

Εργασία 3 - Προγραμματισμός με MPI

Ονοματεπώνυμο: Μάριος Γιαννόπουλος

A.M.: 1115200000032

Γενικές Πληροφορίες

Υπολογιστικό Σύστημα

Όλο το έργο υλοποιήθηκε στο ίδιο υπολογιστικό περιβάλλον:

- Όνομα Υπολογιστικού Συστήματος: Linux12
- Επεξεργαστής: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz
- Αριθμός Πυρήνων: 4
- Λειτουργικό Σύστημα: Linux Ubuntu 20.04.2 LTS
- Έκδοση Μεταγλωττιστή: gcc (Ubuntu 9.4.0-1ubuntu1 20.04.2) 9.4.0

Οδηγίες Εκτέλεσης Python Scripts

Για την εκτέλεση των Python scripts που επεξεργάζονται τα αποτελέσματα, ακολουθήστε τα εξής βήματα:

1. Μεταβείτε στον φάκελο `scripts`.
2. Εγκαταστήστε τις απαραίτητες βιβλιοθήκες:

```
pip install -r requirements.txt
```

3. Εκτελέστε το script που σας ενδιαφέρει:

```
python <test_script>.py
```

Σημείωση: Όλα τα αποτελέσματα στα γραφήματα είναι από την εκτέλεση των πειραμάτων στο εργαστήριο Linux. Κάθε πείραμα εκτελέστηκε 5 φορές και τα αποτελέσματα αναφέρονται στο μέσο όρο των επαναλήψεων.

Άσκηση 3.1

Εισαγωγή

Το Παιχνίδι της Ζωής (Game of Life) είναι ένα κυτταρικό αυτόματο που αναπτύχθηκε από τον μαθηματικό John Horton Conway. Είναι ένα μοντέλο που προσομοιώνει την εξέλιξη ενός πληθυσμού κυττάρων σε ένα δισδιάστατο πλέγμα, όπου κάθε κύτταρο μπορεί να βρίσκεται σε μία από δύο καταστάσεις: ζωντανό ή

νεκρό. Οι κανόνες που διέπουν την εξέλιξη του πληθυσμού βασίζονται στον αριθμό των γειτόνων κάθε κυττάρου. Σε αυτή την άσκηση, υλοποιήθηκε μια παράλληλη έκδοση του Παιχνιδιού της Ζωής χρησιμοποιώντας την MPI (Message Passing Interface) για την κατανομή του υπολογιστικού φόρτου μεταξύ πολλαπλών διεργασιών. Η υλοποίηση βασίζεται στη σειριακή έκδοση του αλγορίθμου που αναπτύχθηκε στην Άσκηση 2.1, χωρίς τη χρήση OpenMP.

Συγχρονισμός

Ο συγχρονισμός μεταξύ των διεργασιών είναι ένα κρίσιμο στοιχείο στην παράλληλη υλοποίηση του Παιχνιδιού της Ζωής. Στον κώδικα, ο συγχρονισμός επιτυγχάνεται με τις εξής λειτουργίες:

1. **MPI_Scatter**: Η διεργασία 0 κατανέμει τα δεδομένα του πλέγματος στις υπόλοιπες διεργασίες.
2. **MPI_Sendrecv**: Κάθε διεργασία ανταλλάσσει γραμμές ορίων (ghost rows) με τις γειτονικές της διεργασίες για να υπολογίσει σωστά την επόμενη γενιά.
3. **MPI_Gather**: Η διεργασία 0 συγκεντρώνει τα αποτελέσματα από όλες τις διεργασίες για να ενημερώσει το τελικό πλέγμα.
4. **MPI_Barrier**: Εξασφαλίζει ότι όλες οι διεργασίες έχουν ολοκληρώσει την τρέχουσα γενιά πριν προχωρήσουν στην επόμενη.

Αυτές οι λειτουργίες εξασφαλίζουν ότι οι διεργασίες συνεργάζονται αποτελεσματικά και ότι τα δεδομένα παραμένουν συνεπή κατά τη διάρκεια της εκτέλεσης.

