Development of Massively Parallel Quantum Chemistry Calculation Program SMASH and its Applications

Kazuya Ishimura
(Institute for Molecular Science)

34th Computational Materials Design(CMD) Workshop
02/21/2019

Outline

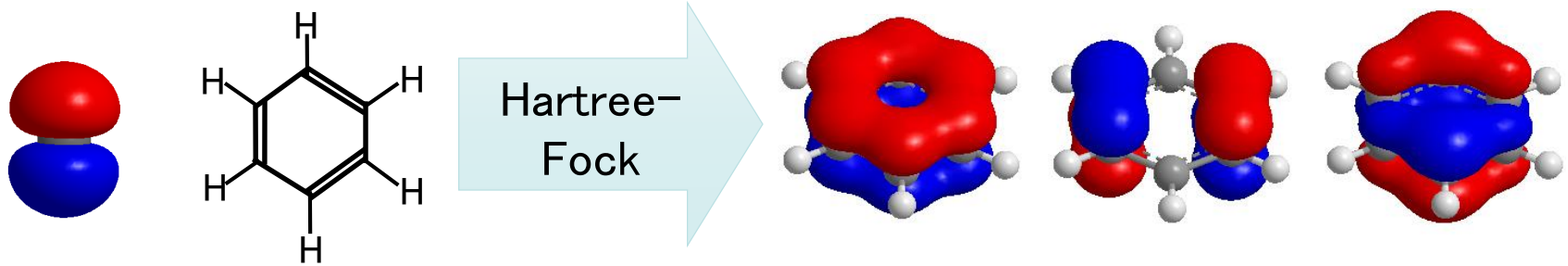
1. Quantum Chemistry Calculation
2. Background of New Program Development
3. SMASH: Massively Parallel Quantum Chemistry Calculation Program
4. Applications
5. Further Improvement for Post-K Computer
6. Summary

Quantum chemistry calculations

- Quantum chemistry calculation:
  - Compute molecular electronic structures
  - Analyze molecular geometries, properties, chemical reactions, and so on
- Input : Atomic electron configurations (atomic orbitals), Atomic coordinates
- Output : Molecular electron configurations (molecular orbitals), Energies
- The computational cost of standard methods increases in proportional to the third to seventh power of the number of atoms (or Atomic orbitals)

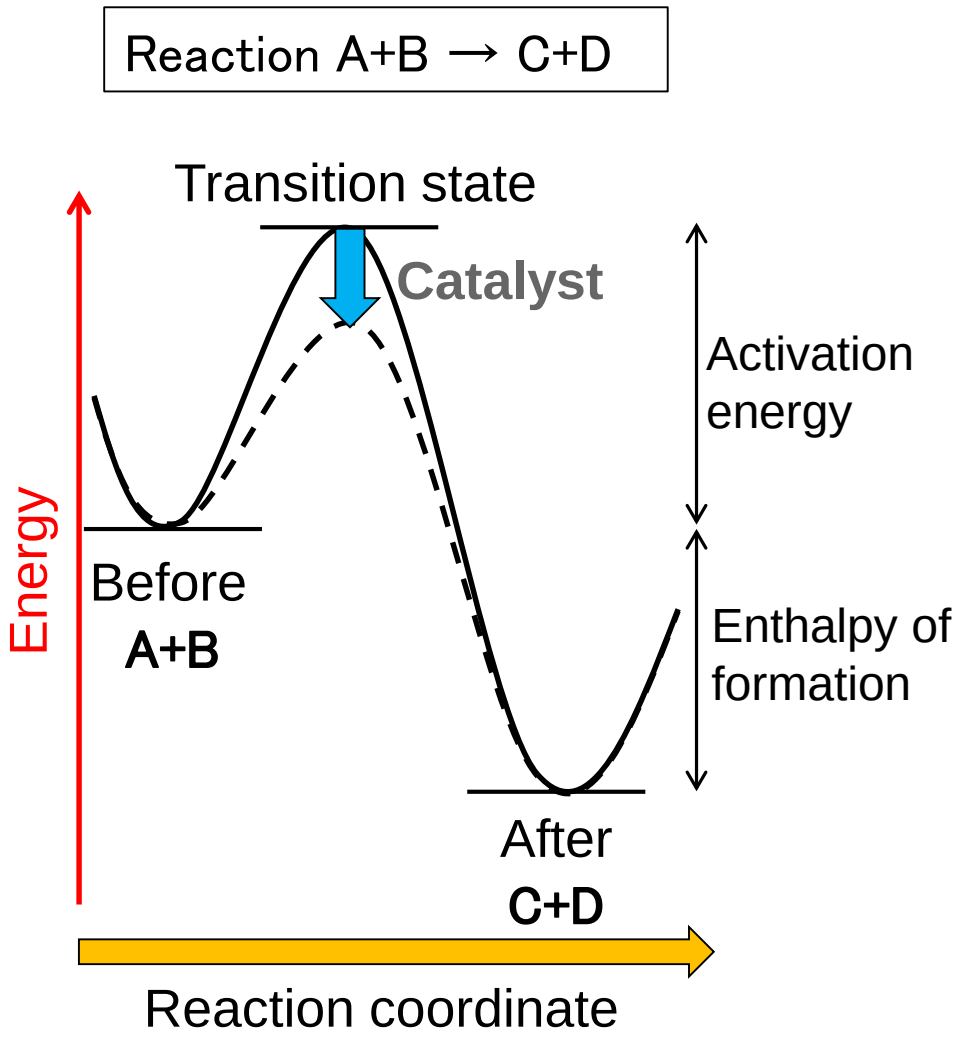
Atomic orbitals (C,H)
Atomic coordinates

Molecular orbitals
(Benzene(C_6H_6))

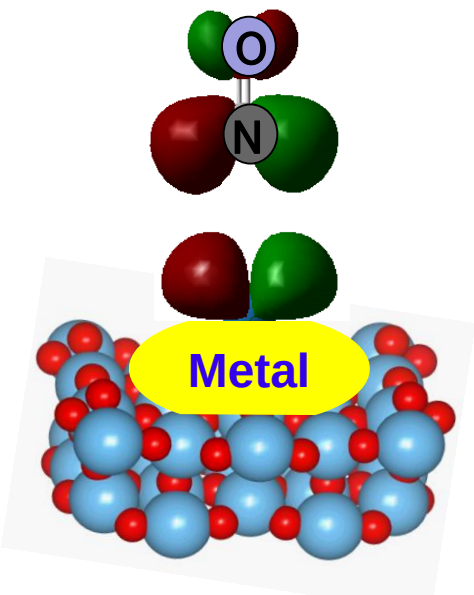


What are obtained from calculations?

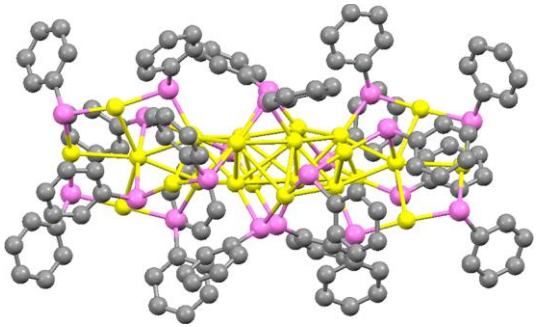
- Molecular energy
- Stable molecular structure
- Transition state structure
- Enthalpy of formation
- Photoelectron spectroscopy
- Vibrational frequency
- NMR spectroscopy
- Atomic charge
- Bond analysis

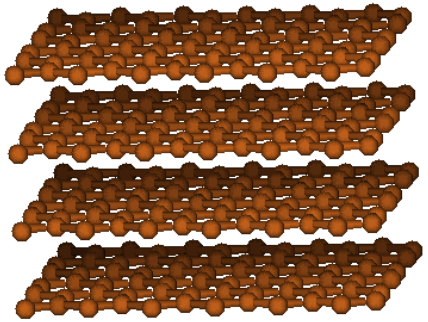


Calculation examples

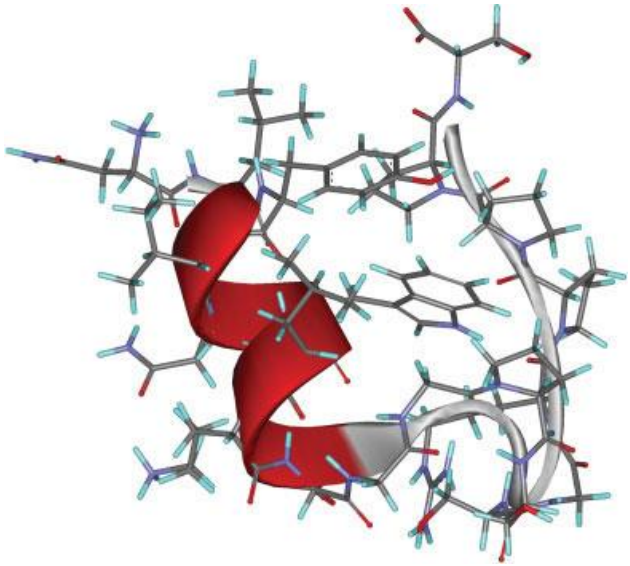


More reactive metal elements are searched by analyzing the interaction of molecules and metals.





Chemical reactions and stable geometries of nano-sized molecules and clusters can be treated.



It is now possible to calculate electronic structures of proteins by the development of new theories and parallel algorithms.

Computational methods

N: Number of atoms or atomic orbitals

Method	Operation	Data	Communication
Hartree-Fock, Density functional Theory(DFT) 2-electron integral terms (on cache) Diagonalization of dense matrix	$O(N^4)$ ($O(N^{2-3})$ by cutoff)	$O(N^2)$	$O(N^2)$
Perturbation(MP2,MP3,..) **Coupled Cluster(CCSD, CCSD(T),..)** Dense matrix-matrix multiplication	$O(N^5\sim)$	$O(N^4\sim)$	$O(N^4\sim)$
Configuration Interaction (CIS, CISD,...) Diagonalization of sparse matrix	$O(N^5\sim)$	$O(N^4\sim)$	$O(N^4\sim)$

SMASH program

SMASH (Scalable Molecular Analysis Solver for High-performance computing systems)

