

MPI/OpenMP Parallelization of the Fragment Molecular Orbitals Method in GAMESS

Vladimir Mironov
Lomonosov Moscow State University,
Department of Chemistry
Leninskie Gory 1/3
Moscow 119991, Russian Federation
vmironov@lcc.chem.msu.ru

Yuri Alexeev
Argonne National Laboratory,
Leadership Computing Facility
Argonne, Illinois 60439, USA
yuri@alcf.anl.gov

Dmitri G. Fedorov
National Institute of Advanced
Industrial Science and Technology,
Research Center for Computational
Design of Advanced Functional
Materials
Tsukuba 305-8568, Japan
d.g.fedorov@aist.go.jp

ABSTRACT

In this work, we present a novel parallelization strategy for the Fragment Molecular Orbital (FMO) method in the quantum chemistry package GAMESS. The original FMO code has been parallelized only with MPI, which limits scalability of the code on multi-core massively parallel machines. To address this problem, we parallelized FMO with a new hybrid MPI-OpenMP scheme that shows excellent scaling up to 2,048 Intel Xeon Phi nodes (131,072 cores) on Theta supercomputer. MPI-OpenMP code not only scales better compared to MPI code, but also performs up to two times faster and has significantly smaller memory footprint.

KEYWORDS

Quantum chemistry, linear scaling method, Fragment Molecular Orbital method, FMO, OpenMP, MPI, GAMESS

1 INTRODUCTION

The subject of this work is the adaptation of the Fragment Molecular Orbital (FMO) method [1, 2] in the General Atomic and Molecular Electronic Structure System (GAMESS) quantum chemistry package for the high-performance multicore CPU architectures. GAMESS [5] is a free quantum chemistry software package maintained by the Gordon research group at Iowa State University. Many quantum chemistry methods in GAMESS are parallelized with MPI only. OpenMP offers an alternative parallelization strategy by threading code using easy to use Application Program Interface (API) that is scalable and portable. The advantage of using OpenMP is that it conveniently enables sharing of the objects in memory. Therefore, OpenMP reduces memory footprint of the program and more easily leverage all of the resources (cores, memory etc.). Hybrid MPI+OpenMP approach also benefits from reduced memory overhead of MPI library and smaller amount of global communication.

The computational complexity and memory requirements of the most quantum chemistry methods scales superlinearly with the size of chemical system. Even for a relatively small molecules massive computational resources are required. One way to address this problem is to use low-order scaling methods like FMO. FMO

takes advantage of the locality of physical interactions by decomposing the system into fragments. Within each fragment all the quantum interactions are handled explicitly by traditional quantum chemistry method(s). The interaction of each fragment with the rest of the system is described as a combination of fragment-to-fragment interactions. As a result, FMO method allows to study very large chemical systems (many thousands of atoms) quickly and accurately.

The parallelization strategy for FMO method is different compared to regular quantum chemistry methods. Each fragment calculation is usually too small to scale up on many cores. However, all fragment calculations are independent of each other and can be computed in parallel inside separate MPI sub-communicators. It makes FMO method highly scalable, up to many thousands of cores [4, 9]. It is implemented in GAMESS by using an interface called Distributed Data Interface Library (DDI)[3], which can work over MPI or socket libraries.

The drawback of the current FMO implementation in GAMESS is legacy code which is mostly parallelized with MPI using static data. Another feature of the legacy GAMESS is that only half of the MPI processes actually doing computations while the other half are used as data-servers. In this setup, the overhead of MPI parallelization (memory, communication) becomes too high for practical use and an alternative approach is needed to handle compute resources. Using MPI+OpenMP parallelization scheme is a viable alternative because it improves utilization of resources of the node without significant rewrite of the legacy code. There were several efforts to parallelize traditional quantum chemistry methods in GAMESS with MPI+OpenMP [6–8]. In this work, we present new MPI+OpenMP scheme to parallelize FMO for the first time.

