Russell Folk
CS 491/CS 521 Parallel Programming
Homework 4 Report

MPI Optimizations

Disclaimer

For these tests, I chose to run on the computers named borg, granville, lamarr, and perlman. These machines were chosen from the ones available because they were equipped with the best processors.

Changes to the algorithm

The most important change that I made to the algorithm was in how I separated the data to be sent to the various nodes. Due to the structure of the algorithm, every row needs to have access to the row before and the row after in order to perform its calculation. Thus, I send the first element from 0 to number of rows being handled + 2. Because the first row actually changed is 1, I have the buffer behind and the buffer in front of the group. I then offset the next group by the number of rows being handled and repeat. In this way all the required data is sent and the calculations are capable of being performed.

Future optimizations to the algorithm

Given additional time I would make the following changes to the algorithm to increase optimization:

- Make changes to the data transmitted so that only the 2 buffer-rows data and a diff is returned after the first iteration. This would allow for minimum messages to be sent and a solution to be discovered quicker.
- Structure the sent data so that each host got all the rows together, that is if one host is running 4 processes it would get the data sequentially to minimize messages back and forth between the buffer-rows.
- Create a hybrid architecture that incorporates OpenMP. MPI would be used to handle the passed messages to the different hosts and then OpenMP would take over to handle the per process operations.

Tests run

For the final experiments I ran a series of 3 tests with each of the assigned configurations: 1 process per node, 2 processes per node, 3 processes per host, and 4 processes per host. To get the best performance I utilized the 3 fastest CPUs in the cluster and 1 of the faster ones. Only 4 hosts were used because of the configurations specified by the assignment, however extensive testing showed that performance degraded as more hosts were added. This run can be simulated if the attached `runner.csh` is launched from the borg host.

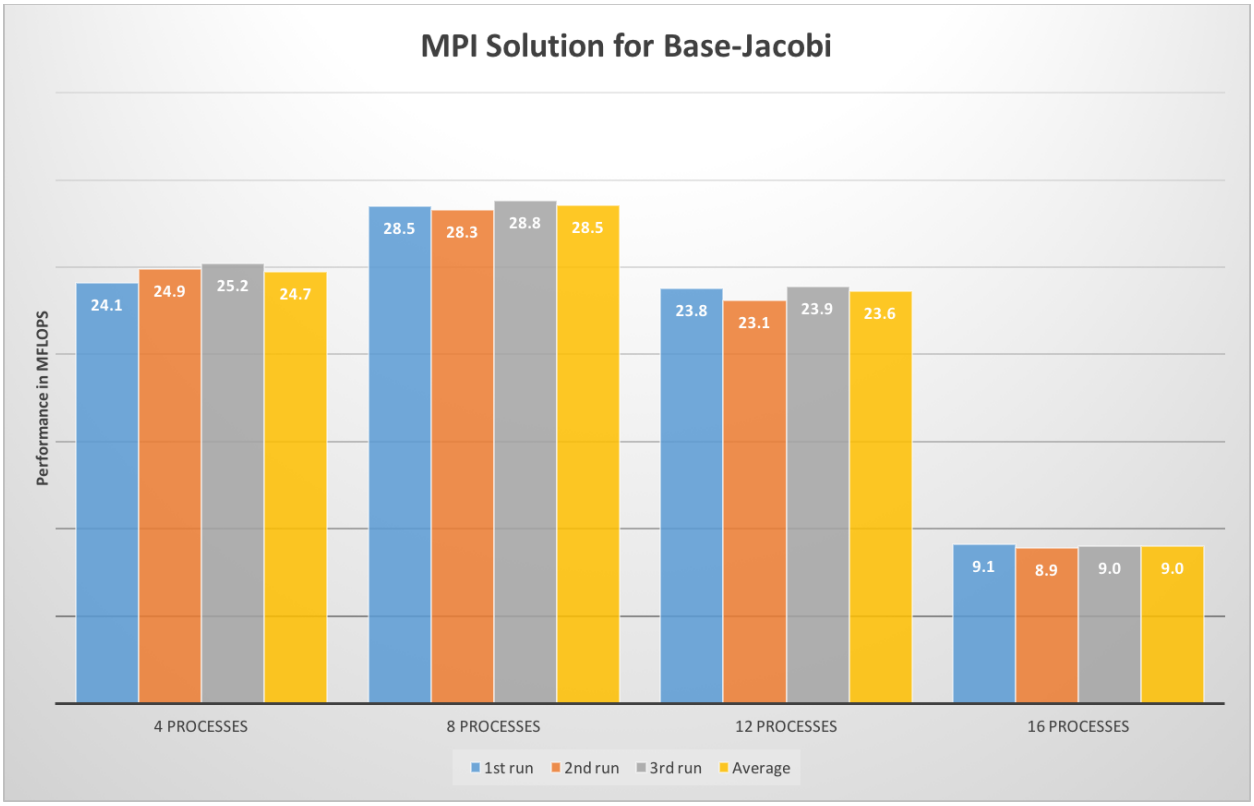

FIGURE 1: PERFORMANCE MEASURED IN MFLOPS OF MPI CONFIGURATIONS THIS GRAPH SHOWS THE THREE RUNS AND AN AVERAGE OF THE 4 CONFIGURATIONS SPECIFIED ON THE MPI CLUSTER.

Results

Obviously, as seen in Figure 1, the results beyond 12 processes is horrible. The best results that I ever saw were when running 2 processes per node. However, as the CPUs that I was running on are quad core, I would have expected the 4 processes per node to be equally good.

I believe that the lack of time to implement the features mentioned above explains the rather lackluster performance of the MPI runs. Furthermore, the MPI cluster happens to be a set of undergrad computers which are in heavy use and could be limiting performance of the program.