Ρόμπερτ Πολοβίνα

23390338

Ομάδα Ε5 – Εργασία 2

Απαντήσεις ερωτημάτων περιγραφής

α) Για το ερώτημα I (και μόνο για αυτό), περιγράψτε αναλυτικά πώς θα μπορούσατε να επεκτείνετε την υλοποίησή σας έτσι ώστε να συμπεριφέρεται σωστά για οποιοδήποτε συνδυασμό 'N' και 'p':

Για οποιοδήποτε συνδυασμό N και p θα μπορούσαμε να κάνουμε άνισο μοίρασμα γραμμών, δηλαδή κάποια ranks να πάρουν μία επιπλέον γραμμή και άλλα 0 γραμμές όταν p>N:

- Τα ranks από το 0 μέχρι το (N%p)-1 θα πάρουν (N/p)+1 γραμμές ενώ
- τα υπόλοιπα ranks από N%p έως p-1 θα πάρουν N/p γραμμές.
- Αν p>N τότε N/p= 0 άρα μόνο οι πρώτες N διεργασίες θα πάρουν 1 γραμμή και οι υπόλοιπες 0 (συμμετέχοντας όμως κανονικά στις κλήσεις των συναρτήσεων του MPI αλλά με rows= 0).
- Ο υπολογισμός του πλήθους των στοιχείων που θα σταλούν (τμήματα) και από ποιο index θα ξεκινήσει το κάθε τμήμα, γίνεται στην rank 0.
- Μετά τους τοπικούς υπολογισμούς των αποτελεσμάτων, θα πρέπει να φροντίσουμε στην σωστή ένωση τους από την rank 0, τηρώντας τον τρόπο με τον οποίο κάναμε την ανάθεση στην αρχή.
- Με τον τρόπο αυτό δεν αλλάζει ο τρόπος υπολογισμού των αποτελεσμάτων, απλώς πρέπει να είμαστε προσεκτικοί στην σωστή ανάθεση των τμημάτων των πινάκων.

β) και πως θα μπορούσατε να βρείτε το μέγιστο σε τιμή στοιχείο του πίνακα αποτελέσματος, καθώς και σε ποια θέση (γραμμή και στήλη του πίνακα) βρίσκεται:

Εφόσον κάθε rank υπολογίζει τοπικά (local_R1) μία γραμμή του τελικού πίνακα R1, μπορούμε να προσθέσουμε και την εύρεση του μεγίστου της αντίστοιχης γραμμής σε κάθε rank καθώς και να κρατήσουμε σε ποια θέση της γραμμής βρίσκεται. Έπειτα θα επιστρέψουμε συλλογικά στην rank 0 τις γραμμές local_R1 και μη συλλογικά το μέγιστο κάθε γραμμής, την θέση του και το rank της διεργασίας.

Η rank 0 θα συγκρίνει κάθε τοπικό μέγιστο (local_max) που λαμβάνει με το δικό της μέγιστο (global_max) και αν το εισερχόμενο είναι μεγαλύτερο, θα κάνει

- global_max= local_max;
- max_i= local_max_i;
- max_rank= local_rank;

Έτσι, έχοντας το τοπικό index του μεγίστου και το rank στο οποίο βρέθηκε, είναι σαν να έχουμε την στήλη και την γραμμή αντίστοιχα του τελικού πίνακα που βρίσκεται το μέγιστο στοιχείο.

γ) Περιγράψτε τέλος επίσης, για το ερώτημα IV, πως θα μπορούσατε να επεκτείνετε την υλοποίησή σας για οποιοδήποτε 'N' ακέραιο πολλαπλάσιο του 'p' (N%p=0):

Για να επεκτείνουμε την υλοποίηση του τέταρτου ερωτήματος, αντί για μια γραμμή των C & D ανά διεργασία, θα μπορούσαμε να χρησιμοποιήσουμε τμήματα γραμμών μεγέθους N/p. Κάθε διεργασία δηλαδή, θα αναλαμβάνει τον υπολογισμό N/p γραμμών του αποτελέσματος και θα λάβει στην αρχή το τμήμα γραμμών των C & D που της αντιστοιχεί.

Ο ring αλγόριθμος δεν θα αλλάξει αλλά σε καθένα από τα p-1 βήματα θα κυκλοφορεί από rank σε rank το τμήμα αυτό των γραμμών της D (αντί για μία γραμμή), με το τμήμα των γραμμών της C που έχει η κάθε διεργασία να μένει αμετάβλητο.

Ενδεικτικά τρεξίματα

Για N=p=4:

A(1x4)= [1 2 3 4]

B(4x1)= [5 6 7 8]*

C(4x4)= [1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16]

D(4x4)= [1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1]

(Μοναδιαίος)

<u>Αποτελέσματα:</u>

Ερώτημα 1:

Επειδή D μοναδιαίος, πρέπει μόνο η διαγώνιος του C να αυξάνεται κατά 1.

```
I. R1= C(4x4)+D(4x4)=
2.00  2.00  3.00  4.00
5.00  7.00  7.00  8.00
9.00  10.00  12.00  12.00
13.00  14.00  15.00  17.00
```

Ερώτημα 2:

- R2[0]= 1·5 + 2·6 + 3·7 + 4·8 = 70
- R2 [1]= 5·5 + 6·6 + 7·7 + 8·8 = 174
- R2 [2]= 9·5 + 10·6 + 11·7 + 12·8 = 278
- R2 [3]= 13·5 + 14·6 + 15·7 + 16·8 = 382

```
II. R2= C(4x4)+B(4x1)=
70.0
174.0
278.0
382.0
```

Ερώτημα 3:

R3= 1·5 + 2·6 + 3·7 + 4·8= 70

```
III. R3= A(1x4)*B(4x1)= 70
```

Ερώτημα 4:

Επειδή D= μοναδιαίος, πρέπει R4= C

```
IV. R4= C(4x4)*D(4x4)=
1.00  2.00  3.00  4.00
5.00  6.00  7.00  8.00
9.00  10.00  11.00  12.00
13.00  14.00  15.00  16.00
```

<u>Για N%p=0 (μόνο ερωτήματα 1 έως 3):</u>

N= 8, p= 4

A(1x8)= [1 2 3 4 5 6 7 8]

B(8x1)= [1 1 1 1 1 1 1 1] * (προγραμματιστικά οι στήλες δεν διαφέρουν απ' τις γραμμές)

C(8x8)= [0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0]

(Μηδενικός)

D(8x8)= [10 0 0 0 0 0 0 0

0 20 0 0 0 0 0 0

0 0 30 0 0 0 0 0

0 0 0 40 0 0 0 0

0 0 0 0 50 0 0 0

0 0 0 0 0 60 0 0

0 0 0 0 0 0 70 0

0 0 0 0 0 0 0 80]

(Διαγώνιος)

<u>Αποτελέσματα:</u>

Ερώτημα 1:

Εφόσον C= 0 => R1= D

```
I. R1= C(8x8)+D(8x8)=
10.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  20.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  30.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  40.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  50.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  60.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  70.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  0.00  80.00
```

Ερώτημα 2:

Εφόσον C= 0 => R2= 0

```
II. R2= C(8x8)+B(8x1)=
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

Ερώτημα 3:

R3= 1+2+3+4+5+6+7+8= 36

```
III. R3= A(1x8)*B(8x1)= 36
```