

ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

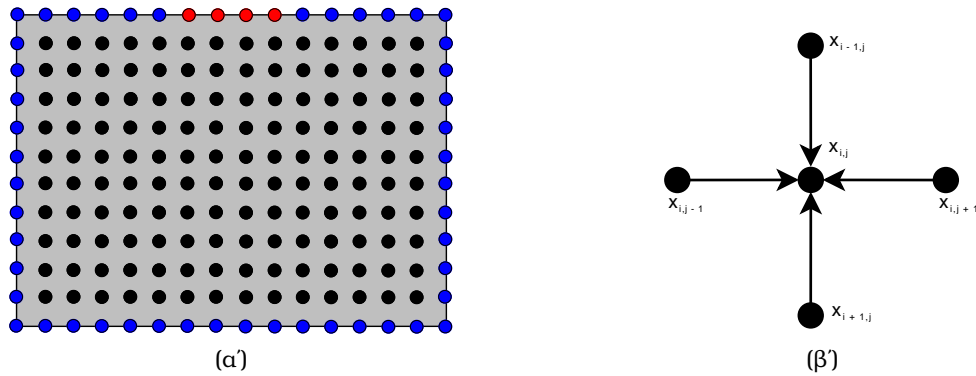
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ ΕΑΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2008

Αντικείμενο της εργαστηριακής άσκησης για το 2008 αποτελεί το πρόβλημα της εύρεσης της κατανομής θερμότητας ενός αντικειμένου σε σταθερή κατάσταση. Μια υπολογιστική μέθοδος η οποία λύνει το συγκεκριμένο πρόβλημα είναι η μέθοδος Jacobi. Η Jacobi είναι επαναληπτική μέθοδος που χρησιμοποιείται για την επίλυση μερικών διαφορικών εξισώσεων. Ο αλγόριθμος Jacobi, παρ' ότι στη απλή του μορφή δεν συγκλίνει ικανοποιητικά γρήγορα σε λύση σε σχέση με άλλους αλγορίθμους, ωστόσο, χαρακτηρίζεται από απλότητα και υψηλό βαθμό παραλληλισμού. Γι' αυτό το λόγο επιλέχθηκε να αποτελέσει την εφαρμογή προς υλοποίηση της φετινής εργαστηριακής άσκησης.

Στη συγκεκριμένη άσκηση θα χρησιμοποιήσετε την μέθοδο Jacobi για την εύρεση της κατανομής θερμότητας σε μια ορθογώνια πλάκα μετάλλου. Η υλοποίηση της μεθόδου θα γίνει χρησιμοποιώντας δυο μοντέλα παράλληλου προγραμματισμού κοινής μνήμης που θα μελετηθούν στη διάρκεια του μαθήματος: τα νήματα και το πρότυπο OpenMP. Σκοπός της εργαστηριακής άσκησης είναι η πρακτική εξοικείωσή σας με σημαντικές πτυχές της παράλληλης επεξεργασίας και η εμβάθυνση σε έννοιες που θα αναλυθούν κατά τη διδασκαλία του μαθήματος. Συγκεκριμένα η εργαστηριακή άσκηση αφορά :

- Πολυεπεξεργαστικά συστήματα με ομοιόμορφη πρόσβαση στην κύρια μνήμη.
- Πολυνηματικό προγραμματισμό με μοντέλο κοινής μνήμης στην παραπάνω αρχιτεκτονική.
- Συγχρονισμό της εκτέλεσης νημάτων σε πολυνηματική εφαρμογή.
- Οργάνωση και καταμερισμό υπολογισμού με σκοπό την επιτάχυνση της εκτέλεσης πολυνηματικής εφαρμογής.

Στη συνέχεια περιγράφεται ο αλγόριθμος Jacobi και τα ζητήματα που θα σας διευκολύνουν στην υλοποίηση της άσκησης.



