Παράλληλος Προγραμματισμός σε Συστήματα Μηχανικής Μάθησης

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Εργαστήριο 2:

Ο αλγόριθμος των κ-μέσων "K-MEANS" με openMP

Δασούλας Ιωάννης – 1053711 – 5ο Έτος

Εισαγωγή

Η αναφορά αφορά την υλοποίηση του αλγορίθμου K-means χρησιμοποιώντας την OpenMP. Για κάθε εργασία έχει δημιουργηθεί ένα ξεχωριστό αρχείο c (par_kmeans_a.c για την Εργασία 1, par_kmeans_b.c για την Εργασία 2 και ούτω καθ' εξής). Αναλυτικά εξηγήσεις και σχόλια για τα βήματα που ακολουθούνται υπάρχουν σε κάθε αρχείο. Για να υπάρχει ακέραια σύγκριση μεταξύ των χρόνων, ο χρόνος κάθε προγράμματος μετρήθηκε για τις 16 επαναλήψεις, όπως δηλαδή και στο πρώτο εργαστήριο.

Εργασία 1

Για την πρώτη εργασία χρησιμοποιήθηκε η σειριακή έκδοση από το προηγούμενο εργαστήριο της άσκησης με τις ίδιες παραμέτρους, δηλαδή N = 100.000, Nv = 1000, Nc = 100.

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243261520.000000
132325432.000000
132211760.000000
132158752.000000
132127800.000000
132107888.000000
132093792.000000
132083080.000000
132074888.000000
132069752.000000
132065528.000000
132061304.000000
132058384.000000
132055984.000000
132053584.000000
132051696.000000
Iterations:16
Algorithm terminated succesfully!

real    11m45,459s
user    11m44,951s
sys     0m0,264s
```

Εικόνα 1: Χρόνος για Εργασία 1

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.29    686.78   686.78 161600000     0.00     0.00  EuclidianDistance
  0.97    693.49     6.71       16     0.42     0.42  EstimateCenters
  0.21    694.93     1.44       16     0.09    43.01  Classification
  0.12    695.78     0.84        1     0.84     0.84  CreateData
  0.00    695.78     0.00       16     0.00     0.00  Terminate
  0.00    695.78     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 2: Αποτελέσματα profiler για Εργασία 1

Παρατηρείται πως σχεδόν όλος ο χρόνος είναι στον υπολογισμό της ευκλείδειας απόστασης των σημείων από τα κέντρα.

Εργασία 2

Βάσει των αποτελεσμάτων του profiler, η πρώτη προσπάθεια ήταν η παραλληλοποίηση της συνάρτησης υπολογισμού της ευκλείδειας απόστασης. Όταν έγινε αυτό, όμως, παρατηρήθηκε ότι ενώ τα αποτελέσματα ήταν σωστά, δεν υπήρχει κέρδος στον χρόνο. Αυτό συνέβαινε επειδή αυτή η συνάρτηση καλείται πάρα πολλές φορές κατά την διάρκεια του αλγορίθμου (συγκεκριμένα 161.600.000 φορές για τις 16 επαναλήψεις) και κάθε φορά η ρύθμιση της OpenMP παίρνει λίγο χρόνο, με αποτέλεσμα να μην μειώνεται ο χρόνος.

Μετά από αυτό, επιχειρήθηκε η χρήση της OpenMP σε όλο το classification (συνάρτηση Classification()) ώστε να μην χρειάζεται πολλές φορές η ρύθμισή της (μόνο 16 για 16 επαναλήψεις) αλλά και να έχει αποτέλεσμα και στις επαναλήψεις για τον υπολογισμό της ευκλείδειας απόστασης, αφού είναι εμφωλευμένες στην ευρύτερη συνάρτηση κατηγοριοποίησης. Έτσι, χωρίς αλλαγή του σειριακού κώδικα η συνάρτηση έγινε ως εξής:

