ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΚ. ΕΤΟΣ 2023-2024

ΟΜΑΔΑ 20

ΔΗΜΗΤΡΙΟΣ ΑΝΔΡΕΑΔΗΣ 03119403

ΕΛΕΥΘΕΡΙΑ ΑΡΚΑΔΟΠΟΥΛΟΥ 03119442

ΑΔΑΜ ΚΑΠΕΤΗΣ 03119034

Κατανεμημένα Συστήματα

Αναφορά Εξαμηνιαίας Εργασίας 2023-2024: BlockChat

Σχεδιασμός Συστήματος

Η υλοποίηση του συστήματος έγινε σε Python, με χρήση REST API για την επικοινωνία μεταξύ κόμβων. Η υλοποίηση για κάθε στοιχείο του BlockChat (node, transaction, block, blockchain, wallet) γίνεται στο αντίστοιχο Python αρχείο. Για να αρχικοποιηθεί κάθε κόμβος σε διαφορετική θύρα και να μπει σε λειτουργία το σύστημα καλούμε την main -p <port>, και για τη διεπαφή του client για κάθε κόμβο καλούμε την client -p <port>.

I. Client

Δίνει στον χρήστη τη δυνατότητα για τις εξής εντολές:

- *t <recipient_id> <message>* : ο χρήστης στέλνει transaction στον κόμβο με id *recipient_id*. Θεωρείται coin transaction αν το *message* είναι αριθμός, αλλιώς θεωρείται message transaction.
- *balance* : τυπώνεται το υπόλοιπο του BC wallet του χρήστη.

- *view* : τυπώνεται το validator id και τα transactions του τελευταίου επικυρωμένου block του blockchain.
- *stake <amount>* : αποδεσμεύει το προηγούμενο stake του κόμβου και καταχωρεί το *amount* ως καινούριο stake του. Αποτυγχάνει αν δεν επαρκούν τα BCC στο wallet του χρήστη για να οριστεί το νέο stake.
- *help* : προβάλει την παραπάνω λίστα με διαθέσιμες εντολές.

II. Δομές

Οι διαφορετικές δομές του συστήματος:

- *ring*: ο κάθε κόμβος διατηρεί ένα ring, στο οποίο περιέχονται όλες οι πληροφορίες που χρειάζεται για τους υπόλοιπους κόμβους. Η δομή του έχει ως εξής:

  ring = {wallet.address: [id, ip, port, BCC, stake]}

  Το ring ανανεώνει τα υπόλοιπα όλων των κόμβων κάθε φορά που γίνεται μια συναλλαγή και τα stake κάθε κόμβου αν αυτός αλλάξει το stake του μέσω της εντολής *stake <amount>* του αντίστοιχου client.

- *node*: ο κάθε κόμβος του συστήματος ισοδυναμεί με ένα node. Στη δομή του κόμβου αποθηκεύονται οι σχετικές πληροφορίες του, καθώς και το wallet, το blockchain όπως έχει διαμορφωθεί μέχρι εκείνη τη στιγμή, και το τρέχον block του blockchain.

- *block*: στη δομή του κάθε block του συστήματος αποθηκεύονται οι σχετικές πληροφορίες του block. Τα fees από κάθε transaction που έχει προστεθεί μέσα στο block κρατούνται επίσης ως πληροφορία του block.

- *blockchain*: διατηρεί τη αλυσίδα από τα blocks που έχουν επικυρωθεί.

- *transaction*: στη δομή του transaction αποθηκεύονται οι σχετικές πληροφορίες του, όπως τα ids του αποστολέα και του παραλήπτη, τα fees που του αντιστοιχούν, το μήνυμα του, και ο τύπος του. Ορίζονται 3 τύποι transaction: coin (type 0), message (type 1), stake (type 2). Το κάθε transaction υπογράφεται με το private key του.

III. Λειτουργίες

Παρακάτω παρουσιάζονται οι λειτουργίες του συστήματος όπως αυτές προκύπτουν από τα διαφορετικά endpoints που έχουν υλοποιηθεί:

- */register_to_ring*: καλείται κατά την εισαγωγή κάποιου νέου κόμβου στο δίκτυο. Αρχικοποιεί τον νέο κόμβο και ενημερώνει το ring με τα νέα στοιχεία.

- */share_ring*: ενημερώνεται το ring των κόμβων για την προσθήκη νέων κόμβων σε αυτό.

- */send_transaction*: δημιουργεί ένα transaction από τον κόμβο του οποίου ο client έστειλε το transaction, το υπογράφει και το κάνει broadcast στο υπόλοιπο δίκτυο. Σε περίπτωση stake transaction, δημουργεί συναλλαγή stake.

- */validate_transaction*: οι υπόλοιποι κόμβοι του δικτύου λαμβάνουν το transaction από το broadcast, επικυρώνουν την υπογραφή του transaction με χρήση του public key του αποστολέα και ελέγχουν αν το υπόλοιπο του αποστολέα επαρκεί για τη συναλλαγή.

