HPCNano06 at Nov. 13, 2006 in Tampa, FL.

Open-architecture Implementation of Fragment Molecular Orbital Method for Peta-scale Computing

Toshiya Takami, J. Maki, J. Ooba,
Y. Inadomi, H. Honda, T. Kobayashi,
R. Nogita, M. Aoyagi

Computing & Communications Center
Kyushu University
Japan

NAREGI

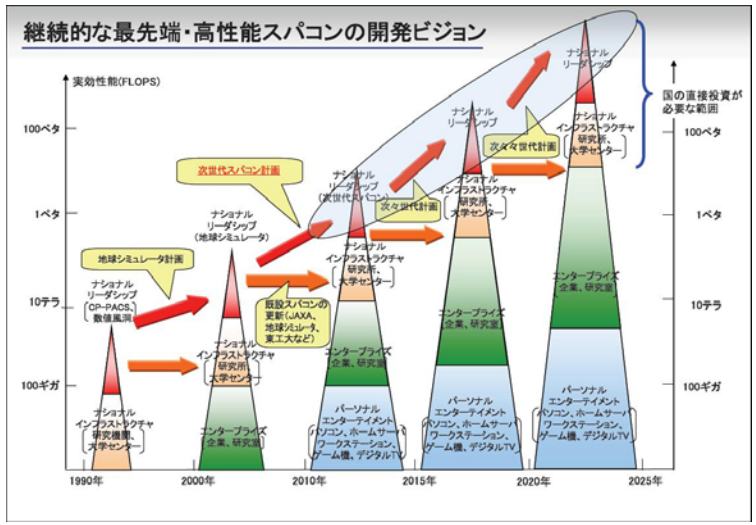
九州大学
KYUSHU UNIVERSITY

Contents

- From Grid to Peta-scale Computing
  - TeraGrid, EGEE, NAREGI
  - Next-generation Supercomputer Project (Riken, Japan)
- Fragment Molecular Orbital Method
  - All electron calculation for large molecules
- Grid-enabled Version of GAMESS-FMO
  - Loosely-coupled FMO on NAREGI test beds
- Performance Prediction of FMO
  - Phenomenological approach to peta computing
- OpenFMO Project for the Next Generation
  - Open architecture implementation
  - Open source software development
  - Open to multi-physics simulations

From Grid to Peta-scale Computing

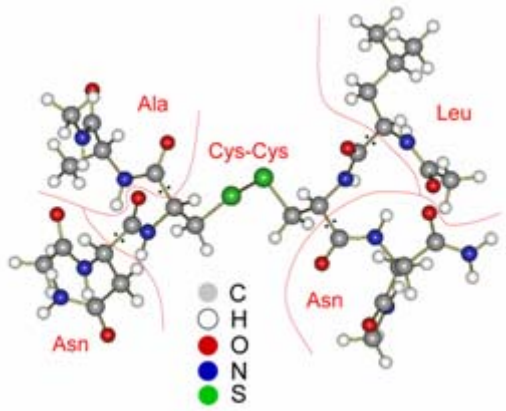
- TeraGrid 
  - http://www.teragrid.org/
- EGEE
  - http://www.eu-egee.org/
- NAREGI 
  - http://www.naregi.org/
- Next-generation Supercomputer Project
  - http://www.nsc.riken.jp/



理化学研究所 次世代スーパーコンピュータ開発実施本部
RIKEN Next-Generation Supercomputer R&D Center

Fragment Molecular Orbital Method (1)

- Fragment MO method is developed by Dr. Kitaura and his co-workers in Tsukuba, Japan, and is known as an approximate method of all electron calculation for large molecules.
- At first, the target molecule is divided into fragments each of which contain one or two residues. Self-consistent field calculation of each fragment is performed under the electro-static potential made by other fragments. After improvements with respect to pairs of fragments, the total energy is obtained.
- This algorithm has been implemented in ABINIT-MP (http://moldb.nihs.go.jp/abinitmp/), GAMESS (http://www.msg.ameslab.gov/GAMESS/), etc.