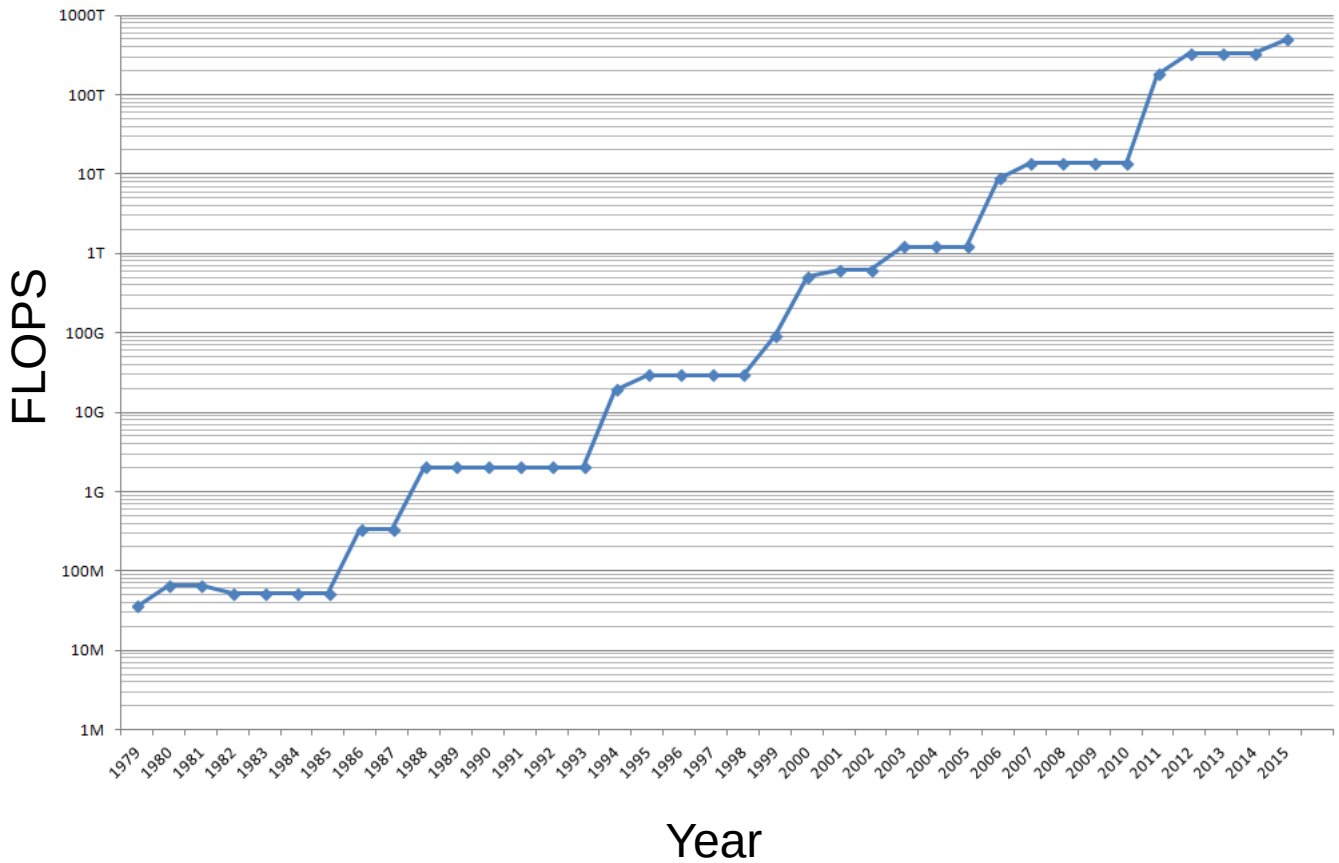
- License: Apache 2.0 (Open source)
- Released on September 1, 2014
- http://smash-qc.sourceforge.net/
- Language: Fortran90/95
- Parallelization : MPI and OpenMP
- Target machines: PC clusters to supercomputers (Scalar CPUs)
- Available calculations: Hartree-Fock, DFT, and MP2 energy and energy gradient
- Modularization of frequently used routines such as 1,2-electron integral calculations

Background of SMASH development 1

History of Computer Center in Institute for Molecular Science

https://ccportal.ims.ac.jp/

- ✓ Updated periodically
- ✓ Improved hundreds of times every 10 years



Year	Performance (TFLOPS)
1979	3.6×10^{-5}
1989	0.002
1999	0.092
2009	13
2015	492
2017	4,076

Background of SMASH development 2

History of Computer Center in Institute for Molecular Science

Year	Machine	Peek performance (TFLOPS)
1979	HITACHI M-180 (2 machines)	3.6×10^{-5}
2000	IBM SP2 (Wide24 machines)	0.007
	IBM SP2 (Thin24 machines)	0.003
	NEC SX-5 (8CPUs)	0.064
	Fujitsu VPP5000 (30PEs)	0.288
	SGI SGI2800 (256CPUs)	0.153
	Total	0.514
2017	NEC LX (Xeon 20cores x2, 2.4GHz) (32560 cores)	2,500
	NEC LX (Xeon 18cores x2, 3.0GHz) (5724 cores)	549
	NEC LX (Xeon 12cores x2, 3.0GHz) (1536 cores)	221
	(+ NVIDIA Tesla P100 x 2)	806
	Total	4,076

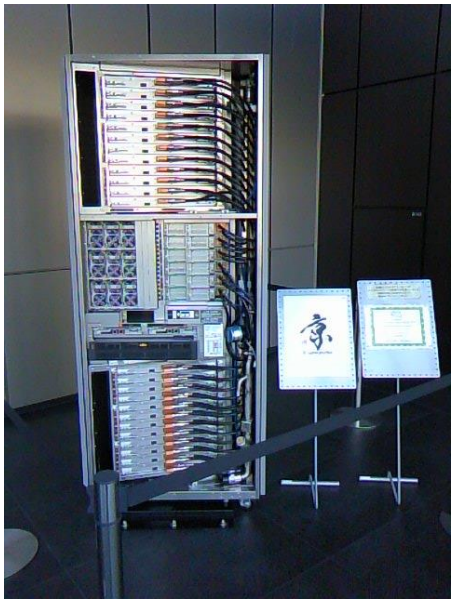
←2 PCs (Core i7 7700K, 4.2GHz,4cores)

← Half of K Computer

✓ Parallelization is necessary to use present and future (super)computers.

Background of SMASH development 3

K Computer (2012-)



- 88128 nodes
- 705024 CPU cores
- 8 CPU cores/node
- 16GB memory/node
- 3 dimensional torus network

Background of SMASH development 4

Development of MPI/OpenMP parallel program for Hartree-Fock and DFT calculations

- Started in 2008
- Based on GAMESS program (originally parallelized by MPI)
- GAMESS code was modified to introduce OpenMP parallelization
- Benchmark calculations using Cray XT5 2048 CPU cores (8cores/node)

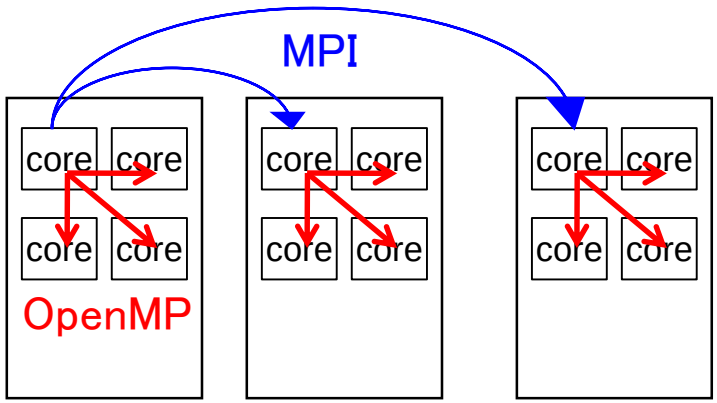
GAMESS

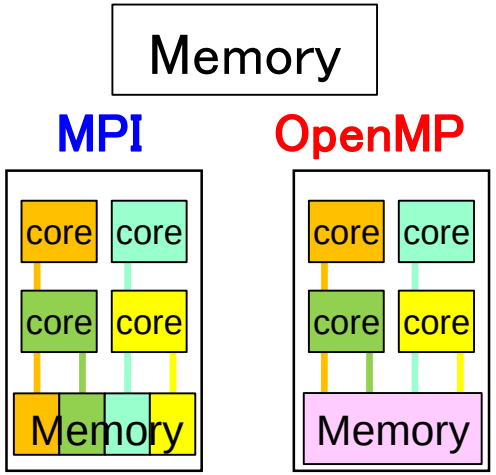
- To obtain good parallel efficiency and computational time, the source code was drastically changed.

  → Only Hartree-Fock and DFT calculations could be performed with the modified GAMESS program.

MPI+OpenMP parallelization

MPI for inter-node、OpenMP for intra-node





Advantage

- High parallel efficiency
  - dynamical load-balancing by OpenMP for intranode
  - Reduction of network communication
- Efficient memory use
  - To share memory space by OpenMP

Disadvantage

- Complicated algorithm and program
- High cost of program development

Hartree-Fock calculation

$$\mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC}$$

F: Fock matrix, **C**: Molecular orbital coefficient
S: Overlap matrix, **ε**: Molecular orbital energy

Fock matrix

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{i,\lambda,\sigma} C_{\lambda i} C_{\sigma i} \underbrace{\{2(\mu\nu\,|\,\lambda\sigma) - (\mu\lambda\,|\,\nu\sigma)\}}_{\text{2-electron integral terms}}$$

$$(\mu\nu|\lambda\sigma) = \int d\boldsymbol{r}_1 \int d\boldsymbol{r}_2 \phi_\mu(\boldsymbol{r}_1)\phi_\nu(\boldsymbol{r}_1)\frac{1}{r_{12}}\phi_\lambda(\boldsymbol{r}_2)\phi_\sigma(\boldsymbol{r}_2)$$

$\phi_\mu(\boldsymbol{r}_1)$: Atomic orbital
Gaussian function

Calculation of initial coefficient **C**

↓

Calculation of 2-electron integral terms +
Addition them to Fock matrix

↓

Diagonalization of Fock matrix

Not converged

↓ Converged

Finish!

MPI/OpenMP parallel algorithm for Fock assembly

K. Ishimura, K. Kuramoto, Y. Ikuta, S. Hyodo, J. Chem. Theory Comp. 2010, 6, 1075.

Fock matrix $F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda,\sigma} D_{\lambda\sigma} \{2(\mu\nu|\lambda\sigma) - (\mu\lambda|\nu\sigma)\}$

2-electron integrals

- ✓ Distribute variables of the outermost loop by OpenMP to reduce overhead
- ✓ Classify all variables in the OpenMP region into private or shared ones

Before parallelization

```
do μ=1, nbasis
 do ν=1, μ
  do λ=1, μ
   do σ=1, λ
    Calculate (μν|λσ) + add
     them to Fock matrix
   enddo
  enddo
 enddo
enddo
```

After parallelization

```
!$OMP parallel do schedule(dynamic,1)
!$OMP private(μν, λstart, …) reduction(+:Fock)
do μ=nbasis, 1, -1                    <----- OpenMP
   do ν=1, μ
     μν=μ*(μ+1)/2+ν
      λstart=mod(μν+mpi_rank,nproc)+1
      do λ=λstart, μ ,nproc           <----- MPI
         do σ=1, λ
            Calculate (μν|λσ) + add them to Fock matrix
         enddo
      enddo
   enddo
enddo
call mpi_allreduce(Fock)
```

Improvement of initial guess calculation

✓ Acceleration and thread (OpenMP) parallelization using BLAS library

- Orbital projection calculation

$$\mathbf{C}_1 = \mathbf{S}_{11}^{-1}\mathbf{S}_{12}\mathbf{C}_2\left[\mathbf{C}_2^t\mathbf{S}_{12}^t\mathbf{S}_{11}^{-1}\mathbf{S}_{12}\mathbf{C}_2\right]^{-1/2}$$

$\mathbf{C}_1$: Initial orbital coefficients
$\mathbf{C}_2$: Coefficients of extended Hückel calculation
$\mathbf{S}_{11}$: Overlap integrals
$\mathbf{S}_{12}$: Overlap integrals between bases for extended Hückel and SCF calculations

D. Cremer and J. Gauss, *J. Comput. Chem.* **7** 274 (1986).
Used in Parallel Quantum Solution (PQS) program

Computational conditions

Quantum chemistry program: GAMESS

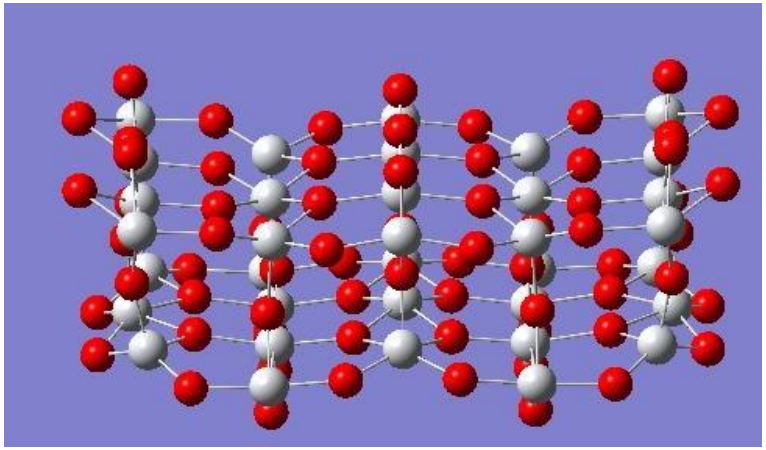
Machine: Cray XT5 2048 cores
(Opteron 2.4GHz, 8cores/node)

Complier: PGI fortran compiler-8.0.2

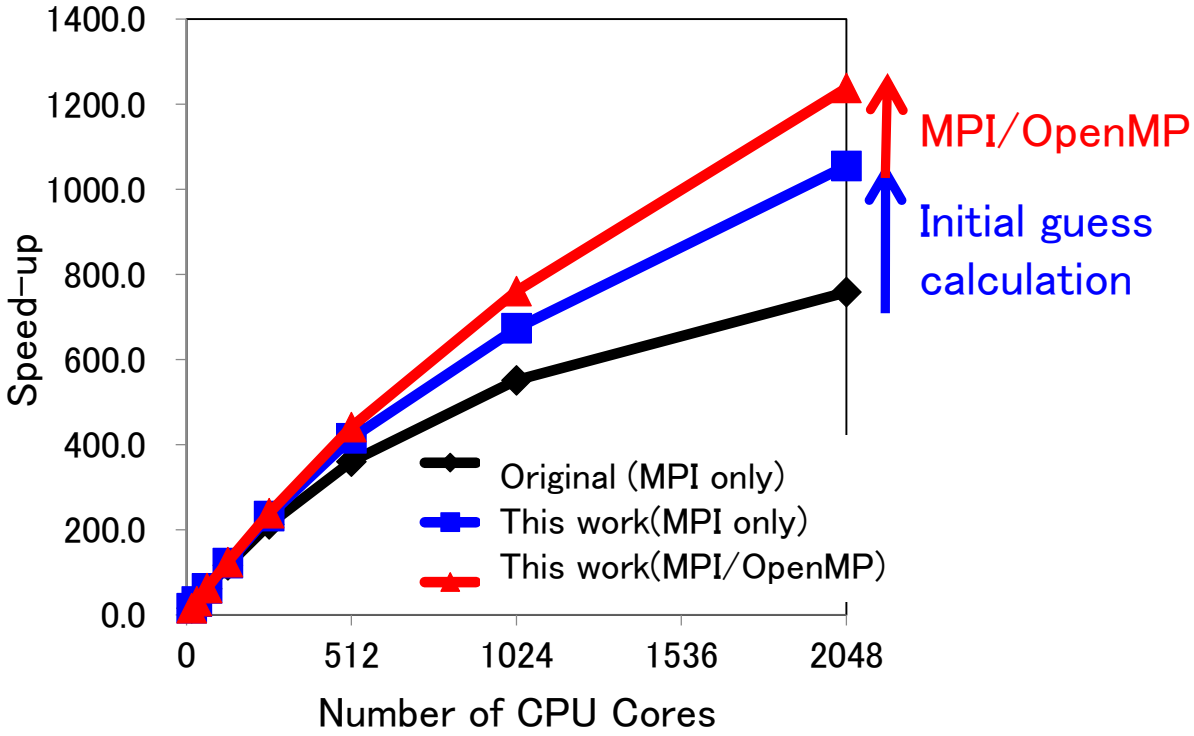
BLAS,LAPACK libraries: XT-Libsci-10.3.3.5

MPI library: xt-mpt-3.1.2.1 (MPICH2-1.0.6p1)

Molecule: TiO_2 cluster($Ti_{35}O_{70}$)
(6-31G basis, 1645 functions)



Speed-up (TiO_2 cluster)



Total elapsed times (sec) and speed-ups in parentheses.

Number of CPU cores		64	256	1024	**2048**
Original GAMESS	MPI only	4739.7 (61.4)	1368.6 (212.5)	527.6 (551.2)	383.5 **(758.3)**
This work	MPI only	4636.6 (62.3)	1241.2 (232.6)	428.7 (673.5)	273.7 **(1054.9)**
This work	MPI/OpenMP	4575.3 (63.4)	1214.6 (238.7)	381.1 (760.8)	**234.2 (1238.0)**

Analysis of TiO_2 cluster calculation

- ✓ Speed-up of Fock Assembly (most time consuming step) is improved by introducing OpenMP.
- ✓ Initial guess cost becomes 40% of total time in MPI only on 2048 cores.

Elapsed times (sec) and speed-ups (in parentheses) of Fock Assembly.

Number of CPU cores		64	256	1024	**2048**
Original	MPI only	4528.7 (63.2)	1175.2 (243.5)	334.0 (856.6)	188.6 **(1517.0)**
This work	MPI only	4570.1 (62.9)	1175.2 (244.4)	360.0 (797.9)	203.1 **(1414.4)**
This work	MPI/OpenMP	4475.8 (63.6)	1150.4 (247.3)	316.4 (899.0)	174.8 **(1627.2)**

Elapsed times (sec) and fractions (in parentheses) of initial guess calculation.

Number of CPU cores		64	256	1024	**2048**
Original	MPI only	148.0 (3.1%)	143.6 (10.5%)	143.6 (27.2%)	143.8 **(37.5%)**
This work	MPI only	18.1 (0.4%)	18.6 (1.5%)	18.9 (4.4%)	19.2 **(7.0%)**
This work	MPI/OpenMP	14.0 (0.3%)	13.2 (1.1%)	13.6 (3.6%)	13.8 **(5.9%)**

Summary of OpenMP introduction

- All variables in OpenMP regions are classified into private or shared ones.
- Many common variables are changed to arguments.

- Only Hartree-Fock and DFT calculations can be performed with the modified GAMESS program.

- A new program with MPI and OpenMP parallelization is necessary to calculate nano-sized molecules.
- OpenMP parallelization should be considered from the beginning of the development.

SMASH program

SMASH (Scalable Molecular Analysis Solver for High-performance computing systems)

- License: Apache 2.0 (Open source)
- Released on September 1, 2014
- http://smash-qc.sourceforge.net/
- Language: Fortran90/95
- Parallelization : MPI and OpenMP
- Target machines: PC clusters to supercomputers (Scalar CPUs)
- Available calculations: Hartree-Fock, DFT, and MP2 energy and energy gradient
- Modularization of frequently used routines such as 1,2-electron integral calculations

SMASH website

http://smash-qc.sourceforge.net/

SMASH download | Sour

https://sourceforge.net/projects/smash-qc/?source=navbar

SOURCEFORGE

SMASH

Massively parallel software for quantum chemistry

Brought to you by: ishimura-smash

Add a Review

Downloads: 2

Download | Get Updates | Share This

Summary | Files | Reviews

Scalable Molecular Analysis Solver for High-performance computing
chemistry calculations. It is currently capable of energy, analytical gr
Density Functional Theory (DFT), and energy calculations of Second-
performed on 100,000 CPU cores of K Computer with high parallel e

Project Activity

Released /smash-2.2.0.tgz

Released /SMASH_user_manual_JP-2.2.0.pdf

Released /smash-2.1.0.tgz

Released /SMASH_user_manual_JP-2.1.0.pdf

Released /smash-2.1.0.tgz

SMASH

smash-qc.sourceforge.net

SMASH

Open-source software for massively parallel quantum chemistry calculations

Download Site SourceForge | Download Latest ver. | 日本語 マニュアル

Hosted on GitHub Pages — Theme by orderedlist

Welcome to SMASH Page

Scalable Molecular Analysis Solver for High-performance computing systems (SMASH) is open-source software for massively parallel quantum chemistry calculations written in the Fortran 90/95 language with MPI and OpenMP. Hartree-Fock and Density Functional Theory (DFT) calculations can be performed on 100,000 CPU cores of K Computer with high parallel efficiency.

What's New?

April 22, 2017

SMASH-2.2.0 released.

* Supported h and i basis functions (except for ECP calculations).

September 5, 2016

SMASH-2.1.0 released.

* Fixed an issue where 7f and 9g (spherical harmonics) integral values are incorrect.
* Changed to write a checkpoint file at each geometry optimization cycle.

July 4, 2016

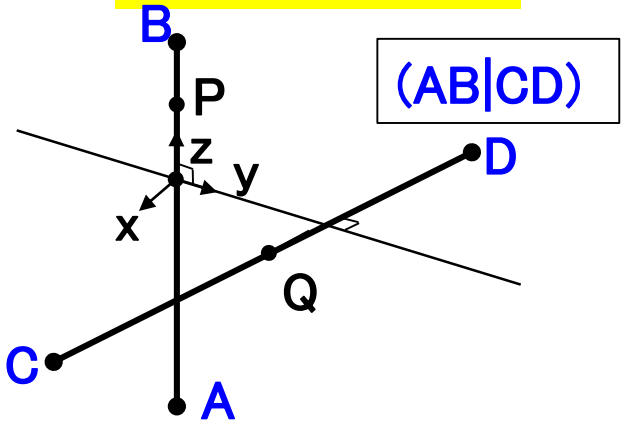
SMASH-2.0.0 released.

* Added closed-shell MP2 energy gradient and geometry optimization calculations.
* Changed B3LYP parameters from VWN5 to VWN3. (job method=B3LYP for VWN3, job method B3LYP5 for VWN5)
* Added an keyword to control computational precision such as cutoffs and the number of grid points for DFT. (control precision=high, medium (default), low)
* Changed default cutoff values. e.g.) Cutint2 is changed from 1.0D-12 to 1.0D-11.

Latest source code and Japanese manual

Program development

Acceleration



2-electron integral calculation
$(ss|ss)^{(m)} \rightarrow (r) \rightarrow (p|q) \rightarrow (AB|CD)$

Rotate axes → Use recurrence relations → Rotate axes to the original position

This algorithm is used in GAMESS.

