

### Τρόπος υλοποίησης c κώδικα (παρόμοιος για τον javascript κώδικα)

Κάνουμε setup το websocket για να ακούμε όταν έρχονται κάποια μηνύματα. Όταν λάβουμε LWS\_CALLBACK\_CLIENT\_RECEIVE, ξέρουμε ότι λάβαμε ένα json μήνυμα που περιέχει την πληροφορία που μας ενδιαφέρει για τις μετοχές. Η writeToFile είναι υπεύθυνη να πάρει τις πληροφορίες από το json και να τις αποθηκεύσει σε ευανάγνωστη μορφή σε ένα αρχείο που έχει όνομα το symbol (την εταιρεία που ανήκει η μετοχή που πήραμε, που βρίσκεται στο key "s" του json που λάβαμε). Επίσης αποθηκεύουμε σε κατάλληλη δομή δεδομένων (deque) τα mean και volume των μετοχών για να τα χρησιμοποιήσουμε στο 3<sup>ο</sup> ερώτημα.

Στο πρόγραμμα κάνουμε spawn ένα ακόμη thread ο οποίος είναι υπεύθυνος να τρέχει κάθε 1 λεπτό μια εντολή που καλεί το candlestick αρχείο, το οποίο είναι το executable του candlestick.c. Το αρχείο αυτό κάνει ένα νέο request για να λάβει τα candlestick data της μετοχής που του δίνουμε στο όρισμα. Για παράδειγμα, μια εντολή που καλεί το αρχείο είναι της μορφής: ./candlestick AMZN 1572651390 1575243390

Το candlestick request γίνεται με βάση τα ορίσματα που καλείται το executable. Το AMZN στο συγκεκριμένο παράδειγμα είναι το σύμβολο της εταιρείας, το πρώτο timestamp 1572651390 είναι από ποια στιγμή και το δεύτερο timestamp είναι μέχρι ποια στιγμή θέλουμε τα candlestick data.

