中国科学院
计算机网络信息中心
Computer Network Information Center,
Chinese Academy of Sciences

IXPUG
INTEL EXTREME PERFORMANCE USERS GROUP

Introduction to Benchmark Test for Multi-scale Computational Materials Software

Shun Xu*, Jian Zhang, Zhong Jin
xushun@sccas.cn

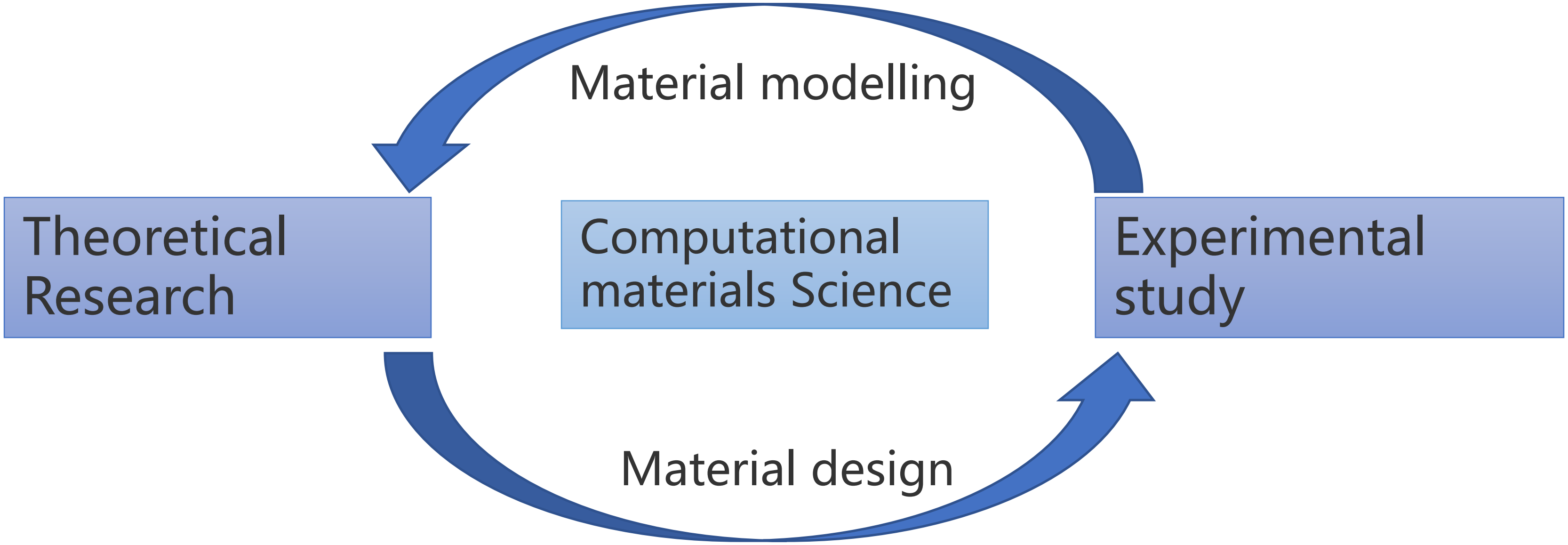
Computer Network Information Center
Chinese Academy of Sciences
(IPCC member)

2019.1.14

Outline

- Introduction to computational materials software
- Benchmark testing routine proposed
- Current progress of benchmark testing system
- Conclusions