K. Ishimura, S. Nagase, Theoret Chem Acc, 2008, 120, 185.

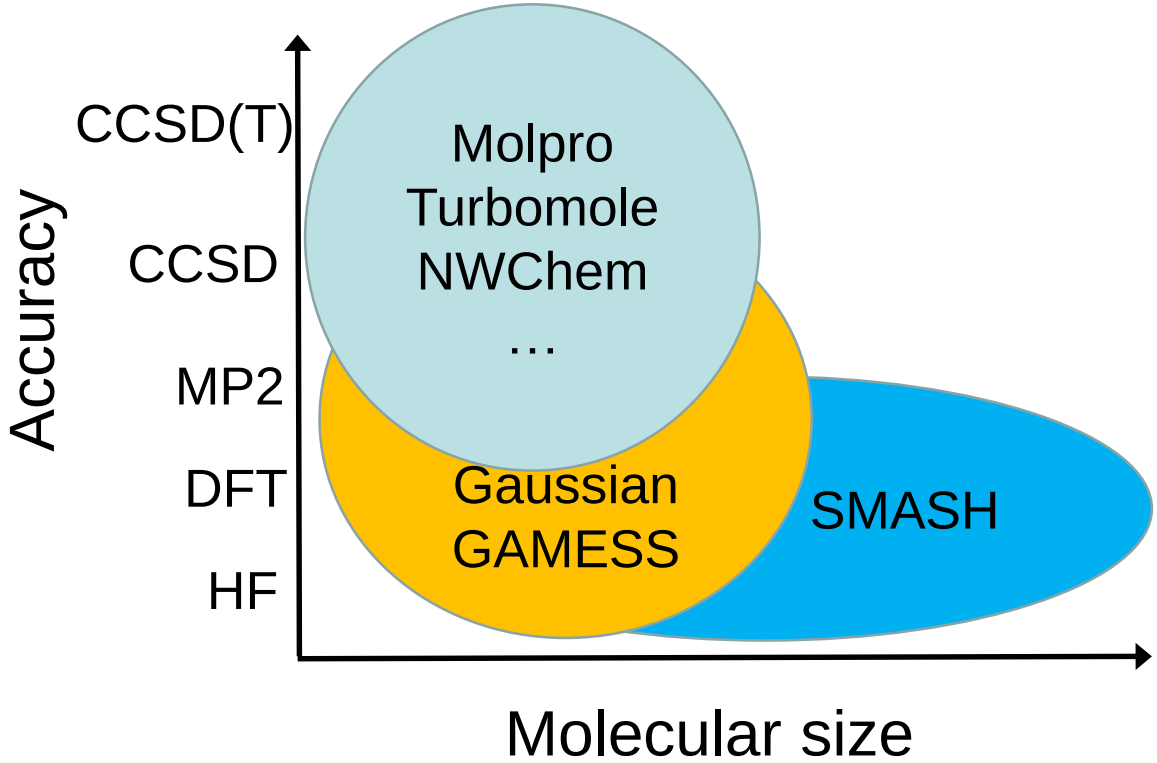
Parallelization

```
!$OMP parallel do schedule(dynamic,1)
do μ=n, 1, -1     <--- OpenMP
   do ν=1, μ
     μν=μ*(μ+1)/2+ν
      λstart=mod(μν+mpi_rank,nproc)+1
      do λ=λstart, μ ,nproc    <--- MPI rank
         do σ=1, λ
              Calculate (μν|λσ ) and add them
              to Fock matrix
          enddo
      enddo
   enddo
enddo
call mpi_allreduce(Fock)
```

Hybrid MPI/OpenMP parallelization

K. Ishimura, K. Kuramoto, Y. Ikuta, S. Hyodo, J. Chem. Theory Comp. 2010, 6, 1075.

Quantum chemistry programs



Target of SMASH: Large molecules with moderate accuracy
Keyword of SMASH: Simple (License, Input, Execution, Development)

Input format

- Set computational methods and conditions in job, control, scf, opt, dft, mp2 sections.
- Molecular coordinates starts after "geom" line and ends at the end of the file or blank line.
- Sample input and output files are in smash/example/
- Basis function and ECPs for each element starts after "basis" and "ecp" lines.
- There is no distinction between uppercase and lowercase letters.

B3LYP/6-31G(d) geometry optimization of H_2O

```
job  runtype=optimize method=b3lyp basis=6-31g(d)
geom
 O   0.000    0.000    0.142
 H   0.000    0.756   -0.462
 H   0.000   -0.756   -0.462
```

How to execute SMASH program

- MPI process (internode) parallelization

```
mpirun -np (# of processes) ./bin/smash < (input file) > (output file)
```

- OpenMP (intranode) parallelization
  - Bash: Add the following lines in your ~/.bashrc file,

```
export OMP_NUM_THREADS=(# of threads)
ulimit -s unlimited
export OMP_STACKSIZE=1G
```

and execute the following command.

```
source ~/.bashrc
```

  - Tcsh: Add the following lines in your ~/.tcshrc file,

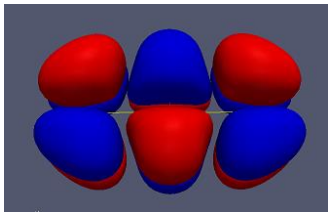
```
setenv OMP_NUM_THREADS (# of threads)
unlimit
setenv OMP_STACKSIZE 1G
```

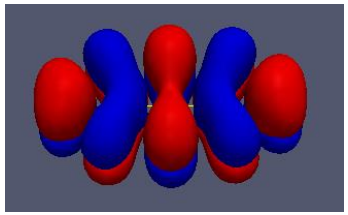
and execute the following command.

```
source ~/.tcshrc
```

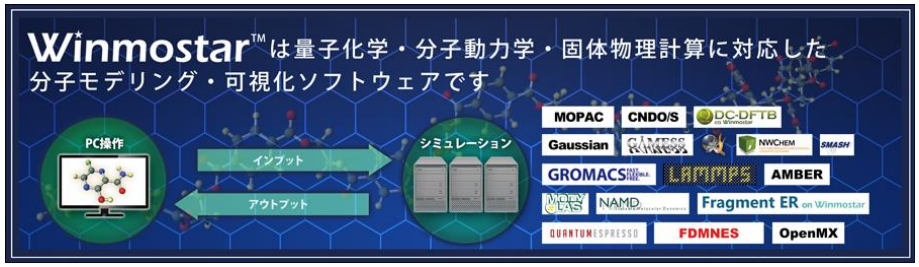
Visualization of calculation results

- Visualization
  - In SMASH version 1, a program to generate vtk files is available. MOs are shown using the free visualization software Paraview.





  - In version 2, a program to generate cube files is available. MOs are shown by GaussView, and so on.
  - The support of modelling and visualization for SMASH by Winmostar software (X-ability Co., Ltd.) is in progress.



Interface of 2-electron integral routine

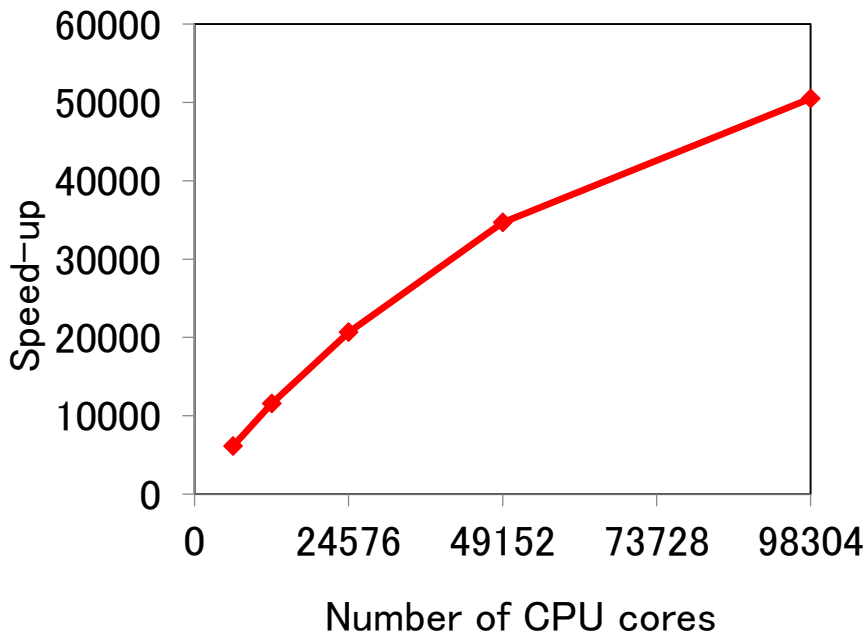
- ✓ Data are accessed by only arguments.
- ✓ Common and module variables are not used.
- ✓ This routine is introduced into other quantum chemistry programs.

subroutine calc2elec(twoeri, exijkl, coijkl, xyzijkl, nprimijkl, nangijkl, nbfijkl, maxdim, mxprsh, threshex)

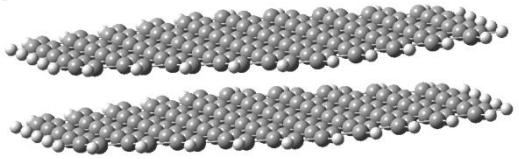
twoeri : Two-electron integrals (Output)
exijkl : Exponents of basis functions (Input)
coijkl : Coefficients of basis functions
xyzijkl : XYZ components of four centers
nprimijkl :Numbers of primitive basis functions
nangijkl : Degrees of angular momenta of basis functions (s=0, p=1, d=2,...)
nbfijkl : Numbers of basis functions of shells (s=1, p=3, d=5 or 6, ...)
maxdim : Maximum number of integral array dimension for twoeri
mxprsh : Maximum number of primitive basis functions for exijkl and coijkl
threshex :Threshold of exponential function ($\exp(-x^2)$)

Parallel performance (DFT energy)

- ✓ 50,000 speed-up on 100,000 cores.
- ✓ Time for 3 diagonalizations is 35 sec on any cores.



Computer : K Computer (2.0GHz, 8cores/node)
Molecule : $(C_{150}H_{30})_2$
Basis set : cc-pVDZ (4500 functions)
Method : B3LYP
Number of SCF cycles : 16
Diagonalization: LAPACK(Divide and Conquer)



Number of cores	6144	12288	24576	49152	**98304**
Elapsed time (sec)	1267.4	674.5	377.0	224.6	**154.2**

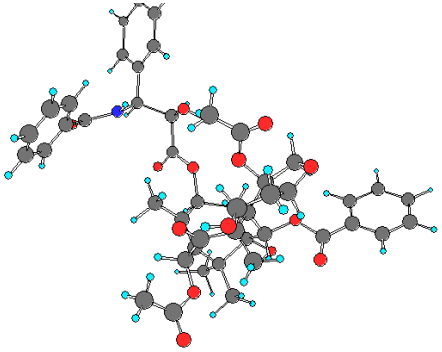
Serial performance (Hartree-Fock energy)

✓ Faster than GAMESS on 1 node.

Comparison with GAMESS

- Xeon E5649 2.53GHz 12core, 1 node
- Hartree-fock calculation time of Taxol($C_{47}H_{51}NO_{14}$) (sec)
- Same conditions(Cutoffs)

Basis set	GAMESS	SMASH
6-31G(d) (1032 functions)	706.4	666.6
cc-pVDZ (1185 functions)	2279.9	1434.3

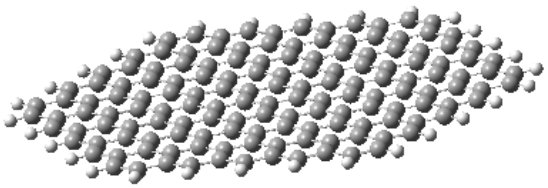


Taxol ($C_{47}H_{51}NO_{14}$)

Parallel performance (DFT energy gradient)

✓ Almost linear speed-up because of no diagonalization in gradient calculation.

Computer : K Computer (2.0GHz, 8cores/node)
Molecule : ($C_{150}H_{30}$)
Basis set : cc-pVDZ (2250 functions)



Computational time for B3LYP energy gradient (sec) and speed-up.

Number of Cores	1024	4096	8192	16384
1-electron terms	60.0 (1024.0)	15.4 (3989.6)	7.9 (7777.2)	4.2 (14628.6)
2-electron terms	342.0 (1024.0)	85.8 (4081.7)	42.9 (8163.4)	21.3 (16441.7)
Total	402.0 (1024.0)	101.2 (4067.7)	50.8 (8103.3)	25.5 (16143.1)

Second-order perturbation (MP2) method

$$E_{MP2} = \sum_{ij}^{occ} \sum_{ab}^{vir} \frac{(ai|bj)\{2(ai|bj) - (aj|bi)\}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

$$\underline{(ai|bj)} = \sum_{\mu\nu\lambda\sigma}^{AO} \underline{C_{\mu a} C_{\nu i} C_{\lambda b} C_{\sigma j}} \underline{(\mu\nu|\lambda\sigma)}$$

Molecular orbital 2-electron term

Molecular orbital coefficient

Atomic orbital 2-electron term

ε_i: Molecular orbital energy
$C_{\mu a}$: Molecular orbital coefficient
i, j: Occupied orbitals
a, b: Unoccupied orbitals

Hartree-Fock calculation

↓

$(\mu\nu|\lambda\sigma)$ (O(N^4))

↓

$(\mu i|\lambda\sigma)$ (O(N^5))

↓

$(\mu i|\lambda j)$ (O(N^5))

↓

$(ai|\lambda j)$ (O(N^5))

↓

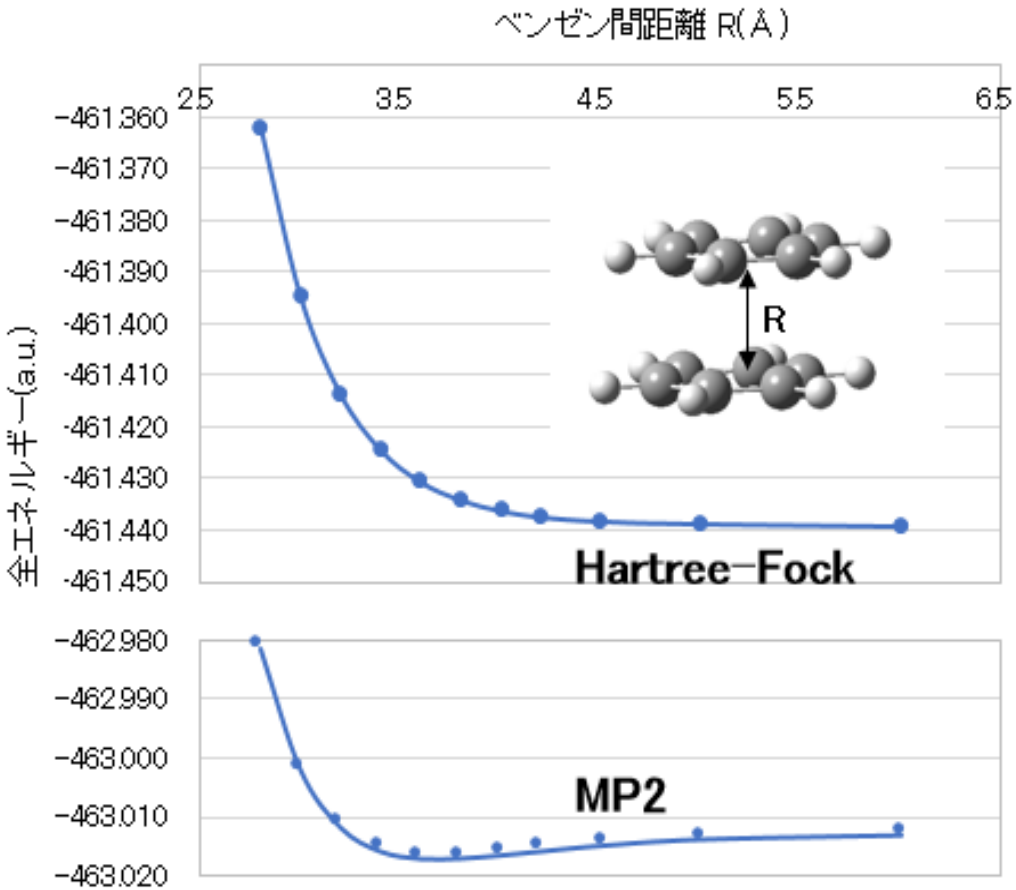
$(ai|bj)$ (O(N^5))

↓

MP2 energy calculation (O(N^4))

Second-order perturbation (MP2) method

- MP2 method can treat non-covalent (dispersion, $\pi-\pi$) interactions which can not describe by Hartree-Fock method.



MP2 energy calculation algorithm

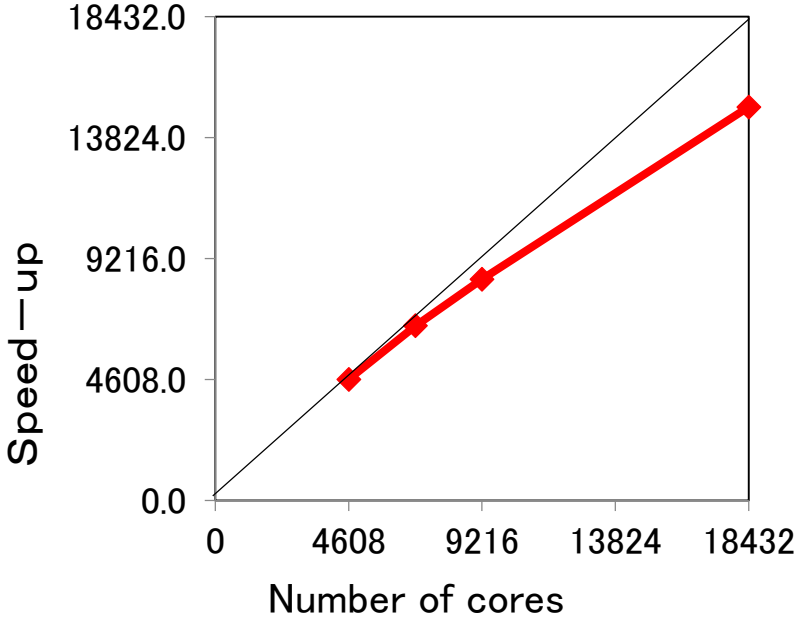
- ✓ All steps are parallelized by MPI and OpenMP.
- ✓ Total amount of network communication is constant on any processes.

$$E_{MP2} = \sum_{ij}^{occ} \sum_{ab}^{vir} \frac{(ai|bj)\{2(ai|bj) - (aj|bi)\}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \qquad (ai|bj) = \sum_{\mu\nu\lambda\sigma}^{AO} C_{\mu a} C_{\nu i} C_{\lambda b} C_{\sigma j} (\mu\nu|\lambda\sigma)$$

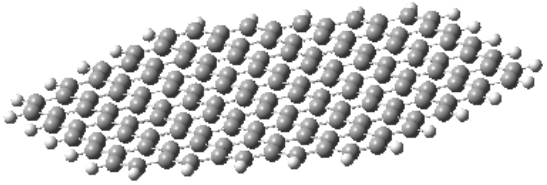
```
do μλ (AO index pair)     Distributed using MPI ranks
!$OMP parallel do schedule(dynamic,1)
  do σ
    two-electron term calculation (μν|λσ)   (all ν)
    1st transformation             (μi|λσ)   (all i)
  enddo
!$OMP end parallel do
  2nd transformation(dgemm)        (μi|λj)   (i≧j)
end do μλ
do ij (MO index pair)     Distributed using MPI ranks
  call MPI_sendrecv (μi|λj)
  3rd transformation(dgemm)        (μi|bj)   (all b)
  4th transformation(dgemm)        (ai|bj)   (all a)
  MP2 energy calculation
end do ij
call mpi_reduce(MP2 energy)
```

Parallel performance (MP2 energy)

✓ Good parallel efficiency on more than 10,000 cores.



Computer : K Computer (2.0GHz, 8cores/node)
Molecule : ($C_{150}H_{30}$)
Basis set : 6-31G(d) (2160 functions)
Method : MP2



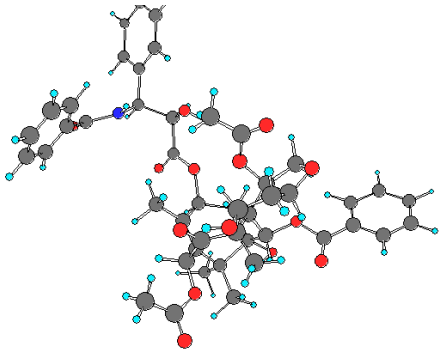
Number of cores	4608	6912	9216	18432
Elapsed time(sec)	152.5	105.7	83.4	46.9
Speed-up	4608.0	6648.2	8425.9	14983.4

Geometry optimization performance

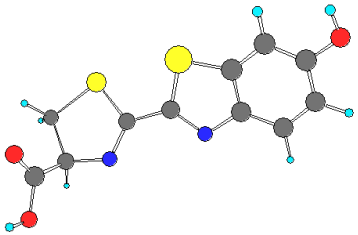
- ✓ The number of geometry optimization cycles is significantly reduced by the introduction of the redundant coordinate system which uses molecular bonds, angles, and torsions.
- ✓ Initial Hessian is calculated using force parameters.

