CSEP 524: Assignment #6

(due prior to class, Tuesday February 26th)

1) **Reading:**
   a) *Technology-Driven, Highly-Scalable Dragonfly Topology,* Kim et al., 2008
   b) Lin & Snyder, Ch. 7 thru Partitioned Global Address Space Languages (pp. 202-229) OR an "Introduction to MPI" Tutorial of your choosing

   Submit 1 question per reading ***in a textfile format*** for consideration in class discussions by Monday evening, 9pm, Feb 18th .

2) **Distributed Memory Reduction:** Starting from the file manual-reduce-mpi.c, rewrite your shared memory reduction algorithm from HW4, Q5c as a distributed memory computation in C+MPI. Essentially, use sends/recvs to coordinate between the tasks (processes) rather than synchronization variables. Note that although MPI supports reduction routines directly, the goal of this exercise is to gain experience with the point-to-point MPI routines; so don't use collectives.

   [Goal: get some initial practice with MPI in a familiar computational setting; note the relationship between synchronization and communication]

3) **Distributed 9-Point Stencil:** Rewrite the 9-point stencil computation from HW5 in C+MPI by following the steps in stencil9-mpi.c. Your implementation should use a 2D block x block distribution (using a version of your block distribution from HW1) and should use the same initial conditions, boundary conditions, and weights as in HW5. Your solution should use ISend/IRecv pairs for the 9-point stencil communication; to coordinate I/O to the console, use Send/Recv pairs. Verify that the results are correct with respect to your OpenMP version. Submit your code and example runs on varying processor grid configurations and problem sizes.

   Optional exercise, for fun: Run your solution on a true distributed memory machine (like the VM cluster provided by UW) to determine how close you can come to linear speedup relative to your OpenMP version (compiled in either sequential or parallel mode).

   Optional exercise: Using the "parallel text file I/O" strategy discussed in lecture, convert the serial console I/O to parallel text-based file I/O.

   [*Goal:* gain more experience with realistic idioms in MPI including ghost cells, communication, reductions, coordinating I/O]