

# Homework 2 - Welcome to parallelism

[Re-submit Assignment](#)

---

**Due** Sep 19 by 11:59pm      **Points** 10      **Submitting** a file upload

---

CIS625 – Concurrent Software Systems

## Homework 2: Intro to parallel programming

**Assigned:** 9/12/18

**Due:** Wednesday, 9/19/18 by midnight

**Individual or group:** Individual

**Points:** 10

### Description

The purpose of this project is to get you started doing parallel programming on a machine like Beocat.

### Task(s)

Your first task is to take an existing program and indicate how the performance improves through using using two standard techniques – pthreads and OpenMP – to parallelize this serial code. Your second task is to do a performance evaluation across a range of iterations to see how the various versions match up in terms of run times, CPU efficiency, memory utilization, etc. You'll want to keep the machines constant, so use the flag for sbatch to confine your jobs to only the 'elf' class nodes.

Parallelize the following code using OpenMP parallel and reduction directives. Ensure that your program prints the output and tracks and prints the time expended by the program.

```
const int NUM_ITER = 100000000;
int i;
double sum = 0.0, x = 0.0;
double st = 1.0/((double) NUM_ITER);
for(i = 0; i < NUM_ITER; i++)
{
    x = (i + 0.25)*st;
    sum += 4.0/(x*x+1);
}
```

Plot the CPU time expended by the serial version of the program and by running the program with 2, 4, 8, and 16 threads for both the pthreads and OpenMP versions (feel free to combine these into one graph). Note that the x axis of your plot should be threads used, and the y-axis of your plot should be time expended in seconds. Discuss the results. Are there any race conditions?

## Resources

Pthreads - <https://computing.llnl.gov/tutorials/pthreads/> ↗ [\(https://computing.llnl.gov/tutorials/pthreads/\)](https://computing.llnl.gov/tutorials/pthreads/)

OpenMP - <https://computing.llnl.gov/tutorials/openMP/> ↗ [\(https://computing.llnl.gov/tutorials/openMP/\)](https://computing.llnl.gov/tutorials/openMP/)

For help using the Beocat scheduler and its command lines, see the general help page

[https://support.beocat.ksu.edu/BeocatDocs/index.php/Main\\_Page](https://support.beocat.ksu.edu/BeocatDocs/index.php/Main_Page) ↗  
[\(https://support.beocat.ksu.edu/BeocatDocs/index.php/Main\\_Page\)](https://support.beocat.ksu.edu/BeocatDocs/index.php/Main_Page)

Compute node specifications are at

[https://support.beocat.ksu.edu/BeocatDocs/index.php/Compute\\_Nodes](https://support.beocat.ksu.edu/BeocatDocs/index.php/Compute_Nodes) ↗  
[\(https://support.beocat.ksu.edu/BeocatDocs/index.php/Compute\\_Nodes\)](https://support.beocat.ksu.edu/BeocatDocs/index.php/Compute_Nodes)

You will probably need to install a shell program on your Windows machine – I use PuTTY, and a program to transfer files back and forth (I use WinSCP, but other options are fine). Linux and Mac OS have terminal and file transfer programs installed by default.

## Submission instructions

Submit a PDF which contains your analysis and a copy of the two versions of your code and your various controlling shell scripts as additional files. Submit this via Canvas. Grading will be split – 75% for the correctness and performance of your implementation (including use of shell scripts), 25% for your performance analysis (probably 1-2 pages, including graphs but excluding appendices, assuming a reasonably compact formatting scheme).