

CPSC 424/524 Assignment 3

yl2335, Yangxiaokang Liu

Task 1: Execute a Simple “Hello World” Program

Q1: Why is the elapsed time reported by the master in task1.c so different from what is reported as “real” time by the time command? What is the master actually measuring? Is it a useful measurement? (If not: think about why not, and what might you do about it?)

A1: Because the elapsed time reported by the master in task1.c measures the message sending time of the master thread. While the "real" time measures the execution time of all the processes. I think the message sending time is useless in this case. We should record the time when all the processes have finished their work instead of recording the time when the master process have finished sending all messages. One possible solution is to set a barrier and wait for every thread to finish its work.

Q2: Could there be any competition for resources (e.g., hardware, software, or data) between your MPI program, the MPI runtime system, and the operating system for this program? If so, which resources? (Hint: Think about the resources used by MPI and for I/O.)

A2: Yes, there is a competition for stdout resources (printf). However, The standard stdio functions, such as printf, perform locking to be thread-safe, so multi-threaded code that contains calls to printf can be serialized to some extent (which would hurt the performance of parallelized code).

Task 2: Modification to Avoid Worker Printouts

Q1: How does the elapsed time reported by the master in this task compare to the time reported as “real” time by the time command? Is the comparison qualitatively different from Task 1? Explain.

A1: The elapsed time reported by the master is very close to the "real" time by the time command. The comparison is qualitatively different from task 1, the elapsed time in task 2 measures the actual working time instead of the message sending time.

Q2: You'll probably see some differences in the total elapsed time among the three runs you made for Task 2. Explain why this happens and describe (in words) how to fix it. A good solution should show little variation in total elapsed time, regardless of the way that work is assigned to the workers.

A2: MPI doesn't preserve the temporal order of messages unless they are transmitted between the same sender/receiver pair. Therefore, ordered receptions would cause delay due to different execution times of different processes.

One solution is to use MPI_ANY_SOURCE instead of a definite source rank. To use MPI_ANY_SOURCE, we need to store all the received messages from the worker processes and print out the messages in order after receiving all the messages.

Task 3: Correcting Performance Issues

Q: Modify task2.c to create a new program task3.c that cures the performance problem identified in Question 2 for Task 2, while still keeping the printouts in order by increasing worker number. Run the new program three times to validate that it works as expected. This task will require adding to your Makefile and extending the Slurm script so that it builds all the programs and runs all three tasks (three times each).

A:

```
char** printouts = (char**) malloc( (size - 1) * sizeof(char*) );
for (int i=1; i<size; i++) {
    printouts[i - 1] = (char*) malloc( 100 * sizeof(char) );
}

for (i=1; i<size; i++) {
    MPI_Recv(message, 100, MPI_CHAR, MPI_ANY_SOURCE, type, MPI_COMM_WORLD,
&status);
    sleep(3);
    sprintf(printouts[i - 1], "Message from process %d: %s", i, message);
}

for (i=1; i<size; i++) {
    printf("%s", printouts[i - 1]);
}

for (int i=1; i<size; i++) {
    free(printouts[i - 1]);
}
free(printouts);
```

I stored all the messages in "printouts" and received messages out of order. Then messages were printed in increasing order. This produced elapsed times with little variation.

Experiments

```
sbatch runtask.sh
```

Currently Loaded Modules:

- 1) StdEnv (S)
- 2) GCCcore/7.3.0
- 3) binutils/2.30-GCCcore-7.3.0
- 4) icc/2018.3.222-GCC-7.3.0-2.30
- 5) ifort/2018.3.222-GCC-7.3.0-2.30
- 6) iccifort/2018.3.222-GCC-7.3.0-2.30
- 7) zlib/1.2.11-GCCcore-7.3.0
- 8) numactl/2.0.11-GCCcore-7.3.0
- 9) XZ/5.2.4-GCCcore-7.3.0
- 10) libxml2/2.9.8-GCCcore-7.3.0
- 11) libpciaccess/0.14-GCCcore-7.3.0
- 12) hwloc/1.11.10-GCCcore-7.3.0
- 13) OpenMPI/3.1.1-iccifort-2018.3.222-GCC-7.3.0-2.30

Where:

S: Module is Sticky, requires --force to unload or purge

Working Directory:

/gpfs/loomis/project/cpsc424/cpsc424_y12335/y12335_ps3_cpsc424

Making task1

```
rm -f task1 task1.o
```

```
rm -f task2 task2.o
```

```
rm -f task3 task3.o
```

```
mpicc -g -O3 -xHost -fno-alias -std=c99 -c task1.c
```

```
mpicc -o task1 -g -O3 -xHost -fno-alias -std=c99 task1.o timing.o rwork.o
```

Node List:

c03n11,c04n03

ntasks-per-node = 2

Run 1

Message printed by master: Total elapsed time is 0.002934 seconds.

