CPS 5401 Fall 2012
Shirley Moore, Instructor
Project 1 – Parallelization of Sieve of Eratosthenes
Due November 20, 2012

The sieve of Eratosthenes is a method for finding the prime numbers below a certain integer. One can do the sieve for small integers by hand. For bigger integers, it becomes necessary to use a computer. To see how the sieve works, we can follow the steps below, using the example of finding the prime numbers under 16.

1. Write out the numbers from 2 to 15.
2. Circle the smallest unmarked, uncircled number in the list.
3. For each number bigger than the biggest circled number, mark the number if it is a multiple of the biggest circled number.
4. Repeat steps 2-4 until all numbers have been circled or marked. The circled numbers will be the primes; the marked numbers will be the composites.

A serial code that implements the sieve can be found in the file sieve.c. You can compile this code on Griffin with

```
gcc –o sieve –lm sieve.c
```

and run it with

```
./sieve –n N
```

where N is the integer for which you want to find the prime numbers below it – e.g.,

```
./sieve –n 60000
```

We will use the Griffin cluster for this assignment. Your task is to optimize the performance of and to parallelize this algorithm to the best of your ability. You may use any programming language and parallel programming model you wish. If you make use of code that you find on the Web or elsewhere, please give credit by citing the source. Note that in addition to parallelizing the code, you will also want to think of possible optimizations to the algorithm and its implementation.

You should time your code and give both strong scaling and weak scaling results and compute parallel speedup and efficiency metrics (see Chapter 5 of Hager and Wellein).

You should turn in your code and a report describing your parallelization strategy and your performance and scaling results.

Two prizes will be given – one for the fastest program for a problem size of N=96,000,000, and one for the largest problem that can be solved in 10 minutes of wallclock time.