Εθνικο & Καποδιστριακο Πανεπιστημιο Αθηνων
Τμημα Πληροφορικης & Τηλεπικοινωνιων
Μαθημα: **Παραλληλα Συστηματα**

Εργασία 2 (υποχρεωτική) – Προγραμματισμός με OpenMP

Ακαδημαϊκο Ετος 2023 – 2024
(εκφωνηση) Σαββατο 9 Δεκεμβριου 2023
(παραδοση στο eclass μεχρι) **Παρασκευη 22 Δεκεμβριου 2023**

Πληροφορίες για τις Υποχρεωτικές Εργασίες του μαθήματος

- Κάθε ομάδα μπορεί να αποτελείται **από 1 ή 2 φοιτητές**. Όλα τα μέλη της ομάδας πρέπει να έχουν ισότιμη συμμετοχή και να γνωρίζουν τις λεπτομέρειες της υλοποίησης της ομάδας.
- Για την εξεταστική Σεπτεμβρίου δε θα δοθούν άλλες εργασίες. Τον Σεπτέμβριο εξετάζεται μόνο το γραπτό.
- Επίσημη υπολογιστική πλατφόρμα του μαθήματος είναι το δίκτυο των υπολογιστών του Τμήματος με λειτουργικό σύστημα Linux Ubuntu (linux01.di.uoa.gr έως linux30.di.uoa.gr).
- Μαζί με τον κώδικά σας καλείστε να υποβάλετε και τα σχετικά αρχεία Makefile.
- Στην αναφορά σας καλείστε να δώσετε πληροφορίες σχετικά με το όνομα του υπολογιστικού συστήματος που χρησιμοποιείτε, καθώς επίσης και το μοντέλο επεξεργαστή, τον αριθμό των πυρήνων, την έκδοση του λειτουργικού συστήματος, και την έκδοση του μεταγλωττιστή. Για τα πειραματικά δεδομένα που παρουσιάζετε, καλείστε να αναφέρετε ρητά στα εκάστοτε σημεία της αναφοράς τις εισόδους που χρησιμοποιήσατε. Καλείστε να εκτελέσετε κάθε πείραμα (το οποίο ορίζεται ως η εκτέλεση ενός προγράμματος για συγκεκριμένο αριθμό νημάτων και παραμέτρους εισόδου) πολλές φορές (για παράδειγμα, 4 φορές) και να παρουσιάσετε τον μέσο όρο των αποτελεσμάτων (σύσταση: δημιουργήστε scripts για την εκτέλεση των πειραμάτων, ακόμα και για την επεξεργασία των αποτελεσμάτων και την δημιουργία γραφημάτων).
- Προαιρετικά, μπορείτε να υποβάλετε και τα scripts που χρησιμοποιήσατε για να τρέξετε τα πειράματα και να δημιουργήσετε τα σχετικά γραφήματα.
- Καλείστε να προσεγγίσετε την κάθε άσκηση στην αναφορά σας ως εξής: περιγραφή προβλήματος, σύντομη περιγραφή της λύσης σας, παράθεση πειραματικών αποτελεσμάτων (χρήση πινάκων ή γραφημάτων), και σχολιασμός αποτελεσμάτων.
- Σε περίπτωση αντιγραφής θα μηδενίζονται όλες οι ομάδες που μετέχουν σε αυτή.
- Η παράδοση της **Εργασίας** πρέπει να γίνει μέχρι τα **μεσάνυχτα της προθεσμίας ηλεκτρονικά** και μόνο στο eclass (να ανεβάσετε ένα μόνο αρχείο zip ή rar με την αναφορά σας σε PDF και τον κώδικά σας). **Μην περιμένετε μέχρι την τελευταία στιγμή.**

Εργασία 2.1

Γράψτε ένα πρόγραμμα OpenMP που να χρησιμοποιεί τη μέθοδο Μόντε Κάρλο για την εκτίμηση της τιμής του π (για την περιγραφή του αλγορίθμου, δείτε την εκφώνηση της Εργασίας 1.1). Το πρόγραμμα θα πρέπει να παίρνει το πλήθος των ρίψεων από τον χρήστη πριν την εκκίνηση νημάτων. Χρησιμοποιήστε έναν όρο reduction για τον υπολογισμό του πλήθους των βελών που πετυχαίνουν το εσωτερικό του κύκλου και φροντίστε ώστε η εκτύπωση του αποτελέσματος να γίνεται μετά την ένωση όλων των νημάτων. Προτιμήστε τον τύπο long long int για το πλήθος των βελών εντός του κύκλου και για το πλήθος των ρίψεων, αφού και οι δύο αυτοί αριθμοί θα πρέπει να είναι πολύ μεγάλοι (για παράδειγμα, το πλήθος ρίψεων να είναι 10^8 ή 10^9) για να πάρετε μια καλή εκτίμηση για το π.