The Computational Materials Science



Material modelling

Theoretical Research

Computational materials Science

Experimental study

Material design

Multiscale mechanics

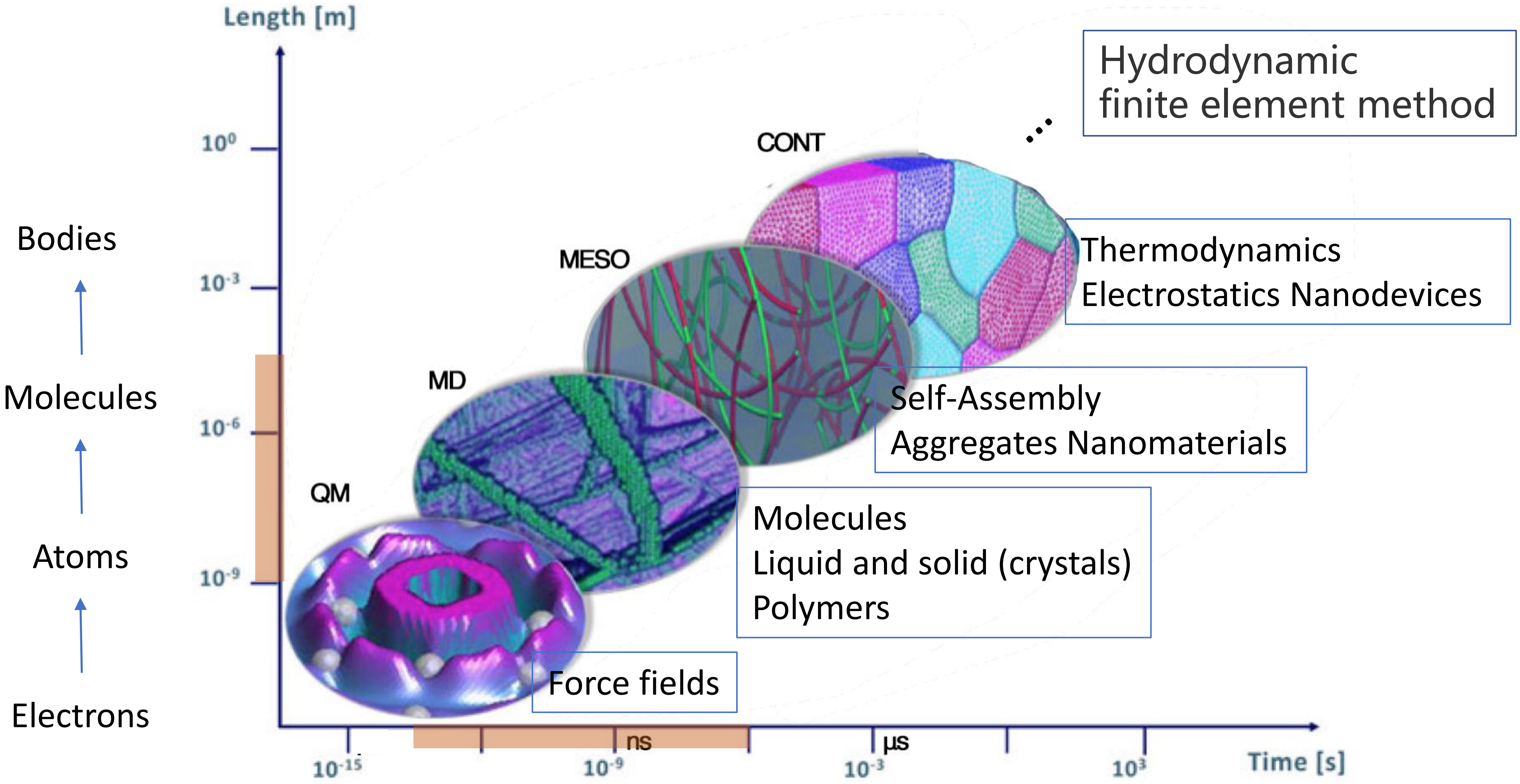
- Statistical Mechanics
- Fluid Mechanics
- Molecular Mechanics
- Quantum Mechanics

Difficulties in Computational Materials Science

- Multidimensional system
- Nonlinear system
- Various of characteristic space scale

These are great challenges of improving computational performance

The temporal and spatial scales of Computational materials



Bodies

Molecules

Atoms

Electrons

Hydrodynamic
finite element method

Thermodynamics
Electrostatics Nanodevices

Self-Assembly
Aggregates Nanomaterials

Molecules
Liquid and solid (crystals)
Polymers

Force fields

Methods used in Computational Materials Science

The commonly used methods include

- First-principles/ab initio method
- Molecular dynamics method
- Monte Carlo method
- Cellular automata method
- Phase field method
- Geometric topological model method
- Finite element analysis
- …

Software developed for Computational Materials Science

- First-principles/ab initio method
  - **NWChem**
  - QUANTUM ESPRESSO
  - Abinit
  - Gaussian
  - VASP
- Molecular dynamics method
  - **LAMMPS**
  - GROMACS
  - CHARMM
  - Material Studio
- Phase field method
  - MICRESS

(incomplete list)

To harness High-Performance Computing facilities

Two typical open source HPC software of computational materials

- **NWChem**: Open Source High-Performance Computational Chemistry
  - nanostructures from quantum to classical
- **LAMMPS**: Large-scale Atomic/Molecular Massively Parallel Simulator
  - classical molecular dynamics on materials modeling

For better use, questions we need answer to are …

- **Where is** the bottleneck in these software running?
  - And how to evaluate it?
- **How to** estimate the best computational performance of these software over current computing resource?
  - And how to achieve the best performance?
- And more …

Resort to benchmark testing!

