Report on the Parallelization and Offloading of a 7-point Stencil and Implicit Moving Squares

Introduction to High Performance Computing CS 1645

Brian Lester bdl20

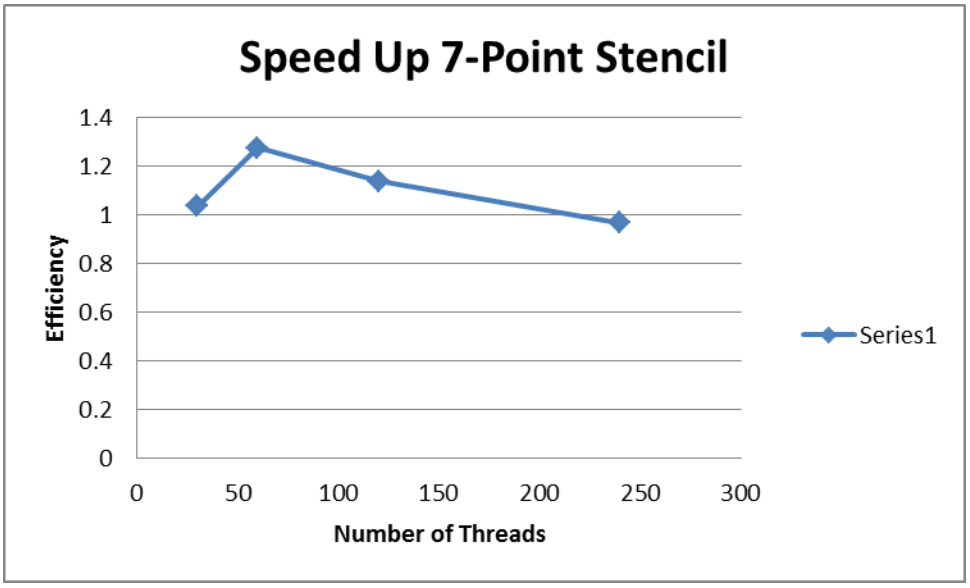
1) Parallelization strategies for each problem.

   7-point Stencil. Obviously a stencil approached is used. The outer-most loop cannot be parallelized because each iteration of the grid is based on the previous iterations grid but the other sets of loops (the first one which computes a value for the new grid from the old one and the second set which copies from one grid to the other) are both parallelizable because there is no data dependencies in the loops.

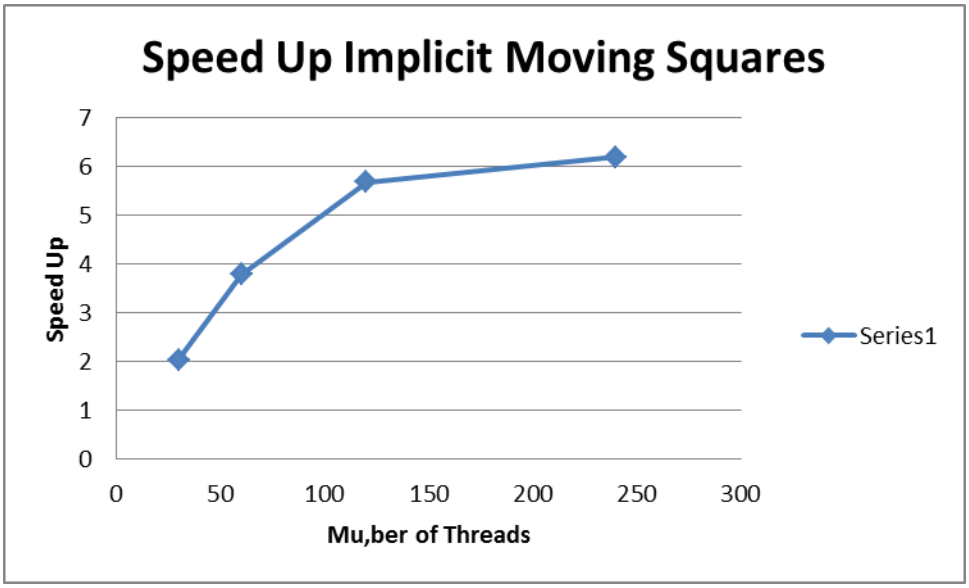
   Implicit Moving Squares. This is parallelized by scatter to gather. Each loop can be parallelized and the whole thing can be offloaded to the accelerator.