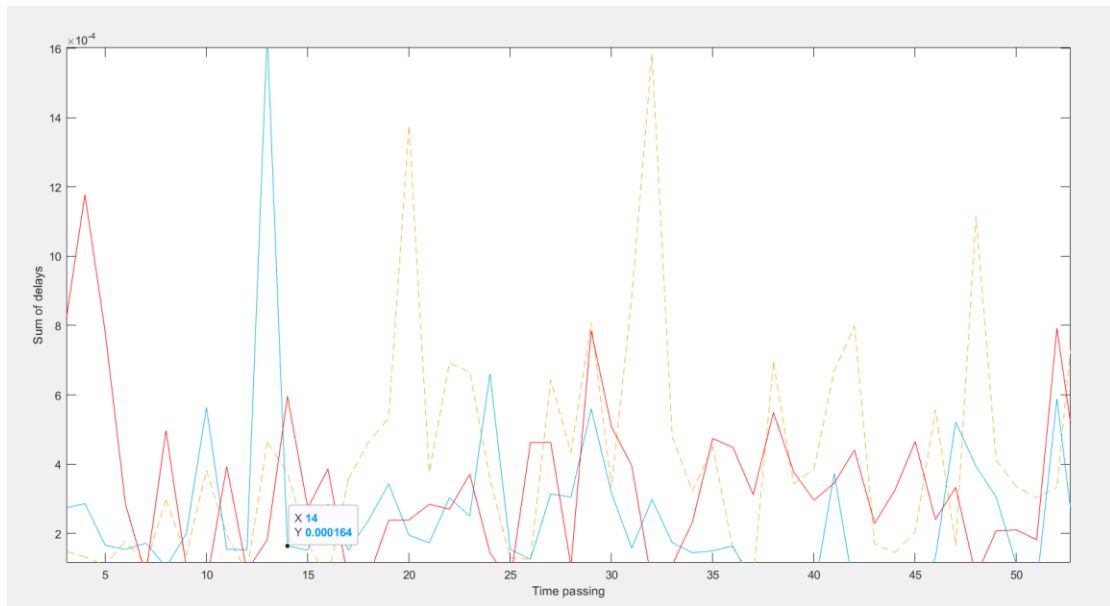
Κάθε ένα λεπτό καλούμε επίσης την

writeMovingMeansAndVolumesForLast15MinutesStockPricesForAllSymbols. Αυτή παίρνει από τα deque τον μέσο όρο των μετοχών και το άθροισμα των volumes και τα γράφει σε αρχεία που έχουν όνομα MOVING\_MEANS\_<SYMBOLNAME>, VOLUMES\_<SYMBOLNAME>. Για τις πληροφορίες αυτές (τον κινούμενο μέσο όρο και το άθροισμα των volume) χρησιμοποιούμε deque. Για τον μέσο όρο (αντίστοιχη υλοποίηση για τα volumes), έχουμε ένα deque ανά σύμβολο το οποίο έχει σε κάθε κελί του το μέσο όρο των stock values ανά λεπτό. Όταν λαμβάνουμε ένα stock value, προσθέτουμε στο μπροστά μέρος του deque το νέο value (χωρίς να αλλάξουμε κελί, δηλαδή το πιο μπροστινό κελί θα έχει το υπάρχον άθροισμα συν την τιμή της καινούργια μετοχής που λάβαμε). Όταν το μέγεθος του deque ξεπεράσει τα 15 (αυτό σημαίνει ότι έχει πληροφορίες για 15 λεπτά), αφαιρούμε το τελευταίο κελί του deque, το οποίο αντιστοιχεί στο πιο παλιό λεπτό. Έτσι επιτυγχάνουμε να έχουμε τις μέσες τιμές των μετοχών για 15 λεπτά κάθε φορά (αντίστοιχα για τα volumes).

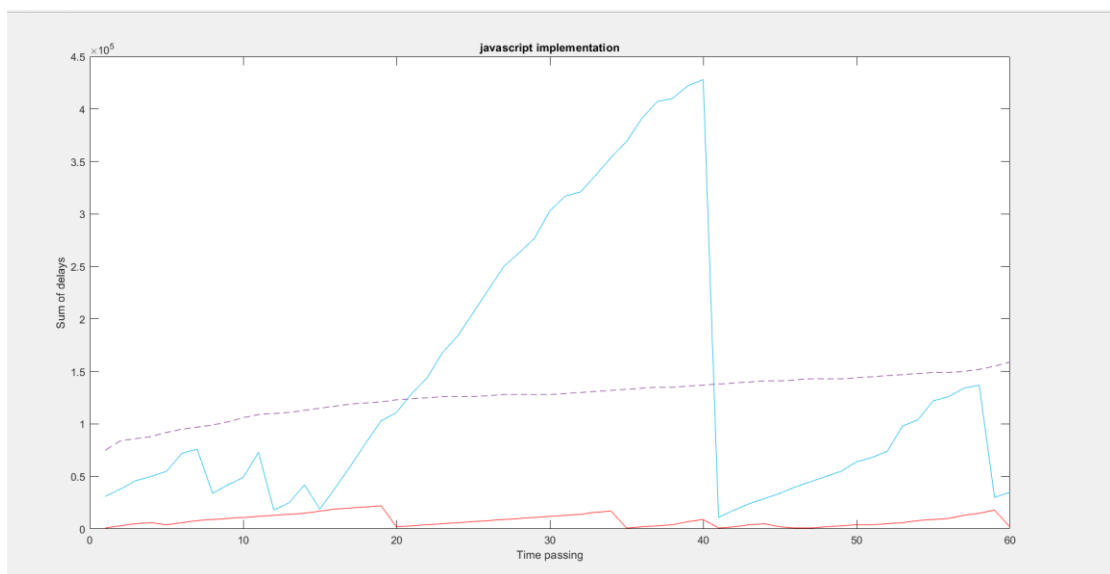
Εναλλακτική προσέγγιση:

Θα μπορούσαμε να υλοποιήσουμε έναν hashmap ανά σύμβολο με key το timestamp και value τον μέσο όρο των μετοχών στο συγκεκριμένο λεπτό που αρχίζει από το timestamp που είναι το key. Όταν περνούσε ένα λεπτό, θα κοιτάγαμε πιο key του map είναι πιο μικρό (πιο παλιό δευτερόλεπτο), και θα το διαγράφαμε από τον hashmap μας (ο οποίος δηλαδή θα είχε πάντα μέχρι 15 key-value pairs).

