Εθνικό Μετσόβιο Πολυτεχνείο

Κατανεμημένα Συστήματα

Θέμα εξαμηνιαίας εργασίας: Ανάπτυξη κρυπτονομίσματος **Noobcash**

Ομάδα (αλφαβητικά):

Κελέσης Δημήτριος A.M.: 03115037

Κοντογιάννης Ανδρέας A.M.: 03115187

Πέππας Παναγιώτης A.M.: 03115146

Εισαγωγή

Στη σημερινή εποχή ολοένα και περισσότερες συναλλαγές τείνουν να γίνονται ηλεκτρονικά. Σε αυτό έχει συμβάλει τόσο η ανάπτυξη της τεχνολογίας όσο και η διάδοση του διαδικτύου ως μέσου επαφής αποκτώντας αναπόσπαστη θέση στην καθημερινότητα μας. Δειλά το 2008 εμφανίστηκε το πρώτο κρυπτονόμισμα (Bitcoin) που έμελλε να αλλάξει τον τρόπο που γίνονται οι συναλλαγές. Το νόμισμα αυτό προσέφερε στους χρήστες του ανωνυμία και ασφάλεια των συναλλαγών τους. Ταυτόχρονα τους απάλλασσε από την ανάγκη ύπαρξης μιας κεντρικής αρχής (τράπεζα, κράτος κλπ.) που θα οργάνωνε και θα διοικούσε το οικονομικό σύστημα. Με τη χρήση του νομίσματος αυτού μπορούσε πλέον ο καθένας να κάνει όσες συναλλαγές ήθελε με όποιον ήθελε στον κόσμο και αυτές να είναι ασφαλείς, εχέμυθες και να μην υπάρχει ανησυχία για τα προβλήματα του double spending ή της πιθανής εισόδου κάποιου κακόβουλου χρήστη που θα οδηγούσε σε κατάρρευση το σύστημα. Για να επιτευχθεί κατάρρευση του συστήματος απαιτείται περισσότεροι από τους μισούς χρήστες να δράσουν όλοι μαζί συνεργατικά για να καταφέρουν να αλλάξουν την πορεία των συναλλαγών και να ξεγελάσουν το σύστημα, κάτι που είναι απίθανο να συμβεί. Επίσης για να διατηρηθεί η αξία του νομίσματος εξασφαλίστηκε ανά τακτά χρονικά διαστήματα να μειώνεται ο ρυθμός δημιουργίας του γεγονός που σημαίνει αύξηση του κόστους παραγωγής του. Από τη δημιουργία του Bitcoin και μετά εμφανίστηκαν πολλά και διαφορετικά κρυπτονομίσματα δείχνοντας την τάση της αγοράς προς μία εναλλακτική μορφή οικονομικών συναλλαγών. Η επιβεβαίωση ήρθε το 2018 όταν το Bitcoin έφτασε στα ιστορικά υψηλότερα του αγγίζοντας τα 20.000 δολάρια.
Με βάση όλα τα παραπάνω κρίνεται σημαντική η ενασχόληση μας με το συγκεκριμένο σύστημα και η προσπάθεια να κατασκευάσουμε μία πολύ απλούστερη έκδοση του που θα πραγματοποιεί τις απολύτως απαραίτητες λειτουργίας δείχνοντας ωστόσο μέρος της ουσίας που κρύβεται από πίσω. Ακολουθώντας τις οδηγίες που μας υποδείχθηκαν υλοποιήσαμε τις συναρτήσεις και εν τέλει το σύστημα του Noobcash που σε πολύ αδρά σημεία μοιάζει με το Bitcoin.

Το Σύστημα

Για την υλοποίηση του συστήματος ακολουθήσαμε μία δενδροειδή μορφή. Δηλαδή μπορεί κανείς να διακρίνει ως βασικά κομμάτια του συστήματος μας τα εξής: Block, Blockchain, Transaction, Node, Rest, Cli. Θα αναλύσουμε καθένα από τα οποία στη συνέχεια.