```
void Classification(void){                              //Αντιστοίχιση δεδομένων με το κοντινότερο κέντρο

    int i, j, k, min_index;
    float min_dist,dist;
    new_dist_sum = 0.0;                                 //Άθροισμα των ελάχιστων αποστάσεων

    #pragma omp parallel for\
             private(i,j,dist,min_dist,min_index)\
             shared(classes,center,vec)\
             reduction(+: new_dist_sum)                 //Omp statement για παραλληλοποίηση
             |
    for(i=0; i<N; i++){                                 //Σάρωση για όλα τα δεδομένα
        min_dist = EuclidianDistance(vec[i],center[0]); //Εύρεση μίας πρώτης απόστασης με το κέντρο 0
        min_index = 0;                                  //Αποθήκευση του index

        for(j=0; j<Nc; j++){                            //Σύγκριση με την απόσταση για κάθε κέντρο
            dist = EuclidianDistance(vec[i], center[j]);
            if(dist<min_dist){                          //Αν βρεθεί μικρότερη απόσταση αποθηκεύεται...
                min_dist=dist;                          //...το μέγεθός της...
                min_index=j;                            //...και το index της
            }
        }
        new_dist_sum += min_dist;                       //Η ελάχιστη απόσταση από κέντρο προστίθεται στην μεταβλητή αυτή...
                                                        //...που χρησιμοποιείται για τον έλεγχο σύγκλισης
        classes[i] = min_index;                         //Η κλάση του σημείου είναι το index του κοντινότερου κέντρου
    }

}
```

Εικόνα 3: Συνάρτηση κατηγοριοποίησης με OpenMP

Στην ρύθμιση της OpenMP ορίστηκαν ως private τα i, j που είναι οι μετρητές των επαναλήψεων ώστε κάθε νήμα να παίρνει από ένα. Αντίστοιχα ορίστηκαν ως private οι αποστάσεις (dist), οι ελάχιστες αποστάσεις (min_dist) και οι τα κέντρα με την ελάχιστη απόσταση (min_index), έτσι ώστε από κάθε νήμα κάθε φορά να προκύπτει μία ελάχιστη απόσταση που θα προστίθεται στην αθροιστική ελάχιστη απόσταση (new_dist_sum) που έχει οριστεί με reduction ώστε όλα τα νήματα να γράφουν σε αυτή

και τα αποτελέσματά τους να προστίθενται. Ως shared ορίστηκαν οι αρχικοί πίνακες των διανυσμάτων, των κέντρων και των κλάσεων.

Αφού διαπιστώθηκε η σωστή λειτουργία, ακολούθησε η χρήση της OpenMP στην συνάρτηση υπολογισμού των νέων κέντρων. Ξανά χωρίς κάποια αλλαγή στον σειριακό κώδικα, η συνάρτηση διαμορφώθηκε ως εξής:

```
void EstimateCenters(void){

    memset(center, 0, sizeof(center));
    int i,j,k,class_members = 0;
    float f;

    #pragma omp parallel for\
            private(i,j,k,class_members)\
            shared(vec,center)

    for(i=0; i<Nc; i++){
        for(j=0; j<N; j++){
            if(classes[j]==i){
                class_members++;
                for(k=0; k<Nv; k++){
                    center[i][k] += vec[j][k];
                }
            }
        }

        f = 1.0/class_members;
        for(j=0; j<Nv; j++){
            center[i][j] *= f;
        }
        class_members = 0;
    }

