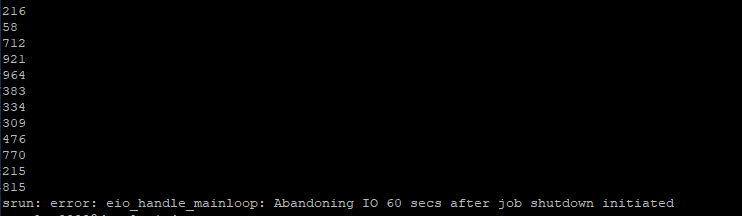
Summary:

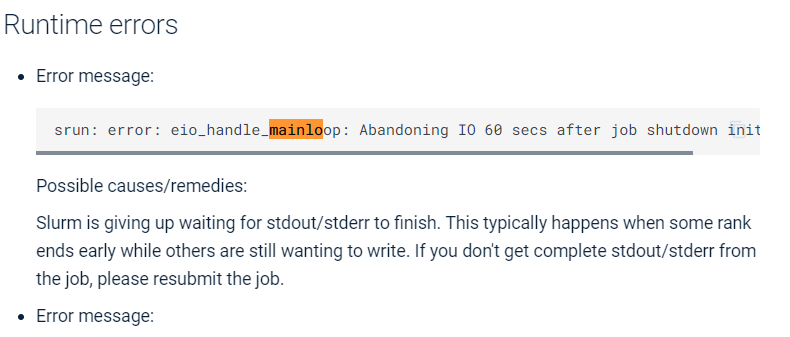
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Machine | # of Processes (Parallel) | Array Size | Parallel Time (s) | Serial Time (s) |
| ASC/DMC | 10 | 1000 | 0.37 | 0.09 |
| ASC/DMC | 10 | 999 | 0.18 | 0.09 |
| Local Windows | 10 | 100,000 | 1.344 | 172.9 |
| Local Windows | 10 | 99,999 | 1.301 | 171.6 |

**Compiling for the MPI program was done using:**  
mpiCC parallelSort.cpp -o pSort.out  
  
  
**Compiling for the serial program:**  
g++ serialSort.cpp -o sSort.out  
  
*script file should have only* ***one*** *argument following* the execution call  
*e.g. using the script: srun --mpi=pmi2 pSort.out 1000*   
creates an array with 1000 randomly generated numbers

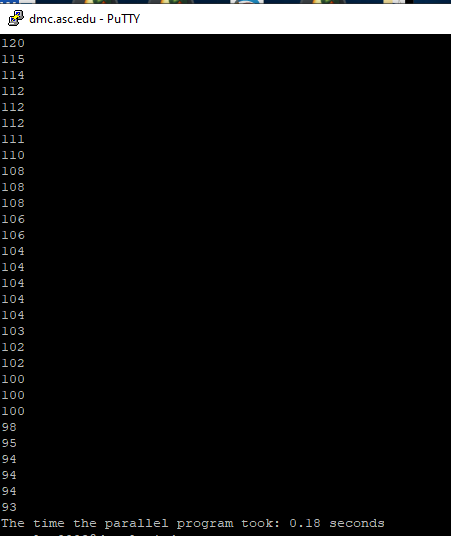
For this assignment, I attempted to sort an array with 10,000 numbers but was given this error by SLURM:

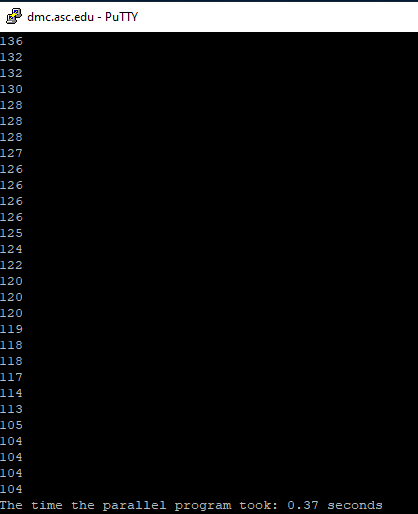


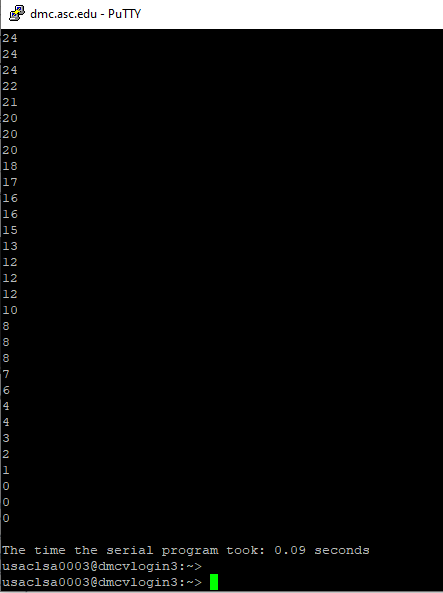
According to this documentation, it seems that there seems to be a limit to the number of things I can print to the IO stream:

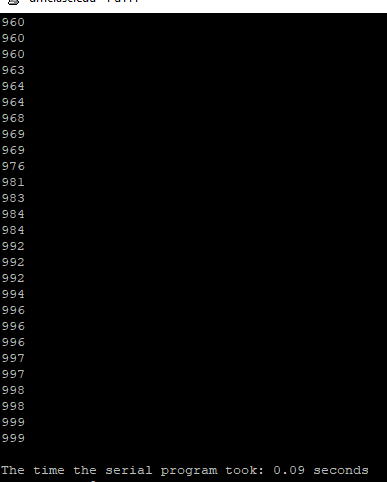


This was resolved reducing the array size. Using 10 processors, trials were completed for an odd example (array size = 999) and an even example (size = 1000). The even trial completed in 0.37 seconds and the odd trial completed in 0.18 seconds as shown:



  
  
After this, even and odd serial sorts were completed. However, the times were 0.09 seconds and \_\_\_ seconds, respectively. This is likely because the size of the arrays isn’t significant to give parallel sorting an advantage, and is possibly due to the fact that using 10 processors for this size causes more communication overhead. The serial results are as shown:





To make this difference more apparent and demonstrate that the parallel algorithm can outperform the serial algorithm with more data, trials were run on a local computer using MPI for Windows for sizes 100,000 and 99,999. As such, we found the serial program took 171.615, and 172.925 seconds. We also found that the parallel program took 1.344 and 1.301 seconds.