2) Speed Up Graphs:

   7-point Stencil: Parallel Time: 17.303

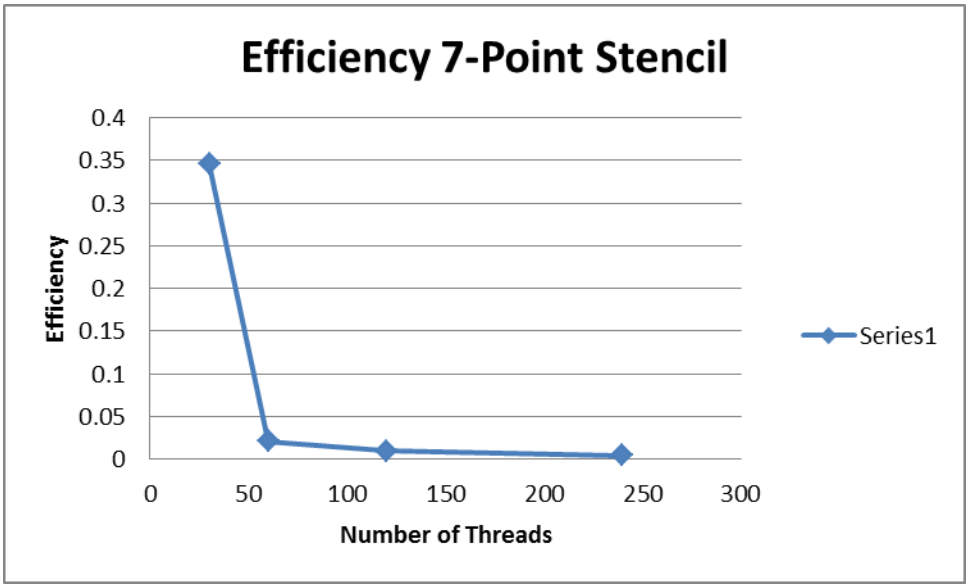


   Implicit Moving Squares: Parallel Time: 5.44708 Seconds

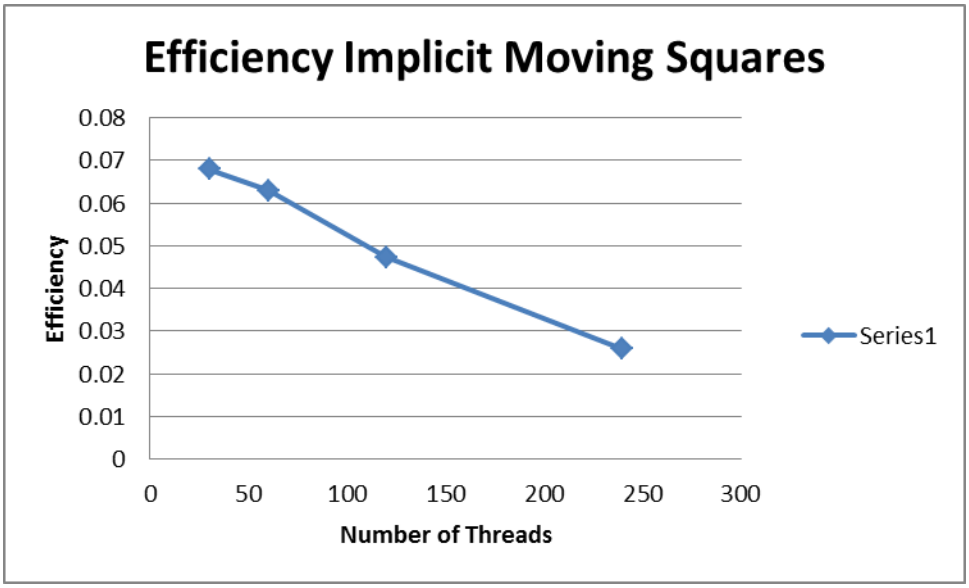


3) Efficiency Graphs:

7-point Stencil



Implicit Moving Squares



4) The bottleneck in the 7 point stencil is the fact that you cannot parallelize the outer most loop. This is because the graph relies on the previous iterations graph so it is in the nature of the problem rather than the algorithm. There is also a large data transmission cost because the grid has to be copied to the accelerator and copied back out.

   The bottleneck in implicit squares is the data transmission, things must be copied and forth which stops the speed up.