

Vertexwise Triangle Counting
Παράλληλα και Διανεμημένα Συστήματα
Assignment 1

Φίλης Χάρης

December 6, 2020

Αποθετήριο Εργασίας :

<https://github.com/harryfilis/Parallel-and-Distributed-Systems-Assignments/tree/master/Vertexwise-triangle-counting-asgmt1>

0.1 Λίγα λόγια για την δομή δεδομένων

Το πρώτο βήμα στην ανάπτυξη κώδικα για τη συγκεκριμένη εργασία ήταν η ανάγνωση Market Matrix. Υπάρχει επιλογή να δώσει ο χρήστης την τιμή 0 για μη δυαδικό πίνακα και 1 για δυαδικό πίνακα (default 1). Έτσι γνωρίζει το πρόγραμμα αν ο coo πίνακας έχει 3 ή 2 στοιχεία ανά γραμμή και με αυτόν τον τρόπο μπορεί να τα διαβάσει δίχως πρόβλημα μνήμης. Ωστόσο δεν δουλεύει για πίνακες με μη μηδενικά στοιχεία στη διαγώνιο.

Ακόμη, για το v3 έχω τους πίνακες σε άνω τριγωνική μορφή, οπότε ήθελα να συγκρίνω την στήλη και τη γραμμή του συμμετρικού πίνακα ώστε, ανάλογα με το τι ήταν μεγαλύτερο, να το περάσω σωστά ως όρισμα στην συνάρτηση coo2csc και να βγει άνω τριγωνικός

0.2 Ανάλυση Αλγορίθμου v3

```
1 V3:
2 for i = 1:n-2
3     for j = adjacent to i % data structure only lists A(i,j)~=0
4         for k = adjacent to j % data structure only lists A(j,k)~=0
```

Στο V3, έχουμε τον πίνακα σε δομή CSC(compressed sparse column).

Ξεκινώντας από τυχαίο στοιχείο πχ (row1,col1) ισχύει ότι:

(row1,col1)->(col1,col2)->(col2,row1)

Στο format CSC κάνω αναζήτηση ανά στήλη βρίσκοντας ποια γραμμή είναι μη μηδενική, επομένως έχω στα χέρια μου το πρώτο στοιχείο (row1, col1). Στη συνέχεια (και επειδή, λόγω συμμετρίας του πίνακα, ισχύει ότι στήλη = γραμμή και το αντίθετο) μπορώ να ψάξω για το στήλη row1 και να βρω ποιες τιμές υπάρχουν ως γραμμές. Συνεπώς θα έχω πλέον στα χέρια μου και το στοιχείο (col2,row1). Προκειμένου να υπάρχει λοιπόν τρίγωνο πρέπει να αναζητήσω αν στη στήλη row1 υπήρχε στοιχείο στη γραμμή με συντεταγμένη col1! Σε περίπτωση που υπάρχει αυξάνω τον αριθμό τριγώνων και τον πίνακα c3 για τις αντίστοιχες κορυφές row1, col1, col2.

0.2.1 Cilk/Openmp

V3

Χρησιμοποίησα το cilkfor/prAGMA omp parallel for για να παραλληλοποιήσω το εξωτερικό for loop και να δουλέψω για κάθε στήλη παράλληλα.

0.3 Αλγόριθμος 3.1 Matlab

```
1 n = 10
2 A = zeros(n,n);
3 for i=1:n
4     for j=i:n
5         if j == i
6             A(i,j) = 0;
7         else
8             A(i,j) = randi(2)-1;
9             A(j,i) = A(i,j);
10        end
11    end
12 end
13 A
14 e = ones(n, 1)
15
16 c3 = (A.*(A*A)) * e ./ 2
```

0.4 Διαγράμματα - Χρόνοι εκτέλεσης κάθε προγράμματος

Οι χρόνοι εκτέλεσης προγραμμάτων μπορούν να βρεθούν στο φάκελο [times](#) στο αποθετήριο ή συγκεντρωτικά

V3

	belgium	youtube	dblp	cielskian	NACA
v3	0.014	4.13	0.15	11.05	0.21
---	---	---	---	---	---
v3_openmp (2)	0.021	2.31	0.20	9.2	0.30
v3_openmp (4)	0.011	1.75	0.26	6.6	0.34
v3_openmp (6)	0.008	1.51	0.36	5.3	0.43
v3_openmp (8)	0.056	1.36	0.4	4.4	0.50
---	---	---	---	---	---
v3_cilk (2)	0.0185	2.01	0.23	5.8	0.41
v3_cilk (4)	0.021	1.02	0.25	2.7	0.35
v3_cilk (6)	0.024	0.82	0.28	2.1	0.38
v3_cilk (8)	0.024	0.83	0.31	1.86	0.55

στον πίνακα.

