

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών

Παράλληλα και Διανεμημένα Συστήματα

Εργασία 1η

Στοιχεία Φοιτητών:

Ον/μα: Μέτα Λουίς Κοσμάς & Ούρδας Αντώνιος

AEM: 9390 & 9358

Emails: louismeta@ece.auth.gr & ourdasav@ece.auth.gr

Github: <https://github.com/lkmeta/Parallel-and-Distributed-Systems>

Σχολιασμός υλοποίησης των v3 και v4 αλγορίθμων

Ο αλγόριθμος της v3 χρησιμοποιώντας την CSC μορφή του κάτω τριγωνικού πίνακα διασχίζει τους κόμβους και ελέγχει αν υπάρχουν κοινοί γειτονικοί κόμβοι. Με αυτή την διαδικασία υπολογίζεται ο πίνακας c3 που είναι το ζητούμενο. Η σειριακή υλοποίηση του αλγορίθμου είχε σχετικά καλούς χρόνους αλλά δεν ικανοποιούσαν όλοι τους απαιτούμενους. Με χρήση παράλληλου προγραμματισμού, καταφέραμε να προσεγγίσουμε λίγο καλύτερα τους επιθυμητούς χρόνους όπως φαίνεται παρακάτω για τους πίνακες με μεγάλο n όπως ο "com-Youtube.mtx" και ο "mycielskian13.mtx".

Ο αλγόριθμος της v4 χρησιμοποιώντας την CSC μορφή ολόκληρου του πίνακα διασχίζει τους κόμβους και υπολογίζει τον πίνακα c3 από τον δοσμένο τύπο. Η σειριακή υλοποίηση του αλγορίθμου είχε σημαντικά καλύτερους χρόνους από αυτούς της v3. Ο παράλληλος προγραμματισμός εδώ δεν είχε τρομερή βελτίωση και μας δυσκόλεψε αρκετά για να πετύχουμε σε κάποιες περιπτώσεις καλύτερους χρόνους.

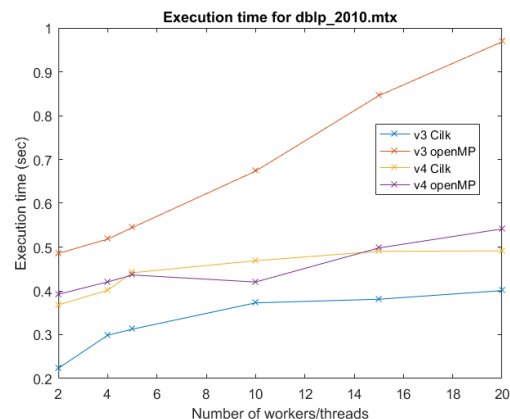
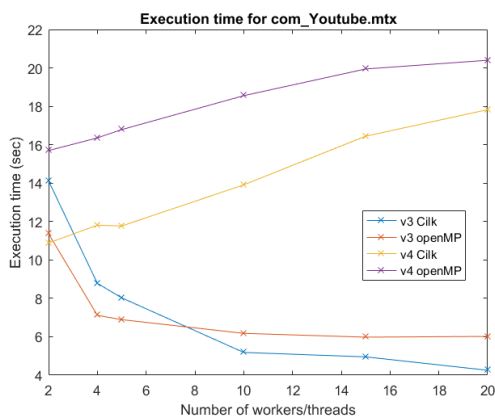
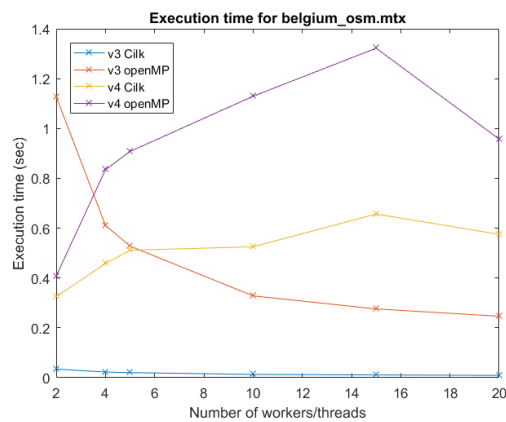
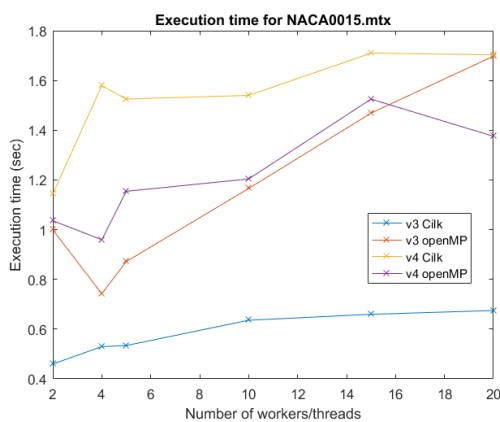
<i>stats.txt</i>	v3	v3_cilk	v3_openMP	v4	v4_cilk	v4_openMP
<i>belgium_osm</i>	0.022009	0.022752	0.611180	0.117482	0.458160	0.834361
<i>com-Youtube</i>	52.206252	8.779148	7.116321	17.027769	11.810333	16.354739
<i>dblp-2010</i>	0.088070	0.298646	0.518053	0.279983	0.401437	0.420376
<i>mycielskian13</i>	21.467567	3.188361	2.240900	4.351176	2.769288	4.038416
<i>NACA0015</i>	0.170284	0.529403	0.742436	1.155859	1.578928	0.959249

