Extra exam questions

Your answers should be to the point: address the questions and omit information that is not asked for.

1. What would change if Floyd’s All pairs Shortest Paths (ASP) algorithm would be parallelized by distributing the rows cyclically instead of blockwise over the different machines? Would the communication pattern and/or performance change?

2. The Traveling Salesman Problem (TSP) can be parallelized using Replicated Workers style parallelism: a master processor searches the TSP tree up to a certain depth (called MAXHOPS) and then dynamically distributes the remaining subtrees to a number of worker processors. The value of MAXHOPS affects the communication overhead, the load imbalance overhead, and the expected search overhead of the parallel TSP program.

   (a) Explain in general what these three types of overhead mean.
   (b) Explain for each type of overhead whether it increases or decreases with larger values of MAXHOPS. Justify your answers. You should assume that the TSP tree is sorted using the nearest-city-first heuristic.

3. Some parallel machines contain multiple networks with different topologies. For example, the Blue Gene has a network with a 3D mesh topology and another network with a tree topology.

   (a) Explain why it is useful to have different networks in one machine.
   (b) Suppose you were to implement the MPI library on the IBM Blue Gene machine. How could you use the different networks of this machine to efficiently implement the different communication mechanisms in MPI?

4. A problem with asynchronous message passing is that the buffer space for storing outgoing messages is finite, so the sender still may have to
be blocked if the buffer space fills up. This is confusing to programmers, who assume that asynchronous sends continue immediately and don’t block. How does MPI deal with this problem?

5. Fortran has operators on entire arrays (e.g. addition of two arrays) built into the language. Explain how HPF (High Performance Fortran) exploits this feature to ease parallel programming.

6. Both branch-and-bound (as used, for example, for the Traveling Salesman Problem) and Barnes-Hut (used for N-body problems) use techniques to cut-off (prune) part of the computations. Parallel branch-and-bound algorithms can sometimes obtain superlinear speedups, because the parallel version may (for certain input problems) perform less work than a sequential algorithm. Can the parallel Barnes-Hut algorithm also obtain superlinear speedups in this way? Explain your answer.