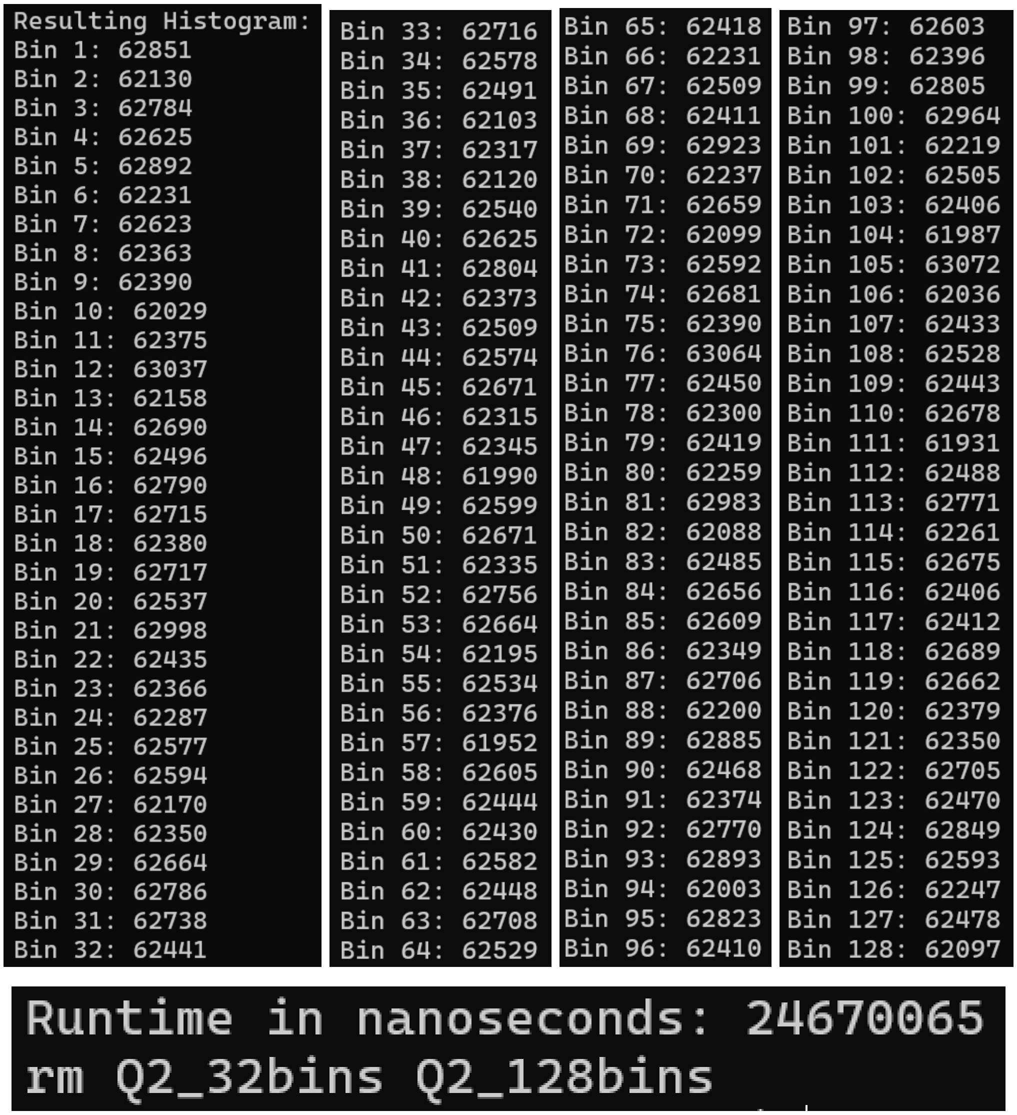
Justin Bahr
NUID: 001592707

High Performance Computing
Homework 4 - Question 2

I ran my programs on the Explorer Cluster with 16 processes per node (except for the run with 32 bins on 4 nodes, which only uses 8 processes per node).

