Αναφορά

Πλήθος Ατόμων Ομάδας: 4
Ονοματεπώνυμο: Δημήτριος Κωστορρίζος A.M: 1054419
Ονοματεπώνυμο: Αλέξανδρος Δεληµιχάλης A.M: 1054324
Ονοματεπώνυμο: Δήμητρα Καλαματιανού A.M: 1054406
Ονοματεπώνυμο: Όμηρος Βασδάρης A.M: 1054429

Ερώτημα 1)

Οδηγίες του πρότυπου OpenMP, οι οποίες αξιοποιήθηκαν, για την υλοποίηση της “Static” προσέγγισης:

- parallel (Χρησιμοποιήθηκε για την δημιουργία τις παράλληλης περιοχής.)
- shared (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής epsilon ως κοινή μεταξύ των thread.)
- ordered (Χρησιμοποιήθηκε ώστε να τηρείται η σειρά αποθήκευσης των αποτελεσμάτων στο vector: SimplifiedAllPolylines, στο παράλληλο πρόγραμμα αντίστοιχα με το σειριακό.)
- private (Χρησιμοποιήθηκε για τον ορισμό των μεταβλητών polyline, simplified_polyline ως ιδιωτικές/ξεχωριστές για κάθε thread.)
- schedule(static) (Χρησιμοποιήθηκε για την επίτευξη του στατικού τρόπου ανάθεσης του φόρτου εργασίας για κάθε thread. Η απουσία δήλωσης τιμής για την μεταβλητή chunk_size, αναγκάζει το πρότυπο OpenMP, να αντιστοιχίσει προσεγγιστικά την default τιμή: Number_of_Loops/Number_of_Threads στην θέση του chunk_size για κάθε thread.)

Οι παρακάτω τιμές, έχουν υπολογισθεί , με epsilon = 0.0001.

i) Για O0 και αρχείο polylines_small.txt.

Num_Threads = 1	1.011397 sec			
Num_Threads = 2	1.057206 sec	1.057246 sec		
Num_Threads = 4	1.008158 sec	1.008171 sec	1.008135 sec	1.008177 sec

ii) Για O3 και αρχείο polylines_small.txt.

Num_Threads = 1	0.547000 sec			
Num_Threads = 2	0.557007 sec	0.557062 sec		
Num_Threads = 4	0.562126 sec	0.562103 sec	0.562152 sec	0.562150 sec

iii) Για O0 και αρχείο polylines_large.txt.

Num_Threads = 1	46.464517 sec			
Num_Threads = 2	47.086990 sec	47.086953 sec		
Num_Threads = 4	46.868617 sec	46.868625 sec	46.868621 sec	46.868622 sec

iii) Για O3 και αρχείο polylines_large.txt.

Num_Threads = 1	24.579368 sec			
Num_Threads = 2	24.796144 sec	24.796195 sec		
Num_Threads = 4	24.569978 sec	24.569974 sec	24.570015 sec	24.570022 sec

Το φαινόμενο, το οποίο παρατηρείται από δεδομένα χρονομέτρησης, είναι αφενός, ότι οι χρόνοι εκτέλεσης των νημάτων καθώς και του παράλληλου προγράμματος είναι παραπλήσιοι του χρόνου εκτέλεσης του σειριακού προγράμματος και αφετέρου, ότι κατά την συγκεκριμένη υλοποίηση, η χρονοβελτίωση που επιτυγχάνεται με την αλλαγή του αριθμού των threads, είναι μηδαμινή. Το συγκεκριμένο φαινόμενο ανάγεται στην απαίτηση επιπλέον χρόνου για την δημιουργία του παράλληλου περιβάλλοντος, τον διαμοιρασμό της εισόδου στα threads, κ.λ.π, εντός του παράλληλου προγράμματος, ενέργειες οι οποίες δεν συμβαίνουν κατά την εκτέλεση του σειριακού προγράμματος.

