Παράλληλα και Διανεμημένα Συστήματα

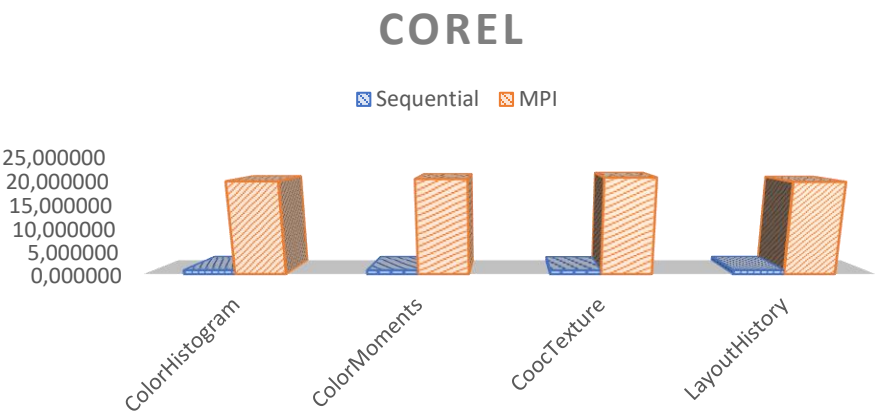
Εργασία 2

1) Intro

Σε αυτήν την εργασία μας ζητήθηκε να αναπτύξουμε τον αλγόριθμο distributed all-knn search. Ως είσοδο έχω έναν πίνακα X, που περιέχει ένα σύνολο των σημείων, όπως και ένας αριθμός n που είναι ο συνολικός αριθμός των σημείων, ο αριθμός των διαστάσεων d και ο αριθμός των γειτόνων k. Ακόμα μας δίνεται ένας πίνακας Y. Ο σκοπός του all kNN είναι να βρούμε τις αποστάσεις όλων των σημείων αυτών με όλα τα υπόλοιπα σημεία του πίνακα. Στην πραγματικότητα ο Y στην ειδική περίπτωση που μας ζητείται είναι ίδιος με τον X, παρ'όλα αυτά το υλοποίησα πιο γενικά όπως ζητείται και στο V0. Οι πίνακες είναι σε μορφή Row Major και βάσει της θεωρίας του Machine Learning, είναι σαν να έχω ως X κάποια training data, και ως Y κάποια testing data. Χρησιμοποιώντας επίσης ως μετρική τον δεδομένο τύπο της ευκλείδειας απόστασης που μας δίνεται, μπορούμε να βρούμε τις ζητούμενες αποστάσεις. Ο knn είναι ένας αλγόριθμος που μπορεί να χρησιμοποιηθεί σε γενικές γραμμές για classification. Εδώ όμως δεν ταξινομούμε ακριβώς νέα σημεία (δεν επιλέγουμε κάποια κλάση του σημείου βάσει των κλάσεων των κοντινότερων γειτόνων), απλά σταματάμε όταν βρίσκουμε τις αποστάσεις. Όσον αφορά το Vantage Point Tree, θεωρητικά θα έπρεπε να έχουμε processes που το κάθε process θα φτιάχνει ένα διαφορετικό δέντρο και με διαφορετικά σημεία το καθένα και θα γίνονται ανταλλαγές όπου χρειάζεται για να μη χάνω γείτονες.

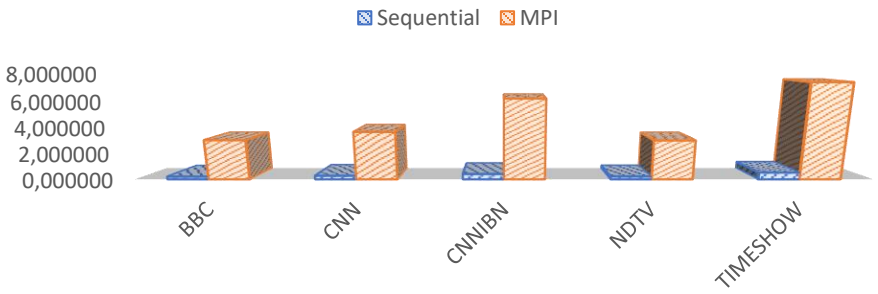
Αρχικά να επισημάνω πως δεν κατάφερα να υλοποιήσω έναν λειτουργικό Vantage Point Tree. Ενώ θεωρώ ότι έχω καταλάβει την εκφώνηση, μάλλον έκανα κάποιο λάθος στον κώδικά μου και δεν μου βγήκε σωστά. Επίσης άλλαξα επίτηδες τον κώδικά μου από Row Major σε Col Major, καθώς ενώ έπαιρνα σωστά αποτελέσματα με Row Major (για κάποιους μικρούς πίνακες που δοκίμασα με το χέρι), δεν περνούσε από τον έλεγχο στο elearning. Φαντάζομαι ότι θα είχε να κάνει με κάποιο ζήτημα του tester ή του πως έβγαζα εγώ τον πίνακα στο ndist. Τέλος, τα simulation έγιναν στον προσωπικό μου υπολογιστή, καθώς το είχα αφήσει για τις τελευταίες μέρες και είχα θέματα με την πρόσβαση στο hpc (φαντάζομαι για λόγους υπερφόρτωσης).

