MONASH University

FIT3143 Lab {Week 10}
Git URL {Specify URL for Lab Week #}

Contribution {To be completed by Students}:

Student ID	Student Name	Contribution
27929795	Luyang Liu	Input file parser, process 1, 2 and 3 MPI code, output file parser, quadratic root calculation code. Report

Self-Evaluation {To be highlighted by Student only}:

Need Help	Work in Progress	Pass	Credit	Distinction	High Distinction

Tutor Evaluation {To be highlighted by Tutor only}:

Incomplete	Needs improvement	Pass	Credit	Distinction	High Distinction

Note: This lab is assessed (Weight: 3%). Please complete the Contribution and Self-Evaluation sections above. In addition, since this is an assessed lab, please submit this E-Folio document and code files into Moodle for grading. Submission link is available in Week 09 of Moodle.

Tasks:

Task 4

MONASH University

Task 4: Observations and Outputs

The first process rank 0 fetches all the coefficients from the input text file. It then calculates the d value. The process passes the data to process of rank 2 to calculate the root values. The type of root (i.e. complex or real) depends on the value of d as it is square rooted in the calculation process.

Finally the roots are passed to process 3 to save to the output.

In total,l process rank 2 has n number of send and receives where n is then number of rows in the text file.

This is also more efficient than without using MPI or any parallelism because task 1 can continue to fetch coefficients and calculate d while process rank 2 and rank 3 can perform operations on previous rows.

Creating of custom data type in MPI:

The key learnings is that you can pass structs in MPI but creating a custom MPI Type by specifying its primitive data types, displacements between internal data and also the length of each data inside.

Creating custom MPI datatype:

```
MPI_Datatype create_root_datatype(root the_root) {
  // The number data type is 1 and just float
  MPI_Datatype array_of_types[1];
  array_of_types[0] = MPI_FLOAT;

  // The number of blocks length are 1 and it is 2 because 2 blocks.
  int array_of_blocklengths[1];
  array_of_blocklengths[0] = 2;

  // The distance between the first and second block
  MPI_Aint array_of_displaysments[1];
  MPI_Aint address1, address2;
  MPI_Get_address(&the_root.real, &address1);
  MPI_Get_address(&the_root.img, &address2);
  array_of_displaysments[0] = address2 - address1;
  MPI_Datatype mpi_root;

  MPI_Type_create_struct(1, array_of_blocklengths, array_of_displaysments,
                         array_of_types, &mpi_root);

  // Need to commit structure type in MPI
  MPI_Type_commit(&mpi_root);
  return mpi_root;
}
```

MONASH University

The main function:

```
int main(int argc, char *argv[]) {
  int rank, size;
  MPI_Comm new_comm;
  MPI_Init(&argc, &argv);

  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);

  char *file_name = "test1.txt";

  MPI_Status status[3];
  MPI_Request request[3];
  int number_of_rows = fetch_number_of_rows(file_name);
  int *lines = init_process_instruction_pointers(file_name, number_of_rows);

  root roots[3];
  MPI_Datatype mpi_root = create_root_datatype(roots[0]);

  if (rank == 0) {
    /*
     * Fetch the row and get coefficients for quadratic equation
     * Calculate the d value
     * Send to node 1
     */

    /*
     * Observations:
     *
     * Need to create a new mpi data type to send struct in openmpi between
     * processes. Otherwise need to extract the values out of the struct and
     * send as array of floats Float is a primitive data structure.
     *
     */

    float send[4];
    write_number_of_rows_to_output(number_of_rows);

    for (int i = 0; i < number_of_rows; i++) {
      float *coef = fetch_next_instruction(lines[i], file_name);
      float d = calculate_d(coef[0], coef[1], coef[2]);
      send[0] = d;
      send[1] = coef[0];
      send[2] = coef[1];
      printf("Row %i, Value d is %.2f.\n", i, d);
      MPI_Send(&send[0], 3, MPI_FLOAT, 1, 0, MPI_COMM_WORLD);
    }
```

```
} else if (rank == 1) {
  /*
   * Receive the d value and coefficients from process rank 1,
   * Calculate the roots.
   * Send to node 2
   */

  /*
   * Observations:
   *
   * Need to convert integer into float in order for the arithmetic to work in c.
   *
   */
  float d;

  for (int i = 0; i < number_of_rows; i++) {
    float recv[3];
    MPI_Recv(&recv[0], 3, MPI_FLOAT, 0, 0, MPI_COMM_WORLD, &status[rank]);
    printf("Row %i: Coffients are d %.2f, a %.2f, b %.2f\n", i, recv[0],
          recv[1], recv[2]);

    root *roots = quadratic_root(recv[1], recv[2], recv[0]);
    printf("Row %i: Roots are %.2f+%.2fi, %.2f-%.2fi.\n", i, roots[0].real,
          roots[0].img, roots[1].real, roots[0].img);
    float send[4] = {roots[0].real, roots[0].img, roots[1].real,
                    roots[1].img};

    MPI_Send(&send[0], 4, MPI_FLOAT, 2, 0, MPI_COMM_WORLD);
  }

} else if (rank == 2) {
  float receive_vals[4];

  for (int i = 0; i < number_of_rows; i++) {
    MPI_Recv(&receive_vals[0], 4, MPI_FLOAT, 1, 0, MPI_COMM_WORLD,
            &status[rank]);
    output_result(receive_vals);
  }
}
```

Input

```
3
a b c
1.0 -4.0 3.0
1.0 3.0 1.0
2.0 2.0 1.0
```

MONASH University

Standard Output:

```
Row 0, Value d is 4.00.
Row 1, Value d is 5.00.
Row 2, Value d is -4.00.
Row 0: Coffients are d 4.00, a 1.00, b -4.00
Row 0: Roots are 3.00+0.00i, 1.00-0.00i.
Row 1: Coffients are d 5.00, a 1.00, b 3.00
Row 1: Roots are -0.38+0.00i, -2.62-0.00i.
Row 2: Coffients are d -4.00, a 2.00, b 2.00
Row 2: Roots are -0.50+0.50i, -0.50-0.50i.
```

Output

```
3
x1    x2
3.00   1.00
x1    x2
-0.38   -2.62
x1    x2    x1_real    x1_img    x2_real    x2_img
          -0.50       0.50     -0.50     -0.50
```

MONASH University

Declaration

I declare that this eFolio tasks and the linked code are my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text and code, nor has any part of this submission been written for me by another person.

Signature of student 1: Luyang Liu

Signature of student 2:_____

For the lab tutor's usage only:

a) Please refer to the maximum marks for each criterion in the first page of the lab specifications in Week 10.
b) Complete the table below based on assessing the student's submitted work above. Include additional feedback in the comments section (if necessary).
c) Note: Partial marks are given for a partially correct or incomplete criterion.

	Code compiles without errors and executed correctly (2 marks)	Sufficient code comments (2 marks)	Questions or instructions fully answered (5 marks)	Proper presentation of results and analysis (3 marks)	Comments
Task 4					

Total marks: