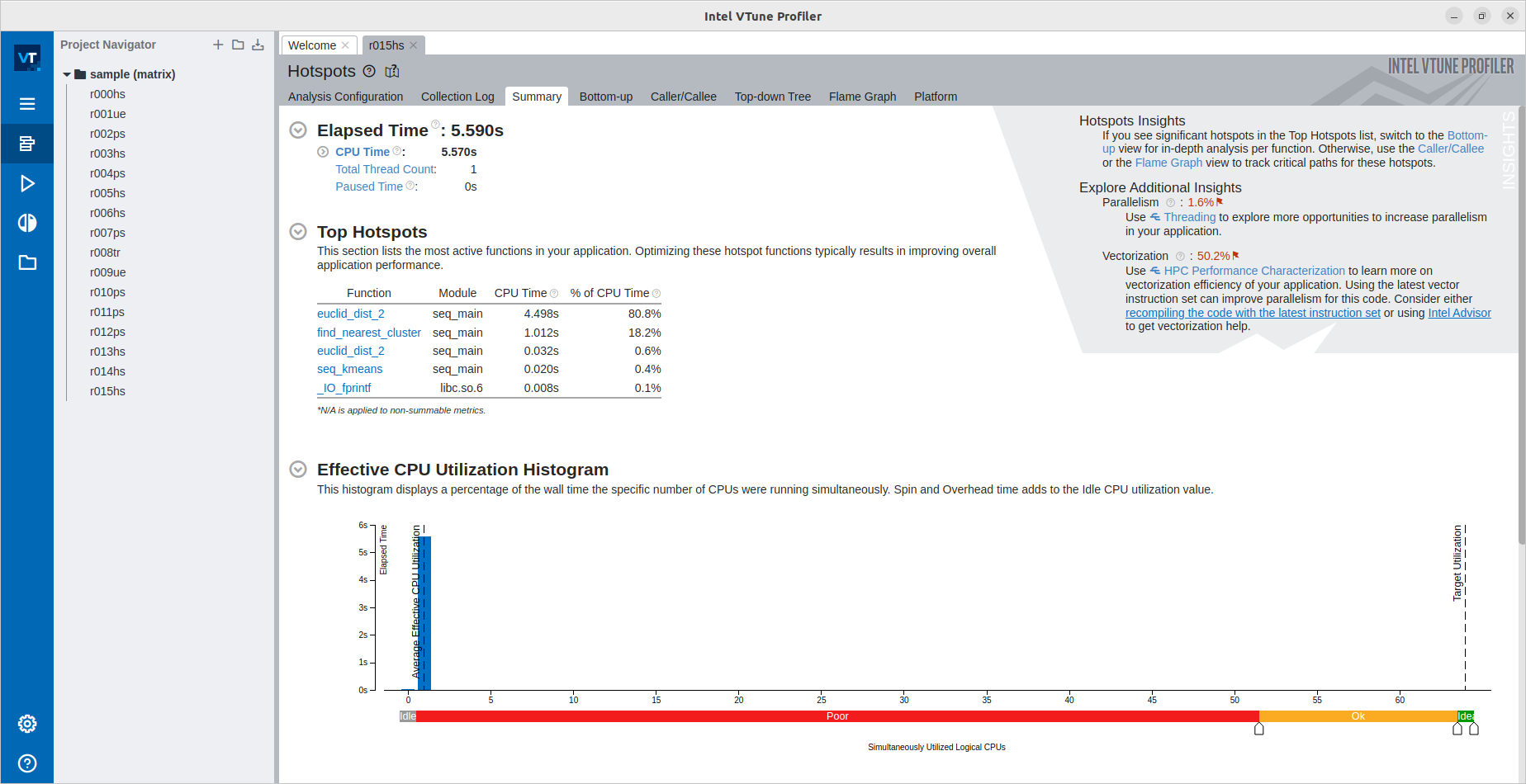
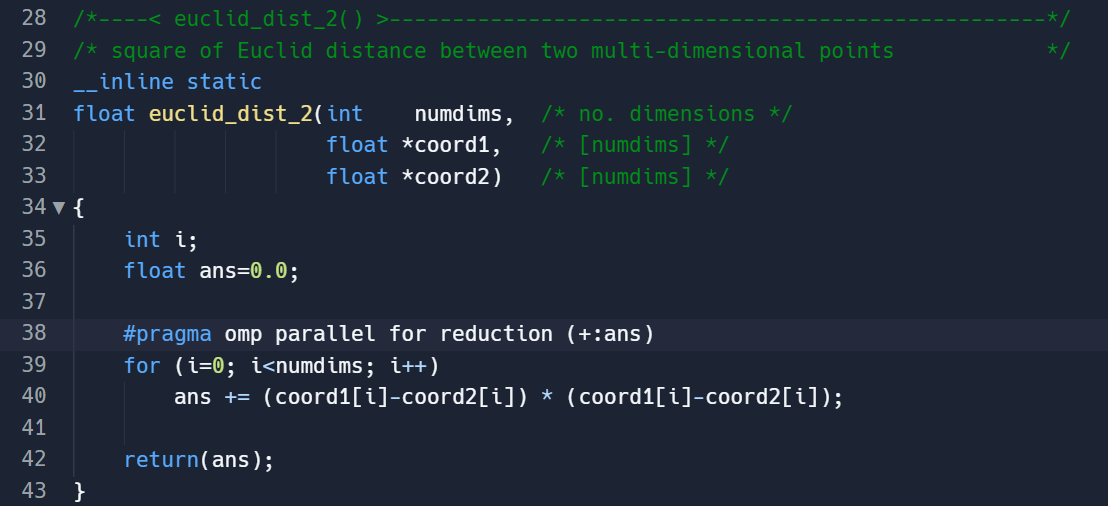
Lab 2