Sum of delays σύγκριση σε διάγραμμα: Κόκκινο: για candlestick data, διακεκομμένη για moving mean ,μπλε για πληροφορίες μετοχων (ερώτημα 1)



Το ίδιο για javascript implementation:



Για να μην σταματάει το πρόγραμμα, «ακούμε» όταν γίνεται closed to client connection (ακούγοντας το LWS\_CALLBACK\_CLIENT\_CLOSED ή όταν δεν μπορέσαμε να κάνουμε connection, LWS\_CALLBACK\_CLIENT\_CONNECTION\_ERROR) και ξαναπροσπαθούμε να ανοίξουμε το connection (ανακατευθύνοντας στο do\_retry σημείο του κώδικα). Έτσι το πρόγραμμα λειτουργεί χωρίς διακοπές.

Ενδεικτικά αποτελέσματα του προγράμματος: (αριστερά βλέπουμε το αποτέλεσμα με javascript κώδικα, και δεξιά στο linux terminal του c κώδικα)

Stock values αρχεία είναι της μορφής:

BINANCEBTCUSD: Last price: 19450.02, timestamp: 1665258305279, volume: 0.00058	Volume: 0.051950
BINANCEBTCUSD: Last price: 19451.25, timestamp: 1665258305968, volume: 0.01075	Last price: 19965.450000
BINANCEBTCUSD: Last price: 19451.23, timestamp: 1665258306268, volume: 0.02661	Trace conditions: 0.000000
BINANCEBTCUSD: Last price: 19451.59, timestamp: 1665258306508, volume: 0.00257	Timestamp:1664904907815.000000
BINANCEBTCUSD: Last price: 19451.27, timestamp: 1665258306912, volume: 0.03651	Volume: 0.051950
BINANCEBTCUSD: Last price: 19451.75, timestamp: 1665258306977, volume: 0.02571	Last price: 19964.700000
BINANCEBTCUSD: Last price: 19451.27, timestamp: 1665258307300, volume: 0.0006	Trace conditions: 0.000000
BINANCEBTCUSD: Last price: 19451.27, timestamp: 1665258307605, volume: 0.00745	Timestamp:1664904908358.000000
BINANCEBTCUSD: Last price: 19452.52, timestamp: 1665258307605, volume: 0.00523	Volume: 0.020460
BINANCEBTCUSD: Last price: 19453.04, timestamp: 1665258307605, volume: 0.01	Last price: 19964.700000
BINANCEBTCUSD: Last price: 19451.27, timestamp: 1665258307625, volume: 0.01983	Trace conditions: 0.000000
BINANCEBTCUSD: Last price: 19451.38, timestamp: 1665258307732, volume: 0.40258	Timestamp:1664904908537.000000
BINANCEBTCUSD: Last price: 19451.04, timestamp: 1665258307755, volume: 0.10285	Volume: 0.029440
BINANCEBTCUSD: Last price: 19451.45, timestamp: 1665258307991, volume: 0.00503	Last price: 19965.200000
BINANCEBTCUSD: Last price: 19451.52, timestamp: 1665258308050, volume: 0.05075	Trace conditions: 0.000000
BINANCEBTCUSD: Last price: 19451.45, timestamp: 1665258308296, volume: 0.011	Timestamp:1664904908693.000000

Candlestick data:

63	23:09:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000
64	23:10:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	Candlestick for minute starting at: 10/07/22 - 11:45PM
65	23:11:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	close price :118.820000
66	23:12:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	high price :121.801600
67	23:13:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000
68	23:14:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	Candlestick for minute starting at: 10/07/22 - 11:45PM
69	23:15:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	close price :118.820000
70	23:16:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	high price :121.801600
71	23:17:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000
72	23:18:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	Candlestick for minute starting at: 10/07/22 - 11:46PM
73	23:19:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	close price :118.820000
74	23:20:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	high price :121.801600
75	23:21:28 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000
76	23:22:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	Candlestick for minute starting at: 10/07/22 - 11:46PM
77	23:23:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	close price :118.820000
78	23:24:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	high price :121.801600
79	23:25:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000
80	23:26:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	Candlestick for minute starting at: 10/07/22 - 11:46PM
81	23:27:29 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	high price :121.801600
82	22:42:59 GMT+0300 (Eastern European Summer Time)	{"s":"no_data"}	low price :118.070000

