

2013.1 Multicore Computing, Project #1

(Due : April 16th 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 graphs that show the execution time for thread number = {1,2,3,4,5,6,7,8}. The document should also contain explanation on the result and why such result is obtained. 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 helloworld.java code) should be done **in a quad-core CPU computer** which is available in our 4th floor laboratory and etc.

The java source file **helloworld.java**, which is accessible from our class website, computes the number of prime numbers between 1 and 200000. The java code creates threads for parallel computation using static load balancing approach. However, The parallel implementation of **helloworld.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 the JAVA code helloworld.java that adopts dynamic load balancing instead of static load balancing and submit the modified JAVA code. 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.

(ii) Write a document that reports the parallel performance of your code. The graph that shows the execution time when using 1,2,3 and 4 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.

