**High Performance Computing**

**Homework #6**

**Due: Tuesday April 26 2016 by 11:59 PM**

**Email-based help Cutoff: 5:00 PM on Mon, April 25 2016**

Points: 5

|  |  |
| --- | --- |
| **Name:** | **Henry Ni** |

## Experimental Platform

## The experiments documented in this report were conducted on the following platform:

|  |  |
| --- | --- |
| Component | Details |
| CPU Model | Intel Xeon X5550 |
| CPU/Core Speed | 2.67GHz |
| Main Memory (RAM) size | 24725392 kB |
| Operating system used | Linux mualhpcp01.hpc.muohio.edu 2.6.32-279.14.1.el6.x86\_64 #1 SMP Tue Nov 6 23:43:09 UTC 2012 x86\_64 x86\_64 x86\_64 GNU/Linux |
| Interconnect type & speed (if applicable) | Not applicable |
| Was machine dedicated to task (yes/no) | Yes (via a qsub job) |
| Name and version of C++ compiler (if used) | icpc 15.0.0, compatible g++ 4.9.2 |
| Name and version of Java compiler (if used) | None |
| Name and version of other non-standard software tools & components (if used) | None |

## Runtime statistics

The runtime statistics show in the adjacent table was collected on a varying number of compute nodes using numbers\_1000.txt. The data from just one run is shown (even though normally Avg±CI from 5 runs would be reported). **Note**: Prior to collecting the data I verified that the output from programs are correct.

I used numbers\_10000.txt as there was no numbers\_1000.txt file in the copied directory.

|  |  |
| --- | --- |
| **Nodes**  **Used** | **Elapsed time**  **(seconds)** |
| 2 | 13.16 |
| 4 | 4.69 |
| 6 | 2.88 |
| 8 | 2.27 |

## Inferences & Discussions

Now, using the data from the above table discuss if your solution is providing good performance improvements as the number of processes/nodes is increased

|  |
| --- |
| Using the data above, it is clear the program is able to take advantage of additional compute power. Although the jump from 6 to 8 nodes doesn’t return as much of a performance gain as the leap from 2 to 4 nodes, there were still improvements at every stage. |