

Παράλληλα και Διανεμημένα Συστήματα

Πρώτη εργασία

Θεόδωρος Κατζάλης

AEM:9282

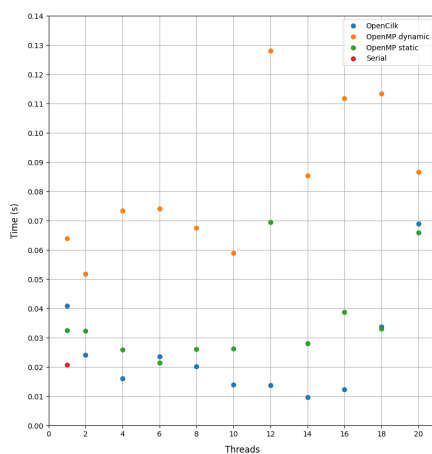
katzalis@auth.gr

6/12/2020

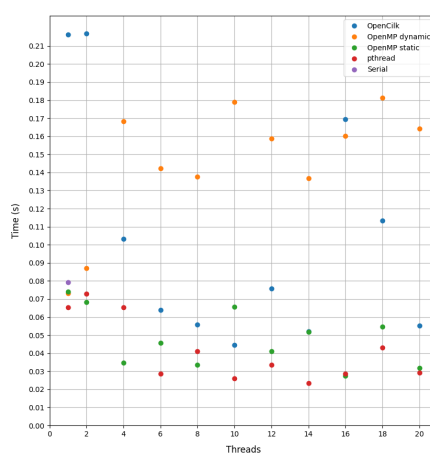
Contents

1	$n = 1.441.295$, $m = 3.099.940$ (belgium_osm.mtx)	1
2	$n = 1.134.890$, $m = 5.975.248$ (com-Youtube.mtx)	2
3	$n = 326,186$, $m = 1,615,400$ (dblp-2010.mtx)	2
4	$n = 6.143$, $m = 1.227.742$ (mycielskian.mtx)	2
5	$n = 1.039.183$, $m = 6.229.636$ (NACA0015.mtx)	3
6	Συμπεράσματα	3
7	Load balancing	3
8	Notes	3

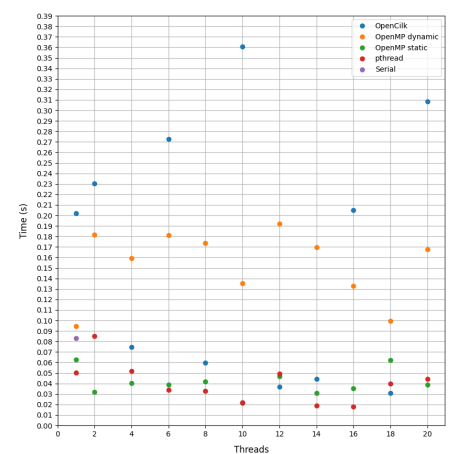
1 $n = 1.441.295$, $m = 3.099.940$ (belgium_osm.mtx)