Block: Αποτελεί τη βασική μονάδα ενώς blockchain και περιέχει όλες εκείνες τις πληροφορίες που θα πρέπει να αποθηκευτούν τελικά στην αλυσίδα. Είναι σημαντικά καθώς αφού αποθηκευτούν οι απαραίτητες πληροφορίες σε αυτά θα πρέπει να ελεγχθούν για την εγκυρότητα τους και στη συνέχεια να αποθηκευτούν στην αλυσίδα αν ικανοποιούν τις συνθήκες σχετικά με την συνέπεια του hash τους. Στην υλοποίηση μας ένα block αποτελείται από το index του που είναι ο αύξοντας αριθμός του εκάστοτε block, σημειώνεται ότι το αρχικό block Genesis έχει αύξοντα αριθμό μηδέν, και οι αριθμοί αυτοί αυξάνονται διαδοχικά. Επίσης το block περιέχει τις συναλλαγές που επιθυμούμε να αποθηκεύσουμε σε αυτό, τον χρόνο δημιουργίας του, το hash του προηγούμενου block στην αλυσίδα καθώς και κάποιους χαρακτήρες nonce που προκύπτουν τυχαία ώστε να ικανοποιηθεί τελικά μία συνθήκη σχετικά με την τιμή του current hash του block, δηλαδή αφού ολοκληρωθεί το block και προστεθούν σε αυτό οι τυχαιές τιμές του nonce όταν hash-αριστεί να έχει τα πρώτα MINING DIFFICULTY ψηφία του ίσα με μηδέν. Έχοντας αυτή τη γενική μορφή για το block οι συναρτήσεις που προσθέσαμε ήταν για να μπορούμε να το διαβάσουμε, δηλαδή προσθέσαμε μία συνάρτηση που μετατρέπει

το block σε μορφή json και αυτό μας ήταν πολύ χρήσιμο για την μετακίνηση του block από κόβμο σε κόμβο. Παράλληλα δημιουργήσαμε μία συνάρτηση για να μπορέσουμε να παράγουμε το hash του block. Σε αυτήν λαμβάνοντας υπόψη όλα τα στοιχεία του block, εκτός από το current hash του το οποίο και ζητάμε να υπολογίσουμε, και αφού τα μετατρέψαμε σε μορφή json υπολογίσαμε το ζητούμενο hash. Στη συνέχεια υλοποιήθηκε απλή συνάρτηση προσθήκης συναλλαγών στο block με εισαγωγή αυτών στην λίστα συναλλαγών του τρέχοντος block. Τέλος η σημαντικότερες συναρτήσεις στο τμήμα αυτό αφορούν το mine. Για την υλοποίηση τους αρχικά υλοποιήσαμε μία συνάρτηση αλήθειας που παράγει το hash του block, από αυτό εξάγει τα πρώτα MINING DIFFICULTY ψηφία και ελέγχει αν όλα αυτά είναι ίσα με το μηδέν επιστρέφοντας την αντίστοιχη αληθοτιμή. Η βασική συνάρτηση mining ξεκινά το nonce από τη μονάδα και το αυξάνει διαρκώς μέχρις ότου η προαναφερθείσα συνάρτηση αλήθειας να επιστρέψει θετικό αποτέλεσμα. Όταν γίνει αυτό ενημερώνεται τόσο το nonce όσο και το current hash του block με τις τελικές τους τιμές που είναι και οι πλέον έγκυρες.

