

Κατανεμημένα Συστήματα

Αναφορά Εξαμηνιαίας Εργασίας

Αχιλλέας Τσιμιχόδημος
03119140

Νικόλαος Μπενέτος
03119028

Νικόλαος - Άγγελος Σκουμιός
03119091

Σχεδιασμός Συστήματος

Είσοδος κόμβου και αρχικοποίηση του συστήματος :

1. Όταν ένας κόμβος εισέρχεται στο σύστημα υπάρχουν 2 δυνατές επιλογές για τις δράσεις του:
 - Αν είναι ο bootstrap: κάνει το initialization του, δημιουργεί το genesis block και την πρώτη transaction.
 - Αν είναι ένας από τους υπόλοιπους κόμβους: «χτυπά» στο endpoint του bootstrap register_node, οπότε και θα προστεθεί στο ring.
2. Όταν όλοι οι κόμβοι εισαχθούν (ο bootstrap ελέγχει με την είσοδο του κάθε κόμβου), τότε θα σταλθούν το ring και το chain σε όλους τους κόμβους του συστήματος.
3. Σε αυτό το σημείο ο κάθε κόμβος θα κληθεί να διαβάσει το αντίστοιχο .txt αρχείο με τα transactions τα οποία θα πρέπει να διεκπεραιώσει.

Ring:

Κάθε κόμβος διατηρεί ένα dictionary που περιέχει όλες τις απαραίτητες πληροφορίες για τις συναλλαγές του με κάθε άλλο κόμβο του συστήματος : id, public_key, port, balance και stake.

Soft state και λίστα με transactions:

Κάθε κόμβος διατηρεί το δικό του soft_state και τη δική του λίστα με transactions, ανανεώνοντάς τα σε κάθε validate_transaction που εκτελεί. Επίσης, αν κατά την προσθήκη ενός transaction στη λίστα παρατηρήσει πώς το μήκος της έχει φτάσει το capacity, προχωρά στην εκτέλεση της mine_block.

Mine block:

Όταν έρθει η ώρα για κάποιον κόμβο να κάνει mine, εκτελεί την συνάρτηση mine_block, όπου αρχικά παίρνει το validator key και ελέγχει αν είναι ο validator. Αν δεν είναι ο validator, τότε συνεχίζει το validation των transactions που λαμβάνει μέχρι ο validator να κάνει validate το block. Αν είναι ο validator, τότε προσθέτει τα capacity πρώτα transactions από την λίστα του στο block, το κάνει broadcast και προχωρά στην validate block.

Broadcast block:

Όταν γίνεται broadcast ενός block από τον validator, τότε αυτός «χτυπά» στο endpoint receive_block του κάθε άλλου node του συστήματος. Σε αυτό το σημείο, όλοι οι υπόλοιποι nodes θα προχωρήσουν και εκείνοι στο validation του block.

Validate block:

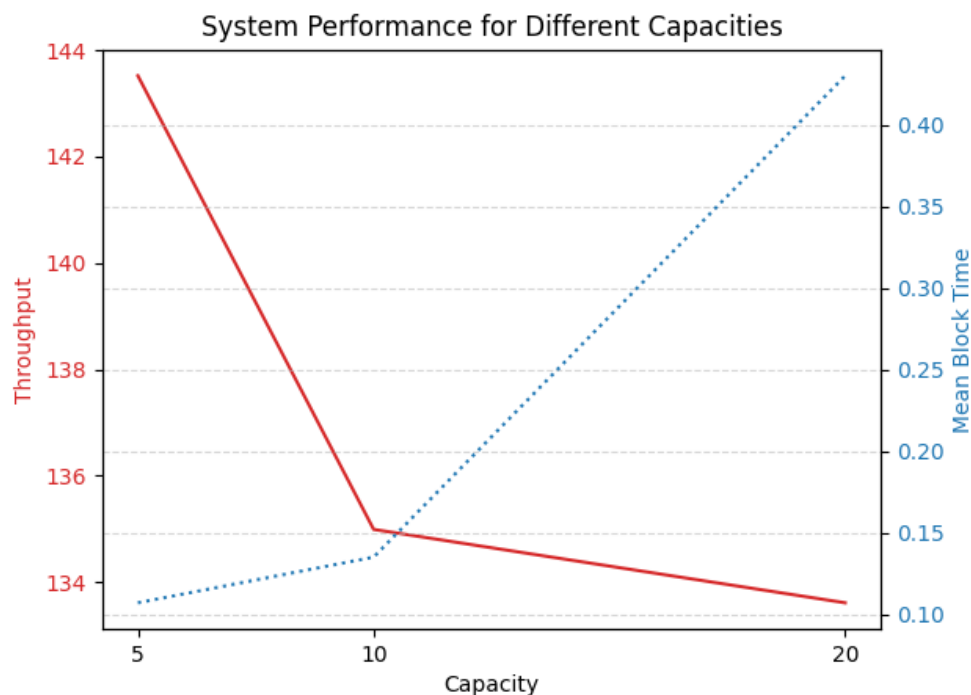
Κάθε κόμβος κάνει validate ενός block ελέγχοντας τα ακόλουθα:

- Αν ο validator του block είναι ο σωστός
- Αν το previous hash του block είναι όντως το current hash του τελευταίου block που έχει προστεθεί στην αλυσίδα.
- Για κάθε transaction μέσα στο block, ελέγχει αν αυτό περιέχεται ήδη στα transactions της λίστας του και αν ναι, το αφαιρεί από την λίστα του.
- Προχωρά στην εκτέλεση όλων των transaction του block βάση του (ενημέρωση ring) που έχει κρατήσει από την προηγούμενη προσθήκη ενός block ελέγχοντας αν μπορούν να γίνουν.
- Τέλος, δημιουργεί το καινούργιο block και το προσθέτει στο chain του.

Αποτελέσματα των Πειραμάτων

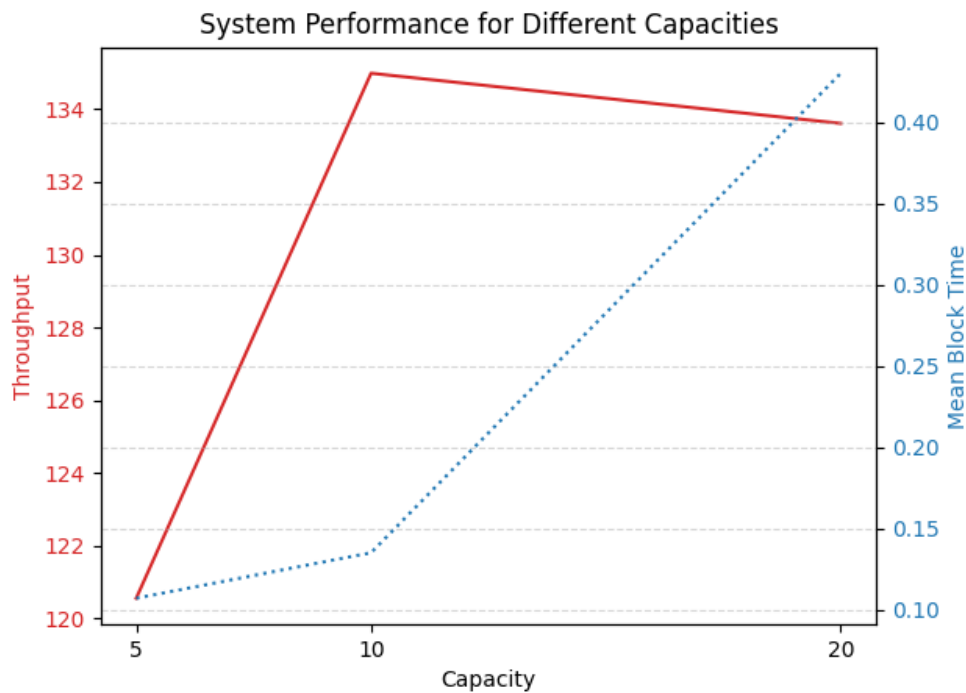
Απόδοση του συστήματος :

Με σταθερό staking 10 BCC και 5 clients



Κλιμακωσιμότητα του συστήματος :

Επανάληψη του πειράματος για 10 clients.



Δικαιοσύνη :

Επανάληψη του πειράματος για 5 clients και capacity 5, με τον bootstrap να έχει staking 100 BCC.

```
Node 0 balance: 1089.0  
Node 1 balance: 1589.0  
Node 2 balance: 516.0  
Node 3 balance: 782.0  
Node 4 balance: 1024.0
```

Same stake

```
Node 0 balance: 2379.0  
Node 1 balance: 705.0  
Node 2 balance: 695.0  
Node 3 balance: 655.0  
Node 4 balance: 566.0
```

Unfair stake

Παρατηρούμε ότι συσσωρεύονται περισσότερα BCC στον bootstrap, καθώς στατιστικά επιλέγεται συχνότερα ως validator, οπότε και λαμβάνει τα fees των συναλλαγών.