ΜΥΕ023-Παράλληλα Σύστημα και Προγραμματισμός

Σετ Ασκήσεων #1

Μποκουράκης Ιωσήφ

E-mail: cs04439@uoi.gr

Όνομα Υπολογιστή	dl380ws02
Επεξεργαστής	Intel Xeon L5520
Πλήθος πυρήνων	4
Μεταφραστής	gcc v11.2.0

Το 1ο σετ ασκήσεων αφορά στον παράλληλο προγραμματισμό με το μοντέλο κοινοχρήστου χώρου διευθύνσεων μέσω του προτύπου OpenMP. Ζητείται η παραλληλοποιήση 3 εφαρμογών. 1η Η παραλληλοποιήση εύρεσης πρώτων αριθμών. 2η Θόλωση εικόνων με τον αλγόριθμο Gaussian Blur. Και 3η μια περιγραφή και άσκηση για την οδηγία της OpenMP taskloop.

Άσκηση 1:

Το πρόβλημα

Στην άσκηση αυτή ζητείται να παραλληλοποιηθει η εύρεση πρώτων. Αυτό πρέπει να γίνει παραλληλοποιόντας τους υπολογισμούς που γίνονται στο αντίστοιχο σειριακό πρόγραμμα.

Μέθοδος Παραλληλοποιήση

Χρησιμοποιήθηκε το σειριακό πρόγραμμα από την ιστοσελίδα του μαθήματος. Για την παραλληλοποιήση του βρόγχου προστέθηκε η

οδηγία #pragma omp parallel for private(i, num, divisor, quotient, remainder) num_threads(i) και #pragma omp critical. Η πρώτη οδηγία μπαίνει πριν την for επανάληψη. Οι μεταβλητές I, num, divisor, quotient, remainder, πρεπει να είναι ιδιωτικες ώστε να έχουν τον δικό τους χώρο και να μην επηρεάζουν τις υπόλοιπες μεταβλητές. Ενώ οι μεταβλητές τις αλλάζει κάθε νήμα διαφορετικά και για την ασφάλεια προσθέτουμε το την δεύτερη οδηγία ώστε να έχουμε αμοιβαίο αποκλεισμό.

Πειραματικά αποτελέσματα – μετρήσεις

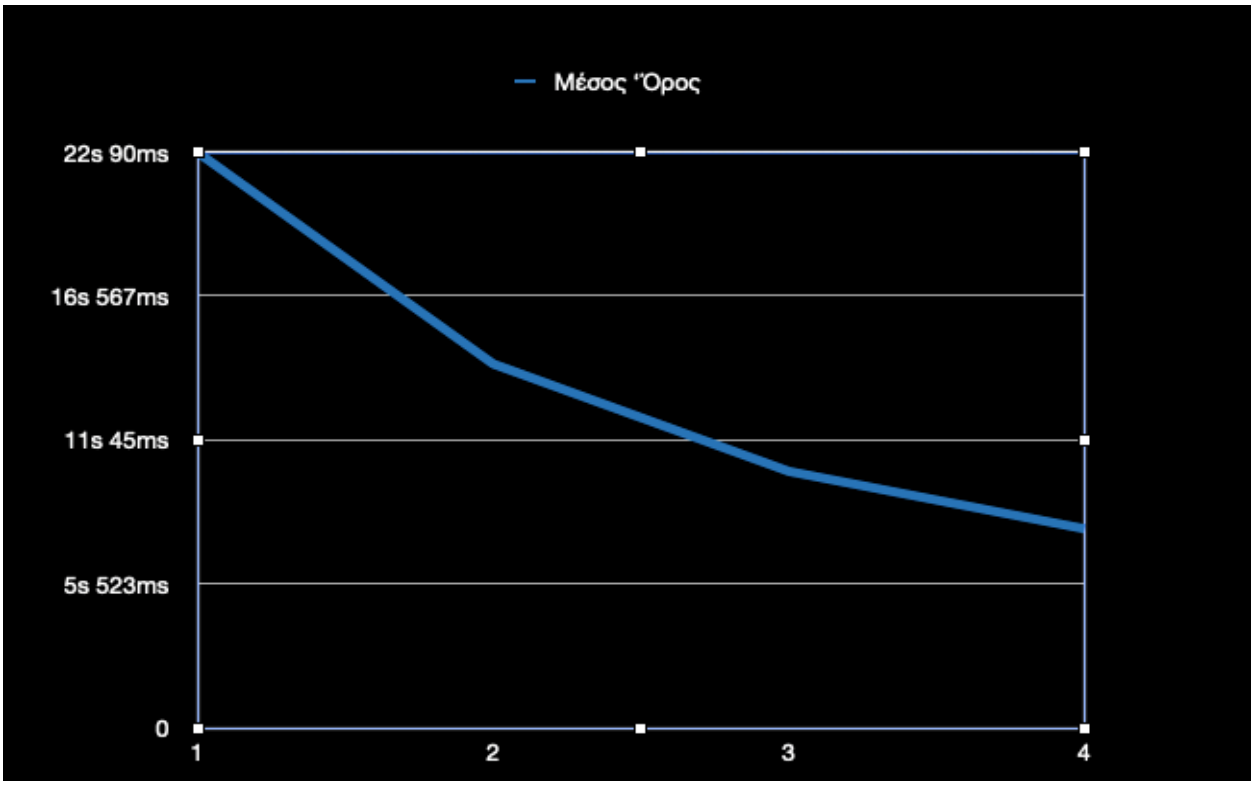
Τα προγράμματα εκτελέστηκαν στο σύστημα που αναφέραμε στην παραπάνω εικόνα και η χρονομέτρηση έγινε χρησιμοποιώντας την συνάρτηση omp_get_wtime()