Ερώτημα 2)

Οδηγίες του πρότυπου OpenMP, οι οποίες αξιοποιήθηκαν, για την υλοποίηση της “Dynamic” προσέγγισης:

- parallel (Χρησιμοποιήθηκε για την δημιουργία τις παράλληλης περιοχής.)
- shared (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής epsilon ως κοινή μεταξύ των thread.)
- ordered (Χρησιμοποιήθηκε ώστε να τηρείται η σειρά αποθήκευσης των αποτελεσμάτων στο vector: SimplifiedAllPolylines, στο παράλληλο πρόγραμμα αντίστοιχα με το σειριακό.)
- private (Χρησιμοποιήθηκε για τον ορισμό των μεταβλητών polyline, simplified_polyline ως ιδιωτικές/ξεχωριστές για κάθε thread.)
- schedule(dynamic) (Χρησιμοποιήθηκε για την επίτευξη του στατικού τρόπου ανάθεσης του φόρτου εργασίας για κάθε thread. Η απουσία δήλωσης τιμής για την μεταβλητή chunk_size, αναγκάζει το πρότυπο OpenMP, να αντιστοιχίσει την default τιμή: 1 στην θέση του chunk_size για κάθε thread.)

Η παράμετρος chunk_size της εντολής schedule(dynamic, chunk_size) , πρέπει να καθοριστεί πειραματικά προκειμένου να επιτευχθεί ο βέλτιστος χρόνος εκτέλεσης της εφαρμογής, λαμβάνοντας υπόψη παραμέτρους όπως το μέγεθος της εισόδου, τον τρόπο εκτέλεσης του προγράμματος, την γλώσσα στην οποία είναι γραμμένο το πρόγραμμα, κ.λ.π.

Οι παρακάτω τιμές, έχουν υπολογισθεί , με epsilon = 0.0001.

i) Για O0 και αρχείο polylines_small.txt.

Num_Threads = 1	1.006752 sec			
Num_Threads = 2	0.560750 sec	0.560788 sec		
Num_Threads = 4	0.399954 sec	0.399990 sec	0.399941 sec	0.399992 sec

ii) Για O3 και αρχείο polylines_small.txt.

Num_Threads = 1	0.539063 sec			
Num_Threads = 2	0.305981 sec	0.306022 sec		
Num_Threads = 4	0.213756 sec	0.213756 sec	0.213660 sec	0.213720 sec

iii) Για O0 και αρχείο polylines_large.txt.

Num_Threads = 1	55.327546 sec			
Num_Threads = 2	29.319486 sec	29.319449 sec		
Num_Threads = 4	19.778994 sec	19.779035 sec	19.779033 sec	19.778994 sec

iii) Για O3 και αρχείο polylines_large.txt.

Num_Threads = 1	23.548604 sec			
Num_Threads = 2	14.479662 sec	14.479713 sec		
Num_Threads = 4	9.450758 sec	9.450795 sec	9.450794 sec	9.450749 sec

Οι χρόνοι εκτέλεσης των νημάτων είναι σημαντικά μειωμένοι σε σύγκριση με τους αντίστοιχους χρόνους του πρώτου ερωτήματος.

Ερώτημα 3)

Οδηγίες του πρότυπου OpenMP, οι οποίες αξιοποιήθηκαν, για την υλοποίηση της “Task1” προσέγγισης:

- parallel (Χρησιμοποιήθηκε για την δημιουργία τις παράλληλης περιοχής.)
- shared (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής epsilon ως κοινή μεταξύ των thread και για τον ορισμό των μεταβλητών epsilon, recResults1 ως κοινών μεταξύ των tasks.)
- private (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής simplified_polyline ως ιδιωτική/ξεχωριστή για κάθε thread.)
- firstprivate (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής polyline ως ιδιωτική/ξεχωριστή για κάθε thread, διατηρώντας ταυτόχρονα την τιμή που είχε πριν την είσοδο στην παράλληλη περιοχή.)
- single (Χρησιμοποιήθηκε ώστε να δηλωθεί ότι το συγκεκριμένο section, θα εκτελεστεί από ένα μόνο thread, όχι απαραίτητα το master thread.)
- nowait (Χρησιμοποιήθηκε ώστε να δηλωθεί ότι στο συγκεκριμένο section, θα αφαιρεθεί το barrier, καθώς, αφού τα thread θα εκτελούν τα tasks που δημιουργούνται, δεν πρέπει να περιμένουμε να φτάσουν όλα τα thread στο barrier. Με τον τρόπο αυτό εξοικονομούμε μερικά microseconds.)
- task (Χρησιμοποιήθηκε για την αναδρομική δημιουργία των tasks, τα οποία θα επεξεργάζονται το πρώτο μισό της προς διάσπαση πολυγραμμής)
- taskwait (Χρησιμοποιήθηκε ώστε το τρέχον task να αναμένει την ολοκλήρωση των task-παιδιών του, αναδρομικά, πριν συνεχίσει στην δημιουργία της απλοποιημένης πολυγραμμής.)

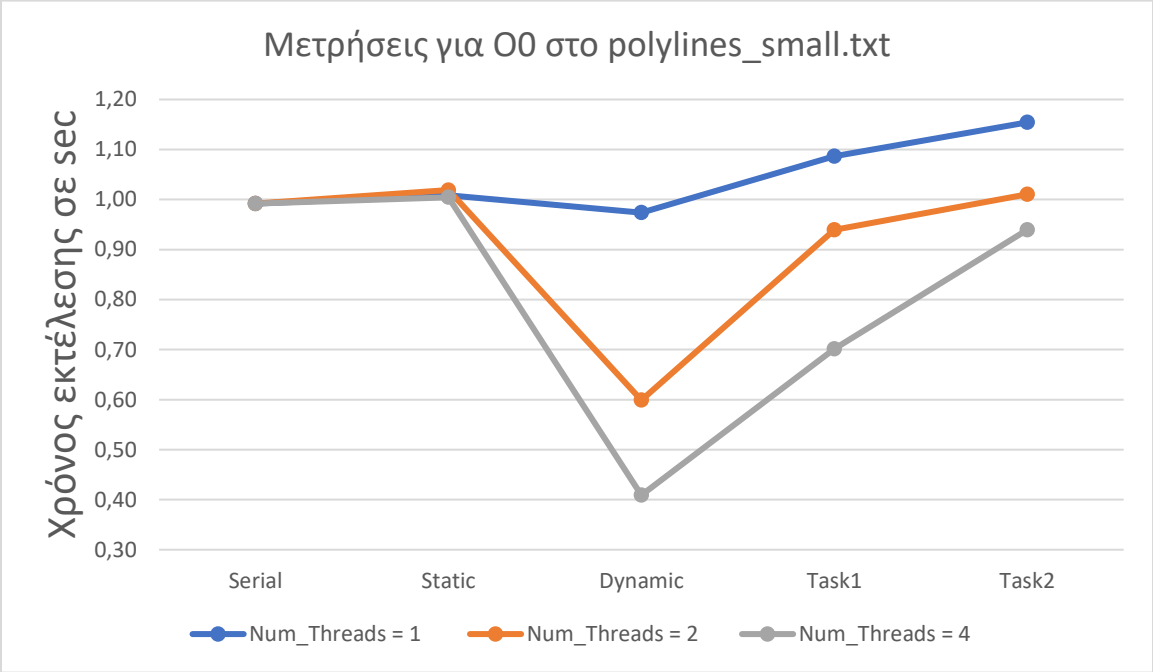
Ερώτημα 4)

