

# Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

Κωτούλας Εμμανουήλ [emmakot@ece.auth.gr](mailto:emmakot@ece.auth.gr) 9697

Repo: [Emkot87/IntegratedSystems \(github.com\)](https://github.com/Emkot87/IntegratedSystems)

Results: [drive](#)

Στόχος της εργασίας ήταν να φτιάξουμε ένα σύστημα που λαμβάνει ασύγχρονα πληροφορία συναλλαγών για ένα σύνολο από σύμβολα. Στην υλοποίηση μου χρησιμοποιήθηκαν τα 4 σύμβολα που είχαν ήδη οριστεί (Amazon, Apple, BitcoinUsd, IcMarkets). Τα δεδομένα αποθηκεύονται στο φάκελο Results και για κάθε σύμβολο δημιουργείται ξεχωριστό directory.

Για να τρέξει το client αρκεί να γίνει compiled με την εντολή make και έπειτα executed με ./client.

Για να σταματήσει χρησιμοποιείται ctrl+z καθώς το ctrl+c δεν το βγάζει από την λούπα.

## Υλοποίηση

Χρησιμοποιείται ένας πίνακας με structs που αποθηκεύουν την πληροφορία για κάθε συναλλαγή. Ο πίνακας αυτός χρησιμοποιείται ώστε να κρατάμε τουλάχιστον τις συναλλαγές του τελευταίου λεπτού. Χρησιμοποιούμε έναν index το οποίο αυξάνουμε κάθε φορά που προσθέτουμε μια συναλλαγή. Το index κάνει wrap around.

Το σύστημα χρησιμοποιεί μια ουρά fifo στην οποία προστίθενται οι συναρτήσεις που πρέπει να κληθούν μαζί με τα ορίσματα τους. Producers είναι η συνάρτηση packetReciever η οποία καλείται για κάθε ένα από τα 4 μέρη της συναλλαγής και η calculateCandlesticks. Η packetReciever προσθέτει το στην ουρά την printInfoToFile με όρισμα το index της συναλλαγής από τον πίνακα με τις συναλλαγές. Η calculateCandlesticks εκτελεί τον κώδικα της μόλις αλλάξει το λεπτό, και προσθέτει στην fifo την calculateFifteen.

Η υλοποίηση της ουράς fifo και των consumer είναι μέρος της πρώτης εργασίας και δεν θα σχολιαστεί η υλοποίηση τους.

Αρχικά δημιουργούνται τα απαραίτητα directories, 1 thread το οποίο εκτελεί την calculateCandlesticks και 8 consumers οι οποίοι εκτελούν στοιχεία από την ουρά fifo. Έπειτα πραγματοποιείται η σύνδεση και λαμβάνεται πληροφορία. Έχει προστεθεί ένα while loop ώστε να ξανασυνδεθεί το σύστημα στην περίπτωση που έχουμε timeout.

## packetReciever

Η συνάρτηση packetReceiver καλείται κάθε φορά που λαμβάνουμε ένα τμήμα πληροφορίας της συναλλαγής. Χρησιμοποιεί ένα counter ώστε να αποθηκεύεται η πληροφορία στην κατάλληλη μεταβλητή. Όταν λάβουμε ολόκληρη την συναλλαγή τότε προσθέτει στην ουρά fifo την printToFile με όρισμα το index της συναλλαγής και την ώρα που έφτασε το πακέτο ώστε να υπολογιστεί η καθυστέρηση.

## initializeCandlestick

Η συνάρτηση αυτή κάνει initialize τα candlestick για ένα συγκεκριμένο index. Τα candlesticks αποθηκεύονται σε έναν πίνακα 4 X 15. Αυτό γίνεται ώστε να είναι συγκεντρωμένα τα candlesticks για κάθε ένα από τα σύμβολα και για να έχουμε πληροφορία για τα τελευταία 15 λεπτά. Ο index για το παρόν candlestick κάνει wrap around.

## printInfoToFile

Η συνάρτηση αυτή δέχεται σαν όρισμα το index της συναλλαγής ώστε να το τυπώσει και υπολογίζει και την καθυστέρηση από το χρόνο που της δίνεται μέχρι να το τυπώσει.