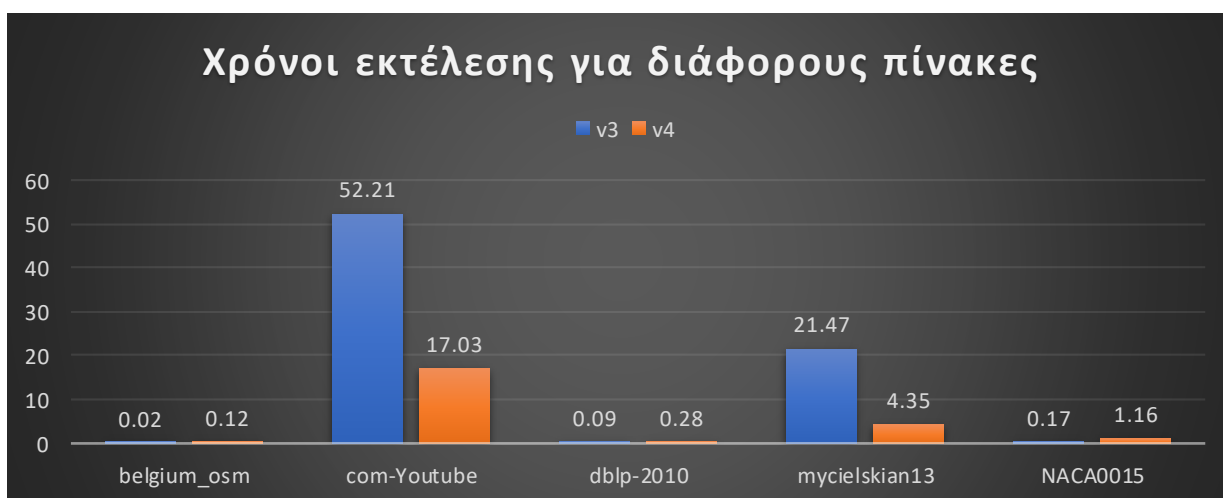
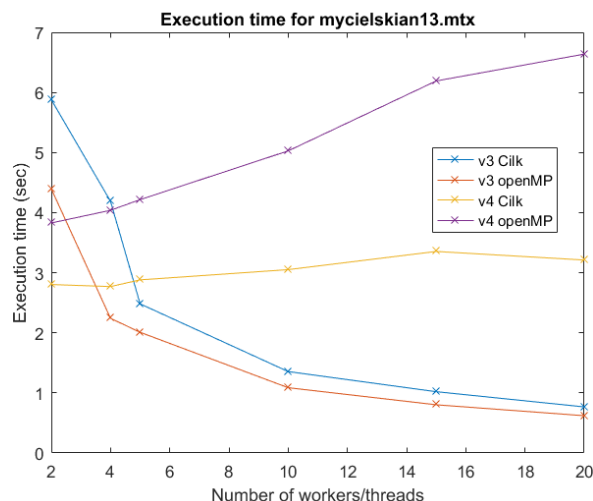
Παρατηρήσεις

- Για μικρό αριθμό κόμβων η παραλληλοποίηση δεν έκανε κάποια ιδιαίτερη διαφορά στους χρόνους και κάποιες φορές μάλιστα γινόταν και πιο αργή.
- Ένας χρόνος που μας δυσκόλεψε να τον προσεγγίσουμε στον επιθυμητό ήταν για το "com-Youtube.mtx" καθώς είναι ένας αρκετά πυκνός πίνακας.
- Ο "mycielskian13.mtx" πίνακας με σειριακή v3 καθυστέρησε αρκετά αλλά με παραλληλοποίηση και με τον αλγόριθμο του v4 έφτασε σε σημαντικά καλύτερους χρόνους.
- Η υλοποίηση του pthreads για το v4 μας δυσκόλεψε και δεν καταφέραμε να έχουμε κάποιον αποδοτικό κώδικα ώστε να τον προσθέσουμε στην παράδοση αυτής της εργασίας.

Διαγράμματα

Τα παρακάτω διαγράμματα προέκυψαν για τους πέντε πίνακες με χρήση των v3_cilk, v3_openMP, v4_cilk και v4_openMP αλγορίθμων για διαφορετικούς αριθμούς πυρήνων. Οι μετρήσεις έγιναν με την χρήση scripts στην συστοιχία και τα διαγράμματα μέσω του Matlab.





Σχολιασμός διαγραμμάτων

Η παραλληλοποίηση του αλγορίθμου v3 παρατηρώντας τα παραπάνω διαγράμματα φαίνεται να έχει καλύτερη απόδοση από την παραλληλοποίηση του v4 γεγονός που πιθανόν να οφείλεται στην αποδοτικότητα των αλγορίθμων. Συγκεκριμένα, ο σειριακός αλγόριθμος της v4 ήταν συγκριτικά με τον αντίστοιχο της v3 σημαντικά αποδοτικότερος στον χρόνο για μεγάλους πίνακες όπως φαίνεται στο τελευταίο διάγραμμα. Αντίθετα, για μικρότερους πίνακες ο v3 υπολόγισε ταχύτερα το ζητούμενο αποτέλεσμα.

Η OpenMP δεν λειτούργησε εξίσου ικανοποιητικά όσο η Cilk παρόλο που δοκιμάσαμε και πειραματιστήκαμε με αρκετούς διαφορετικούς συνδυασμούς πάνω στον αλγόριθμο μας. Ευελπιστούμε στο μέλλον να αποκτήσουμε μεγαλύτερη εξοικείωση με τα παραπάνω εργαλεία.