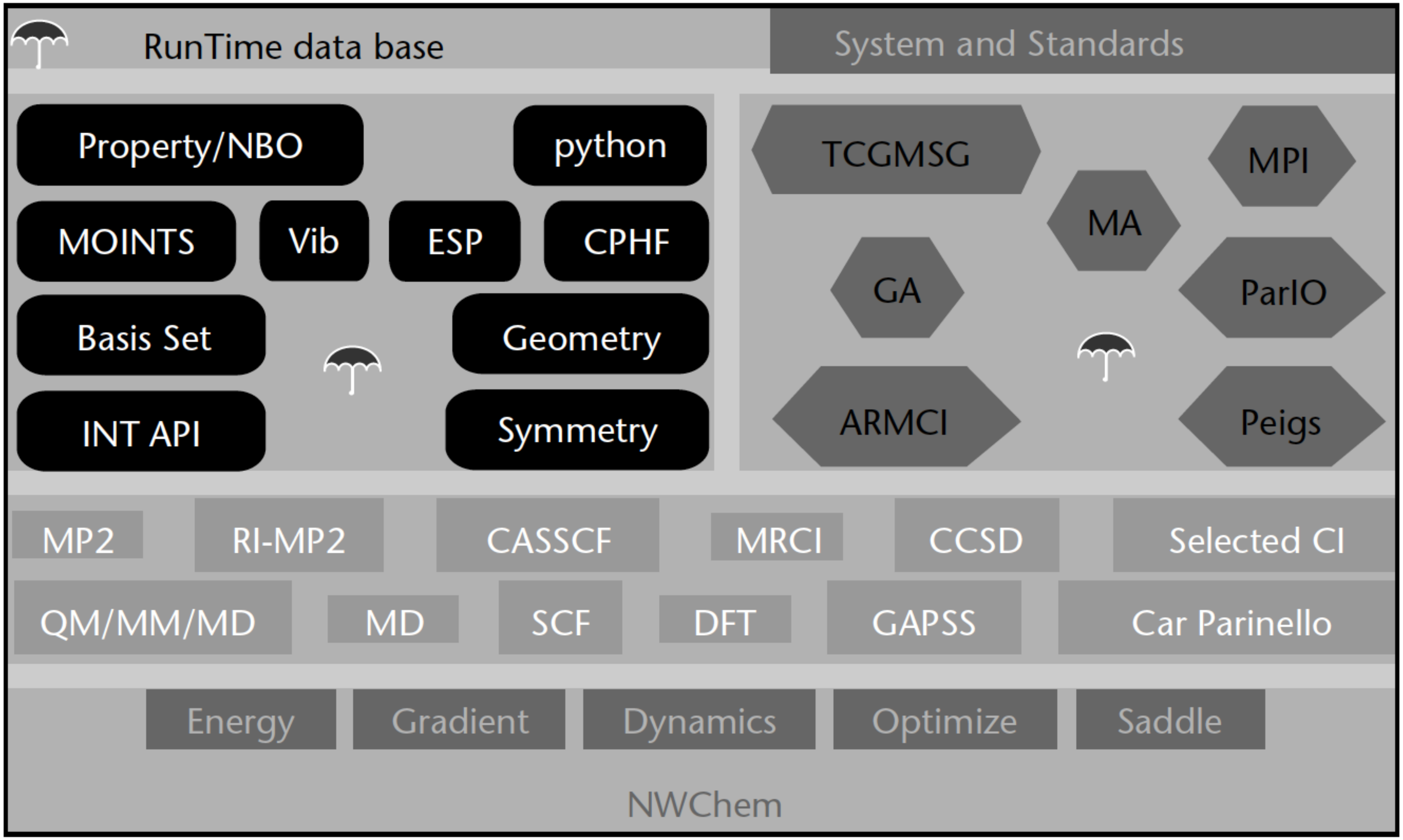
We propose a routine to benchmark testing

1. **To prepare for benchmark**
    - Collect **hardware** information
    - Collect **operation system** information
    - Choose **compiler** environments
    - Do **basic benchmarks** for system
2. **In benchmark testing of application software**
    - Locate common problems in target software
    - Write & profile a **prototype** program
    - Prepare typical use cases
    - Run & compare with **typical** use cases
3. **Over benchmark testing**
    - Do a way of batch processing
    - Setup a platform of performance analysis

NWChem: benchmark testing

- **Locate common problems** in target software
    - SCF: self-consistent field
    - Density functional theory
    - Ab initio molecular dynamics
    - Perturbation theory
    - Coupled cluster
    - MCSCF: multiconfiguration self-consistent field
    - CI: configuration interaction
    - Molecular mechanics
    - Molecular dynamics
    - Free energy simulation

NWChem: benchmark testing



The NWChem architecture representing general functionality within NWChem

(citation from: sourcebook of parallel computing)

NWChem: Features considered in Parallel performance

- The Global Array (GA) Toolkit
    - for one-sided access to shared data structures
    - for data locality sensitive
    - for high-level operations on distributed arrays for out-of-core array–based algorithms
    - for simulations driven by dynamic load balancing
- Peigs: a parallel eigen-solver
- MA: a portable memory allocator
- RTDB:The run-time database (a storage of key/value pairs)

