Εργασία 1
Παράλληλα και Διανεμημένα Συστήματα .
Υλοποίηση ενός Vantage point Tree στην C και παραλληλοποίηση του με την χρήση Pthreads, Cilk και OpenMP.

Μιχάλης Καρατζάς
AEM:9137
email:mikalaki@ece.auth.gr
mikalaki@it.auth.gr

Κυριάκος Μαραντίδης
AEM:9095
email: kyrmarkos@ece.auth.gr

Gitlab link για το πρότζεκτ :
https://gitlab.com/mikalaki/parallels-and-distributed-systems-exercise-1/
Dropbox link(το αρχείο που πέρασε από τον έλεγχο ορθότητας):
https://www.dropbox.com/s/katd2mx59bblfcs/code.tar.gz?dl=0

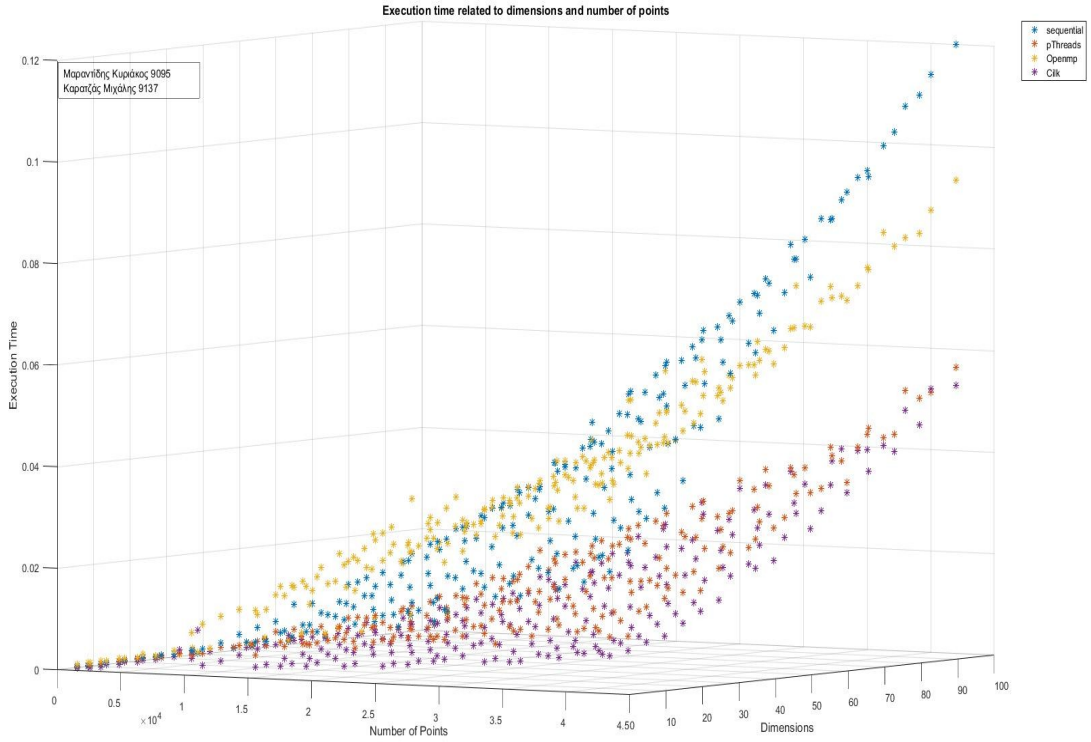
1. Σύντομη Περιγραφή της υλοποίησης μας.
   - Sequential: Στην Ακολουθιακή έκδοση της C , ακολουθήσαμε σε μεγάλο βαθμό της υλοποίηση που μας δόθηκε στο matlab. Χρησιμοποιήσαμε τις αρχές του δομημένου προγραμματισμού και σπάσαμε το πρόγραμμα μας σε επιμέρους συναρτήσεις.
   - Pthreads: Στην υλοποίηση με Pthreads ακολουθήσαμε τις αρχές που διέπουν την ακολουθιακή μας υλοποίηση ωστόσο παραλληλίσαμε τον υπολογισμό των αποστάσεων και το "χτίσιμο" των επιμέρους υπό-δέντρων. Ακολουθήσαμε την μέθοδο peer threads , καθώς το καλόν thread καλεί threads ώστε να δουλέψουν ,ωστόσο εκτελεί και αυτό μέρος της δουλειάς . Για τις αποστάσεις καθε thread γεμίζει τον πινακα των αποστάσεων εναλάξ. Για την περίπτωση των υπο-δέντρων , δίνουμε σε ένα thread τα δεδομένα, προκειμένου να υπολογίζει τις παράμέτρους του εξωτερικού υποδέντρου και να το χτίσει και το καλών thread (main thread) υπολογίζει τις παραμέτρους και χτίζει το εσωτερικό υποδεντρο. Για να κάνουμε το πρόγραμμα μας πιο γρήγορο ορίζουμε τιμή κατωφλιού (threshold) για τον αριθμό των σημείων τόσο για τις αποστάσεις όσο και για τα υπο-δέντρα ,κάτω από την οποία καλείται η ακολουθιακή έκδοση του κώδικα .
   - OpenMp: Για την υλοποίηση με openMp βάλαμε κατάλληλα annotations στον κώδικα της sequential έκδοσης μας , προκείμενου να παραλληλιστεί η for λούπα των ευκλείδειων αποστάσεων και στην συνέχεια για να παραλληλίσουμε την διαδικασία χτισίματος των υποδέντρων καλείται το χτίσιμο των επιμέρους υποδέντρων απο δύο διαφορετικά thread.
   - Cilk: Για την υλοποίηση με openMp βάλαμε κατάλληλα annotations στον κώδικα της sequential έκδοσης μας ,cilk_for , για την "λούπα" των ευκλείδειων αποστάσεων και για την "λούπα" της απόκτησης δεδομένων των υποδέντρων (αν και αυτό αποτελεί αρκετά ριψοκίνδυνη τεχνική,καθώς μοιράζονται και χρησιμοποιούνται από τις επαναλήψεις κοινές μεταβλητές, ωστόσο ακόμα και με αυτήν την τεχνική η cilk έκδοση μας, παρόλο που επιταχύνθηκε σε μεγάλο βαθμό , πέρασε τον validator tester και επαληθεύτηκε και με δικά μας τεστ σε μικρά datasets) και cilk spawn για τον παραλληλισμό της αναδρομικής κλήσης του χτισίματος των υποδέντρων .
   - *Για περισσότερες λεπτομέρειες σχετικά με τον κώδικα μπορείτε να δείτε τα σχόλια που εμπεριέχονται μέσα σε αυτόν.*