}
```

Εικόνα 4: Συνάρτηση υπολογισμού κέντρων με OpenMP

Ξανά ορίζονται οι μετρητές επαναλήψεων (i, j, k) ως private ώστε κάθε νήμα να κάνει μία πλήρη επανάληψη, καθώς και η μεταβλητή class_members που μετράει πόσα διανύσματα ανήκουν στην κατηγορία που εξετάζεται κάθε φορά. Ως shared ορίζονται

οι αρχικοί πίνακες που είναι κοινοί για όλα τα νήματα, δηλαδή ο πίνακας με τα κέντρα και ο πίνακας με τα διανύσματα.

Για τις συναρτήσεις αρχικοποίησης των δεδομένων δεν έγινε χρήση της OpenMP καθώς είναι πολύ σύντομες σε σχέση με τους υπόλοιπους υπολογισμούς και γίνονται μόνο μία φορά.

Η σωστή λειτουργία φαίνεται από τα αποτελέσματα. Για παράδειγμα, για 2 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 2
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243394080.000000
132335352.000000
132219312.000000
132165440.000000
132135376.000000
132116800.000000
132103424.000000
132093800.000000
132086816.000000
132080960.000000
132076752.000000
132073072.000000
132070200.000000
132067664.000000
132065904.000000
132064128.000000

Επαναλήψεις:16

real    5m48,979s
user    11m33,763s
sys     0m0,288s
```

Εικόνα 5: Αποτελέσματα Εργασίας 2 με 2 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.13    663.79   663.79 161275348     0.00     0.00  EuclidianDistance
  1.28    672.38     8.60       16     0.54    42.02  Terminate
  0.12    673.16     0.77        1     0.77     0.77  CreateData
  0.00    673.18     0.03       16     0.00     0.00  Classification
  0.00    673.20     0.02       16     0.00     0.00  EstimateCenters
  0.00    673.20     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 6: Αποτελέσματα profiler Εργασίας 2 για 2 νήματα

Αντίστοιχα, για χρήση 4 νημάτων:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 4
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243398544.000000
132339944.000000
132223328.000000
132170360.000000
132142432.000000
132120448.000000
132106944.000000
132097384.000000
132090304.000000
132084744.000000
132080384.000000
132076944.000000
132074256.000000
132071664.000000
132069872.000000
132068368.000000

Επαναλήψεις:16

real    2m57,071s
user    11m41,649s
sys     0m0,249s
```

Εικόνα 7: Αποτελέσματα Εργασίας 2 με 4 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.15    488.31   488.31 158462560     0.00     0.00  EuclidianDistance
  1.23    494.35     6.04       16     0.38    30.90  Terminate
  0.16    495.13     0.77        1     0.77     0.77  CreateData
  0.00    495.14     0.01       16     0.00     0.00  EstimateCenters
  0.00    495.14     0.01       16     0.00     0.00  Classification
  0.00    495.14     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 8: Αποτελέσματα profiler Εργασίας 2 για 4 νήματα

Τέλος, για 8 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 8
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
244161232.000000
132375952.000000
132258720.000000
132206360.000000
132170648.000000
132149296.000000
132134480.000000
132124128.000000
132116096.000000
132110176.000000
132105392.000000
132101904.000000
132098968.000000
132096400.000000
132094320.000000
132092504.000000

Επαναλήψεις:16

real    1m31,186s
user    11m48,478s
sys     0m0,208s
```

Εικόνα 9: Αποτελέσματα Εργασίας 2 με 8 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 98.57    228.94   228.94 102352761     0.00     0.00  EuclidianDistance
  1.58    232.60     3.66       16     0.23    14.54  Terminate
  0.38    233.47     0.87        1     0.87     0.87  CreateData
  0.01    233.49     0.02       16     0.00     0.00  EstimateCenters
  0.00    233.49     0.01       16     0.00     0.00  Classification
  0.00    233.49     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 10: Αποτελέσματα profiler Εργασίας 2 για 8 νήματα

Στα αποτελέσματα γίνεται εμφανές ότι οι αθροιστικές ελάχιστες αποστάσεις είναι όλο και πιο κοντά μεταξύ τους καθώς τρέχει ο αλγόριθμος, που σημαίνει ότι ο αλγόριθμος συγκλίνει, κάτι που επαληθεύεται αν βγει το όριο των 16 επαναλήψεων και ο αλγόριθμος ολοκληρωθεί. Επίσης, γίνεται εμφανής η αντίστροφη αναλογία νημάτων και χρόνων εκτέλεσης που δείχνει ότι έγινε επιτυχώς η παραλληλοποίηση. Τα αποτελέσματα είναι συγκεντρωμένα στον παρακάτω πίνακα.