(a) V3 binary search

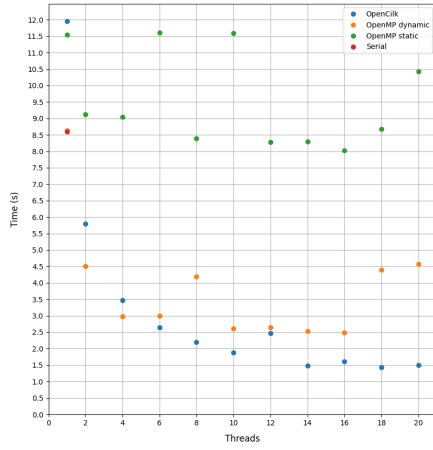


(b) V4 binary search

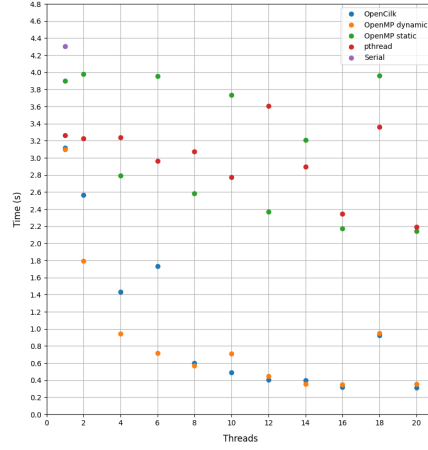


(c) V4 linear search

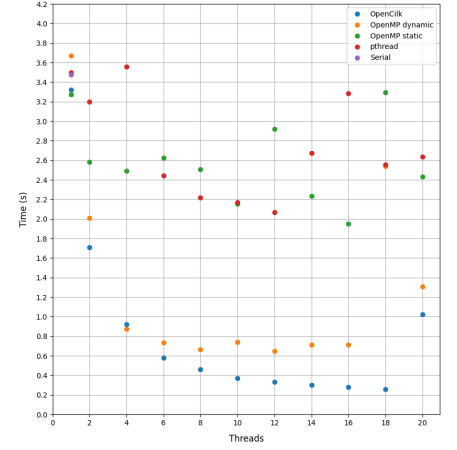
2 $n = 1.134.890$, $m = 5.975.248$ (com-Youtube.mtx)



(a) V3 binary search

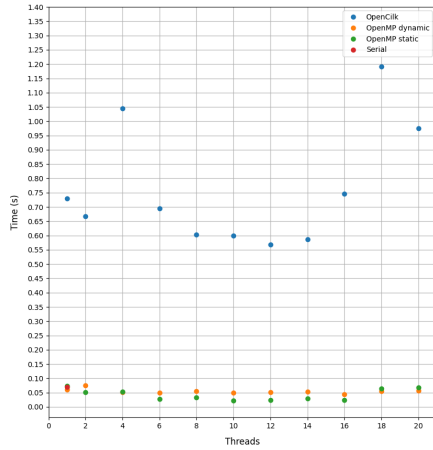


(b) V4 binary search

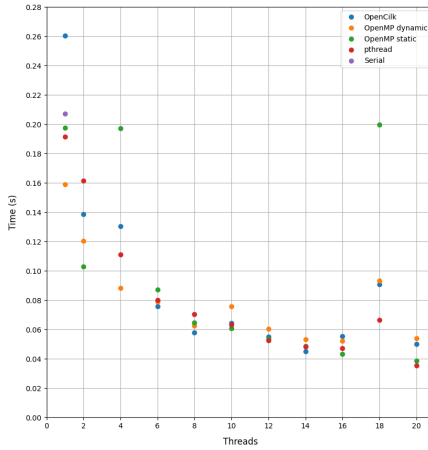


(c) V4 linear search

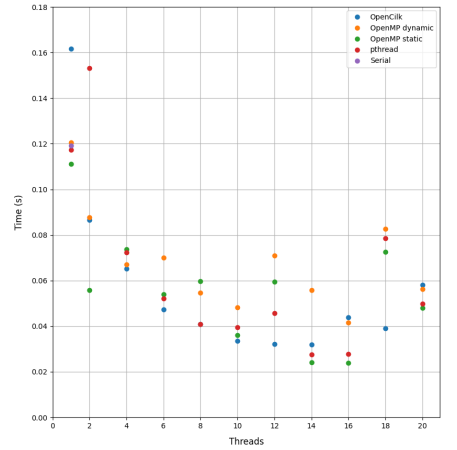
3 $n = 326,186$, $m = 1,615,400$ (dblp-2010.mtx)



