Uppsala University
Department of Information Technology
Scientific Computing

Programming of Parallel Computers, 2008-06-02

Time: $14^{00} - 19^{00}$
Help: None
Each of the six problems below can give up to five points. For maximum points, you must give detailed answers and motivate your assumptions.

1. Why don't we get perfect speedup when running a parallel program with OpenMP on a distributed shared memory computer? Give at least five reasons for this (the parallel overhead) and explain each of them.

2. For each of the parallel overheads above (problem 1) describe or discuss techniques for how the overheads can be reduced.

3. Let $V = \{v_1, v_2, \ldots, v_m\}$ be a set of linearly independent vectors but not orthogonal. An orthogonal set of vectors $Q = \{q_1, q_2, \ldots, q_m\}$ can then be constructed by using the modified Gram-Schmidt algorithm:

   program Modified-Gram-Schmidt
   declare
   real $v(1:n, 1:m)$, $q(1:n, 1:m)$
   begin
   $q(:,1) = v(:,1)/\|v(:,1)\|_2$;

   for $i = 2$ to m do
   for $j = i$ to m do
   $\sigma = q(:, i-1)^T v(:, j)$
   $v(:, j) = v(:, j) - \sigma q(:, i-1)$;
   $q(:, i) = v(:, i)/\|v(:, i)\|_2$;
   end

   Note: We are using Matlab notation, i.e., colon notation means for all elements.
   '$q(:,1) = \ldots$' equals to 'for k=1 to n, $q(k,1) = \ldots$'
   The dot-product is defined as $q(:,i)^T v(:,j) = \sum_{k=1}^{n} q(k,i) \cdot v(k,j)$
   and the norm $\|v(:,i)\|_2 = \sqrt{v(:,i)^T v(:,i)}$.

   Analyze the loop dependencies and parallelize the algorithm with OpenMP directives. Discuss what factors will limit the parallel performance of your parallelization.

4. In the course we have been discussing different metrics used for evaluating parallel algorithms and programs. Describe and explain these different metrics.

5. What is *collective communication* in MPI and what is characteristic for a collective communication call? Give at least five examples of collective communication operations and explain their effect.

6. Assume that we have a set of heterogeneous multi-core nodes with different number of cores within different nodes but all cores over all nodes are identical. The nodes are connected with some kind of interconnect. This gives us a heterogenous distributed (local address space) memory parallel computer. We can use MPI to communicate between the nodes and OpenMP to parallelize over the cores within a node. Assume that we want to do parallel matrix-matrix multiplication on this parallel machine when we have three nodes with 6 cores, 2 cores, and 4 cores, respectively. The matrix sizes are 1200x1200. Construct an efficient parallel algorithm for this problem using all cores and nodes.

Good Luck!