Σειριακό	2 Νήματα	4 Νήματα	8 Νήματα
11m45s	5m48s	2m57s	1m31s

Εργασία 3

Στην εργασία 3 προστέθηκαν δηλώσεις schedule(static) στις δηλώσεις openmp. Στον συγκεκριμένο αλγόριθμο δεν υπήρχε ανάγκη για δήλωση schedule(dynamic) μιας και οι επαναλήψεις των βρόγχων έχουν περίπου ίδια διάρκεια όλες. Με τη δήλωση αυτή δεν παρατηρήθηκε μεγάλο κέρδος χρόνου. Οι καλύτεροι χρόνοι επιτεύχθηκαν για δήλωση schedule(static,1), ενώ όταν επιλέγονταν μεγαλύτερες παράμετροι υπήρχε συχνά χειρότερη επίδοση από αυτή της εργασίας 2. Τα αποτελέσματα επαληθεύουν τη σωστή λειτουργία.

Αρχικά, με 2 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 2
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
244087968.000000
132314864.000000
132202080.000000
132148064.000000
132117704.000000
132097760.000000
132083904.000000
132073792.000000
132066728.000000
132061096.000000
132056912.000000
132053136.000000
132049712.000000
132047456.000000
132045784.000000
132043728.000000

Επαναλήψεις:16

real    5m39,600s
user    11m16,494s
sys     0m0,181s
```

Εικόνα 11: Αποτελέσματα Εργασίας 3 με 2 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.24    646.94   646.94 161188128     0.00     0.00  EuclidianDistance
  1.14    654.35     7.41       16     0.46    40.90  Terminate
  0.11    655.06     0.71        1     0.71     0.71  CreateData
  0.01    655.11     0.05       16     0.00     0.00  Classification
  0.00    655.11     0.00       16     0.00     0.00  EstimateCenters
  0.00    655.11     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 12: Αποτελέσματα profiler Εργασίας 3 για 2 νήματα

Αντίστοιχα, για 4 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 4
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243617360.000000
132318280.000000
132203536.000000
132148704.000000
132117680.000000
132098152.000000
132084032.000000
132074352.000000
132067440.000000
132062160.000000
132057560.000000
132056184.000000
132050544.000000
132047856.000000
132045664.000000
132044352.000000

Επαναλήψεις:16

real    2m57,679s
user    11m42,700s
sys     0m0,188s
```

Εικόνα 13: Αποτελέσματα Εργασίας 3 με 4 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.21    487.20   487.20 152508727     0.00     0.00  EuclidianDistance
  1.10    492.60     5.40       16     0.34    30.79  Terminate
  0.17    493.44     0.84        1     0.84     0.84  CreateData
  0.01    493.48     0.04       16     0.00     0.00  EstimateCenters
  0.00    493.50     0.02       16     0.00     0.00  Classification
  0.00    493.50     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 14: Αποτελέσματα profiler Εργασίας 3 για 4 νήματα

Τέλος, για 8 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 8
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243757920.000000
132362856.000000
132244912.000000
132188888.000000
132156440.000000
132135800.000000
132122040.000000
132111800.000000
132103952.000000
132097672.000000
132092896.000000
132088976.000000
132085472.000000
132082752.000000
132080408.000000
132078536.000000

