GPU Acceleration for the Impurity Solver in GW+DMFT Packages

Kwangmin Yu
Brookhaven National Laboratory
Upton, NY
kyu@bnl.gov

Patrick Sémon
Brookhaven National Laboratory
Upton, NY
psemon@bnl.gov

Nicholas D'Imperio
Brookhaven National Laboratory
Upton, NY
dimperio@bnl.gov

ABSTRACT

The combination of dynamical mean field theory (DMFT) and GW (or density functional theory) has become a powerful tool to study and predict properties of real materials with strongly correlated electrons, such as high temperature superconductors. At the core of this combined theory lies the solution of a quantum impurity model, and continuous-time quantum Monte Carlo (CT-QMC) has proven an indispensable algorithm in this respect. However, depending on the material, this algorithm is computationally very expensive, and enhancements are crucial for bringing new materials within reach of GW+DMFT. Based on a CPU implementation, GPU acceleration is added and two times speedup is achieved. New techniques are invented and implemented to deal with various GPU acceleration environment.

1 METROPOLIS-HASTING FOR QUANTUM IMPURITY SOLVER

Materials with strongly correlated electrons often show surprising and technologically very useful properties, for example high temperature superconductivity. Understanding these materials in order to predict their properties is hence of great interest, but turns out to be challenging because of the strong correlations. Traditional methods such as GW or density functional theory, which are very successful for weakly correlated materials, fail in the presence of strong correlations. This failure can be remedied by combining these traditional methods with dynamical mean field theory (DMFT) [1]. DMFT captures the strongly correlated electrons with the help of a quantum impurity model, which consists of an atom immersed in a sea of non-interacting electrons. This assumes that only electrons surrounding the same atom are correlated, which most of time shows to be an excellent approximation.

Solving a quantum impurity model is not trivial, and up to date, only continuous-time quantum Monte-Carlo [2] (CT-QMC) provides (statistically) exact solutions of quantum impurity models relevant for GW+DMFT. CT-QMC starts by writing the solution of the quantum impurity model as the expectation value of a random variable $O(\mathbf{c})$ with distribution $p(\mathbf{c})$. Sampling this distribution $p(\mathbf{c})$ with the help of a Metropolis-Hasting Markov Chain algorithm then yields an estimate of the solution. There are several choices for $p(\mathbf{c})$, and the one obtained by a "continuous-time hybridization expansion" is the most suitable in the context of GW+DMFT. In this case, a configuration $\mathbf{c}$ is given by a vector of indices α, and the probability of a configuration $\mathbf{c} := (\alpha_1, \alpha_2, \ldots, \alpha_{2k})$, where $k \geq 0$ is a natural number, is proportional to the product of a trace of matrices (which are specified by the indices) and a determinant,

$$p(\mathbf{c}) \propto \mathrm{Tr}[\mathbf{F}_{\alpha_1}\mathbf{F}_{\alpha_2}\cdots\mathbf{F}_{\alpha_{2k}}] \times \mathrm{Det}\mathbf{M}(\mathbf{c}). \tag{1}$$

The matrices $\mathbf{F}$ encapsulate the atomic degrees of freedom, while the $k\times k$ matrix $\mathbf{M}$ (which depends on the configuration) encapsulates the hybridization of the atom with the sea of electrons. The space of configurations is given by the set

$$\{(\alpha_1, \ldots, \alpha_{2k}) \,|\, 0 \leq k < \infty\} \tag{2}$$

of all possible vectors of indices. Proposing a new configuration $\mathbf{c}'$ in the Metropolis-Hasting algorithm by inserting/removing two indices α and α' into/from $\mathbf{c}$ is thus (in general) sufficient to explore all the space of configuration. In case of an insertion, the ratio

$$r := \frac{\mathrm{Tr}[\mathbf{F}_{\alpha_1}\cdots\mathbf{F}_{\alpha}\cdots\mathbf{F}_{\alpha'}\cdots\mathbf{F}_{\alpha_{2k}}]}{\mathrm{Tr}[\mathbf{F}_{\alpha_1}\cdots\mathbf{F}_{\alpha_{2k}}]} \times \frac{\mathrm{Det}\mathbf{M}(\mathbf{c}')}{\mathrm{Det}\mathbf{M}(\mathbf{c})} \tag{3}$$

needs to be calculated to decide whether to accept or reject the new configuration $\mathbf{c}'$ in the Metropolis-Hasting algorithm. The ratio of the determinants takes $O(k^3)$ operations (which can be reduced to $O(k^2)$ operations by using the Shermann-Morrison formula), while the calculation of the matrix product takes $O(N^3k)$ operations, where N is the dimension of the $\mathbf{F}$ matrices (which is independent of k). For the quantum impurity models, we consider here (which are relevant to study Plutonium based compounds), the dimension of the $\mathbf{F}$ matrices is typically a hundred while k is peaked around a few hundreds, so that the calculation of the product of matrices is the bottleneck in the simulation.