Πειραματική Διαδικασία

• Παραμετροποίηση:

- Μέγεθος πλέγματος: 64×64 , 1024×1024 , 4096×4096 .
- Αριθμός γενιών: 1000.

• Εκτέλεση:

- Τα πειράματα εκτελέστηκαν 5 φορές για κάθε συνδυασμό παραμέτρων.
- Καταγράφηκε ο χρόνος εκτέλεσης για κάθε πείραμα.
- Τα δεδομένα αποθηκεύτηκαν σε CSV αρχείο.

• Αυτοματοποίηση:

- Αναπτύχθηκαν Python scripts για την εκτέλεση των πειραμάτων και την καταγραφή των δεδομένων.
- Χρησιμοποιήθηκαν Python scripts για την επεξεργασία των αποτελεσμάτων και τη δημιουργία γραφημάτων.

Αποτελέσματα

Από τα αποτελέσματα των μετρήσεων, παρατηρούμε τα εξής:

1. Για μικρά μεγέθη πλέγματος (64x64):

- Η χρήση περισσότερων διεργασιών δεν οδηγεί πάντα σε μείωση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 4 διεργασιών οδήγησε σε μεγαλύτερο χρόνο εκτέλεσης (2.20885 s) σε σύγκριση με τη χρήση 2 διεργασιών (1.72495 s). Αυτό οφείλεται στο γεγονός ότι το κόστος επικοινωνίας μεταξύ των διεργασιών είναι υψηλό σε σχέση με τον χρόνο υπολογισμού για τόσο μικρά πλέγματα.

- Η χρήση 8 και 16 διεργασιών οδήγησε σε ακόμη μεγαλύτερους χρόνους εκτέλεσης (7.92210 s και 10.79159 s αντίστοιχα), γεγονός που επιβεβαιώνει ότι για μικρά πλέγματα η παράλληλη εκτέλεση δεν είναι αποτελεσματική.

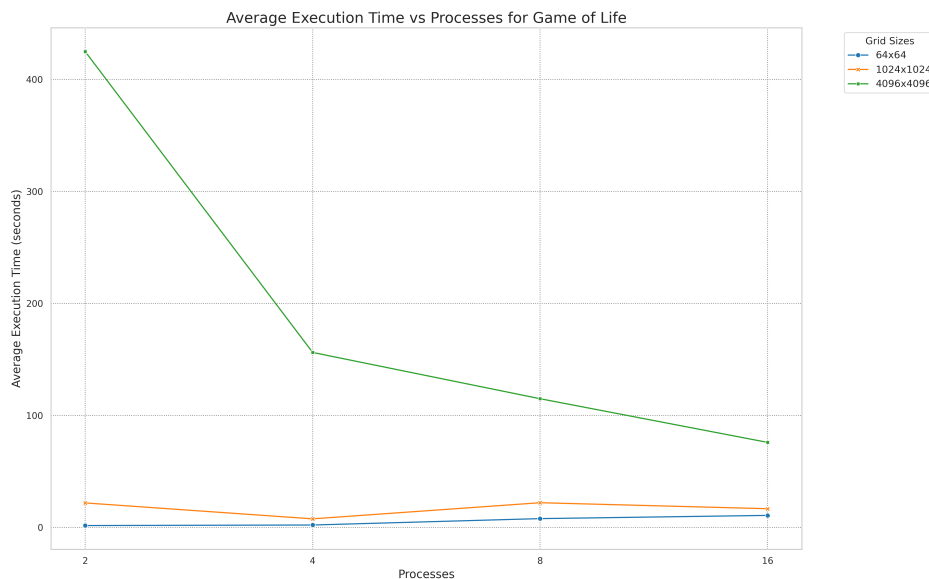
2. Για μεσαία μεγέθη πλέγματος (1024x1024):

- Η χρήση 4 διεργασιών οδήγησε σε σημαντική μείωση του χρόνου εκτέλεσης (7.71714 s) σε σύγκριση με τη χρήση 2 διεργασιών (21.93556 s). Αυτό δείχνει ότι για μεσαία μεγέθη πλέγματος, η παράλληλη εκτέλεση αρχίζει να γίνεται αποτελεσματική.
- Ωστόσο, η χρήση 8 και 16 διεργασιών οδήγησε σε αύξηση του χρόνου εκτέλεσης (22.11091 s και 16.76202 s αντίστοιχα), γεγονός που υποδηλώνει ότι το κόστος επικοινωνίας αρχίζει να γίνεται σημαντικό για μεγαλύτερο αριθμό διεργασιών.

