

Παράλληλα και Διανεμημένα Συστήματα

Εργασία 2 kNN Search

Φοιτητής : Ζηκόπης Ευάγγελος

AEM : 8808

Email : vagzikopis@gmail.com / ezikopis@ece.auth.gr

5/12/2019

Κώδικας

Ο κώδικας βρίσκεται στο παρακάτω github repository. Για την ορθή εκτέλεσή του παρακαλώ ακολουθήστε τα βήματα που περιγράφονται στο README του repository: <https://github.com/vagzikopis/kNN-MPI.git>

Περιγραφή Υλοποίησης

Η σειριακή εκτέλεση αποτελεί την υλοποίηση του αλγορίθμου k-Nearest-Neighbor Search και στα αποτελέσματά της θεωρεί πάντα κοντινότερο γείτονα ενός σημείου τον εαυτό του. Για τον υπολογισμό των αποστάσεων μεταξύ των σημείων χρησιμοποιείται η βιβλιοθήκη OpenBLAS και πιο συγκεκριμένα η συνάρτηση `cbblas_dgemm`. Αφού έχουν υπολογιστεί οι αποστάσεις, οι κοντινότεροι γείτονες κάθε σημείου επιλέγονται με χρήση του αλγορίθμου `quickselect`. Το τελικό αποτέλεσμα επιστρέφεται σε ένα struct που περιέχει ταξινομημένες σε αύξουσα σειρά τις αποστάσεις των γειτόνων από κάθε σημείο σε ένα διάνυσμα $m \times k$, τους δείκτες των κοντινότερων γειτόνων σε ένα διάνυσμα $m \times k$ και τα μεγέθη m και k .

Οι παράλληλες υλοποιήσεις κάνουν χρήση της σειριακής, για να εφαρμόσουν το kNN στο dataset που έχουν σε κάθε επανάληψη.

Στη σύγχρονη(blocking) υλοποίηση τα processes επικοινωνούν σε μορφή ring, όπου τα process με ζυγό id κάνουν πρώτα send και μετά receive και τα process με μονό id το αντίστροφο. Για την επικοινωνία χρησιμοποιούνται οι blocking συναρτήσεις `MPI_Send` και `MPI_Recv`. Επίσης χρησιμοποιούνται συνολικά 3 buffer για την προσωρινή αποθήκευση και αποστολή των dataset.

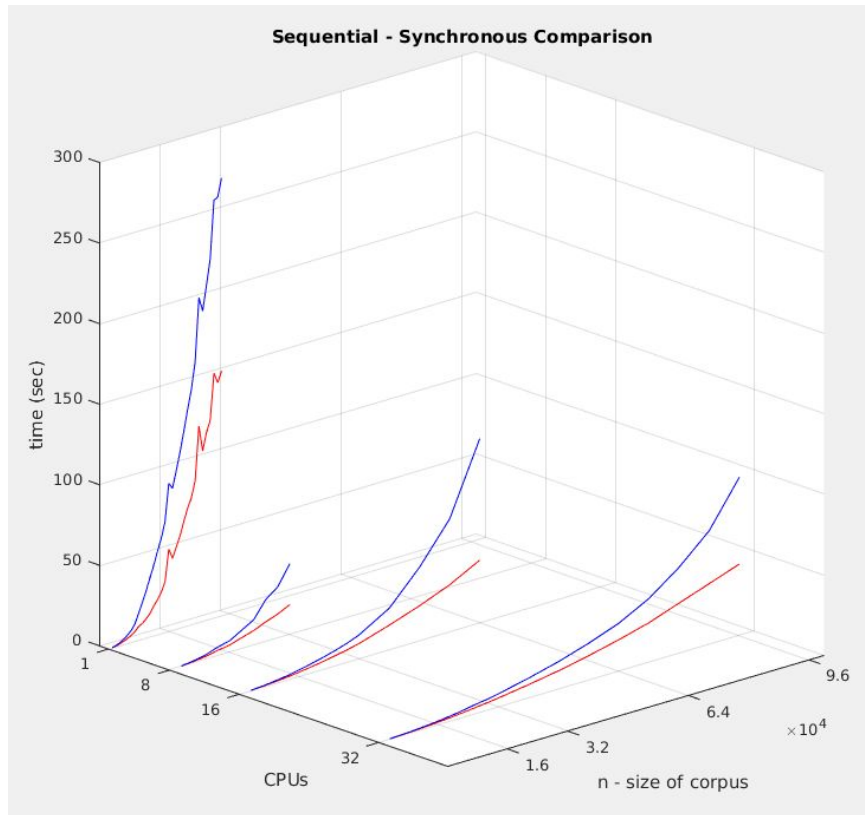
Στην ασύγχρονη υλοποίηση τα process επικοινωνούν και πάλι σε μορφή ring, αλλά σε αυτήν την περίπτωση η επικοινωνία "κρύβεται" πίσω από τον υπολογισμό του kNN. Τα processes δηλαδή κάνουν αίτημα για send/receive με το process που πρέπει και ξεκινούν τον υπολογισμό του kNN αμέσως. Και εδώ χρησιμοποιήθηκαν 3 buffer, ωστόσο από τα πειράματα παρατηρήθηκε ότι η χρήση ενός επιπλέον buffer για την προσωρινή αποθήκευση των δεδομένων που πρέπει να σταλούν στο επόμενο process αλλά και να χρησιμοποιηθούν για το kNN δεν προσθέτει χρονική καθυστέρηση (οι χρόνοι προέκυπταν περίπου ίδιοι).

