

Εργασία MPI 2017-2018 Συνέλιξη Εικόνων

ΑΝΔΡΙΑΝΟΣ ΧΡΗΣΤΟΣ - 1115201300010

ΠΟΤΑΜΙΑΣ ΣΩΚΡΑΤΗΣ – 1115201400166

1.Εισαγωγή

Στα πλαίσια της εργασίας έγινε χρήση των βιβλιοθηκών MP I και OpenMP για την παραλληλοποίηση εργασιών σε πολλαπλές διεργασίες ή και σε πολλαπλά νήματα. Στη συγκεκριμένη εργασία απαιτείται η παραλληλοποίηση της πράξης της συνέλιξης εικόνων, μιας πολύ χρήσιμης πράξης στα γραφικά. Απαιτείται επίσης η σύγκριση των μετρήσεων και των αποτελεσμάτων που είχαμε με βάση την ανάλυση της θεωρητικής απόδοσης καθώς και τη κλιμάκωση.

Για την εκπόνηση της εργασίας βρισκόμασταν συστηματικά, οπότε δεν ανατέθηκε κάποια συγκεκριμένη αρμοδιότητα στον καθένα καθώς η υλοποίηση γινόταν απο κοινού.

Έχουν υλοποιηθεί τα κομμάτια:

- Δυναμική δέσμευση δύο πινάκων κάθε διεργασίας σε συνεχόμενη μνήμη για την αποθήκευση δεδομένων πριν και μετά την συνέλιξη
- Δυναμική δέσμευση πινάκων για αποστολή/λήψη δεδομένων προς και απο τους γείτονες
- Επανάληψη της διαδικασίας συνέλιξης 100 φορές
- Non-Blocking επικοινωνία για υπολογισμό πράξεων συνέλιξης στα σημεία που δεν χρειάζονται γειτονικά στοιχεία από άλλες διεργασίες

- Non-Blocking επικοινωνία για υπολογισμό πράξεων συνέλιξης στα σημεία που χρειάζονται γειτονικά στοιχεία από άλλες διεργασίες
- Χρήση Virtual Topology και Mapping για την βελτιστοποίηση επικοινωνίας
- Χρήση MPI_Isend - MPI_Irecv - MPI_Waitall για επικοινωνία των διεργασιών

Δεν έχει υλοποιηθεί το κομμάτι CUDA γιατί δεν είχαμε πρόσβαση στο απαραίτητο hardware.

2. Ανάπτυξη Κώδικα

Για το κομμάτι του MPI :

Αρχικά παίρνουμε τον αριθμό διεργασιών και δημιουργούμε έναν τετράγωνο πίνακα διαστάσεων $(1920/\text{αριθμό διεργασιών})+2$ καθώς δεν έχουμε υλοποιήσει το parallel I/O για να παίρνουμε δεδομένα από την εικόνα. Το κομμάτι $1920/\text{αριθμό διεργασιών}$ έχει την πληροφορία και περιφεριακά υπάρχει κενό ώστε να δεχτεί τις πληροφορίες από τους γείτονες. Επίσης έχουμε άλλον ένα πίνακα $1920/\text{αριθμό διεργασιών}$ για το αποτέλεσμα μετά την συνέλιξη.

Στη συνέχεια ξεκινούν οι απαραίτητες διεργασίες για την εύρεση των γειτόνων κάθε διεργασίας.

Σε επόμενο βήμα γίνεται η αποστολή των περιφεριακών στοιχείων προς τους κατάλληλους γείτονες και επεξεργασία των εσωτερικών στοιχείων που δεν χρειάζονται πληροφορία από τους γείτονες.

Με τη χρήση της MPI_Waitany εξασφαλίζουμε ότι δεν θα υπάρχει χρόνος αναμονής του προγράμματος καθώς μόλις λάβει κάποια πληροφορία κάνει τις απαραίτητες ενέργειες.

Για τον υπολογισμό της συνέλιξης σε οριακά σημεία της εικόνας που δεν έχουν γείτονες, χρησιμοποιούμε την τιμή του εκάστοτε σημείου σαν τιμή των γειτόνων που δεν υπάρχουν, σύμφωνα με την εκφώνηση.

