Blair Todd

CS4380

# Project 2 Report

Question 2.1a) Check the contents of the collatz_MPI12.sub submission script. How many processes are being launched? How many compute nodes do these processes run on?

- MPI12 launches 12 processes and runs on 1 node

*Question 2.1b) What compute times do you get (in seconds)? Present the times in a table for increasing process counts from left to right and increasing problem sizes from top to bottom. Use three digits after the decimal point for all compute times (even if the last digit is a zero).

| Compute Times | 1 | 12 | 24 |
|---|---|---|---|
| 500000 | 0.133 | 0.013 | 0.008 |
| 5000000 | 1.589 | 0.168 | 0.100 |
| 50000000 | 18.409 | 1.868 | 1.166 |

Question 2.1c) Based on the compute times, calculate the speedups relative to running the MPI code with one process. Present the speedups in the same format as the compute times but show only two digits after the decimal point.

| Speedups | 1 | 12 | 24 |
|---|---|---|---|
| 500000 | 1 | 10.23 | 16.625 |
| 5000000 | 1 | 9.45 | 15.89 |
| 50000000 | 1 | 9.85 | 15.78 |

Question 2.1d) Based on the speedups, calculate the efficiencies relative to running the MPI code with one process. Present the efficiencies in the same format as the speedups.

| Efficiencies | 1 | 12 | 24 |
|---|---|---|---|
| 500000 | 1 | .85 | .69 |
| 5000000 | 1 | .78 | .66 |
| 50000000 | 1 | .82 | .65 |

Question 2.1e) Explain why no barrier is needed before stopping the timer, i.e., why the compute time printed by process 0 is guaranteed to be the compute time of the slowest process.

- No barrier is needed due to the MPI_Reduce function used which has implied synchronization.

Question 2.2a) Run the code with the submission scripts at /home1/00976/burtsche/Parallel/fractal_MPI*.sub. What compute times do you get (in seconds)? Present them in a table for increasing process counts from left to right and “increasing” problem sizes from top to bottom (i.e., use the same order as in the submission script). Use three digits after the decimal point.

| Compute Time | 1 (Serial) | 16 |
|---|---|---|
| Frames: 64<br>Width: 512 | 3.810 | 0.293 |
| Frames: 128<br>Width: 512 | 9.361 | 1.002 |
| Frames: 64<br>Width: 1024 | 15.215 | 1.173 |
| Frames: 128<br>Width: 1024 | 37.418 | 3.961 |

Question 2.2b) Based on the efficiencies (which you must compute), does the code scale well to 16 processes for all four inputs?

| Efficiencies | 1(Serial) | 16 |
|---|---|---|
| Frames:64<br>Width:512 | 1 | .81 |
| Frames:128<br>Width:512 | 1 | .58 |
| Frames:64<br>Width:1024 | 1 | .81 |
| Frames:128<br>Width:1024 | 1 | .59 |

- The code scales well for all four inputs (depending on frames time does change but still scales)

Question 2.2c) How many cores does a “normal” compute node in Lonestar 5 have? How many sockets does it have?

- 24 cores
- Dual Sockets (2)