Παράλληλα Συστήματα :
Εργασία MPI και MPI+openMP

Φοιτητής 1: Γεώργιος Μητράκης
A.M. : 1115201400107
SDI: sdi1400107

Φοιτητής 2: Δημήτριος Σιδέρης
A.M. : 1115201400182
SDI: sdi1400182

Μεταγλώττιση : make

Καθαρισμός : make clean

Εκτέλεση :

• **MPI**:
mpiexec -n <procs> -f machines
./project_mpi
-p <path_to_input>
-w <width>
-h <height>
-f <number> for fraction of the image / -m <number> for image multiple (optional)
-g for black-and-white / -c for colored input
-r to run with reduce (optional)

• **MPI + openmp**
mpiexec -n <procs> -f machines
./project_hybrid
-p <path_to_input>
-w <width>
-h <height>
-f <number> for fraction of the image / -m <number> for image multiple (optional)
-g for black-and-white / -c for colored input
-r to run with reduce (optional)

Σημαντική σημείωση: Δεν υλοποιήθηκε ο μηχανισμός του parallel I/O από κάποιο αρχείο εικόνας. Η επεξεργασία γίνεται σε πίνακες με τυχαίους ακεραίους εύρους [0, 255] ο καθένας (ή το κάθε χρωματικό κανάλι του καθενός). Για αυτό στο όρισμα -p path_to_file δεν ελέγχεται αν υπάρχει το file που εισήχθη, ωστόσο, για τυπικούς λόγους ορθότητας, το όρισμα απαιτείται από το χρήστη.

Αρχεία και περιεχόμενά τους:

• project_mpi.c
Περιέχει τη main συνάρτηση του προγράμματος με υλοποίηση MPI, καθώς και τις δηλώσεις των inline συναρτήσεων συνέλιξης και τους ορισμούς συναρτήσεων για την κατασκευή των halo points ως Datatype του MPI.

• project_hybrid.c
Περιέχει τη main συνάρτηση του προγράμματος με υλοποίηση MPI+openmp , καθώς και τις δηλώσεις των inline συναρτήσεων συνέλιξης και τους ορισμούς συναρτήσεων για την κατασκευή των halo points ως Datatype του MPI.

• tools.h
Header file που περιέχει τον ορισμό της συνάρτησης για τον έλεγχο ορθότητας των ορισμάτων εισόδου του εκάστοτε προγράμματος.

• black_and_white_header.h
Header file, το οποίο περιλαμβάνει τους ορισμούς των inline συναρτήσεων συνέλιξης των σημείων της ασπρόμαυρης εικόνας.

• rgb_header.h
Header file, το οποίο περιλαμβάνει τους ορισμούς των inline συναρτήσεων συνέλιξης, και την δομή struct pixel, για την αποθήκευση των στοιχείων της έγχρωμης εικόνας

• machines
Ένα text file με τα ονόματα των διαθέσιμων μηχανών και των πυρήνων τους για το -f όρισμα του mpiexec.

• askdate.csh
Ένα c-shell script που επιστρέφει την ημερομηνία από όλα τα διαθέσιμα μηχανήματα του εργαστηρίου.

Δομή και τρόπος λειτουργίας του προγράμματος:

Κατά την εκτέλεση του προγράμματος, αρχικά ελέγχεται αν τα ορίσματα που έδωσε ο χρήστης για την εκτέλεση είναι ορθά, καθώς και αν η είσοδος μπορεί να διαιρεθεί ακριβώς σε √<procs> * √<procs> κομμάτια. Αν δεν ισχύει κάτι από τα παραπάνω, η διεργασία με rank 0 εκτυπώνει ανάλογο μήνυμα και το πρόγραμμα κλείνει.

Μετά τον έλεγχο, αναλόγως με το αν η είσοδος είναι μια ασπρόμαυρη ή μια έγχρωμη εικόνα, καλείται η συνάρτηση black_and_white ή η rgb_image αντίστοιχα για την συνέλιξη της εικόνας.