Επαναλήψεις:16

real    1m31,152s
user    11m49,037s
sys     0m0,256s
```

Εικόνα 15: Αποτελέσματα Εργασίας 3 με 8 νήματα

- Τα αποτελέσματα του profiler:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 98.73    228.76   228.76 105131722     0.00     0.00  EuclidianDistance
  1.43    232.07     3.32       16     0.21    14.50  Terminate
  0.33    232.84     0.76        1     0.76     0.76  CreateData
  0.00    232.84     0.01       16     0.00     0.00  EstimateCenters
  0.00    232.84     0.00       16     0.00     0.00  Classification
  0.00    232.84     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 16: Αποτελέσματα profiler Εργασίας 3 για 8 νήματα

Στα αποτελέσματα γίνεται ξανά εμφανές ότι οι αθροιστικές ελάχιστες αποστάσεις είναι όλο και πιο κοντά μεταξύ τους καθώς τρέχει ο αλγόριθμος. Επίσης, γίνεται εμφανής η αντίστροφη αναλογία νημάτων και χρόνων εκτέλεσης που δείχνει ότι έγινε επιτυχώς η παραλληλοποίηση. Τα αποτελέσματα είναι συγκεντρωμένα στον παρακάτω πίνακα και δείχνουν ότι ουσιαστικά δεν υπήρχε βελτίωση με τη δήλωση schedule().

Σειριακό	2 Νήματα	4 Νήματα	8 Νήματα
11m45s	5m37s	2m57s	1m31s

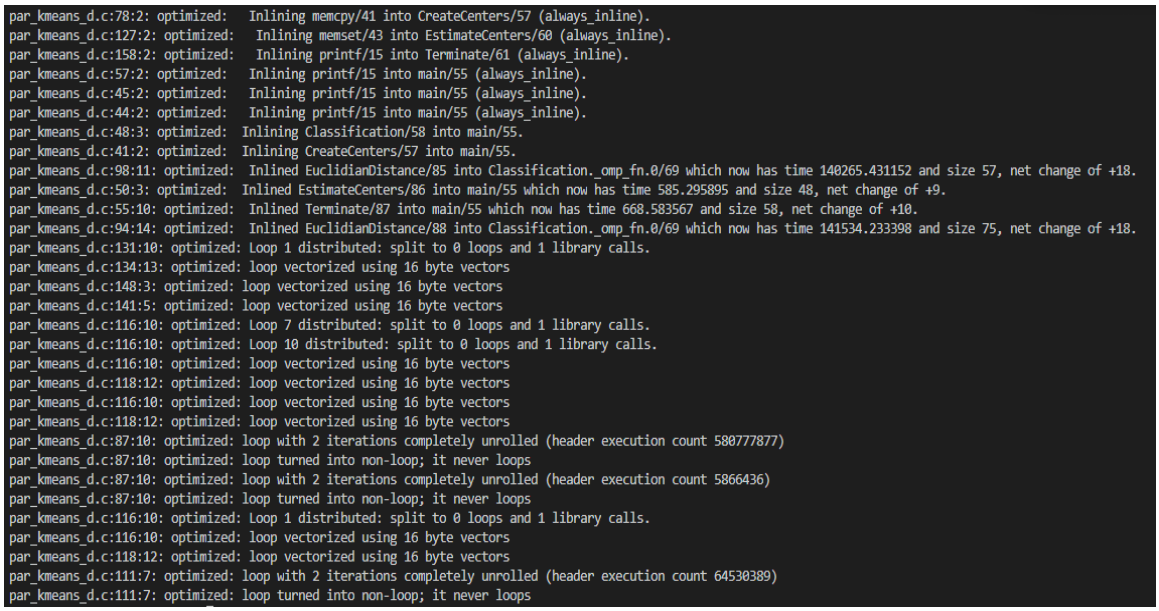
Εργασία 4

Στην εργασία 4 προστέθηκε η δήλωση simd μέσω της οποίας επιτρέπεται πολλαπλές επαναλήψεις των βρόγχων να γίνονται ταυτόχρονα μέσω SIMD εντολών. Αυτό έγινε στην συνάρτηση EuclidianDistance() που είναι η πιο χρονοβόρα και στην οποία επιτρέπεται να χρησιμοποιηθεί η simd εντολή μιας περιέχει έναν βρόγχο χωρίς άλλους εμφωλευμένους με σύνθετες εντολές μεταξύ τους, όπως γίνεται στην συνάρτηση Classification() και EstimateCenters(). Σε αυτές τις συναρτήσεις δεν θα είχε νόημα η χρήση της simd καθώς περιέχουν εντολές ανάμεσα στους εμφωλευμένους βρόγχους. Πλέον η συνάρτηση υπολογισμού της ευκλείδειας απόστασης διαμορφώθηκε ως εξής χωρίς αλλαγή στον κώδικα:

