

Parallelization: Sieve of Eratosthenes

By Aaron Weeden, Shodor Education Foundation, Inc.

Rubric for Student Assessment

This module can be graded for 100 total points, which allows for easy conversion to a different points-based grade or a percent-based grade depending on the instructor's preference.

Quick Review Questions

The Quick Review Questions can be graded for 24 total points; each question can be graded for 1 point, except for questions 9, 10, 13, and 17, which can be graded for 2 points each, and question 14, which can be graded for 4 points. It is recommended that students provide hand-written, not typed, responses. Examples of acceptable answers for these questions are provided below.

1. 1, 2, 3, 4, 6, and 12
2. 2, 3, 5, and 7
3. `./sieve.serial -n 5000`
4. The program would print a usage message and exit with an error.
5. 0
6. Entities that share memory are known as **threads**. Entities that have distributed memory are known as **processes**.
7. **Message passing** is a method of communicating data between processes. It is needed in a distributed memory situation in which one or more processes must access the memory of one or more other processes.
8. Parallelism is impossible on a von Neumann computer because there is only one CPU to run a series of instructions. Parallelism requires at least two CPUs.
9. **Cores** are the processors created by splitting a CPU. They often share RAM (1 point).
Compute nodes are computers connected on a network; they do not share RAM (1 point).
10. **Amdahl's Law** says that the speedup of a parallel program will be limited by the time it takes to execute its serial regions. As the number of processors devoted to the problem increases, the advantages of parallelism diminish as the serial regions become the only parts of the code that take significant time to execute (1 point).
Gustafson's Law says that bigger problems can be modeled in the same amount of time as smaller problems if the processor count is increased (1 point).
11. In **strong scaling**, the problem size stays constant as the number of

processors increases. In **weak scaling**, the problem size varies as the number of processors increases.

12. `list1` contains the numbers **2** through `sqrtN`, so the biggest number will be `sqrtN`, which in this case is $\sqrt{16}$, or 4.
13. $S = (N - (\text{sqrt}N + 1)) / p = (16 - (\sqrt{16} + 1)) / 4 = (16 - (4 + 1)) / 4 = (16 - 5) / 4 = 11/4$, which truncates to **2**
(1 point)
This means **R** is $11 \% 4 = 3$.
(1 point)
14. Rank 0's **L** = $\text{sqrt}N + r * S + 1 = \sqrt{16} + 0 * 2 + 1 = 4 + 0 + 1 = 5$
Rank 0's **H** = $L + S - 1 = 5 + 2 - 1 = 6$
(1 point)
Rank 1's **L** = $\text{sqrt}N + r * S + 1 = \sqrt{16} + 1 * 2 + 1 = 4 + 2 + 1 = 7$
Rank 1's **H** = $L + S - 1 = 7 + 2 - 1 = 8$
(1 point)
Rank 2's **L** = $\text{sqrt}N + r * S + 1 = \sqrt{16} + 2 * 2 + 1 = 4 + 4 + 1 = 9$
Rank 2's **H** = $L + S - 1 = 9 + 2 - 1 = 10$
(1 point)
Rank 3's **L** = $\text{sqrt}N + r * S + 1 = \sqrt{16} + 3 * 2 + 1 = 4 + 6 + 1 = 11$
Since Rank 3 is the last process, Rank 3's **H** = $L + S - 1 + R = 11 + 2 - 1 + 3 = 15$
(1 point)
15. 3 is less than $\sqrt{16}$, so it will be stored in `list1` at position **3**.
16. 8 is greater than $\sqrt{16}$, so it will be stored in `list2` at position $8 - L = 8 - (\text{sqrt}N + 1) = 8 - (4 + 1) = 3$.
17. `MPI_Recv(list, 3, MPI_FLOAT, 4, 5, MPI_OUR_COMM, MPI_STATUS_IGNORE);`
(2 points if all arguments are correct and in the right order, 1 point if most arguments are correct and in the right order, 0 points if most arguments are incorrect or in the wrong order)

Exercise 1

Exercise 1 can be graded for 18 total points, as follows: 6 points for handing in `hello.c` or `hello.F90`, 6 points for handing in `hello.qsub`, 6 points for handing in `hello.out`.

Exercise 2

Exercise 2 can be graded for 20 points, as follows: 10 points for handing in Table 1, 10 points for handing in Table 2.

Projects

It is encouraged that each student completes one project, either alone or in groups. The project can be graded for 38 points based on accuracy and presentation of the solution.