Πειράματα στη συστοιχία Aristotelis

Προσπάθησα να εκτελέσω όσο το δυνατόν περισσότερα πειράματα ώστε οι χρόνοι να είναι αντιπροσωπευτικοί. Σε αρκετές υλοποιήσεις έχουν ληφθεί οι μέσοι όροι των χρόνων εκτέλεσης, ωστόσο για άλλες δεν πρόλαβα να εκτελέσω τα πειράματα πολλές φορές καθώς οι πόροι έπρεπε να μοιραστούν ισόποσα σε όλους τους φοιτητές. Τα πειράματα εκτελέστηκαν όλες τις φορές(εκτός από τη σειριακή) σε 4 nodes με διαφορετικό αριθμό task-per-node κάθε φορά.

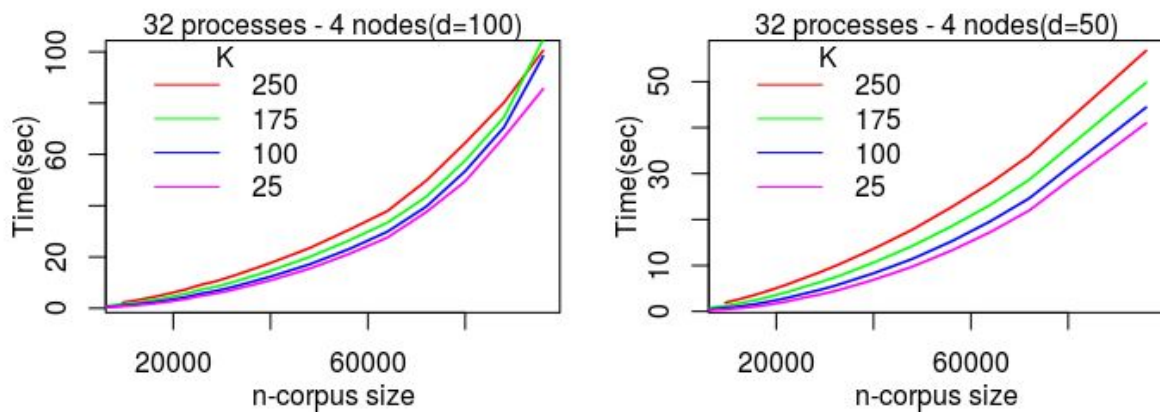
Σύγκριση σειριακής και παράλληλης υλοποίησης

Παρακάτω βλέπουμε τη σύγκριση της σύγχρονης και της σειριακής υλοποίησης για **K=100** και **D = 50(κόκκινο)** και **D = 100(μπλέ)**. Οι υλοποιήσεις με περισσότερα processes κατάφεραν να εκτελέσουν τα πειράματα για πολύ μεγαλύτερα dataset. Η σειριακή υλοποίηση ήταν η πιο χρονοβόρα και γι'αυτό εκτελέστηκε μια φορά, οπότε προέκυψαν κάποια καρφιά στο διάγραμμα. Οι υπόλοιπες εκτελέστηκαν πιο πολλές φορές και γι'αυτό η κλίση τους είναι πιο ομαλή. Το διάγραμμα για τη σύγκριση σειριακού-ασύγχρονου είναι ίδιο οπότε αποφάσισα να μην το βάλω για να μην υπερβώ το όριο των 3 σελίδων.