```
float EuclidianDistance(float * v, float * c){

    float dis=0.0, temp;
    int i;

    #pragma omp simd reduction(+: dis) private(temp)
    for(i=0; i<Nv; i++){
        temp = (v[i] - c[i]);
        dis += temp*temp;
    }

    return dis;
}
```

Εικόνα 17: Χρήση simd στον υπολογισμό απόστασης

Η μεταβλητή temp δηλώνεται ως private αφού είναι ξεχωριστή για κάθε νήμα, αλλά όλα τα νήματα γράφουν στην μεταβλητή dis που περιέχει την τελική απόσταση και δηλώνεται με reduction. Αυτό είναι σωστό αλγοριθμικά για τον συγκεκριμένο αλγόριθμο μιας και οι υπολογισμοί των επαναλήψεων δεν αλληλεξαρτούνται. Με την βελτιστοποίηση -O3, οι βελτιστοποιήσεις ήταν οι εξής:



Εικόνα 18: Βελτιστοποίησεις στην Εργασία 4 με χρήση simd και δήλωση -O3

Συγκεκριμένα, έγιναν inlined η αρχικοποίηση των κέντρων με την συνάρτηση memcpy(), η συνάρτηση EuclidianDistance() στην συνάρτηση Classification() από την οποία καλείται, η συνάρτηση memset() που μηδενίζει τον πίνακα των κέντρων κάθε φορά που πρέπει να υπολογιστούν, οι συναρτήσεις printf() που καλούνται για την επίδειξη των αποτελεσμάτων και τέλος οι συναρτήσεις Classification(), CreateCenters(), EstimateCenters() και Terminate() στην συνάρτηση main() από την οποία καλούνται.

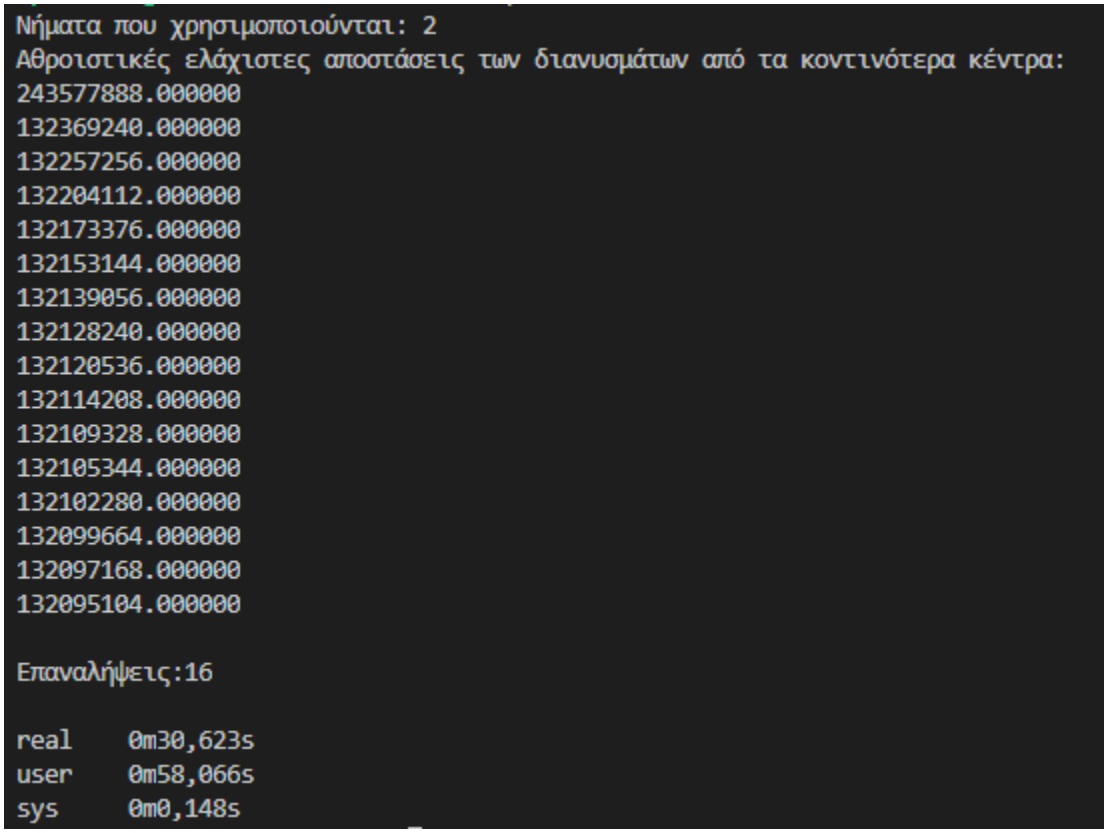
Επίσης έγινε loop distribution (διαχωρισμός παραλληλοποιήσιμων και μη παραλληλοποιήσιμων statements σε διαφορετικούς βρόγχους) για τα 3 #pragma ώστε να ξεχωριστούν τα κομμάτια κώδικα που μπορούν να παραλληλοποιηθούν.

Τέλος, έγινε vectorization και unrolling των βρόγχων στις συναρτήσεις Classification(), και EstimateCenters() συμπεριλαμβανομένης και της EuclidianDistance() που καλείται από την Classification.

Τα αποτελέσματα επιβεβαιώνουν τη σωστή λειτουργία του αλγορίθμου με μεγάλη βελτίωση χρόνου. Ο έλεγχος του profiler εδώ δεν ήταν δυνατός διότι εμφάνιζε χρόνο frame dummy λόγω της -O3 βελτιστοποίησης.

Για 2 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:



Εικόνα 19: Αποτελέσματα Εργασίας 4 με 2 νήματα

Αντίστοιχα, για 4 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 4
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243368704.000000
132355960.000000
132241408.000000
132186896.000000
132155416.000000
132134992.000000
132120840.000000
132110896.000000
132103552.000000
132098008.000000
132093392.000000
132089712.000000
132086496.000000
132083952.000000
132082200.000000
132080280.000000

Επαναλήψεις:16

real    0m18,966s
user    0m58,272s
sys     0m0,164s
```