Συγκρίνετε την απόδοση του παράλληλου προγράμματος σας που χρησιμοποιεί OpenMP για διαφορετικό αριθμό νημάτων και διαφορετικό αριθμό πλήθος ρίψεων σε σχέση με τον σειριακό αλγόριθμο. Παρατηρείτε επιτάχυνση και γιατί; Συγκρίνετε επίσης την απόδοση του παράλληλου προγράμματος σας που χρησιμοποιεί OpenMP με το αντίστοιχο παράλληλο πρόγραμμα που γράψατε για την Εργασία 1.1 με Pthreads. Παρατηρείτε διαφορές στην επίδοση;

Άσκηση 2.2

Σε αυτή την άσκηση καλείστε να τροποποιήσετε το πρόγραμμα πολλαπλασιασμού μήτρας-διανύσματος του βιβλίου «Εισαγωγή στον Παράλληλο Προγραμματισμό» που σας δίνεται (omp_mat_vect_rand_split.c), ώστε να υπολογίζει αποδοτικά (δηλαδή να αποφεύγει αχρείαστους υπολογισμούς) τον πολλαπλασιασμό άνω τριγωνικού πίνακα με διάνυσμα, με χρήση OpenMP. Υπενθυμίζεται ότι άνω (κάτω) τριγωνικός λέγεται ένας πίνακας στον οποίο είναι μη μηδενικά μόνο τα στοιχεία πάνω (κάτω) από την κύρια διαγώνιο, καθώς και αυτά της κύριας διαγωνίου. Παρατηρείτε διαφορές στην επίδοση καθώς αυξάνεται ο αριθμός των νημάτων; Μπορείτε να βελτιώσετε την επίδοση μέσω της επιλογής ανάθεσης επαναλήψεων σε νήματα; Ποια είναι η επιλογή που μειώνει περισσότερο το χρόνο εκτέλεσης; Ίσως σας φανεί χρήσιμος ο όρος schedule(runtime) και η μεταβλητή περιβάλλοντος OMP_SCHEDULE.

Εργασία 2.3

Για την επίλυση μεγάλων γραμμικών συστημάτων, συχνά χρησιμοποιούμε την απαλοιφή Gauss ακολουθούμενη από μια αντικατάσταση προς τα πίσω. Η απαλοιφή Gauss μετατρέπει ένα γραμμικό σύστημα n × n σε άνω τριγωνικό χρησιμοποιώντας τις παρακάτω πράξεις στις γραμμές:

- Πρόσθεση ενός πολλαπλάσιου μίας γραμμής σε άλλη γραμμή
- Αντιμετάθεση δύο γραμμών
- Πολλαπλασιασμό μίας γραμμής με μια μη μηδενική σταθερά

Ένα άνω τριγωνικό σύστημα έχει μόνο μηδενικούς συντελεστές κάτω από τη διαγώνιο που εκτείνεται από την άνω αριστερή γωνία τού συστήματος στην κάτω δεξιά.

Για παράδειγμα, το γραμμικό σύστημα:

$$\begin{aligned} 2x_0 - 3x_1 &= 3 \\ 4x_0 - 5x_1 + x_2 &= 7 \\ 2x_0 - x_1 - 3x_2 &= 5 \end{aligned}$$

μπορεί να απλοποιηθεί στην άνω τριγωνική μορφή:

$$\begin{aligned} 2x_0 - 3x_1 &= 3 \\ x_1 + x_2 &= 1 \\ -5x_2 &= 0 \end{aligned}$$

Θεωρούμε ότι η «δεξιά πλευρά» του συστήματος αποθηκεύεται στη συστοιχία b, οι συντελεστές των αγνώστων αποθηκεύονται στη διδιάστατη συστοιχία A, και οι λύσεις αποθηκεύονται στη συστοιχία x. Ο παρακάτω αλγόριθμος υλοποιεί την απαλοιφή Gauss:

```
for (i = 0; i < n-1; i++) {
      for (j = i+1; j < n; j++) {
            ratio = A[j][i]/A[i][i];
            for (k = i; k < n; k++)
                  A[j][k] -= (ratio*A[i][k]);
            b[j] -= (ratio*b[i]);
      }
}
```

Χρησιμοποιήστε την OpenMP για να υλοποιήσετε ένα παράλληλο πρόγραμμα που εφαρμόζει την απαλοιφή Gauss. Το πρόγραμμά σας θα πρέπει να δέχεται σαν ορίσματα: (α) το μέγεθος του γραμμικού συστήματος n, (β) το αν θα εκτελεστεί ο σειριακός ή ο παράλληλος αλγόριθμος της ταξινόμησης, και (γ) τον αριθμό των νημάτων που θα χρησιμοποιηθούν στην παράλληλη απαλοιφή. Θεωρήστε ότι το σύστημα που το πρόγραμμά σας θα δέχεται ως είσοδο δεν θα χρειάζεται αντιμεταθέσεις γραμμών. Εξετάστε ποιος βρόχος του αλγορίθμου που υλοποιεί την απαλοιφή Gauss μπορεί να παραλληλοποιηθεί και συμφέρει περισσότερο όσον αφορά την επίδοση του προγράμματός σας.