2. Γραφήματα, πίνακες και σχολιασμός των χρόνων εκτέλεσης :

Το μεγαλύτερο μέρος της υλοποίησης πραγματοποιήθηκε σε ένα λάπτοπ με επεξεργαστή i5-2410m(2 πυρήνες 4 threads) με 4gb ram και 5gb swap και λειτουργικό ubuntu linux 18.10. Το configuration και οι δοκιμές κατά την διαδικασία υλοποίησης έγιναν σε αυτόν τον υπολογιστή. Παρόλα αυτά για η συλλογή δεδομένων για την απόδοση του προγράμματος έγιναν σε Desktop με

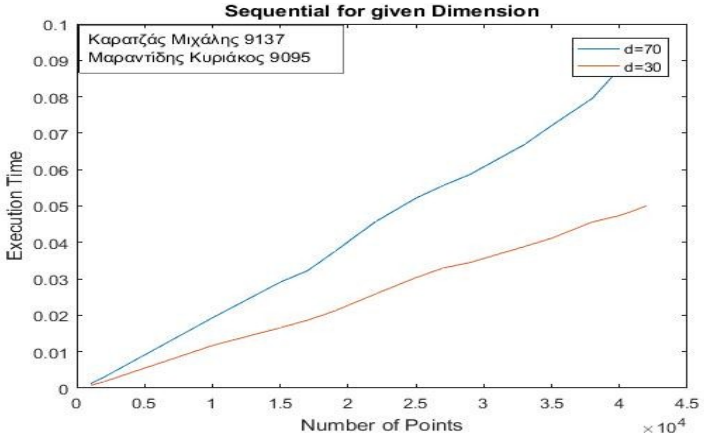
επεξεργαστή AMD Ryzen 7 2700X(8 πυρήνες 16 threads) με 16 gb ram και 2 gb swap και λειτουργικό ubuntu linux 19.04 με μικρές αλλαγές στα #define attributes των αρχείων παράλληλου κώδικα (π.χ.#define MAX_N_OF_ACTIVE_THREADS 4 → 16 στο vptree_pthreads.c). Παρακάτω βλέπουμε τα ανάλογα γραφήματα(τα δεδομένα αφορούν το Desktop που αναφέρθηκε):

