ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΑΡΑΛΛΗΛΟ ΥΠΟΛΟΓΙΣΜΟ - ΑΣΚΗΣΗ ΙΙ 2024-2025

<REDACTED>

Build and run

Για building και debugging του project χρησιμοποιήθηκε CMake, του οποίου το CMakeLists.txt συμπεριλαμβάνεται στα παραδοτέα αρχεία για λόγους αναφοράς, δεν είναι όμως σε καμία περίπτωση απαραίτητη η χρήση CMake για building του project.

Υπάρχουν δύο (εξίσου εύκολοι) τρόποι να γίνει compile το project:

1) Παρέχεται Makefile που το κάνει αυτόματα με τις εξής εντολές (από το root του project folder):

```
make
```

Υπάρχει και δυνατότητα διαγραφής των object files με:

```
make clean
```

2) Απευθείας compile με το gcc wrapper του MPI (mpicc) με:

```
mpicc ./src/*.c -I ./include -o bin
```

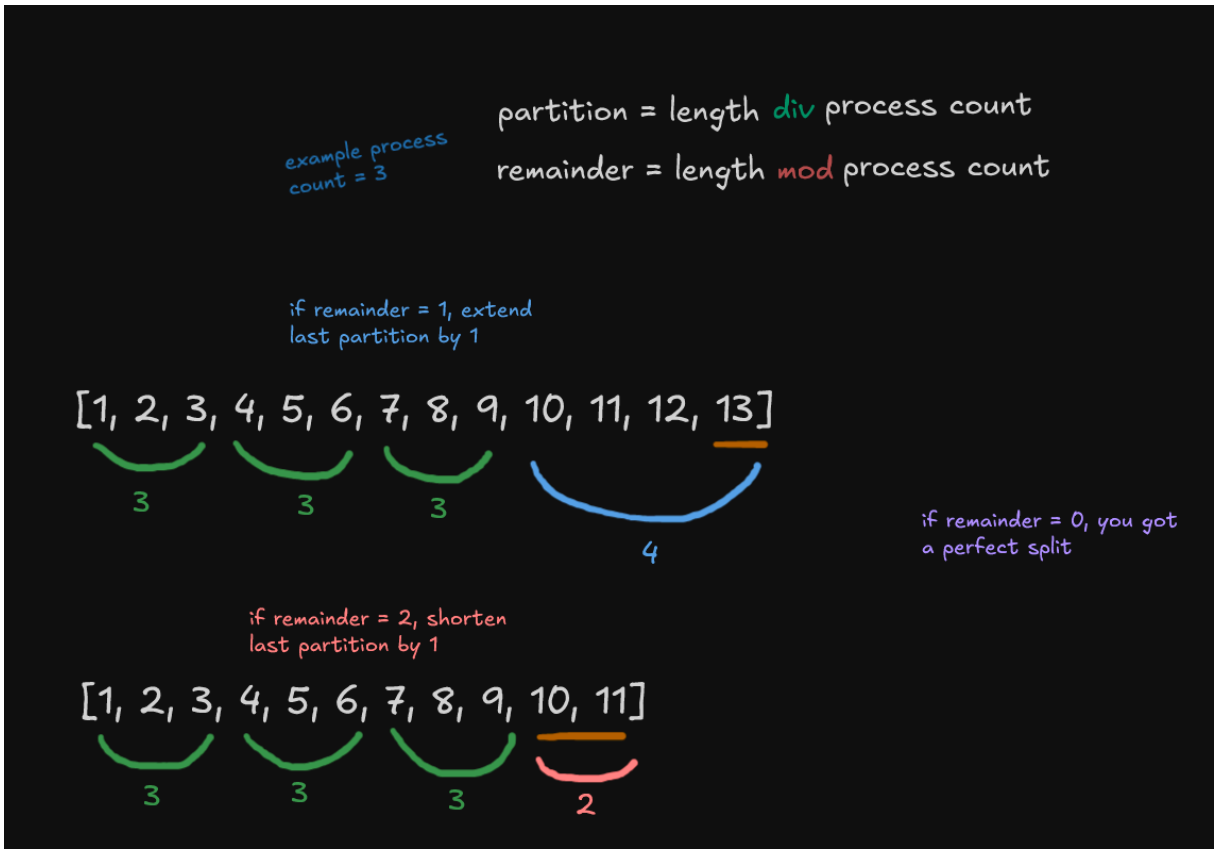
Εκτέλεση του προγράμματος γίνεται με (όπου <process_count> ακέραιο νούμερο των processes, χωρίς angled brackets, πχ. 2):

```
mpirun -np <process_count> bin
```

