

Εργασία #3 (MPI και υβριδικός προγραμματισμός)

Θόλωση εικόνων με MPI (35%)

Σας δίνεται ένα σειριακό πρόγραμμα το οποίο εφαρμόζει Gaussian blur προκειμένου να θολώσει (ή να ομαλοποιήσει) μία εικόνα. Η συνάρτηση που κάνει τη θόλωση είναι η `gaussian_blur_serial()`, η οποία παίρνει μία εικόνα `imgin` και παράγει τη θολωμένη της εκδοχή `imgout`, βάσει μίας ακτίνας θόλωσης `radius` (όσο μεγαλύτερη η ακτίνα, τόσο πιο έντονο το θόλωμα). Σας ζητείται να γίνεται η θόλωση παράλληλα χρησιμοποιώντας το MPI ως εξής:

- Θα πρέπει να εισάγετε όλες τις απαραίτητες ενέργειες για την αρχικοποίηση/τερματισμό του MPI στη συνάρτηση `main()`
- Θα πρέπει να δημιουργήσετε μία συνάρτηση `gaussian_blur_mpi()`, η οποία με τις κατάλληλες εντολές αποστολής/λήψης μηνυμάτων του MPI, η θόλωση να πραγματοποιείται παράλληλα από τις διεργασίες των διαθέσιμων κόμβων. Συγκεκριμένα, θα πρέπει:
 - να παραλληλοποιηθεί το εξωτερικό `for loop`, με την κάθε διεργασία να αναλαμβάνει τη θόλωση ενός συνεχόμενου μπλοκ γραμμών της εικόνας,
 - η συνάρτηση `gaussian_blur_mpi()` να δέχεται, εκτός των υπόλοιπων ορισμάτων, το ID της διεργασίας και το πλήθος των διεργασιών.

Υβριδικός πολλαπλασιασμός πινάκων MPI/OpenMP (35%)

Ο υβριδικός προγραμματισμός χρησιμοποιεί συνδυασμό προγραμματιστικών μοντέλων για την παραλληλοποίηση μίας εφαρμογής. Η πιο συνηθισμένη μορφή του σε υπολογιστικές συστάδες (`clusters`) είναι να χρησιμοποιείται το MPI με μία διεργασία ανά κόμβο, και μέσα στον κάθε (πολυπύρηνο) κόμβο η διεργασία να παραλληλοποιεί τον υπολογισμό της με νήματα είτε μέσω της βιβλιοθήκης νημάτων POSIX, είτε μέσω OpenMP.

Υλοποιήστε και χρονομετρήστε ένα προγράμματα για τον πολλαπλασιασμό πίνακα επί πίνακα χρησιμοποιώντας το MPI και παραλληλοποίηση του εξωτερικού βρόχου (ο οποίος διατρέχει τις γραμμές). Η παραλληλοποίηση αυτή είναι γνωστή και ως μέθοδος διαχωρισμού γραμμών (`strip-partitioning`) όπου κάθε κόμβος αναλαμβάνει τον υπολογισμό μια «λωρίδας» συνεχόμενων γραμμών του αποτελέσματος.

Θα πρέπει να έχετε 1 διεργασία ανά κόμβο και η διεργασία θα εκμεταλλεύεται τα πολλαπλά `cores` του κόμβου μέσω OpenMP. Με άλλα λόγια, αν η μηχανή σας αποτελείται από K κόμβους των N πυρήνων ο καθένας, αντί να φτιάξετε $K \times N$ διεργασίες MPI, θα πρέπει να φτιάξετε K διεργασίες. Κάθε μία από αυτές θα παραλληλοποιεί τον υπολογισμό που της αντιστοιχεί με χρήση OpenMP, όπου η παράλληλη περιοχή θα εκτελείται από N νήματα.

Θα πρέπει να χρονομετρηθεί τόσο ο συνολικός χρόνος των προγραμμάτων όσο και ο χρόνος που σπαταλήθηκε σε επικοινωνίες και να γίνει σύγκριση με τα αντίστοιχα αποτελέσματα από το σειριακό πρόγραμμα. Σε κάθε περίπτωση να γίνεται χρήση συλλογικών επικοινωνιών, όπου και εφόσον είναι εφικτό.

Μονόπλευρες επικοινωνίες (30%)

Η επικοινωνία μεταξύ δύο διεργασιών, όπως την έχουμε δει μέχρι στιγμής, θεωρείται αμφίπλευρη (`two-sided`) διότι, για να ολοκληρωθεί, πρέπει να ενεργήσουν και οι δύο εμπλεκόμενες διεργασίες: η μία διεργασία πρέπει να εκτελέσει αποστολή και η άλλη λήψη. Στη λεγόμενη μονόπλευρη επικοινωνία (`one-sided communication`), ενεργεί μόνο ένα από τα δύο μέρη, είτε ο αποστολέας είτε ο παραλήπτης. Στην περίπτωση αυτή, εκείνος που εκτελεί την επικοινωνία πρέπει να προσδιορίσει όλες τις απαραίτητες παραμέτρους. Για παράδειγμα, στη μονόπλευρη αποστολή, η πηγή πρέπει να καθορίσει τον χώρο που βρίσκεται το μήνυμα που θα σταλθεί αλλά και τον χώρο (στον προορισμό) όπου θα αποθηκευτεί και η αποστολή θα γίνει χωρίς τη μεσολάβηση του παραλήπτη. Οι μονόπλευρες επικοινωνίες είναι ευρύτερα γνωστές με τον όρο απομακρυσμένη προσπέλαση μνήμης (`Remote Memory Access, RMA`), όπου μία διεργασία μπορεί να προσβεί, γράφοντας ή διαβάζοντας, άμεσα τη μνήμη μίας άλλης.