Η καθεμία από τις <procs> διεργασίες έχει ένα τμήμα της αρχικής εικόνας, το οποίο τμήμα έχει διαστάσεις total_width/√<procs> * total_height/√<procs>. Στην ασπρόμαυρη εικόνα, τα στοιχεία είναι τύπου unsigned char, μεγέθους 1 byte και με εύρος τιμών [0,255], ενώ στην έγχρωμη, τα στοιχεία είναι τύπου struct pixel, το οποίο περιλαμβάνει 3 unsigned chars, 1 για κάθε χρώμα.

Οι δύο συναρτήσεις ακολουθούν ακριβώς τα ίδια βήματα για την επεξεργασία της εισόδου. Δημιουργείται μια καρτεσιανή τοπολογία, κάθε διεργασία βρίσκει τις γειτόνικές της σε αυτήν την τοπολογία και κατασκευάζονται οι MPI τύποι vertical_halo_point και horizontal_halo_point για την ανταλλαγή βορείων, νοτίων, ανατολικών και δυτικών ακριανών σημείων μεταξύ των διεργασιών.

Έπειτα, γίνεται επαναληπτικά χρήση της μεθοδολογίας του Foster (επικάλυψη επικοινωνίας με υπολογισμούς) για την αποδοτικότερη συνέλιξη των στοιχείων της εικόνας, με την ακόλουθη σειρά:
• Ξεκινάνε οι διαδικασίες αποστολής και λήψης των halo points της κάθε διεργασίας προς και από τις γειτονικές της διεργασίες στην τοπολογία.
• Μέχρι να ολοκληρωθεί αυτή η ανταλλαγή πληροφορίας, κάθε διεργασία κάνει συνέλιξη του εσωτερικού της εικόνας της, για την οποία δεν απαιτείται πληροφορία από άλλες διεργασίες.
• Η διεργασία αναμένει για να λάβει πληροφορία. Στην λήψη, ελέγχεται ποιο halo point ήταν αυτό που ελήφθη και πραγματοποιείται συνέλιξη της αντίστοιχης περιοχής της εικόνας.
• Η διεργασία αναμένει μέχρι να στείλει όλη την πληροφορία που χρειάζονται οι γείτονές της.

• Αν δεν υπάρχει καμία αλλαγή, ή αν οι διεργασίες έχουν ολοκληρώσει τον μέγιστο αριθμό επαναλήψεων, το πρόγραμμα τερματίζει.Στην περίπτωση του υβριδικού προγράμματος, οι εσωτερικοί
υπολογισμοί διαμοιράζονται σε πλήθος νημάτων, το οποίο είναι ίσο με τον συνολικό αριθμό των διεργασιών στον οποίο εκτελείται το πρόγραμμα. Τα νήματα δημιουργούνται κάθε φορά που έχουμε υπολογισμούς πολλαπλών στοιχείων του πίνακα σε for, δηλαδή στα εσωτερικά αλλά και σε εξωτερικά, σε περίπτωση που το αντίστοιχο halo point έχει ληφθεί.

Τέλος όσον αφορά την επιλογή τμήματος ή πολλαπλασίου της εικόνας, ισχύουν τα ακόλουθα:
• Για την επιλογή τμήματος της εικόνας, ο χρήστης μπορεί να επιλέξει μόνο το 1⁄2 ή το 1⁄4 αυτής. Για το πολλαπλάσιο της εικόνας, μπορεί να μεγεθύνει την εικόνα με μια δύναμη του 2 (X2, X4, X8, X16...). Οι παρακάτω μετρήσεις ωστόσο φτάνουν μέχρι το X16.
• Η επιλογή τμήματος της εικόνας γίνεται ως εξής:
◦ για το 1⁄2, οι διαστάσεις του θα είναι το 1⁄2 του ύψους * το πλάτος της αρχικής,
◦ ενώ για το 1⁄4 θα είναι το 1⁄2 του ύψους * το 1⁄2 του πλάτους της αρχικής.
• Η μεγέθυνση της εικόνας γίνεται με αντίστοιχο τρόπο:
◦ για το X2 η νέα εικόνα θα ειναι ύψος * 2*πλάτος της αρχικής,
◦ για το X4 θα είναι 2*ύψος * 2*πλάτος της,
◦ για το X8 θα είναι 2*ύψος * 4*πλάτος της, κ.ο.κ.