3. Για μεγάλα μεγέθη πλέγματος (4096x4096):

- Η χρήση περισσότερων διεργασιών οδήγησε σε σημαντική μείωση του χρόνου εκτέλεσης. Για παράδειγμα, η χρήση 16 διεργασιών οδήγησε σε μέσο χρόνο εκτέλεσης 75.97231 s, σε σύγκριση με 425.03920 s για 2 διεργασίες.
- Αυτό δείχνει ότι για πολύ μεγάλα πλέγματα, η παράλληλη εκτέλεση είναι πολύ αποτελεσματική, καθώς ο χρόνος υπολογισμού κυριαρχεί έναντι του χρόνου επικοινωνίας.

Γραφήματα



Γράφημα 1: Χρόνοι εκτέλεσης για το Παιχνίδι της Ζωής με MPI

Συμπεράσματα

Από τα αποτελέσματα και τα γραφήματα, μπορούμε να εξάγουμε τα εξής συμπεράσματα:

1. Επίδραση του μεγέθους πλέγματος:

- Για μικρά μεγέθη πλέγματος, η παράλληλη εκτέλεση δεν είναι αποτελεσματική λόγω του υψηλού κόστους επικοινωνίας σε σχέση με τον χρόνο υπολογισμού.
- Για μεσαία και μεγάλα μεγέθη πλέγματος, η παράλληλη εκτέλεση γίνεται ολοένα και πιο αποτελεσματική, καθώς ο χρόνος υπολογισμού κυριαρχεί έναντι του χρόνου επικοινωνίας.

2. Επίδραση του αριθμού των διεργασιών:

- Για μικρά πλέγματα, η αύξηση του αριθμού των διεργασιών οδηγεί σε αύξηση του χρόνου εκτέλεσης λόγω του κόστους επικοινωνίας.
- Για μεγάλα πλέγματα, η αύξηση του αριθμού των διεργασιών οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης, καθώς ο χρόνος υπολογισμού είναι πολύ μεγαλύτερος από το κόστος επικοινωνίας.

3. Βέλτιστος αριθμός διεργασιών:

- Για κάθε μέγεθος πλέγματος, υπάρχει ένας βέλτιστος αριθμός διεργασιών που εξαρτάται από το μέγεθος του πλέγματος και το κόστος επικοινωνίας. Για παράδειγμα, για πλέγμα 1024×1024 , ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι 4, ενώ για πλέγμα 4096×4096 , ο βέλτιστος αριθμός διεργασιών φαίνεται να είναι 16.

Συνολικά, η παράλληλη υλοποίηση του Παιχνιδιού της Ζωής με MPI αποδεικνύεται πολύ αποτελεσματική για μεγάλα προβλήματα, ενώ για μικρά προβλήματα η σειριακή εκτέλεση μπορεί να είναι επαρκής. Η επιλογή του βέλτιστου αριθμού διεργασιών εξαρτάται από το μέγεθος του πλέγματος και το κόστος επικοινωνίας μεταξύ των διεργασιών.

Άσκηση 3.2

Εισαγωγή

Πειραματική Διαδικασία

Αποτελέσματα

Γραφήματα

Συμπεράσματα

Άσκηση 3.3

Εισαγωγή

Πειραματική Διαδικασία

Αποτελέσματα

Γραφήματα

Συμπεράσματα

Άσκηση 3.4

Εισαγωγή

Πειραματική Διαδικασία

Αποτελέσματα

Γραφήματα

Συμπεράσματα