

# Παράλληλα και Διανεμημένα Συστήματα

## Πρώτη εργασία

Θεόδωρος Κατζάλης  
AEM:9282  
katzalis@auth.gr

6/12/2020

## Contents

### 1 Version 4 pthreads

1

## 1 Version 4 pthreads

Το **load balancing** είναι ιδιαίτερα σημαντικό. Παρατηρούμε ότι όταν διαμοιράζουμε την δουλειά με βάση το index, δεν βλέπουμε βελτίωση στην απόδοση, μιας και την περισσότερη δουλειά την κάνει μόνο ένα thread. Συνεπώς θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί με το scheduling των threads. Τα ίδια συμπεράσματα μπορούσαμε να εξάγουμε και απο την openmp υλοποίηση για static scheduling. Οπότε με dynamic scheduling σε pthreads και openmp έχουμε πολύ καλύτερα αποτελέσματα... Βασική διαφορά είναι ότι σε pthreads είναι λίγο πιο δύσκολο να υλοποιηθεί (στην openmp είναι μονο ένα keyword static -> dynamic). Το πρώτο thread κάνει σχεδόν όλη την δουλειά.

```
-----Version 4-----
Tic: 10730 seconds and 891511404 nanoseconds
Toc: 10733 seconds and 98114700 nanoseconds
Time elapsed (seconds): 2.206603
Total number of triangles: 3056386

-----Version 4 Pthread is called-----
Tic: 10733 seconds and 98406652 nanoseconds
Current stack size -> 8388608
Hello, I am 1
Start is 141861 and end is 283722
Hello, I am 2
Hello, I am 0
Start is 0 and end is 141861
Hello, I am 3
Start is 425583 and end is 567444
Hello, I am 4
Start is 567444 and end is 709305
Start is 283722 and end is 425583
Hello, I am 5
Start is 709305 and end is 851166
Hello, I am 7
Start is 993027 and end is 1134890
Hello, I am 6
Start is 851166 and end is 993027
Finished id: 6. Elapsed time: 0.004249
Finished id: 7. Elapsed time: 0.013671
Finished id: 5. Elapsed time: 0.015528
Finished id: 3. Elapsed time: 0.066082
Finished id: 2. Elapsed time: 0.076644
Finished id: 4. Elapsed time: 0.128299
Finished id: 1. Elapsed time: 0.148125
Finished id: 0. Elapsed time: 2.041982
Toc: 10735 seconds and 156086076 nanoseconds
Time elapsed (seconds): 2.057679
Total number of triangles: 3056386
```

Figure 1: Load balancing pthreads, static scheduling. Matrix com-Youtube.mtx