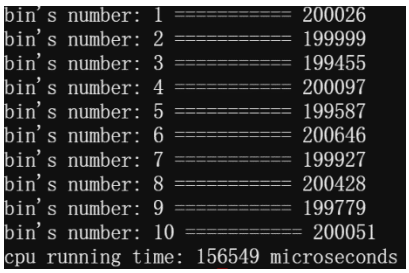
2. Develop a parallel histogramming program using C/C++ and OpenMPI. A histogram is used to summarize the distribution of data values in a data set. The most common form of histogramming splits the data range into equal-sized bins. For each bin, the number of data values in the data set that falls into 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.

a) 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.

```
bin's number: 1 ========== 19946
bin's number: 2 ========== 19967
bin's number: 3 ========== 19935
bin's number: 4 ========== 20140
bin's number: 5 ========== 19878
bin's number: 6 ========== 19919
bin's number: 7 ========== 20124
bin's number: 8 ========== 20004
bin's number: 9 ========== 19928
bin's number: 10 ========== 20185
bin's number: 11 ========== 20211
bin's number: 12 ========== 20060
bin's number: 13 ========== 19812
bin's number: 14 ========== 20113
bin's number: 15 ========== 20040
bin's number: 16 ========== 19925
bin's number: 17 ========== 19986
bin's number: 18 ========== 19929
bin's number: 19 ========== 20006
bin's number: 20 ========== 19917
bin's number: 21 ========== 19993
bin's number: 22 ========== 20198
bin's number: 23 ========== 19881
bin's number: 24 ========== 19924
bin's number: 25 ========== 20007
bin's number: 26 ========== 19636
bin's number: 27 ========== 19952
bin's number: 28 ========== 19918
bin's number: 29 ========== 19905
bin's number: 30 ========== 20041
bin's number: 31 ========== 19980
bin's number: 32 ========== 20048
bin's number: 33 ========== 20121
bin's number: 34 ========== 20179
bin's number: 35 ========== 19841
bin's number: 36 ========== 19978
bin's number: 37 ========== 19944
bin's number: 38 ========== 20055
bin's number: 39 ========== 19984
bin's number: 40 ========== 19967
bin's number: 41 ========== 20006
bin's number: 42 ========== 20053
bin's number: 43 ========== 20048
bin's number: 44 ========== 19993
bin's number: 45 ========== 19770
bin's number: 46 ========== 19934
bin's number: 47 ========== 19915
bin's number: 48 ========== 19935
bin's number: 49 ========== 20063
bin's number: 50 ========== 19870
bin's number: 51 ========== 20230
bin's number: 52 ========== 20101
bin's number: 53 ========== 20122
bin's number: 54 ========== 20204
bin's number: 55 ========== 20059
bin's number: 56 ========== 19824
bin's number: 57 ========== 20196
bin's number: 58 ========== 19838
bin's number: 59 ========== 20155
bin's number: 60 ========== 19917
bin's number: 61 ========== 20267
bin's number: 62 ========== 20210
bin's number: 63 ========== 20029
bin's number: 64 ========== 19895
bin's number: 65 ========== 20006
bin's number: 66 ========== 19990
bin's number: 67 ========== 19727
bin's number: 68 ========== 20046
bin's number: 69 ========== 19764
bin's number: 70 ========== 19993
bin's number: 71 ========== 19914
bin's number: 72 ========== 20223
bin's number: 73 ========== 20254
bin's number: 74 ========== 19841
bin's number: 75 ========== 20097
bin's number: 76 ========== 19888
bin's number: 77 ========== 19842
bin's number: 78 ========== 20027
bin's number: 79 ========== 20097
bin's number: 80 ========== 20245
bin's number: 81 ========== 19997
bin's number: 82 ========== 19974
bin's number: 83 ========== 19937
bin's number: 84 ========== 19920
bin's number: 85 ========== 19803
bin's number: 86 ========== 20032
bin's number: 87 ========== 19901
bin's number: 88 ========== 20166
bin's number: 89 ========== 20084
bin's number: 90 ========== 19965
bin's number: 91 ========== 19937
bin's number: 92 ========== 20022
bin's number: 93 ========== 19881
bin's number: 94 ========== 19912
bin's number: 95 ========== 19948
bin's number: 96 ========== 20284
bin's number: 97 ========== 20001
bin's number: 98 ========== 20144
bin's number: 99 ========== 20065
bin's number: 100 ========== 19857
cpu running time: 230052 microseconds
```

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

b) 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.

c) 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.