

## 2014.1 Multicore Computing, Project #1

(Due : April 14th 11:59pm)

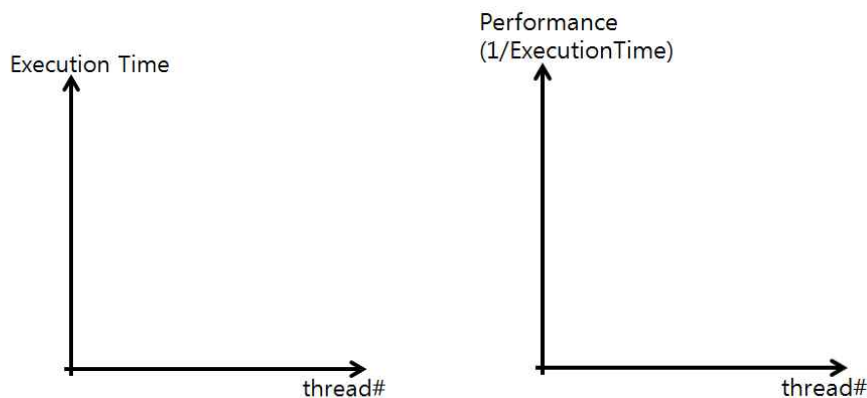
### Submission Rule

1. Create a directory "proj1"
2. Insert (i) JAVA source code, (ii) a document that reports the parallel performance of your code, and (iii) readme.txt into the directory "proj1". The document that reports the parallel performance should contain (a) in what environment (e.g. CPU type, memory size, OS type ...) the experimentation was performed, (b) tables and graphs that show the execution time (unit:millisecond) for thread number = {1,2,4,8,16}. (c) The document should also contain explanation on the results and why such results are obtained. (d) The document should also contain your entire JAVA source code. In `readme.txt` file, you should briefly explain how to compile and execute the source codes you submit. You should use JAVA language, but C language using pthread library is also allowed.
4. zip the directory "proj1" into "proj1.zip" and submit the zip file into eClass homework board.  
\* The experimentation (execution of your code and original ex4.java code) should be done **in a quad-core CPU computer**. **Quad-core CPU PCs** are available in our 4<sup>th</sup> floor laboratory and etc.

In our lab. class, we looked at the JAVA program (ex4.java) that computes the number of 'prime numbers' between 1 and 200000. The java code (ex4.java) creates threads for parallel computation using static load balancing approach. However, The parallel implementation of `ex4.java` does not give satisfactory performance because of bad load balancing. The problem is that (i) higher ranges have fewer primes and (ii) larger numbers are harder to test. Therefore thread workloads become uneven and hard to predict. For better performance, we consider dynamic load balancing approach where each thread takes a number one by one and test whether the number is a prime number.

(i) Modify `ex4.java` to adopt dynamic load balancing instead of static load balancing and submit the modified JAVA code (name it "`ex4_dynamic.java`"). Your dynamic load balancing code should use JAVA synchronization method. Your code also should print the execution time of each thread and execution time for the entire process. When writing your JAVA code, `NUM_THREAD` variable should be used as a constant value in the program (just like `ex4.java`).

(ii) Write a document that reports the parallel performance of your code. The graph that shows the execution time when using 1,2,4,8,16 threads. There should be at least two graphs, one for static load balancing and the other for dynamic load balancing. Your document also should mention which CPU (dualcore? or quadcore?, clock speed) was used for executing your code.



exec time	1	2	4	8	16
static					
dynamic					

performace (1/exec time)	1	2	4	8	16
static					
dynamic					