ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΑΡΑΛΛΗΛΟ ΥΠΟΛΟΓΙΣΜΟ

ΑΣΚΗΣΗ – ΙΙ

Ζωβοΐλης Δημήτριος-Μάριος

ΑΜ: 19390064

Τμήμα: Ε1 (Δευτέρα 12-2μμ)
19/01/22

ΠΕΡΙΕΧΟΜΕΝΑ

Πίνακας περιεχομένων

1. Εισαγωγή – περιγραφή της εργασίας

Σκοπός της εργασίας είναι η κατασκευή ενός προγράμματος MPI σε γλώσσα C δέχεται ως είσοδο έναν δισδιάστατο πίνακα A(NxN), να υπολογίζει παράλληλα σε περιβάλλον 'p' επεξεργαστών και να τυπώνει στην οθόνη (ως έξοδο) τα ακόλουθα:

α) αν ο πίνακας A είναι αυστηρά διαγώνια δεσπόζων (strictly diagonally dominant): $|a_{ii}| > \sum_{j=0, j\neq i}^{N-1} |a_{ij}|,\ i=0..N-1$. Στην περίπτωση που δεν είναι σταματάει εδώ το πρόγραμμα, αλλιώς συνεχίζει και υπολογίζει τα υπόλοιπα.

β) το μέγιστο κατ' απόλυτη τιμή στοιχείο της διαγωνίου του πίνακα A (m=max(|Aii|).

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: $Bij = m - |A_{ij}|$ *για* $i \neq j$ *και* $B_{ij} = m$ *για* $i = j$.

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.

Επίσης, το πρόγραμμα έπρεπε να λυθεί με την χρήση μόνο συναρτήσεων συλλογικής επικοινωνίας και θεωρώντας αρχικά ότι το 'N' είναι ακέραιο πολλαπλάσιο του 'p'. Στη συνέχεια, επεκτάθηκε το πρόγραμμά έτσι ώστε να συμπεριφέρεται σωστά για οποιονδήποτε συνδυασμό τιμών 'N' και 'p' (με χρήση των συναρτήσεων scatterv/gatherv).

2. Τεκμηρίωση της εργασίας

Η λύση της εργασίας βασίζεται στην ιδέα ότι το σύνολο του απαιτούμενου υπολογιστικού φόρτου θα ισοκατανεμηθεί στους 'p' επεξεργαστές του παράλληλου περιβάλλοντός. Έτσι, κάθε επεξεργαστής θα λαμβάνει ένα μέρος του αρχικού πίνακα για να εκτελέσει ότι πράξεις χρειάζονται. Επίσης, να σημειωθεί πως η είσοδος των δεδομένων στο πρόγραμμα θα γίνεται με την χρήση μόνο ενός 'κεντρικού' επεξεργαστή. Στην προκειμένη περίπτωση, ως 'κεντρικός' επεξεργαστής επιλέγχηκε αυτός που έχει `rank==0`. Τέλος, τα αποτελέσματα του προγράμματος συγκεντρώνονται στον 'κεντρικό' αυτό επεξεργαστή και μέσω αυτού παρουσιάζονται στο χρήστη.

Στο αρχείο ex.c βρίσκεται η λύση του προβλήματος όπου το 'N' είναι ακέραιο πολλαπλάσιο του 'p' και στο ex2.c βρίσκεται η επέκταση της λύσης όπου δουλεύει για οποιονδήποτε συνδυασμό τιμών 'N' και 'p'. Στα scripts ex.sh και ex2.sh γίνεται το compile και run των αρχείων ex.c και ex2.c αντίστοιχα. Μέσα στα scripts με την χρήση heredoc τρέχουν τα προγράμματα με έτοιμες εισόδους για να μην χάνεται χρόνος στα `scanf`.

2.1 Σχεδιασμός και πληροφορίες για την υλοποίηση του κώδικα

Η δομή της main συνάρτησης του προγράμματος είχε την εξής μορφή:

1. Αρχικά, διαβάζεται τον πίνακα A και το N από τον επεξεργαστή με `rank==0` με την χρήση της συνάρτησης `read_data()`.

2. Έπειτα στέλνουμε το N του διανύσματος και το μέρος του πίνακα που αντιστοιχεί σε κάθε επεξεργαστή με τη χρήση της `MPI_Bcast` και της `MPI_Scatter` (ή της `MPI_Scatterv`) αντίστοιχα.

3. Έπειτα υπολογίσουμε αν ο πίνακας είναι αυστηρά διαγώνια δεσπόζων (strictly diagonally dominant). Σε περίπτωση που είναι συνεχίζουμε στα επόμενα βήματα, αλλιώς τερματίζει η εκτέλεση του προγράμματος.

4. Κάθε επεξεργαστής υπολογίζει το $\max(|A_{ii}|)$ και έπειτα με την χρήση της `MPI_Allreduce` (εκτελώντας την πράξη `MPI_MAX`) όλοι οι επεξεργαστές έχουν στην τοπική τους μνήμη το $m = \max(|A_{ii}|)$.

5. Μετά, κάθε επεξεργαστής δημιουργεί τον τοπικό πίνακα B με βάση το αρχικό μέρος του πίνακα A που έλαβε και το m που υπολογίστηκε προηγουμένως.

6. Έπειτα, με την χρήση της `MPI_Gather` (ή της `MPI_Gatherv`) κατασκευάζουμε όλο τον πίνακα B στον επεξεργαστή με `rank == 0` και των εκτυπώνουμε.

7. Επόμενο βήμα είναι ο υπολογισμός του ελάχιστου στοιχείου και της θέσης του στον τοπικό πίνακα B. Αυτό επιτυγχάνεται με την χρήση της συνάρτησης find_min.

8. Με την `MPI_Reduce` (καλώντας την με πράξη την `MPI_MINLOC`) γυρίζουμε στον επεξεργαστή με `rank == 0` το ολικό ελάχιστο στοιχείο και την θέση του στον πίνακα B.

9. Τέλος, απελευθερώνεται ο χώρος που έιχε δεσμευτεί δυναμικά μέσω της `malloc()`.

2.2 Προβλήματα που αντιμετωπίστηκαν

Ένα πρόβλημα στο οποίο έπεσα προσπαθώντας να λύσω την άσκηση ήταν στο ερώτημα Δ, πως θα περνάω και την θέση του μικρότερου στοιχείου. Για να το λύσω αυτό, κοίταξα τις πράξεις που μπορούν να γίνουν με την χρήση της `MPI_Reduce` και βρήκα την πράξη `MPI_MINLOC` με την οποία μπορώ να περνάω και την θέση του στοιχείου τελικά. Να σημειωθεί επίσης πως απαντήθηκαν όλα τα ερωτήματα πλήρως τόσο στο βασικό πρόγραμμα όσο και στην επέκταση του προγράμματος.