Ainsley Uhlenkott
CSCI 322 - Winter 2014
Homework 3
Due February 28

Compiling and Running

Included in the project are one program and one shell script.

For compilation, use:

```
make pmerge
```

To run, use:

```
./run_pmerge
```

You will be prompted for the number of processors and size of array to use.

Results

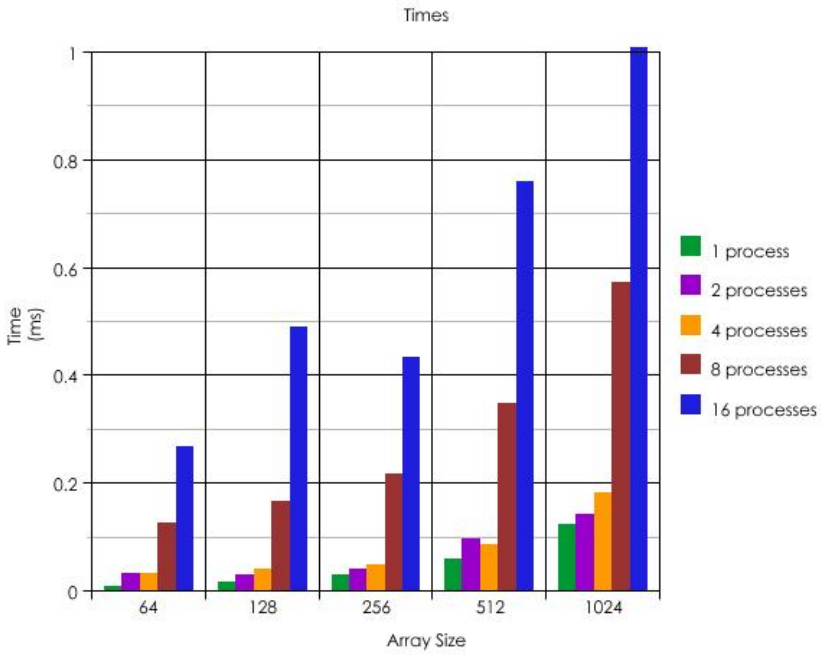
The program was run 5 times per processor/array-size configuration, and the minimum run times for each configuration were recorded. All times are in milliseconds.

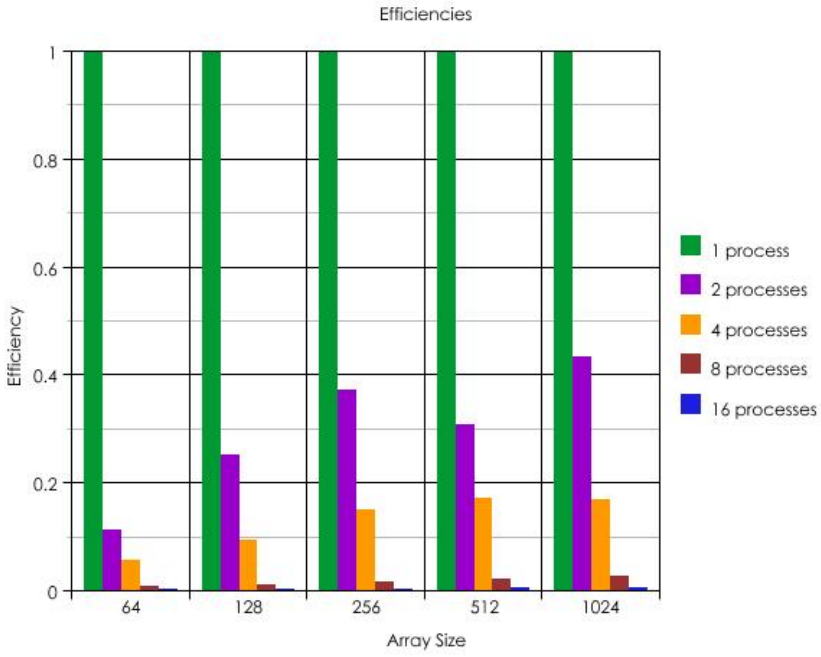
Times

number of processors	size of array				
	64	**128**	**256**	**512**	**1024**
1	0.007	0.015	0.029	0.059	0.122
2	0.031	0.030	0.039	0.096	0.141
4	0.032	0.040	0.048	0.086	0.181
8	0.126	0.165	0.217	0.348	0.571
16	0.286	0.490	0.433	0.759	1.322

Efficiencies

number of processors	size of array				
	64	**128**	**256**	**512**	**1024**
1	1.000	1.000	1.000	1.000	1.000
2	0.113	0.250	0.372	0.307	0.433
4	0.055	0.094	0.151	0.172	0.169
8	0.007	0.011	0.017	0.021	0.027
16	0.002	0.002	0.004	0.005	0.006





Analysis

Clearly this implementation of parallel merge sort is no better than a serial implementation for these numbers - in fact, it is significantly worse. We can conclude that the overhead of message passing for these processor/array configurations far outweighs any potential benefits of the parallelization.

However, the graphs show an upward trend in the efficiencies as the array size increases. Perhaps with a large enough array, this parallel implementation might show improvement on the serial implementation.