Different strategies have been proposed to reduce the cost of calculating the product of matrices.[3–8] When calculating the ratio in Eq. 3, the new and the old product of matrices differ only by two matrices, $\mathbf{F}_\alpha$ and $\mathbf{F}_{\alpha'}$. An idea is thus to store sub-products of matrices between successive Metropolis-Hasting steps in order to avoid recalculating the whole product of matrices from scratch at every trial step. This is the strategy followed in Refs. [3–5], using binary trees or skip lists to organize the sub-products. Another strategy, which can lead to a drastic speed-up, consists in finding a cheap bound to the acceptance ratio Eq. 3 (using some sub-multiplicative matrix norm, e.g. the Frobenius norm) in order to quickly reject trial configurations with low probability[6]. These two strategies are combined in Ref. [4] and implemented in the present code.

2 GPU ACCELERATION

By the profiling of the CPU code, the most time consuming parts are matrix-matrix multiplication (82 %) and Frobenius norm computation (14%). To accelerate those parts, cuBLAS library is used for matrix-matrix multiplication and CUDA kernel functions are implemented for Frobenius norm and others.

The implementations show more than ten times speed-up when it is compared with one CPU core. But the acceleration is not sufficient in hybrid MPI+GPU environment when multiple MPI processes share one GPU resource. Nowadays, it is becoming increasingly common for multiple MPI processes to share a limited number of GPU devices in a GPU accelerated HPC system due to the high CPU core count per node in current systems. This is still true even though a high degree of GPU density per node is the current trend. Therefore, the code acceleration is considered mainly on hybrid MPI+GPU environment. To share GPU resource more efficiently, CUDA Multi-Process Service (MPS) is applied and a new method (concurrent run of CPU and GPU) is invented and implemented.

2.1 Concurrent Run of CPU & GPU

When multiple MPI processes share a limited number of GPU devices in hybrid MPI+GPU environment, the sharing would be inefficient or even it would not be possible mainly because of GPU memory deficiency. By the concurrent run mode, some portion of MPI processes can utilize GPU resources in a node. In the method, an optimal number of MPI processes utilizes the GPU devices in a node. The remaining MPI processes not utilizing the GPU devices work in conjunction with the MPI processes utilizing the GPU devices. Since the Impurity Solver code implements for each MPI process to have its own random walk and each random walk is independent, each MPI process is independent to others and the computing time depends on the iteration number of the random walk. Therefore, when an optimal number of MPI process are chosen to use GPU and others run without GPU acceleration, good load balancing can be achieved by assigning more iterations to MPI processes using GPU. The number of more iterations can be estimated by the speed-up factor by the GPU.

3 RESULTS

The code has been tested on several clusters with a Plutonium example.

First, one node of HPC1 cluster at BNL has Intel Xeon CPU E5-2670 0 @ 2.60GHz with sixteen cores and one NVIDIA K40 GPU. It is not possible to run the code with sixteen MPI process sharing a GPU device in the system. But the concurrent run of eight MPI process using GPU and eight MPI process not using GPU shows 2.31 times speed-up.

Second, on Institutional Cluster at BNL, one node is composed of Intel Xeon CPU E5-2695 v4 @ 2.10GHz (thirty six cores) and two NVIDIA K80 GPUs. On this system, nine MPI processes can share one GPU device in a node. When nine MPI processes share one GPU device, we have 1.94 times speed-up. When six MPI processes use one GPU device and other three MPI processes run without GPU, we have 2.09 times speed-up.

Third, Titan (sixteen cores per node) at ORNL shows 2.3 times speed-up with four MPI process using GPU and twelve MPI process not using GPU. Because of the memory deficiency of GPU, only four MPI processes can utilize GPU in a node.

In conclusion, the concurrent run mode of CPU & GPU makes GPU acceleration possible and efficient.

REFERENCES

[1] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. Electronic structure calculations with dynamical mean-field theory. *Rev. Mod. Phys.*, 78:865–951, Aug 2006.

[2] Emanuel Gull, Andrew J. Millis, Alexander I. Lichtenstein, Alexey N. Rubtsov, Matthias Troyer, and Philipp Werner. Continuous-time monte carlo methods for quantum impurity models. *Rev. Mod. Phys.*, 83:349–404, May 2011.

[3] Emanuel Gull. *Continuous-time quantum Monte Carlo algorithms for fermions.* PhD thesis, ETH Zurich, 2008.

[4] P. Sémon, Chuck-Hou Yee, Kristjan Haule, and A.-M. S. Tremblay. Lazy skip-lists: An algorithm for fast hybridization-expansion quantum monte carlo. *Phys. Rev. B*, 90:075149, Aug 2014.

[5] Priyanka Seth, Igor Krivenko, Michel Ferrero, and Olivier Parcollet. Triqs/cthyb: A continuous-time quantum monte carlo hybridisation expansion solver for quantum impurity problems. *Computer Physics Communications*, 200:274 – 284, 2016.

[6] Chuck-Hou Yee. *Towards an ab initio description of correlated materials.* PhD thesis, Rutgers University, New Brunswick, NJ, USA, 2012.

[7] Kristjan Haule. Quantum Monte Carlo impurity solver for cluster dynamical mean-field theory and electronic structure calculations with adjustable cluster base. *Phys. Rev. B*, 75(15):155113, 2007.

[8] Hiroshi Shinaoka, Michele Dolfi, Matthias Troyer, and Philipp Werner. Hybridization expansion monte carlo simulation of multi-orbital quantum impurity problems: matrix product formalism and improved sampling. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(6):P06012, 2014.