## calculateCandlesticks

Η συνάρτηση αυτή ορίζει κάποιες μεταβλητές και έπειτα περιμένει μέχρι την αλλαγή του λεπτού. Αποθηκεύει τον χρόνο και έπειτα διατρέχει όλες τις συναλλαγές που λήφθηκαν το τελευταίο λεπτό και κάνει για κάθε μια τους απαραίτητους υπολογισμούς. Προσθέτει στην ουρά την calculateFifteen με όρισμα το candlestickIndex και τον χρόνο που κράτησε ώστε να υπολογίσει η calculateFifteen την δική της καθυστέρηση. Τυπώνει το candlestick και υπολογίζει και τυπώνει την καθυστέρηση της. Τέλος υπολογίζει την μέση καθυστέρηση των συναλλαγών του τελευταίου λεπτού και την τυπώνει, μηδενίζει τις τιμές ώστε να υπολογίσουμε ξανά τις μέσες καθυστερήσεις των συναλλαγών και κάνει increment τον candlestickIndex.

## calculateFifteen

Η συνάρτηση αυτή υπολογίζει το συνολικό volume και το μέσο κόστος συναλλαγής για τα τελευταία 15 λεπτά. Τα πρώτα 15 λεπτά υπολογίζει από την αρχή μέχρι όποιο λεπτό έχουμε φτάσει και δεν περιμένει μέχρι τα 15 λεπτά ώστε να βγάλει αποτέλεσμα. Τυπώνει τα αποτελέσματα της και τέλος υπολογίζει τον χρόνο ώστε να τυπώσει την καθυστέρηση της.

## Σχόλια και παρατηρήσεις

Την λειτουργία του προγράμματος για μεγάλα χρονικά διαστήματα την καθιστούν δυνατή η στατική δέσμευση μνήμης, που καθιστά αδύνατο να έχουμε memory leaks και ο κώδικας που προστέθηκε στην ws\_service\_callback σε συνδιασμό με μια while loop στην main σε περίπτωση που υπάρξει timeout από πλευράς του API ώστε το πρόγραμμα να επανασυνδεθεί και να συνεχίσει να συλλέγει δεδομένα.

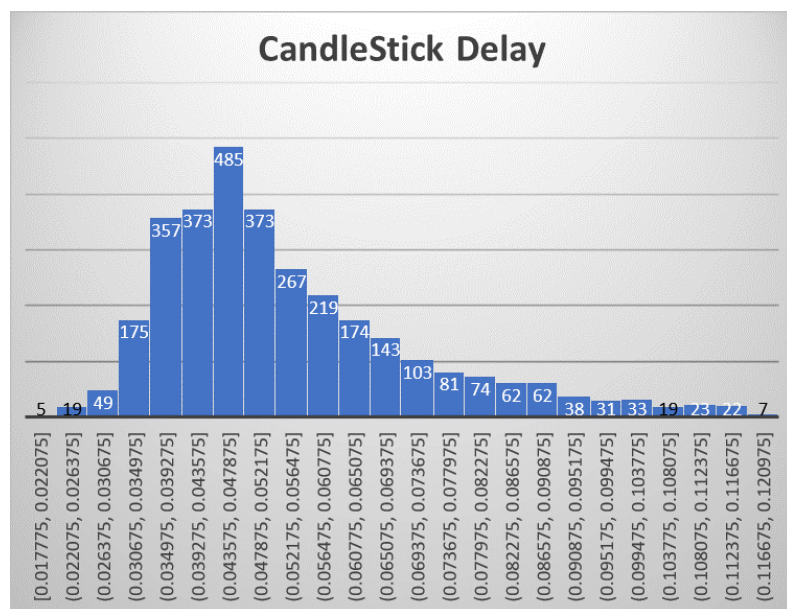
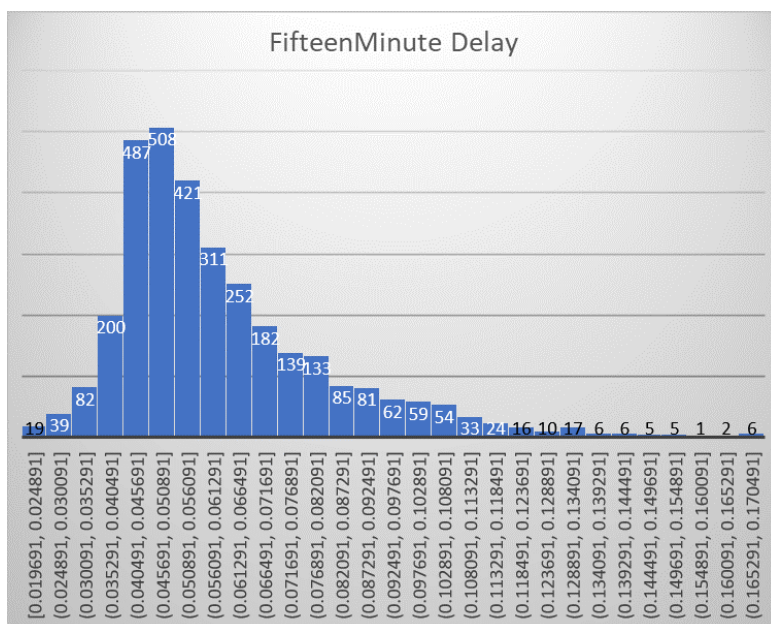
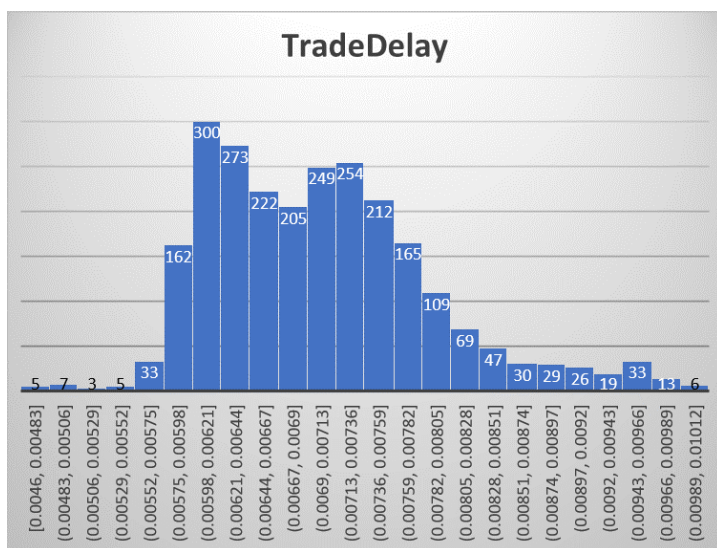
Επίσης η χρήση της ουράς fifo καθιστά δυνατή την παράλληλη εκτύπωση των δεδομένων ώστε το main thread να μπορεί να λαμβάνει συνεχώς δεδομένα. Η fifo χρησιμοποιείται και στην calculateCandles για να κληθεί η calculateFifteen επειδή θέλουμε πρώτα να έχει υπολογιστεί το candlestick, ωστόσο δεν θέλουμε να καθυστερήσει η εκτύπωση των candlestick ούτε να καθυστερήσει η calculateFifteen από την εκτύπωση των candlestick και έτσι τα κάνουμε παράλληλα.

