2022-23

ΗΜΜΥ, ΡΟΗ Υ, 9ο εξάμηνο: Συστήματα Παράλληλης Επεξεργασίας

1η εργαστηριακή ασκηση

ομαδα 14 : ΠΑναγιωτησ Νταγκασ, ορφεασ φιλιππολουλοσ, Μαρια-Ηως Γλαρου

1 Σκοπός της Άσκησης

Σκοπός της συγκεκριμένης άσκησης είναι η εξοικείωση με τις υποδομές του εργαστηρίου (πρόσβαση στα συστήματα, μεταγλώττιση προγραμμάτων, υποβολή εργασιών κλπ) μέσα από την παραλληλοποίηση ενός απλού προβλήματος σε αρχιτεκτονικές κοινής μνήμης.

2 Conway's Game of Life

**Υλοποίηση Παραλληλοποίησης**

Η κατάσταση κάθε κελιού καθορίζεται από εκείνη του ιδίου και των γειτόνων του σύμφωνα με τους κανόνες που περιγράφονται στην εκφώνηση. Έτσι με δεδομένη την σειριακή υλοποίηση του παιχνιδιού μπορούμε να παραλληλοποιήσουμε την διαδικασία ενημέρωσής της κατάστασης του κάθε κελιού με την βοήθεια του OpenMP. Συγκεκριμένα, χρησιμοποιήσαμε την εντολή:

pragma omp parallel for shared(N, previous, current) private(i, j, nbrs)

Προσθέτοντας την εντολή αυτή στον κώδικα που υλοποιεί σειριακά το παιχνίδι της ζωής, όπως φαίνεται στο παρακάτω screenshot απαιτούμε την παραλληλοποίηση του εξωτερικού βρόγχου που διασχίζει το grid και φροντίζουμε τα νήματα να μοιράζονται τις μεταβλητές N, previous, current εφόσον μπορούν τα νήματα να έχουν από κοινού πρόσβαση όλα τα νήματα στους πίνακες με τα προηγούμενα και τα τωρινά δεδομένα αλλά όχι τα i, j, nbrs δεδομένου ότι κάθε thread κρατάει και χειρίζεται τους δικούς του δείκτες για την κίνηση στους πίνακες και γράφει στην δική του μεταβλητή nbrs.

Graphical user interface, text

Description automatically generated

**Αποτελέσματα**

Για να ερμηνεύσουμε τα αποτελέσματα οπτικοποιήσαμε τις μετρήσεις με χρήση διαγραμμάτων. Συγκεκριμένα αναπαριστούμε τον χρόνο και το speedup του παράλληλου προγράμματος για ένα μόνο thread σε σχέση με εκείνο με πολλαπλά threads (δηλαδή πλοτάρουμε τον λόγο του χρόνου της παράλληλης υλοποίησης με ένα μόνο thread προς εκείνη με πολλαπλά).

* Grid 64 x 64

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Παρατηρούμε ότι ενώ το speedup σημειώνει σημαντική αύξηση αρχικά, όταν αυξάνουμε τα νήματα από 4 σε 6 η αλλαγή δεν είναι μεγάλη ενώ από 6 σε 8 αυξάνεται ο χρόνος εκτέλεσης. Αυτή η αύξηση ερμηνεύεται από το overhead που προκύπτει από την δημιουργία τέτοιου αριθμού threads, ο οποίος δεν χρειάζεται δεδομένου ότι το grid είναι σχετικά μικρό (64 x 64) και επομένως και λίγες οι γραμμές που παραλληλοποιούμε. Για το μέγεθος αυτό μάλλον δεν μας κερδίζουμε σε χρόνο χρησιμοποιώντας περισσότερα των 6 threads και ανάλογα με τα διαθέσιμα resources ίσως αξίζει να αρκεστούμε στα 4 threads.

* Grid 1024 x 1024

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Παρατηρούμε ότι το speedup είναι σημαντικό όσο αυξάνουμε τον αριθμό των χρησιμοποιούμενων threads. Μάλιστα διπλασιάζοντας των αριθμό των threads διπλασιάζεται και το speedup σε σχέση με την χρήση ενός μόνο thread που είναι η καλύτερη δυνατή επιτάχυνση που θα μπορούσαμε να περιμένουμε με την παραλληλοποίηση.

* Grid 4096 x 4096

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Παρατηρούμε ότι έχουμε επιτάχυνση όσο αυξάνουμε τον αριθμό των threads. Όμως όσο μεγαλύτερη είναι η τιμή του πλήθους των threads στην οποία αναφερόμαστε τόσο μικρότερο είναι το speedup ου πετυχαίνουμε. Ο χρόνος χρησιμοποιώντας 6 και 8 threads βλέπουμε ότι δεν διαφοροποιείται παρά πολύ λίγο. Αυτό συμβαίνει καθώς το κάθε thread πρέπει να φέρει πολλά δεδομένα (4096 bytes) δεδομένου ότι κάθε thread αναλαμβάνει μία γραμμή του grid) στην cache με αποτέλεσμα να προκαλούνται πολλά misses στα επόμενα threads τα οποία καθυστερούν την εκτέλεση του προγράμματος.

Σημείωση: Για να επιβεβαιώσουμε τους ισχυρισμούς μας για το speedup όταν κάθε νήμα χειρίζεται μεγάλο πλήθος δεδομένων δοκιμάσαμε και μεγαλύτερο grid οπότε παρατηρήσαμε ότι η μεταβολή του χρόνου για τις διάφορες τιμές threads είναι παρόμοια.

* Σύγκριση αποτελεσμάτων υλοποίησης παράλληλου προγράμματος με 1 thread με το σειριακό πρόγραμμα

Παρατίθενται στον παρακάτω πίνακα οι τιμές των μετρήσεων.

|  |  |  |  |
| --- | --- | --- | --- |
| Χρόνος (sec) | | | |
| Υλοποίηση\μέγεθος grid | 64 | 1024 | 4096 |
| Σειριακή | 0.020356 | 10.809425 | 173.575566 |
| Παράλληλη | 0.023138 | 10.970185 | 175.962971 |

Μετρώντας τον χρόνο της σειριακής εκτέλεσης του προγράμματος παρατηρούμε λοιπόν ότι είναι ελάχιστα λιγότερος από εκείνον για την παράλληλη εκτέλεση του προγράμματος με ένα thread. Σε καμία περίπτωση δεν έχουμε παραλληλοποίηση των υπολογισμών του παιχνιδιού όμως στην περίπτωση που δουλεύουμε με τις βιβλιοθήκες για τα νήματα έχουμε ένα overhead για την δημιουργία του thread που αναλαμβάνει τους υπολογισμούς που εξηγεί τα αποτελέσματα που προέκυψαν.