Χρησιμοποιήσαμε από 1 έως 4 νήματα.

Κάθε πείραμα εκτελέστηκε 4 φορές και υπολογίστηκαν οι μεσοί οροί για κάθε νήμα μετα τις 4 εκτελέσεις στο καθένα. Τα αποτελέσματα φαίνονται στον πίνακα παρακάτω.

Νήματα	1η Εκτέλεση	2η Εκτέλεση	3η Εκτέλεση	4η Εκτέλεση	Μέσος Όρος
1	22s 46ms	22s 8ms	22s 147ms	22s 121ms	22s 81ms
2	14s 62ms	13s 977ms	13s 880ms	13s 938ms	13s 964ms
3	9s 870ms	9s 832ms	9s 827ms	9s 835ms	9s 841ms
4	7s 546ms	7s 779ms	7s 581ms	7s 645ms	7s 638ms

Με βαση τους πινακες προκυπτει τον παρακατω γραφημα.



Σχόλια

Με βάση τα αποτελέσματα βλέπουμε ότι η αύξηση του αριθμού νημάτων μειώνει σημαντικά τους χρόνους εκτέλεσης. Επίσης η πολιτική schedule έχει επίδραση στον διαμοιρασμό των επαναλήψεων στα νήματα.

Όλα τα παραπάνω είναι μάλλον αναμενόμενά διότι οι επαναλήψεις μοιράστηκαν ισόποσα και κάθε μια είχε παρόμοιο φόρτο υπολογισμών με τις άλλες. Επομένως σε κάθε νήμα αντιστοιχεί ίσος αριθμών επαναλήψεων (τουλάχιστον όσο ίσος αριθμός είναι δυνατός).

Άσκηση 2:

Το Πρόβλημα

Στο πρόβλημα αυτό ζητείτε να γίνει θόλωση μιας εικόνας χρησιμοποιώντας το Gaussian Blur και να παραλληποιηθει η διαδικασία χρησιμοποιώντας την βιβλιοθήκη OpenMP. Πρέπει να παραλληλοποιηθει η συνάρτηση gaussian_blur_omp_loops() και η συνάρτηση gaussian_blur_omp_tasks().

Μέθοδος Παραλληλοποιήση

Χρησιμοποιήθηκε το σειριακό πρόγραμμα από την ιστοσελίδα του μαθήματος. Για την παραλληλοποιήση της κάθε συνάρτησης έχουν χρησιμοποιηθεί οι εντολές που αναφέρονται παρακάτω.

Για την παραλληλοποιήση της συνάρτησης gaussian_blur_omp_loops() χρησιμοποιήθηκε η εντολή

#pragma omp parallel for private(j, row, col, weightSum, redsum, greenSum, blueSum) num_threads(i) οπου i είναι ο αριθμος των επιθυμητων νηματων.