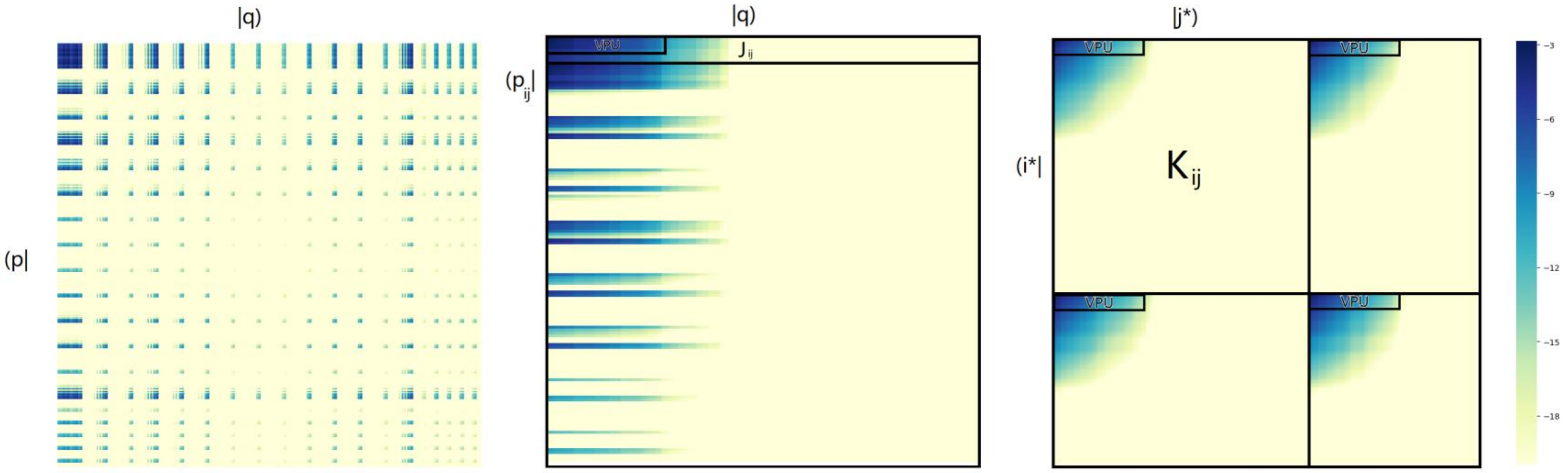
Write & profile a prototype program of HF-SCF

- Although a broad range of computational chemistry methods, **Hartree-Fock self-consistent field** (HF-SCF) is usual a common module as foundation.
- The complexity of HF-SCF
  - The cost of solving Fock eigen matrix scales with **$O(N^3)$,** where N is for basis functions.
  - But the cost of building Fock eigen matrix scales with the number of integrals, which is formally **$O(N^4)$** for N basis functions.
- HILQC: A prototype Program of HF-SCF

About HILQC

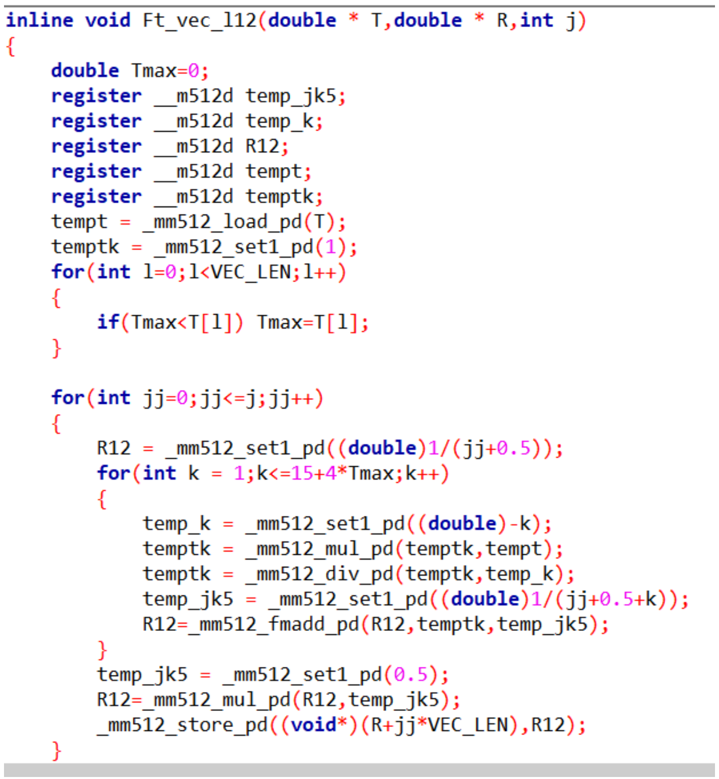
- HILQC is a **H**eterogeneous **I**ntegral **L**ibrary for **Q**uantum **C**hemistry
- Optimized in prescreening and presort stage on Intel Platform (MIC/KNL)
- Optimizing Two-Electron Repulsion Integral Calculation on Knights Landing Architecture *(Presented at IXPUG workshop 2018 by Dr. Yingqi Tian from our group.)*
- Integrated with Gaussian function based BDF (Beijing Density Functional) program package *(finished)*

