Practice questions for MPI

August 22, 2018

Overview

1. These questions are to familiarize you to MPI parallel computing environment
2. Some of the code might have bugs but not all of them do
3. You can assume the syntax of the code (e.g order of arguments in MPI calls, array allocation calls etc.) is correct
4. Focus on the logic, you don't have to care about performance issues
5. There are two variables re-defined in these code:
   (a) *size* is the total number of MPI tasks
   (b) *rank* is the id of each the MPI task

Question 1

Given the snippet below:

Listing 1: code snippet for question 1

```
int arr [10];
// calculating the number of iterations that each process will execute
int range = 10/size;

// calculating the start and end value of the for loop for each process
int start =  rank*range;
int end = start + range;

// at rank 0, initializing the value for array "arr"
if (rank == 0) {
        for (int i =0; i < 10; i++) {
                arr[i] = i;
        }
}

// Scatter portion of the array to other MPI processes
int recvbuf[range];
MPI_Scatter(arr,range, MPI_INT, recvbuf, range,MPI_INT,0, MPI_COMM_WORLD);

// Increase the value of each element by 1
for (int i = 0 ; i < range ; i++) {
  recvbuf[i] += 1;
}

// Gather the updated elements from all processes to get the final array
MPI_Gather(recvbuf,range, MPI_INT,arr, range,MPI_INT,0, MPI_COMM_WORLD);
```

Running with total **3** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, what will be the value of *arr* at rank 0 after the gather call :

Question 2

Given 2 snippets below: The first snippet code is the serial (original) program, and the second snippet code is its parallel version (i.e trying to do the same thing)

Listing 2: code snippet (serial) for question 2

```
int sum = 1;

// increase sum by 10 using a loop
for (int i = 0; i < 10; i ++) {
        sum +=1;
}
```

Listing 3: code snippet (parallel) for question 2

```
// calculating the number of iterations that each process will execute
int range = 10/size;

// calculating the start and end value of the for loop for each process
int p_start_value = rank*range;
int p_end_value = p_start_value  + range;

int sum =1;

// increase sum in parallel in total of 10 times
for (int i = p_start_value; i <  p_end_value; i++) {
   sum  +=1;
 }

// Reduce (compute total sum) from the partly "sum" values of every processes
int receive_placeholder = 0;
MPI_Allreduce(&sum,&receive_placeholder,1,MPI_INT,MPI_SUM,MPI_COMM_WORLD);

// Reassign total sum value to "sum" variable
sum = receive_placeholder;
```

Running with total **2** MPI tasks.

1. Will the parallel program terminate without error ?
2. If the program terminates, will the value of *sum* at rank 0 in the paralel programbe the same as the value of *sum* in the serial program ?

Question 3

Given the snippet below:

Listing 4: code snippet for question 3

```
int arr [10];
int rev [10];

// calculating the number of iterations that each process will execute
int range = 10/size;

// calculating the start and end value of the for loop for each process
int p_start_value = rank*range;
int p_end_value = p_start_value + range;

for (int i = p_start_value ; i < p_end_value; i++) {

        // compute all elements at even indexes first
        if (i < 5) {
         arr[i*2] =2 ;
        }

        // compute all elements at odd indexes second
        else {
         arr[(i-5)*2+1] =3;
        }
}

// Gathering the elements from each process to get the final array
MPI_Gather(&arr[rank*range],range, MPI_INT,rev, range,MPI_INT,0,
     MPI_COMM_WORLD);
```

Running with total **5** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, what will be the value of *rev* at rank 0 after the gather call:

Question 4

Given the snippet below:

Listing 5: code snippet for question 4

```
int *arr ;
// calculating the number of iterations that each process will execute
int range = 10/size;

if (rank == 0) {

        // allocating and initializing array "arr"
        arr = (int*) malloc(sizeof(int)*10);
        for (int i =0; i < 10; i++) {
                arr[i] = i;
        }
}

// Scatter portion of the array to other MPI processes
int recvbuf[range];
MPI_Scatter(arr,range, MPI_INT, recvbuf, range,MPI_INT,0, MPI_COMM_WORLD);
```

Running with total **2** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, what will be the value of *recvbuf* at rank 1 after the scatter call:

Question 5

Given the snippet below

Listing 6: code snippet for question 5

```
int arr [9];
int rev[9];
// calculating the number of iterations that each process will execute
int range = 5/size;

// calculating the start and end value of the for loop for each process
int p_start_value = rank*range;
int p_end_value = p_start_value + range;


for (int i =  p_start_value; i < p_end_value; i++) {

        // update 2 elements at a time unless this is the very last iteration
        // of the last rank MPI task
        arr[i*2]=1;
        if (i < 4) {
                arr[i*2+1] =2;
        };
}

// Gathering the elements from each process to get the final array

MPI_Gather(&arr[rank*range],range*2, MPI_INT,rev, range*2,MPI_INT,0,
     MPI_COMM_WORLD);
```

Running with total **5** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, what will be the value of *rev* at rank 0 after the gather call:

Question 6

Given the snippet below:

Listing 7: code snippet for question 6

```
// calculating the number of iterations that each process will execute
int range = (20/size);

// calculating the start and end value of the for loop for each process
int start =  rank*range;
int end = start + range ;

// adding extra iterations to the last process
if (rank == size - 1) {
        end = end + (20 % size);
}

// looking to see if i ever equals 19
for (int i =start; i < end; i++) {
        int flag = 0;

        // if i equals 19, set flag to 1 and broadcasting
        // the result to other MPI processes
        if (i  == 19) {
                flag =1;
        }
        int result = 0;
        // the allreduce calls are used to broadcasting the flag
        MPI_Allreduce(&flag,&result,1,MPI_INT,MPI_MAX,MPI_COMM_WORLD);

        // break the loop if the broadcasting is set
        if (result == 1) {
                break;
        }
}
```

Running with total **3** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, will the break statement ever be executed ?

Question 7

Given 2 snippets below: The first snippet code is the serial (original) program, and the second snippet code is its parallel version (i.e trying to do the same thing.

Listing 8: code snippet (serial) for question 7

```
int sum = 1;

// increase sum by 5 using a loop
for (int i = 0; i < 5; i ++) {
        sum +=1;
}
```

Listing 9: code snippet (parallel) for question 7

```
// calculating the number of iterations that each process will execute
int range = 5/size;

// calculating the start and end value of the for loop for each process
int p_start_value = rank*range;
int p_end_value = p_start_value + range;

int sum =1;

// increase sum in parallel
for (int i = p_start_value; i <  p_end_value; i++) {
   sum  +=1;
}

// Reduce (compute total sum) from the partly "sum" values of every processes
int receive_placeholder = 0
MPI_Allreduce(&sum,&receive_placeholder,1,MPI_INT,MPI_SUM,MPI_COMM_WORLD);

// Reassign total sum value to "sum" variable
sum = receive_placeholder;
```

Running with total **10** MPI tasks.

1. Will the parallel program terminate without error ?
2. If the program terminates, will the value of *sum* at rank 0 be the same as the value of *sum* in the serial program ?

Question 8

Given the snippet below:

Listing 10: code snippet for question 8

```
int arr [12];

// initializing array "arr" at rank 0
if (rank = 0) {
        for (int i =0; i < 10; i++) {
                arr[i] = i;
        }
}

// calculating the number of iterations that each process will execute
int range = (6/size);

// calculating the start and end value of the for loop for each process
int start =  rank*range;
int end = start + range ;

// adding extra iterations to the last process
if (rank == size - 1) {
        end = end + (6 % size);
}

// calculating the number of elements each process is getting
// as well their displacements in regard to the original array
int displacements[size], count[size];
for (int i = 0; i < size ; i++) {

        // The formula for "displacement" and "count" is similar to
        // computing start and end (i.e each process has the same
        // number of elements, only the last process may have extra ones)
        displacement[i] = rank*(12/size);
        if (i < size - 1) {
                count[i] = 12/size;
        }
        else {
                count[i] = 12 - rank*(12/size);
        }
}

// Scatter portion of the array to other MPI processes
int recvbuf[count[rank]];
MPI_Scatterv(arr,count,displacement, MPI_INT, recvbuf,count[rank],MPI_INT,0,
     MPI_COMM_WORLD);

// increase the elements by one ( 2 at a time)
for (int i = 0 ; i < range ; i++) {
  recvbuf[i] += 1;
  recvbuf[i+1] += 1;
}

// Gathering the elements from each process to get the final array
MPI_Gatherv(recvbuf,range, MPI_INT,arr, count,displacement,MPI_INT,0,
     MPI_COMM_WORLD);
```

Running with total **4** MPI tasks.

1. Will this program terminate without error ?
2. If the program terminates, what will be the value of *arr* at rank 0 after the gather call: