



**Πανεπιστήμιο Δυτικής Αττικής Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών**

**ΕΡΓΑΣΤΗΡΙΟ
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΑΡΑΛΛΗΛΟ ΥΠΟΛΟΓΙΣΜΟ**

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ - 21390174

Ημερομηνία παραδοσης:

Κυριακή 21/1/2024.

ΕΡΓΑΣΙΑ 2 MPI

Ένας πίνακας $A(N \times N)$ λέγεται αυστηρά διαγώνια δεσπόζων (strictly diagonally dominant) εάν για κάθε γραμμή του πίνακα του A ισχύει: Σας ζητείται να γράψετε και να τρέξετε ένα MPI πρόγραμμα σε C (θεωρώντας ένα παράλληλο περιβάλλον p επεξεργαστών), το οποίο δοθέντος ενός δισδιάστατου πίνακα $A(N \times N)$:

Αρχικά ξεκινήσα το προγράμμα αρχικοποιώντας τις μεταβλητές που χρειαζομαι για να ξεκινήσω το MPI_Init η ιδέα μου για την ασκήση είναι απλή να χρησιμοποιήσω τις συναρτησεις συλογγικού υπολογισμού όπως MPI_Gather, MPI_Scatterv κτλπ που θα δείτε στον κωδικά μου για να σπασώ τον διδισδιαστατό πινακά στις υπολοιπές διεργασίες και να κάνω τους απαραίτητους υπολογισμούς που μου ζητείται άπο τα παρακάτω ερωτημάτα.

Ερώτημα 1

A. Θα ελέγχει αρχικά με παράλληλο τρόπο αν ο πίνακας A είναι αυστηρά διαγώνια δεσπόζων (ο επεξεργαστής '0' θα πρέπει να διαβάσει από την οθόνη τον πίνακα A και στο τέλος να τυπώνει το αποτέλεσμα – 'yes' ή 'no').

Αρχικοποιώ τους πινάκες sendcnt και offset1:

sendcnt[i] υποδεικνύει πόσα στοιχεία θα λάβει κάθε διεργασία κατά τη λειτουργία διασποράς. Το offset1[i] υποδεικνύει τον αρχικό δείκτη των στοιχείων για κάθε διεργασία στο buffer αποστολής.

Μετά η διεργασία 0 φτιάχνει τον διασδιαστατό πινάκα καθώς ζηταεί τις τιμές από τον χρήστη Διασκορπίζει τον πίνακα x κατά γραμμές σε όλες τις διεργασίες με την Scatterv ετσι καθε διεργασία θα παρει από έναν μονοδιαστατό πινάκα για να κναουμε τους υπολογισμούς παραλλήλα.Μετά υπολογίζω αν ο πίνακας είναι αυστήρα διαγωνιός δεσπόζων και θετώ ένα flag=1 για οποιά διεργασία είναι αληθής η συνθήκη και τελός τα κάνω όλα gather σε ένα πινακά και αν ο πινάκας περιέχει μόνο 1 τοτέ ειναι σωστό και τυπώνω 'YES' αλλιώς 'NO'.

Ερώτηση 2

B. Στην περίπτωση που αυτό ισχύει (είναι δηλ. ο πίνακας A αυστηρά διαγώνια δεσπόζων) το πρόγραμμα θα πρέπει στη συνέχεια να υπολογίζει παράλληλα το μέγιστο κατ' απόλυτη τιμή στοιχείο της διαγωνίου του πίνακα A ($m = \max(|A_{ii}|)$).

Για την 2 ερώτηση αρχικά κάνω broadcast σε όλες τις διεργασίες το big_flag το οποίο ελέγχουμε αν θα συνεχίσουμε ή ο χρήστης edwse λάθος input στον πίνακα μας. Μετά υπολογίζω το max με τους πίνακες που είχαμε διαμειράσει από πριν. Κάνω gather όλα το local_max καθε διεργασίας σε έναν τελικό πίνακα final_max βρίσκω το max και τυπωνώ στην μητρική διεργασία

Ερώτημα 3

Γ. Και ακολούθως με βάση αυτό (m) να φτιάχνει παράλληλα ένα νέο πίνακα B NxN (τον οποίον θα τυπώνει επίσης ο '0' στο τέλος στην οθόνη) όπου: $B_{ij} = m - |A_{ij}|$ για $i < j$ και $B_{ij} = m$ για $i = j$

Αρχικά μοιραζώ σε όλες τις υπολοιπές διεργασίες με broadcast το max ώστε να μπορούν ολοι να εκτελέσουν τους υπολογισμούς. Αρχικοποιώ τους πίνακες recvnt και offset2 τους οποίους θα χρησιμοποιησσουμε στο gathern ώστε να μπορούμε να μαζέψουμε τα δεδομένα μας με την σειρά στον πίνακα και την σωστή θέση.

Ερώτημα 4

Δ. Για τον παραπάνω πίνακα B ζητείται επίσης να υπολογιστεί παράλληλα (και να τυπώνεται στο τέλος από τον '0' επίσης στην οθόνη) το ελάχιστο σε τιμή στοιχείο του, καθώς και σε ποιά θέση (i,j) του πίνακα B βρίσκεται.

Τελός για την τελευταία ερωτήση βρήκα τοπικά τα min στους πίνακες που είχαμε φτιαξεί καθώς και την θέση την οποία βρεθηκε ο min και τα κάνω gather τις θέσεις και τα τοπικά min σε ξεχωριστούς πίνακες. Ο 0 θα ανατρέξει τους πίνακες για να βρει το μικροτερό και επειδή οι δύο πίνακες είναι παραλληλή θα παρουμε και την αντιστοιχή θέση και ετσι θα βρουμε το min και τις θέσεις του στον πίνακα.

ΕΝΔΕΙΚΤΙΚΑ ΤΡΕΞΙΜΑΤΑ.

Pinakas A:

```
6|1|1|1
1|7|1|1
1|1|8|1
1|1|1|9

MPI_Gather(&min_loc, 1, MPI_INT, final_min, 1, MPI_INT, mpi_root, 0);
MPI_Gather(&min_pos, 1, MPI_INT, pos, 1, MPI_INT, mpi_root, 0);
if (me == mpi_root) {
    fin_min = final_min[0];
    min_pos = pos[0];
    'YES'
```

To MAX se apoliti timi einai: 9

0 kainougiuos Pinakas B:

```
3|8|8|8
8|2|8|8
8|8|1|8
8|8|8|0
```

To MIN se apoliti timi einai 0 se thesi i=3 kai j=3 se thesi i=

~/De/MP/MPI_/Collective communication

✓ 13s

Pinakas A:

```
1|1|1|1
1|1|1|1
1|1|1|1
1|1|1|1

if (fin_min > final_min[i]) {
    fin_min = final_min[i];
    i_pos = i;
    min_pos = pos[i];
}
```

'NO'

~/De/MP/MPI_/Collective communication