a. Below is an example result of program run with 128 bins:



```
Resulting Histogram:
Bin 1: 62851
Bin 2: 62130
Bin 3: 62784
Bin 4: 62625
Bin 5: 62892
Bin 6: 62231
Bin 7: 62623
Bin 8: 62363
Bin 9: 62390
Bin 10: 62029
Bin 11: 62375
Bin 12: 63037
Bin 13: 62158
Bin 14: 62690
Bin 15: 62496
Bin 16: 62790
Bin 17: 62715
Bin 18: 62380
Bin 19: 62717
Bin 20: 62537
Bin 21: 62998
Bin 22: 62435
Bin 23: 62366
Bin 24: 62287
Bin 25: 62577
Bin 26: 62594
Bin 27: 62170
Bin 28: 62350
Bin 29: 62664
Bin 30: 62786
Bin 31: 62738
Bin 32: 62441
```

```
Bin 33: 62716
Bin 34: 62578
Bin 35: 62491
Bin 36: 62103
Bin 37: 62317
Bin 38: 62120
Bin 39: 62540
Bin 40: 62625
Bin 41: 62804
Bin 42: 62373
Bin 43: 62509
Bin 44: 62574
Bin 45: 62671
Bin 46: 62315
Bin 47: 62345
Bin 48: 61990
Bin 49: 62599
Bin 50: 62671
Bin 51: 62335
Bin 52: 62756
Bin 53: 62664
Bin 54: 62195
Bin 55: 62534
Bin 56: 62376
Bin 57: 61952
Bin 58: 62605
Bin 59: 62444
Bin 60: 62430
Bin 61: 62582
Bin 62: 62448
Bin 63: 62708
Bin 64: 62529
```

```
Bin 65: 62418
Bin 66: 62231
Bin 67: 62509
Bin 68: 62411
Bin 69: 62923
Bin 70: 62237
Bin 71: 62659
Bin 72: 62099
Bin 73: 62592
Bin 74: 62681
Bin 75: 62390
Bin 76: 63064
Bin 77: 62450
Bin 78: 62300
Bin 79: 62419
Bin 80: 62259
Bin 81: 62983
Bin 82: 62088
Bin 83: 62485
Bin 84: 62656
Bin 85: 62609
Bin 86: 62349
Bin 87: 62706
Bin 88: 62200
Bin 89: 62885
Bin 90: 62468
Bin 91: 62374
Bin 92: 62770
Bin 93: 62893
Bin 94: 62003
Bin 95: 62823
Bin 96: 62410
```

```
Bin 97: 62603
Bin 98: 62396
Bin 99: 62805
Bin 100: 62964
Bin 101: 62219
Bin 102: 62505
Bin 103: 62406
Bin 104: 61987
Bin 105: 63072
Bin 106: 62036
Bin 107: 62433
Bin 108: 62528
Bin 109: 62443
Bin 110: 62678
Bin 111: 61931
Bin 112: 62488
Bin 113: 62771
Bin 114: 62261
Bin 115: 62675
Bin 116: 62406
Bin 117: 62412
Bin 118: 62689
Bin 119: 62662
Bin 120: 62379
Bin 121: 62350
Bin 122: 62705
Bin 123: 62470
Bin 124: 62849
Bin 125: 62593
Bin 126: 62247
Bin 127: 62478
Bin 128: 62097
```

```
Runtime in nanoseconds: 24670065
rm Q2_32bins Q2_128bins
```