(a) V3 binary search

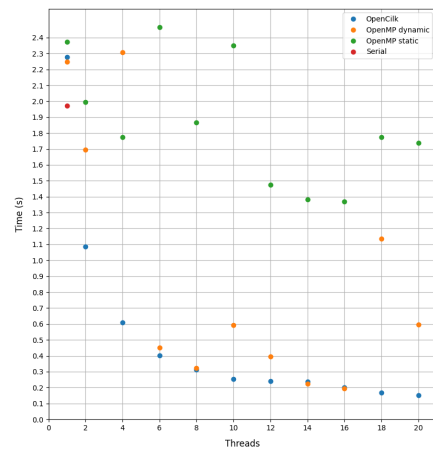


(b) V4 binary search

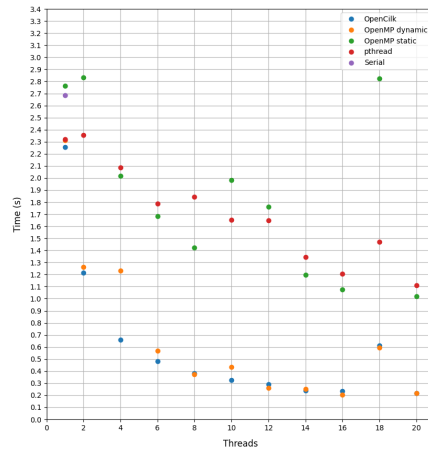


(c) V4 linear search

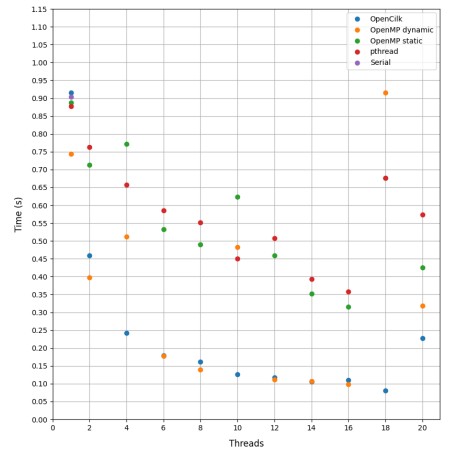
4 $n = 6.143$, $m = 1.227.742$ (mycielskian.mtx)



(a) V3 binary search

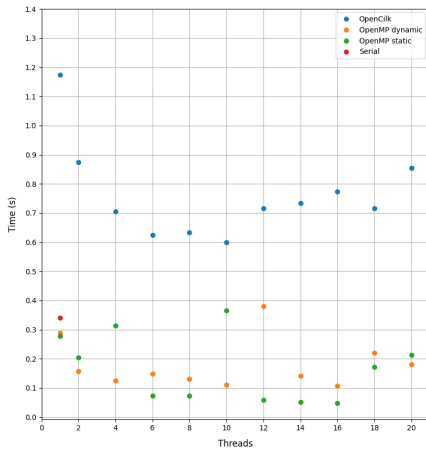


(b) V4 binary search

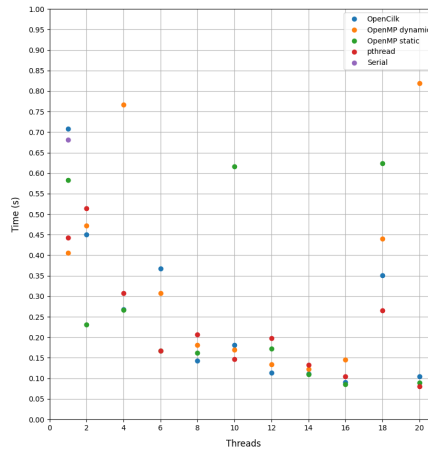


(c) V4 linear search

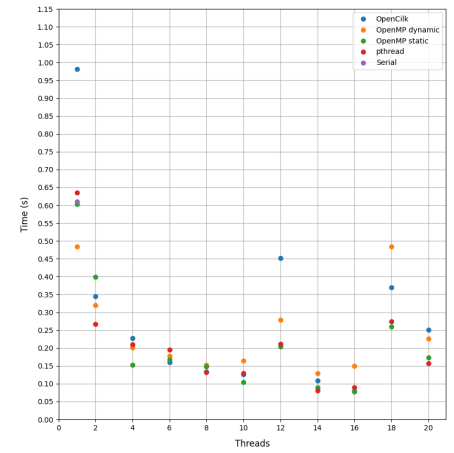
5 $n = 1.039.183$, $m = 6.229.636$ (NACA0015.mtx)



(a) V3 binary search



(b) V4 binary search



(c) V4 linear search

6 Συμπεράσματα

7 Load balancing

The more the loops are balanced, the better our **static** implementation will be. Για παράδειγμα, παρατηρώντας τις δύο παρακάτω εικόνες¹, το πρώτο thread κάνει σχεδόν όλη την δουλειά προκαλώντας load imbalance και επηρεάζοντας σημαντικά το speedup. Στα πλαίσια του χρονικού διαστήματος της εργασίας, λείπει μια δυναμική έκδοση των pthreads. Εικάζουμε ότι σε τέτοιες περιπτώσεις θα ήταν αρκετά πιο αποδοτική (βλέποντας και την απόκριση του schedule dynamic σε openmp υλοποίηση).