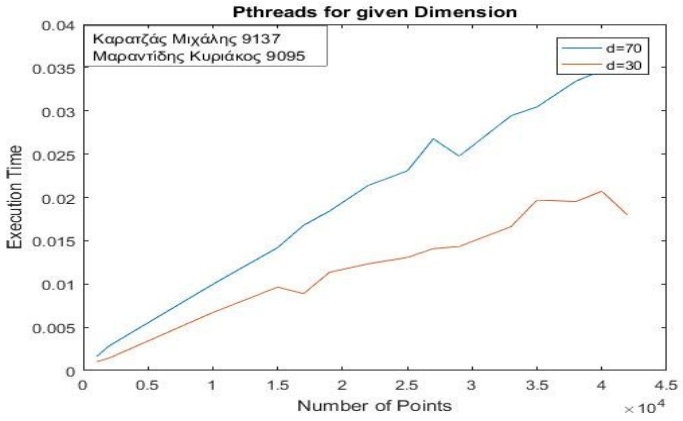
1. 3D γράφημα του χρόνου εκτέλεσης συναρτήσει του πλήθους των σημείων(* 10^4) και των διαστάσεων για κάθε υλοποίηση:

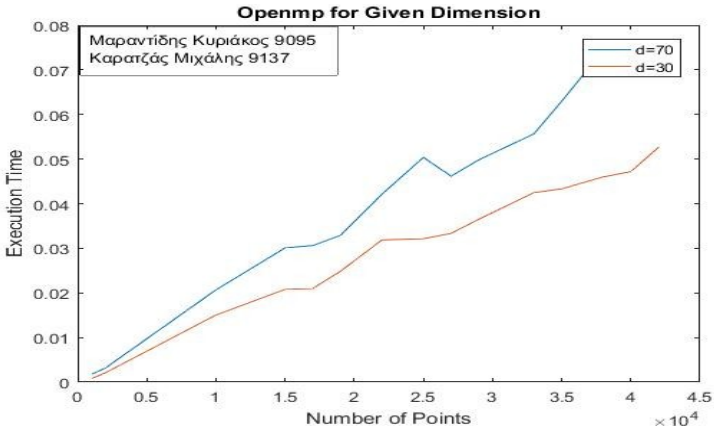


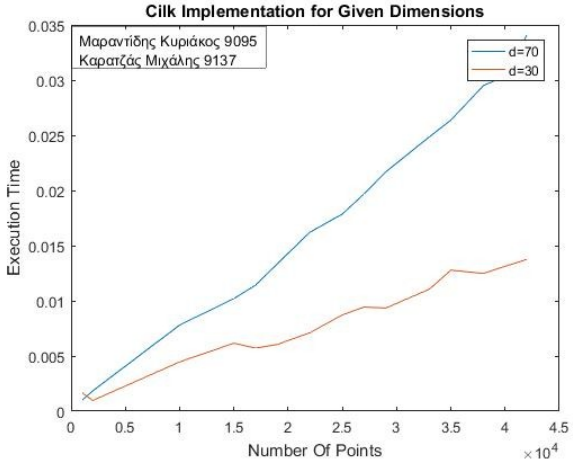
Στο παραπάνω γράφημα βλέπουμε ότι για μικρό πλήθος σημείων και μικρές διαστάσεις δεν έχουμε ιδιαίτερο όφελος από τον παραλληλισμό ,ωστόσο καθώς ο αριθμός των σημείων και των διαστάσεων αυξάνεται , έχουμε αυξανόμενο όφελος. Μας προβλημάτισε λίγο η υλοποίηση του openMP σε αυτό το γράφημα ωστόσο βλέπουμε ότι πάλι για μεγάλα σύνολα έχουμε σαφές προβάδισμα από το sequential, επίσης για τεστ που έγιναν στο λάπτοπ είδαμε ότι αξιοποιεί μονό <200% του cpu δηλαδή τα 2 από τα 4 thread , ενώ στο Desktop είδαμε ότι αξιοποιεί μαξιμουμ 3-4 threads. Ιδιαίτερο ενδιαφέρον παρουσιάζει και η υλοποίηση σε cilk καθώς είναι πολύ γρηγορότερη από κάθε άλλη για μεγάλα σύνολα ,αλλά είναι αρκετά γρήγορη ακόμα και για μικρά σύνολα (σε αυτά συνέβαλε η τεχνική που αναφέρθηκε παραπάνω στην ενότητα 1).