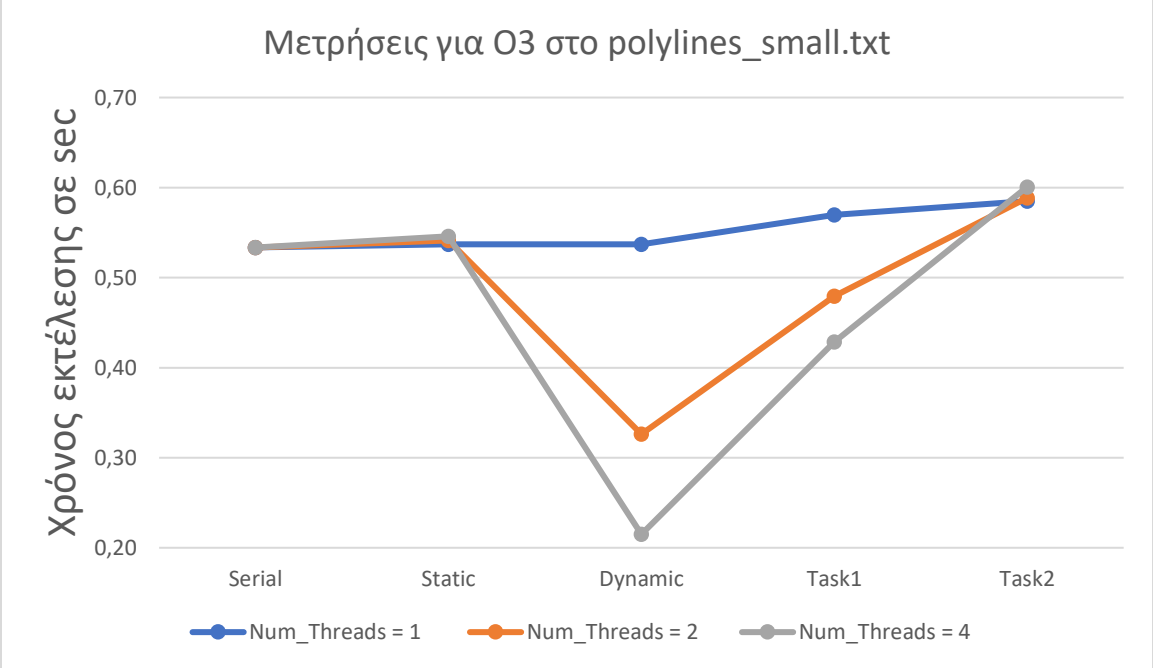
Οδηγίες του πρότυπου OpenMP, οι οποίες αξιοποιήθηκαν, για την υλοποίηση της “Task1” προσέγγισης:

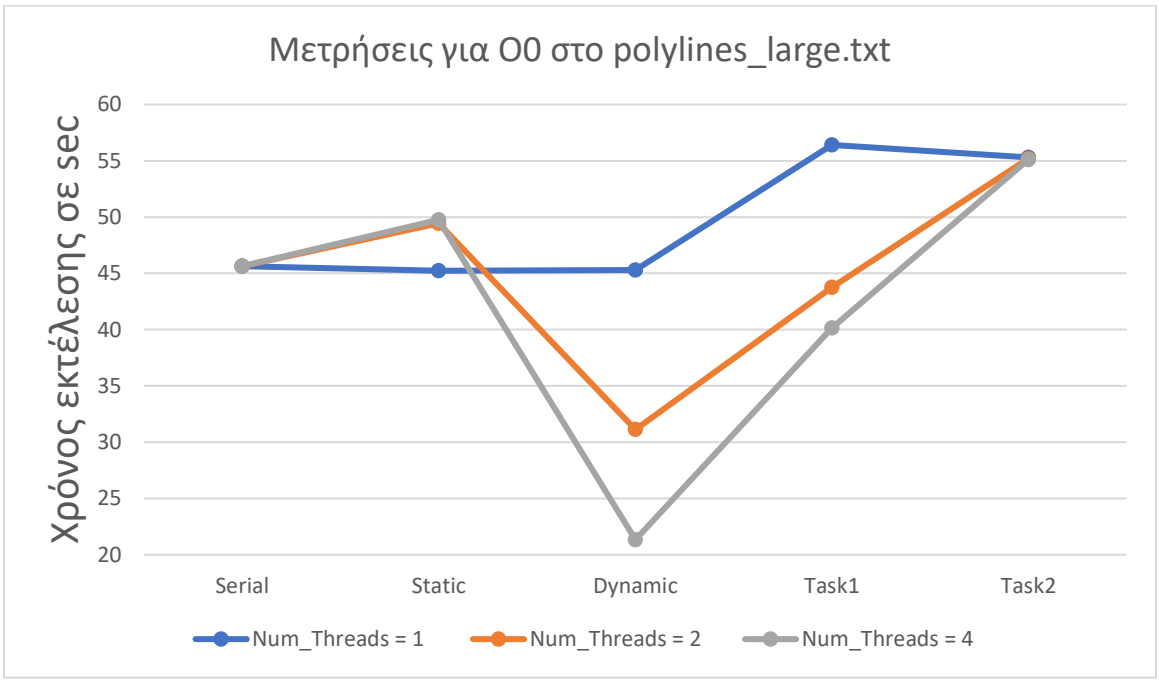
- parallel (Χρησιμοποιήθηκε για την δημιουργία τις παράλληλης περιοχής.)
- shared (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής epsilon ως κοινή μεταξύ των thread, για τον ορισμό των μεταβλητών epsilon, recResults1 ως κοινών μεταξύ των tasks που επεξεργάζονται την πρώτη πολυγραμμή και για τον ορισμό των μεταβλητών epsilon, recResults2 ως κοινών μεταξύ των tasks που επεξεργάζονται την δεύτερη πολυγραμμή.)
- private (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής simplified_polyline ως ιδιωτική/ξεχωριστή για κάθε thread.)
- firstprivate (Χρησιμοποιήθηκε για τον ορισμό της μεταβλητής polyline ως ιδιωτική/ξεχωριστή για κάθε thread, διατηρώντας ταυτόχρονα την τιμή που είχε πριν την είσοδο στην παράλληλη περιοχή.)
- single (Χρησιμοποιήθηκε ώστε να δηλωθεί ότι το συγκεκριμένο section, θα εκτελεστεί από ένα μόνο thread, όχι απαραίτητα το master thread.)
- nowait (Χρησιμοποιήθηκε ώστε να δηλωθεί ότι στο συγκεκριμένο section, θα αφαιρεθεί το barrier, καθώς, αφού τα thread θα εκτελούν τα tasks που δημιουργούνται, δεν πρέπει να περιμένουμε να φτάσουν όλα τα thread στο barrier. Με τον τρόπο αυτό εξοικονομούμε μερικά microseconds.)
- task (Χρησιμοποιήθηκε για την αναδρομική δημιουργία των tasks, τα οποία θα επεξεργάζονται τα δύο μισά της προς διάσπαση πολυγραμμής)
- taskwait (Χρησιμοποιήθηκε ώστε το τρέχον task να αναμένει την ολοκλήρωση των task-παιδιών του, αναδρομικά, πριν συνεχίσει στην δημιουργία της απλοποιημένης πολυγραμμής.)