Blockchain: Ανεβαίνοντας ένα επίπεδο αφαιρετικότητας πιο πάνω συναντάμε το Blockchain που όπως υποδηλώνει το όνομα του αποτελείται από ένα σύνολο από blocks. Τα κύρια μέρη της κλάσης αυτής είναι η λίστα από blocks, οι λίστα από συναλλαγές που θα μπουν στο block που θα δημιουργηθεί πριν αυτό προστεθεί στο chain και από ένα thread που εκτελεί την εργασία του mining. Συγκεκριμένα η κάθε αλυσίδα αντιστοιχεί σε έναν χρήστη αφού ο καθένας έχει την δικιά του αντίληψη για τους υπόλοιπους. Επίσης υπάρχει η συνάρτηση δημιουργίας της genesis αλυσίδας, δηλαδή αλυσίδα μήκους ένα με μόνο block το genesis block. Αυτή δημιουργείται για τον bootstrap κόμβο δίνοντας του αρχικά το απαραίτητο πλήθος χρημάτων για να τα μοιράσει στους υπόλοιπους. Υλοποιήθηκε η συνάρτηση προσθήκης συναλλαγής στην αλυσίδα η οποία για κάθε συναλλαγή στην είσοδο της ελέγχει αν με αυτή τη συναλλαγή το πλήθος των συναλλαγών που περιμένουν να μπουν στην αλυσίδα έφτασε το CHAIN CAPACITY, δηλαδή το μέγιστο αριθμό συναλλαγών που χωράνε σε κάθε block. Εάν έγινε αυτό τότε δημιουργεί ένα νέο block με αυτές τις συναλλαγές, κατάλληλο αύξοντα αριθμό και previous hash σύμφωνα με το τελευταίο block της αλυσίδας, καθαρίζει τον χώρο στον οποίο βρίσκονταν μέχρι πρότινος οι συναλλαγές και ξεκινά το mine thread ώστε να βρεθεί κατάλληλο nonce για το νέο block, παράλληλα θέτει την σημαία no mine σε κατάλληλη τιμή για να ενημερώσει όλους τους χρήστες ότι γίνεται mining. Η συνάρτηση που εκτελεί το thread που αναφέραμε πραγματοποιεί mining στο block και όταν αυτό ολοκληρωθεί αν μέχρι τότε δεν έχει γίνει ήδη mine από κάποιον άλλο χρήστη τότε σημαίνει ότι ο κόμβος αυτός έχει βρει πρώτος έγκυρο nonce και συνεπώς θα πρέπει να ενημερώσει όλους τους άλλους για το νέο block ώστε να το προσθέσουν όλοι στην αλυσίδα τους.

Transaction: Υλοποιήσαμε τις συναλλαγές ώστε σε αυτές να υπάρχουν οι βασικές πληροφορίες σχετικά με τους εμπλεκόμενους, το ποσό καθώς και τα inputs και τα outputs όπως αυτά ορίζονται σε ένα κρυπτονόμισμα, επίσης προστέθηκαν τα πεδία της υπογραφής καθώς και του μοναδικού αναγνωριστικού για κάθε συναλλαγή. Όπως και στο block έτσι κ εδώ υλοποιήσαμε συνάρτηση που μετατρέπει την κάθε συναλλαγή σε μορφή json ώστε να είναι δυνατή η προβολή της αλλά και η μεταφορά της. Παράλληλα για να βρούμε το id κάθε συναλλαγής χρησιμοποιήσαμε όπως αναφερόταν στην εκφώνηση τα υπόλοιπα στοιχεία της συναλλαγής τα οποία και hash-άραμε καταλλήλως ώστε να είναι μοναδικό το αναγνωριστικό της με μεγάλη πιθανότητα, σχεδόν μοναδιαία. Για να εξασφαλιστεί η μοναδικότητα όπως και σε όλες τις αντίστοιχες περιπτώσεις χρησιμοποιήσαμε την παράμετρο sorte keys ώστε ίδιες συναλλαγές να αποτυπώνονται σε ίδια json αρχεία ώστε όταν αυτές κωδικοποιηθούν να αντιστοιχί-