Presort and J、 K matrix testing



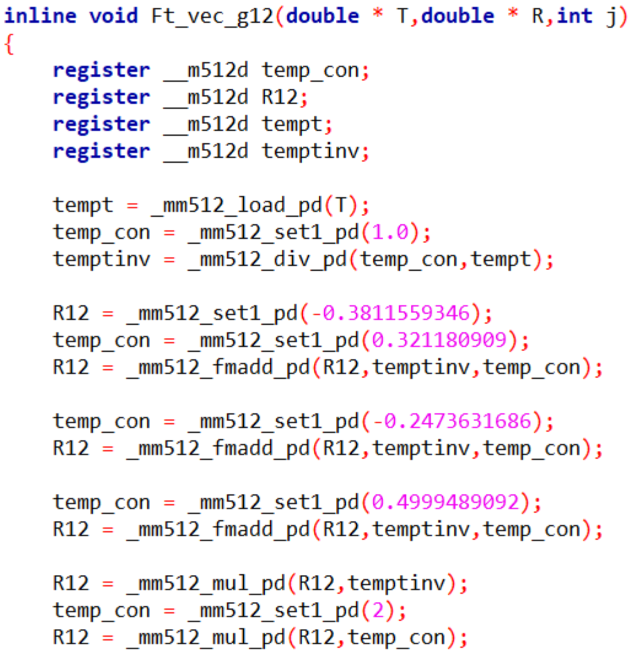
	Haswell	Broadwell	KNL	GAMESS
caffeine	0.14	0.12	0.13	0.1
cocaine	0.44	0.33	0.46	0.4
taxol	2.78	1.62	1.55	3.8
valinomycin	4.48	2.71	2.12	10.8

AVX-512 Intrinsic of incomplete gamma function



```
inline void Ft_vec_l12(double * T,double * R,int j)
{
    double Tmax=0;
    register __m512d temp_jk5;
    register __m512d temp_k;
    register __m512d R12;
    register __m512d tempt;
    register __m512d temptk;
    tempt = _mm512_load_pd(T);
    temptk = _mm512_set1_pd(1);
    for(int l=0;l<VEC_LEN;l++)
    {
        if(Tmax<T[l]) Tmax=T[l];
    }

    for(int jj=0;jj<=j;jj++)
    {
        R12 = _mm512_set1_pd((double)1/(jj+0.5));
        for(int k = 1;k<=15+4*Tmax;k++)
        {
            temp_k = _mm512_set1_pd((double)-k);
            temptk = _mm512_mul_pd(temptk,tempt);
            temptk = _mm512_div_pd(temptk,temp_k);
            temp_jk5 = _mm512_set1_pd((double)1/(jj+0.5+k));
            R12=_mm512_fmadd_pd(R12,temptk,temp_jk5);
        }
        temp_jk5 = _mm512_set1_pd(0.5);
        R12=_mm512_mul_pd(R12,temp_jk5);
        _mm512_store_pd((void*)(R+jj*VEC_LEN),R12);
    }
```



```
inline void Ft_vec_g12(double * T,double * R,int j)
{
    register __m512d temp_con;
    register __m512d R12;
    register __m512d tempt;
    register __m512d temptinv;

    tempt = _mm512_load_pd(T);
    temp_con = _mm512_set1_pd(1.0);
    temptinv = _mm512_div_pd(temp_con,tempt);

    R12 = _mm512_set1_pd(-0.3811559346);
    temp_con = _mm512_set1_pd(0.321180909);
    R12 = _mm512_fmadd_pd(R12,temptinv,temp_con);

    temp_con = _mm512_set1_pd(-0.2473631686);
    R12 = _mm512_fmadd_pd(R12,temptinv,temp_con);

    temp_con = _mm512_set1_pd(0.4999489092);
    R12 = _mm512_fmadd_pd(R12,temptinv,temp_con);

    R12 = _mm512_mul_pd(R12,temptinv);
    temp_con = _mm512_set1_pd(2);
    R12 = _mm512_mul_pd(R12,temp_con);
```

The calculation of all double electron integrals based on (GTOs) and (STOs) basis functions can be transformed into a series of incomplete Gamma functions

LAMMPS: benchmark testing

- Locate common problems in target software
  - classical molecular dynamics of atoms/particles
  - Interatomic potentials (force fields)
    - pair style
    - bond style
    - angle style
    - dihedral style
    - improper style
    - kspace style
  - Ensembles, constraints, and boundary conditions
  - Integrators
    - velocity-Verlet integrator
    - Brownian dynamics
    - rigid body integration
    - energy minimization

LAMMPS: Features considered in Parallel performance

GPU Package	for NVIDIA GPUs as well as OpenCL support
USER-INTEL Package	for Intel CPUs and Intel Xeon Phi
KOKKOS Package	for Nvidia GPUs, Intel Xeon Phi, and OpenMP threading
USER-OMP Package	for OpenMP threading and generic CPU optimizations
OPT Package	generic CPU optimizations

Many-core CPUs	USER-INTEL, KOKKOS, USER-OMP, OPT packages
NVIDIA GPUs	GPU, KOKKOS packages
Intel Phi	USER-INTEL, KOKKOS packages

