

ECE 493/693  
Fall 2004  
Assignment 4

Task: Write a parallel matrix/vector multiply

Due: Monday, November 15, 2004

Details:

Write a C/MPI program that implements one of the following Monte Carlo simulations: neutron transport, plate temperature, icing model, room assignment, parking garage, traffic circle, or another Monte Carlo problem of your choosing, approved by the instructor. You should provide appropriate command line arguments, based on the problem (for example, the parking garage problem could have the number of cars to simulate, the number of parking spots, and the average inter-arrival time as arguments). Incorrect arguments should result in an appropriate usage statement. Simulations requiring IO (such as the plate temperature (boundary temps), icing model (initial configuration), room assignment (like/dislike matrix)) may use any IO mechanism you choose, including MPI-IO, master process, etc.

This project must use a parallel formulation, which means the final results must be appropriately reduced before the results are reported. This also means a suitable parallel random number generator is needed. Rather than use one of the RNG in the library, implement the Lagged Fibonacci RNG. Next choose a scheme for generating parallel random streams. Details of these choices and designs must be well documented.

Produce final formatted output to the screen with the results of the simulation and the runtime of the simulation. Do not include IO, argument processing, or output formatting in your runtime.

You must use MPI collective communication, communicators, groups, and topologies if, when, and where appropriate. You should use `getopt` (man 3 `getopt`) to parse command line arguments.

Turn in:

- Printout of program code, commented so the grader can easily follow your design logic. Pay special attention to the RNG.
- A one page code summary that lists: identification (name, etc.), code purpose, degree of completeness of the code, summarized results of testing, any other relevant comments. Include a section on the parallel RNG design.
- Timing of executing the program on between 1 and 8 nodes on class cluster suitably graphed as a speedup chart (processors versus speedup) for presentation. On a separate chart graph efficiency.