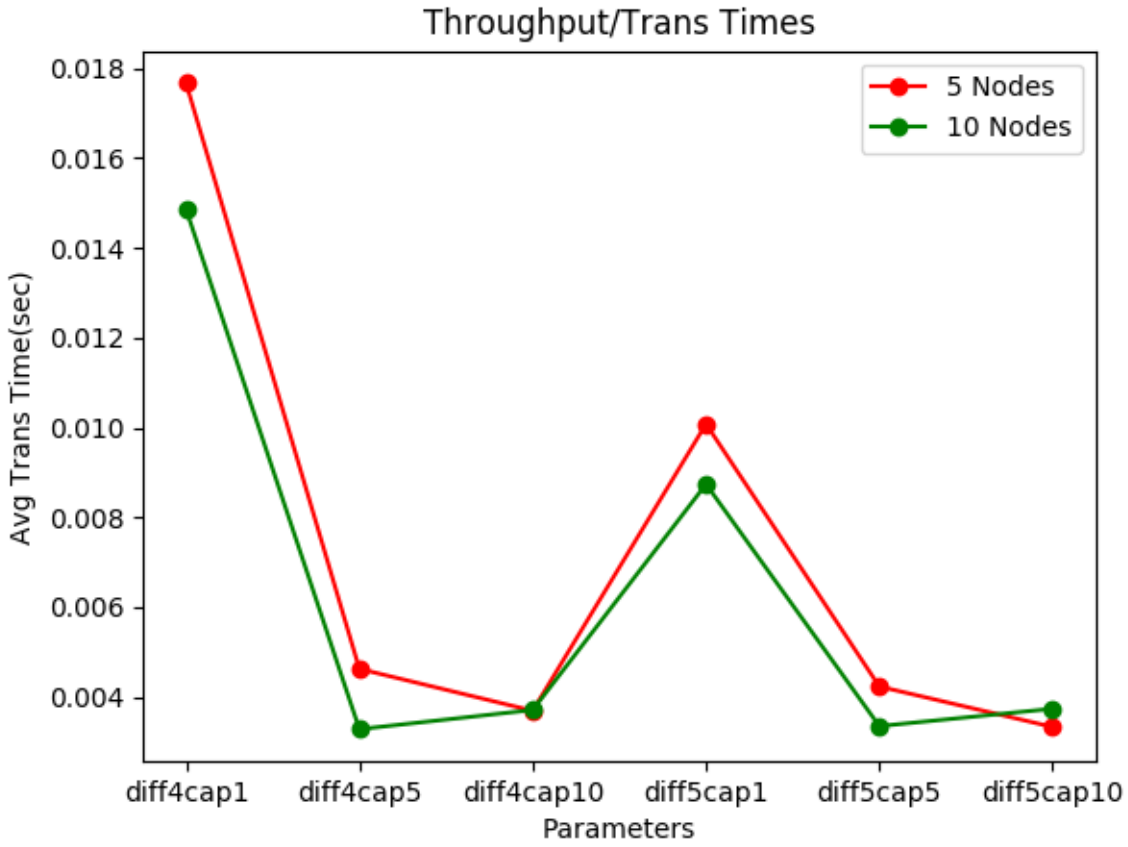
σουν στην ίδια τελική τιμή. Επίσης υλοποιήσαμε συνάρτηση υπογραφής για κάθε συναλλαγή ώστε αυτή να είναι έγκυρη και όποιος την λάβει να μπορεί να διαπιστώσει ότι υπεγράφη από τον νόμιμο δικαιούχο της και όχι από κάποιον τρίτο που ενδεχομένως δρα κακόβουλα. Για να το κάνουμε αυτό χρησιμοποιήσαμε RSA ζευγάρι αριθμών που αντιστοιχεί στο ιδιωτικό και δημόσιο κλειδί του κάθε χρήστη και με κατάλληλη συνάρτηση υπογράψαμε την συναλλαγή. Ταυτόχρονα σχηματίστηκε συνάρτηση επιβεβαίωσης της υπογραφής η οποία για μία συναλλαγή και έχοντας ως γνώση της μόνο το δημόσιο κλειδί μπορεί να επιβεβαιώσει αν η συναλλαγή υπεγράφη από τον νόμιμο δικαιούχο και να επιστρέψει κατάλληλη αληθοτιμή για αυτό.