Για την παραλληλοποιήση της συνάρτησης gaussian_blur_omp_tasks() χρησιμοποιήθηκαν οι εντολές.

#pragma omp single πριν από τις επαναλήψεις. Και μέσα στις επαναλήψεις χρησιμοποιήθηκε η εντολή.

#pragma omp task firstprivate(i, j)

Πειραματικά αποτελέσματα – μετρήσεις

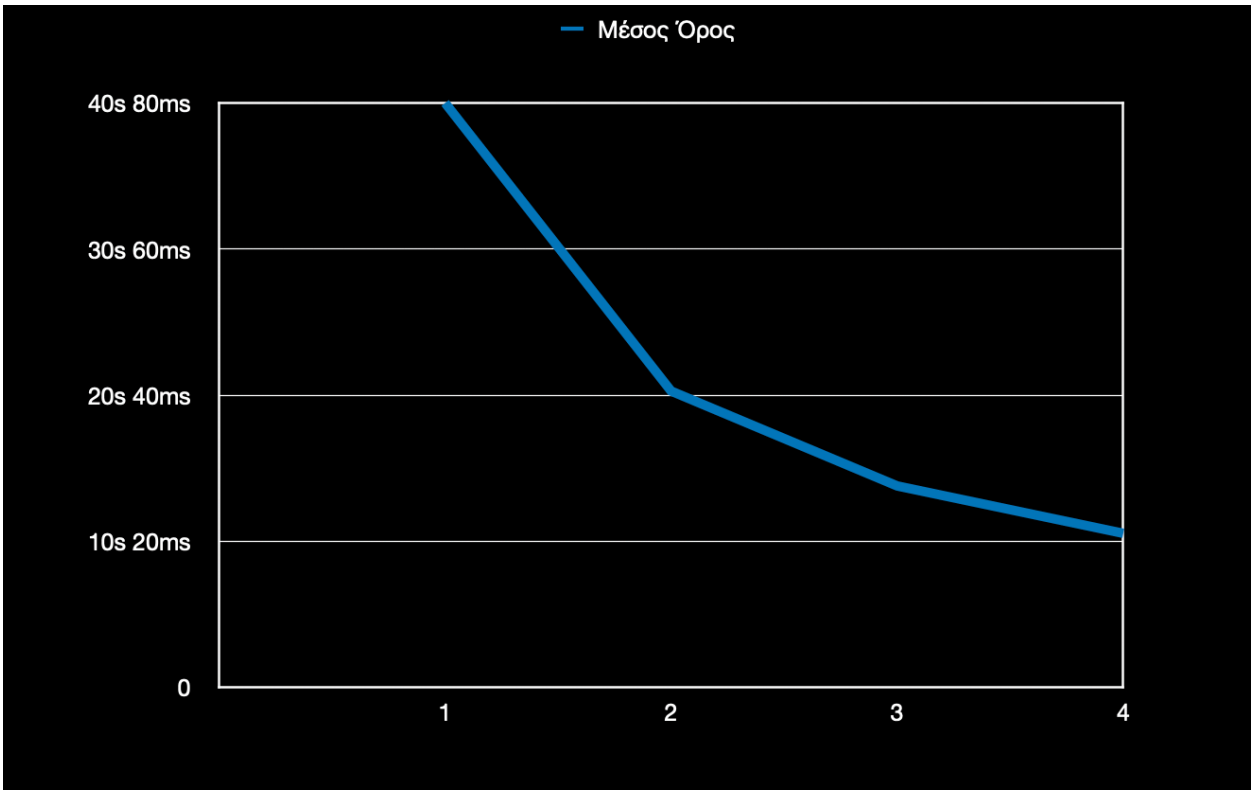
Τα προγράμματα εκτελέστηκαν στο σύστημα που αναφέραμε στην εισαγωγή και η χρονομέτρηση έγινε με την συνάρτηση gettimeofday().

Χρησιμοποιήθηκαν από 1 μέχρι 4 νήματα και κάθε πείραμα εκτελεστικέ 4 φορές και υπολογίστηκαν οι μέσοι οροί. Τα αποτελέσματα δίνονται παρακάτω στον πίνακα (οι χρόνοι είναι σε seconds).

