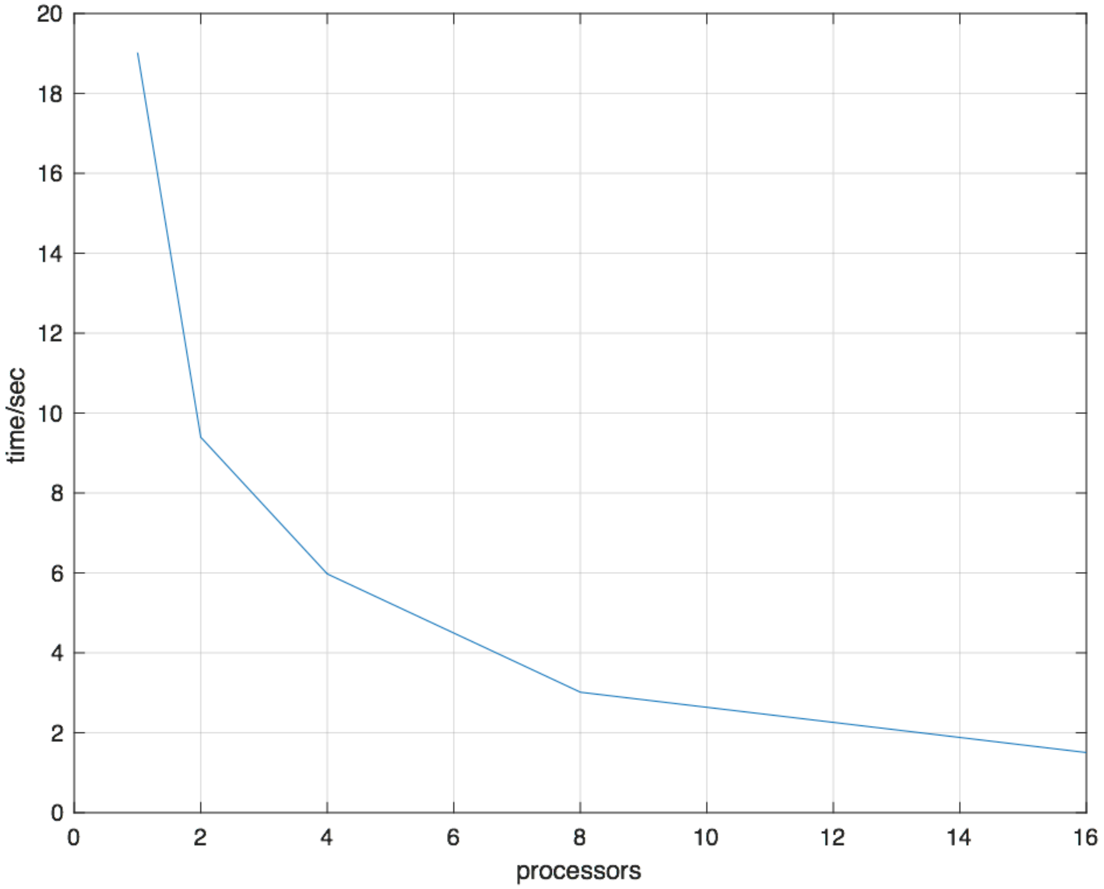
CSE6220-Proj1

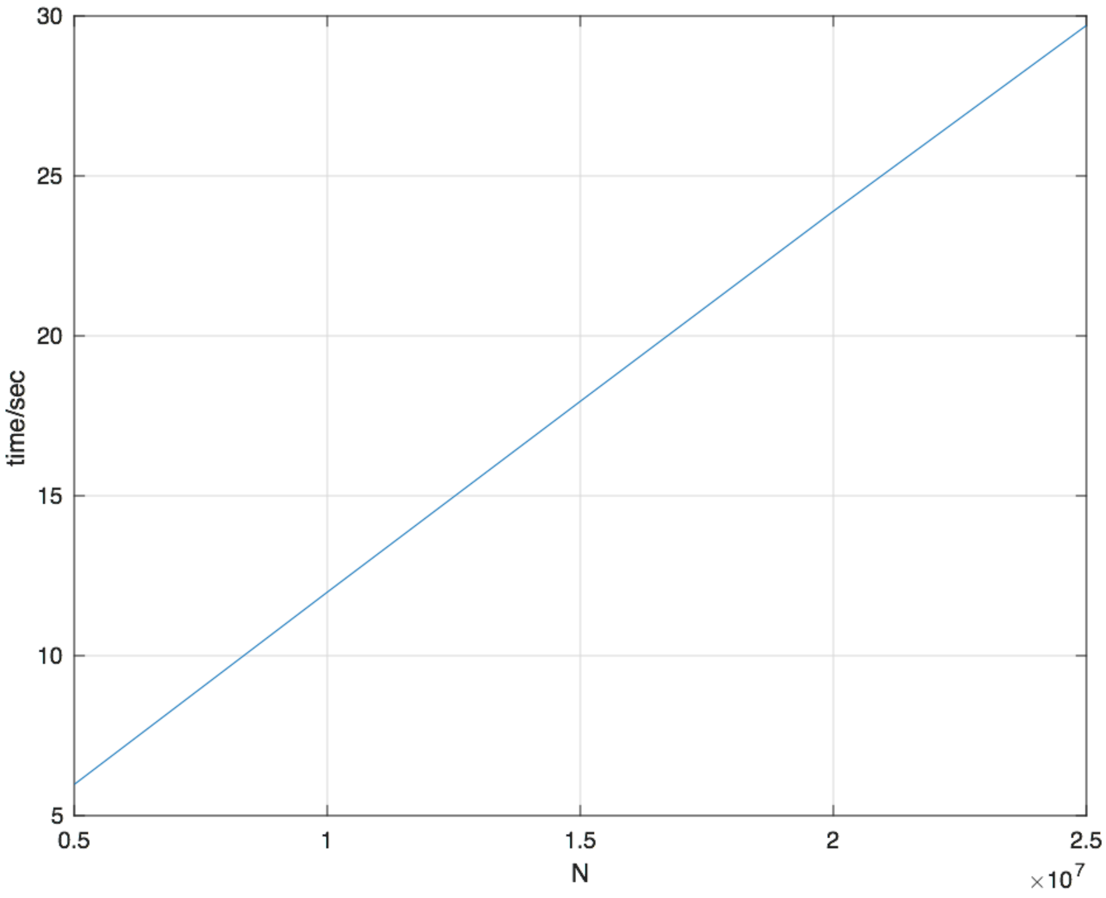
Jingfan Sun, 903141537

Runtime VS. Processors



As shown in the figure above, as the number of processors increase, the runtime decreases. However, it is not a linear relationship. The more processors involved, the longer it takes to communicate between processors. The `reduce` function also has to wait for the last processor to send its calculated value back. So it makes sense that the speedup drops as the number of processors increase.

Runtime VS. N



The variable N is increased from 5000000 to 25000000. Since it is just a `for` loop inside, the runtime increase linearly as N increases.

Readme

These results as well as those in `output.txt` file are based on execution on the laptop, 2.7 GHz Intel Core i5 processor.

The environment on the laptop is

```
# mpicxx --version
Apple LLVM version 10.0.0 (clang-1000.10.44.4)
Target: x86_64-apple-darwin17.7.0
# mpirun --version
HYDRA build details:
    Version:                                  3.2
    CC:                               clang
    CXX:                              clang++
```

On the Pace cluster, the compilation is good using `-std=c++11` but calculation is not correct. I still didn't findout the problem. That's why the local results are submitted.