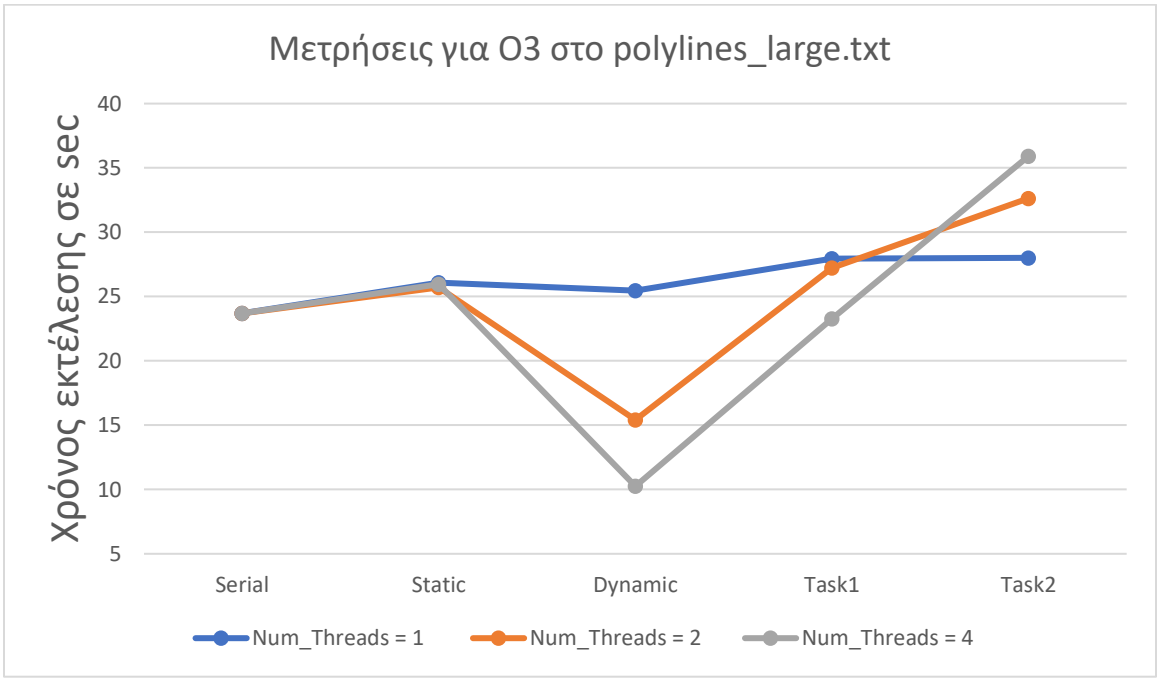
  Η προσέγγιση “task2”, στις περισσότερες περιπτώσεις εκτέλεσης, έχει παραπλήσιο χρόνο εκτέλεσης με την “static” προσέγγιση, αλλά μεγαλύτερο χρόνο εκτέλεσης από τις “task1” και “dynamic" προσεγγίσεις. Το φαινόμενο αυτό έγκειται στην δημιουργία μεγάλου αριθμού tasks, μερικά εκ των οποίων θα εκτελούνταν πιο γρήγορα ως σειριακά τμήματα κώδικα. Το φαινόμενο αυτό μπορεί να βελτιωθεί, εισάγοντας συνθήκες if(), οι οποίες θα ωθούν στην δημιουργία tasks, όταν ο προβλεπόμενος αριθμός αναδρομών κατά το σπάσιμο της πολυγραμμής στα δύο είναι μεγαλύτερος από ένα όριο, ενώ θα αποφεύγεται η δημιουργία tasks και θα εκτελείται σειριακά το συγκεκριμένο κομμάτι κώδικα, όταν ο αριθμός των αναδρομών είναι μικρότερος από το όριο αυτό.

Παρακάτω παραθέτονται τα διαγράμματα χρονοβελτιώσεων, για τα δύο αρχεία εισόδου, χωρίς βελτιστοποιήσεις/ με μέγιστες βελτιστοποιήσεις, για epsilon = 0.0001.






Μετρήσεις για O0 στο polylines_large.txt
Χρόνος εκτέλεσης σε sec
60
55
50
45
40
35
30
25
20
Serial
Static
Dynamic
Task1
Task2
Num_Threads = 1
Num_Threads = 2
Num_Threads = 4


Μετρήσεις για O3 στο polylines_large.txt
Χρόνος εκτέλεσης σε sec
40
35
30
25
20
15
10
5
Serial
Static
Dynamic
Task1
Task2
Num_Threads = 1
Num_Threads = 2
Num_Threads = 4