Η λογική του προγράμματος ξεκινάει με το “κύριο process” (στην περίπτωση μας 0) να διαβάζει από τον χρήστη το μήκος και τα ίδια τα στοιχεία του διανύσματος. Έπειτα γίνεται προετοιμασία για την κατακερμάτιση του φόρτου με την χρήση της MPI_Scatterv(). Χονδρικά το τμήμα που θα πάρει κάθε process είναι το μήκος του διανύσματος δια το πλήθος των processes. Εαν η διαίρεση έχει υπόλοιπο μηδέν σημαίνει πως η τμηματοποίηση του φόρτου είναι τέλεια. Σε περίπτωση που το υπόλοιπο δεν είναι μηδέν υπάρχουν 2 περιπτώσεις (το υπόλοιπο είναι το μήκος του διανύσματος mod το πλήθος των processes):

1) Εάν το υπόλοιπο = 1, θα γίνει επέκταση του τελευταίου τμήματος κατά 1.

2) Εάν το υπόλοιπο = 2, θα γίνει μείωση του τελευταίου τμήματος κατά 1.



Τα νούμερα που δε χωράνε τέλεια στο προκαθορισμένο partition μοιράζονται ξανά κάνοντας προσπέλαση τα process ranks από την αρχή

Γίνεται επίσης ο υπολογισμός των displacements για το MPI_Scatterv ως εξής:

```
// Displacements calculation for Scatterv
for(int i = 1; i < process_count; ++i) {
    displacements[i] = displacements[i - 1] + receive_counts[i
- 1];
}
```

Αυτό γίνεται καθώς η Scatterv δε ξέρει τι offset από την διεύθυνση των τιμών του διανύσματος να χρησιμοποιήσει για n νούμερα που θα λάβει το κάθε process. Το min, το max και η μέση τιμή υπολογίζονται με τα αντίστοιχα MPI_Reduce() operations (MPI_MIN, MPI_MAX και MPI_SUM + διέρεση με το πλήθος νούμερων που έλαβε η συνάρτηση). Τα αποτελέασματα αποστέλονται πίσω σε όλα τα processes με την χρήση MPI_Bcast() καθώς χρησιμοποιούνται για τον υπολογισμό της διασποράς των στοιχείων του διανύσματος, ο οποίος πάλι γίνεται επί των τμημάτων φόρτου και επιστρέφεται στο κύριο process. Ο υπολογισμός της διασποράς γίνεται πλην την διαίρεση διά του πλήθους των στοιχείων του διανύσματος. Τα υπομέρη του αριθμιτή αθροίζονται με χρήση MPI_Reduce() (με MPI_SUM operation) και διαιρούνται με n από το κύριο process.

```
float variance_subsection = 0;
for(int i = 0; i < n_count; i++) {
float x = receive_buffer[i] - vector_mean;
    variance_subsection += x * x;
}
```

Για τον υπολογισμό των στοιχείων του διανύσματος Δ αρκεί να δεσμεύσουμε νέα μνήμη για να κρατήσουμε τα αποτελέσματα και να κάνουμε τις αντίστοιχες πράξεις στο ήδη υπάρχον buffer.

```
// Calculating the value of each element. One extra space is
reserved to store the rank of the process that holds the data
float *delta_subsection = malloc(sizeof(float) * n_count);
for(int i = 0; i < n_count; i++) {
    delta_subsection[i] = ((receive_buffer[i] - vector_min) /
    (vector_max - vector_min)) * 100;
}
```

Η εύρεση του max στο διάνυσμα Δ είναι ημιτελής. Η εύρεση της ίδιας της τιμής δουλεύει μέσω του MPI_Reduce() σε συνδιασμό με το MAX_LOC operation, όμως η θέση του στοιχείου στο διάνυσμα δεν διατηρείται (η λογική έχει υλοποιηθεί αλλά χρειάζεται debugging). Λειτουργεί βρίσκοντας τοπικά την μέγιστη τιμή του τμήματος που κατέχει το κάθε process και εφαρμόζοντας reduction με το MAX_LOC operation πάνω στο τοπικό μέγιστο. Η ταύτιση της floating point (float) τιμής του στοιχείου και της ακέραιας θέσης του (int) γίνεται με την χρήση struct.

Έγινε απόπειρα υλοποίησης του prefix sum με την χρήση MPI_Scan όμως ο κώδικας είναι δυσλειτουργικός και χρειάζεται debugging, συνεπώς το αντίστοιχο τμήμα στο source έχει γίνει commented out.