Number of Optimization cycles at B3LYP/cc-pVDZ level
(Initial Geometry: HF/STO-3G)

	Cartesian	Redundant
Luciferin($C_{11}H_8N_2O_3S_2$)	63	11
Taxol ($C_{47}H_{51}NO_{14}$)	203	40

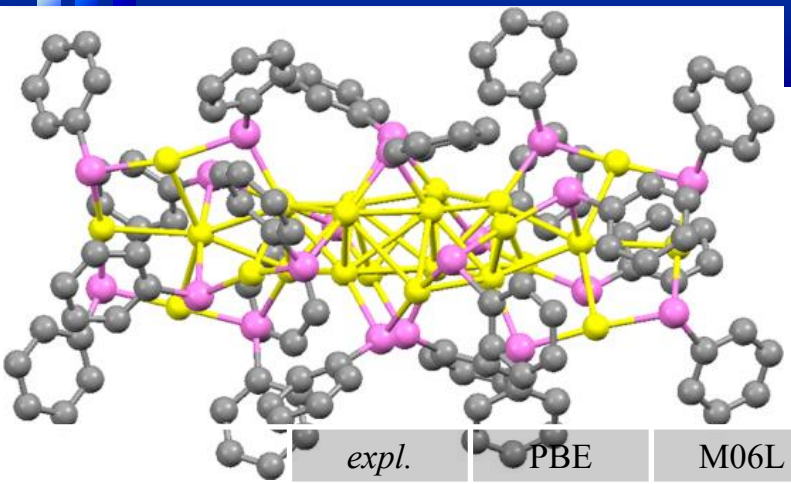


Taxol ($C_{47}H_{51}NO_{14}$)

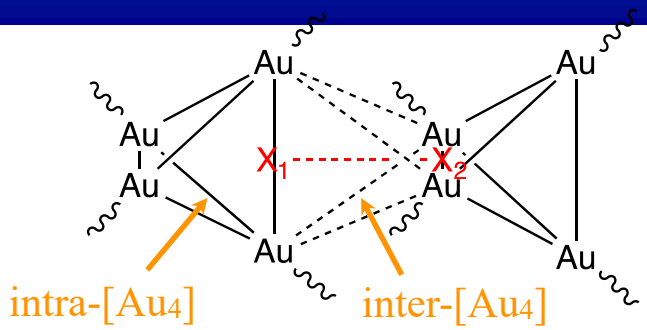


Luciferin($C_{11}H_8N_2O_3S_2$)

Selenolate-Protected Au Nanocluster; $Au_{24}(SePh)_{20}$



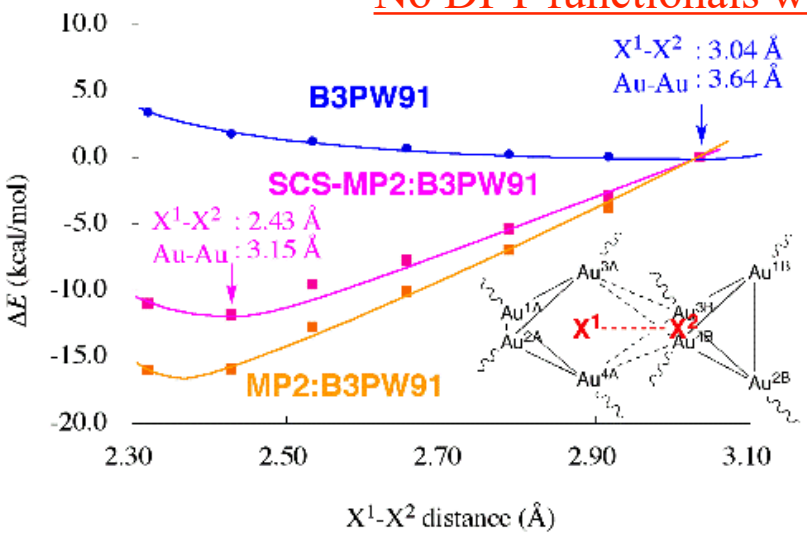
$Au_{24}(SePh)_{20}$



	expl.	PBE	M06L	B3LYP	B3PW91	TPSSh	M06	B3LYP-D	B3LYP-D3
inter-[Au_4]	*3.223*	3.401	3.173	3.926	3.637	3.17	3.296	3.386	3.220
intra-[Au_4]	*2.741*	2.831	2.857	2.855	2.82	2.811	2.899	3.000	2.927
[Au_4]-Se	*2.512*	2.561	2.610	2.577	2.545	2.572	2.595	2.524	2.543
Au-Se in [Au]$_3$	*2.402*	2.497	2.516	2.520	2.488	2.493	2.528	2.503	2.505
Au-Se in [Au]$_5$	*2.422*	2.512	2.534	2.525	2.499	2.510	2.540	2.496	2.503

No DFT functionals were found to describe the geometry.

† distances are in Å



Difference from the experimental values;
< 0.04Å
0.04Å-0.09Å
> 0.09Å

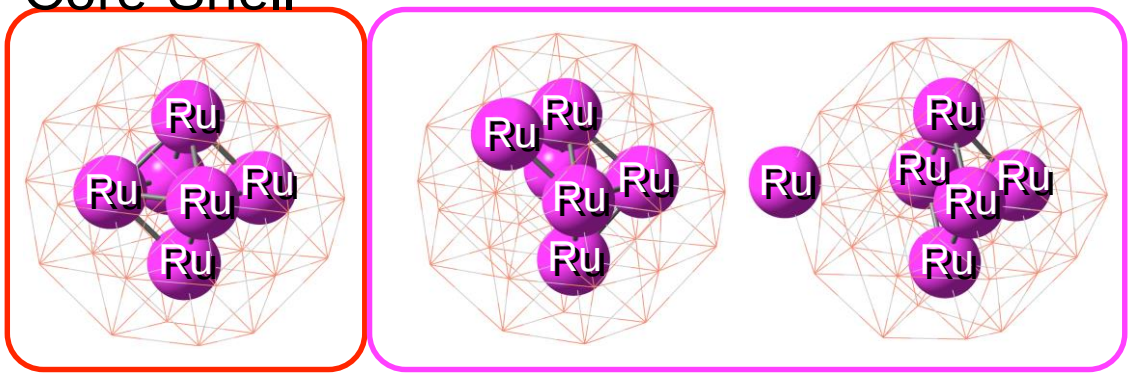
A energy minimum is located around Au-Au = 3.15Å at SCS-MP2 (3.22Å in expl.)(264atoms).

N. Takagi, K. Ishimura, et al., JACS 137 (2015) 8593.

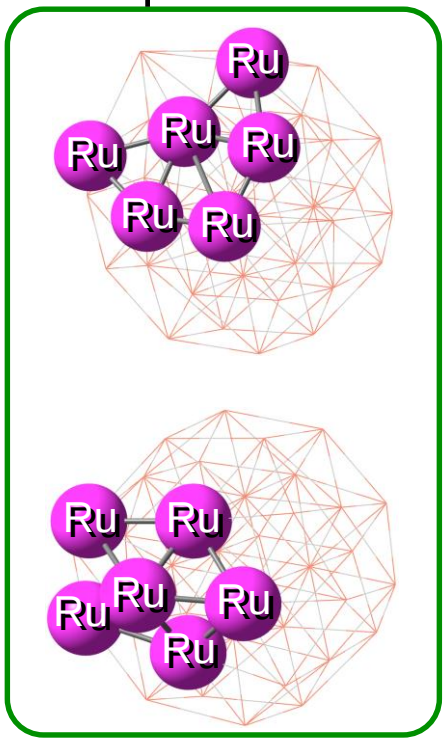
Models of Core-shell, Alloy and Phase-separated Structures of $Cu_{32}Ru_6$

N. Takagi, K. Ishimura, et al., JPCC 121 (2017) 10514.

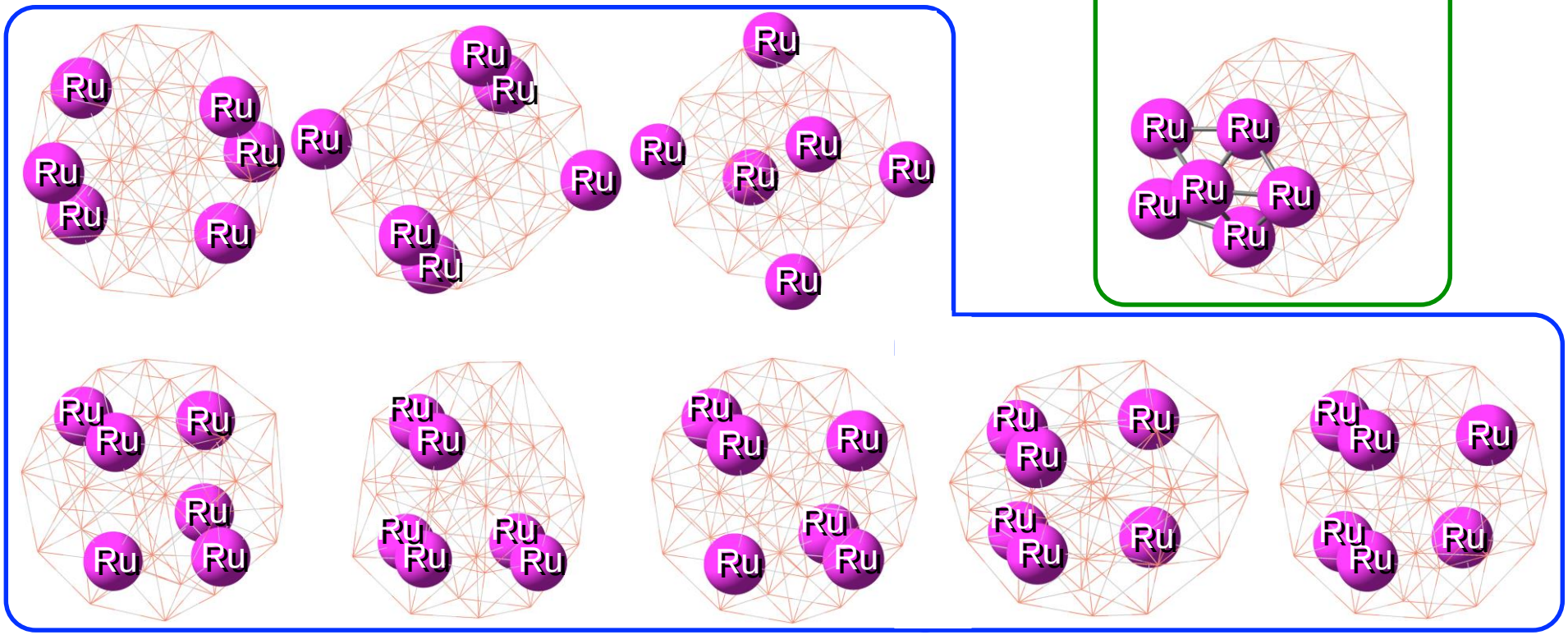
Core-Shell



Phase-separated

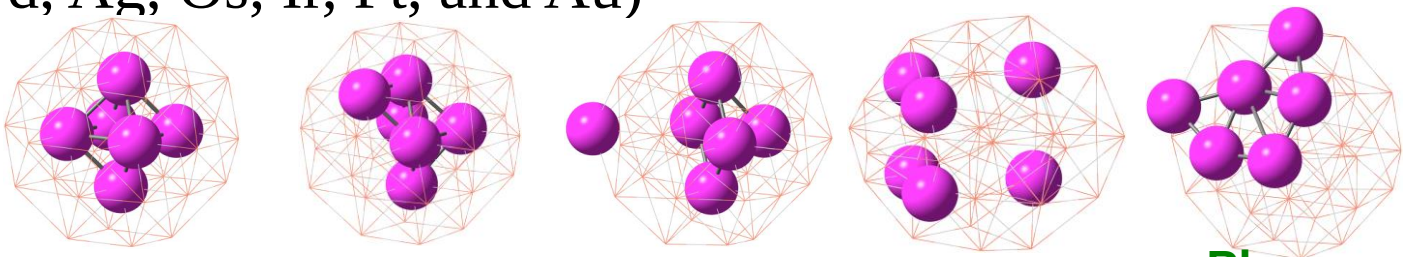


Fused-alloy



Relative Stability of $Cu_{32}M_6$

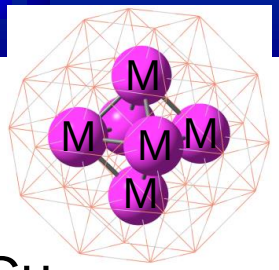
(M = Ru, Rh, Pd, Ag, Os, Ir, Pt, and Au)



(in kcal/mol)	Core-shell			Fused-alloy	Phase-separated
$Cu_{32}Ru_6$ (triplet)	0.0	+16.4	+34.8	+80.0	+101.1
$Cu_{32}Rh_6$ (singlet)	0.0	+19.5	+18.3	+47.0	+104.3
$Cu_{32}Pd_6$ (singlet)	0.0	-6.1	-7.6	-33.4	+24.9
$Cu_{32}Ag_6$ (triplet)	0.0	-16.9	-19.0	-109.6	-108.1
$Cu_{32}Os_6$ (singlet)	0.0	+33.0	+76.5	+157.3	+128.9
$Cu_{32}Ir_6$ (triplet)	0.0	+29.4	+38.9	+93.6	+139.1
$Cu_{32}Pt_6$ (singlet)	0.0	-5.5	-12.4	-51.6	+35.3
$Cu_{32}Au_6$ (triplet)	0.0	-23.7	-26.9	-141.2	-97.6

✓ Core-shell structures are stable for M = Ru, Rh, Os, Ir.

