Homework 5

Parallel programming with OpenSHMEM

Deadline: March 22 2017
Laplace's equation - OpenSHMEM

\[ U_{i,j}^{n+1} = \frac{1}{4} \left( U_{i-1,j}^n + U_{i+1,j}^n + U_{i,j-1}^n + U_{i,j+1}^n \right) \]
• Assuming you have a 2 dimensional matrix distributed in row-major format (also called 2d block-cyclic format familiar to ScaLAPACK users), compute the 1000 iterations of the computation of the Laplace equation using multiple OpenSHMEM processes.
  – Special attention should be payed to minimize the extra memory requirements
  – Hint: the algorithm is highly parallelizable (it should be visible from your performance graphs). This requires fine grained synchronizations between the different processes.

• Originally the matrix is initialized with 0 everywhere except the boundaries (first and last row and first and last column) which are randomly initialized.

• Highlight the performance of your implementation by providing weak (fixed size per process) and strong scaling (fixed problem size independent of the number of processors) results.

• Benchmarking of the Laplace algorithm should be measured excluding the data and OpenSHMEM initialization.

• This homework is obviously extremely similar to the previous one. The reason is that you should compare the 2 programming approaches in terms of performance [objectively] and also programmability [subjectively] (justify).