Figure 1. Screenshot of 128 Bins Result

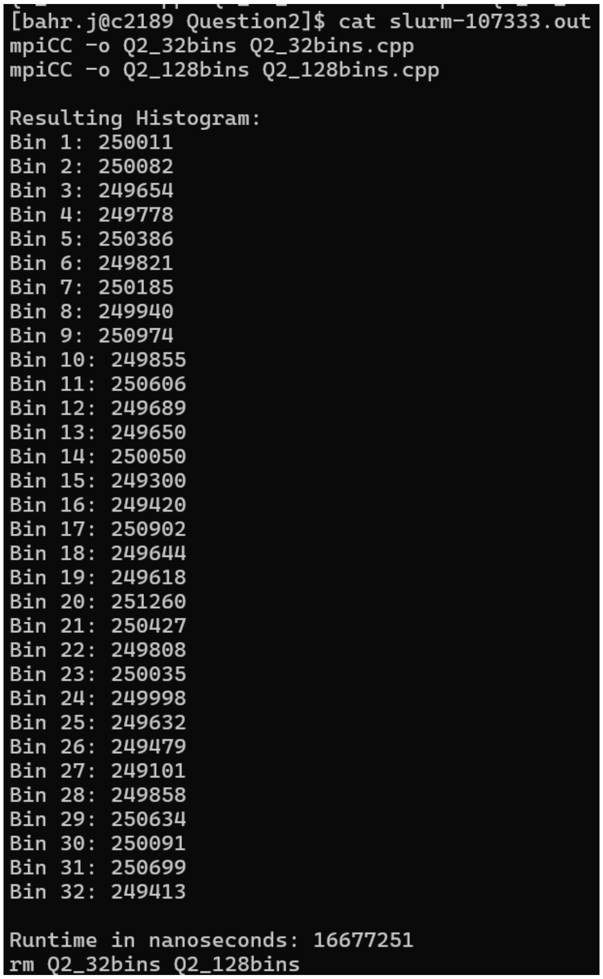
Since there are 8 millions random integers to histogram and 128 bins, each bin should have about 8 million / 128 = 62,500 integers, which resembles the results above.

128 Bins	2 Nodes	4 Nodes	8 Nodes
Trial	Runtime (ms)	Runtime (ms)	Runtime (ms)
1	18.539344	24.811686	37.20247
2	18.650524	24.706874	27.847574
3	18.715954	24.656571	28.290499
Average	18.635274	24.72504367	31.11351433

Figure 2. Runtimes for 128 Bins with 2, 4, and 8 Nodes

In this experiment, increasing the number of nodes used also increased the runtime. While increasing the number of nodes used decreases the amount of work each node needs to do, it also increases the overhead cost of MPI communication between nodes, which is significantly longer than shared memory latency. However, this is using a fixed size of 8 million integers to judge strong scaling. If the problem size is increased past a point where the overhead communication costs are negligible, it will likely become advantageous to employ MPI across multiple nodes.

b. Below is an example result of program run with 32 bins



```
[bahr.j@c2189 Question2]$ cat slurm-107333.out
mpiCC -o Q2_32bins Q2_32bins.cpp
mpiCC -o Q2_128bins Q2_128bins.cpp

Resulting Histogram:
Bin 1: 250011
Bin 2: 250082
Bin 3: 249654
Bin 4: 249778
Bin 5: 250386
Bin 6: 249821
Bin 7: 250185
Bin 8: 249940
Bin 9: 250974
Bin 10: 249855
Bin 11: 250606
Bin 12: 249689
Bin 13: 249650
Bin 14: 250050
Bin 15: 249300
Bin 16: 249420
Bin 17: 250902
Bin 18: 249644
Bin 19: 249618
Bin 20: 251260
Bin 21: 250427
Bin 22: 249808
Bin 23: 250035
Bin 24: 249998
Bin 25: 249632
Bin 26: 249479
Bin 27: 249101
Bin 28: 249858
Bin 29: 250634
Bin 30: 250091
Bin 31: 250699
Bin 32: 249413

Runtime in nanoseconds: 16677251
rm Q2_32bins Q2_128bins
```

Figure 3. Screenshot of 32 Bins Result

Since there are 8 millions random integers to histogram and 32 bins, each bin should have about 8 million / 32 = 250,000 integers, which resembles the results above.

32 Bins	2 Nodes	4 Nodes
Trial	Runtime (ms)	Runtime (ms)
1	16.940292	23.131445
2	16.740868	24.088682
3	16.802406	23.62041
Average	16.82785533	23.61351233

Figure 2. Runtimes for 32 Bins with 2 and 4 Nodes

Similar to part (a), increasing the from 2 to 4 nodes also increased the runtime due to the overhead cost of MPI communication across nodes.

c. For both 2 and 4 nodes, histogramming ran faster with 32 bins compared to with 128 bins. This is because the local histogram arrays for each process only require 32 indices compared to 128. Increasing the size of the local histogram increases the size of the messages passed between MPI processes during histogram reduction, leading to longer runtimes. I confirmed this by increasing the number of bins to 256, which led to an increased runtime of 24.82 ms with 2 nodes compared to 18.64 ms with 2 nodes and 128 bins.