

## E-85: Ε.Θ.Α: Προγραμματισμός Συστημάτων Υψηλών Επιδόσεων Χειμερινό Εξάμηνο 2009-2010

### 2<sup>ο</sup> Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Παρασκευή 22/1/2010, 20:00

#### 1. Μέτρηση των overheads του OpenMP

Στα πλαίσια της πρώτης άσκησης θα πρέπει να μετρήσετε τις καθυστερήσεις κάποιων βασικών οδηγιών του μοντέλου προγραμματισμού OpenMP. Για το σκοπό αυτό θα πρέπει να χρησιμοποιήσετε τα διαθέσιμα EPCC microbenchmarks, τα οποία είναι διαθέσιμα στη σελίδα [http://www2.epcc.ed.ac.uk/computing/research\\_activities/openmpbench/openmp\\_index.html](http://www2.epcc.ed.ac.uk/computing/research_activities/openmpbench/openmp_index.html).

Θα πρέπει να εγκαταστήσετε τα microbenchmarks, να θέσετε σωστά το Makefile για τον OpenMP compiler που έχετε στη διάθεσή σας και να μετρήσετε τις καθυστερήσεις 3 οδηγιών του OpenMP χρησιμοποιώντας το πρόγραμμα syncbench των microbenchmarks. Επιλέξτε οι ίδιοι ποιες οδηγίες θα εξετάσετε από που είναι διαθέσιμες στον κώδικα. Οι μετρήσεις θα γίνουν χρησιμοποιώντας 1, 2, 4 και 8 OpenMP threads (OMP\_NUM\_THREADS) και θα πρέπει να παρουσιαστούν σε πίνακες και αντίστοιχες γραφικές παραστάσεις.

#### 2. Πολυνηματική συνάρτηση

Έστω η συνάρτηση **find\_max**, η οποία υπολογίζει το μέγιστο στοιχείο καθώς και τη θέση ενός διδιάστατου πίνακα A μεγέθους NxN. Επομένως, το πρωτότυπο της συνάρτησης είναι ως εξής: **double find\_max(double \*\*A, int N, int \*maxi, int \*maxj)**; Υλοποιήστε την παράλληλη έκδοση της συνάρτησης χρησιμοποιώντας το μοντέλο προγραμματισμού OpenMP.

#### 3. Προγραμματιστικό παράδειγμα αφέντη – εργάτη

Στην άσκηση αυτή θα πρέπει να παραλληλοποιήσετε την παρακάτω εφαρμογή χρησιμοποιώντας MPI και το προγραμματιστικό παράδειγμα αφέντη-εργάτη. Συγκεκριμένα, μία MPI διεργασία λειτουργεί ως αφέντης, εκτελώντας τον κώδικα της main και μοιράζοντας δυναμικά δουλιές (επαναλήψεις του for) στις υπόλοιπες MPI διεργασίες – εργάτες. Κάθε διεργασία-εργάτης δέχεται τα δεδομένα εισόδου, εκτελεί τη συνάρτηση και επιστρέφει τα αποτελέσματα στον αφέντη. Αυτό επαναλαμβάνεται μέχρι να ολοκληρωθούν όλες οι εργασίες, οπότε ο αφέντης προχωρεί στον τερματισμό των εργατών.

---

```
#include <stdio.h>
#include <unistd.h>

void work(int x, int *y)
{
    sleep(x);
    *y = x*x;
}

int main()
{
    int I, N, *Y;
    scanf("%d", &N);
    Y = malloc(N*sizeof(int));

    for (I = 0; I < N; I++) work(I, &Y[I]);

    for (I = 0; I < N; I++) print("%d^2 = %d\n", I, Y[I]);

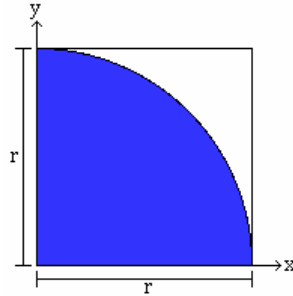
    return 0;
}
```

---

- α) Βεβαιωθείτε για την ορθή και αποδοτική λειτουργία της υλοποίησής σας.  
 β) Χρονομετρήστε και παρουσιάστε σε πίνακες την απόδοση της υλοποίησης σας για διάφορες τιμές του αριθμού των δουλειών (N) και των διεργασιών-εργατών (P).

#### 4. Υπολογισμός του $\pi$

Αντικείμενο της άσκησης είναι ο υπολογισμός του  $\pi$  ( $\approx 3.14..$ ) με χρήση OpenMP και MPI. Για τον παραπάνω σκοπό θα χρησιμοποιηθεί μία μέθοδος Monte Carlo, που υπολογίζει την τιμή του  $\pi$  με τη βοήθεια τυχαίων αριθμών και στατιστικών πιθανοτήτων. Επιλέγοντας τυχαία σημεία μέσα στο τετράγωνο του παρακάτω σχήματος, αποδεικνύεται εύκολα ότι ο λόγος του αριθμού των σημείων που βρίσκονται στην χρωματισμένη περιοχή προς το συνολικό αριθμό τους ισούται με  $\frac{1}{4} \pi$ . Από τη παραπάνω σχέση προκύπτει ο τρόπος υπολογισμού της τιμής του  $\pi$ .