Σχήμα 1: Διακριτοποίηση για τον υπολογισμό της κατανομής θερμότητας στην μεταλλική πλάκα

Περιγραφή Αλγορίθμου

Για λόγους απλότητας ο αλγόριθμος θα εφαρμοστεί στις δυο διαστάσεις. Γι' αυτό το λόγο θεωρούμε πως το αντικείμενο για το οποίο ζητείται η κατανομή θερμότητας σε σταθερή κατάσταση είναι μια ορθογώνια πλάκα μετάλλου. Για να μοντελοποιήσουμε την κατανομή της θερμότητας της πλάκας υπολογιστικά, θεωρούμε την πλάκα ως ένα δισδιάστατο πλέγμα διακριτών σημείων, για κάθε ένα από τα οποία ορίζεται μια ορισμένη θερμοκρασία (Σχήμα 1.(α)). Η θερμοκρασία ενός τέτοιου σημείου στο πλέγμα εξαρτάται από τη θερμοκρασία των γειτονικών του σημείων (Σχήμα 1.(β)). Συγκεκριμένα η θερμοκρασία ενός σημείου $x_{i,j}$ υπολογίζεται ως ο μέσος όρος των γειτονικών θερμοκρασιών:

$$x_{i,j} = \frac{x_{i-1,j} + x_{i+1,j} + x_{i,j+1} + x_{i,j-1}}{4}$$

Με δεδομένο ότι η θερμοκρασία των ακραίων σημείων στις πλευρές της πλάκας παραμένει σταθερή, το ζητούμενο του προβλήματος είναι να υπολογιστεί η τελική κατανομή θερμότητας στην πλάκα υπό τη συνεχή εφαρμογή της συγκεκριμένης θερμότητας από το περιβάλλον. Το κύριο τμήμα του ακολουθιακού αλγορίθμου Jacobi που υπολογίζει την τελική κατανομή παρουσιάζεται παρακάτω στο Σχήμα 2.

Ζητήματα Υλοποίησης

Για την υλοποίηση των παράλληλων εκδόσεων του αλγορίθμου Jacobi θα πρέπει να λάβετε υπόψιν τα εξής:

- Η τιμή της θερμοκρασίας κάθε σημείου της πλάκας αποθηκεύεται σε μεταβλητή τύπου `double`. Μπορείτε να ξεκινήσετε την υλοποίηση σας θέτοντας δικές σας τιμές, π.χ. με τη μορφή που φαίνεται στο Σχήμα (α).

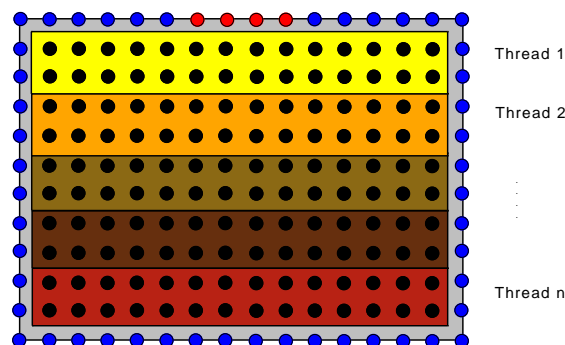
```

1  for ( ;; ) {
2      diff = 0.0;
3
4      for ( i = 1; i < N - 1; i++ ) {
5          for ( j = 1; j < M - 1; j++ ) {
6              after[i][j] = 0.25 * ( before[i-1][j] + before[i+1][j] +
7                  before[i][j-1] + before[i][j+1] );
8
9              if ( fabs( after[i][j] - before[i][j] ) > diff )
10                 diff = fabs( after[i][j] - before[i][j] );
11          }
12      }
13
14      if ( diff <= EPSILON )
15          break;
16
17      for ( i = 1; i < N - 1; i++ )
18          for ( j = 1; j < M - 1; j++ )
19              before[i][j] = after[i][j];
20 }

