Άσκηση 1

Παράλληλα και Διανεμημένα Συστήματα

Θωμάς Πλιάκης tpliakis@ece.auth.com
ΑΕΜ: 9018

Αντώνης Μαυρομανώλης antomavr@ece.auth.gr
ΑΕΜ: 9010

Link κώδικα :
https://github.com/thpliakis/PDS_Exercise1.git

Σειριακός κώδικας V4:

Η σειριακή υλοποίηση για την εύρεση των τριγώνων που συμμετέχει κάθε κόμβος βασίζεται αρκετά στην δομή του CSC προτύπου. Ουσιαστικά ο αλγόριθμος κάνει 2 *for loops* για να βρει το διάνυσμα C3, ενώ ταυτόχρονα μπορεί να απαριθμεί τα τρίγωνα του γράφου μία φορά το καθένα. Ο αλγόριθμος ακολουθάει τα εξής βήματα:

1. Για κάθε κόμβο (1 for) βρες με ποιους άλλους κόμβους έχει ακμή (1 λίστα)
2. Για κάθε έναν από αυτούς (2 for) βρες τους κόμβους με τους οποίους συνδέεται (2 λίστα)
3. Σύγκρινε τις 2 λίστες και για κάθε κοινό στοιχείο που υπάρχει αύξησε των αριθμό των τριγώνων του κάθε κόμβου.

Παραληλοποίηση:

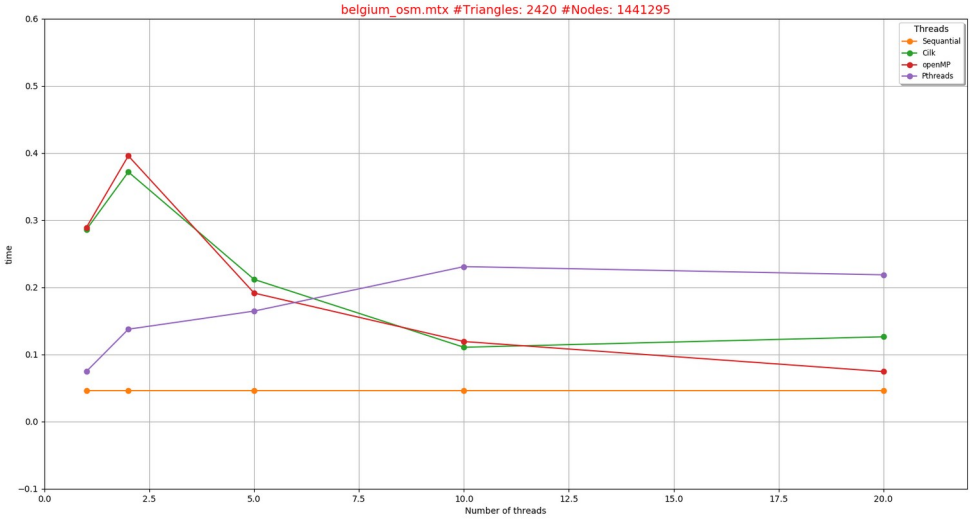
1. Με **phtreads** έγινε η παραλληλοποίση του πρώτου **for** ώστε να μετράνε παράλληλα τρίγωνα τα διάφορα νήματα, δηλαδή κάθε νήμα μετράει σε διαφορετική ομάδα κόμβων. Αρκετές μεταβλητές γίνανε global ώστε να έχουν πρόσβαση όλα τα νήματα. Φυσικά για τις μεταβλητές που τα νήματα γράφουν χρησιμοποιήθηκαν κλειδιά με τις mutex_lock, mutex_unlock. Επίσης όρισαμε μια stuct για να περνούνται τα κατάλληλα ορίσματα σε κάθε νέο νήμα ώστε να υπαρχει έλεγχος και καταγραφή τους.
2. Στην **Cilk** κάναμε το εσωτερικό του πρώτου for νέα συνάρτηση ώστε να χρησιμοποιήσουμε την εντολή *cilk_for στο εξωτερικό και* να δημιουργείται κάθε φορά καινούργιο νήμα για μια ομάδα κόμβων. Βάλαμε και τις *__cilkrts_end_cilk, __cilkrts_set_param* για να θέσουμε τον αριθμό νημάτων που θέλουμε κάθε φορά.
3. Ομοίως με την **Cilk** έγινε η παραλληλοποίηση στην **OpenMP** όπου χρησιμοποίησαμε τις εντολές ***#pragma omp parallel*** , ***#pragma omp for*** για την παραλληλοποίηση του πρώτου for.