Τέλος πριν αρχίσει η επόμενη επανάληψη μεταφέρεται ο πίνακας με το αποτέλεσμα της συνέλιξης στον μεγαλύτερο πίνακα για να αρχίσει πάλι η ίδια διαδικασία. Επίσης στη μεταβλητή **foundS** κρατάμε τον αριθμό της επανάληψης στην οποία ο μέσος ορός του πίνακα είναι μικρότερος απο την προηγούμενη, γιατί τότε δεν παρουσιάζεται αλλαγή στην εικόνα.

Για το κομμάτι του Open Mp:

Η ενσωμάτωση του OpenMP στο κεντρικό πρόγραμμα του MPI έγινε σε διαφορετικό αρχείο **openmp.c**.

Οι βρόχοι που παραλληλοποιήθηκαν είναι οι εξής:

- Βρόχος συνέλιξης εσωτερικών σημείων
- Βρόχος συνέλιξης ακραίων σημείων (εκτός των γωνιακών)

Στο προγράμμα μας χρησιμοποιούμε `schedule(auto)`.

3. Μετρήσεις

Για το κομμάτι του MPI :

Μετρήσεις για πλήθος επαναλήψεων 100

Processes	Time(seconds)	Speedup	Efficiency
1	~7,706	1	1
4	~0.537	14.3	3.57
16	~0.082	93.9	5.56
36	~0.064	120.4	3.34
64	~0.062	124.2	1.94
81	~0,059	130.6	1.61
100	~0,052	148.1	1.48
121	~0,051	151.1	1.24

Για το κομμάτι του Open Mp:

- Παράδειγμα για 25 processes

Threads	Time(seconds)	Speedup
1	~0.026	1
2	~0.036	0.72
3	~0.048	0.54
4	~217.4	0.00001
5	~4.562	0.005
6	~8.841	0.002

- Παράδειγμα για 16 processes

Threads	Time(seconds)	Speedup
1	~0.509	1
2	~0.785	0.64
3	~0.820	0.62
4	~35.86	0.014
5	~2.112	0.24
6	~1.025	0.49

4. Συμπεράσματα και επεκτάσεις

Η επιτάχυνση είναι αρκετά καλή όσο αυξάνονται οι διεργασίες αλλά όσο αυξάνονται οι διεργασίες τόσο λιγότερο αυξάνεται η επιτάχυνση. Παρατηρούμε ότι στις 100 και 121 διεργασίες πετυχαίνουμε ελάχιστη επιτάχυνση. Είναι προφανές πως όταν αυξάνονται οι διεργασίες η καθυστέρηση οφείλεται στην επικοινωνία των διεργασιών λόγω συμφόρησης παρά στην επεξεργασία των δεδομένων. Καταλήγουμε στο συμπέρασμα πως το πρόβλημα μας δεν είναι ισχυρά επεκτάσιμο γιατί όσο αυξάνονται οι διεργασίες τόσο λιγότερο εκμεταλευόμαστε την επεξεργαστική ισχύ.

Μια επέκταση που θα μπορούσε να γίνει στο πρόγραμμά μας είναι η υλοποίηση του parallel I/O.

5. Μεταγλώτιση και εκτέλεση

Για την μεταγλώτιση των προγραμμάτων χρησιμοποιείτε την εντολή **make mpi** : για το πρόγραμμα μόνο με mpi

make openmp : για το hybrid πρόγραμμα.

Κατά την εκτέλεση:

1. Για το πρόγραμμα mpi:

- `mpiexec -f machines -n <number of processes> mpi`

2. Για το πρόγραμμα openmp:

- `mpiexec -f machines2 -n <number of processes> openmp`

*το αρχείο `machines2` αναθέτει ένα process σε κάθε διαθέσιμο υπολογιστή σε αντίθεση με το `machines` που αναθέτει 4.

**Η αλλαγή των threads γίνεται στην γραμμή 27 του αρχείου `openmp.c`.