```

Σχήμα 2: Ακολουθιακός αλγόριθμος Jacobi

- Οι τιμές για τις οποίες θα πάρετε μετρήσεις θα σας σταλούν σε ένα αρχείο κειμένου με την παράδοση του λογαριασμού σας στο υπολογιστικό σύστημα του εργαστηρίου.
- Ο αλγόριθμος Jacobi δεν συγκλίνει κάτω από οποιεσδήποτε αρχικές συνθήκες. Γι' αυτό θα πρέπει να επιλέξετε σωστές αρχικές συνθήκες τόσο για τα άκρα όσο και για τα εσωτερικά σημεία. Για τα εσωτερικά σημεία μπορείτε να θέσετε αρχικά τιμή ίση με τη μέση τιμή των ακραίων σημείων σε κάθε πλευρά.
- Στις παράλληλες εκδόσεις η ανάθεση τμημάτων της πλάκας σε νήματα (είτε χρησιμοποιώντας Pthreads, είτε OpenMP) θα γίνεται κατά γραμμές όπως φαίνεται στο Σχήμα 3.
- Ιδιαίτερη προσοχή πρέπει να δοθεί στη συνθήκη τερματισμού του αλγορίθμου. Όπως φαίνεται από το ακολουθιακό τμήμα παραπάνω ο αλγόριθμος τερματίζει όταν για κάθε σημείο, η διαφορά σε απόλυτη τιμή της προηγούμενης από την καινούργια θερμοκρασία δεν ξεπερνά ένα όριο (EPSILON). Το όριο είναι ανάλογο με τις αρχικές συνθήκες και θα δοθεί σαν δεδομένο με το αρχείο εισόδου (π.χ EPSILON = 0.01). Στην περίπτωση των νημάτων η συνθήκη τερματισμού θα πρέπει να ορίζεται από κοινά δεδομένα, γεγονός που σηματοδοτεί την ανάγκη για συγχρονισμένη πρόσβαση σε αυτά.

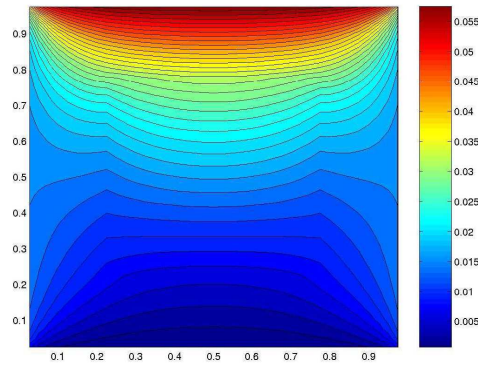


Σχήμα 3: Ανάθεση ανά νήματα

Συνολικά στο πλαίσιο της άσκησης θα υλοποιήσετε τις παρακάτω εκδόσεις του αλγορίθμου Jacobi:

1. **Ακολουθιακό πρόγραμμα.** Στηριζόμενοι στο τμήμα κώδικα που παρουσιάστηκε προηγουμένως θα υλοποιήσετε την ακολουθιακή εκδοχή του αλγορίθμου που θα αποτελέσει τη βάση με την οποία θα συγκριθούν οι παράλληλες υλοποιήσεις.
2. **Υλοποίηση με Pthreads.** Η πρώτη παράλληλη εκδοχή του αλγορίθμου Jacobi θα γίνει σε χαμηλό επίπεδο, απευθείας με τη χρήση νημάτων Pthreads. Σε αυτή την έκδοση θα υλοποιηθούν δυο περιπτώσεις. Στην πρώτη περίπτωση το πλέγμα των σημείων προς ανανέωση θα χωρίζεται σε ισομερή τμήματα το πλήθος των οποίων θα ισούται με τον αριθμό των νημάτων (στατική εκτέλεση). Στην δεύτερη περίπτωση κάθε νήμα θα επεξεργάζεται μια γραμμή κάθε φορά μέχρι να ανανεωθούν όλες οι γραμμές για κάθε γύρο εκτέλεσης.
3. **Υλοποίηση με OpenMP.** Χρησιμοποιώντας το OpenMP θα υλοποιήσετε μια υψηλότερου επιπέδου πολυνηματική εκδοχή. Και σε αυτή την εκδοχή θα διακρίνετε επίσης δυο περιπτώσεις, θέτοντας την παράμετρο δρομολόγησης (schedule) στην στατική (static) επιλογή και στην επιλογή οδήγησης (guided) αντίστοιχα.

