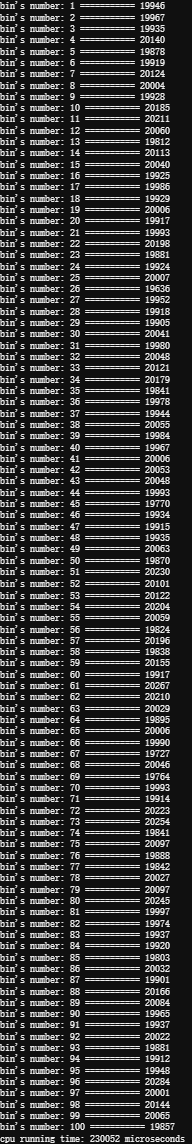
2. Develop a parallel histogramming program using C/C++ and OpenMPI. A histogram is used to summarize the distribution of data values in a data set. The most common form of histogramming splits the data range into equal-sized bins. For each bin, the number of data values in the data set that falls into that class are totaled. Your input to this program will be integers in the range 1-1,000,000 (use a random number generator that first generates the numbers first). Your input data set should contain 2 million integers. You will vary the number of bins. You should have as many OpenMPI processes as bins. You are suggested to use the sample batch script provided on Canvas

for specifying your OpenMPI configuration and running your program. Make sure to

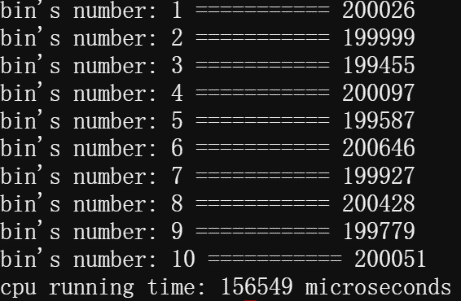
use the express partition, versus the short partition.

1. Assume there are 100 bins. Perform binning across nodes and processes using OpenMPI, and then perform a reduction on the lead node, combining your partial results. Run this on 2 and 4 nodes on Discovery. Your program should print out the number of values that fall into each bin. Compare the performance between running this on 2 and 4 nodes. Comment on the differences.



The running time of 2 nodes is 230052 microseconds, and the running time of 4 nodes is 1870 microseconds.

1. For this part, assume you have 20 bins. Perform binning on each process using OpenMPI, and then perform a reduction on the lead node, combining your partial results. Run this on 2 and 4 nodes on Discovery. Your program should print out the number of values that fall into each bin. Compare the performance between running this on 2 and 4 nodes. Comment on the differences.



The running time of 2 nodes is 156549 microseconds, and the running time of 4 nodes is 1269 microseconds.

1. Compare the performance measured in parts a.) and b.). Try to explain why one is faster than the other and run additional experiments to support your claims.

The results show that the running time is smaller when the number of bins is smaller than when the number of bins is larger. When the number of bins increases significantly, more comparisons are needed to determine which bin the number falls in. Also it becomes more difficult for processes to access memory and to communicate between nodes.