Development of GPU-accelerated Fragment Molecular Orbital Program

Center for Computational Sciences, University of Tsukuba

Hiroaki Umeda[1*], Toshihiro Hanawa[2], Mitsuo Shoji[1], Taisuke Boku[1], and Yasuteru Shigeta[1]

[1]Center for Computational Sciences, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305–8577, Japan. [2]Information Technology Center, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8589, Japan. *E-mail address: *umeda@ccs.tsukuba.ac.jp*

Abstract

GPU-accelerated FMO (fragment molecular orbital) program has been implemented with CUDA. Performance benchmarks were executed on HA-PACS base-cluster, and shows 3.8x speedups from CPU on-the-fly FMO calculation for middle-sized protein. As an example of large-scale FMO calculation, FMO/HF/6-31G(d) calculation of 23,460-atomic molecule was performed with 256GPUs of HA-PACS base-cluster, and it was successfully executed in two hours.

Introduction

Ab initio molecular orbital (MO) calculation is a standard tool to obtain molecular properties and to analyze reaction mechanisms in quantum chemistry, and it is wanted to perform MO calculation much faster for large-scale molecule, such as proteins. Fragment molecular orbital (FMO) method [1] is one of such MO calculation technique. Besides this computational technique, it is also required to execute MO calculation on modern computer architecture, such as an accelerator computer, in order to realize large-scale MO calculation.

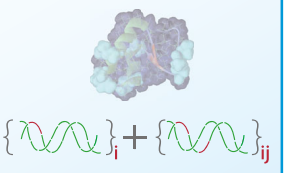
We have been developed GPU-accelerated routines on a FMO program, OpenFMO [2], which is a simple FMO program and is designed for massively parallel computer, with CUDA language.

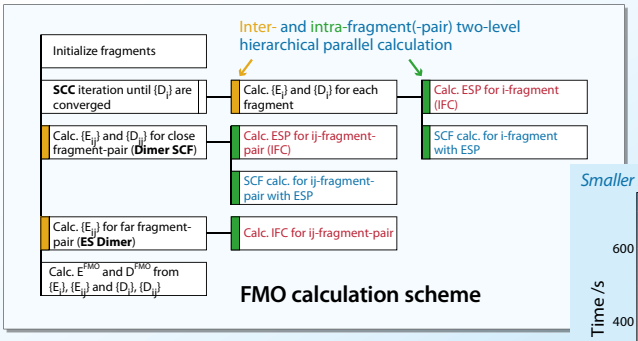
In this presentation, we briefly introduce FMO method and shows some benchmark results of our GPU-enable FMO calculation on HA-PACS GPU cluster, including the first large-scale GPU-accelerated FMO calculation for 23,460-atomic molecular system.

[1] K. Kitaura et al., *Chem. Phys. Lett.*, **312**, 319 (1999).
[2] OpenFMO; http://www.openfmo.org

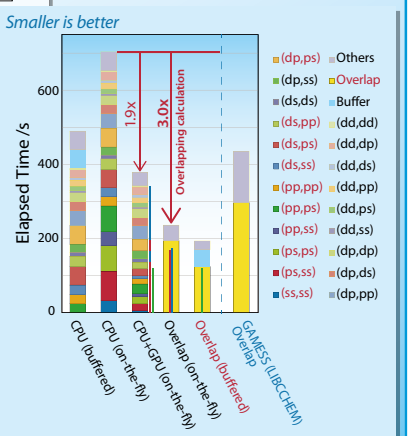
FMO Method and its Hotspot

The FMO (fragment molecular orbital) method is one of the computational technique to perform large MO calculation, where large system is treated as a ensemble of smaller fragments and fragment-pairs with self-consistent electrostatic charge condition, and is expected to be a quantum chemistry application on the modern parallel computer system. Because both inter- and intra-fragment(-pair) calculation can be parallelized independently, FMO calculation can be executed with two-level hierarchical parallel execution model for massively parallel computer.





Target to GPGPU
- **Fock matrix construction**
- **Inter-fragment Coulomb interaction calculation (IFC)**



Performance of GPU-accelerated HF calculation [3]

Elapsed time for HF/6-31G(d) calculations of model DNA molecule $(CG)_2$ (126 atoms, 1,282AO) with a single HA-PACS base-cluster node (16 CPU core and 4 GPU cards) w/wo overlap. cf.) GAMESS(2013May)+libcchem with CPU&GPU.

3.0x speedups by CPU&GPU overlapping calculations for Fock matrix construction

[3] H. Umeda, et al., *IPSJ Transactions on Advanced Computing Systems*, **6**,4, 26-37 (2013); H. Umeda, et al., *HPC in Asia poster presentation in ISC'13*, 19a (2013).

Performance of GPU-accelerated ESP calculation [4]

No.	2e-Type	CPU Etime [s]	GPU Etime [s]	Larger is better Speedups
1	(ps,ps)	27.36	2.91	9.4
2	(ss,ss)	18.99	2.88	6.6
3	(ss,ps)	16.05	1.95	8.2
4	(ps,ss)	15.20	2.45	6.2
5	(ds,ps)	8.67	2.48	3.5
6	(ps,ds)	7.79	1.65	4.7
7	(ps,pp)	7.30	3.06	2.4
8	(pp,ps)	6.97	3.75	1.9
9	(ds,ss)	6.54	0.89	7.4
10	(dp,ps)	6.52	3.67	1.8
11	(ss,ds)	6.16	0.64	9.6
12	(ps,dp)	5.69	3.06	1.9
13	(ss,pp)	5.50	0.75	7.3
14	(pp,ss)	4.99	1.01	4.9
15	(dp,dp)	4.70		
16	(ds,pp)	3.93	2.80	1.4
17	(dp,ss)	3.84	0.96	4.0
18	(ds,ds)	3.71	1.92	1.9
19	(pp,pp)	3.63		
20	(ss,dp)	3.60	0.61	5.9
22	(pp,ds)	3.33	2.40	1.4
36	(ss,dd)	0.80	0.29	2.8
Total	CPU	199.15	37.00	
	GPU		40.13	

Accumulated elapsed time of 4C-IFC (fragment-monomer ESP calculation) for each integral type in a SCC process of FMO-HF/6-31G(d) calculation of $(Ala)_{10}$ (112 atom) with a single HA-PACS base-cluster node (16 CPU core and 4 M2090 GPU cards)

5x speedups are expected by CPU&GPU overlapping calculations for 4C-IFC ESP calculation

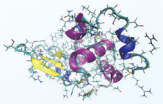
[4] H. Umeda, et al., *IPSJ Sig Technical Reports*, 2015-HPC-151, No.26 (2015).

Performance Benchmark

Middle-sized protein

Elapsed time [sec.] for FMO-HF/6-31G(d) calculations of lysozyme molecule with HA-PACS GPU cluster. The calculations are performed with 15 worker groups of 2 (4, 8) ranks. (1 rank = 4 (5 for TCA) CPU core and 1 GPU).

HA-PACS	Base cluster					TCA
	CPU	CPU+GPU				CPU+GPU
	On-the-fly	*On-the-fly*	***Speedups***	*On-the-fly*	*On-the-Fly*	*On-the-Fly*
#node (#GPU)	8	8 (32)		16 (64)	32 (128)	8 (32)
SCC [s]	3,070.5	827.5	***3.7***	450.2	308.2	457.6
Dimer SCF [s]	6,246.2	1,674.8	***3.7***	898.4	530.7	960.8
ES Dimer [s]	407.3	77.8	***5.2***	42.2	24.8	39.8
Total [s]	9,770.5	2,596.8	***3.8***	1,429.6	902.0	1,498.9



Lysozyme (1,961 atoms, 57 fragments)

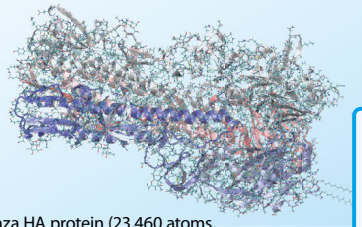
3.8x speedups by GPUs than CPU on-the-fly FMO calculation

Large-scale FMO application

A large-scale FMO calculation is examined with our GPU-enable FMO program. The application is FMO-HF/6-31G(d) calculation of influenza HA protein (HA), which is considered as acting important roles with influenza virus. This calculation is performed on 64 nodes of HA-PACS base cluster.

For comparison, we also shows the benchmark results of FMO-RI-MP2/6-31G(d)+cc-pVDZ calculation of this molecule with 24,576 nodes of K computer using customized GAMESS program with SIMD-enchanced ESP routine provided by X-Ability Co. Ltd.

Application	HA (23,460 atoms)	
HPC system	HA-PACS	K
FMO Theor. Level	HF	RI-MP2
#nodes (#GPU)	64 (256)	24,576
FLOPS(HWpeak)	192 TF	3.1 PF
#groups (SCC/dimer)	84/84	358/1529
SCC [min]	31.5	3.8
Dimer SCF [min]	54.3	--
ES Dimer [min]	27.2	(1.3)
Total [min]	**118.2**	10.7



Influenza HA protein (23,460 atoms, 721 fragments) [5]

Two hours for FMO calculation of large protein (23,460 atom) with 256 GPUs

[5] T. Sawada, et al., *J Phys. Chem. B*, **114**, 15700-15705 (2010).

HA-PACS GPU cluster system

HA-PACS Base-cluster (268 nodes, 802TFLOPS)
- CPU: 2 Intel E5-2680 2.6GHz (8core), 20.8GFLOPS/core
- GPU: 4 NVIDIA Tesla M2090 (Fermi), 665GFLOPS/GPU
- Network: 2xQDR IB

HA-PACS/TCA (64 nodes, 364TFLOPS)
- CPU: 2 Intel E5-2680v2 2.8GHz (10core), 22.4GFLOPS/core
- GPU: 4 NVIDIA Tesla K20Xm (Kepler), 1.23TFLOPS/GPU
- Network: 2xQDR IB

HA-PACS GPU cluster system (Left: HA-PACS/TCA, Right: base-cluser)

Softwares: Intel C Compiler 15.0, CUDA Toolkit 6.5, Intel MPI 5.0

Summary

We have implemented a GPU-accelerated FMO program and its performance shows 3.8 times faster than CPU on-the-fly calculation for middle-sized protein on the HA-PACS base-cluster. For a large-scale MO application, FMO calculation of 23,460 atomic protein was examined, and was successfully executed in 2 hours with 256 GPUs. As compared with FMO-application on K-computer, our GPU-accelerated program has reasonable efficiency.

Acknowledgements: This study is partially supported by the project "Research and Development on Unified Environment of Accelerated Computing and Interconnection for Post-Petascale Era" of Basic Research Programs: CREST "Development of System Software Technologies for post-Peta Scale High Performance Computing," Japan Science and Technology Agency, and FMO benchmarks have been carried out under the "Interdisciplinary Computational Science Program" in CCS, University of Tsukuba. The model of Influenza HA protein is provided by T. Sawada (AIST, Fujifilm Corporation).

筑波大学 University of Tsukuba

Japan Science and Technology Agency