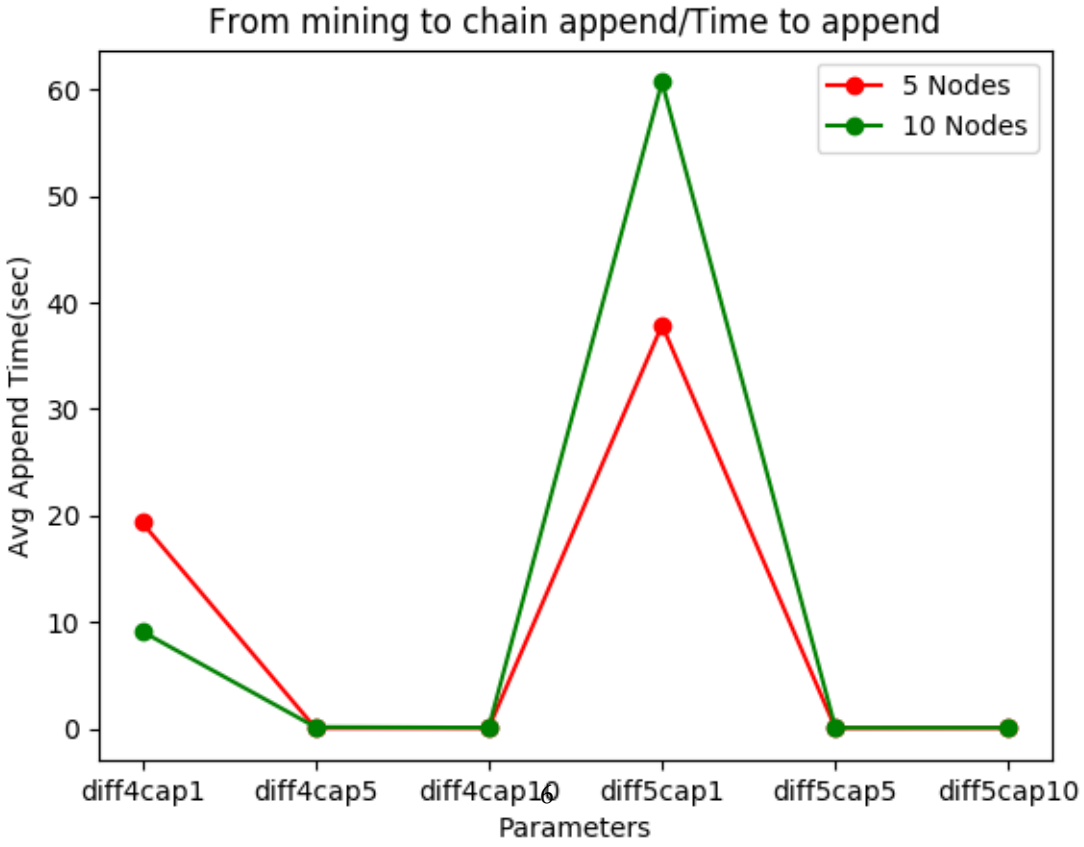
Node: Η κλάση αυτή αποτελεί το πυρήνα του συστήματος μας και αντιπροσωπεύει τον κάθε χρήστη καθώς και τις ενέργειες που θα ακολουθηθούν στην αρχή ώστε το σύστημα να λειτουργήσει ομαλά. Αρχικά κάθε χρήστης έχει στη διάθεση του μια πληθώρα πληροφοριών: την δικιά του αλυσίδα που αντιπροσωπεύει την μέχρι στιγμής αντίληψη του για τον κόσμο, το port που ακούει καθώς και την ip του, τις πλήρεις διευθύνσεις των υπολοίπων συμμετεχόντων καθώς και το πλήθος αυτών. Επίσης κατέχει μία λίστα που περιέχει τα unspent χρήματα του με τα οποία μπορεί να πραγματοποιήσει συναλλαγές, ένα λεξικό που περιέχει πληροφορίες για όλους τους χρήστες του συστήματος και τα χρήματα που αυτοί κατέχουν ώστε να μπορεί να ελέγξει αν οι συναλλαγές που λαμβάνει είναι έγκυρες ή όχι. Φυσικά, διαθέτει ένα πορτοφόλι που αποτελείται από το ιδιωτικό και το δημόσιο κλειδί του. Τέλος για να επιτευχθεί το ζητούμενο της άσκησης σχετικά με ταυτόχρονες συναλλαγές ο κάθε χρήστης έχει έναν buffer για τις συναλλαγές καθώς και ένα λεξικό που ενημερώνεται μόνο κατά τη διάρκεια του consensus σχετικά με τις αλυσίδες των υπολοίπων χρηστών. Η διαδικασία που ακολουθεί το σύστημα είναι η εξής:
Όλοι οι κόμβοι ξεκινούν το σύστημα και αναμένουν μέχρι να εισέλθουν όλοι σε αυτό. Ο κάθε κόμβος εισέρχεται είτε με αναγνωριστικό πατέρα (bootstrap) είτε με αναγνωριστικό παιδιού. Ο πατέρας αποκτά id ίσο με μηδέν και δημιουργεί για τον εαυτό του το genesis block. Ταυτόχρονα ενεργοποιεί ένα thread που αναμένει μέχρι να μπουν όλα τα παιδιά. Τα παιδιά με τη σειρά τους κατά την ενεργοποίηση τους αποστέλλουν