

# MPI+OpenMP paralelization of density functional theory method in GAMESS

Vladimir Mironov  
Lomonosov Moscow State University,  
Department of Chemistry  
Moscow, Russian Federation  
vmironov@lcc.chem.msu.ru

Yuri Alexeev  
Argonne National Laboratory,  
Computational Science Division  
Argonne, Illinois, USA  
yuri@alcf.anl.gov

Dmitri G. Fedorov  
National Institute of Advanced  
Industrial Science and Technology  
Tsukuba, Japan  
d.g.fedorov@aist.go.jp

## ABSTRACT

In this work, Density Functional Theory (DFT) method is parallelized with MPI-OpenMP in quantum chemistry package GAMESS. It has been implemented in both regular and Fragment Molecular Orbital (FMO) based DFT codes. The scalability of the FMO-DFT code was demonstrated on Cray XC 40 Theta supercomputer. We demonstrated excellent scalability of the code up 2,048 Intel Xeon Phi nodes (131,072 cores). Moreover, the developed DFT code is about twice faster than the original code because of our new grid integration algorithm.

## KEYWORDS

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

## 1 INTRODUCTION

General atomic and molecular electronic structure system (GAMESS) [1, 2] is a free quantum chemical software maintained by the Gordon research group at Iowa State University. Most of the methods in GAMESS code are parallelized with MPI or a native socket library using the framework of the distributed data interface (DDI) [3]. OpenMP offers an alternative parallelization strategy by threading the code. OpenMP supports sharing of arrays in memory and reduces the memory footprint. A hybrid MPI+OpenMP approach also benefits from reduced memory overhead of MPI and a smaller amount of global communication. The purpose of this work is to develop a hybrid MPI/OpenMP parallelization for the density functional theory (DFT) in GAMESS.

Typically, the computational cost and memory requirements of quantum chemical methods scale at least quadratically with the chemical system size depending on theory. The fragment molecular orbital (FMO) method [4–6] solves this scaling problem by decomposing a large system into fragments. Fragments are computed explicitly with quantum chemical methods. As a result, the scaling of FMO is nearly linear, making large scale simulations feasible. Fragment calculations can be well parallelized [7, 8] using generalized distributed data interface (GDDI)[9].

In DDI only one half of the MPI ranks does computations, and the other half is used as data servers. The memory and communication overhead of such parallelization is high, especially when there are many cores per node. Replacing intranode MPI parallelization with OpenMP in a hybrid MPI+OpenMP parallelization is an attractive way to improve the efficiency. In the earlier work [10–13]

MPI+OpenMP code was developed for HF. HF, however, is less accurate than DFT in general, and for practical applications it is very desirable to use DFT. In this work, an MPI+OpenMP parallelization is developed for the DFT method in GAMESS.

## 2 METHODOLOGY

The basic FMO algorithm consists of two steps, the calculation of self-consistent field (SCF) for monomers and dimers. Monomers SCF step is done iteratively until the convergence of all fragments with respect to the polarizable embedding. Each dimer is calculated only once. FMO-HF code was parallelized with MPI+OpenMP earlier [13].

DFT is a quantum chemical method, which can predict properties of chemical systems accurately with a relatively low computational cost. The DFT algorithm is very similar to the Hartree-Fock (HF) method. HF in GAMESS was parallelized with MPI+OpenMP earlier [11, 12]. DFT requires an integration of the exchange-correlation (XC) functional over real space, in addition to the steps in HF calculations. The integration of the functional is commonly done numerically, and it takes a long time.

The algorithm of the DFT-specific code is shown in Algorithm 1. In DFT, the Fock matrix includes the Coulomb and exchange contributions from HF (the exchange part may be scaled or set to zero), and to this the exchange and correlation (XC) terms specific to DFT are added. In this work, both energy and its analytic gradient were parallelized, and the gradient is omitted from Algorithm 1 for simplicity.

There are various choices of grid point distributions in DFT. In this work, the Becke’s atom-centered scheme [14] is used, with

---

### Algorithm 1 Schematic algorithm for DFT

---

```
1: Set up numerical grid ( $\mathbf{r}_i, \omega_i$ )
2: repeat ▷ Begin SCF loop
3:   Compute Coulomb and optionally exact exchange terms
4:   for each ( $\mathbf{r}_i, \omega_i$ ) do
5:     Compute electron density  $\rho(\mathbf{r}_i)$ 
6:     (Optional, GGA) compute density gradient  $\nabla\rho(\mathbf{r}_i)$ 
7:     (Optional, mGGA) compute kinetic energy density  $\tau(\mathbf{r}_i)$ 
8:     Compute  $f_i^{XC}(\rho, \nabla\rho, \tau)$ 
9:     Update the Fock matrix using
        $F_{mn} += \omega_i f_i^{XC} \phi_m(\mathbf{r}_i) \phi_n(\mathbf{r}_i)$  ▷ rank-2 update
10:   end for
11:   Diagonalize the Fock matrix, update  $\rho$  and  $\epsilon$ 
12: until  $|\mathbf{D} - \mathbf{D}^{old}| < conv$  or  $|\epsilon - \epsilon^{old}| < conv$ 
```

---

the Lebedev angular distribution of points, combined with a radial distribution.

