Oblig3 IN3030/IN4330 v2025: Prime Numbers

Eric Jul, February 26th, 2025

This is mandatory assignment number 3 (Oblig3) for IN3030/IN4330, Spring 2025. The subject is calculation of prime numbers and the unique factorization of a given number into the product of only primes. You must program and run timings of a sequential and a parallel version of the Sieve of Eratosthenes and write a short report on your experience. You are to take the Java sequential version of the sieve of Eratosthenes that will be published on the IN3030 web site and produce a Java parallel version that achieves speedups greater than one. The program must also sequentially calculate the factorization of large numbers based on the primes that your program generates.

Your program must contain both the sequential and a parallel version of the sieve of Eratosthenes and of the factorization of large numbers. The program must take a parameter, N, greater than 16, and then first generate all the primes less than N, and thereafter calculate the factorization of the 100 largest numbers less than N * N. You must print the factorizations using the precode that we post on the web. Do not modify the precode. Simply place the file in the same folder as your own code, and use it as shown in the comments of the source file.

The program must also take a second parameter, k, the number of threads to be used, if zero then the program uses the number of cores on the machine. Note that your program must contain both the sequential and a parallel version of the sieve of Eratosthenes and both the sequential and a parallel version of factorizing a single number. You must test that your program (in both the sequential version and the parallel version) correctly calculates the primes less than 100 by printing the primes and manually checking them. You must run your program in the full version that finds all primes up to N and factorizes the largest 100 numbers less than N*N for the following four values of N: 2 million, 20 million, 200 million, 2 billion (note: 2 milliarder in Norwegian). For each run, you must print the execution times and speedups obtained. In the report, you must include the median times for a set of runs (at *least* 7 runs) and speedups achieved when generating primes and when factorizing the top 100 numbers. The 100 factorizations must be printed to a file using a class (our precode) that will be uploaded to the course web site. The order of the factors must be monotonically increasing. You MUST parallelize EACH factorization, i.e., use multiple threads even for a single factorization. You program should work for any number of cores.

Further requirements:

- Your program must *test* that the sequential and the parallel versions generate the SAME prime numbers.
- The report must be submitted in PDF and must include two graphs of the speedup, one for each algorithm, where the X-axis is the four values of N (perhaps log-scale), and the Y-axis is the speedup.
- The execution time for the sieve should not exceed 30 seconds – if it does, you are doing something wrong. Similarly, the time for factorization should not exceed 60 seconds.
- When printing timing values make sure to print the correct unit, *e.g.*, s, ms, ns, or whatever SI-unit you use.
- The program should print good error messages, if it is given incorrect/missing parameters.
- Use good programming style, *e.g.*, proper indentations, good variable names, *etc*.

The report must contain:

1. Introduction – *what this report is about.*
2. User guide – *how to run your program (short, but essential), include a very simple example.*
3. Parallel Sieve of Eratosthenes – *how you did the parallelization – consider including drawings.*
4. Parallel factorization of a large number – *how you did the parallelization – consider including drawings.*
5. Implementation – *a somewhat detailed description of how your Java program works & how tested.*
6. Measurements – *includes discussion, tables, graphs of speedups for the four values of N, number of cores used.*
7. Conclusion – *just a short summary of what you have achieved.*
8. Appendix – *the output of your program.*

Deadline

Deliver in devilry: Deadline is March 25th, 2025 at 23:59 CET – that is 23:59:00 local time in Oslo.