μήνυμα στον πατέρα με τα στοιχεία τους (διεύθυνση και public key) και περιμένουν. Όταν ο πατέρας λάβει τόσα μηνύματα όσα και τα παιδιά έχει πλέον ενημερώσει ολόκληρο το ring με τα στοιχεία όλων των συμμετεχόντων και ξεκινά το thread που μέχρι αυτό το σημείο περίμενε. Κατά την εκκίνηση του ο πατέρας αναθέτει ids στα παιδιά σειριακά με τη σειρά που έλαβε από αυτά τα μηνύματα τους, ενημερώνει την αλυσίδα του και αφού σχηματίσει εγγραφές για αυτά στο λεξικό του που αφορά τον κόσμο στέλνει σε όλα μήνυμα σχετικά με το ring, την ταυτότητα τους, τα public keys όλων και το genesis block. Στη συνέχεια τους δίνει και από 100 NBC. Τέλος ενεργοποιεί τον buffer του αναμένοντας εισερχόμενες συναλλαγές. Τα παιδία λαμβάνοντας τα μηνύματα αυτά αρχικά δημιουργούν την αλυσίδα τους με το genesis block που έλαβαν και ενημερώνουν τα τοπικά τους στοιχεία για τον υπόλοιπο κόσμο, τις διευθύνσεις των συμμετεχόντων, τα public keys τους κλπ. Αφού έχουν γίνει αυτά τα παιδιά λαμβάνουν και τα πρώτα χρήματα τους και ενεργοποιούν με τη σειρά τους τον buffer και αυτά ώστε να μπορούν να δεχθούν συναλλαγές.
Έχοντας ξεκινήσει να στέλνονται συναλλαγές όλοι τις τοποθετούν στον buffer και σιγά σιγά τις κάνουν issue και ενημερώνουν τα αντίστοιχα πεδία. Σημειώνεται σε αυτό το σημείο ότι μόλις και το τελευταίο παιδί λάβει τα αρχικά χρήματα του τότε ξεκινά η επαναληπτική διαδικασία διαβάσματος συναλλαγών από τα αρχεία και η εκτέλεση αυτών.
Υλοποιήθηκε συνάρτηση δημιουργίας συναλλαγής η οποία αφού λάβει τα δεδομένα σχετικά με το ποσό αλλά και τον παραλήπτη ελέγχει αν τα χρήματα που πρόκειται αν δαπανηθούν