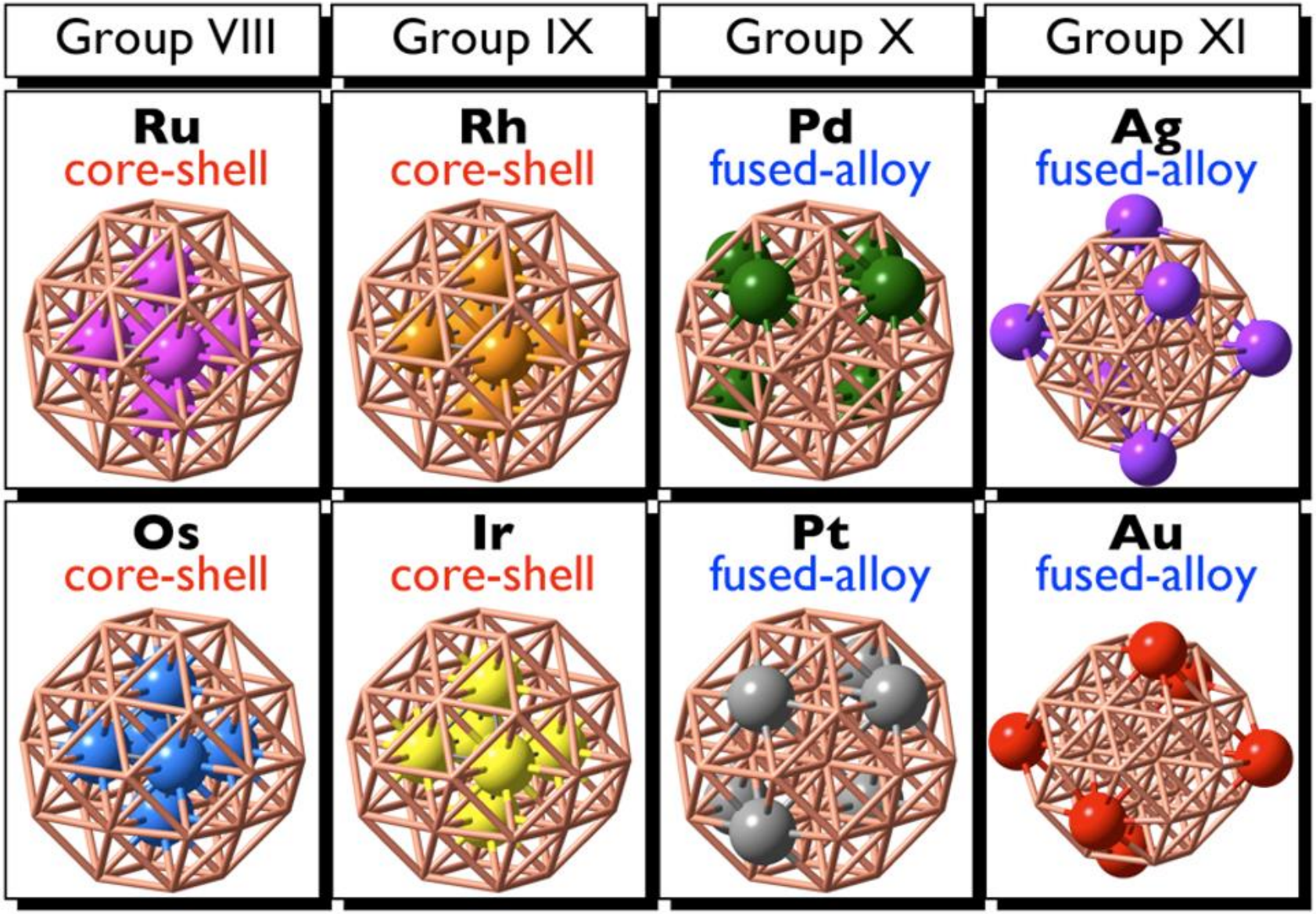
Core-shell Structure of $Cu_{32}M_6$



	d-electron population of M (*e*)	Unstabilization energy of Cu_{32} shell (kcal/mol)	M-M distance (Å)
Cu_{38} (triplet)	9.86 (-0.14)	0.0	2.572 (0.0)
$Cu_{32}Ru_6$ (triplet)	8.06 (+1.06)	19.9	2.684 (+0.112)
$Cu_{32}Rh_6$ (singlet)	8.90 (+0.90)	23.9	2.772 (+0.200)
$Cu_{32}Pd_6$ (singlet)	9.62 (+0.62)	40.3	2.880 (+0.308)
$Cu_{32}Ag_6$ (triplet)	9.90 (-0.10)	46.8	2.837 (+0.265)
$Cu_{32}Os_6$ (singlet)	7.72 (+0.72)	22.9	2.657 (+0.085)
$Cu_{32}Ir_6$ (triplet)	8.66 (+0.66)	22.3	2.738 (+0.166)
$Cu_{32}Pt_6$ (singlet)	9.60 (+0.60)	50.9	2.946 (+0.374)
$Cu_{32}Au_6$ (triplet)	9.88 (-0.12)	61.7	2.937 (+0.365)

	Stabilization by Charge transfer	**Deformation energy**		**Core-shell structure**
Ru, Rh, Os, Ir	**Large**	**small**	➡	**Stable**
Pd, Ag, Pt, Au	**Small**	**Large**	➡	**Unstable**

Stable Structure of $Cu_{32}M_6$ Cluster

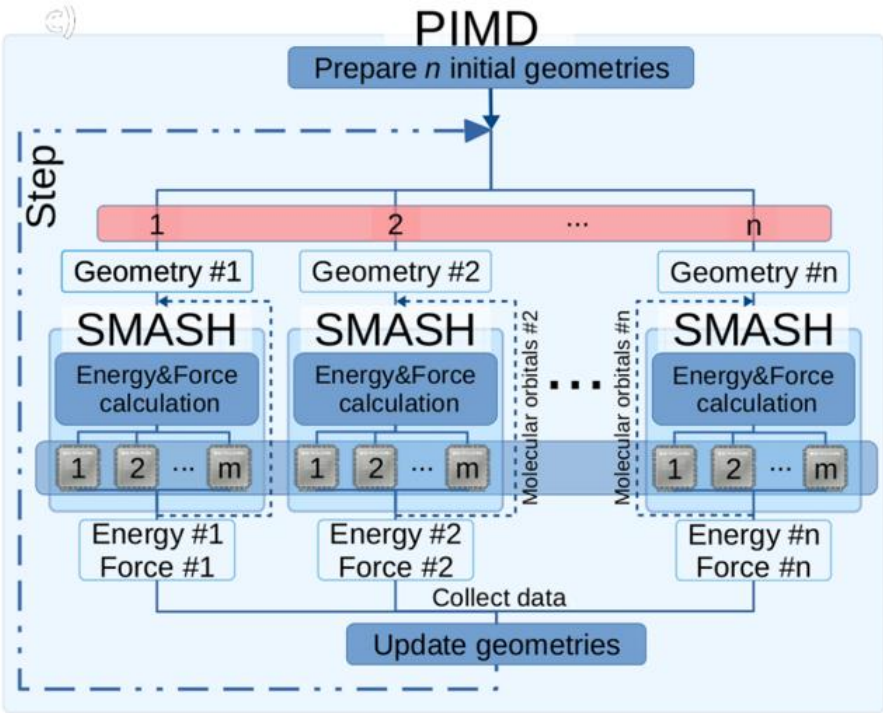


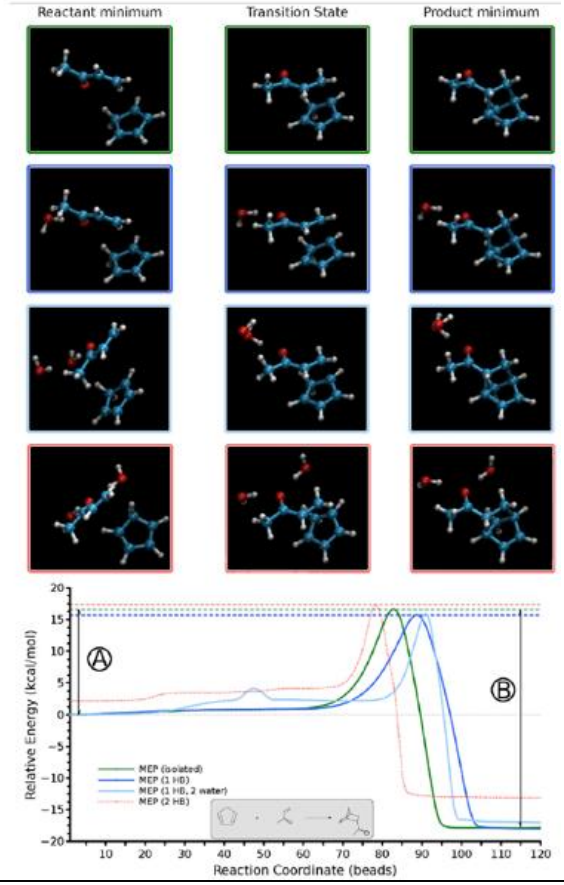
Electronic structure + Dynamics calculations

S. Ruiz-Barragan, K. Ishimura, M. Shiga, Chem. Phys. Lett. 646 (2016) 130.

+ SMASH

- Combination of PIMD program developed by Dr. M. Shiga (JAEA) and SMASH program
- Open source license (Apache 2.0)
- http://ccse.jaea.go.jp/ja/download/pimd/index.html





Diels-Alder reaction in aqueous solution (cyclopentadiene + butenone)
Minimum energy path – String method with 120 images.
B3LYP/6-31G*

Experiment: DA reaction is accelerated in aqueous solution
$\Delta\Delta G$ = 4 kcal/mol
Breslow et al. J. Amer. Chem. Soc. 102 (1980) 7816.