Έλεγχος αποτελεσμάτων: Γράφοντας κατλάλληλα τον κώδικα τον δοκιμάσαμε στο έλεγχο ορθότητας στο elearning όπου πήραμε **Pass all tests!** αν και την μέτρηση του αριθμού των τριγώνων χρησιμοποιήσαμε την V3 και έαν πολύ απλό γράφο για να σιγουρευτούμε για τα σωστά αποτελέσματα.

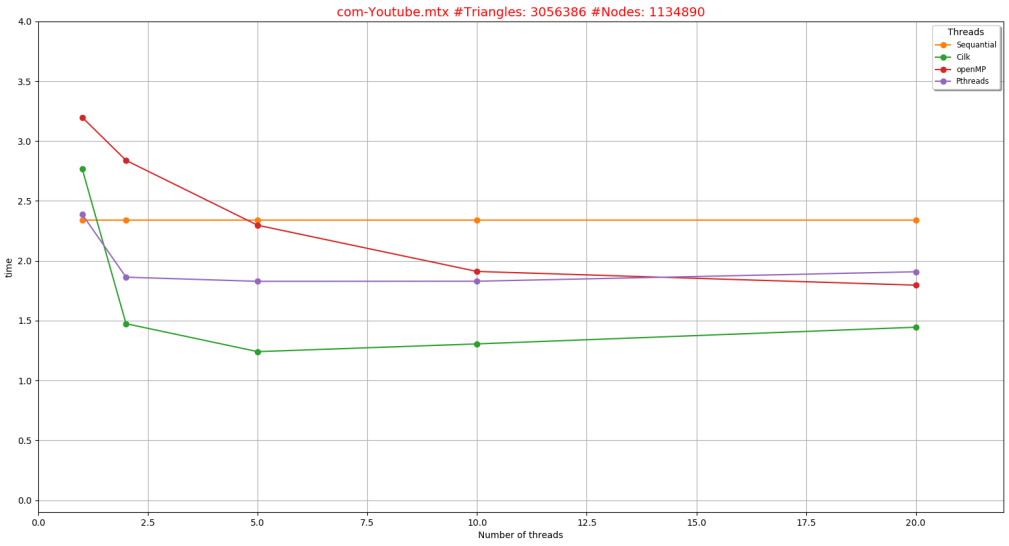
Διάβασμα γραφημάτων : Κάθε γράφημα δείχνει τους χρόνους, τα νηματά και την μέθοδο παραλληλοποίησης που χρησιμοποίθηκε σε κάθε γράφο. Ο κάθετος άξονας δείχνει τον χρόνο που χρειάστηκε το καμμάτι του κώδικα για την μέτρηση των τριγώνων. Ο οριζόντιος άξονας αναγράφει το πλήθος των νημάτων που έχουν δημιουργηθεί.

Παρατηρήσεις στα αποτελέσματα: Στην υλοποίηση με τα P**threads** παιζεί μεγάλο ρόλο η δομή του γράφου αφού σε ανάλογους γράφους σε αριθμό τριγώνων και σε αριθμό κόμβων μια είναι πιο γρήγορή από την σειριακή και μια πιο αργή. Η εξήγηση αυτού είναι ότι το overhead που δημιουργείται είναι ανάλογο της δομής του γράφου. Οι υλοποίησεις με C**il**k και **openMP** είναι τις πιο πολλές φορές πιο γρήγορες από την σειριακή εκτός εάν το πλήθος των κομβων είναι πολύ περισσότερο από αυτών των τριγώνων.