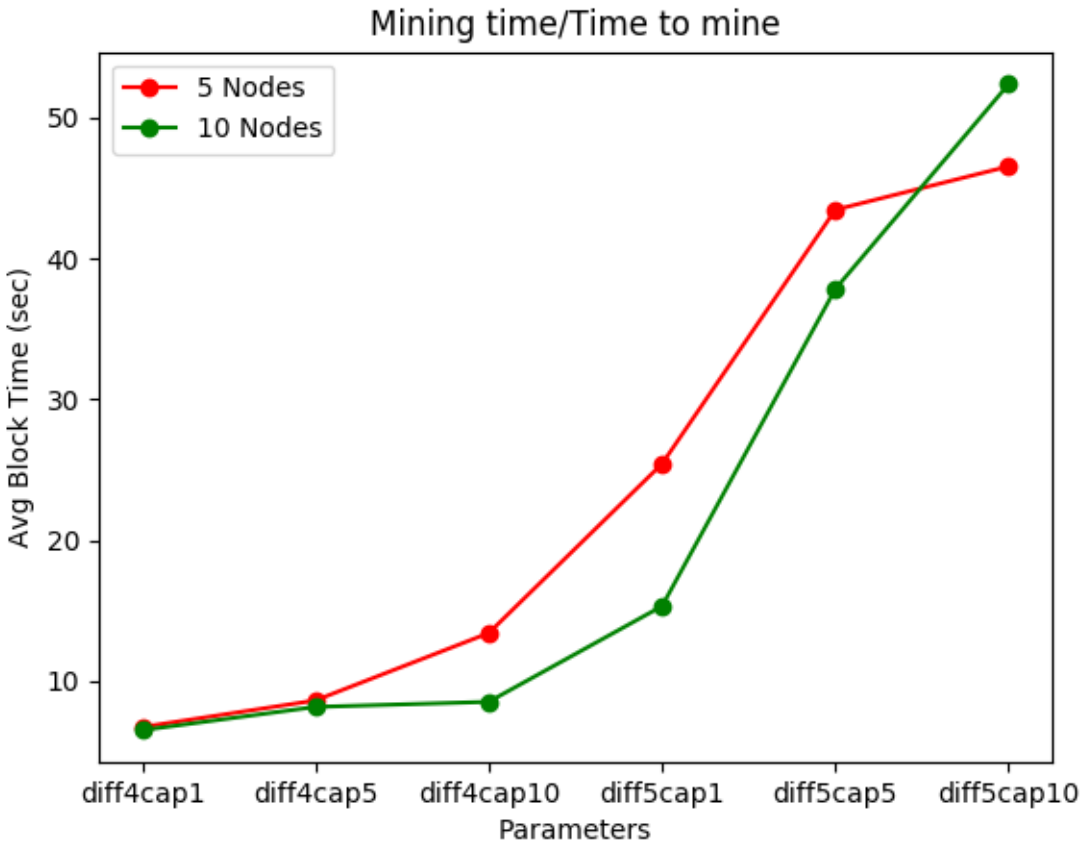
υπάρχουν σε unspent και αν ναι τότε χρησιμοποιεί αυτά τα unspent ώστε να σχηματιστούν να χρήματα που είναι να αποσταλούν αλλά και τα ρέστα στον παραλήπτη, αν υπάρχουν, αυτά αποθηκεύονται στη συναλλαγή στο πεδίο output. Κατόπιν δημιουργεί ένα νέο αντικείμενο transaction και στο οποίο εκχωρούνται όλες οι παραπάνω πληροφορίες και ενημερώνεται το λεξικό trans dict του αποστολέα σχετικά με τη νέα κατάσταση που πρόκειται αν λάβει ο κόσμος γύρω του, δηλαδή ότι κάποιος θα έχει περισσότερα χρήματα και ο ίδιος θα έχει λιγότερα. Έπειτα αφού υπογράψει τη συναλλαγή την κάνει broadcast σε όλους και την προσθέτει στην δικιά του έκδοση της αλυσίδας.
Όταν κάποιος λάβει μία συναλλαγή θα πρέπει για αρχή να επιβεβαιώσει την προέλευση της, δηλαδή να ελέγξει για την υπογραφή που υπάρχει πάνω σε αυτή και ότι όλα τα inputs της συναλλαγής δηλαδή τα unspent που αυτή χάλασε για να πραγματοποιηθεί ήταν όντως υπαρκτά και συμφωνούν με την άποψη που έχει ο παραλήπτης για τον κόσμο. Με άλλα λόγια αν ο παραλήπτης στην άποψη του για τον κόσμο θεωρεί ότι ο αποστολέας δεν είχε τόσο χρήματα όσα χάλασε τότε θεωρεί την συναλλαγή άκυρη.
Υλοποιήθηκε συνάρτηση που τοποθετεί όλες τις συναλλαγές που λαμβάνει στον buffer και στη συνέχεια επαναληπτικά αν δεν γίνεται mine, consensus και υπάρχει συναλλαγή στον buffer την κάνει issue/receive. Το παραπάνω γίνεται επαναληπτικά και σε αυτό το σημείο όταν λάβει τη πρώτη συναλλαγή του το τελευταίο παιδί ενεργοποιεί την επαναληπτική διαδικασία για εισαγωγή συναλλαγών από το αρχείο.
Για κάθε συναλλαγή που λαμβάνει ένας χρήστης ελέγχει αν αυτή είναι έγκυρη χρησιμοποιώντας την προαναφερθείσα συνάρτηση και αν είναι έγκυρη τότε την προσθέτει στην αλυσίδα του και ενημερώνει την κατάσταση που έχει ο ίδιος για τον κόσμο γύρω του, επίσης αν η συναλλαγή τον αφορά ενημερώνει τα unspent χρήματα του προσθέτοντας το έξτρα ποσό.
Κατά τη λήψη ενός block υλοποιήθηκε συνάρτηση ελέγχου και επιβεβαίωσης αυτού. Αν το previous hash του block που λάβαμε είναι ίδιο με αυτό του current hash του τελευταίου block στην αλυσίδα μας τότε είναι έγκυρο block και το προσθέτουμε στην αλυσίδα σταματώντας παράλληλα τυχόν mining. Σε αντίθετη περίπτωση στέλνουμε μήνυμα για consensus και ζητάμε ενημέρωση από όλους. Όλοι όσοι λάβουν μήνυμα στέλνουν στον αποστολέα την αλυσίδα τους και αυτός διαλέγει την μεγαλύτερη την οποία και κάνει δικιά του.