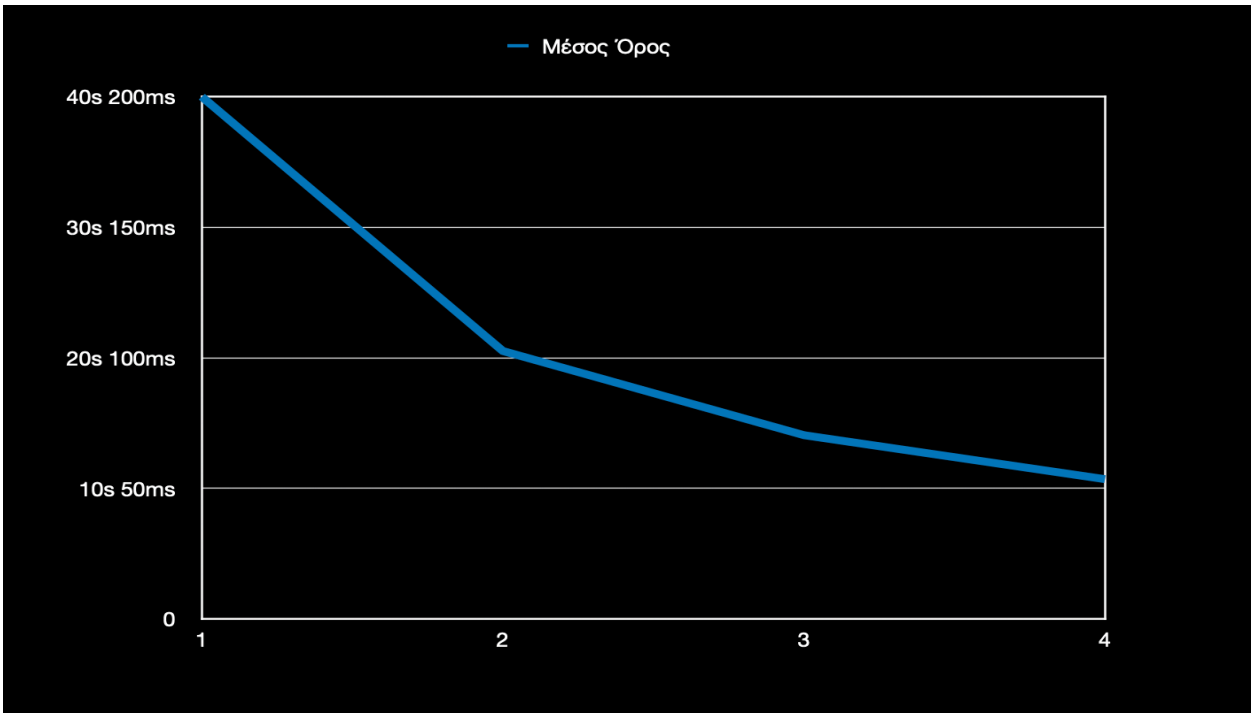
Νήματα	η Εκτέλεση	3η Εκτέλεση	4η Εκτέλεση	Μέσος Όρος
Omp loops				
1	40s 510ms	40s 30ms	39s 950ms	40s 78ms
2	20s 290ms	20s 330ms	20s 380ms	20s 315ms
3	13s 840ms	13s 820ms	13s 770ms	13s 810ms
4	10s 700ms	10s 500ms	10s 510ms	10s 560ms
Omp tasks				
1	39s 990ms	40s 210ms	40s 280ms	40s 185ms
2	20s 560ms	20s 670ms	20s 580ms	20s 617ms
3	14s 70ms	14s 30ms	14s 320ms	14s 140ms
4	10s 700ms	10s 900ms	10s 750ms	10s 755ms

Με βάση τον παραπάνω πίνακα βγαίνουν και οι δυο γραφικές παραστάσεις που φαίνονται παρακάτω.

Η πρώτη γραφική παράσταση είναι για την συνάρτηση gaussian_blur_omp_loops(). Η δεύτερη γραφική παράσταση είναι για την συνάρτηση είναι για την συνάρτηση gaussian_blur_omp_tasks().