2 METHODOLOGY

The basic FMO algorithm (Algorithm 1) consists of two steps. First step is calculation of self-consistent field (SCF) for monomers and the second step is calculation of dimers. Monomers SCF step is done iteratively until the convergence of wavefunctions of all fragments (up to several tens of iterations). The second step is similar to the first step except that the loop runs over all fragment pairs and is done only once. All individual calculations are done in separate MPI sub-groups. We did not change the existing MPI or MPI+OpenMP parallelization of a quantum method in each individual calculation in the MPI sub-group. FMO parallelization with MPI+OpenMP is done only in FMO code with full support of the

Algorithm 1 Algorithm of FMO method

```

1: Calculate one-electron Hamiltonian of each fragment
2: Guess initial density matrix for each fragment  $D_i$ 
   ▶ Monomer calculation:
3: repeat
4:   for  $i = 1, NumFrag$  do           ▶ GDDI load balancing
5:     Compute embedding ESP matrix
6:     Compute wavefunction of the fragment  $i$ , update
       energy and density matrix of fragment  $i$  ( $\epsilon_i, D_i$ )
7:     Compute fragment properties
8:     Broadcast results (energy, density, charges)
9:   end for
10: until  $\max(|\epsilon_i^{old} - \epsilon_i|) < conv$ 
   ▶ Dimer calculation:
11: for  $i = 1, NumFrag$  do           ▶ GDDI load balancing
12:   for  $j = 1, i - 1$  do           ▶ GDDI load balancing
13:     Compute embedding ESP of the dimer
14:     Compute wavefunction, get energy ( $\epsilon_{ij}$ ) and
       density matrix( $D_{ij}$ ) of the fragment pair  $ij$ 
15:   end for
16: end for
17: Gather results of fragment pairs calculation
18: Calculate whole system properties, print results

```

underlying parallelization scheme (MPI or MPI+OpenMP) of the used quantum chemistry method. In this study, we used Hartree-Fock method for our benchmarks, which was already parallelized with MPI+OpenMP in our previous work [7].

It is important to note that the support of the parallelization scheme in the underlying quantum chemistry method is not enough to run FMO simulations efficiently. To achieve it, we parallelized FMO-specific parts of code regardless of the used underlying quantum chemistry method. To be more specific, we parallelized embedding electrostatic potential (ESP) and one-electron integrals (OEI) calculations. The ESP is computed for all fragment pairs and requires calculation of the specific subset of the electron-repulsion integrals (ERIs) of the two-fragment (“dimer”) system. The parallelization strategy resembles the one that have been used for MPI-OpenMP Hartree-Fock algorithm described in [7].

Another target for parallelization are nuclear attractions of one-electron integrals. In traditional quantum chemistry calculations, the number of one-electron integrals is negligible in comparison to other types of integrals. However, unlike other integral types they cannot be easily pre-screened by their value and in large-scale FMO simulations these calculation takes significant portion of compute time. In addition to OpenMP threading, the subroutines for integrals calculation were partially vectorized.

3 RESULTS

The benchmarks reported in this paper were performed on Intel Xeon Phi supercomputers Stampede2 located in the Texas Advanced Computing Center(TACC) and Theta located in Argonne Leadership Computing Facility (ALCF), which is a part of the U.S. Department of Energy (DOE) Office of Science (SC) Innovative and Novel Computational Impact on Theory and Experiment (INCITE)

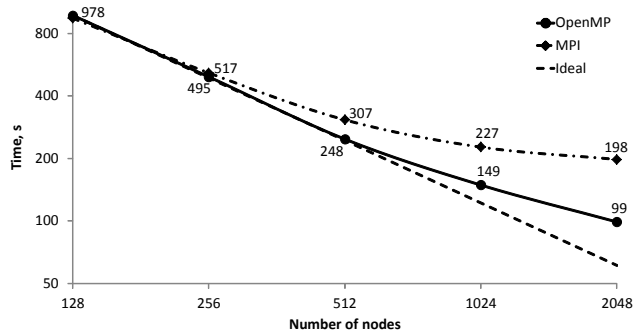


Figure 1: Comparison of the new MPI+OpenMP and legacy MPI-only implementations of the FMO method on Theta supercomputer. Times in seconds are given for the full FMO2 calculation of the 30 Å water droplet chemical system.

program. All calculations were performed in quadrant-cache mode because our previous study [7] shows that different memory and clustering modes have little impact on the time to solution for the Hartree-Fock method implemented in GAMESS.

