Your Special Challenge

Develop a parallel application that does find prime numbers by using the sieve of Eratosthenes with C and MPI

A prime number is a natural number that has exactly two distinct natural number divisors: 1 and itself. The sieve of Eratosthenes is a simple algorithm for finding all prime numbers up to a given limit.

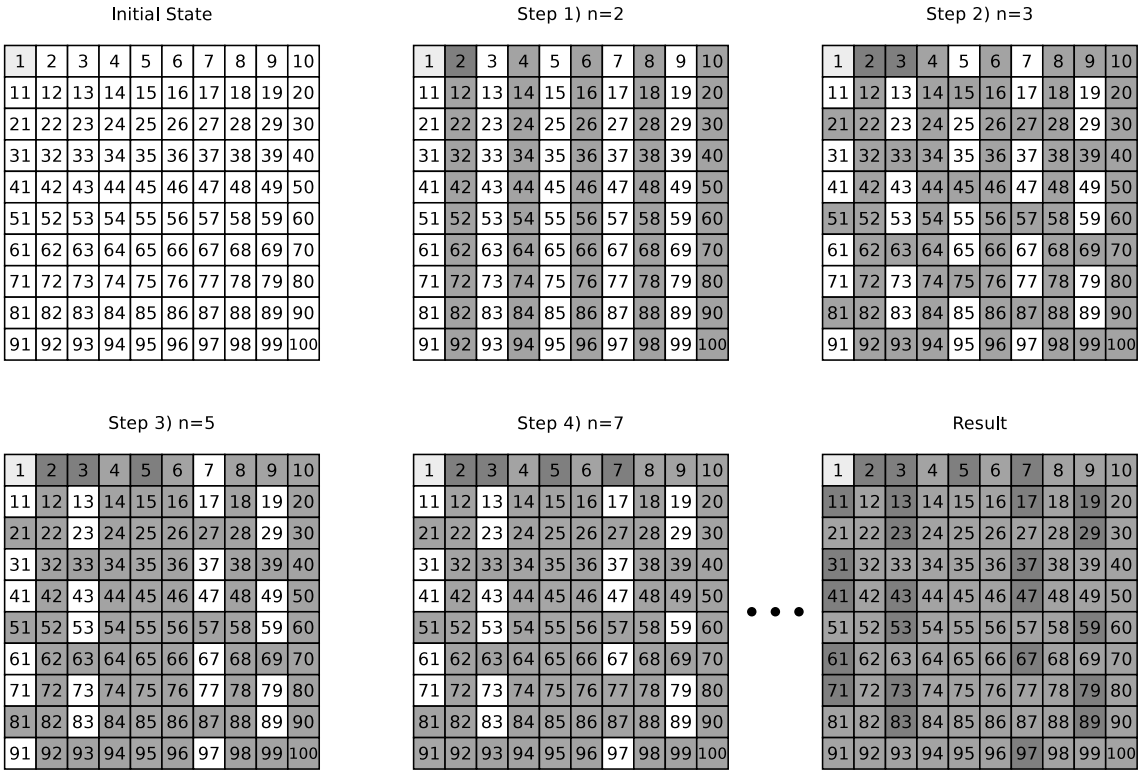

Figure 1: The sieve of Eratosthenes (steps for primes below 101)

Wikipedia explains [1] the functioning of the algorithm this way: *To find all the prime numbers less than or equal to a given integer n by Eratosthenes' method:*

- *Create a list of consecutive integers from 2 through n: (2, 3, 4,. . . , n).*
- *Initially, let p equal 2, the smallest prime number.*
- *Enumerate the multiples of p by counting to n from 2p in increments of p, and mark them in the list (these will be 2p, 3p, 4p,. . . ; the p itself should not be marked).*

- *Find the first number greater than p in the list that is not marked. If there was no such number, stop. Otherwise, let p now equal this new number (which is the next prime), and repeat from step 3.*
- *When the algorithm terminates, the numbers remaining not marked in the list are all the primes below n.*

The individual steps of your task

1. **Investigate** how to implement with the programming language C and with MPI an application, that finds prime numbers by using the sieve of Eratosthenes algorithm and benefits from the resources of a parallel computer [2]. Literature provides many helpful sources, covering how to implement the algorithm in C with MPI [3][4][5][6][7][8][9].
2. **Deploy** a MPI environment for testing and development on physical hardware, inside virtual machines [10][11] or inside a public cloud infrastructure service like EC2 [12][13][14].
3. **Develop** your application with C and MPI.
4. **Test** your application on your own MPI cluster and test it on our 128 node cluster with the 512 CPU cores. Test your application...
   - with sufficient large problem sizes and
   - with different numbers of cores (1, 2, 4, 8, 16... 512) and

   calculate the speedup. Your outcomes may prove some of the laws and limitations we already discussed during class in slide set 1.
5. **Create** a presentation (max. 30 Minutes) with maximum 15 slides and additionally a live demonstration. Give your presentation during class or during an exercise session.

Some final words to motivate you

In real life, one or two weeks is a typical amount of time to do such a task in parallel to your daily tasks. In this course you have several weeks and you are not alone.

Last but not least, questions of the exam will cover this special task!

References

[1] "Sieve of Eratosthenes." https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes.

[2] T. Bräunl, *Parallele Programmierung: Eine Einführung*. Vieweg, 1993.

[3] "MPI Sieve of Eratosthenes." https://github.com/marius92mc/sieve-of-eratosthenes-with-MPI.

[4] M. Cordeiro, "Parallelization of the Sieve of Eratosthenes," tech. rep. https://mmfcordeiro.files.wordpress.com/2012/10/mmfcordeiro-parallelization-of-the-sieve-of-eratosthenes.pdf.

[5] "Count Primes Using MPI." http://people.sc.fsu.edu/~jburkardt/c_src/prime_mpi/prime_mpi.html.

[6] "Prime Number Counter (Sieve of Eratosthenes) Using MPI." https://stackoverflow.com/questions/23589313/prime-number-counter-sieve-of-eratosthenes-using-mpi-too-slow.

[7] "Mpi and the Sieve of Eratosthenes." http://acc6.its.brooklyn.cuny.edu/~cisc7340/examples/mpisieves16.pdf.

[8] J. Frisch, "Parallel Programming 1: MPI," tech. rep. https://www5.in.tum.de/forschung/simlab/course2010_files/course_mat/Frisch_MPI_ParProg.pdf.

[9] X. Cai, "The Sieve of Eratosthenes," tech. rep. http://www.uio.no/studier/emner/matnat/ifi/INF3380/v12/undervisningsmateriale/inf3380-sieve.pdf.

[10] "MPI Tutorials 01." https://www.youtube.com/watch?v=2rpWEZY0aPo.

[11] "Create test MPI cluster using VirtualBox images." http://www.zamaudio.com/?p=837.

[12] "Launching an Amazon EC2 MPI Cluster." http://mpitutorial.com/tutorials/launching-an-amazon-ec2-mpi-cluster/.

[13] "Using MPICH in Amazon EC2." https://wiki.mpich.org/mpich/index.php/Using_MPICH_in_Amazon_EC2.

[14] "MPI on EC2." https://www.youtube.com/watch?v=049CVTGHO1k.