2. Γραφήματα χρόνου εκτέλεσης ,δεδομένων διαστάσεων.









3. Πίνακες με μετρήσεις για σταθερό αριθμό σημείων.

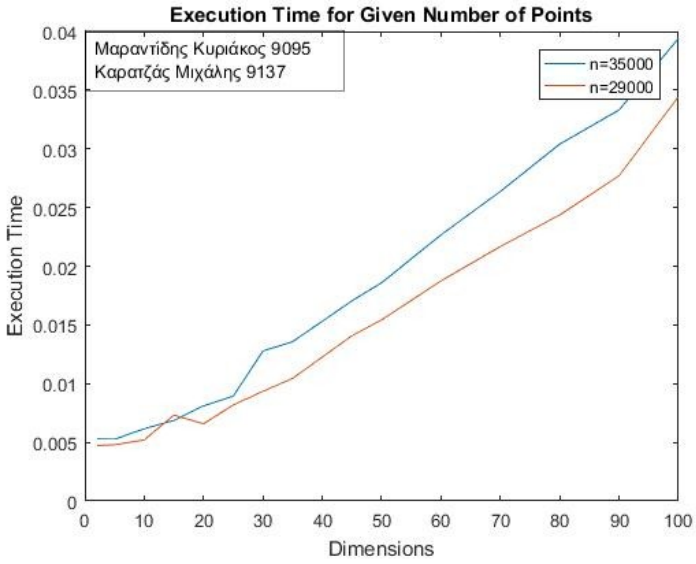
Sequential for Given Number Of Points		
Dimensions	Execution Time N=35000	Execution Time N=19000
100	0,094359	0,048587
90	0,085427	0,045379
80	0,078445	0,041157
70	0,072105	0,037473
60	0,064169	0,032934
50	0,055943	0,028497
45	0,052246	0,026767
35	0,039464	0,021626
30	0,041176	0,021183
25	0,032756	0,016892
20	0,028538	0,015137
15	0,025525	0,013251
10	0,023073	0,011778
5	0,018039	0,009089
2	0,020607	0,00934

Pthreads for Given Number of points		
Dimensions	Execution Time N=35000	Execution Time N=19000
100	0,042469	0,022801
90	0,040647	0,021107
80	0,036113	0,018697
70	0,03046	0,018423
60	0,026835	0,0152
50	0,026262	0,013035
45	0,020327	0,012885
35	0,018621	0,010068
30	0,019702	0,011354
25	0,013872	0,008476
20	0,013295	0,008853
15	0,012751	0,006507
10	0,014466	0,007637
5	0,011485	0,006965
2	0,012184	0,006962

Openmp for given Number Of Points		
Dimensions	Execution Time N=35000	Execution Time N=19000
100	0,075234	0,038995
90	0,072289	0,041245
80	0,064934	0,036219
70	0,062961	0,032906
60	0,053814	0,033274
50	0,050849	0,027018
45	0,051097	0,027983
35	0,04375	0,025317
30	0,043328	0,024844
25	0,042676	0,023251
20	0,036359	0,020333
15	0,043812	0,023147
10	0,030997	0,018832
5	0,030065	0,019971
2	0,037129	0,018435

Cilk for Given Number of Points		
Dimensions	Execution Time N=35000	Execution Time N=19000
100	0,003843	0,002325
90	0,003175	0,002029
80	0,002985	0,001725
70	0,002766	0,001626
60	0,002815	0,001424
50	0,002313	0,001382
45	0,002173	0,001298
35	0,001631	0,001157
30	0,001618	0,001092
25	0,001249	0,000956
20	0,001348	0,000796
15	0,001302	0,000885
10	0,001021	0,000701
5	0,000992	0,000722
2	0,000936	0,000684

CILK


Execution Time for Given Number of Points
Μαραντίδης Κυριάκος 9095
Καρατζάς Μιχάλης 9137
n=35000
n=29000
Execution Time
Dimensions
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0
10
20
30
40
50
60
70
80
90
100