.Γραφική Παράσταση 1



.Γραφική Παράσταση 2

Σχόλια

Με βάση τα αποτελέσματα βλέπουμε ότι η αύξηση του αριθμού των νημάτων μειώνει σημαντικά τον χρόνο εκτέλεσης του προγράμματος.

Όλα τα παραπάνω είναι αναμενόμενά διότι οι επαναλήψεις μοιράστηκαν ισόποσα (όσο το δυνατόν περισσότερο γίνεται).

Άσκηση 3:

Στην άσκηση αυτή μας ζητείται να μελετήσουμε την οδηγία taskloop που υποστηρίζει η βιβλιοθήκη openMP καθώς και να συντάξουμε ένα μικρό δοκιμαστικό πρόγραμμα που να δείχνει την λειτουργία της.

Περιγραφή

Η οδηγία taskloop είναι μια task generated οδηγία. Οπότε ένα νήμα βρίσκε μια οδηγία taskloop, η οδηγία διαμερίζει τις επαναλήψεις σε συγκεκριμένες οδηγίες για παράλληλη εκτέλεση. Τα δεδομένα από την παραγόμενη διαδικασία δημιουργουνται σύμφωνα με τον διαμερισμό δεδομένων που έχει οριστεί στην οδηγία taskloop. Η σειρά δημιουργίας των επαναλήψεων είναι απροσδιόριστη. Τα προγράμματα που βασίζονται στην σειρά εκτέλεσης λογικών επαναλήψεων είναι μη συμβατά. Από προεπιλογή η οδηγια taskloop εκτελεί σαν να ήταν σε κλειστό πεδίο. Άμα το nongroup υπάρχει τότε κανένα συγκεκριμένο πεδίο δημιουργείτε.

Εάν το reduction υπάρχει τότε το taskloop συμπεριφέρεται σαν να είναι task_reduction με το ίδιο τελεστή μείωσης και απαριθμεί τα στοιχεία που εφάρμοσε στο συγκεκριμενο taskgrou που περικυκλώνει την οδηγία taskloop. Η οδηγία taskloop εκτελεί σαν κάθε μια από τις παραγόμενες εργασίες η μέλη από το reduction από τον ορό task_reduction.

Εάν υπάρχει ο ορός grainsize τότε το νούμερο από κάθε λογική επανάληψη που ανατέθηκε σε κάθε εργασία είναι μεγαλύτερο ή ίσο από την ελάχιστη τιμή της grain-size έκφρασης και το νουμερο από τις λογικές επαναλήψεις άλλα λιγότερο από το διπλάσιο της τιμής της grain-size έκφρασης.

Η παράμετρος του όρου grainsize πρέπει να είναι ενας θετικός ακέραιος αριθμός. Εάν το num_tasks είναι ορισμένο, η οδηγία taskloop δημιουργεί όσες εργασίες όσο η ελάχιστη τιμή της έκφρασης num_tasks και της τιμής των λογικών επαναλήψεων.

Ο ορός collapse μπορεί να χρησιμοποιηθεί για να ορίσει το πόσες επαναλήψεις μπορούν να συσχετιστούν με την οδηγία taskloop. Η παράμετρος αυτή πρέπει να είναι ένας θετικός ακέραιος αριθμός. Εάν δεν υπάρχει ο όρος είτε εάν ισούται με 1 τότε μόνο μια επανάληψη μπορεί να συσχετιστεί με την οδηγία taskloop.

Μια επανάληψη τύπου taskloop έχει μέγεθος από 1 μέχρι N-1 όπου N είναι ο αριθμός των επαναλήψεων. Και η λογική αρίθμηση δείχνει την σειρά με την οποία οι επαναλήψεις θα εκτελεστούν.

Πρόγραμμα Εμπεδωσης

Ένα πρόγραμμα που δείχνεις την λειτουργία της οδηγίας taskloop δίνεται μαζί με τα αρχεία κώδικά από τις δυο προηγούμενες ασκήσεις. Το όνομα του αρχείου είναι taskloopExample.c και είναι ένα πρόγραμμα το οποίο υπολογίζει το άθροισμα τετράγωνων των στοιχείων μέσα σε μια λίστα. Και για να γίνει αυτό χρησιμοποιούμε την οδηγία taskloop.