

Lab1 Report

Χουλιαράς Ανδρέας AEM:2143

Compiler: icc

Compiler flags: -qopenmp -g -fast (επιπλέον σε αυτά που ήδη υπήρχαν)

Για αρχή, εκτελώντας profiling με το VTune διαπιστώθηκε ότι τα σημεία που επαναλαμβάνονται περισσότερο είναι οι συναρτήσεις `euclid_dist_2` και `find_nearest_cluster`.

Πρώτα δοκίμασα να παραλληλοποιήσω το `for loop` της `euclid_dist_2` προσθέτοντας OpenMP directives, χωρίς καμία επιτυχία. Το overhead και τα spins ήταν τόσα πολλά που έκανε το πρόγραμμα πιο αργό από το αρχικό. Έψαξα να βρω από ποιά σημεία του κώδικα καλείτε η συνάρτηση αυτή. Απ' ότι φαίνεται η `euclid_dist_2` καλείτε μόνο από την `find_nearest_cluster` και μάλιστα μια φορά μέσα σε ένα `for loop`. Έτσι εξηγείτε το overhead, γιατί τα νήματα δημιουργούνται και καταστρέφονται συνέχεια.

Προχωράμε λοιπόν σε παραλληλοποίηση του `for loop` της `find_nearest_cluster`. Ίδια ιστορία, πολύ overhead και spins χωρίς καμία βελτίωση στον χρόνο εκτέλεσης. Αυτή τη φορά παρατήρησα γρήγορα ότι όπως και πριν η `find_nearest_cluster` καλείτε μόνο από `for loops` της `seq_kmeans`. Οπότε θα επικεντρώσουμε την προσοχή μας εκεί.

Αναμένουμε λοιπόν να εξαλειφθούν από hotspots οι `euclid_dist_2` και `find_nearest_cluster` παραλληλοποιώντας το σημείο που καλείτε η `find_nearest_cluster` από την `seq_kmeans`.

Παραλληλοποιώντας το πρώτο `for loop` μέσα στο `do while` πέρνουμε για πρώτη φορά βελτίωση στο χρόνο εκτέλεσης σε σχέση με το ακολουθιακό. Το overhead είναι ακόμα πολύ και τα spins πολλά οπότε ακολουθούν δοκιμές για την πολιτική χρονοδρομολόγησης και το chunk. Το μεγαλύτερο hotspot πλέον είναι λειτουργίες του OpenMP, που υποψιάζομαι ότι έχουν σχέση είτε με την δημιουργία και καταστροφή νημάτων, γιατί το παράλληλο τμήμα είναι μέσα στο `do while`, είτε με το γεγονός ότι δύο εντολές γίνονται ατομικά.

Οι δύο ατομικές εντολές δεν μπορούν να αλλάξουν και να εκτελούνται παράλληλα γιατί σταματάει η ορθότητα του κώδικα, οπότε θα συμβιβαστούμε ότι αναγκαστικά θα υπάρχουν καθυστερήσεις για θέματα συγχρονισμού και αμοιβαίου αποκλεισμού ανάμεσα στα νήματα.

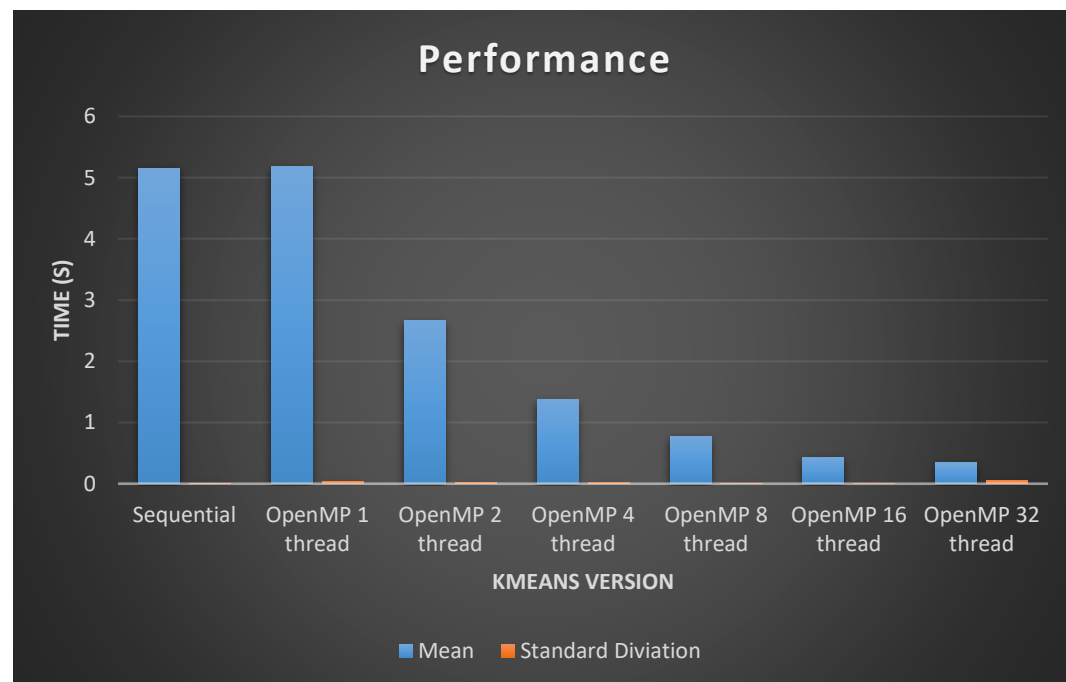
Δοκίμασα λοιπόν, προκειμένου να μειώσω το overhead λόγω δημιουργίας και καταστροφής νημάτων, να μην παραλληλοποιήσω το πρώτο `for loop` μέσα στο `do while`, αλλά αντίθετα να παραλληλοποιήσω το ίδιο το `do while`. Μετά από πολλές δοκιμές έφτασα στο συμπέρασμα ότι δεν γίνεται, ή τουλάχιστον δεν γίνεται χωρίς να μειωθεί η ορθότητα του κώδικα. Οπότε εγκατέλειψα αυτήν την προσπάθεια.

Σκέφτηκα όμως το εξής: αφού τελικά δεν μπορώ να αποφύγω την δημιουργία και καταστροφή νημάτων σε κάθε κύκλο του do while ας μεγαλώσω την παραλληλοποίηση βάζοντας τα ίδια νήματα να εκτελέσουν παράλληλα και το επόμενο for loop. Οπότε τα νήματα που δημιουργούνται μέσα στο do while εκτελούν παράλληλα το πρώτο μεγάλο for loop και μετά αφού τελειώσουν με αυτό, εκτελούν παράλληλα και το εξωτερικό for loop από τα δύο εμφωλευμένα.

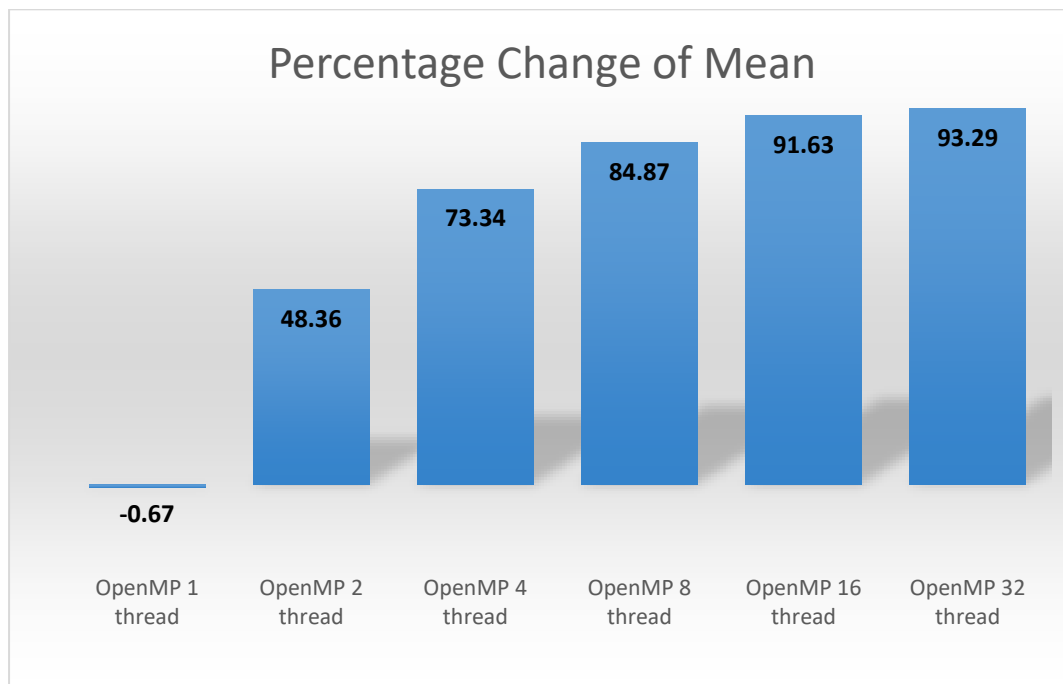
Ο χρόνος εκτέλεσης φαίνεται να βελτιώθηκε λίγο, όμως είναι λογικό γιατί το κομμάτι αυτό του κώδικα είναι πιό ελαφρύ σε υπολογιστική ισχύ και δεν έχει καθόλου κλήσεις συναρτήσεων.

Παρατήρηση: στο πλαίσιο της εργασίας εκτελούμε το πρόγραμμα με 1000 clusters και το εύρος του for loop στο δεύτερο παράλληλο κομμάτι κώδικα εξαρτάται από τον αριθμό τών clusters. Αυτό σημαίνει ότι περιμένουμε ακόμα μεγαλύτερη επίδοση για περισσότερα clusters, σε σχέση με τη προηγούμενη έκδοση. Αλλά όσο μικρότερος ο αριθμός από clusters τόσο μικρότερη η επίδοση που κερδίζουμε, φτάνοντας καπρία στιγμή σε ένα σημείο όπου το επιπλέον overhead για την παραλληλοποίηση και αυτού του loop είναι περισσότερο από την επίδοση που κερδίζουμε. Σε εκείνο το σημείο θα αρχίσουμε να βλέπουμε λιγότερη επίδοση σε σχέση με την προηγούμενη έκδοση.

Στο ακόλουθο γράφημα φαίνονται οι μέσοι χρόνοι εκτέλεσης και οι τυπικές αποκλίσεις για το ακολουθιακό και το παράλληλο για 1,2,4,8,16,32 νήματα



Στο ακόλουθο γράφημα φαίνεται η ποσοστιαία μείωση του μέσου όρου των χρόνων εκτέλεσης ανάλογα με τον αριθμό των νημάτων σε σχέση με την αρχική έκδοση:



Όλες οι χρονομετρήσεις βρίσκονται αναλυτικά στο αρχείο excel Lab2_Statistics. Έκανα 12 πειράματα για κάθε πείραμα όπως τις οδηγίες και τα έβαλα στο αρχείο χωρίς τη μεγαλύτερη και τη μικρότερη τιμή.