

Jeremy Bonnell
CS4230
Programming Assignment #4

README

I was able to parallelize the code using MPI and minor amounts of openmp but was unsuccessful in getting any kind of speedup. There were several openmp modifications I made that only worked part of the time. I did not include these modifications in the final version of my code because I was unsure if I was gathering false data or if it was something weird happening on CHPC. The first version of my code used MPI to parallelize the inner-k loop to solve for alpha, beta, and gamma using MPI_Allreduce(MPI_SUM). I later expanded the code to the outer-j loop and used this version because it was much faster. This method has each core solve for entire rows to get 'c' and 's' values and then used MPI_Allgather() to combine all the 'c' and 's' values for one 'i' iteration into an array. I also used MPI_Allreduce(MPI_MAX) to allow each core to have the proper value of 'converge'. I tried both a cyclic and block distribution but settled on the block distribution because the results were more refined. The timing was comparable for both distributions. The number of threads used in my code is '4' and is declared within the code (not user input).

TO use my compile, run, and Validate my code, I used the following commands:

- mpic++ -O3 seq_SVD.cpp -o seq_SVD
- mpic++ -O3 -fopenmp par_SVD.cpp -o par_SVD
- g++ -g -Wall -o Validation Validation.cpp
-
- mpirun -np 1 ./seq_SVD 1024 1024 -t -d
- mpirun -np 8 ./par_SVD 1024 1024 -t -d
- ./Validation -p

The file "par_SVD.cpp" was done using a block distribution. The file "par_cyclic_SVD.cpp" can be compiled to test cyclic results with:

- mpic++ -O3 -fopenmp par_cyclic_SVD.cpp -o par_cyclic_SVD
-
- mpirun -np 8 ./par_cyclic_SVD 1024 1024 -t -d
- ./Validation -p

SEQUENTIAL RESULTS:

```
$ mpirun -n 1 ./seq_SVD 1024 1024 -t -d
```

iterations: 15

Time: 53059 ms.

MPI/OPENMP (32-NODE [8 nodes - 4 cores/node]) RESULTS:

```
$ mpirun -n 8 ./par_SVD 1024 1024 -t -d
```

iterations: 20
Time: 139175 ms.

VALIDATION RESULTS:

NOT VALID!

difference in U: 8.29459e-10
difference in V: 1.18307e-08
difference in S: 9.09495e-12

**RESULTS OFF ONLY SLIGHTLY. ALWAYS GOT VALID RESULTS WHEN RUNNING ANYTHING
UNDER 1024. I'M NOT SURE WHY.**