Να σχολιαστεί ότι μετά από timeout ή έλλειψη σύνδεσης είναι πιθανό να μην ληφθούν δεδομένα για 1-5 λεπτά ωστόσο δεν γίνεται πάντα αυτό.

Στην ws\_service\_callback προστέθηκε στο case το LWS\_CALLBACK\_CLIENT\_CLOSED το οποίο καλείται όταν έχουμε timeout και βγάζει το πρόγραμμα από το while loop στην γραμμή 350.

Τέλος οι δοκιμές έγιναν στον προσωπικό μου υπολογιστή και όχι σε Rpi καθώς με τον parser που δίνονταν λάμβανα error -23 το οποίο δεν μπόρεσα να λύσω.

## Αποτελέσματα



Μέση καθυστέρηση για την εγγραφή των candlestick : 0.057978 second

Μέση καθυστέρηση για την εγγραφή των fifteenMinute τιμών : 0.063073 second

Μέση καθυστέρηση για την εγγραφή των trade : 0.011155 second

Για την συλλογή των δεδομένων το πρόγραμμα έτρεξε για 3279 λεπτά.

Δεν μετρήθηκε ο idle χρόνος της CPU επειδή το πρόγραμμα έτρεχε παράλληλα με άλλες δραστηριότητες που εκτελούσα στον υπολογιστή και καθώς οι συναλλαγές bitcoin έρχονταν με αρκετά γρήγορο ρυθμό όπου δεν υποθέτω ότι υπήρχε σημαντικός idle χρόνος.

Χρησιμοποιήθηκαν τα 4 σύμβολα που υπάρχουν και στο παράδειγμα με τις περισσότερες κατά μεγάλο βαθμό συναλλαγές να είναι αυτές για BitcoinUsd. Χωρίς να είναι σφάλμα του προγράμματος το API σταμάτησε να στέλνει δεδομένα για trades μετοχών της Apple, αν και σε παλαιότερες δοκιμές λειτουργούσε κανονικά. Τέλος τα IcMarkets δεν παρείχαν πληροφορίες για το volume οπότε οι 15λεπτες τιμές δεν έχουν κάποια πληροφορία.