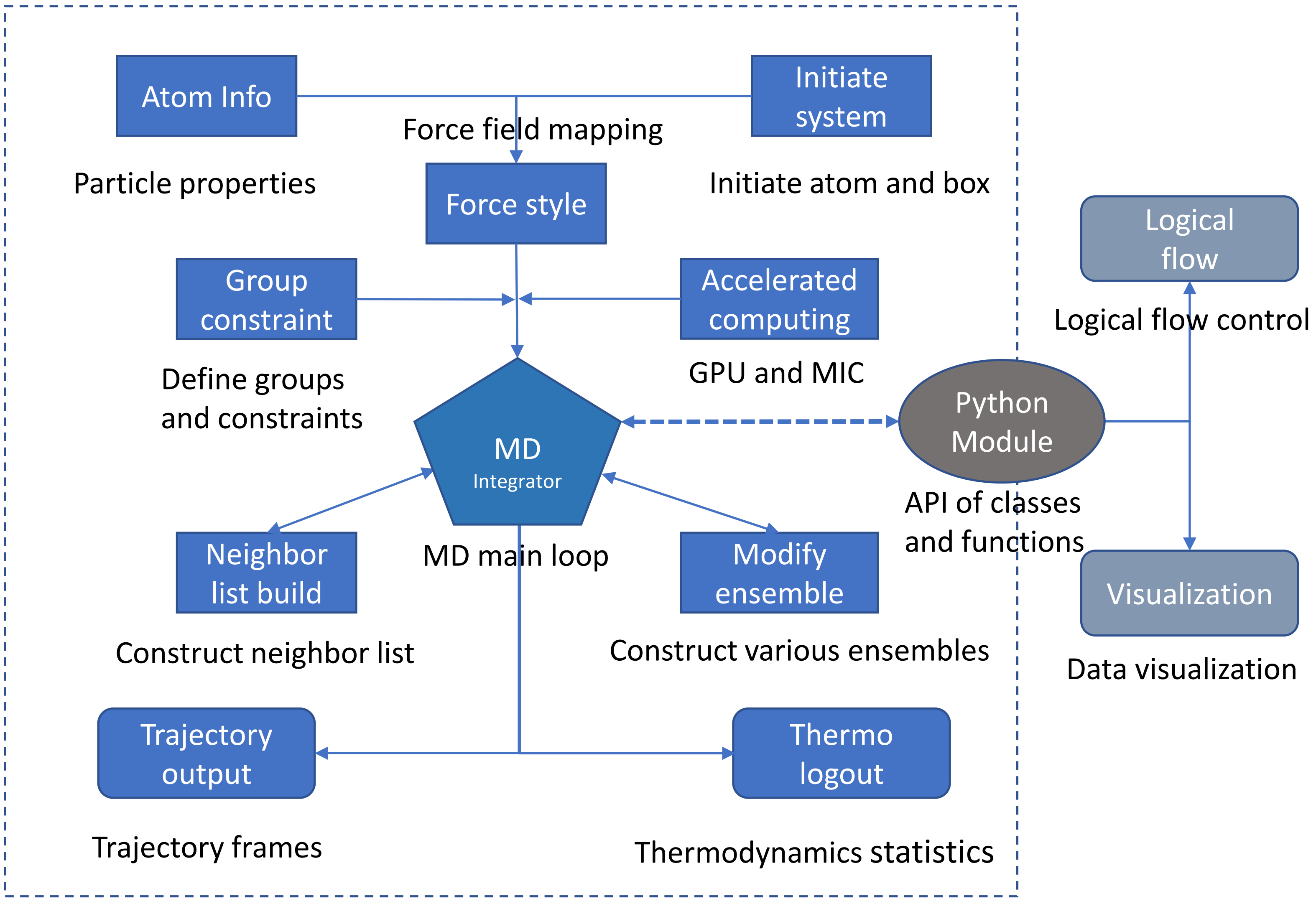
Write & profile a prototype program of MD

- The calculation of interatomic potentials (force fields) is a typical bottleneck.
- The complexity of Molecular Dynamics (MD)
    - The cost of building neighbor list of many-body short-range potentials
    - The cost of calculating long-range interactions for charge.
- eMD: A prototype Program of MD