Σταμούλος Αλέξανδρος 02954

Αχιλλέας Γαλάνης 02941

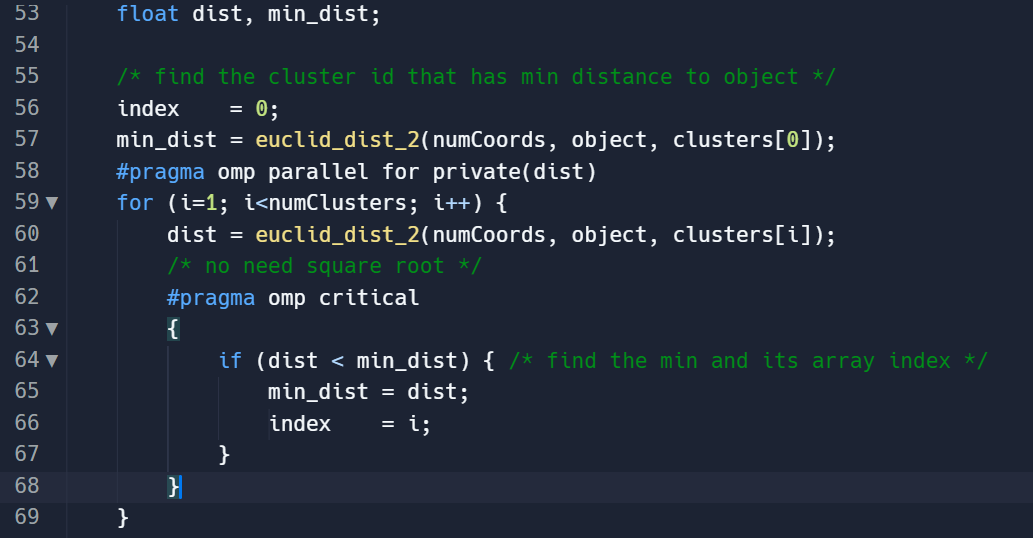


Αρχικά κάνουμε profiling των κώδικα και από το hotspot analysis παρατηρούμε ότι η συνάρτηση Euclid\_dist\_2 χαλάσει τον περισσότερο χρόνο όποτε ξεκινάμε παραλληλοποιωντας το for loop της συναρτησης.