From process 2: I worked for 5 seconds after receiving the following message:

Hello, from process 0.

From process 1: I worked for 10 seconds after receiving the following message:

Hello, from process 0.

From process 3: I worked for 15 seconds after receiving the following message:

Hello, from process 0.

real 0m15.786s

user 0m0.683s

sys 0m0.476s

Run 2

Message printed by master: Total elapsed time is 0.001680 seconds.

From process 3: I worked for 5 seconds after receiving the following message:

Hello, from process 0.

From process 2: I worked for 10 seconds after receiving the following message:

Hello, from process 0.

From process 1: I worked for 15 seconds after receiving the following message:

Hello, from process 0.

real 0m15.540s

user 0m0.544s

sys 0m0.427s

Run 3

Message printed by master: Total elapsed time is 0.001654 seconds.

From process 2: I worked for 5 seconds after receiving the following message:

Hello, from process 0.

From process 3: I worked for 10 seconds after receiving the following message:

Hello, from process 0.

From process 1: I worked for 15 seconds after receiving the following message:

Hello, from process 0.

real 0m15.625s

user 0m0.537s

sys 0m0.413s

Making task2

mpicc -g -O3 -xHost -fno-alias -std=c99 -c task2.c

mpicc -o task2 -g -O3 -xHost -fno-alias -std=c99 task2.o timing.o rwork.o

Run 1

Message from process 1: Hello master, from process 1 after working 5 seconds

Message from process 2: Hello master, from process 2 after working 15 seconds

Message from process 3: Hello master, from process 3 after working 10 seconds

Message printed by master: Total elapsed time is 21.000402 seconds.

real 0m22.060s

user 0m12.579s

sys 0m0.432s

Run 2

Message from process 1: Hello master, from process 1 after working 10 seconds

Message from process 2: Hello master, from process 2 after working 5 seconds

Message from process 3: Hello master, from process 3 after working 15 seconds

Message printed by master: Total elapsed time is 19.000569 seconds.

real 0m19.546s

user 0m10.370s

sys 0m0.398s

Run 3

Message from process 1: Hello master, from process 1 after working 10 seconds

Message from process 2: Hello master, from process 2 after working 15 seconds

Message from process 3: Hello master, from process 3 after working 5 seconds

Message printed by master: Total elapsed time is 21.000333 seconds.

real 0m21.546s

user 0m12.449s

sys 0m0.387s

Making task3

mpicc -g -O3 -xHost -fno-alias -std=c99 -c task3.c

mpicc -o task3 -g -O3 -xHost -fno-alias -std=c99 task3.o timing.o rwork.o

Run 1

Message from process 1: Hello master, from process 1 after working 5 seconds

Message from process 2: Hello master, from process 2 after working 10 seconds

Message from process 3: Hello master, from process 3 after working 15 seconds

Message printed by master: Total elapsed time is 18.006579 seconds.

real 0m18.557s

user 0m9.508s

sys 0m0.402s

Run 2

Message from process 1: Hello master, from process 1 after working 5 seconds

Message from process 2: Hello master, from process 2 after working 10 seconds

Message from process 3: Hello master, from process 3 after working 15 seconds

Message printed by master: Total elapsed time is 18.002166 seconds.

real 0m19.644s

user 0m9.489s

sys 0m0.419s

Run 3

Message from process 1: Hello master, from process 3 after working 5 seconds

Message from process 2: Hello master, from process 2 after working 10 seconds

Message from process 3: Hello master, from process 1 after working 15 seconds

Message printed by master: Total elapsed time is 18.015121 seconds.

real 0m18.567s

user 0m9.259s

sys 0m0.319s