - Πρόκειται για ακόμα μία κλάση που προσθέσαμε εμείς. Περιέχει μία λίστα από RingNodes, κάθε ένας από τους οποίους αποτελεί έναν κόμβο του δικτύου. Αποθηκεύουμε σε αυτό το id, τη διεύθυνση, το public key και το balance του κάθε κόμβου.
        - Το χρησιμοποιούμε ώστε να εκτελούμε πιο εύκολα όσες λειτουργίες σχετίζονται με το δίκτυο, όπως για παράδειγμα η εύρεση μιας διεύθυνσης.

- Txo:
  - Το Txo είναι μία κλάση που χρησιμοποιούμε για τα outputs των Transactions. Περιέχει σαν πεδία το id του transactions από το οποίο προήλθε, το id του παραλήπτη και ποσό της συναλλαγής.
  - Τα χρησιμοποιούμε σαν transaction outputs και για να συμβολίζουμε τα unspent transactions (utxo).

- **Όσον αφορά το front-end**:
  - Το front-end χτίστηκε ως ανεξάρτητο React-app από το back-end
  - Τα βασικά εξωτερικά πακέτα που χρησιμοποιήσαμε είναι τα πακέτα axios, material-ui, react-router
  - Το front-end αποτελεί στην ουσία ένα dashboard, όπου ο χρήστης μπορεί:
    - Να δει διαγράμματα με στατιστικά στοιχεία για τις εισερχόμενες και εξερχόμενες δοσοληψίες
    - Να δει αναλυτικές λίστες σχετικά με τις δοσοληψίες που τον αφορούν (διαφορετικές όψεις για τις εισερχόμενες και τις εξερχόμενες δοσοληψίες)
    - Να δει το υπόλοιπό του
    - Να δει τα βασικά στοιχεία του (public key, id κόμβου)
    - Να πραγματοποιήσει μία νέα συναλλαγή προς κάποιον άλλον κόμβο του συστήματος

- **Όσον αφορά το cli**:
  - Χτίστηκε, ομοίως με το front-end, ως ένα ανεξάρτητο πρόγραμμα που επικοινωνεί με το back-end μέσω του rest-api που προσφέρει το τελευταίο
  - Ουσιαστικά παρέχει τις ίδιες λειτουργικότητες με το front-end, αν εξαιρέσουμε τα στατιστικά στοιχεία με την μορφή διαγραμμάτων που απουσιάζουν για ευνόητους λόγους

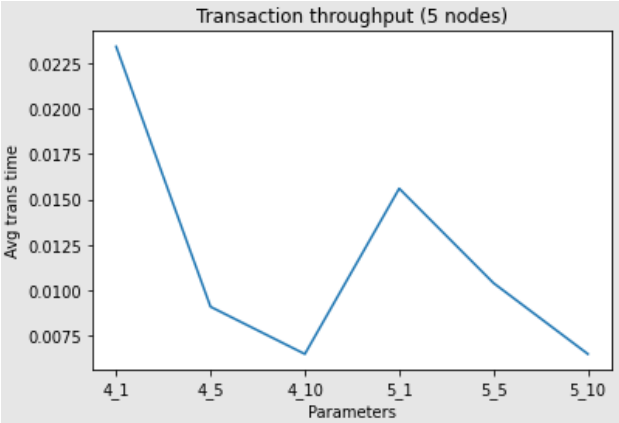
Πειράματα

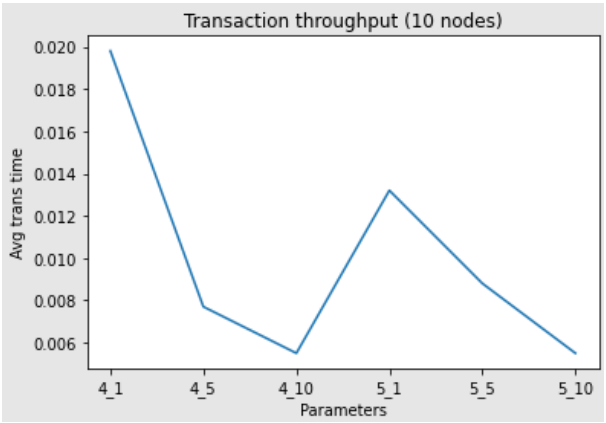
Έπειτα, προχωρήσαμε σε εκτέλεση των ζητούμενων πειραμάτων τοπικά, λόγω της αδυναμίας να φτιάξουμε εικονικές μηχανές στον Ωκεανό. Αυτό έχει ως αποτέλεσμα να μην είναι καλό μέτρο σύγκρισης τα αποτελέσματα σε σχέση με αντίστοιχες εργασίες, ωστόσο μας δείχνει μία σαφή εικόνα όσον αφορά το scalability.

