

# System Programming HW3

Ορέστης Μελκονιάν – 1115201000128

## Συμβάσεις

- Η παράμετρος *SIZE* πάντα μετατρέπεται στον επόμενο ακέραιο που είναι δύναμη του 2
- Αν δοθεί η παράμετρος *-h ADDRESS* η σύνδεση θα γίνει στη θύρα *PORT*
- Σε περίπτωση διαφορετικών ορισμάτων, ο client θα τερματίσει με μήνυμα λάθους
- Το makefile έχει παραχθεί αυτόματα από το Eclipse
- Μπορούν να δοθούν και οι παράμετροι *[-s SEED] [-d DIFFICULTY] [-noserve]*

## Οδηγίες μεταγλώττισης/εκτέλεσης

- compile: `$ cd Debug/ ; make all`
- clean: `$ cd Debug/ ; make clean`
- run: `$ cd Debug/ ; ./invoke-oracle-multithreaded <PARAMETERS>`
- measure: `$ cd Debug/ ; ./measure.sh`

## Ενδεικτική εκτέλεση (seed = 1, easy mode)

### SERVER - linux07.di.uoa.gr

```
$ ./invoke 10000000 4 50 9006 logfile.txt
$ Main Logfile initialized
$ Main Seeds initialized
$ Main Oracle initialized
$ Main Server initialized
$ Thread 3035310912 Starting search
$ Main Threads created
$ Thread 3052096320 Starting search
$ Thread 3043703616 Starting search
$ Thread 3020946240 Starting search
$ Server created socket
$ Server binded socket
$ Server Listening for connections to port 9005
$ Server Accepted connection
$ Server Parameters sent
$ Server Bloom filter locked
$ Server Bloom filter sent
$ Server Bloom filter unlocked
$ Thread 3035310912 Secret word is
    nwlrbmqbhcdarzowkkyhiddqscdx
$ Thread 3035310912 Finished
$ Thread 3035310912 Outputting statistics
$ Thread 3052096320 Finished
$ Thread 3020946240 Finished
$ Thread 3043703616 Finished
$ Thread 3052096320 Outputting statistics
$ Thread 3020946240 Outputting statistics
$ Thread 3043703616 Outputting statistics
$ Main Threads terminated
```

### CLIENT - linux22.di.uoa.gr

```
$ ./invoke 10000000 4 50 9006 logfile.txt -h linux07.di.uoa.gr
$ Client socket created
$ Client Connected to linux07.di.uoa.gr:9005
$ Client Got parameters
$ Client Cross-checked
$ Client Got bloom filter
$ Client Finished
$ Main Logfile initialized
$ Main Seeds initialized
$ Main Oracle initialized
$ Main Server initialized
$ Server created socket
$ Server binded socket
$ Server Listening for connections to port 9005
$ Thread 3050306368 Starting search
$ Main Threads created
$ Thread 3039816512 Starting search
$ Thread 3031423808 Starting search
$ Thread 3023031104 Starting search
$ Thread 3031423808 Secret word is
    nwlrbmqbhcdarzowkkyhiddqscdx
$ Thread 3031423808 Finished
$ Thread 3031423808 Outputting statistics
$ Thread 3023031104 Finished
$ Thread 3050306368 Finished
$ Thread 3039816512 Finished
$ Thread 3023031104 Outputting statistics
$ Thread 3050306368 Outputting statistics
$ Thread 3039816512 Outputting statistics
$ Main Threads terminated
```

## Πειραματική Αποτίμηση

### ΧΡΟΝΟΣ

- Οι χρονικές μετρήσεις έγιναν αυτόματα με ένα bash script (*/Debug/measure-many.sh*) ταυτόχρονα σε διαφορετικά μηχανήματα Linux της σχολής. Η ποικιλία στο φόρτο εργασίας του κάθε μηχανήματος, σε συνδυασμό με την τυχαία αρχικοποίηση της πρώτης λέξης κάθε νήματος καθώς και την τυχαία σειρά που θα γίνουν scheduled τα threads, οδηγούν σε αρκετά αναξιόπιστα αποτελέσματα.
- Για τις μετρήσεις, τα seeds για την rand\_r αρχικοποιούνται με σταθερή τιμή (0..N, όπου N τα threads) ώστε να ελαχιστοποιηθεί ο μη-ντετερμινισμός στις εκτελέσεις.
  - πχ. Αν προσθέσουμε ένα thread ακόμα(N+1) στην επόμενη εκτέλεση, τα N πρώτα threads θα αρχίσουν με την ίδια συμβολοσειρά και το N+1 θα ξεκινήσει από καινούρια “τυχαία”.
- Στο αρχείο **measures.txt** φαίνονται τα αποτελέσματα για seed=[1, 100] και threads=[1, 10] σε διαφορετικά μηχανήματα της σχολής. Οι σταθερές παράμετροι είναι οι εξής:
  - SIZE: 100000000
  - L: 5
  - PORT: 9006
  - SECTION\_SIZE: 512 bits
  - T: 200
  - HASH\_NO: 3
- Στον φάκελο /logfiles/ δίνονται τα logfiles από διάφορες εκτελέσεις σε easy\_mode με διαφορετικό seed και αριθμό threads.  
Τα αρχεία είναι ονομασμένα ως: logfile<SEED>-<#THREADS>.txt

### ΣΥΜΠΕΡΑΣΜΑΤΑ

- Δεν κερδίζουμε ιδιαίτερα όταν τα threads είναι περισσότερα από τα logical threads του επεξεργαστή(2), διότι η λειτουργία που εκτελούμε είναι τελείως CPU-heavy και συνεπώς δεν θα εκμεταλλευτούμε άλλου είδους παραλληλίες από άλλα περιφερειακά(ΙΟ, GPU, etc...). Πέρα από αυτό, θα υπάρξει επιβράδυνση λόγω του context-switch από thread σε thread.
- Ειδικότερα, όταν η αναζήτηση είναι πολύ γρήγορη (< 2 sec) περισσότερα thread θα καθυστερήσουν την εκτέλεση λόγω των mutex για το logfile και για printing.
- Γενικότερα, σε δύσκολες αναζητήσεις (> 20 sec), βλέπουμε σημαντική βελτίωση με περισσότερα threads, τουλάχιστον μέχρι τα 4.
- Μακράν το μεγαλύτερο speedup παρατηρείται από 1 στα 2 threads.

### ΜΝΗΜΗ

- Δεν υπάρχουν memory leaks και η μνήμη που καταλαμβάνει το πρόγραμμα παραμένει σταθερή καθ'όλη τη διάρκεια εκτέλεσης. Αυτή η σταθερή ποσότητα αποτελείται κυρίως από το bloom filter και τα mutex των sections του bloom filter.
- Το μέγεθος των sections είναι 512 bytes=4096 bits και κάθε mutex καταλαμβάνει sizeof(pthread\_mutex\_t)=40. Δηλαδή, ο χώρος είναι περίπου ο διπλάσιος από το μέγεθος του bloom filter.
- Η αναζήτηση γίνεται κατά βάθος (DFS), οπότε η χωρική πολυπλοκότητα είναι  $O(n)$ , αντί για  $O(n^2)$  στην περίπτωση της αναζήτησης κατά πλάτος (BFS). Οπότε οι λέξεις που παραμένουν στην στοίβα δεν καταλαμβάνουν σημαντικό χώρο μνήμης.