

ΣΧΟΛΗ  
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Παράλληλα Συστήματα Επεξεργασίας

---

**Ενδιάμεση Αναφορά**

---

Κουτσούκος Δημήτριος  
(ΑΜ: 03110078)  
Χαρισόπουλος Βασίλειος  
(ΑΜ: 03110046)

2 Δεκεμβρίου 2014

# 1 Υλοποίηση MPI

## 1.1 Μέθοδος Jacobi

Η επαναληπτική μέθοδος Jacobi χρησιμοποιεί τις τιμές των 4 γειτονικών σημείων του πλέγματος, οι οποίες αφορούν την προηγούμενη χρονική στιγμή του υπολογισμού. Ο τύπος της είναι

$$u_{x,y}^{t+1} = \frac{u_{x-1,y}^t + u_{x,y-1}^t + u_{x+1,y}^t + u_{x,y+1}^t}{4}$$

Επομένως στον υπολογισμό των στοιχείων για την χρονική στιγμή  $t + 1$  για τυχαίο σημείο του πλέγματος δεν υπάρχουν εξαρτήσεις από άλλους υπολογισμούς που μπορεί να γίνονται ταυτόχρονα για άλλο σημείο.

### 1.1.1 Διαμοιρασμός πίνακα στο MPI

Μπορούμε να διαμοιράσουμε είτε κατά στήλες/γραμμές (1D partition), είτε να χωρίσουμε τον πίνακα σε block (2D partition). Για λόγους ευκολίας και περιορισμού της επικοινωνίας έχουμε επιλέξει τον πρώτο τύπο διαμοιρασμού.

Οι συναρτήσεις του MPI που χρησιμοποιούμε για το διαμοιρασμό των στοιχείων του πίνακα είναι οι `MPI_Send`, `MPI_Recv`.

### 1.1.2 Ανταλλαγή στοιχείων μεταξύ επεξεργαστών

Εφόσον ακολουθήσουμε διαμέριση κατά στήλες, οι επεξεργαστές πρέπει να ανταλλάσσουν είναι τα γειτονικά τους, δηλαδή για το block που περιλαμβάνει τις στήλες  $i \dots i + k$ , πρέπει να αποστείλει τις στήλες  $i, i + k$  μέσω της `MPI_Send` και να ανακτήσει τις στήλες  $i - 1, i + k + 1$  μέσω της `MPI_Recv`. Γενικά σε κάθε τύπο διαμέρισης, 2 επεξεργαστές ανταλλάσσουν στοιχεία μονάχα αν "γειτνιάζουν" στο partition του πίνακα που ακολουθείται.

Η ανταλλαγή δεδομένων μπορεί να γίνεται στην αρχή της φάσης των υπολογισμών για κάθε χρονική στιγμή  $t_i$ . Εναλλακτικά ο κάθε επεξεργαστής μπορεί να υπολογίζει τα στοιχεία που είναι εφικτό να υπολογίσει χωρίς δεδομένα που ανήκουν σε άλλο επεξεργαστή και η ανταλλαγή δεδομένων να γίνεται μετά από αυτή τη διαδικασία.

Τα δεδομένα που ανταλλάσσονται, όταν στέλνονται μέσω της `MPI_Send` αποθηκεύονται στον MPI Buffer, απ'όπου ανακτώνται όταν καλείται η `MPI_Recv`.

### 1.1.3 Έλεγχος Σύγκλισης

Κάθε διεργασία κάνει έλεγχο σύγκλισης στα δεδομένα που έχει υπολογίσει τοπικά μέσω της `converge()`. Η `converge` επιστρέφει

- 0 εάν συναντήσει αποκλίνουσες τιμές (άρα αποτυχία σύγκλισης)
- 1 εάν διαπιστώσει σύγκλιση

Επομένως, έχοντας χωρίσει τον πίνακα  $A$  σε κομμάτια  $A_i, i \in \#Processors$ , συνολική σύγκλιση συνεπάγεται να έχουν συγκλίνει όλα τα κομμάτια, άρα ουσιαστικά  $\min\{converge(A_i)\} = 1$  (το `min()` αντιστοιχεί στο λογικό AND για συναρτήσεις σαν την `converge()`). Έτσι, εύκολα διαπιστώνουμε ότι μπορούμε να ελέγξουμε για συνολική σύγκλιση μέσω της `MPI_Reduce`, χρησιμοποιώντας για τελεστή τον `MPI_MIN`.

## 1.2 Μέθοδος Red-Black SOR

Στη μέθοδο Red-Black SOR το πλέγμα χωρίζεται σε 2 σύνολα σημείων, τα κόκκινα και τα μαύρα. Το layout που προκύπτει είναι παρόμοιο με αυτό μιας σκακιέρας, δηλαδή κάθε σημείο του ενός χρώματος γειτνιάζει οριζοντίως και καθέτως με 4 σημεία του έτερου χρώματος. Η μέθοδος περιλαμβάνει 2 διακριτές φάσεις υπολογισμού. Η πρώτη είναι η ανανέωση των "red" σημείων του πλέγματος, και δίνεται από τον τύπο

$$u_{x,y}^{t+1} = u_{x,y}^t + \omega \frac{u_{x-1,y}^t + u_{x,y-1}^t + u_{x+1,y}^t + u_{x,y+1}^t - 4u_{x,y}^t}{4}, (x+y) \% 2 == 0$$

Παρατηρούμε ότι τα red στοιχεία υπολογίζονται μονάχα από black στοιχεία, και μάλιστα χρησιμοποιούν παρελθοντικές τιμές. Επομένως η φάση αυτή δεν περιέχει αλληλοεξαρτήσεις ή χρονικές προτεραιότητες υπολογισμών (σε αντίθεση με τη μέθοδο gauss-seidel). Η επόμενη φάση υπολογίζει τα black στοιχεία του πλέγματος για χρονική στιγμή  $t+1$  με βάση την ανανεωμένη τιμή των red στοιχείων (δηλαδή για  $t+1$ ).

$$u_{x,y}^{t+1} = u_{x,y}^t + \omega \frac{u_{x-1,y}^{t+1} + u_{x,y-1}^{t+1} + u_{x+1,y}^{t+1} + u_{x,y+1}^{t+1} - 4u_{x,y}^t}{4}, (x+y) \% 2 == 1$$

Όμως οι ανανεωμένες τιμές των red στοιχείων έχουν υπολογιστεί στην αμέσως προηγούμενη φάση και άρα είναι διαθέσιμες εξ'αρχής. Επομένως και πάλι έχουμε ευχέρεια όσον αφορά την παραλληλοποίηση της εκτέλεσης των υπολογισμών.