Συναρτήσεις:

• Build_vertical_halo_point
Φτιάχνει ένα MPI_Datatype το οποίο θα αντιστοιχεί σε ένα κάθετο (δυτικό ή ανατολικό) halo point του πίνακα που αποτελείται από MPI_elem_type στοιχεία.

• Build_horizontal_halo_point
Φτιάχνει ένα MPI_Datatype το οποίο θα αντιστοιχεί σε ένα οριζόντιο (βόρειο ή νότιο) halo point του πίνακα που αποτελείται από MPI_elem_type στοιχεία.

Συναρτήσεις ασπρόμαυρης εικόνας:

• black_and_white
Συνάρτηση για επεξεργασία και συνέλιξη ασπρόμαυρης εικόνας.

• elem_calc
Συνάρτηση για συνέλιξη ενός εσωτερικού σημείου του πίνακα ασπρόμαυρης εικόνας.

• northest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην πρώτη σειρά του πίνακα ασπρόμαυρης εικόνας.

• southest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην τελευταία σειρά του πίνακα ασπρόμαυρης εικόνας.

• eastest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην τελευταία στήλη του πίνακα ασπρόμαυρης εικόνας.

• westest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην πρώτη στήλη του πίνακα ασπρόμαυρης εικόνας.

• northeastern_elem_calc

Συνάρτηση για συνέλιξη του τελευταίου σημείου της πρώτης σειράς του πίνακα ασπρόμαυρης εικόνας.

• northwestern_elem_calc
Συνάρτηση για συνέλιξη του πρώτου σημείου του πίνακα ασπρόμαυρης εικόνας.

• southeastern_elem_calc
Συνάρτηση για συνέλιξη του τελευταίου σημείου του πίνακα ασπρόμαυρης εικόνας.

• southwestern_elem_calc
Συνάρτηση για συνέλιξη του πρώτου σημείου της τελευταίας σειράς του πίνακα ασπρόμαυρης εικόνας.

Συναρτήσεις έγχρωμης εικόνας:

• rgb_image
Συνάρτηση για επεξεργασία και συνέλιξη έγχρωμης εικόνας.

• rgb_elem_equal
Συνάρτηση που ελέγχει αν δύο στοιχεία τύπου struct pixel είναι ίσα.

• rgb_elem_calc
Συνάρτηση για συνέλιξη ενός εσωτερικού σημείου του πίνακα έγχρωμης εικόνας.

• rgb_northest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην πρώτη σειρά του πίνακα έγχρωμης εικόνας.

• rgb_southest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην τελευταία σειρά του πίνακα έγχρωμης εικόνας.

• rgb_eastest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην τελευταία στήλη του πίνακα έγχρωμης εικόνας.

• rgb_westest_elem_calc
Συνάρτηση για συνέλιξη ενός σημείου στην πρώτη στήλη του πίνακα έγχρωμης εικόνας.

• rgb_northeastern_elem_calc
Συνάρτηση για συνέλιξη του τελευταίου σημείου της πρώτης σειράς του πίνακα έγχρωμης εικόνας.

• rgb_northwestern_elem_calc
Συνάρτηση για συνέλιξη του πρώτου σημείου του πίνακα έγχρωμης εικόνας.

• rgb_southeastern_elem_calc
Συνάρτηση για συνέλιξη του τελευταίου σημείου του πίνακα έγχρωμης εικόνας.

• rgb_southwestern_elem_calc
Συνάρτηση για συνέλιξη του πρώτου σημείου της τελευταίας σειράς του πίνακα έγχρωμης εικόνας.

Μετρήσεις

Το πρόγραμμα εκτελεί 100 επαναλήψεις με και χωρίς την κλήση του reduce.
Οι μετρήσεις βρίσκονται στα αρχεία mpi_timings.pdf, hybrid_timings_1.pdf και hybrid_timings_2.pdf

Παρατηρήσεις και Συμπεράσματα

MPI:

Αρχικά παρατηρούμε ότι στις περισσότερες μετρήσεις οι λόγοι των χρόνων με και χωρίς reduce προσεγγίζουν το 1, οπότε συμπεραίνουμε ότι προγραμματιστικά υπάρχει reduce.