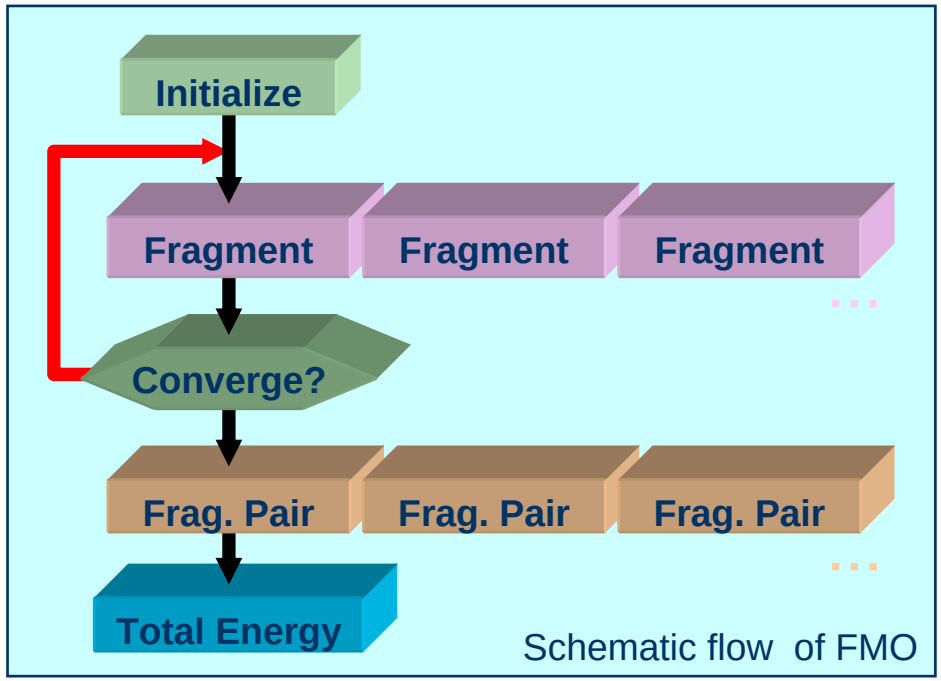
Works on FMO by Dr. Kitaura's group:

K. Kitaura, E. Ikeo, T. Asada, T. Nakano and M. Uebayasi, *Chem. Phys. Lett.* **313**, 701 (1999).
K. Kitaura, S. Sugiki, T. Nakano, Y. Komeiji and M. Uebayasi, *Chem. Phys. Lett.* **336**, 163 (2001).
D.G. Fedorov and K. Kitaura, *J. Chem. Phys.* **120**, 6832 (2004).
D.G. Fedorov and K. Kitaura, *J. Chem. Phys.* **121**, 2483 (2004).

Fragment Molecular Orbital Method (2)

- The flow of FMO calculation can be represented in this figure.
  - The main part is a self-consistent loop of SCF calculations of fragments under the electro-static potential made by other fragments.
  - This is executed until the total electro-static potential is converged.
  - After the convergence, the fragment-pair calculation is carried out over all combinations of two fragments in order to improve the result.

- Parallel execution
  - Calculations of fragments and fragment-pairs can be parallelized.
  - Since the SCF calculation itself can be parallelized, FMO is executed under a hierarchical parallelization scheme.

