Quiz-4 Nov 4 results for Shiba Inu

Score for this attempt: **11** out of 15

Submitted 20 Nov at 16:10

This attempt took 70 minutes.

Question 1

2 / 2 pts

Given that the content of array `a` (in row major order) is `{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0},` what is the content of array `b` for each process after executing the following code segment using 4 MPI processes ?

```
    #define SIZE 4
    ......
    float a[SIZE][SIZE];
    int myrank, comm_sz;
    float b[SIZE] = {0};

    MPI_Datatype newtype;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

    MPI_Type_vector(SIZE, 1, SIZE, MPI_FLOAT, &newtype);
    MPI_Type_commit(&newtype);

    if(myrank == 0){
        for(int i = 1; i < comm_sz; i++)
            MPI_Send(&a[0][i], 1, newtype, i, 0, MPI_COMM_WORLD);
    }
    else
        MPI_Recv(b, SIZE, MPI_FLOAT, 0, 0, MPI_COMM_WORLD,
MPI_STATUS_IGNORE);
    ......
```

Correct!

● Process 0: `b = {0, 0, 0, 0}`
Process 1: `b = {2.0, 6.0, 10.0, 14.0}`
Process 2: `b = {3.0, 7.0, 11.0, 15.0}`
Process 3: `b = {4.0, 8.0, 12.0, 16.0}`

○ Process 0: `b = {1.0, 2.0, 3.0, 4.0}`
Process 1: `b = {5.0, 6.0, 7.0, 8.0}`
Process 2: `b = {9.0, 10.0, 11.0, 12.0}`
Process 3: `b = {13.0, 14.0, 15.0, 16.0}`

○ Process 0: `b = {0, 0, 0, 0}`
Process 1: `b = {1.0, 5.0, 9.0, 13.0}`
Process 2: `b = {2.0, 6.0, 10.0, 14.0}`
Process 3: `b = {3.0, 7.0, 11.0, 15.0}`

○ Process 0: `b = {1.0, 5.0, 9.0, 13.0}`
Process 1: `b = {2.0, 6.0, 10.0, 14.0}`
Process 2: `b = {3.0, 7.0, 11.0, 15.0}`
Process 3: `b = {4.0, 8.0, 12.0, 16.0}`

Question 2

2 / 2 pts

Given that the content of array `a` (in row major order) is `{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0},` what is the content of array `b` for each process after executing the following code segment using 4 MPI processes ?

```
#define SIZE 4
```

```
    float a[SIZE][SIZE];
    int myrank, comm_sz;
    float b[SIZE] = {0};
    int blocklengths[SIZE], displacements[SIZE];
    MPI_Datatype newtype;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
     for(int i=0; i<SIZE; i++) {
        blocklengths[i] = 1;
        displacements[i] = i*SIZE + i;
    }

    MPI_Type_indexed(SIZE, blocklengths, displacements, MPI_FLOAT,
&newtype);
    MPI_Type_commit(&newtype);

    if(myrank == 0){
        for(int i = 1; i < comm_sz; i++)
            MPI_Send(&a[0][0], 1, newtype, i, 0, MPI_COMM_WORLD);
    }
    else
        MPI_Recv(b, SIZE, MPI_FLOAT, 0, 0, MPI_COMM_WORLD,
MPI_STATUS_IGNORE);
```

○ The content of array `b` is {`4.0, 7.0, 10.0, 13.0`} on all processes.

○ The content of array `b` is {`4.0, 8.0, 12.0, 16.0`} on all processes.

Correct!

●

The content of array `b` is {`1.0, 6.0, 11.0, 16.0`} on all processes except on process 0.

○

The content of array `b` is {`1.0, 5.0, 9.0, 13.0`} on all processes except on process 0.

Question 3

3 / 3 pts

Consider the following code fragment, in which process i, $0 \leq i < P - 1$ where P is the number of processes in the communicator, sends a message to process $i + 1$, and process $i = P - 1$ sends a message to process 0; furthermore, process i, $0 < i < P$, also receives a message from process $i - 1$, and process 0 receives a message from process $P - 1$.

```
        int a[10], b[10], nprocs, myrank;
        MPI_Status status;
        ...
        MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
        MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
        MPI_Send(a, 10, MPI_INT, (myrank + 1) % nprocs, 1, MPI_COMM_WORLD);
        MPI_Recv(b, 10, MPI_INT, (myrank - 1 + nprocs) % nprocs, 1,
MPI_COMM_WORLD, &status);
        ...
```

Which of the following statements is (are) TRUE?

□ The calls to `MPI_Send()` and `MPI_Recv()` in the code fragment will never cause a deadlock.

Correct!

☑ We can rewrite this code using `MPI_Sendrecv` to make the code safe.

□ The execution of `MPI_Send()` and `MPI_Recv()` in the code fragment always causes a deadlock.

Correct!

☑ The execution of `MPI_Send()` and `MPI_Recv()` in the code fragment might cause a deadlock.

□ We can't use non-blocking send and receive functions for this case.

Question 4

0 / 2 pts

Which of the following statements is (are) TRUE?

orrect answer

□

`MPI_Comm_create` must be called by all the processes associated with the communicator given in the function call.

Correct!

☑

A message sent in one context cannot be received in another context.

□

`MPI_Comm_split` can partition the group associated with a given communicator into overlapping subgroups.

ou Answered

☑

If two communicators have exactly the same group of processes, then these two communicators are identical.

Question 5

0 / 2 pts

Given the following code segment,

```
......
int num_procs, orig_rank, new_size, new_rank, sum = 0;
MPI_Comm new_comm;

MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &num_procs);
MPI_Comm_rank(MPI_COMM_WORLD, &orig_rank);

MPI_Comm_split(MPI_COMM_WORLD, orig_rank%2, 0, &new_comm);

MPI_Comm_size(new_comm, &new_size);
MPI_Comm_rank(new_comm, &new_rank);

MPI_Reduce(&new_rank, &sum, 1, MPI_INT, MPI_SUM, 0, new_comm);
......
```

if the code is ran using `mpiexec` with option `-n 8`, what is the rank of process 4 from `MPI_COMM_WORLD` in `new_comm`?

○ 3

○ 4

ou Answered ● 1

orrect answer ○ 2

Question 6

2 / 2 pts

Given the code segment in Question 5, what is the value of `new_size`, respectively, for processes 0 and 7 in `MPI_COMM_WORLD`?

○ Process 0: `new_size = 8`; process 7: `new_size = 8`;

○ Process 0: `new_size = 8`; process 7: `new_size = 4`;

○ Process 0: `new_size = 2`; process 7: `new_size = 2`

Correct! ● Process 0: `new_size = 4`; process 7: `new_size = 4`;

Question 7

2 / 2 pts

Given the code segment in Question 5, how many communicators are there when it is executed?

○ 4

Correct!

● 3

○ 1

○ 2

Quiz score: **11** out of 15