526E$_1$

Efficient algorithm development for problems in condensed matter physics

K P M K Silva[1] D N Ranasinghe[1*] G Fernando[2] and W R M U K Wickramasinghe[2]

[1]*University of Colombo School of Computing, University of Colombo, Colombo 03, Sri Lanka.*

[2]*Department of Physics, University of Connecticut, USA*

The study of advanced materials and especially transition-metal compounds for their quantum and statistical mechanical properties have been shown to be highly computationally dependent. Quantum Monte Carlo methods are able to deal with strongly correlated electronic systems and have in fact provided solutions to important problems in many body physics. It has been observed by timing measurements of the serial QMC code that there are several instances of code that consumes substantial amount of time which directly correspond to matrix multiplications. With larger dimensions the computational overhead becomes very heavy especially on single processor machines running a sequential algorithm.

The original algorithm has been made efficient in two ways: parallelization of matrix multiplications and code optimization. Parallel algorithms have been developed to run on two distinct paradigms: one which uses Open MP on a shared memory multiprocessor and one that uses MPI on a cluster. In handling large data blocks as in the case of matrices, cache optimization attempts to optimize the placing of the blocks of data in cache so that the overhead in transferring blocks between main memory and the cache is reduced. This would lead to a substantial saving in running time.

Timing results are obtained on a cluster of machines consisting of four nos Dual Intel Pentium II processors (200 MHz) with 256 MB RAM and 1 MB cache running MPI-Fortran and on a Sun Solaris shared memory 64 processor system running Open MP. In case of the sequential version, a clear timing improvement can be seen with cache optimization especially for large dimensional matrices of the order of 1024. Typical block algorithms have been used for the MPI version with and without cache optimization and in the former a further timing improvement is visible. Similar improvements of timing results are visible in the case of the shared memory implementation. As emphasis is placed on low cost scalable solutions to the large dimensional QMC problem, distributed memory versions are of special interest. It is hoped that the measurements will also reinforce the view that parallelization combined with code optimization offers a real solution to the computationally intensive QMC simulation problem and other similar computationally intensive calculations.

*dnr@ucsc.cmb.ac.lk Tel: 011 2589123