Το παρακάτω πρόγραμμα υπολογίζει την τιμή του  $\pi$ , θεωρώντας στην παραπάνω μέθοδο ότι η ακτίνα  $r$  ισούται με τη μονάδα.

---

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

const double pi_true = 3.1415926535897932384626433;

int main()
{
    double x, y, dist, pi;
    unsigned long i, n, hits = 0;

    printf("give n:");
    scanf("%ld", &n);

    srand(time(NULL));
    for (i = 0; i < n; i++) {
        x = drand48(); y = drand48();
        dist = sqrt(x*x + y*y);
        if (dist <= 1.0) hits++;
    }

    pi = 4.0*((double)hits/(double)n);

    printf ("PI = %lf\n", pi);
    printf ("error = %lf\n", fabs(pi-pi_true));

    return 0;
}
```

---

Παραλληλοποιήστε τον παραπάνω κώδικα σύμφωνα με τα ακόλουθα μοντέλα προγραμματισμού:

- α) OpenMP  
 β) MPI  
 γ) MPI + OpenMP

Χρονομετρήστε και παρουσιάστε σε πίνακες την απόδοση της υλοποίησης σας.

## 5. Παραλληλοποίηση Εφαρμογής

Στην άσκηση αυτή θα πρέπει να παραλληλοποιήσετε έναν κώδικα που υλοποιεί μία τεχνική Βελτιστοποίησης με Σμήνη Σωματιδίων (Particle Swarm Optimization). Η σειριακή έκδοση του κώδικα είναι διαθέσιμη στη σελίδα του μαθήματος, ενώ η παραλληλοποίηση του κώδικα θα πρέπει να γίνει με το μοντέλο προγραμματισμού OpenMP. Αιτιολογήστε τις αποφάσεις παραλληλοποίησης που πήρατε και μετρήστε το χρόνο εκτέλεσης για αριθμό νημάτων ίσο με 1, 2, 3 και 4.

## 6. Παραδοτέα

Η παράδοση της άσκησης πρέπει να γίνει ηλεκτρονικά με αποστολή στη διεύθυνση ηλεκτρονικού ταχυδρομείου του διδάσκοντα (**phadjido (at) cs.uoi.gr**) μέχρι τη **Παρασκευή 22/1/2010** και ώρα **8μμ. (20:00)**. Μετά την συγκεκριμένη διορία, δεν θα γίνεται δεκτή καμία άσκηση. Το θέμα του μηνύματος θα είναι **E-85-EX2 (AM:X1,X2)**, όπου X1, X2 οι αριθμοί μητρώου του κάθε μέλους της ομάδας σας. Συνημμένα θα υπάρχει συμπιεσμένο αρχείο το οποίο θα περιέχει:

1. Πηγαίο κώδικα, γραμμένο σε C, για κάθε περίπτωση, με τα απαραίτητα **σχόλια**.
2. **Αναλυτική αναφορά** για την εργασία σας όπου θα συμπεριλαμβάνεται το ονοματεπώνυμό σας, ο αριθμός μητρώου σας και η διεύθυνση ηλεκτρονικού ταχυδρομείου που χρησιμοποιείτε. Θα πρέπει να αναφέρετε τα βασικά χαρακτηριστικά του υπολογιστικού συστήματος στο οποίο εκτελέσατε τα προγράμματά σας. Στην αναφορά θα τεκμηριώνετε την υλοποίηση και τις σχεδιαστικές αποφάσεις σας, θα παρουσιάσετε τα αποτελέσματά σας, ενώ μπορείτε να προσθέσετε οτιδήποτε θεωρήσετε σημαντικό.

Οι ασκήσεις θα πραγματοποιηθούν σε ομάδες 1-2 ατόμων. Οι ομάδες αυτές θα ισχύσουν και τα επόμενα σύνολα ασκήσεων. Σε περίπτωση αντιγραφής η άσκηση θα μηδενίζεται.

Για τυχόν απορίες ή διευκρινήσεις μπορείτε να χρησιμοποιήσετε τις ώρες διδασκαλίας του μαθήματος ή να επικοινωνήσετε μαζί μου μέσω e-mail ή προσωπικά. Αν απαιτείται, θα προστεθούν διευκρινήσεις, βοηθητικές πληροφορίες και απαντήσεις στην ιστοσελίδα του μαθήματος.

**Καλή επιτυχία!**

**Παναγιώτης Χατζηδούκας (Π.Δ. 407/80)**