

Template για την εργασία σε MPI

Σε συνδυασμό με τις πληροφορίες που σας δίνονται στην εκφώνηση της εργασίας MPI και τις επεξηγήσεις οι οποίες σας έχουν δοθεί στις διαλέξεις, μπορείτε να βασιστείτε στο παρακάτω template για την παραλληλοποίηση της επίλυσης της εξίσωσης διάχυσης του Poisson σε μια διάσταση.

```
#include
```

Ορίστε πλήθος τμημάτων και παραμέτρους της εξίσωσης

```
main( )
```

Καθορίστε τιμές για `residual`, για οριακές συνθήκες, δηλώστε το `unorm` όπου κάθε διεργασία θα υπολογίζει την ποσότητα $\sum_i (u_i - u_i^{t-1})^2$ για τα τμήματα i που της έχουν ανατεθεί και που χρησιμοποιείται για τον έλεγχο της απαιτούμενης ακρίβειας σύγκλισης, αλλά δηλώστε και το `global_unorm`, όπου οι διεργασίες θα αθροίζουν τα επιμέρους `unorm` που έχουν υπολογίσει. Δηλώστε u και u_{new} , όπου κάθε διεργασία θα τοποθετεί τις θερμοκρασίες των τμημάτων που την ενδιαφέρουν, κατά τις χρονικές στιγμές $t - 1$ και t , αντίστοιχα. Ορίστε πίνακα `u_all` όπου η διεργασία 0 θα συγκεντρώνει τα αποτελέσματα u όλων των διεργασιών.

Αρχικοποίηση MPI, εύρεση αριθμού διεργασιών και `rank` κάθε διεργασίας, καθορισμός πλήθους τμημάτων που θα διαχειρίζεται κάθε διεργασία.

Δυναμική εκχώρηση μνήμης για το `u_all` (προσοχή, πρέπει να προβλέπονται δυο επιπλέον θέσεις στα άκρα του πίνακα, για την αποθήκευση των οριακών συνθηκών). Τοποθέτηση των αρχικών συνθηκών στον πίνακα `u_all`.

Δυναμική εκχώρηση μνήμης για τα u , u_{new} κάθε διεργασίας, με επιπλέον δυο θέσεις στα άκρα κάθε διανύσματος, όπου η διεργασία i θα τοποθετεί, στη θέση 0 τη θερμοκρασία του δεξιότερου τμήματος της διεργασίας $i - 1$, στη δε τελευταία θέση τη θερμοκρασία του αριστερότερου τμήματος της διεργασίας $i + 1$ (εκτός της διεργασίας 0 και της τελευταίας διεργασίας που στο αριστερό και δεξι άκρο, αντίστοιχα, θα τοποθετούν τις οριακές συνθήκες). Αρχικοποιήστε το u στην τιμή 0 για όλες τις διεργασίες, εκτός από τη θέση 0 της διεργασίας 0 και την τελευταία θέση της τελευταίας διεργασίας, όπου θα μπου οι οριακές συνθήκες του προβλήματος.

Μέσω ενός `while`, όσο η τετραγωνική ρίζα του `global_unorm` είναι μεγαλύτερη της τετραγωνικής ρίζας του `residual`, συνεχίστε να καλείτε τη συνάρτηση `poisson_discrete()` στην οποία κάθε διεργασία υπολογίζει τις νέες θερμοκρασίες των τμημάτων που της έχουν ανατεθεί και επιστρέφει την τιμή

`unorm` για τα τμήματα που της έχουν ανατεθεί. Εκτύπωση, σε κάθε επανάληψη, του αριθμού επανάληψης και της τρέχουσας τιμής του `global_unorm`. Εντός του `while`, όλες οι διεργασίες, σε κάθε επανάληψη, αθροίζουν τα επιμέρους `unorm` που έχουν υπολογίσει, στη μεταβλητή `global_unorm`.

Συγκέντρωση όλων των `u`, από όλες τις διεργασίες, στο `uall`. Προσοχή, γιατί κάθε διεργασία θα πρέπει να στέλνει τις θερμοκρασίες των τμημάτων που της έχουν ανατεθεί στο `uall` και όχι τις τιμές των δυο ακραίων θέσεων του δικού της `u`, στις οποίες αποθηκεύει τις θερμοκρασίες των γειτονικών διεργασιών που χρειάζεται για τους υπολογισμούς της. Προσοχή επίσης για το που θα τοποθετηθούν οι τιμές των `u` στο `uall`, αφού στα άκρα του `uall` πρέπει να αποθηκεύονται οι οριακές συνθήκες του προβλήματος.

Εκτύπωση, από τη διεργασία 0, του τελικού `global_unorm`, εκτύπωση των τελικών `u` που υπολόγισε κάθε διεργασία και ιστογράμματος της λύσης από τη διεργασία 0.

Απελευθέρωση δυναμικά εκχωρημένης μνήμης.

Τέλος MPI.

Τέλος `main()`

`poisson_discrete()`

Ορίστε δυο buffers `sendbuf` και `recvbuf`, όπου κάθε διεργασία θα τοποθετεί κάποια τιμή που χρειάζεται να στείλει σε κάποια γειτονική της διεργασία και όπου κάθε διεργασία θα λαμβάνει κάποια τιμή που κάποια γειτονική της διεργασία της έστειλε, αντίστοιχα. Από τον `recvbuf` η λαμβανόμενη τιμή θα περνάει στην κατάλληλη θέση της πίνακα `u` της διεργασίας, ενώ η αποστελλόμενη τιμή θα περνάει από την κατάλληλη θέση του πίνακα `u` της διεργασίας, στον `sendbuf`.

Εύρεση αριθμού διεργασιών και `rank` κάθε διεργασίας.

Υπολογισμός παραμέτρου `F`.

Κάθε διεργασία υπολογίζει το νέο διάνυσμα `unew` από το διάνυσμα `u`, με βάση την επαναληπτική σχέση υπολογισμού των θερμοκρασιών των τμημάτων.

Κάθε διεργασία υπολογίζει το `unorm` για τα τμήματα που της έχουν ανατεθεί.

`u=unew`, ώστε οι υπολογισμένες στην τρέχουσα επανάληψη τιμές `unew` να είναι οι τιμές `u` της επόμενης επανάληψης (επαναληπτική σχέση).

Ανταλλαγή απαιτούμενων πληροφοριών μεταξύ των διεργασιών, σύμφωνα με τον πίνακα που σας έχει δοθεί στην εκφώνηση της εργασίας, ώστε κάθε διεργασία, κατά τους υπολογισμούς της επόμενης επανάληψης που θα κάνει, να έχει διαθέσιμες τις τιμές των γειτονικών της διεργασιών που χρειάζεται. Να χρησιμοποιήσετε μηνύματα `MPI_Send()` και `MPI_Recv()`. Πιο συγκεκριμένα:

Αν η διεργασία είναι περιττή

... (συμπληρώστε κώδικα εδώ)

Αν η διεργασία δεν είναι η τελευταία

... (συμπληρώστε κώδικα εδώ)

```
Τέλος Αν
Αλλιώς (αν δηλαδή η διεργασία είναι άρτια)
  Αν η διεργασία δεν είναι η τελευταία
    ... (συμπληρώστε κώδικα εδώ)
  Τέλος Αν
  Αν η διεργασία δεν είναι η πρώτη
    ... (συμπληρώστε κώδικα εδώ)
  Τέλος Αν
Τέλος Αν
```

Επιστροφή της τιμής της `unorm`.

Τέλος `poisson_discrete()`