Introduction to Parallel Computing
Exercise 2 (Deadline: June 21)

Wolfgang Schreiner
RISC-Linz

May 7, 1999

Write on the Convex SPP a parallel (Fortran or C) program solving the "All Pairs Shortest Path" problem.Take the sequential program on the course page as the starting point and use the MPI library to develop the parallel program in the message passing model.

1. Compile the sequential program with `-O2`. Measure the execution times of function `path` for matrix sizes 256 *and* 384 and take these times as the base times of all your comparisons with the parallel program. Make sure that you apply the same optimizations to the sequential code that you apply to the parallel code for a fair comparison.

2. Write a MPI program that implements a row/column-oriented solution organizing all processors in a ring (as for the matrix multiplication algorithm) discussed in class. After each "communication round", the result has to be *gathered* and *scattered* for the next round.

   For developing the MPI program

   - call `mpa -sc parallel` to move to subcomplex "parallel",
   - include `/usr/convex/mpich/bin` in your PATH,
   - include `/usr/convex/mpich/man` in your MANPATH,
   - compile the program with `mpicc` or `mpif77` (`-O2`),
   - call `mpirun -np` *n program* to run *program* with *n* processes.
   - use `MPI_Wtime` to measure wall clock times,

   Manual pages are available for `MPI`, `mpicc`, `mpif77`, `mpirun`, `mpiclean`, and for all the MPI library calls. Various environment variables can be used to customize the execution (see the man pages).

3. Compile the program with `-O2` and and measure the execution times for both input sizes and 1, 2, 4, 8, 12 processors.

4. Determine the asymptotic *isoefficiency function* for the algorithm using $T(s) = s+1$ as the time required for sending a message of s words (assume that a scatter to n processors takes the same time as sending n partial messages). By which factor should the dimension of the matrix grow to maintain constant efficiency if we increase the processor number from 1 to 2, 4, 8, 12?

   Test this claim by running benchmarks with 1, 2, 4, 8, 12 processors increasing your matrix size (from 256 respectively 384) correspondingly.

5. Write a report documenting your results including

   - the source code,
   - execution times (numerical tables and graphical diagrams),
   - speedups (numerical tables and graphical diagrams),
   - efficiencies (numerical tables and graphical diagrams),
   - the isoefficiency analysis and the corresponding benchmarks,
   - other findings, problems, comments, etc.