Problem set

Cpt S 483: Introduction to Parallel Computing

School of EECS, Washington State University

Fall 2015

Updated: Thursday 8th October, 2015

1 Active problems

2. *(5 points) - due: Thursday, October 15, 2015*

   Matrix transpose: As you may recall, the *transpose* of a matrix $A[m, n]$ is the matrix $A^T[n, m]$ s.t., $A^T[i][j] = A[j][i]$, for any $0 \leq i \leq m-1$, $0 \leq j \leq n-1$.

   Let us say I have a very large matrix A for which I need to perform a transpose (in parallel). For this I need to write a transpose function in MPI.

   Before the transpose operation, the matrix $A[m, n]$ is stored in a distributed manner evenly among p processors, s.t., each processor initially holds a distinct submatrix of size $\frac{m}{\sqrt{p}} \times \frac{n}{\sqrt{p}}$. (Assume that p is a perfect square.) In other words, the matrix $m \times n$ is kept distributed in a mesh of $\sqrt{p} \times \sqrt{p}$ processors, such that processor rank r holds the submatrix defined by $A[k'\frac{m}{\sqrt{p}} \ldots (k'+1)\frac{m}{\sqrt{p}}, k\frac{n}{\sqrt{p}} \ldots (k+1)\frac{n}{\sqrt{p}}]$, where $k' = \lfloor \frac{r}{\sqrt{p}} \rfloor$ and $k = r \ modulo \ \sqrt{p}$.

   After the transpose operation, the matrix $A^T[n, m]$ should be made available in the same distributed manner as matrix A.

   It should be easy to see that all that's needed to do the transpose is for specific pairs of processors to exchange their respective submatrices. The question for you in this homework problem is to design a function *int GetMyMate(int r)*, which takes as input a given processor's rank r and returns the rank of the processor that it should exchange its submatrix with, to complete the transpose. You can assume that your function knows the total number of processors in the system (p).

   Provide the function in the form of a C code. You don't need to implement it or test it but it should work if I were to plug in your function's code to the rest of my transpose code.

2 Retired problems

3. *(10 points) - due: Tuesday, September 8, 2015*

   Consider the problem of computing the sum of n numbers on p processors, where $n >> p$. For the parallel algorithm we discussed in class, the parallel runtime, denoted by $T(n, p)$, is given by:

   $$T(n, p) \ = \ \mathcal{O}(\frac{n}{p} + \lg p)$$

   Also, let ω denote the sequential work performed by the *best* serial algorithm for this problem. For this problem:

   $$\omega \ = \ n$$

   Given the above, answer the following questions:

   i) Provide the expression for the parallel algorithm's speedup (S) on p processors.
   ii) Provide the expression for the parallel algorithm's efficiency (E) on p processors.
   iii) Provide the expression for the parallel algorithm's overhead ($T_o(n, p)$) on p processors.
   iv) Provide the *Isoefficiency function* for the parallel algorithm.
   v) If we double the number of processors (i.e., use $2 \times p$ instead of p processors), then using the isoefficiency function provided above, determine by how much factor should we increase ω (and hence, n) in order to maintain efficiency.