Further improvement for post-K computer

- ✓ Improvement of OpenMP parallel performance for many-core machines including post-K
- ✓ Reduction of Memory Usage

Post-K computer architecture
https://www.r-ccs.riken.jp/en/postk/project/outline

Architecture	Armv8.2-A SVE 512bit With the following Fujitsu's extensions: Hardware barrier, Sector cache, and Prefetch
Core	48 cores for compute and 2 or 4 cores for OS activities 4 CMGs (NUMA nodes)
	DP: 2.7+ TF, SP: 5.4+ TF, HP: 10.8 TF
Cache	L1D: 64 KiB, 4way, 230 GB/s (load), 115 GB/s (store)
	L2: 8 MiB, 16way, 115 GB/s (load), 57 GB/s (store)
Memory	HBM2 32 GiB, 1024 GB/s
Interconnect	Tofu Interconnect D (28 Gbps x 2 lane x 10 port)
I/O	PCIe Gen3 x16
Technology	7nm FinFET

Improvement of OpenMP performance1

✓ Increase the number of distributed tasks for OpenMP by combining two loops.

Hartree-Fock calculation

Before

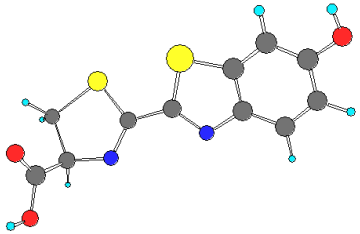
```
!$OMP parallel do schedule(dynamic,1) &
!$OMP reduction(+:Fock)
do μ=nbasis, 1, -1
  do ν=1, μ
    μν=μ(μ+1)/2+ν
    λstart=mod(μν+mpi_rank,nproc)+1
    do λ=λstart, μ ,nproc
      do σ=1, λ
         Calculate (μν|λσ) and
         add them to Fock matrix
      enddo
    enddo
  enddo
enddo
!$OMP end parallel do
call mpi_allreduce(Fock)
```

After

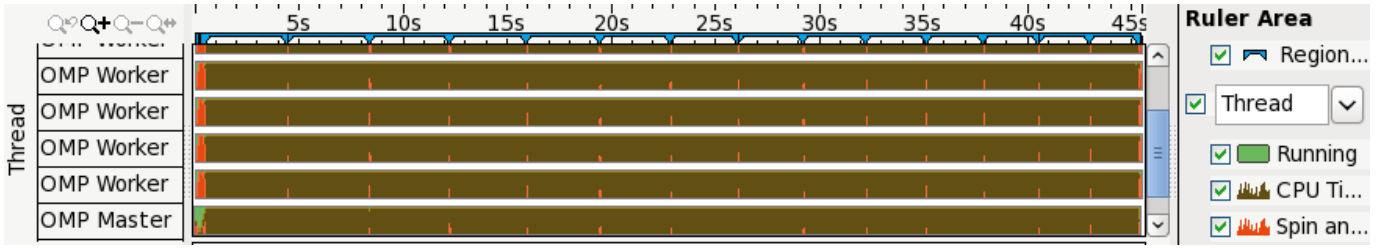
```
!$OMP parallel do schedule(dynamic,1) &
!$OMP reduction(+:Fock)
do μν=nbasis*(nbasis+1)/2, 1, -1
  Calculate μ and ν from μν
   λstart=mod(μν+mpi_rank,nproc)+1
    do λ=λstart, μ ,nproc
      do σ=1, λ
         Calculate (μν|λσ) and
          add them to Fock matrix
      enddo
    enddo

enddo
!$OMP end parallel do
call mpi_allreduce(Fock)
```

Improvement of OpenMP performance2

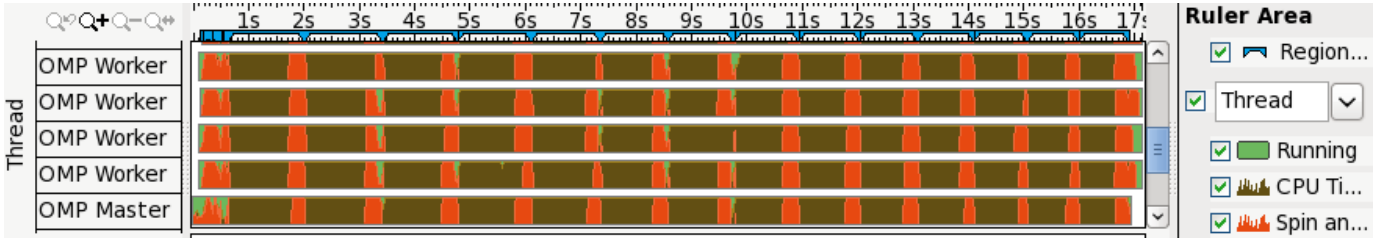
- Machine：Fujitsu PRIMEGY CX2500 (Xeon E5-2697, 28cores/node)
- Analysis of Intel VTune Amplifier2015
  - Molecule：Luciferin($C_{11}H_8N_2O_3S_2$)
  - Method：Hartree-Fock/cc-pVDZ(300 functions)



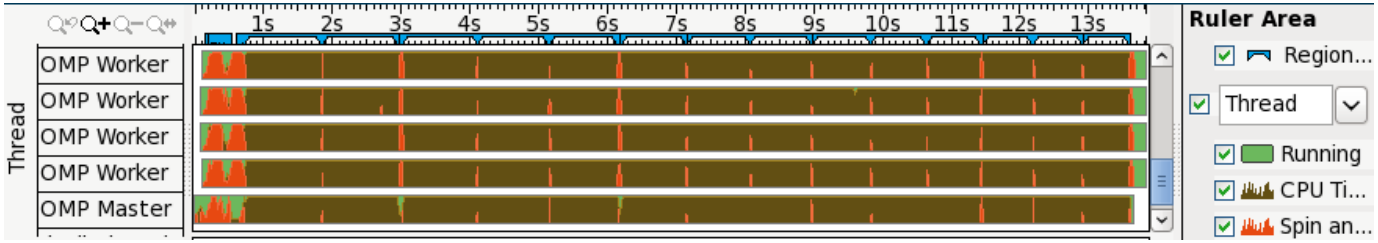
Before 8cores



Before 28cores



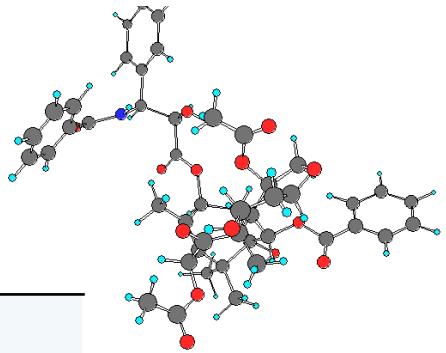
After 28cores



Orange: CPU waiting
Brown : CPU working

Improvement of OpenMP performance3

- Xeon Phi(Knights Corner, 5110P, 60cores, 1011GFlops)
- Elapsed time of Hartree-Fock calculation (sec)
- Molecule: Taxol($C_{47}H_{51}NO_{14}$)



	Before		After	
Number of threads	Time	Speed-up	Time	Speed-up
15	9034.6	15.0	9057.9	15.0
30	4559.0	29.7	4565.8	29.8
60	2391.9	56.7	2302.6	59.0
120	2652.9	51.1	1592.9	85.3
240	2888.6	46.9	1333.5	101.9

Reduction of memory usage1

- ✓ The ij-transformation is divided in order to reduce memory usage.
- ✓ The penalty is to calculate two-electron repulsion terms several times.

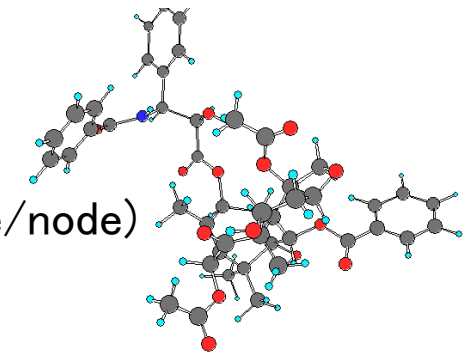
$$E_{MP2} = \sum_{ij}^{occ} \sum_{ab}^{vir} \frac{(ai|bj)\{2(ai|bj) - (aj|bi)\}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \qquad (ai|bj) = \sum_{\mu\nu\lambda\sigma}^{AO} C_{\mu a} C_{\nu i} C_{\lambda b} C_{\sigma j} \underbrace{(\mu\nu|\lambda\sigma)}_{\text{two-electron term}}$$

```
do μλ (AO index pair)
!$OMP parallel do schedule(dynamic,1)
  do σ
    two-electron term calculation(μν|λσ)  (all ν)
    1st transformation              (μi|λσ)  (all i)
  enddo
  2nd transformation(dgemm) (μi|λj)   (i≧j)
end do μλ
do all ij (MO index pair)
  call MPI_sendrecv (μi|λj)
  3rd transformation(dgemm)     (μi|bj)  (all b)
  4th transformation(dgemm)   (ai|bj)   (all a)
  MP2 energy calculation
end do ij
call mpi_reduce(MP2 energy)
```

```
do ij-block
 do μλ (AO index pair)
!$OMP parallel do schedule(dynamic,1)
   do σ
     two-electron term calculation (μν|λσ)  (all ν)
     1st transformation              (μi|λσ)  (partial i)
   enddo
   2nd transformation(dgemm) (μi|λj)   (i≧j)
 end do μλ
 do partial ij (MO index pair)
   call MPI_sendrecv (μi|λj)
   3rd transformation(dgemm)   (μi|bj)  (all b)
   4th transformation(dgemm)   (ai|bj)   (all a)
   MP2 energy calculation
 end do ij
end do ij-block
call mpi_reduce(MP2 energy)
```

Reduction of memory usage2

- Total memory usage $O^2N^2/2 \rightarrow O^2N^2/(2n_{ij\text{-}block})$ (O,N: Numbers of occupied orbitals and basis functions, $n_{ij\text{-}block}$:Number of ij-divisions)
  - Molecule：Taxol ($C_{47}H_{51}NO_{14}$)
  - Method：MP2/6-31G(d)
  - Machine：Fujitsu PRIMEGY CX2500 (Xeon E5-2697, 28core/node)



MP2 energy calculation time(sec) and memory usage(GB)

Number of ij-divisions	1	2	3
Total time	**765.7**	**943.4**	**1121.5**
Time for two-electron term calculation and 1st transformation	618.8	803.6	981.2
Memory	**101.0**	**50.5**	**33.7**

20%, 40% up

50%, 67% down

Reduction of memory usage3

- Speed-up of MP2 energy derivative calculation
  - Computational condition：$C_{150}H_{30}$、MP2/cc-pVDZ
  - Machine: K Computer
  - Super-linear speed-up was achieved. The number of divisions reduced as the number of nodes increases because the total amount memory increases.

MP2 energy derivative calculation time (sec) and Speed-up

Number of CPU cores	1536	3072	6144	12288
Number of ij-divisions	3	2	1	1
Elapsed time	4253.6	2002.0	743.6	382.0
Speed-up	1536.0	3263.2	8785.6	17103.2

Summary

- The SMASH program has been developed and distributed under the Apache License 2.0.
- The speed-up of the DFT calculation (360atoms, 4500 dimensions) is more than 50,000 on 100,000 cores.
- It is now practical to calculate electronic structures of nano-sized molecular systems.
- To distribute source codes under an open source license is one of the approaches to reduce development costs and to share programming techniques in the community.
- It is important to improve OpenMP parallel performance for many-core machines.
- The reduction of memory usage is necessary to calculate large molecular systems with accurate methods.