Σαν γενική παρατήρηση, μπορούμε να πούμε ότι με σταθερά difficulty και #nodes, όσο αυξάνεται το Capacity των Blocks, αυξάνεται και η ρυθμαπόδοση του συστήματος. Αυτό έχει να κάνει με τον φόρτο στο δίκτυο. Όσο μικρότερο Capacity έχει το Block, τόσο πιο γρήγορα γίνεται mine ένα καινούριο, με αποτέλεσμα να υπάρχει ανάγκη είτε για αλγόριθμο consensus, είτε απλά για προσθήκη του block στο Blockchain, γεγονότα που καθυστερούν την εξυπηρέτηση των Transactions. Αξίζει να σημειωθεί ότι αν και η αύξηση των κόμβων οδηγούν σε μεγαλύτερο φόρτο αιτημάτων, αυτό δε σημαίνει και μείωση της απόδοσης, καθώς εξυπηρετούνται περισσότερα αιτήματα ανά μονάδα χρόνου.

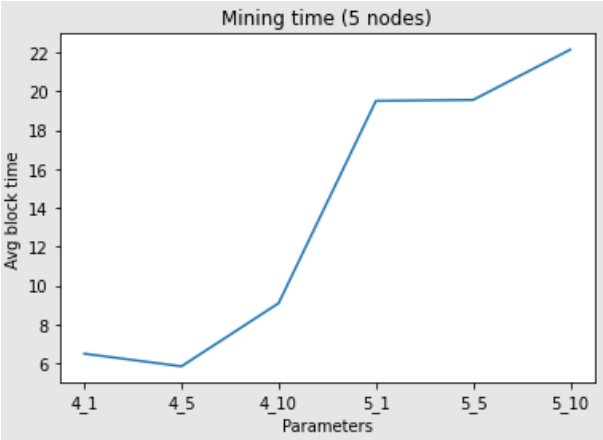
Τέλος, ο χρόνος για να γίνει mine ένα block εξαρτάται καθαρά από τον αριθμό των κόμβων, καθώς όσοι περισσότεροι miners, τόσο το καλύτερο, και από το minining difficulty. Αυτό εξηγεί το γεγονός ότι οι χρόνοι είναι παραπλήσιοι ανεξάρτητα του Capacity.

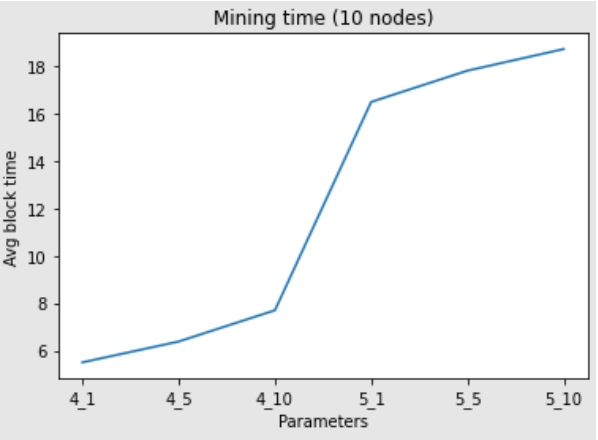
Ρυθμαπόδοση Συστήματος





Block mining time





Στις παραπάνω γραφικές, οι τιμές 4_1, 4_5 κ.ο.κ. αναφέρονται σε mininig diffulty 4 και block capacity 1,5. Σε γενικές γραμμές παρατηρούμε ότι τα αποτελέσματα είναι πολύ κοντά στα αναμενόμενα. Στον χρόνο για mine ενός Block, οι χρόνοι είναι σχεδόν σταθεροί για ίδιο difficulti, ενώ αυξάνονται όταν μεγαλώνει. Επίσης, στην περίπτωση των 10 κόμβων οι χρόνοι είναι λίγο χαμηλότεροι. Στην περίπτωση της ρυθμαπόδοσης, οι χρόνοι είναι λίγο καλύτεροι στην περίπτωση των 10 κόμβων, ενώ όσο αυξάνεται το Block Capacity, μειώνεται ο χρόνος εξυπηρέτησης ενός transaction.