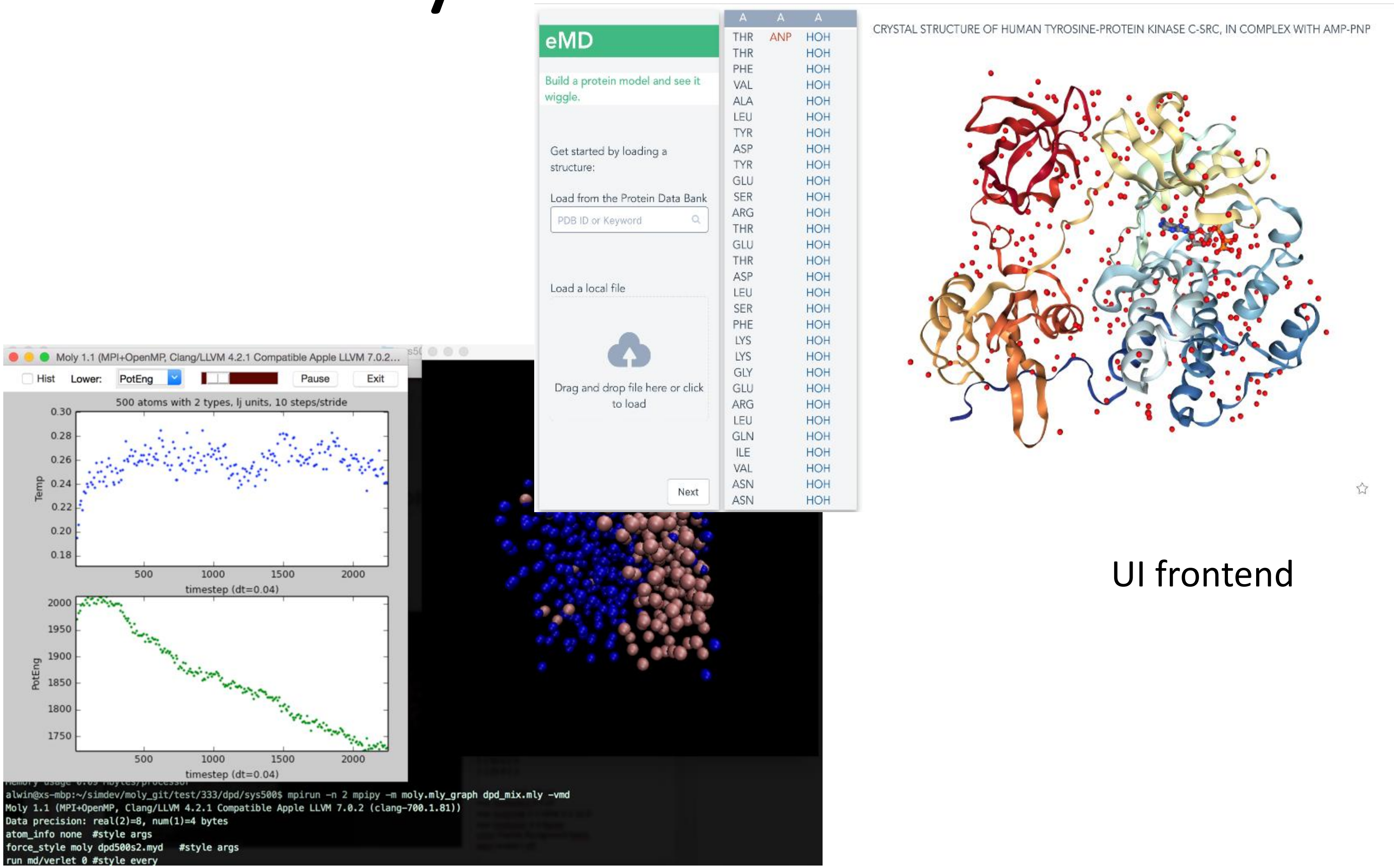
About eMD

- An easy-to-use MD software for biomacromolecules
- Standard MD simulation process, developed by CNIC, CAS
- Main Features
    - Based on Moly HPC Framework
    - Custom Force Field
    - Advanced Modeling
    - Heterogeneous Computing

The eMD software architecture for high performance Molecular Dynamics simulations



eMD & Moly



eMD

Build a protein model and see it wiggle.

Get started by loading a structure:

Load from the Protein Data Bank

Load a local file

Drag and drop file here or click to load

A	A	A
THR	ANP	HOH
THR		HOH
PHE		HOH
VAL		HOH
ALA		HOH
LEU		HOH
TYR		HOH
ASP		HOH
TYR		HOH
GLU		HOH
SER		HOH
ARG		HOH
THR		HOH
GLU		HOH
THR		HOH
ASP		HOH
LEU		HOH
SER		HOH
PHE		HOH
LYS		HOH
LYS		HOH
GLY		HOH
GLU		HOH
ARG		HOH
LEU		HOH
GLN		HOH
ILE		HOH
VAL		HOH
ASN		HOH
ASN		HOH

CRYSTAL STRUCTURE OF HUMAN TYROSINE-PROTEIN KINASE C-SRC, IN COMPLEX WITH AMP-PNP

Moly 1.1 (MPI+OpenMP, Clang/LLVM 4.2.1 Compatible Apple LLVM 7.0.2...

```
alwin@xs-mbp:~/simdev/moly_git/test/333/dpd/sys500$ mpirun -n 2 mpipy -m moly.mly_graph dpd_mix.mly -vmd
Moly 1.1 (MPI+OpenMP, Clang/LLVM 4.2.1 Compatible Apple LLVM 7.0.2 (clang-700.1.81))
Data precision: real(2)=8, num(1)=4 bytes
atom_info none  #style args
force_style moly dpd500s2.myd   #style args
run md/verlet 0 #style every
```

UI frontend

Computation backend

About Benchmark testing in NWChem and LAMMPS

- Locate calculation hotspot and bottleneck
  - In theoretical way
  - by profiling tools (Intel Vtune Amplifier)
- Write prototype program highlight it
  - Computation
  - Network
  - Memory and Disk
- Run & compare typical use cases
  - Compare results between prototype program and application software
  - Tell us how it works efficiently