The central idea of Becke’s algorithm is to adjust quadrature weights for regions, where grids of different atoms intersect. In this algorithm, the real space is separated using fuzzy Voronoy cells. The weight of each atomic quadrature point is scaled by the Becke’s factor, where each point “belongs” to an atom. The Becke’s factor for a point is close to unity in the vicinity of its atom and is zero near other atoms. The drawback of the original Becke’s algorithm is its high  $O(N^3)$  scaling. However, the scaling can be improved by using different criteria for Voronoy partitioning [15].

In this work, special attention was paid to finding points where Becke’s factors are equal to zero or one. Then one can skip the calculation of Becke’s factors and reduce the grid size in memory for points with a zero weight. We also used the fact that points which are close to each other should have similar Becke’s factors. Thus, we split the atomic grid to localized clusters of points. Each atomic grid point is assigned to a layer depending on the distance to the nucleus; each layer is divided on 3D tiles of varying size. The advantage of this setup is that points can be manipulated in groups. It simplifies the memory management and makes it possible to use BLAS level 3 (instead of BLAS level 2) subroutines for updating the Fock matrix. Additionally, all in-house linear algebra subroutines were rewritten to use BLAS libraries as well.

### 3 RESULTS

In the first series of benchmarks the new and original DFT implementations are compared. A dual-CPU Intel Xeon Gold 6254 node was used for this benchmark. A series of B3LYP/6-31G\* energy and gradient calculations of phenylalanine amino acid were performed using various numbers of cores. A high precision numerical grid (radial quadrature: 96-point radial Murray-Handy-Laming grid and 1202-point angular Lebedev grid) was used. Timings of the main steps in DFT are shown in Table 1. The measured speedup factor for DFT steps is 2 or more for the energy, and up to 5 for the DFT gradient step.

The parallel scaling benchmarks reported in this work were performed on Cray XC40 Intel Xeon Phi supercomputer Theta located at 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) program. The calculations were performed in quadrant-cache mode because a previous study [12] showed that different memory and clustering modes have little impact on the time to solution for HF implemented in GAMESS. For massively parallel scaling benchmarks, a spherical water droplet with the 30 Å radius was used and a good scalability up to 2,048 Theta KNL nodes was obtained (Figure 1).

### 4 CONCLUSIONS

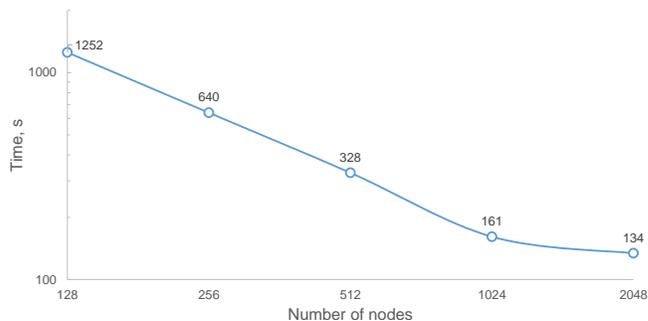
In this work, Density Functional Theory (DFT) method has been parallelized with MPI-OpenMP in quantum chemistry package GAMESS. The parallel performance has been measured on a single Intel Xeon Phi node and supercomputer Theta (Cray XC40 based on Intel Xeon Phi). The developed hybrid MPI+OpenMP code scales well up to 2,048 Intel Xeon Phi nodes. Moreover, we also rewrote

**Table 1: Wall-clock timings in seconds for the new and original DFT implementations in GAMESS computed on a dual-CPU Intel Xeon Gold 6254 server for chemical system phenylalanine at the level of B3LYP/6-31G\*.** “Init.” column means grid setup, ( $E_{xc}$ ) is energy, ( $\nabla E_{xc}$ ) is nuclear gradient, and “Tot.” is total DFT timing<sup>1</sup>.

$N_c^2$	New DFT code				Original DFT code			
	Init.	$E_{xc}$	$\nabla E_{xc}$	Tot. <sup>1</sup>	Init.	$E_{xc}$	$\nabla E_{xc}$	Tot. <sup>1</sup>
1	2.1	108.7	21.2	167.7	6.3	253.9	81.8	376.7
2	1.1	55.3	10.1	85.3	3.3	135.9	43.9	201.1
4	0.59	32.6	5.1	48.0	1.7	69.3	22.4	102.6
8	0.33	18.3	2.6	26.1	0.90	35.1	11.3	52.1
12	0.24	12.7	1.8	18.6	0.67	25.4	8.3	38.1
18	0.19	9.4	1.3	13.5	0.50	17.9	5.7	27.2
24	0.15	7.9	1.1	11.3	0.50	16.6	5.3	25.8
36	0.15	6.7	0.91	10.6	0.40	14.4	4.6	22.9

<sup>1</sup> The total timing includes the HF step, which is not shown separately.