Παρατηρουμε οτι ο χρονος εκτελεσης αυξανεται σημαντικα . Αυτο μαλλον συμβαινει επειδη η συναρτηση καλειται πολλες φορες(σε καθε επαναληωη το καθε object την καλει για number of clusters φορές ) έτσι δημιουργείται μεγάλο overhead.

Αφού ο χρόνος αυξήθηκε προχέαμε στο επόμενο σημείο που χαλάσει χρόνο το πρόγραμμα την συνάρτηση find\_nearest\_cluster , όμοια περιμένουμε ο χρόνος να αυξηθεί αφού και αυτή η συνάρτηση καλείται πολλές φορές δημιουργώντας overhead.

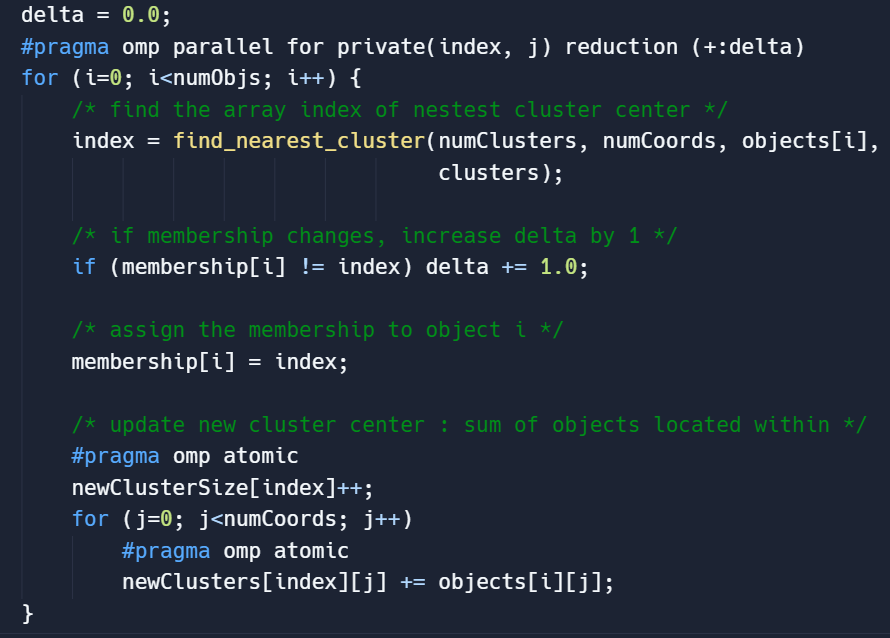


Το dist πρέπει να είναι private και το σημείο που βρίσκουμε το min πρέπει να είναι critical αφού πολλά threads πάνε να γράψουν στην μεταβλητή min\_dist.

Όντως μετά από τα πειράματα ο χρόνος εκτέλεσης αυξάνεται όπως και περιμέναμε .

Όποτε μας μένει η βασική συνάρτηση kmeans() , παραλληλοποιησαμε το for loop που γίνεται η αρχικοποίηση του πινάκα membership με μικρή μείωση στον χρόνο εκτέλεσης

Έπειτα το πρώτο for μέσα do



Το update new cluster πρέπει να είναι atomic καθώς δυο threads μπορεί να έχουν το ίδιο index και j με αποτέλεσμα να πάνε να γράψουν στο ίδιο κελί.

Προσπαθήσαμε να παραλληλοποιησουμε και το δεύτερο for (*average the sum and replace old cluster center with newClusters*) όμως αύξησε τον χρόνο εκτέλεσης ακόμα και όταν το βάλαμε σε ένα parallel region μαζί με το άλλο for.

Μετα τα πειραματα παιρνουμε τα παρακατω αποτελεσματα.

Παρατηρουμε ότι το παραλληλο με ένα thread είναι πιο αργο λογω του επιπλεον overhead.

Τα flags για το compile :

-Wall -Ofast -ipo -xHost -DNDEBUG -g -qopenmp ( -static εβγαζε warning)