Benchmark testing in batch processing

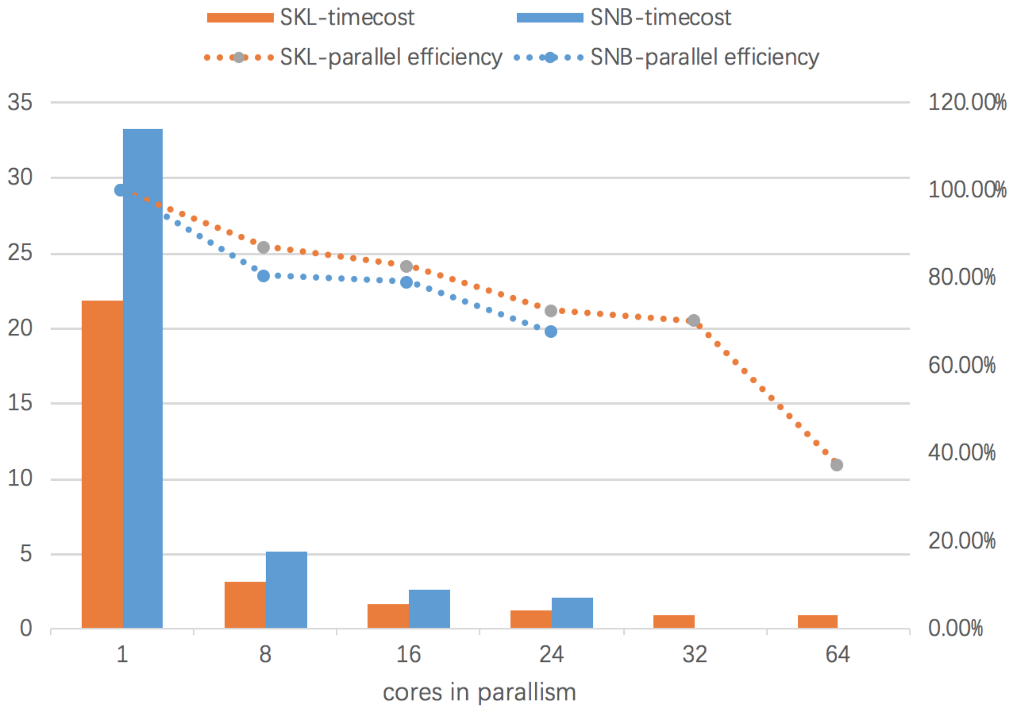
Some slides on this content are not posted online for privacy reason

A benchmark platform for performance analysis (under work)

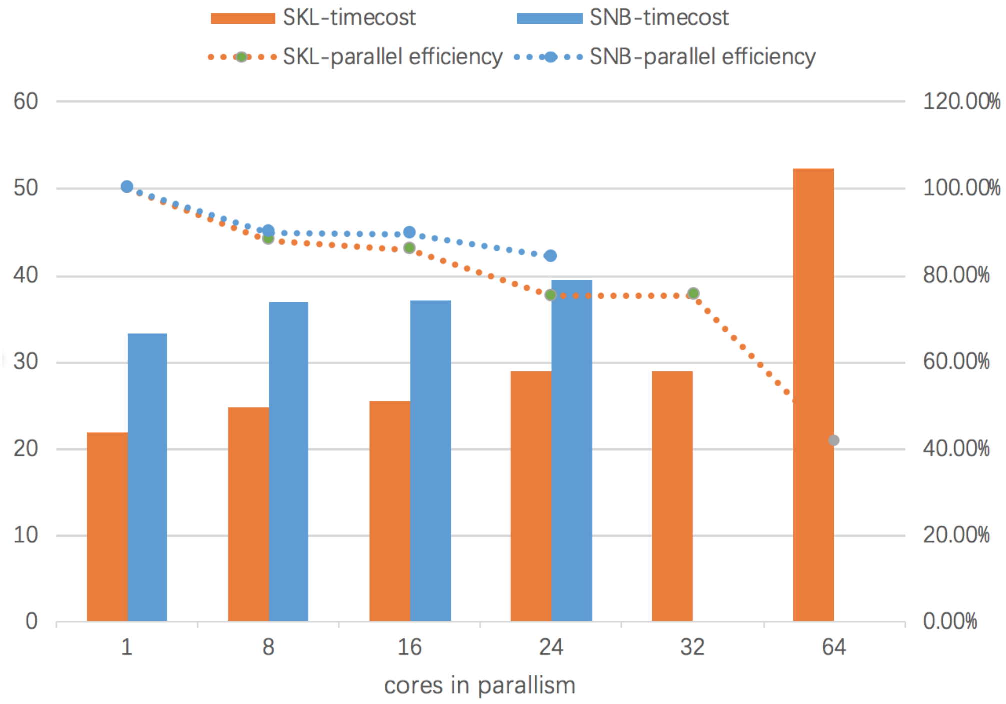
- To publish performance data in a uniform format
- To give a way to compare performance in testing systems
- To create a machine learning method for performance estimation
- To publish any information to help us well understand the software and hardware performance

Output figures from benchmark testing

LAMMPS benchmark of **rhodo** case by **only one command**



number of particles fixed (Amdahl law)



number of particles scaled (Gustafson law)

Conclusions

- To build **an efficient way** to well learn the performance of software and hardware, by benchmark testing system
- It is feasible for the benchmark testing system used from computational materials **to a broad range** of scientific applications
- A part of this work is summited **as a proposal** of IPCC project

Thank you!