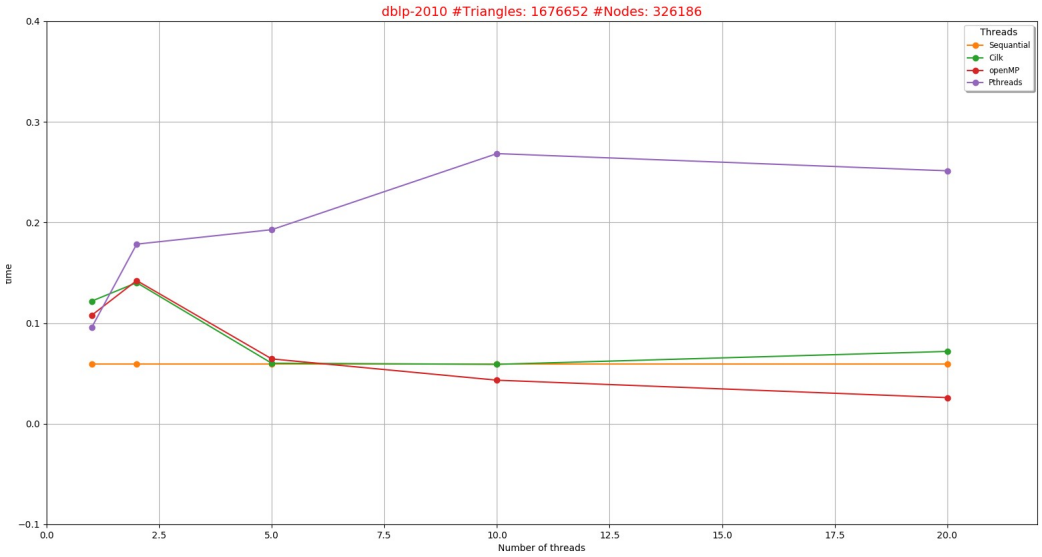
belgium_osm.mtx



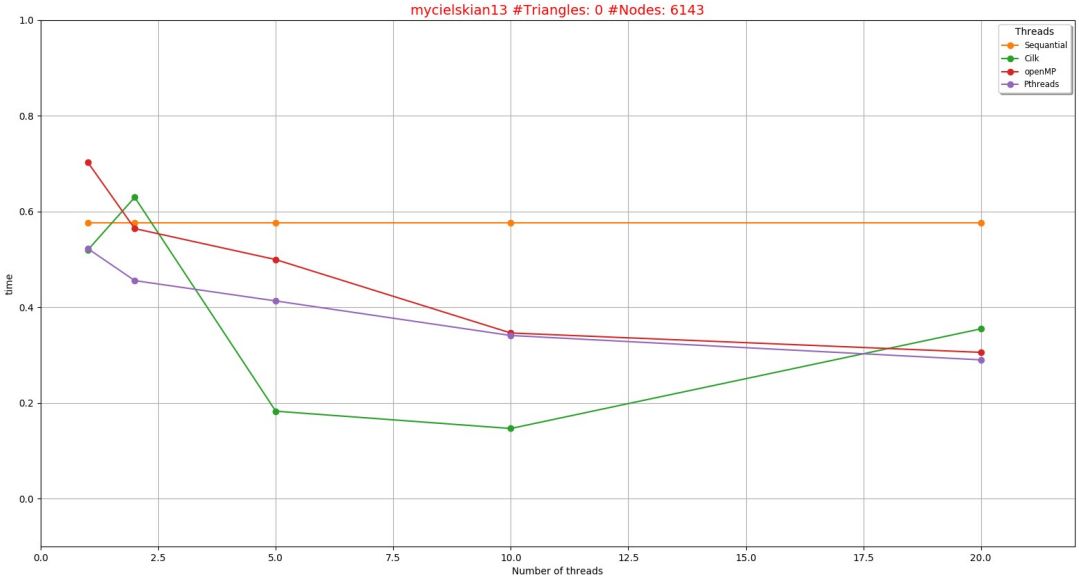
com-Youtube.mtx



dblp-2010.mtx



mycielskian13.mtx



NACA0015.mtx