



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ
9ο εξάμηνο ΗΜΜΥ, ακαδημαϊκό έτος 2009-2010

ΑΣΚΗΣΗ 1

Προθεσμίες παράδοσης:
Παράδοση ενδιάμεσης αναφοράς: **εως 9 Νοεμβρίου 2009**
Επίδειξη προγραμμάτων: **εως 24 Νοεμβρίου 2009**
Παράδοση τελικής αναφοράς: **εως 12 Ιανουαρίου 2009**

Μέρος Α

Γενικά

Δίνεται ο παρακάτω κώδικας που αποτελεί πυρήνα επίλυσης της τρισδιάστατης εξίσωσης διάχυσης:

```
while (steps < MAX_STEPS && !converged){
    for (x=1; x<X-1; x++)
        for (y=1; y<Y-1; y++)
            for (z=1; z<Z-1; z++)
                A[steps+1][x][y][z] = 1/7*(A[steps][x][y][z] + A[steps][x-1][y][z]
                    + A[steps][x+1][y][z] + A[steps][x][y-1][z]
                    + A[steps][x][y+1][z] + A[steps][x][y][z-1]
                    + A[steps][x][y][z+1]);
    converged = check_convergence();
    steps++;
}
```

Ζητούμενα

1. Αναπτύξτε σειριακό πρόγραμμα που να υλοποιεί τους παραπάνω υπολογισμούς.
2. Υλοποιήστε παράλληλη έκδοση του παραπάνω κώδικα στο μοντέλο ανταλλαγής μηνυμάτων με χρήση της βιβλιοθήκης MPI. Χρησιμοποιήστε σαν πλατφόρμα εκτέλεσης τη συστοιχία των *twins* (12 διπλοεπεξεργαστικοί κόμβοι).

3. Παρουσιάστε τη στρατηγική παραλληλοποίησης (π.χ. με ψευδοκώδικα) στην ενδιάμεση αναφορά.
4. Πραγματοποιείτε μετρήσεις επίδοσης με βάση σενάριο που θα σας δοθεί στο εργαστήριο.
5. Συγκενρώστε τα αποτελέσματα και τις παρατηρήσεις σας στην τελική αναφορά.