2) Results



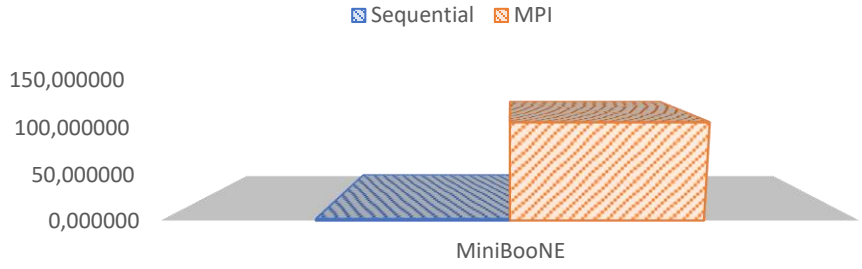
	ColorHistogram	ColorMoments	CoocTexture	LayoutHistory
Sequential	0,828687	0,798852	0,785949	0,841038
MPI	19,709245	20,174004	20,485216	19,584142

TV



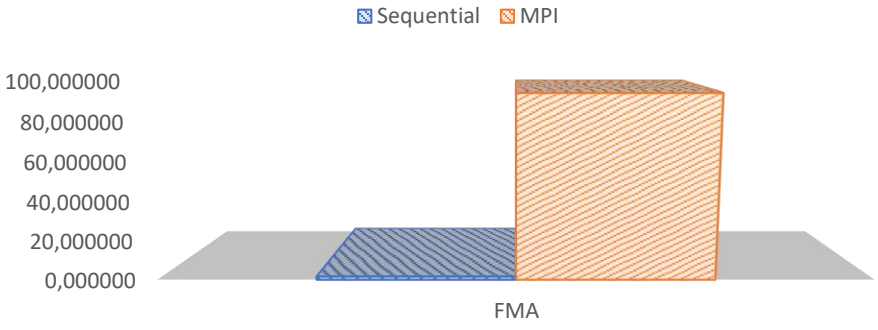
	BBC	CNN	CNNIBN	NDTV	TIMESHOW
Sequential	0,208272	0,275064	0,419950	0,225920	0,529801
MPI	3,041314	3,671483	6,169533	3,022385	7,378062

MINIBOONE



	MiniBooNE
Sequential	1,621429
MPI	104,336109

FMA



	FMA
Sequential	1,657365
MPI	93,553419

3) Regarding BLAS

Το BLAS (Basic Linear Algebra Subprograms) είναι στην πράξη κάποιες ρουτίνες που μας βοηθούν να κάνουμε κάποιες βασικές πράξεις σε διανύσματα ή πίνακες. Όσον αφορά την εργασία χρησιμοποίησα τη cblas_dgemm με flag CblasRowMajor και προφανώς με αλλαγές στις διαστάσεις, αλλά κατέληξα στην CblasColMajor, καθώς όπως προείπα ο κώδικάς μου δεν περνούσε αρχικά από το elearning.

4) Regarding MPI

Η MPI έχει αρκετά διαφορετική λογική από την pthread, την cilk ή την OpenMP. Κάποια από τα χαρακτηριστικά της έχουν να κάνουν με το γεγονός ότι η MPI, ξεκινώντας με το 0 ως parent process δίνει σε αύξοντα αριθμό ακεραίων έναν αριθμό, για κάθε νέο process που δημιουργείται. Το κάθε process ID καλείται rank και επιπλέον υπάρχουν ρουτίνες που μας δίνουν τη δυνατότητα να μπορεί κάθε process να δει το process ID του ή τον αριθμό των processes που έχουν δημιουργηθεί (MPI_Comm_rank, MPI_Comm_size). Οι πιο σημαντικές-βασικές εντολές στην MPI είναι η MPI_Send και η MPI_Recv που στέλνουν ή λαμβάνουν κάποιο «μήνυμα» από κάποιο process σε κάποιο άλλο. Πέρα από αυτές τις δύο χρησιμοποίησα και τις MPI_Isend και MPI_Irecv, οι οποίες είναι για nonblocking μηνύματα. Τέλος χρησιμοποίησα την MPI_Reduce για να περάσω το local max-min distance στο global.

5) Conclusion

Σε γενικές γραμμές, η εργασία μου φάνηκε αρκετά πιο απαιτητική απ’ ότι η πρώτη εργασία. Λόγω προγράμματος δεν πρόλαβα να τελειώσω σωστά το VPT. Παρ’όλα αυτά, θεωρώ πως η εργασία μου έδωσε κάποιες βάσεις σχετικά με το MPI αλλά και λίγο με το BLAS. Σχετικά με το MPI όντως αποτελεί ένα πολύ χρήσιμο εργαλείο για να μπορώ να χρησιμοποιήσω resource-hungry προγράμματα για προσπέλαση και πράξεις σε μεγάλα δεδομένα.

Ηλιάδης-Αποστολίδης Δημοσθένης 8811 https://github.com/iliadisd/PDS