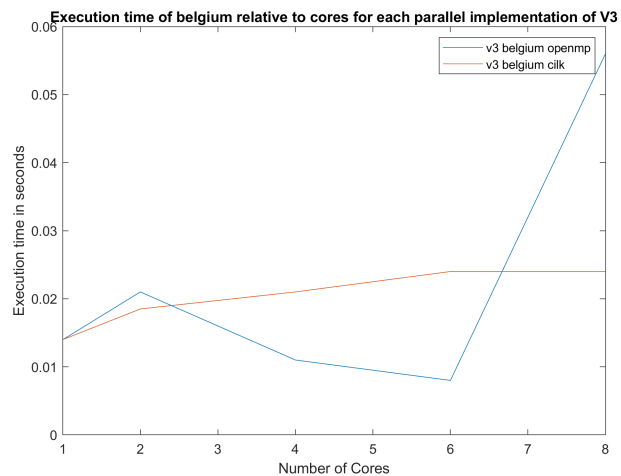


Figure 1: belgium

Αποθετήριο Εργασίας :

<https://github.com/harryfilis/Parallel-and-Distributed-Systems-Assignments/tree/master/Vertexwise-triangle-counting-asgmt1>

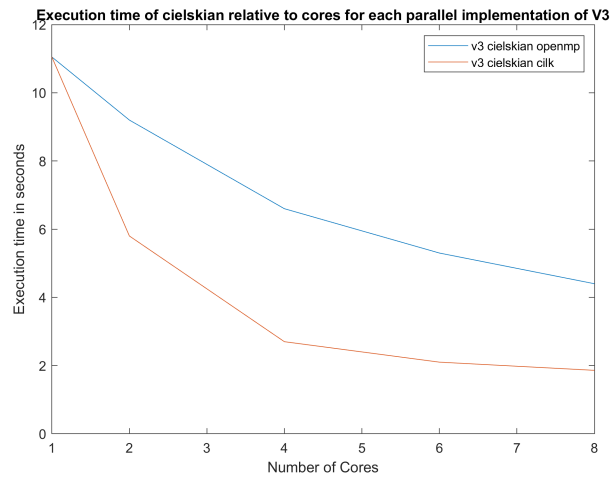


Figure 2: cielskian

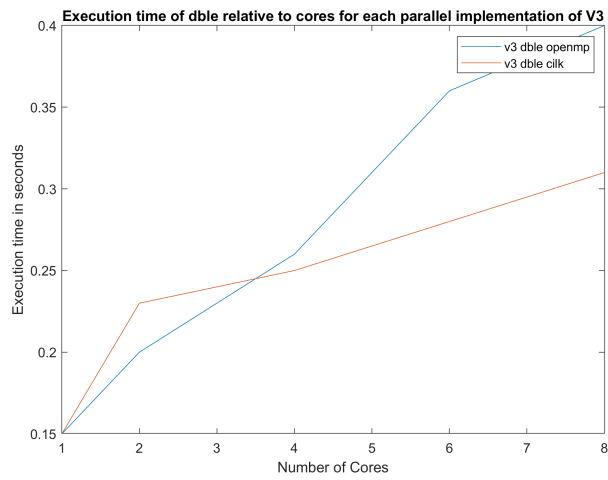


Figure 3: dble

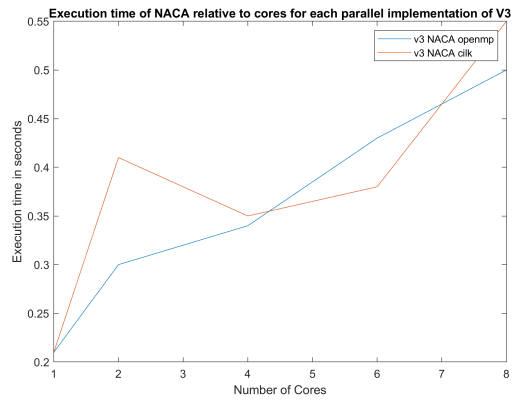


Figure 4: NACA

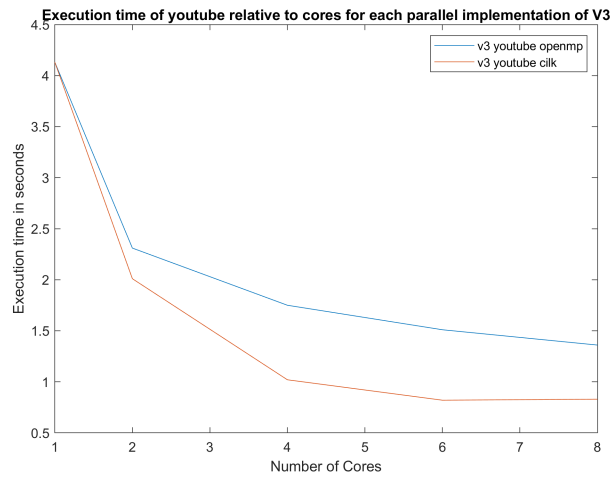


Figure 5: youtube