Καλείστε να υλοποιήσετε την άσκηση χρησιμοποιώντας τη γλώσσα προγραμματισμού C, τη βιβλιοθήκη νημάτων Pthreads και τον μεταφραστή GCC από την έκδοση 4.2 και έπειτα που υποστηρίζει το πρότυπο OpenMP. Για τη χρήση φράγματος θα χρησιμοποιήσετε την ήδη υπάρχουσα υλοποίηση των Pthread barriers (κλήσεις της μορφής `pthread_barrier_*`) η οποία είναι διαθέσιμη στο Linux.



Σχήμα 4: Διάγραμμα κατανομής θερμότητας

Πειράματα - Μετρήσεις

Η πειραματική αξιολόγηση της υλοποίησης σας θα γίνει σε πολυεπεξεργαστικό σύστημα στο οποίο θα αποκτήσετε πρόσβαση μετά την δήλωση της ομάδας σας. Στα πειράματα που θα εκτελέσετε θα πάρετε μετρήσεις για τις πέντε παραπάνω εκδοχές της υλοποίησης σας. Θα συγκρίνετε από κοινού τις πέντε εκδοχές σε διαγράμματα μετρήσεων (χρόνοι εκτέλεσης και αντίστοιχες επιταχύνσεις), ενώ καλείστε να σχολιάσετε τις διάφορες υλοποιήσεις και ως προς τα υπόλοιπα, πέραν της απόδοσης, χαρακτηριστικά τους. Επίσης θα παρουσιάσετε και ένα διάγραμμα παρόμοιο με αυτό του Σχήματος 4, στο οποίο θα εμφανίζεται γραφικά η κατανομή της θερμότητας της μεταλλικής πλάκας όπως την υπολογίσατε. Για κάθε ομάδα τα δεδομένα εισόδου ενδέχεται να διαφέρουν.

Παραδοτέα

Για την ολοκλήρωση της εργαστηριακής άσκησης καλείστε να παραδώσετε ένα συμπιεσμένο φάκελο σε μορφή `.tar.gz` με όνομα `PROJECT_PARAL_xx.tar.gz` (όπου `xx` ο αριθμός της ομάδας σας), ο οποίος θα περιέχει:

- Όλα τα αρχεία του πηγαίου κώδικα σας (source code).
- Επιπλέον απαραίτητα αρχεία για την μετάφραση ή την τεκμηρίωση του κώδικα σας (π.χ. Makefiles)
- Τεχνική αναφορά σε μορφή `.pdf`, `.ps`, ή `.doc` στην οποία θα παρουσιάζονται οι μετρήσεις και τα διαγράμματα σας και θα τεκμηριώνονται τα συμπεράσματα σας από την πειραματική αξιολόγηση. Επίσης, θα εκτιμηθεί ο εύστοχος σχολιασμός των πιο αξιοσημείωτων από τη σκοπιά της παράλληλης επεξεργασίας

σημείων της υλοποίησης σας. Επιπλέον, είναι δεκτά και οποιαδήποτε σχόλια ή παρατηρήσεις σχετικά με την άσκηση.

Στο φάκελο δε θα πρέπει να περιέχονται αρχεία ενδιάμεσης αναπαράστασης .ο ή εκτελέσιμα αρχεία.

Τα παραπάνω παραδοτέα τα αποστέλλετε στην διεύθυνση ηλεκτρονικού ταχυδρομείου `par_sys@hpclab.ceid.upatras.gr` απαραίτητως με θέμα: `PROJECT_PARAL_xx.tar.gz` (όπου xx ο αριθμός της ομάδας σας). Έπειτα από επιτυχή παράδοση θα σας αποσταλεί αυτόματο μήνυμα επιβεβαίωσης.

Τυπικά Ζητήματα

- Η ημερομηνία παράδοσης της άσκησης θα ανακοινωθεί. Η ημερομηνία της προφορικής εξέτασης στην άσκηση θα ανακοινωθεί μετά την παράδοση των ασκήσεων.
- Επικοινωνία για απορίες:
 - ☞ Forum μαθήματος στο My.Ceid οποιαδήποτε μέρα και ώρα.
 - ☞ Υπεύθυνος καθηγητής: Ελευθέριος Πολυχρονόπουλος
(`edp@hpclab.ceid.upatras.gr`)
 - ☞ Μεταπτυχιακός: Καρανιάσης Κώστας (`kik@hpclab.ceid.upatras.gr`)