**Portland State University – ECE 588/688**

**Winter 2024**

**Homework 4 Notes**

Due Date: Week 8 Sunday, Mar 3, 2024 Before 11:59 PM

Submit to Canvas (text and C files)

Points: 6

Notes are marked in blue color below in this file.

For this homework, you will write a parallel program using MPI. For more information on MPI, you can check the tutorial: <https://hpc-tutorials.llnl.gov/mpi/>

There are a few steps that you need to do before you begin working on your homework:

1. Make sure you can login to mo.ece.pdx.edu OR auto.ece.pdx.edu. If you don’t have access to that machine, you should contact CAT ([support@cat.pdx.edu](mailto:support@cat.pdx.edu)).

(use Putty to login)

Type “pwd” at the command prompt, it will show current directory is /u/huangy

(Note the current directory above is instructor’s current directory.)

1. Create a subdirectory under your home directory, and copy this file

mkdir hw4

cd hw4

cp /home/huangy/redhat5/hw4/mpi\_sum.c .

The program mpi\_sum.c represents a naïve implementation of for a parallel program sums up the integers from 1 to MAX. MAX is a constant defined in the program

1. Compile the sample program with this command:

mpicc -o sum mpi\_sum.c

This compiles the program using the MPI library to the executable file “sum” using the default C compiler. You should make sure that you have /usr/bin in your path so that you can find the C compiler.

1. Setup your mpi runtime environment. First, you need to create a file “.mpd.conf” in your home directory and write a line to it:

cd $HOME (Your home directory)

touch .mpd.conf

chmod 600 .mpd.conf

Yuchen notes: after I type cd $HOME, then type pwd, the screen shows /u/huangy

Just type the above 3 commands one after another, the above 3 commands are:

cd $HOME

touch .mpd.conf

chmod 600 .mpd.conf

Then use an editor to write a line in your .mpd.conf file like this:

MPD\_SECRETWORD=your\_secret\_word (use your own secret word)

Q&A: How to use an editor to write a line in .mpd.conf file?

Answer: just use the WinSCP software to edit and save. Click at the right bottom of the WinSCP window on the 2 hidden files, then the .mpd.conf file will show (initially this file won’t show). Double-click file .mpd.conf and write a line MPD\_SECRETWORD=YourSecretWord into this file. Then do Ctrl-S to save. Note here the “YourSecretWord” is the secret word you choose to write in.

Run the sample program using different parameters to experiment with its performance.

cd hw4

mpirun --oversubscribe -H localhost -np 1 ./sum

mpirun --oversubscribe -H localhost -np 16 ./sum

There seems to be some issue with running the MPI install across multiple nodes, so make sure the host list is always localhost or the name of the machine the user is running on (in this case, it is running on mo.ece.pdx.edu).

The above 2 mpirun commands shall be replaced by the following commands if user runs the mpirun command on auto.ece.pdx.edu while user login into mo.ece.pdx.edu:

mpirun --oversubscribe -H auto.ece.pdx.edu -np 1 ./sum

mpirun --oversubscribe -H auto.ece.pdx.edu -np 16 ./sum

These commands sum up the numbers from 1 to 1 billion on 1 and 16 processors, respectively, then print the sum and the execution time. The number of processors is specified after the “-np” flag.

Notes:

After I type:

mpirun --oversubscribe -H localhost -np 16 ./sum

The screen shows (starting MPI task 11 has started below):

huangy@mo:~/hw4$ mpirun --oversubscribe -H localhost -np 16 ./sum

MPI task 11 has started...

MPI task 12 has started...

MPI task 13 has started...

MPI task 14 has started...

MPI task 15 has started...

MPI task 4 has started...

MPI task 5 has started...

MPI task 6 has started...

MPI task 7 has started...

MPI task 8 has started...

MPI task 9 has started...

MPI task 10 has started...

MPI task 0 has started...

MPI task 1 has started...

MPI task 2 has started...

MPI task 3 has started...

sum = 500000000500000000.0

Time = 195118034 nanoseconds (0.195118034 sec)

After I type:

mpirun --oversubscribe -H localhost -np 1 ./sum

The screen shows:

huangy@mo:~/hw4$ mpirun --oversubscribe -H localhost -np 1 ./sum

MPI task 0 has started...

sum = 500000000500000000.0

Time = 2710886610 nanoseconds (2.710886610 sec)

# Assignment

Either modify the sample program or write your own program to redo homework 3’s problem using MPI.

You should turn in the following in Canvas:

1. A text file called “output4.txt” that contains a table showing the parallel speedup when running on 1, 2, 3 … 16 processors. Each line in the file should contain the following three fields separated by a tab: number of threads, time (in seconds), and speedup (vs. 1 thread)
2. A file containing your C program. The file name has to be “YourLastname\_homework4.c”. Your program should run, similar to mpi\_sum.c should be able to compile it and run it using similar commands to the example above. When your program runs, it should print temperatures for the required points every 200 time steps. At the end, it should print the elapsed time:

Time = 4507377000 nanoseconds (4.507377000 sec)

Notes: If you start from the sum.c example, then you have all the necessary calls you need to use to do send and receive stuff. So it’s up to you if you start to write your own program which will have a deeper learning curve or if you want to start from sum.c as the basis.