

Porting Lattice QCD Programs from High Performance Fortran to MPI

Peter Moran, 2006/07

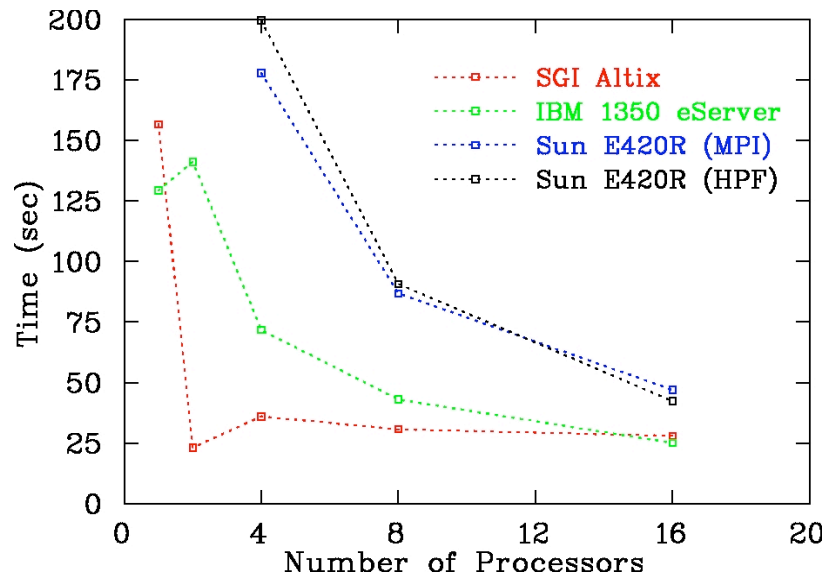
For my project I worked on porting some legacy HPF code to MPI using Waseem Kamleh's library. The port was successful, and we now have a working and tested Highly-Improved Cooling and Over-Improved Smearing program written in MPI. Also, as part of my internship I presented a series of lectures, 3 in total, on the basic use of Waseem Kamleh's library. I am now in a position to help other researchers to make or modify MPI programs for their own lattice studies.

The program that I worked with takes a Quantum Chromodynamics (QCD) gauge field and applies either a Highly Improved Cooling or an Over-Improved Smearing algorithm to links between lattice sites. This involves finding products of closed loops with different sizes, either 1x1, 1x2, 2x2, 1x3, or 3x3 loops. These loops are added together and depending on whether we want to cool or smear the links, a different algorithm is applied and the original links are updated. When smearing the gauge field we are allowed to update all links simultaneously. However, when cooling we can not update all links simultaneously and therefore have to use a masking technique to ensure that neighbouring links do not affect each other. The masks are implemented as arrays of logical variables that tell each processor when it is safe to update a certain link. After the links have been smoothed by either cooling or smearing the program analyses the resulting gauge field.

The legacy code needed to be ported to MPI because the HPF version would only run on the Sun E420R supercomputer (Orion), which is currently being phased out. We also hoped to see some speedup when switching to MPI. It was also important that this code be ported to MPI in order to provide both an example of writing MPI programs for other researchers, and also to provide them with a working cooling and smearing program.

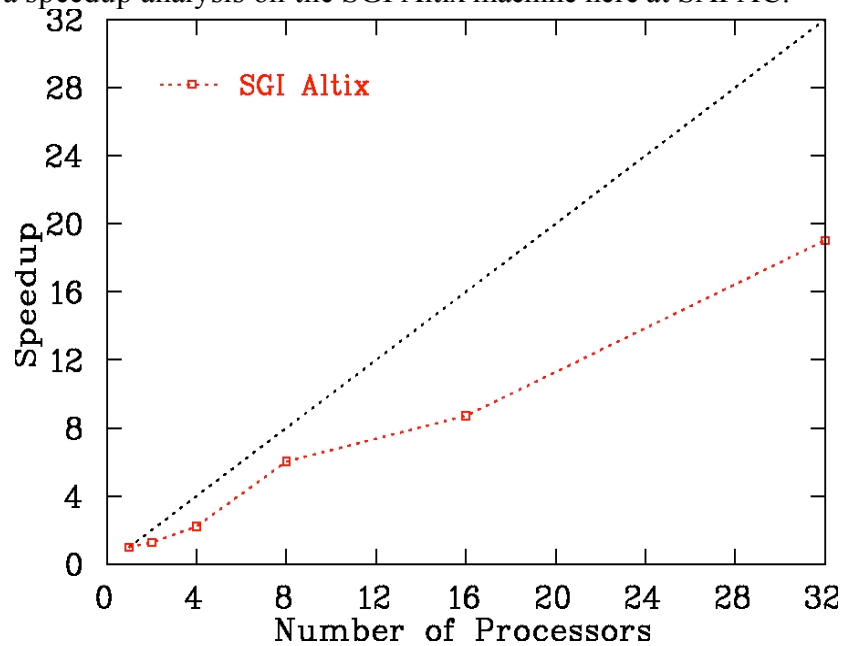
In regards to the actual coding, I was forced to rewrite the majority of the program. The gauge fields are now stored as 3x3 complex matrices, whereas previously, they were stored as two separate arrays, one for the real part and one for the imaginary. This meant that virtually none of the existing code could be easily used in the new program. Since the program now uses MPI, I also had to add in new lines of code to perform operations such as; broadcasting variables, shadowing the gauge field across the processors, and performing reduction operations. I also had to modify parts of Waseem Kamleh's library, because the program requires the calculation of 3x3 loops of links. 3x3 links requires triple gauge field shadowing, and previously the library was only able to handle double shadowing.

I did some preliminary benchmarking of the code on the SUN E420R, IBM 1350 eServer, and SGI Altix supercomputers here at SAPAC. A comparison of the running time for a small $16^3 * 32$ lattice is shown below:



I was unable to run the program using just 1 or 2 processors on the Sun E420R supercomputer. I suspect this was because of memory restraints. The program naturally ran much faster on the newer supercomputers. The strange behaviour at large numbers of processors is due to the small lattice size. There was not enough time available to re-run the benchmarks with a larger lattice size.

I also performed a speedup analysis on the SGI Altix machine here at SAPAC.



This data was obtained using a very large $28^3 * 96$ lattice. In this case the speedup is less than linear, however, when scaling to larger numbers of processors this performance should improve as there will be an increased use of cache memory.

In conclusion, the project was a success, with the legacy HPF code being successfully ported to MPI. The code has not yet been optimized, however, and it may be possible to obtain some speedup by tweaking the program. A proper speedup analysis should also be performed at the APAC Facility, with hundreds of processors, where we expect to see super-linear speedup.