Rest & Cli: Το rest αποτελείται από ένα από API με τις αναγκαίες λειτουργηκότητες για το σύστημα μας και τα κατάλληλα μηνύματα και responses αναλόγως τον τρόπο που αυτό καλείται. Κάθε χρήστης ξεκινά ένα rest πρόγραμμα στη διεύθυνση που ακούει δίνοντας τα κατάλληλα ορίσματα σχετικά με τα παιδιά και τον πατέρα. Από εκεί και πέρα το σύστημα τρέχει αυτόματα μέσω του node.
Το cli αποτελείται από ένα command line tool με τις ζητούμενες λειτουργικότητες.

Πειραματικά Αποτελέσματα







Τα γραφήματα κατά σειρά παρουσιάζουν:
Το χρόνο για να εκτελεστεί μία συναλλαγή από τη στιγμή που θα βγει από τον **buffer** μέχρι να ολοκληρωθεί το **receive** της. Μετρημένο για όλες τις συναλλαγές και στη συνέχεις υπολογίζοντας τον μέσο χρόνο αυτών.
Μέσος χρόνος από τη στιγμή που έγινε το **mine** ενός **block** μέχρι αυτό να προστεθεί στην αλυσίδα, κάνοντας ή όχι **consensus**.
Μέσος χρόνος που απαιτείται για να γίνει **mine** ένα **block**.