Accelerating the B-spline evaluation in Quantum Monte Carlo

Ye Luo
Argonne National Laboratory
9700 S. Cass Avenue
Lemont, IL 60439
yeluo@anl.gov

Anouar Benali
Argonne National Laboratory
9700 S. Cass Avenue
Lemont, IL 60439
benali@anl.gov

Vitali Morozov
Argonne National Laboratory
9700 S. Cass Avenue
Lemont, IL 60439
morozov@anl.gov

ABSTRACT

In Quantum Monte Carlo simulations, the many-body Schrödinger equation can be solved with its wavefunction represented by B-splines which is computationally less intensive $O(N^2)$ than the commonly used planewaves $O(N^3)$. Despite the high efficiency of B-splines, the wavefunction evaluation still takes over 20% of the total application time. We recently improved the algorithm by fully taking advantage of the vectorization and optimizing the memory access on BG/Q and achieved about 3-fold speedup in the subroutines calculating multiple B-splines. Threading capability is also added to the new algorithm to maximize the single node performance. According to the specifications of the upcoming HPC systems (long vector units, more integrated cores, higher memory bandwidth), all the methods used to design the new algorithm make it ready to efficiently exploit the new features of these systems.

Categories and Subject Descriptors

G.4 [**Mathematical software**]: Algorithm design and analysis, parallel and vector implementations; G.1.10 [**Numerical analysis**]: Applications

General Terms

Algorithms

Thanks to fast evolving high performance supercomputers, first principle simulations of realistic systems with large numbers of atoms nowadays become affordable for better understanding the physics behind peculiar phenomena of new materials. Quantum Monte Carlo (QMC) is one of the most accurate and scalable methods for these simulations. In QMC calculations, the many-body Shrödinger equation $\hat{H}\Psi(\mathbf{r}) = E\Psi(\mathbf{r})$ is solved by stochastic algorithms, where $\hat{H}$ is the Hamiltonian describing the physical system, E is its energy and $\Psi(\mathbf{r})$ is its associated wavefunction. A wavefunction is usually represented in a planewave basis, which comes at high computational cost $O(N^3)$ with the number of electrons N. In our software package QMCPACK [1], the cost of evaluating the wavefunction at a given electronic configuration has been reduced to a quadratic scaling $O(N^2)$ by representing it with B-splines, real space localized cubic splines centered on a regular grid. Despite the high efficiency of B-splines, the computational cost of the wavefunction evaluation is still high (over 20% of the total application time) and we further reduce it with a new algorithm described in this work.

According to the evolution trend of top supercomputers in the world, the total number of nodes stops increasing, but more performance is obtained with increasing on-node concurrency such as multi-core, hardware threads and SIMD vectorization. Meanwhile, the memory bandwidth, limited by the frequency of DDR, grows very slowly during the past decade. The key challenge for application developers is to improve the single node performance by exploiting all the parallel capabilities and improving the utilization of the limited bandwidth of the memory subsystem.

We recently improved the B-spline evaluation subroutines by fully taking advantage of the vectorization and optimizing the memory access on BG/Q and achieved about 3-fold speedup. The codes are transformed in three steps, see Figures 1,2. In the first step, the second innermost loop is unrolled and interchanged with the innermost loop and redundant calculations are also reduced. The improvement in this step shows up on both BG/Q and Xeon processors. In the second step, vectorization is applied on the inner most loop over all the B-splines. On the Intel platform, the vectorized code is automatically generated by the compiler while on BG/Q it can be better coded with vector intrinsic functions. Thanks to vectorization, the code further speeds up significantly. On the last step which is dedicated to BG/Q, memory prefetching intrinsics are added to hide the memory latency. With these three steps of evolution, this new algorithm shows a significant increase on the memory bandwidth utilization and a much higher total Flop rates. Moreover, threading capabilities have been added to the new algorithm and can be activated to reduce the time to solution on large simulations when necessary.

With the hardware and software specifications of the upcoming and future HPC systems (long vector units, more integrated cores, higher memory bandwidth), all the methods used to design this new algorithm make it ready to run

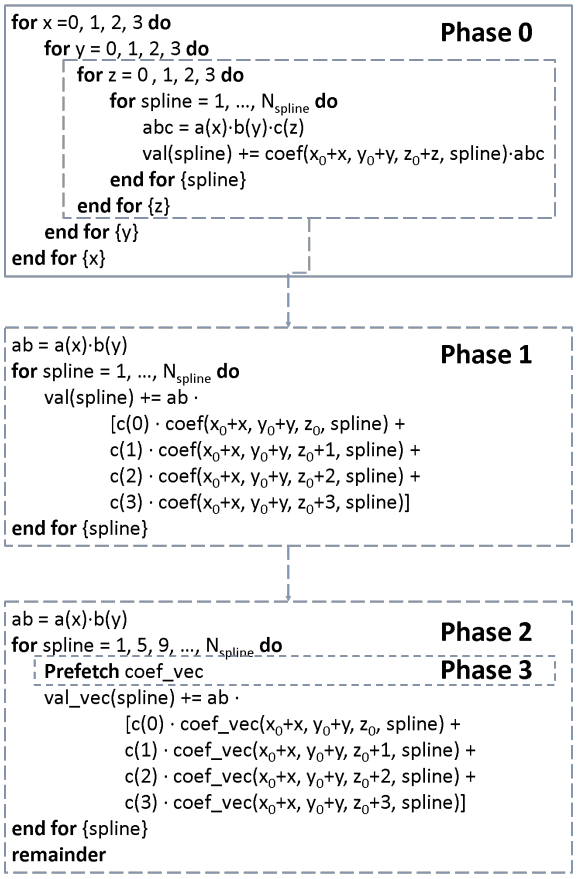

Figure 1: Improving multiple B-spline evaluation in three steps.

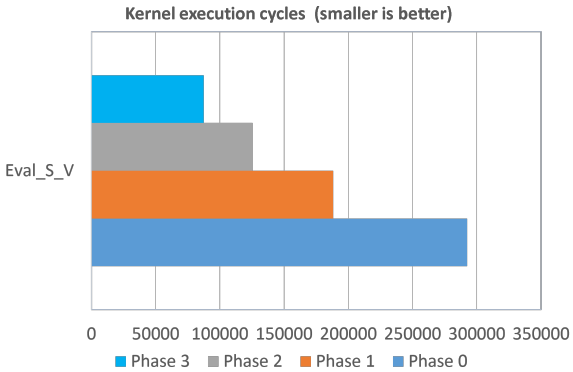

Figure 2: Time spent on subroutine Eval_S_V evaluating 503 orbitals.

efficiently by exploiting all the features of these new systems and enable us earlier access to the solution of scientific challenges.

1. REFERENCES

[1] J. Kim, K. P. Esler, J. McMinis, M. A. Morales, B. K. Clark, L. Shulenburger, and D. M. Ceperley. Hybrid algorithms in quantum monte carlo. *Journal of Physics: Conference Series*, 402(1):012008, 2012.