Ainsley Uhlenkott

CSCI 322 - Winter 2014
Assignment 3 Writeup

# Compiling and Running

There are 3 programs included in the project: `central.c`, `symmetric.c`, and `ring.c`.

For compilation, use:

```
make central
make symmetric
make ring
```

Each program has a corresponding shell script to simplify startup. To run the programs, use:

```
./run_central
./run_symmetric
./run_ring
```

You will be prompted for the number of processes and number of loop iterations to use.

# Results

Each program was run 5 times per process/iteration configuration, and the minimum run times for each configuration were recorded. All times are in seconds.

**central**

| number of processes | number of iterations | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 0.484 | 0.566 | 1.133 | 2.613 | 4.924 |
| **2** | 0.484 | 0.566 | 1.207 | 2.757 | 5.242 |
| **4** | 0.484 | 0.642 | 1.428 | 3.107 | 6.127 |
| **8** | 0.484 | 0.792 | 1.580 | 3.314 | 6.436 |
| **16** | 0.492 | 0.939 | 1.961 | 3.889 | 7.572 |

**symmetric**

| number of processes | number of iterations | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 0.484 | 0.566 | 1.133 | 2.613 | 4.924 |
| **2** | 0.484 | 0.566 | 1.207 | 2.757 | 5.242 |
| **4** | 0.484 | 0.642 | 1.428 | 3.107 | 6.127 |
| **8** | 0.484 | 0.817 | 1.578 | 3.310 | 6.424 |
| **16** | 0.492 | 0.930 | 1.923 | 3.859 | 7.514 |

**ring**

| number of processes | number of iterations | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 0.484 | 0.566 | 1.133 | 2.613 | 4.924 |
| **2** | 0.484 | 0.566 | 1.207 | 2.757 | 5.242 |
| **4** | 0.484 | 0.642 | 1.428 | 3.107 | 6.127 |
| **8** | 0.484 | 0.792 | 1.579 | 3.310 | 6.427 |
| **16** | 0.492 | 0.930 | 1.934 | 3.879 | 7.519 |

These times were used to calculate the following efficiencies.

**central**

| number of processes | number of iterations | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| **2** | 1.000 | 1.000 | 0.469 | 0.474 | 0.470 |
| **4** | 1.000 | 0.220 | 0.198 | 0.210 | 0.201 |
| **8** | 1.000 | 0.089 | 0.090 | 0.099 | 0.096 |
| **16** | 0.061 | 0.038 | 0.036 | 0.042 | 0.041 |

**symmetric**

| **number of processes** | **number of iterations** | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| **2** | 1.000 | 1.000 | 0.469 | 0.474 | 0.470 |
| **4** | 1.000 | 0.220 | 0.198 | 0.210 | 0.201 |
| **8** | 1.000 | 0.087 | 0.090 | 0.099 | 0.096 |
| **16** | 0.061 | 0.038 | 0.037 | 0.042 | 0.041 |

**ring**

| **number of processes** | **number of iterations** | | | | |
|---|---|---|---|---|---|
| | **1** | **2** | **4** | **8** | **16** |
| **1** | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| **2** | 1.000 | 1.000 | 0.469 | 0.474 | 0.470 |
| **4** | 1.000 | 0.220 | 0.198 | 0.210 | 0.201 |
| **8** | 1.000 | 0.089 | 0.090 | 0.099 | 0.096 |
| **16** | 0.061 | 0.038 | 0.037 | 0.042 | 0.041 |

# Analysis

As the efficiency tables make clear, the performance quickly degrades with each increase of the number of processors. This makes sense, as every additional processor necessitates at least two more MPI messages regardless of message strategy.

We can also see that the performances of the three programs are nearly identical. Somewhat surprisingly, the ring strategy is no more effective than the centralized or symmetric strategies when the number of loop iterations is increased. This suggests that the runtimes are dominated by the randomly generated wait times, which are constrained to be between 0 and 500 milliseconds.