Έπειτα, παρατηρούμε ότι, για συγκεκριμένο αριθμό διεργασιών, όσο κινούμαστε προς τα δεξιά του πίνακα και διπλασιάζουμε την εικόνα και αυξάνουμε και το μέγεθος των μηνυμάτων μεταξύ των διεργασιών, πετυχαίνουμε και αντίστοιχη αύξηση του χρόνου περάτωσης.

Επίσης, παρατηρούμε ότι, για συγκεκριμένη ανάλυση, όσο αυξάνουμε τον αριθμό των διεργασιών για το ίδιο πλήθος δεδομένων, όσο δηλαδή αυξάνουμε τους κόμβους επικοινωνίας και μειώνουμε το εσωτερικό πλήθος υπολογισμών κάθε διεργασίας, οι χρόνοι μειώνονται αναλογικά με το ποσοστό αύξησης των διεργασιών, πράγμα που φαίνεται κυρίως για μεγάλες αναλύσεις εικόνας, όπου επιτυγχάνεται καλύτερη επικάλυψη επικοινωνίας με υπολογισμούς.

Συνεπώς το πρόγραμμα παρουσιάζει κλιμάκωση, γιατί όσο αυξάνεται το πλήθος των υπολογισμών, μπορούμε να έχουμε μικρούς χρόνους περάτωσης αυξάνοντας το πλήθος των διεργασιών.

Τέλος, οι μετρήσεις της ασπρόμαυρης εικόνας είναι περίπου 3 φορές μικρότερες από αυτές της έγχρωμης, πράγμα που οφείλεται στο ότι οι υπολογισμοί γίνονται σε 1 χρωματικό κανάλι αντί για 3.

MPI+OpenMP :

- Εκτέλεση με comm_sz νήματα ανά διεργασία, μέχρι 4 διεργασίες ανά επεξεργαστή, με μέγιστο αριθμό διεργασιών 100 (Πίνακες 2.1 και 2.2):

Από τα αποτελέσματα του προγράμματος παρατηρούμε τα εξής:
1)όταν έχουμε reduce, ο χρόνος εκτέλεσης είναι λίγο μεγαλύτερος. Αυτό το περιμέναμε, δεδομένου ότι πρόκειται για επιπλέον πράξεις. Μάλιστα η διαφορά χρόνου είναι μεγαλύτερη όσο αυξάνεται ο αριθμός των διεργασιών. Αυτό είναι αναμενόμενο γιατί γίνονται παραπάνω πράξεις.
2)Γενικώς οι χρόνοι εκτέλεσης τριπλασιάζονται όταν πάμε από ασπρόμαυρη σε έγχρωμη εικόνα. Επίσης διπλασιάζονται κάθε φορά που

αυξάνουμε το μέγεθος της εικόνας. Αυτό σημαίνει ότι το πρόγραμμά μας δεν έχει καλή κλιμάκωση.
3) Γενικώς παρατηρούμε βέλτιστους χρόνους για μικρά n ενώ για μεγαλύτερα n έχουμε λίγο μεγαλύτερους χρόνους. Αυτό γίνεται λόγω καθυστέρησης πρόσβασης στην μνήμη λόγω ανταγωνισμού (race condition). Όλες οι μετρήσεις του hybrid προγράμματος (mpi+openmp) έγιναν μέσω putty στα linux της σχολής.

- Εκτέλεση με 4 νήματα ανά διεργασία, μέχρι 1 διεργασία ανά επεξεργαστή,μέχρι 36 διεργασίες (Πίνακες 2.3 και 2.4):

Παρατηρείται ότι έχουμε παρόμοιους χρόνους με αυτούς του MPI προγράμματος, έχοντας όμως 4 νήματα για κάθε διεργασία αντί για 1 για τους εσωτερικούς υπολογισμούς και έχοντας την κάθε διεργασία να επικοινωνεί από διαφορετικό μηχάνημα με τις υπόλοιπες. Το πρόγραμμα παρουσιάζει παρόμοια κλιμάκωση με αυτό του MPI, ωστόσο δεν υποτετραπλασιάζονται, ή έστω, μειώνονται αναλογικά οι χρόνοι, όπως θα περιμέναμε.