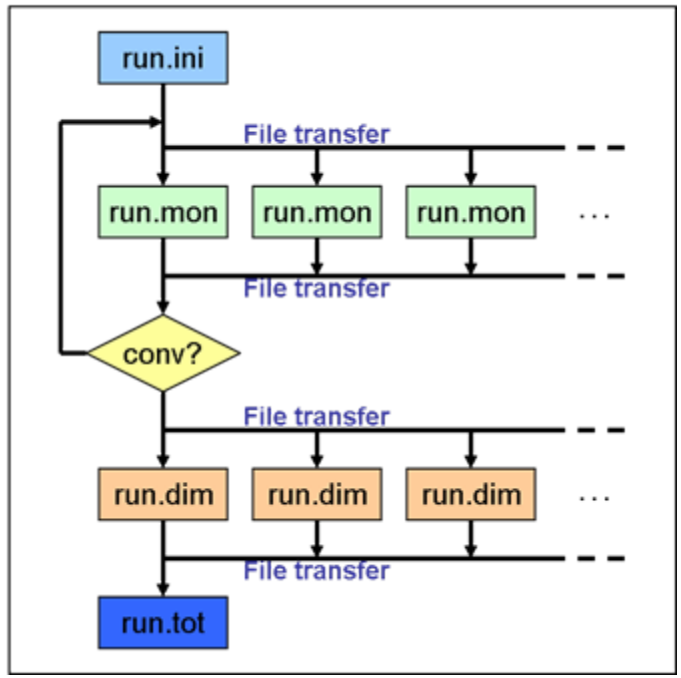


Schematic flow of FMO

D.G..Fedorov, R.M. Olson, K. Kitaura, M.S. Gordon, and S. Koseki, *J. Comp. Chem.* **25**, 872 (2004).

Grid-enabled Version of FMO (1)

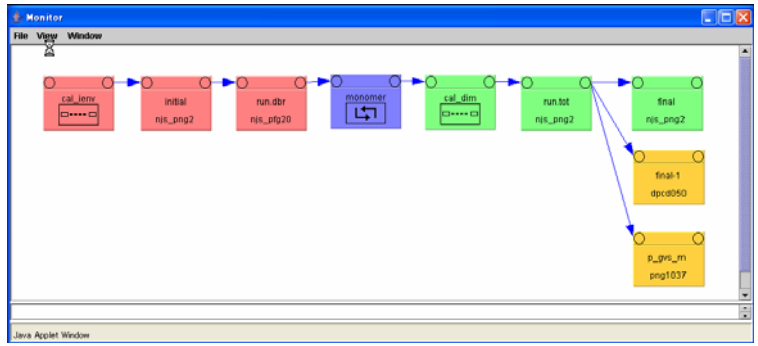
- In order to execute FMO calculations on Grid environments, we have developed the Grid-enabled version of GAMESS-FMO (called Grid-FMO or The Loosely-coupled FMO) on NAREGI Grid.
- The loosely-coupled FMO is constructed by dividing the original GAMESS-FMO into following modules:
  - **run.ini**: initialization
  - **run.mon**: calculation of a fragment
  - **run.dim**: calculation of a fragment-pair
  - **run.tot**: total energy calculation
- Those modules are invoked from a control shell program.
- These modules are loosely-coupled through file transfers.
- After the reconstruction of the program, we can execute it on grid environment .



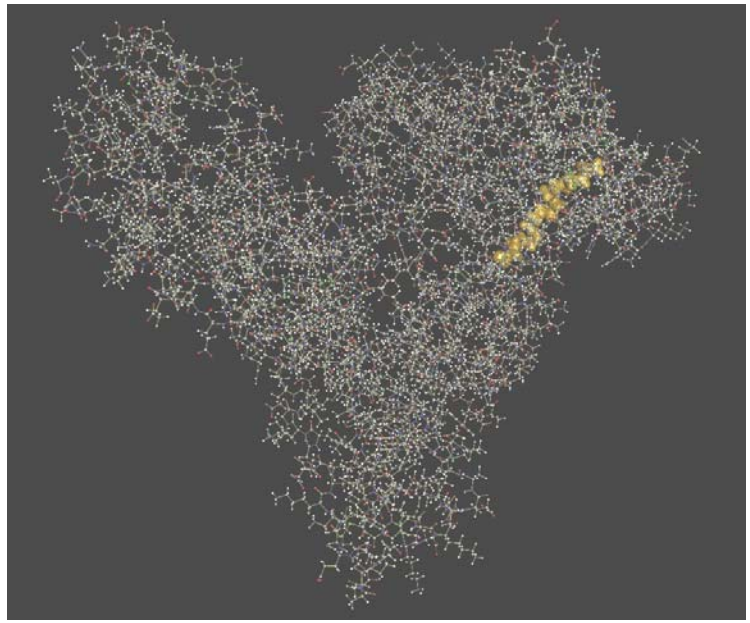
S. Matsuoka, S. Shimojo, M. Aoyagi, S. Sekiguchi, H. Usami, and K. Miura, *Proc. IEEE* **93**, 522 (2005).

Grid-enabled Version of FMO (2)

- Since the data transfer between those modules are done through files, a stable execution on distributed machines, PC clusters, large-scale grids, etc., is realized.
- The large-scale parallel execution on the NAREGI testbeds can be done by writing the loosely-coupled FMO on NAREGI Workflow Tool.
- By the help of the NAREGI grid middleware, an effective execution is guaranteed.
- We have also developed an initial value database of the fragment electron density to reduce the SCF loop counts and the total MO calculation with the grid visualization functionality.



NAREGI Workflow Tool



fatty-acid albmin

Grid-enabled Version of FMO (3)

- It is inevitable in general that the total elapsed time increases when we reconstruct the application to a loosely-coupled form. This increase of the execution time is considered as a cost for grid-enabling.
- In the present case, however, the extra computational time on a grid-environment is relatively small (about 10%) compared to the total elapsed time of the original FMO. Thus, our program can be executed effectively on the grid.

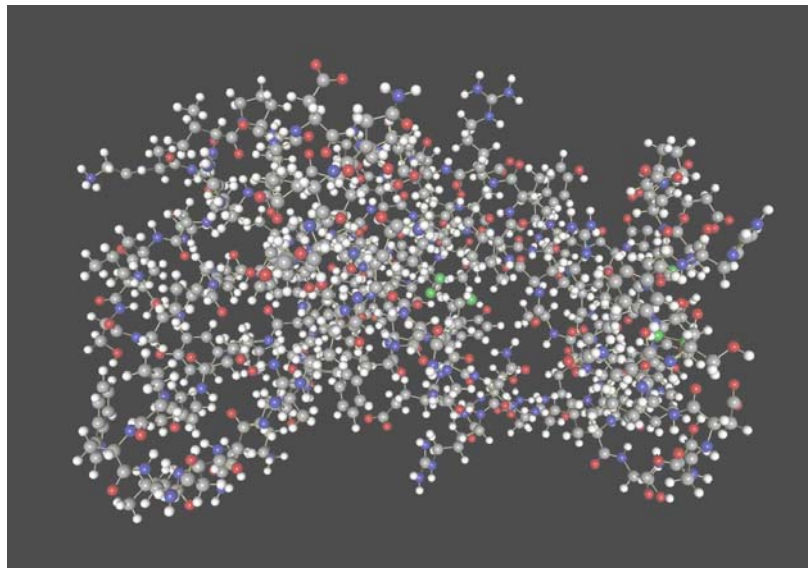
TABLE I

THE ELAPSED TIME FOR CHICKEN EGG WHITE CYSTATIN.

	Grid-enabled FMO	GAMESS-FMO
Initial Guess	37s	4s
Monomer	1h11m	59m
Dimer	2h16m	2h11m
Energy	4s	< 1s
Total	3h28m	3h10m

Xeon 3GHz 16CPUs in NAREGI Grid with Ninf-G2.40 on GT4.0.1

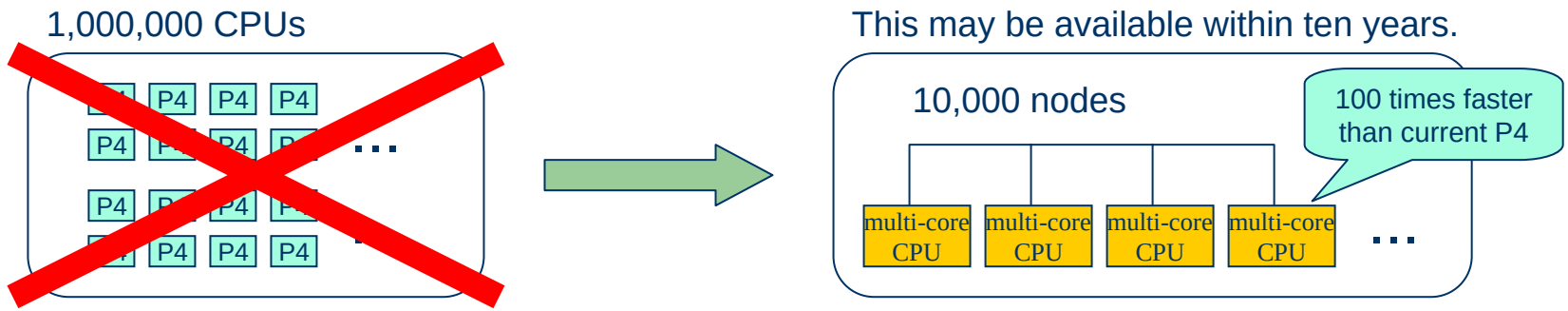
MO Basis set: 3-21G
20 monomer iterations until convergence



Chicken egg white cystatin: 1CEW
1701 atoms, 106 fragments

Performance Prediction of FMO (1)

- **The fictitious peta-scale computer**
  - Since the general purpose peta-performance computer is not available currently, we assume a fictitious architecture for the peta-scale computer from the following consideration.
  - If we configure 10peta-Flops peak computer from the current P4 (about 10 GFlops peak), 1,000,000 CPUs are necessary. This is not realistic since the current maximum scalability is the order of one thausand.
  - Then, the peta-scale computer will be constructed from some hierarchical structure. If we could use 100 times faster CPU-nodes, we can construct the peta-computer from 10,000 nodes



In the followings, we assume this hierarchical computer.

Performance Prediction of FMO (2)

- Phenomenological Model
  - The total amount of computation is represented by three terms of the function of N_f.

$$F_{\text{total}}(N_f) = F_m(N_f) + F_d(N_f) + F_{es}(N_f)$$

  - Each term is assumed as the following expressions.

$$\begin{aligned} F_m(N_f) &= \left[f_m^{(0)} + f_m^{(1)} N_f \right] N_f I_m \\ F_d(N_f) &= \left[f_d^{(0)} + f_d^{(1)} N_f \right] N_d(N_f) \\ F_{es}(N_f) &= f_{es}^{(0)} N_{es}(N_f), \end{aligned}$$

TABLE II
TIMING DATA OBTAINED IN A SINGLE NODE OF IBM P5 1.9GHZ.

Input		1cew	1ao6_half	1ao6	1ao6_dim
No. Atom		1701	9121	18242	36484
N_f		106	561	1122	2244
$N_d(N_f)$		690	4192	8416	16832
$N_{es}(N_f)$		4875	152888	620465	2499814
I_m		17	17	17	17
Time (sec)	monomer (Average)	1356 (0.752)	13364 (1.40)	40005 (2.10)	140810 (3.69)
	SCF-dimer (Average)	2037 (2.95)	20689 (4.94)	70465 (8.37)	186901 (11.1)
	ES-dimer (Average)	398 (0.0816)	13772 (0.0901)	55955 (0.0902)	208627 (0.0835)
Elapsed Time (sec)		3799	47886	166601	536898

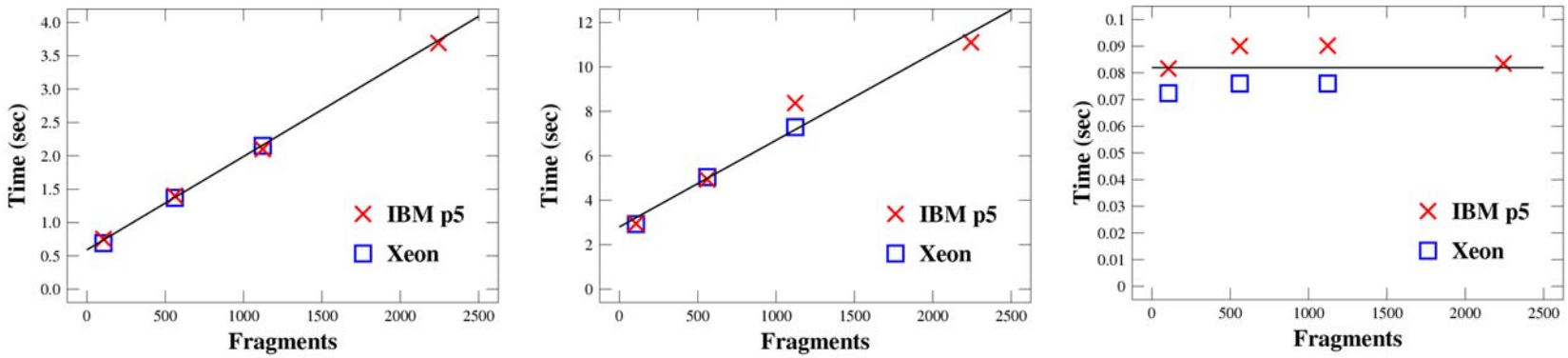
TABLE III
TIMING DATA OBTAINED IN 16 CPUS OF XEON 3GHZ.

Input		1cew	1ao6_half	1ao6
No. Atoms		1701	9121	18242
N_f		106	561	1122
$N_d(N_f)$		690	4192	8416
$N_{es}(N_f)$		4875	152888	620465
I_m		17	17	17
Time (sec)	monomer (Average)	1030.4 (9.72)	10808.9 (19.3)	33989.2 (30.3)
	SCF-dimer (Average)	1677.4 (41.3)	17517.6 (71.0)	50819.2 (102.7)
	ES-dimer (Average)	293.1 (1.02)	9594.8 (1.07)	39133.4 (1.07)
Elapsed Time (sec)		3003.5	38065.9	126330.9

These model functions are obtained under an assumption that the number of atoms in a fragment is not changed even for the peta-scale computer.

Performance Prediction of FMO (3)

- Least-square method to the model equations
  - Execution times for SCF calculations show linear dependence on the number of fragments, while non-SCF calculation of the fragment-pair is constant.
  - All the parameters are determined by these results.

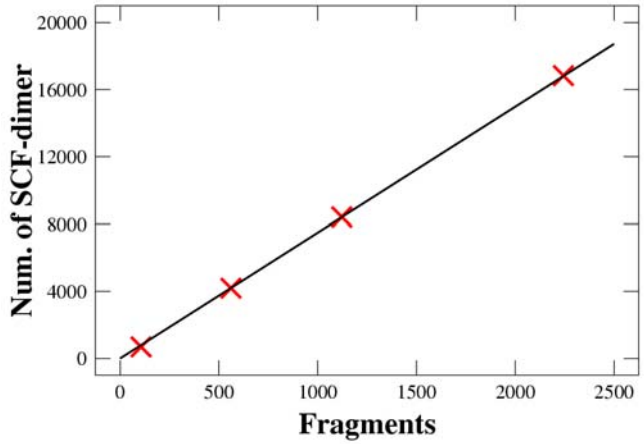


The fragment number dependence of an averaged calculation time of a fragment (left), a fragment-pair (center), and a fragment-pair under an electro-static approximation (right).

Performance Prediction of FMO (4)

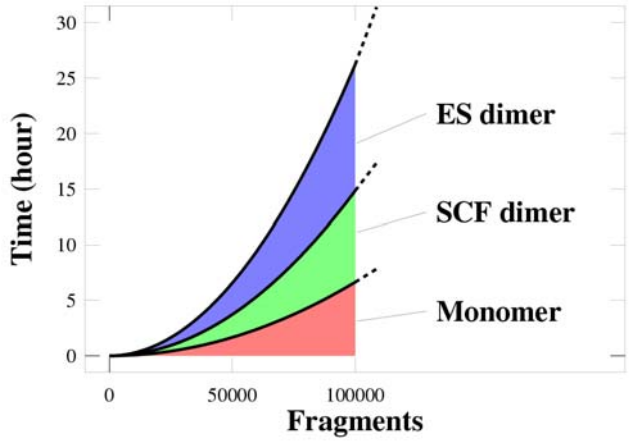
- The number of SCF-pairs
  - can also be determined by the test executions, and is linearly dependent on the number of fragments.
  - is finally represented in the function:

$$N_{es}(N_f) = \frac{(N_f - 1)N_f}{2} - 7.50\ N_f$$



The number of SCF fragment-pairs

- On the fictitious peta-scale computer (10,000 nodes of 100 core CPUs)
  - The total execution time is represented in a quadratic function of N_f .
  - The execution time can be estimated on the 10,000 nodes computer with a sufficient performance in each node.
  - All electron calculation of a molecule with 100,000 fragments (approx. 2,000,000 atoms) can be executed by FMO in a day.



Total execution time of FMO

OpenFMO Project (1)

It is revealed that FMO calculations can exhibit peta-scale performance in the next-generation supercomputer. However, it is also known that the present applications have significant problems in:

- the memory allocation
- communications between processes.

and may not be executed in the peta-scale machines.

Then, we began a project named OpenFMO. The main objective of this project is to construct the FMO program which can be executed in peta-scale computers.

OpenFMO Project (2)

This project stands for the following "Openesses":

A) ***Open Architecture Implementation of Skeleton and APIs***
(Dr. Inadomi, Dr. Maki, Dr. Honda)
- FMO is divided into two layers, the skeleton which control the whole flow, and the molecular orbital (MO) APIs to provide the charge distribution of each fragment. We first determine these spec of interfaces, and opened publicly.

B) ***Open Interface to Multi-physics Simulations***
(Dr. Kobayashi, T.T. (myself))
- FMO can also be opened to multi-physics simulations. Since it is based on electro-static interaction between fragments, each fragment can be substituted by the general object which can provide a static charge distribution.

C) ***Open Source License***
- The source code of the skeleton program of FMO is publicly opened according to some open-source license.

Currently, this project is supported by Kyushu University. And, the current head of the project is Professor M. Aoyagi. In the near future, the project management itself may be opened to public user and developer.

OpenFMO Project (3)

- Future Schedule of The Project:
  - within this year
    - A skeleton program based on the parallelization scheme of GAMESS-FMO (already done by Dr. Maki).
    - Interfaces of MO-APIs are fixed (by Dr. Inadomi and Dr. Maki).
    - Web pages of OpenFMO (will be done by T.T. (myself))
  - The first half of 2007
    - The new style skeleton which can be executed on the peta-scale resources (this will be done by Dr. Inadomi).
    - beta codes of all APIs are released.
    - The multi-physics applications (Dr. Kobayashi, and me)
  - The latter half of 2007
    - beta release of the multi-physics application

Please visit the OpenFMO web-site:
http://www.OpenFMO.org/

Summary

- In accordance with the trend from Grid to Peta, we showed the recent progresses on the FMO method.
  - Loosely-coupled FMO on NAREGI testbeds
  - Performance prediction of FMO
  - OpenFMO Project
- On the multi-physics/multi-scale problems, we study various kinds of systems including quantum mechanics, classical statistical mechanics, bio-molecular simulations, coarce-grained dynamics of cells, etc. (Please refer our recent work below)

T. Takami, J. Maki, J. Ooba, T. Kobayashi, R. Nogita, and M. Aoyagi, "*Interaction and Localization of One-electron Orbitals in an Organic Molecule: the Fictitious Parameter Analysis for Multi-physics Simulations*," J. Phys. Soc. Jpn. **76**, 013001 (2007)