Διευκρινίσεις

- Για οδηγίες σύνδεσης, μεταγλώττισης, εκτέλεσης κ.λ.π. των προγραμμάτων σας συμβουλευτείτε τις “ΟΔΗΓΙΕΣ ΕΡΓΑΣΤΗΡΙΟΥ” που σας έχουν δοθεί. Το αρχείο με τις οδηγίες είναι διαθέσιμο επίσης στο <http://www.cslab.ece.ntua.gr/courses/pps/files/pps-lab-guide.pdf>. **Δώστε ιδιαίτερη σημασία στον τερματισμό των διεργασιών που έχουν διακοπή “απότομα” και στον καθαρισμό του συστήματος σε αυτή την περίπτωση!!!**
- Η μνήμη που θα χρησιμοποιήσετε θα δεσμεύεται **δυναμικά** (π.χ. με `malloc`). Διατηρείστε τις τιμές μόνο για το τρέχον και το προηγούμενο υπολογιστικό βήμα.
- Η κατανομή των υπολογισμών θα γίνεται σε πλέγμα επεξεργαστών και θα δίνεται η επιλογή δημιουργίας πλέγματος σε μία, δύο ή τρεις διαστάσεις. Π.χ. όταν χρησιμοποιούνται 8 επεξεργαστές, το μονοδιάστατο πλέγμα θα τους τοποθετεί γραμμικά, το δισδιάστατο σε διάταξη 4×2 (ή 2×4) και το τρισδιάστατο σε διάταξη $(2 \times 2 \times 2)$. Σε περίπτωση που υπάρχουν πολλές επιλογές για τη δημιουργία του πλέγματος (π.χ. για 16 επεξεργαστές σε δισδιάστατο πλέγμα υπάρχουν οι δυνατές διατάξεις 16×1 , 8×2 , 4×4 , 2×8 , 1×16) θα επιλέγεται η διάταξη που ισοκατανέμει κατά το δυνατόν τους επεξεργαστές ανά διάσταση (στο παράδειγμά μας 4×4).
- Αρχικοποιείτε ως εξής: $A[0][x][y][z] = c_t$, $A[steps][0][y][z] = c_{x0}$, $A[steps][X-1][y][z] = c_{x1}$, $A[steps][x][0][z] = c_{y0}$, $A[steps][x][Y-1][z] = c_{y1}$, $A[steps][x][y][0] = c_{z0}$ και $A[steps][x][y][Z-1] = c_{z1}$ με c_t , c_x , c_y , $c_z = \text{σταθ}$. Προαιρετικά μπορείτε να επιλέξετε για αρχικοποίηση μη σταθερές τιμές (π.χ. συναρτήσεις των x , y , z).
- Οι υπολογισμοί έχουν συγκλίνει αν ισχύει η ιδιότητα $|A[steps+1][x][y][z] - A[steps][x][y][z]| < \epsilon$ για όλα τα x, y, z του χωρίου. **Πραγματοποιείτε έλεγχο σύγκλισης κάθε K υπολογιστικά βήματα.**
- Το πρόγραμμά σας θα πρέπει να είναι **παραμετρικό** ως προς το μέγεθος του χωρίου (X , Y , Z), τον μέγιστο αριθμό βημάτων για τη σύγκλιση (`MAX_STEPS`), τη συχνότητα ελέγχου σύγκλισης (K), τον αριθμό των επεξεργαστών, το πλέγμα των επεξεργαστών, την ακρίβεια (ϵ) κλπ.
- Για τη μέτρηση των χρόνων εκτέλεσης προτείνεται να χρησιμοποιηθεί η συνάρτηση βιβλιοθήκης `gettimeofday` του `sys/time.h`. Παρατηρείστε ότι κατά την μέτρηση χρόνων ενδιαφέρει **μόνο** το υπολογιστικό κομμάτι του αλγορίθμου, και όχι η φάση αρχικοποίησης ή π.χ. εκτύπωσης των αποτελεσμάτων. Για το λόγο αυτό πραγματοποιείται κατάλληλος συγχρονισμός των διεργασιών πριν τις μετρήσεις χρόνου.

Μέρος Β

Ο αλγόριθμος FW υπολογίζει τα ελάχιστα μονοπάτια ανάμεσα σε όλα τα ζεύγη ενός γράφου με V κόμβους. Σε ψευδοκώδικα ο αλγόριθμος έχει τη μορφή:

```
for (k=0; k<V; k++)
  for (i=0; i<V; i++)
    for (j=0; j<V; j++)
      A[i][j] = min(A[i][j], A[i][k]+A[k][j]);
```

Ζητούμενα

1. Αναπτύξτε σειριακό πρόγραμμα που να υλοποιεί τον αλγόριθμο FW.
2. Επισημάνετε τους βρόχους του αλγορίθμου που μπορούν να παραλληλοποιηθούν και αναπτύξτε παράλληλο πρόγραμμα στο μοντέλο κοινής μνήμης (shared memory) με τη βοήθεια της βιβλιοθήκης OpenMP για κάθε περίπτωση παραλληλοποίησης των βρόχων.
3. Παρουσιάστε την παραπάνω ανάλυση στην ενδιάμεση αναφορά.
4. Πραγματοποιήστε μετρήσεις επίδοσης με βάση συγκεκριμένο σενάριο που θα σας δοθεί στο εργαστήριο.
5. Συγκεντρώστε τις παρατηρήσεις και τα αποτελέσματά σας σε συνοπτική αναφορά.

Διευκρινίσεις

- Το πρόγραμμά σας να είναι παραμετρικό ως προς το μέγεθος του γράφου και τον αριθμό των νημάτων.
- Αρχικοποιήστε τον πίνακα γειτνίασης (A) σε τυχαίες τιμές.