Assignment 4 – CSE536, Summer 2016

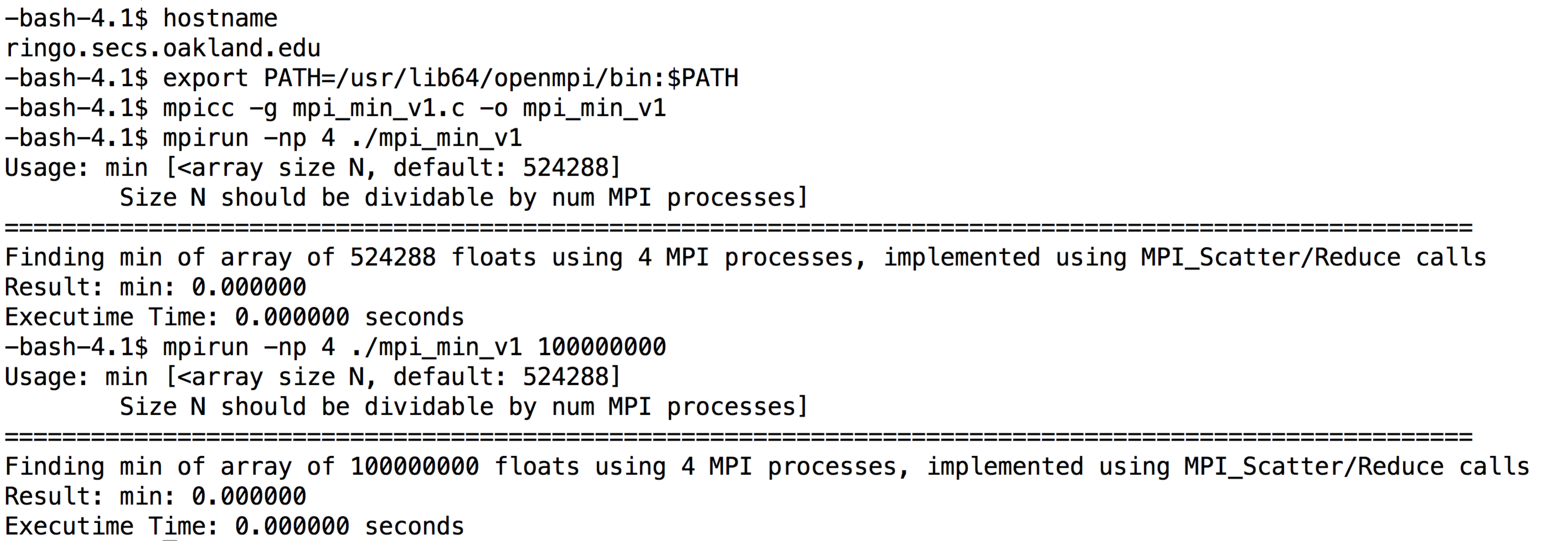
**Due: 11:55PM July 1 Friday, 2016**

**MPI Exercise for Finding the Minimum of an Array of Float**

In this assignment, you will implement two versions of procedures using MPI to find the minimum value from an array of float numbers. The program should output this minimum number. In the program, process 0 should initialize the array with random numbers and then distribute the array to other processes. Each process, including process 0, will then compute to find the minimum of its own local portion of the array. After that, each non-zero process should send its results back to the process 0 and process 0 collect them, and process 0 will then compute to find the final minimum value. If there are more than one location that have the minimum value, the program only need to output one of them.

The first version should be implemented mainly using MPI collective calls, i.e. MPI\_Scatter for distributing the array and MPI\_Reduce (MPI\_MIN reduction operations) for collecting the results. The second version should be implemented using only MPI\_Send and MPI\_Recv calls for array distribution and collecting results.

A skeleton file, mpi\_min.c, is provided, but feel free to create your own versions. I recommend you create two files, one for each of the two versions, so each version can be independently compiled and run. The MPI program we discussed during class will be a good starting point for you, e.g. scatter\_reduce.c and mpisendrecv.c. Use ringo/lennon.secs.oakland.edu to compile, run and collect results. Please export OpenMPI installation to your environment using “export PATH=/usr/lib64/openmpi/bin:$PATH” before making any mpi call. See below screen shot for a list of command to compile and run the program:



Your submission should include your implementation files and a max 3-page report for describing your implementation of each of the two versions, as well as for discussing the performance results. Please collect the performance results using 100M (100,000,000)-sized array, e.g. “mpirun –np 4 ./mpi\_min\_v1 100000000”

The assignment4-plot.xlsx file will help you to generate the figures from the results you will collect. While the development can be done from your laptop or any other computers, the results in the report should be collected from ringo.secs.oakland.edu (access info from http://cto.secs.oakland.edu/docs/pdf/linuxServers.pdf, and http://secs.oakland.edu/docs/pdf/vpn.pdf) using 1,2,4,8 and 16 processes. Please be noted that the machine is shared resource, overloaded use of the machine may cause incorrect performance results.

**Grading:**

**Functions implementations**

1. **90 points for implementation.**
2. **Report: 30 points.**

**For non-compliable code, you only receive max 60% of function implementations points. For compliable, but with execution errors and incorrectness, you receive max 70% of function implementation points. Please refer to the next page for the policy of academic conduct.**

Assignment policy:

Programming assignments are to be done individually. You may discuss assignments with others, but you must code your own solutions and submit them with a write-up in your own words. Indicate clearly the name(s) of student(s) you collaborated with, if any. Although homework assignments will not be pledged, per se, the submitted solutions must be your work and not copied from other students' assignments or other sources.

You may not transmit or receive code from anyone in the class in any way--visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading your code to someone), or in any other way.

You may not collaborate with people who are not your classmates, TAs, or instructor in any way. For example, you may not post questions to programming forums.

You may search the web and use any information that you find. However, you cannot take more than two lines of code from an external resource and actually include it in one of your assignments. Changing variable names or rewriting code you find does not void the "two line rule."

Any violations of these rules will be reported to the honor council. Check the syllabus for the late policy and academic conduct.