

## Homework 2

**Programs:** Submit the program solutions electronically. See the end of this assignment for information on electronic turnin. The programming exercises are worth 64 points.

### Programming Exercises:

**Due:** Friday, March 14, 8 pm.

Write two solutions to the Jacobi problem: one in MPD and the other using either C or Java threads and semaphores.

Use the same algorithm for both solutions. This will allow you to better compare the execution times of MPD vs. your other choice.

### Reading from textbook:

Sections 11.1.1 to 11.1.3, pages 534 to 541 (and the figure on page 542).

### Command-line arguments:

There are two required:

**N** = the grid will be of size **N x N**.

**numProcs** = number of worker processes (threads) to create to solve the problem. Your program should work for values of **numProcs** from 1 to 16.

The remaining command-line arguments are up to you. They should have default values so the user can choose to supply only the required two. As an example, my current solution has 4 optional arguments that are the fixed values for the left, top, right, and bottom edges of the grid, and one additional argument which is the epsilon value used to determine when to stop the computation. With my solution, typing:

```
jacobi 32 4
```

is equivalent to typing:

```
jacobi 32 4 1.0 1.0 80.0 80.0 1
```

which gives:

```
N = 32
```

```
numProcs = 4
```

```
left = 1.0, top = 1.0, right = 80.0, bottom = 80.0
```

```
epsilon = 1.0
```

### Output:

Print to a file: the settings (N, numProcs, edge values, epsilon, anything else that seems useful), and the final grid contents.

Print to standard out the number of iterations and the computation time.

### Compilation:

Submit a **Makefile** that has a target named **all**. The command **make all** should produce the executables for both of your solutions. If you are using Java, then **make all** should produce all needed **.class** files.

**Timing tests:**

Run timing tests on voltron for both programs. Run timing tests for 1, 2, 3, 4, etc. workers. Some questions that your timing results should be able to answer:

Do you see speed up as the number of workers increases?

How do the times for your MPD solution compare with your times for C or Java?

Do the times start to increase as you go beyond 4 workers?

Submit a file (using **turnin**) named *timing.txt* or *timing.pdf*. This should contain the results of your timing runs. We should be able to figure out the answers to the above questions from the timing results that you supply. Charts can be included where useful. Include in this file the command-lines that produced the timing runs reported, along with an explanation of the command-line arguments and (where appropriate) the default values for missing arguments.

**Turnin:**

Use the **turnin** program to turn in **Makefile**, **timing.txt** or **timing.pdf**, and all needed source files. Use **422assign2** as the assignment name for turnin.

See the man page for the **turnin** program for details on what turnin can do and how you can confirm that your files were turned in.