Moving mean results (για BINANCE symbol) (αντίστοιχα για volume):

secondClient, client1, client2, # BINANCEBTCUSD X main.js, utilities.js	
MOVING MEANS RESULTS > # BINANCEBTCUSD X	
346	Mean for 1 stock values is 19801.74576000001
347	Mean for 1 stock values is 19801.74576000001
348	Mean for 1 stock values is 19801.74576000001
349	Mean for 1 stock values is 19801.74576000001
350	Mean for 1 stock values is 19801.74576000001
351	Mean for 1 stock values is 19801.74576000001
352	Mean for 1 stock values is 19801.74576000001
353	Mean for 1 stock values is 19801.74576000001
354	Mean for 1 stock values is 19801.74576000001
355	Mean for 1 stock values is 19801.74576000001
356	Mean for 1 stock values is 19801.74576000001
357	Mean for 1 stock values is 19801.74576000001
358	Mean for 1 stock values is 19801.74576000001
359	Mean for 1 stock values is 19801.74576000001
360	Mean for 1 stock values is 19801.74576000001
361	Mean for 1 stock values is 19801.74576000001
362	Mean for 1 stock values is 19801.74576000001
363	Mean for 1 stock values is 19801.74576000001
364	Mean for 1 stock values is 19801.74576000001
365	Mean for 1 stock values is 19801.74576000001
366	Mean for 1 stock values is 19801.74576000001
367	Mean for 1 stock values is 19801.74576000001
368	Mean for 1 stock values is 19801.74576000001
369	Mean for 1 stock values is 19801.74576000001
370	Mean for 1 stock values is 19801.74576000001
371	Mean for 1 stock values is 19801.74576000001
372	Mean for 1 stock values is 19801.74576000001
373	Mean for 1 stock values is 19801.74576000001
374	Mean for 1 stock values is 19801.74576000001
375	Mean for 1 stock values is 19801.74576000001
376	Mean for 1 stock values is 19801.74576000001
377	Mean for 1 stock values is 19801.74576000001
378	Mean for 1 stock values is 19801.74576000001
379	Mean for 1 stock values is 19801.74576000001
380	Mean for 1 stock values is 19801.74576000001
381	Mean for 1 stock values is 19801.74576000001
382	Mean for 1 stock values is 19801.74576000001
383	Mean for 1 stock values is 19801.74576000001
384	Mean for 1 stock values is 19801.74576000001
385	Mean for 1 stock values is 19801.74576000001
386	Mean for 1 stock values is 19801.74576000001
387	Mean for 1 stock values is 19801.74576000001
388	Mean for 1 stock values is 19801.74576000001

sofi@DESKTOP-MGUKRET: ~/enigmatomona/ergasia  
moving mean for last 4 minutes: 4214040.750000  
moving mean for last 1 minutes: 4901745.000000  
moving mean for last 2 minutes: 4470615.000000  
moving mean for last 3 minutes: 4292142.000000  
moving mean for last 4 minutes: 4172398.000000  
moving mean for last 5 minutes: 4449207.800000  
moving mean for last 6 minutes: 4266242.500000  
moving mean for last 7 minutes: 4132848.285714  
moving mean for last 8 minutes: 4056694.125000  
moving mean for last 9 minutes: 4007795.000000  
moving mean for last 10 minutes: 3907825.200000  
moving mean for last 11 minutes: 3916358.545455  
moving mean for last 12 minutes: 3859669.333333  
moving mean for last 13 minutes: 3979631.230769  
moving mean for last 14 minutes: 3990856.642857  
moving mean for last 15 minutes: 3985273.533333  
moving mean for last 15 minutes: 3958068.000000  
moving mean for last 15 minutes: 3928007.666667  
moving mean for last 15 minutes: 3983042.933333  
moving mean for last 1 minutes: 3619649.000000  
moving mean for last 2 minutes: 4031205.000000  
moving mean for last 3 minutes: 4078791.666667  
moving mean for last 4 minutes: 4322934.750000  
moving mean for last 5 minutes: 4308790.800000  
moving mean for last 6 minutes: 4727399.166667  
moving mean for last 7 minutes: 5097627.714286  
moving mean for last 8 minutes: 5116795.625000  
moving mean for last 9 minutes: 5108202.555556  
moving mean for last 10 minutes: 5164553.400000  
sofi@DESKTOP-MGUKRET: ~/enigmatomona/ergasia\$

Επίσης συγκρίνουμε την απόδοση των δυο implementation, με βάση τα delays (με κόκκινο το js implementation, με μπλέ το c). Προφανώς το δεύτερο έχει πολύ καλύτερη απόδοση.