<sup>2</sup> The number of MPI compute processes (not counting data servers) or OpenMP threads.



**Figure 1: Scalability of the hybrid MPI+OpenMP FMO-DFT code in GAMESS on Theta supercomputer for the 30 Å water droplet. Times in seconds are given for a single FMO2 energy calculation. “Time” is total time in seconds and “Number of nodes” is the number Xeon Phi nodes.**

DFT grid integration code, resulting in about 2x performance improvement compared to the original code.

### ACKNOWLEDGMENTS

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. 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. VM thanks the Intel Parallel Computing program for funding and providing access to its hardware and software.

## REFERENCES

- [1] Michael W. Schmidt, Kim K. Baldrige, Jerry A. Boatz, Steven T. Elbert, Mark S. Gordon, Jan H. Jensen, Shiro Koseki, Nikita Matsunaga, Kiet A. Nguyen, Shujun Su, et al. General Atomic and Molecular Electronic Structure System. *Journal of Computational Chemistry*, 14(11):1347–1363, 1993.
- [2] Mark S. Gordon and Michael W. Schmidt. Advances in electronic structure theory: GAMESS a decade later. In C. E. Dykstra, G. Frenking, K. S. Kim, and G. E. Scuseria, editors, *Theory and Applications of Computational Chemistry: the first forty years*, pages 1167–1189. Elsevier, Amsterdam, 2005.
- [3] Graham D. Fletcher, Michael W. Schmidt, Brett M. Bode, and Mark S. Gordon. The distributed data interface in GAMESS. *Computer Physics Communications*, 128(1):190–200, 2000.
- [4] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi. Fragment molecular orbital method: an approximate computational method for large molecules. *Chem. Phys. Lett.*, 313:701–706, 1999.
- [5] Dmitri G. Fedorov and Kazuo Kitaura. Extending the power of quantum chemistry to large systems with the fragment molecular orbital method. *The Journal of Physical Chemistry A*, 111(30):6904–6914, 2007.
- [6] Dmitry G. Fedorov and K. Kitaura, editors. *The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems*. CRC Press, Boca Raton, FL, USA, 2009.
- [7] Graham D. Fletcher, Dmitri G. Fedorov, Spencer R. Pruitt, Theresa L. Windus, and Mark S. Gordon. Large-Scale MP2 Calculations on the Blue Gene Architecture Using the Fragment Molecular Orbital Method. *Journal of Chemical Theory and Computation*, 8(1):75–79, 2012.
- [8] Spencer R. Pruitt, Hiroya Nakata, Takeshi Nagata, Maricris Mayes, Yuri Alexeev, Graham Fletcher, Dmitri G. Fedorov, Kazuo Kitaura, and Mark S. Gordon. 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):1423–1435, 2016.
- [9] Dmitri G. Fedorov, Ryan M. Olson, Kazuo Kitaura, Mark S. Gordon, and Shiro Koseki. 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):872–880, 2004.
- [10] Kazuya Ishimura, Kei Kuramoto, Yasuhiro Ikuta, and Shi-aki Hyodo. Mpi/openmp hybrid parallel algorithm for hartree-fock calculations. *Journal of Chemical Theory and Computation*, 6(4):1075–1080, 2010.
- [11] Vladimir Mironov, Alexander Moskovsky, Michael D’Mello, and Yuri Alexeev. An efficient MPI/OpenMP parallelization of the Hartree-Fock-Roothaan method for the first generation of Intel Xeon Phi processor architecture. *The International Journal of High Performance Computing Applications*, 33(1):212–224, 2017.
- [12] 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*, pages 1–12, New York, New York, USA, 2017. ACM Press.
- [13] Vladimir Mironov, Yuri Alexeev, and Dmitri G. Fedorov. Multithreaded parallelization of the energy and analytic gradient in the fragment molecular orbital method. *International Journal of Quantum Chemistry*, 119(12):e25937, 2019.
- [14] A. D. Becke. A multicenter numerical integration scheme for polyatomic molecules. *The Journal of Chemical Physics*, 88(4):2547–2553, 1988. ISSN 00219606.
- [15] R. Eric Stratmann, Gustavo E. Scuseria, and Michael J. Frisch. Achieving linear scaling in exchange-correlation density functional quadratures. *Chemical Physics Letters*, 257(3-4):213–223, 1996. ISSN 00092614.