Αναζητήστε πληροφορίες για το τι ακριβώς είναι οι μονόπλευρες επικοινωνίες, τι παρέχει το MPI και πώς τις χρησιμοποιεί κάποιος σε μία εφαρμογή προκειμένου να γράψετε έναν μικρό οδηγό (2-3 σελίδες το πολύ) για αυτές στην αναφορά σας.

Λεπτομέρειες

Απαιτούμενα

- Θα πρέπει να παραδώσετε πλήρη αναφορά, περιλαμβάνοντας και γραφικές παραστάσεις χρονομετρήσεων καθώς και συζήτηση γύρω από τα αποτελέσματα.
- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά) θα πρέπει να τα παραδώσετε με `turnin set3@mye023`. Πληροφορίες στην ιστοσελίδα του μαθήματος.
- Για τη χρονομέτρηση θα πρέπει να χρησιμοποιηθεί η κλήση χρονομέτρησης που παρέχει το ίδιο το MPI (`MPI_Wtime()`).
- Τα προγράμματά σας θα πρέπει να τα εκτελέσετε σε τρεις διαφορετικές “εικονικές” μηχανές (που φτιάχνετε μέσω του αρχείου `hostfile`) με 2, 4 και 8 υπολογιστές η κάθε μία, και να συγκρίνετε τους χρόνους εκτέλεσης με τον χρόνο που απαιτήθηκε για το σειριακό πρόγραμμα (βλ. και τις Παρατηρήσεις παρακάτω).
- Για κάθε περίπτωση, το πρόγραμμα θα εκτελείται 4 φορές και ο τελικός χρόνος θα είναι ο μέσος όρος των τεσσάρων χρόνων. Θα πρέπει επομένως να ετοιμάσετε ένα πίνακα αποτελεσμάτων για κάθε ένα από τα προγράμματά σας, ακριβώς όπως στις προηγούμενες εργασίες σας.
- Μαζί με τους παραπάνω πίνακες, θα πρέπει να ετοιμάσετε και μία γραφική παράσταση των (μέσων) χρόνων συναρτήσει του αριθμού των διεργασιών.
- Τέλος, θα πρέπει να σχολιάσετε τους χρόνους που βρήκατε. Στο σχολιασμό θα πρέπει να περιλαμβάνονται και τα παρακάτω: Είναι τα αποτελέσματα κοντά στα ιδεώδη; Γιατί; Πόσο χρόνο σπαταλούν οι διεργασίες στους υπολογισμούς και πόσο στις επικοινωνίες;

Παρατηρήσεις

1. Η ανάπτυξη των προγραμμάτων σας μπορεί να γίνει οπουδήποτε αλλά η εκτέλεση και χρονομέτρηση των πειραμάτων σας θα πρέπει να γίνει σε υπολογιστές του τμήματος. Θα είναι στη διάθεσή σας οι υπολογιστές `opti3060ws18–opti3060ws30`.
2. Σε κάθε κόμβο της εικονικής μηχανής που χρησιμοποιείτε να κάνετε χρήση όλων των πυρήνων. Δηλαδή, αν στην εικονική μηχανή συμμετέχουν 2 κόμβοι, κάθε ένας εκ των οποίων είναι τετραπύρηνος, το πλήθος των διεργασιών που θα πρέπει να δημιουργηθούν είναι $2 \times 4 = 8$ (εκτός βέβαια της περίπτωσης του υβριδικού προγράμματος, όπου θα δημιουργείτε 1 διεργασία ανά κόμβο).
3. Οδηγίες χρήσεως του MPI θα βρείτε στην ιστοσελίδα του μαθήματος. Οι υπολογιστές του τμήματος διαθέτουν 3 υλοποιήσεις του MPI εκ των οποίων η βασική είναι αυτή του OpenMPI.
4. Για τον πολλαπλασιασμό πινάκων, δοκιμάστε πίνακες ακεραίων 2048×2048 στοιχείων (δίνονται στην ιστοσελίδα του μαθήματος). Τα στοιχεία τους θα πρέπει να τα αποθηκεύσετε σε αρχεία ώστε να μπορέσετε να ελέγξετε τη σωστή λειτουργία των προγραμμάτων σας συγκρίνοντας με τα αποτελέσματα του σειριακού (εγγυημένα σωστού) προγράμματος. **Προσοχή:** μην χρονομετρήσετε το μέρος του κώδικα που διαβάζει / γράφει στα αρχεία.
5. Θα πρέπει να χρησιμοποιήσετε (όπου χρειάζεται) τις κλήσεις για συλλογικές επικοινωνίες και θα πρέπει να χρονομετρήσετε ξεχωριστά τον χρόνο που σπαταλήθηκε σε επικοινωνίες.

Προθεσμία παράδοσης:

Παρασκευή, 6 Ιουνίου 2025

Βασίλειος Δημακόπουλος