Με αυτά τα γραφήματα μπορούμε να εξηγήσουμε την διαγορευτική απόδοση μεταξύ dynamic και static scheduling στους παραπάνω πίνακες.

```
-----Version 4-----
Tic: 11544 seconds and 94231585 nanoseconds
Toc: 11546 seconds and 301739210 nanoseconds
Time elapsed (seconds): 2.207508
Total number of triangles: 3056386

-----Version 4 Pthread is called-----
Tic: 11546 seconds and 302021981 nanoseconds
Current stack size -> 8388608
Hello, I am 0
For id: 0. Start is 0 and end is 141861
Hello, I am 1
For id: 1. Start is 141861 and end is 283722
Hello, I am 3
For id: 3. Start is 425583 and end is 567444
Hello, I am 4
For id: 4. Start is 567444 and end is 709305
Hello, I am 7
For id: 7. Start is 993027 and end is 1134890
Hello, I am 2
For id: 2. Start is 283722 and end is 425583
Finished id: 7. Elapsed time: 0.019409
Hello, I am 5
For id: 5. Start is 709305 and end is 851166
Hello, I am 6
For id: 6. Start is 851166 and end is 993027
Finished id: 6. Elapsed time: 0.004363
Finished id: 5. Elapsed time: 0.021700
Finished id: 3. Elapsed time: 0.065244
Finished id: 2. Elapsed time: 0.077734
Finished id: 1. Elapsed time: 0.146834
Finished id: 4. Elapsed time: 0.152418
Finished id: 0. Elapsed time: 2.032460
Toc: 11548 seconds and 350684285 nanoseconds
Time elapsed (seconds): 2.048662
Total number of triangles: 3056386
```

(a) Load balancing pthreads, Scheduling: static, Threads: 8. Matrix com-Youtube.mtx. 10% speedup

```
-----Version 4-----
Tic: 12223 seconds and 621036786 nanoseconds
Toc: 12223 seconds and 930843929 nanoseconds
Time elapsed (seconds): 0.309807
Total number of triangles: 7978861

-----Version 4 Pthread is called-----
Tic: 12223 seconds and 930932289 nanoseconds
Current stack size -> 8388608
Hello, I am 0
For id: 0. Start is 0 and end is 65536
Hello, I am 7
For id: 7. Start is 458752 and end is 524288
Hello, I am 1
For id: 1. Start is 65536 and end is 131072
Hello, I am 2
For id: 2. Start is 131072 and end is 196608
Hello, I am 6
For id: 6. Start is 393216 and end is 458752
Hello, I am 5
For id: 5. Start is 327680 and end is 393216
Hello, I am 3
For id: 3. Start is 196608 and end is 262144
Hello, I am 4
For id: 4. Start is 262144 and end is 327680
Finished id: 7. Elapsed time: 0.097889
Finished id: 5. Elapsed time: 0.095962
Finished id: 0. Elapsed time: 0.116595
Finished id: 2. Elapsed time: 0.123036
Finished id: 6. Elapsed time: 0.126428
Finished id: 3. Elapsed time: 0.123485
Finished id: 1. Elapsed time: 0.134692
Finished id: 4. Elapsed time: 0.124316
Toc: 12224 seconds and 81524391 nanoseconds
Time elapsed (seconds): 0.150592
Total number of triangles: 7978861
```

(b) Load balancing pthreads. Scheduling: static, Threads: 8. Matrix rgg_n_2_19_s0.mtx. 50% speedup.

8 Notes

- Αναγνώριση χρονοβόρων διαδικασιών/επαναλήψεων
- Αφαίρεση εξαρτήσεων μεταξύ των επαναλήψεων (loop carrier dependency)

¹ Δεν έχει τρέξει στην συστοιχία αλλά σε προσωπικό σύστημα. FIX ME!

- Αναγνώριση βέλτιστου scheduling (κατανομή "βάρους")
- Binary vs Linear search sorted not very different