Στη συνέχεια, αυτό το σύστημα μπορεί να λυθεί εύκολα: βρίσκουμε το x_2 χρησιμοποιώντας την τελευταία εξίσωση, μετά βρίσκουμε το x_1 από τη δεύτερη εξίσωση και, τέλος, βρίσκουμε το x_0 χρησιμοποιώντας την πρώτη εξίσωση. Μπορούμε να χρησιμοποιηθούν διάφοροι σειριακοί αλγόριθμοι για την αντικατάσταση προς τα πίσω. Ένας τέτοιος αλγόριθμος που διατρέχει το σύστημα πρώτα κατά γραμμή έχει τη μορφή:

```
for (row = n-1; row >= 0; row--) {
      x[row] = b[row];
      for (col = row+1; col < n; col++)
            x[row] -= A[row][col]*x[col];
      x[row] /= A[row][row];
}
```

Εξετάστε αν ο εξωτερικός βρόχος του «κατά γραμμή» αλγορίθμου μπορεί να παραλληλοποιηθεί. Αντίστοιχα, εξετάστε αν ο εσωτερικός βρόχος του «κατά γραμμή» αλγορίθμου μπορεί να παραλληλοποιηθεί.

Επεκτείνετε το προηγούμενο πρόγραμμα OpenMP που γράψατε για την απαλοιφή Gauss ώστε να υλοποιεί και την επίλυση του συστήματος παράλληλα. Ίσως σας φανεί χρήσιμη η οδηγία single – όταν ένα μπλοκ κώδικα εκτελείται παράλληλα και ένα υπομπλοκ στο εσωτερικό του πρέπει να εκτελεστεί από ένα μόνο νήμα, το υπομπλοκ μπορεί να προσδιοριστεί με μια οδηγία #pragma omp single. Τα νήματα της ομάδας που εκτελεί τον κώδικα θα μείνουν μπλοκαρισμένα στο τέλος του υπο-μπλοκ κώδικα μέχρι να φτάσουν σε αυτό το σημείο όλα τα νήματα.

Σε αυτή την εργασία παρουσιάστε ξεχωριστά τα αποτελέσματά σας για τα δύο αντίστοιχα τμήματα της επίλυσης γραμμικών συστημάτων, δηλαδή για την απαλοιφή Gauss και για την αντικατάσταση προς τα πίσω, για κατάλληλα μεγέθη πινάκων (π.χ. n=10000).

Εργασία 2.4 (προαιρετική)

Σε αυτή την άσκηση καλείστε να παραλληλοποιήσετε τον αλγόριθμο της ταξινόμησης με συγχώνευση (mergesort), και πιο συγκεκριμένα την από πάνω προς τα κάτω υλοποίηση του αλγορίθμου κάνοντας χρήση της OpenMP και της οδηγίας task. Το πρόγραμμά σας θα πρέπει να δέχεται σαν ορίσματα: (α) το μέγεθος του πίνακα ακεραίων που θα ταξινομήσει, (β) το αν θα εκτελεστεί ο σειριακός ή ο παράλληλος αλγόριθμος της ταξινόμησης, και (γ) τον αριθμό των νημάτων που θα χρησιμοποιηθούν στην παράλληλη ταξινόμηση. Στη συνέχεια, το πρόγραμμά σας θα πρέπει να παράγει με τυχαίο (ντετερμινιστικό) τρόπο τον προς ταξινόμηση πίνακα ακεραίων και έπειτα να τον ταξινομεί παράλληλα. Τέλος, το πρόγραμμά σας επιβεβαιώνει ότι ο πίνακας είναι ταξινομημένος και τυπώνει το χρόνο εκτέλεσης της ταξινόμησης. Δοκιμάστε να εκτελέσετε το πρόγραμμά σας για διαφορετικά μεγέθη πινάκων (π.χ. 10^7 και 10^8). Παρατηρείτε επιτάχυνση και γιατί; Ίσως σας φανεί χρήσιμος ο όρος if() της οδηγίας task.