For our benchmarks, water chemical system was chosen. This system is ideal because water is often computed with FMO and the size of the system can be easily changed. The model chemical system is a spherical water droplet with 30 Å radius. We used two setups to test FMO parallelization. The first setup was single iteration of FMO1 run which was carried out on Stampede2. Another setup was full FMO2 calculation using the same chemical system which was carried out on Theta.

The FMO1 setup is relatively simple and scales fairly well up to 64 KNL nodes with $\approx 76\%$ efficiency. The FMO2 setup is computationally much more expensive. We observed excellent scalability up to 2,048 Theta KNL nodes (Figure 1). The new code not only scales better than the legacy MPI-only implementation but is up to two times faster than the latter.

4 CONCLUSION

In this work, we present novel hybrid MPI+OpenMP scheme to parallelize FMO code in GAMESS. The code was benchmarked on Intel Xeon Phi supercomputers Theta and Stampede2. Our new hybrid MPI+OpenMP code significantly outperforms the legacy MPI-only code in GAMESS. MPI+OpenMP code scales excellently up to 2,048 of Intel Xeon Phi nodes and runs up to 2 times faster than MPI-only version. Moreover, MPI+OpenMP has smaller memory footprint.

ACKNOWLEDGMENTS

This research used the resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy (DOE) Office of Science User Facility supported under Contract DE-AC02-06CH11357. The authors thank the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources. We also thank the Intel® Parallel Computing Centers program for funding.

REFERENCES

- [1] Dmitri G Fedorov and Kazuo Kitaura. 2007. Extending the power of quantum chemistry to large systems with the fragment molecular orbital method. *The Journal of Physical Chemistry A* 111, 30 (2007), 6904–6914.
- [2] D G Fedorov and K Kitaura (Eds.). 2009. *The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems*. CRC Press, Boca Raton, FL, USA.
- [3] Dmitri G. Fedorov, Ryan M. Olson, Kazuo Kitaura, Mark S. Gordon, and Shiro Koseki. 2004. A new hierarchical parallelization scheme: Generalized distributed data interface (GDDI), and an application to the fragment molecular orbital method (FMO). *Journal of Computational Chemistry* 25, 6 (apr 2004), 872–880.
- [4] Graham D. Fletcher, Dmitri G. Fedorov, Spencer R. Pruitt, Theresa L. Windus, and Mark S. Gordon. 2012. Large-Scale MP2 Calculations on the Blue Gene Architecture Using the Fragment Molecular Orbital Method. *Journal of Chemical Theory and Computation* 8, 1 (jan 2012), 75–79.
- [5] Mark S Gordon and Michael W. Schmidt. 2005. Advances in electronic structure theory: GAMESS a decade later. In *Theory and Applications of Computational Chemistry: the first forty years*, CE Dykstra, G. Frenking, KS Kim, and GE Scuseria (Eds.). Elsevier, Amsterdam, 1167–1189.
- [6] Kazuya Ishimura, Kei Kuramoto, Yasuhiro Ikuta, and Shi-aki Hyodo. 2010. MPI/OpenMP hybrid parallel algorithm for Hartree-Fock calculations. *Journal of Chemical Theory and Computation* 6, 4 (2010), 1075–1080.
- [7] Vladimir Mironov, Yuri Alexeev, Kristopher Keipert, Michael D’Mello, Alexander Moskovsky, and Mark S. Gordon. An efficient MPI/OpenMP parallelization of the Hartree-Fock method for the second generation of Intel Xeon Phi processor. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC ’17*. ACM Press, New York, New York, USA, 1–12.
- [8] Vladimir Mironov, Alexander Moskovsky, Michael D’Mello, and Yuri Alexeev. 2017. An efficient MPI/OpenMP parallelization of the Hartree-Fock method for the first generation of Intel Xeon Phi processor architecture. *The International Journal of High Performance Computing Applications* (Oct 2017).
- [9] Spencer R. Pruitt, Hiroya Nakata, Takeshi Nagata, Maricris Mayes, Yuri Alexeev, Graham Fletcher, Dmitri G. Fedorov, Kazuo Kitaura, and Mark S. Gordon. 2016. Importance of Three-Body Interactions in Molecular Dynamics Simulations of Water Demonstrated with the Fragment Molecular Orbital Method. *Journal of Chemical Theory and Computation* 12, 4 (apr 2016), 1423–1435.