CSE-745 2017 Course Project

Your task is to write an MPI version of the serial code ~syam/ces745/Project/primes_count.c. This is a code to count the number of prime numbers within a given interval. All prime numbers can be expressed as 6*k-1 or 6*k+1, k being an integer. The code defines the range of k via the macro parameters KMIN and KMAX. You can choose one of the three levels of difficulty, with Level 1 being the easiest (and giving you the lowest maximum grade, 80%), Level 2 being of an intermediate difficulty (maximum grade 90%), and Level 3 being the most difficult (maximum grade 100%). (Any bugs/mistakes/lack of commenting etc. will further reduce the grade.) Here is a more detailed description of the three leveles:

- Level 1. Maximum grade 80%. Use static workload balancing, with all ranks participating in prime number search (including rank 0).

- Level 2. Maximum grade 90%. Use dynamic work load balancing scheme, with the master (rank 0) not participating in prime number search.

- Level 3. Maximum grade 100%. Use dynamic work load balancing scheme with the master (rank 0) taking part in prime number search.

Please indicate the Level chosen in the header of your solution. Check the parallel code correctness - it should produce the same result as the serial version, for the same range KMIN...KMAX. Try to make the parallel code as efficient as possible. More detailed instructions are given in the header of the file primes_count.c. You are expected to achieve a speedup close to the number of cpu cores (threads) used. For example, on orca development nodes (orc-dev1, ..., orc-dev3, with 24 cores each), the speedup should be close to 24. **You should put comments inside your code explaining what you are doing.**

Marks will be taken off for code bugs (some or all the results are wrong), for insufficient commenting in the code, and for poor performance. It must be your own work and you are responsible for adhering to the Senate Policy Statement on Academic Ethics.