- */receive_transaction*: το πλέον επικυρωμένο transaction προστίθεται στο τρέχον block κάθε κόμβου και ενημερώνεται το ring. Στην περίπτωση stake transaction, απλά ενημερώνεται το ring. Αν το block με την προσθήκη του τρέχοντος transaction έφτασε σε capacity, επιστρέφεται status 205.

- */select_validator*: καλείται κατά τη λήψη status 205 από το /receive_transaction. Εκτελείται Proof-of-Stake με τα τρέχοντα stakes όλων των κόμβων του δικτύου και ο καθένας εκλέγει για τον ίδιο το validator του block σύμφωνα με αυτό. Χρησιμοποιώντας ως seed το hash του previous_block (ώστε να είναι ίδιο για όλους τους κόμβους αλλα διαφορετικό για κάθε επόμενο block που πρέπει να επαληθευτεί), μέσω τυχαίας γεννήτριας σε εύρος όσο και το συνολικό stake όλων των κόμβων, επιλέγεται ένα κατώφλι (threshold). Στη συνέχεια αντιστοιχίζεται η τιμή του κατωφλίου με έναν από τους κόμβους (αυτί γίνεται αθροίζοντας τα stakes των κόμβων με τη σειρά έως ότου ξεπεραστεί το κατώφλι) και ο κόμβος αυτός επιλέγεται ως validator.

- /validate_block: σε περίπτωση που ο κόμβος εκλέξει τον εαυτό του για validator, υπολογίζει το hash του block.

- */receive_block*: οι κόμβοι ελέγχουν αν έλαβαν από τον validator που εξέλεξαν το σωστό previous hash για το τρέχον block. Στη συνέχεια προσθέτουν το πλέον επικυρωμένο block στο blockchain.

- */balance*: όταν κληθεί από τη διεπαφή client κάποιου κόμβου τυπώνει το υπόλοιπο του wallet του.

- */view_block*: όταν κληθεί από τη διεπαφή client κάποιου κόμβου τυπώνει για το τελευταίο επικυρωμένο block του blockchain τα εξής στοιχεία: index του block, validator, capacity, λίστα των transactions που έχουν προστεθεί σε αυτό στη μορφή {sender id, receiver id, message} και τα συνολικά fees που προστέθηκαν στο wallet του validator του block.

Πειράματα και Αποτελέσματα

I. Απόδοση του συστήματος

Θεωρούμε ένα BlockChat με 5 clients, ο καθένας εκ των οποίων διαβάζει το αντίστοιχο αρχείο transX.txt. Εκτελούν και στέλνουν στο δίκτυο ταυτόχρονα τα transactions του αρχείου, έχοντας σταθερό staking 10 BCC. Τα αποτελέσματα του πειράματος:

- capacity 5:

  Block time: 0.3265 sec

  Total time: 30 sec

  Throughput: 16.67 transactions/sec

- capacity 10:

  Block time: 0.6078 sec

  Total time: 28 sec

  Throughput: 17.86 transactions/sec

- capacity 20:

  Block time: 1.16 sec

  Total time: 27.7 sec

  Throughput: 18.05 transactions/sec

II. Κλιμακωσιμότητα του συστήματος

Επαναλαμβάνουμε το πείραμα για 10 clients.

- capacity 5:

  Block time: 0.18 sec

  Total time: 35.5 sec

  Throughput: 28.17 transactions/sec

- capacity 10:

  Block time: 0.37 sec

  Total time: 35.9 sec

Throughput: 27.86 transactions/sec

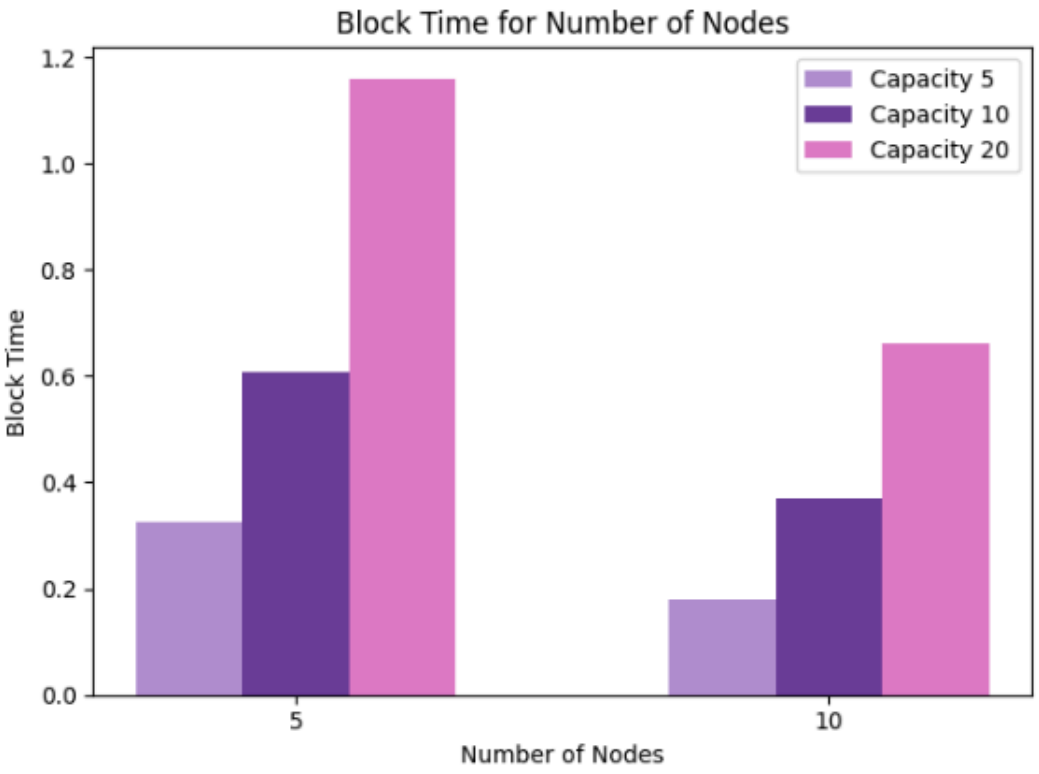
- capacity 20:

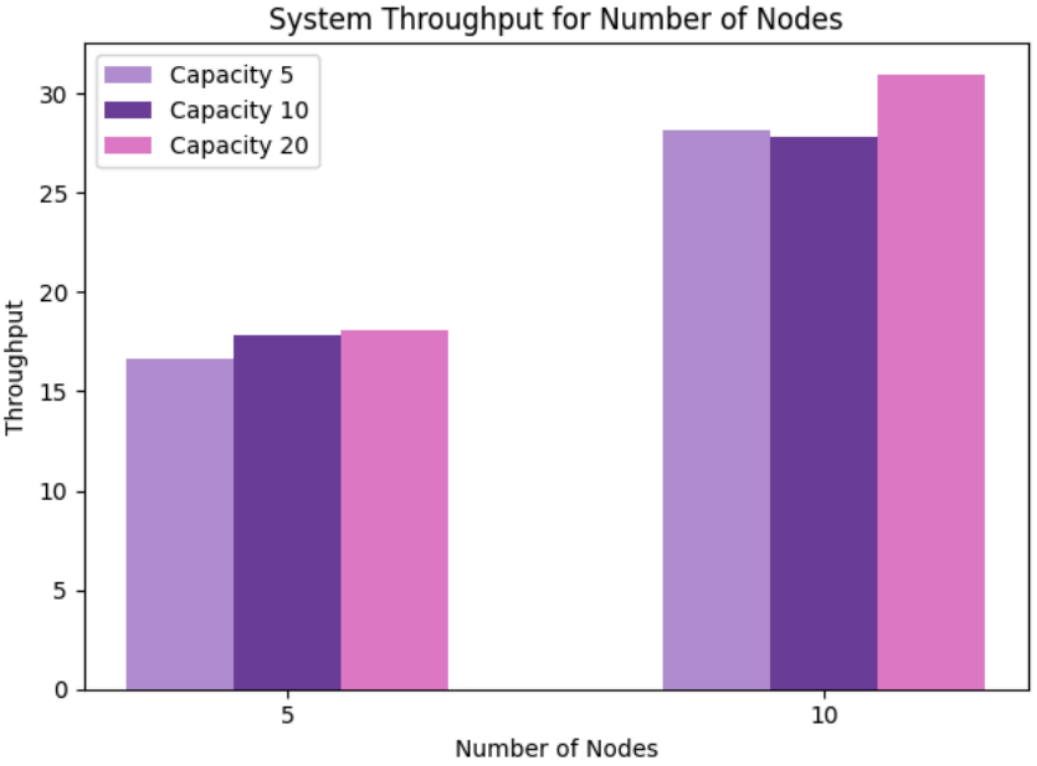
  Block time: 0.667 sec

  Total time: 32.28 sec

  Throughput: 30.98 transactions/sec

Τα γραφήματα σύγκρισης μετρικών για 5 και 10 nodes:





III. Δικαιοσύνη

Επαναλαμβάνουμε το πείραμα με 5 clients και capacity 5, με τον κόμβο 0 να έχει stake 100 BCC.

Παρατηρούμε πως αυτό έχει ως αποτέλεσμα σχεδόν πάντα να εκλέγεται ο κόμβος 0 ως validator, καθώς έχει κατά πολύ μεγαλύτερο stake και άρα πολύ περισσότερα “shares” για τη λοτταρία του Proof-of-Stake. Κατά συνέπεια, το πορτοφόλι του validator κόμβου, συλλέγοντας τα block fees από τα message transactions που εκτελούνται, έχει δυσανάλογα μεγαλύτερο υπόλοιπο σε σχέση με των υπόλοιπων κόμβων. Με αυτόν τον τρόπο, το σύστημα παύει να είναι δίκαιο για τους υπόλοιπους κόμβους.