Shane Alvarez

CS 491 Parallel Programming

Homework 4

How optimization was performed

The optimization is heavily based off the past optimization which used pthreads. First, global data is initialized. Then, MPI is initialized, cloning to the other processes. Using the rank of each processes, access to the xold and xnew arrays are partitioned evenly between all processes. The working loop is then broken into the following sections:

1. Begin Calculations of xnew within the partition. For each newly calculated row:
    a. If it is the first row, send that row to the process with the next lowest rank (async)
    b. If it is the last row, send that row to the process with the next greater rank (async)
2. Begin receiving the data sent from the other processes into xold
3. Aggregate the maximum difference from each process to find the global max (blocking)
4. Copy partition of xnew into partition of xold (synchronous)
5. Have the median process send the center of xnew to the root process (synchronous)
6. Have the root process check for the stop condition and increment iter
7. Synchronize the iteration (blocking)
8. Have all processes await the data transactions from set 1 and 2 (blocking)

The I/O overhead of the sending and receiving were the greatest issue, and thus were handled asynchronously with many intermediate operations to make the best use of time. Additionally, since each partition only needs to access itself and a row above and below them, only the boundary rows of each partition needed to be synchronized.

Based on the results, it would seem that this problem is not very well suited for MPI distributed on many machines. The I/O delay is much performance gain of the concurrent sections.

Running the Code

The code can be built and run using the following commands:

```
mpicc -std=c99 -g 491hw4-optimized.c -o jacobi
mpirun -H cauchy,fermat,mckusick,naur -npernode 4 ./Jacobi
```

Results

4 Processes

Execution #	MFLOPS	Time
1	936.8	1.181
2	929.4	1.190
3	917.1	1.206
Average	**927.8**	**1.192**

8 Processes

Execution #	MFLOPS	Time
1	288.9	3.829
2	295.6	3.742
3	292.5	3.781
Average	**292.3**	**3.784**

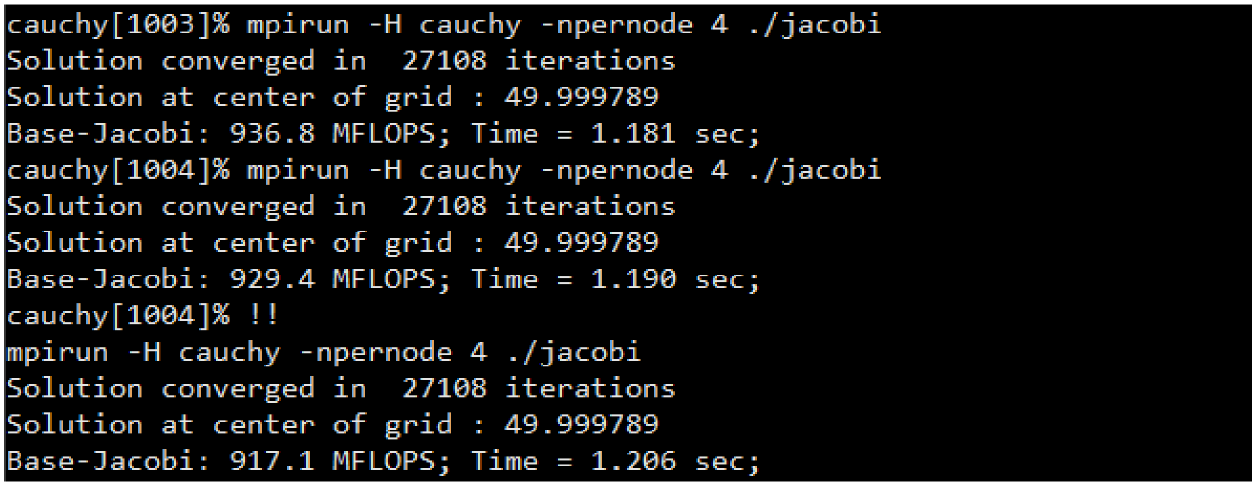
12 Processes

Execution #	MFLOPS	Time
1	260.8	4.241
2	249.4	4.435
3	244.3	4.528
Average	**251.5**	**4.401**

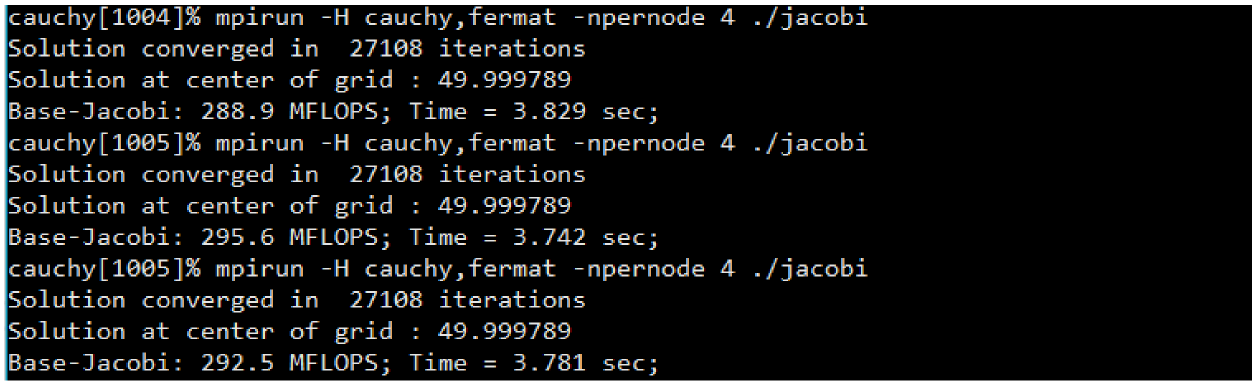
16 processes

Execution #	MFLOPS	Time
1	126.0	8.780
2	130.0	8.509
3	126.1	8.775
	127.4	**8.688**

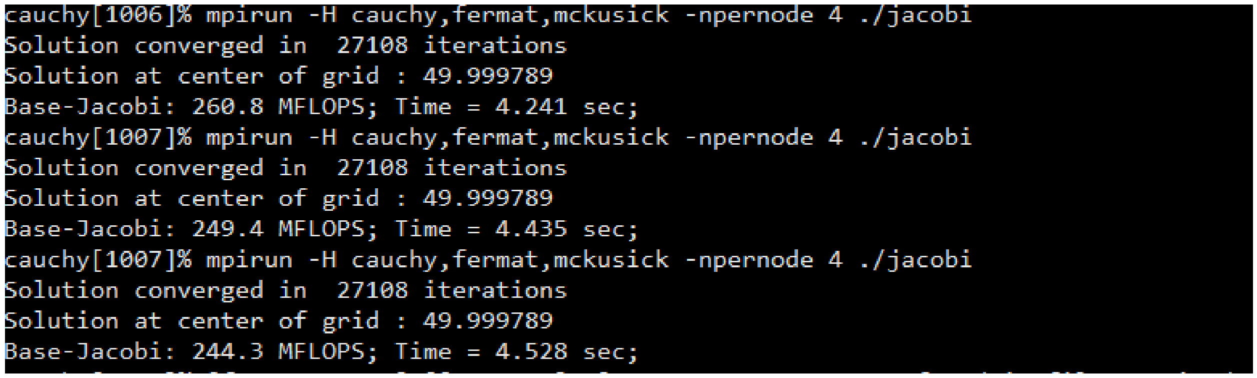
Raw Data



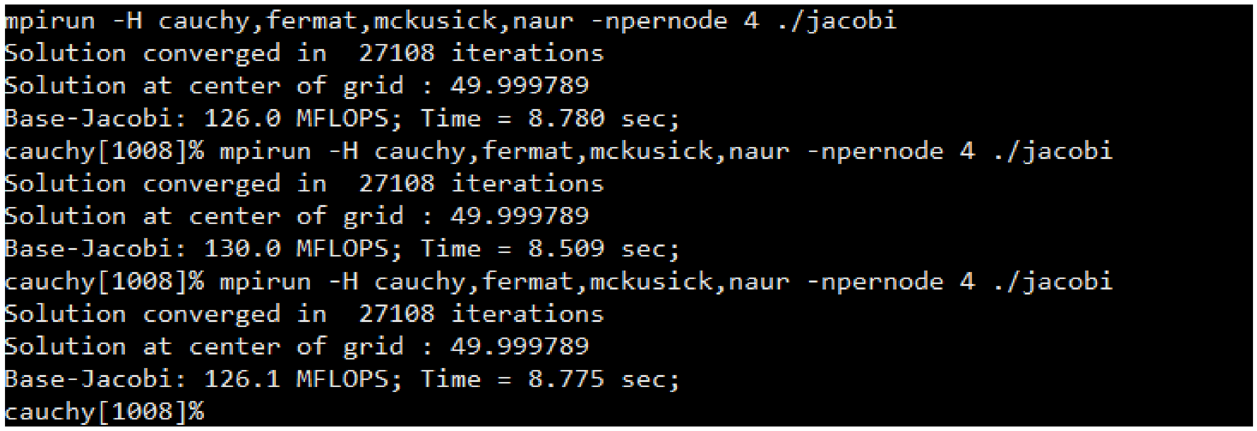
```
cauchy[1003]% mpirun -H cauchy -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 936.8 MFLOPS; Time = 1.181 sec;
cauchy[1004]% mpirun -H cauchy -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 929.4 MFLOPS; Time = 1.190 sec;
cauchy[1004]% !!
mpirun -H cauchy -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 917.1 MFLOPS; Time = 1.206 sec;
```



```
cauchy[1004]% mpirun -H cauchy,fermat -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 288.9 MFLOPS; Time = 3.829 sec;
cauchy[1005]% mpirun -H cauchy,fermat -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 295.6 MFLOPS; Time = 3.742 sec;
cauchy[1005]% mpirun -H cauchy,fermat -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 292.5 MFLOPS; Time = 3.781 sec;
```



```
cauchy[1006]% mpirun -H cauchy,fermat,mckusick -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 260.8 MFLOPS; Time = 4.241 sec;
cauchy[1007]% mpirun -H cauchy,fermat,mckusick -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 249.4 MFLOPS; Time = 4.435 sec;
cauchy[1007]% mpirun -H cauchy,fermat,mckusick -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 244.3 MFLOPS; Time = 4.528 sec;
```



```
mpirun -H cauchy,fermat,mckusick,naur -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 126.0 MFLOPS; Time = 8.780 sec;
cauchy[1008]% mpirun -H cauchy,fermat,mckusick,naur -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 130.0 MFLOPS; Time = 8.509 sec;
cauchy[1008]% mpirun -H cauchy,fermat,mckusick,naur -npernode 4 ./jacobi
Solution converged in  27108 iterations
Solution at center of grid : 49.999789
Base-Jacobi: 126.1 MFLOPS; Time = 8.775 sec;
cauchy[1008]%
```