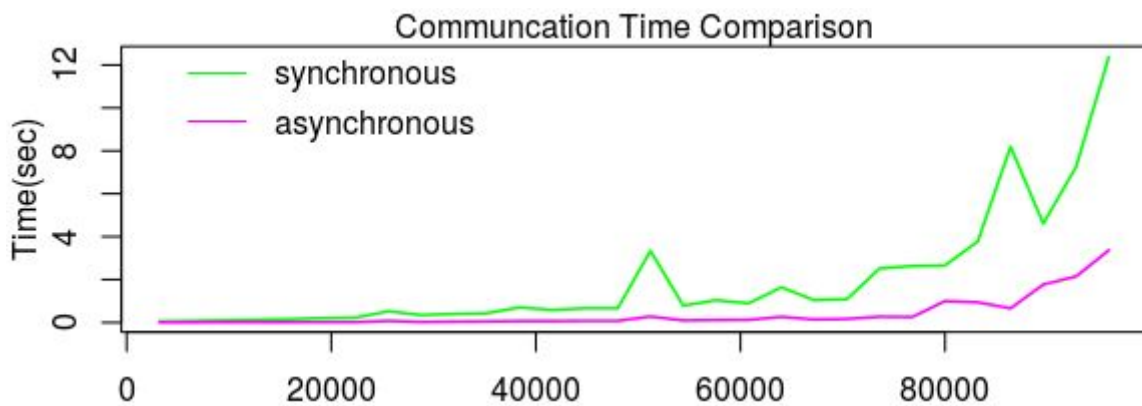
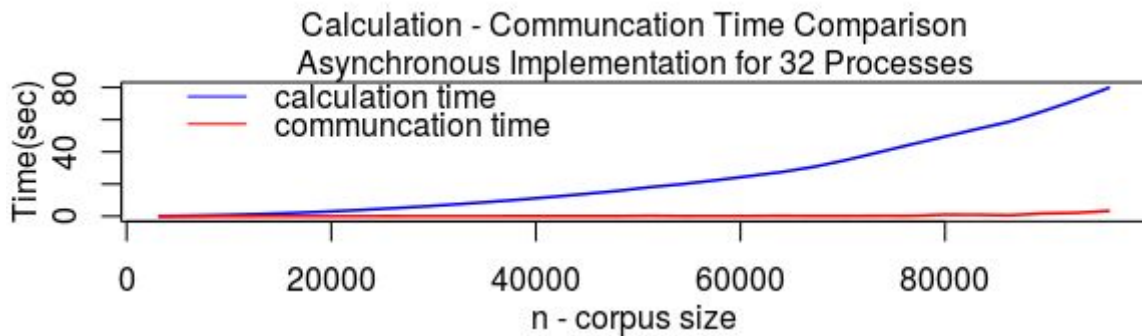
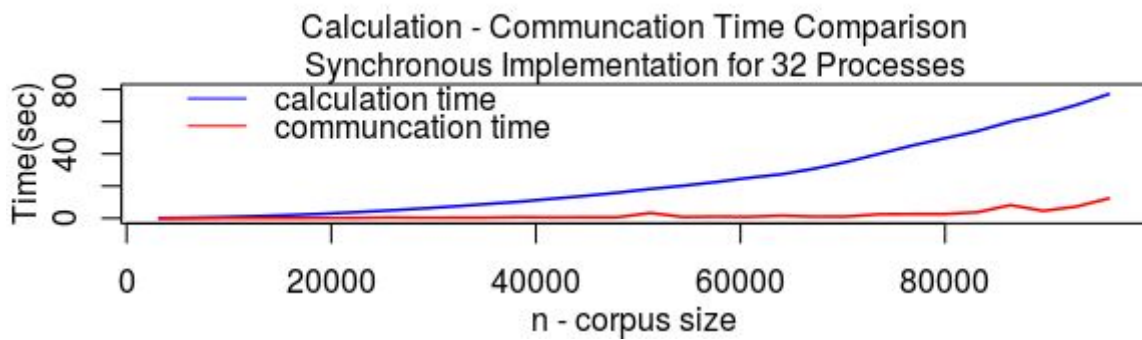
Μεταβαλλόμενη παράμετρος K

Παρακάτω βλέπουμε πως μεταβάλλεται ο χρόνος εκτέλεσης του προγράμματος ανάλογα με τον αριθμό των κοντινότερων γειτόνων που αναζητούμε. Η μεταβολή είναι όμοια για 16 και 8 processes οπότε τα διαγράμματα παραλείπονται για εξοικονόμηση χώρου.



Σύγκριση χρόνου υπολογισμών και χρόνου επικοινωνίας

Τα επόμενα δύο γραφήματα απεικονίζουν τους χρόνους υπολογισμών και επικοινωνίας για 32 processes. Το τρίτο γράφημα συγκρίνει τους χρόνους επικοινωνίας της σύγχρονης και ασύγχρονης υλοποίησης. Τα αποτελέσματα προέρχονται από τον μέσο όρο του χρόνου στον οποίο κάθε process πραγματοποιούσε υπολογισμούς και τον μέσο όρο του χρόνου στον οποίο κάθε process ήταν blocked εξαιτίας της επικοινωνίας.



Σύγκριση σύγχρονης και ασύγχρονης υλοποίησης

Στο παρακάτω πείραμα απεικονίζονται οι συνολικοί χρόνοι για τις σύγχρονες και ασύγχρονες υλοποιήσεις. Παρατηρούμε ότι η χρήση ενός επιπλέον buffer που αποθηκεύει το set που πρόκειται να στείλει η MPI_Isend δεν επηρεάζει τον χρόνο εκτέλεσης του ασύγχρονου προγράμματος. Γενικά, παρότι υπάρχει διαφορά στους χρόνους επικοινωνίας των δύο υλοποιήσεων, αυτή δεν είναι εμφανής στο συνολικό χρόνο. Αυτό συμβαίνει επειδή ο συνολικός χρόνος είναι μεγαλύτερος κατά αρκετές τάξεις μεγέθους σε σχέση με τον χρόνο επικοινωνίας μεταξύ των processes. Η εκτέλεση για 8 processes έγινε χωρίς επιπλέον buffer ενώ για 16 processes και 32 processes έγινε με επιπλέον buffer. Η διαφορά σύγχρονου και ασύγχρονου προγράμματος βλέπουμε ότι μένει σταθερή και μικρή. Στο πείραμα οι παράμετροι d και k είναι σταθερές και ίσες με 50 και 100.