Εικόνα 20: Αποτελέσματα Εργασίας 4 με 4 νήματα

Τέλος, για 8 νήματα:

- Για τις 16 επαναλήψεις, τα αποτελέσματα ήταν τα εξής:

```
Νήματα που χρησιμοποιούνται: 8
Αθροιστικές ελάχιστες αποστάσεις των διανυσμάτων από τα κοντινότερα κέντρα:
243426912.000000
132339824.000000
132228440.000000
132175880.000000
132145416.000000
132125640.000000
132111512.000000
132100984.000000
132093288.000000
132087392.000000
132082760.000000
132078680.000000
132075544.000000
132072832.000000
132070408.000000
132068184.000000

Επαναλήψεις:16

real    0m9,159s
user    0m59,045s
sys     0m0,180s
```

Εικόνα 21: Αποτελέσματα Εργασίας 4 με 8 νήματα

Στα αποτελέσματα γίνεται ξανά εμφανές ότι οι αθροιστικές ελάχιστες αποστάσεις είναι όλο και πιο κοντά μεταξύ τους καθώς τρέχει ο αλγόριθμος. Επίσης, γίνεται εμφανής η αντίστροφη αναλογία νημάτων και χρόνων εκτέλεσης που δείχνει ότι έγινε επιτυχώς η παραλληλοποίηση. Τα αποτελέσματα είναι συγκεντρωμένα στον παρακάτω πίνακα και δείχνουν φανερή βελτίωση σε σχέση με τις προηγούμενες εργασίες.

Σειριακό	2 Νήματα	4 Νήματα	8 Νήματα
11m45s	0m30s	0m18s	0m9s