Optimization of GENESIS/QSimulate on Fugaku for High-Performance QM/MM-MD Simulation using Current LLVM/Clang++ Compiler

162s1

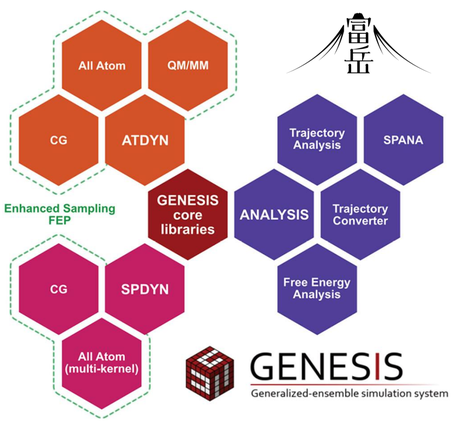
○Shingo Ito[1], Chigusa Kobayashi[1], Kiyoshi Yagi[2], Yuji Sugita[1,3,4]

[1] Computational Biophysics Research Team, RIKEN Center for Computational Science
[2] Department of Chemistry, Institute of Pure and Applied Sciences, University of Tsukuba
[3] Theoretical Molecular Science Laboratory, RIKEN Pioneering Research Institute
[4] Graduate School of Science, University of Tokyo

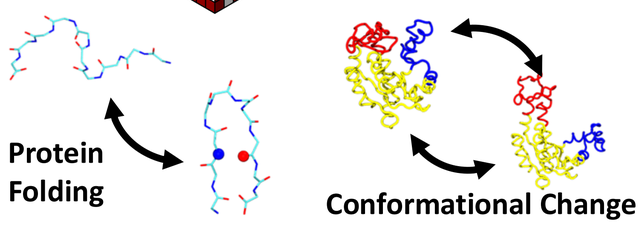
Background

The GENESIS/QSimulate, combined molecular dynamics (MD) program, GENESIS, and Quantum Mechanics/Molecular Mechanics (QM/MM) program, QSimulate-QM (https://qsimulate.com/academic), demonstrates excellent parallel scalability for QM/MM-MD simulations on multi-node supercomputers. High-performance QM/MM-MD simulations using this program have elucidated the mechanisms of biological functions, including enzymatic reactions, on Intel-based systems. However, its performance on the Fugaku, which employs Arm CPUs and the Fujitsu compiler, remains suboptimal due to insufficient code optimization.

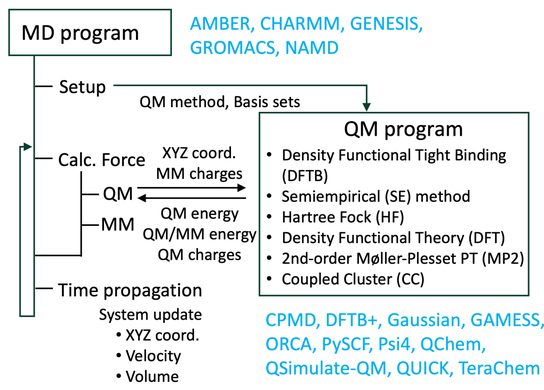
GENESIS/QSimulate



J. Jung et al., JPCB, 128, 6028−6048 (2024)



GENESIS (GENeralized-Ensemble SImulation System) is a high-performance molecular dynamics software package designed for large-scale biomolecular simulations, supporting advanced enhanced-sampling methods, and **particularly well optimized for the Fugaku supercomputer**.



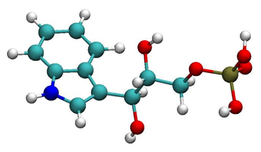
K. Yagi et al., JPCB, 125, 4701−4713 (2021)

High Performance QM/MM-MD

QSimulate is a high-performance quantum chemistry software package for large-scale electronic structure calculations.
By coupling **GENESIS** with **QSimulate**, efficient and scalable **QM/MM molecular dynamics** simulations are enabled, allowing accurate modeling of enzymatic reactions and ligand–protein interactions, which is valuable for **structure-based drug design**.

Optimization of GENESIS/QSimulate on Fugaku

Single-Point Calculations

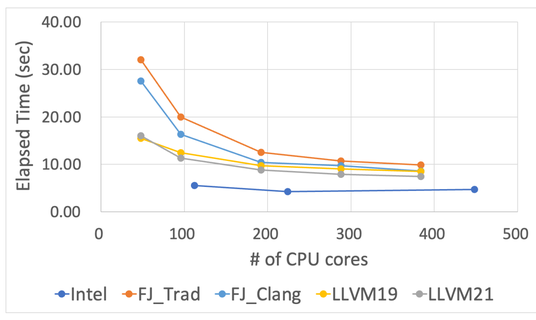


IGP (33 QM atoms)

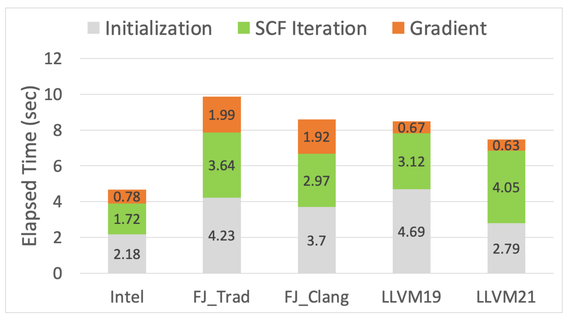
First, we evaluated the performance of **QSimulate** on **Fugaku** (CPU=Intel Xeon Max 9480) and **HOKUSAI BigWaterfall2** (so-called "HBW2", CPU=A64FX) using different compilers, **based on single-point calculations for small systems**.

System: IGP
QM Level: B3LYP-D3/cc-pVDZ

Time vs. Cores



Computational Time Breakdown
(448 cores (HBW2), 384 cores (Fugaku))

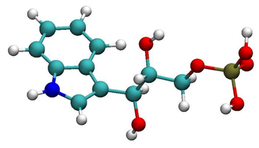


Efficiency Comparison with an Intel-Based Machine

Machine	Compiler	CPU core	Elapsed Time (sec)	Runtime Ratio
HBW2	Intel	448	4.68	x 1.00
Fugaku	FJ (Traditional)	384	9.68	x 2.07
Fugaku	FJ (Clang)	384	8.59	x 1.84
Fugaku	LLVM19	384	8.48	x 1.81
Fugaku	LLVM21	384	**7.47**	**x 1.60**

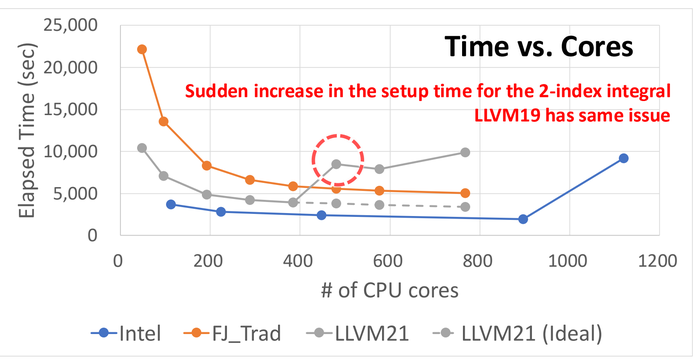
The LLVM compilers (ver. 19 and 21) did not improve the computational time of the **SCF iterations**, which are one of the most time-consuming parts of QM calculations. In contrast, the LLVM compilers significantly reduced the computational time of the **gradient calculations**, another major time-consuming component.

QM/MM-MD Simulations



IGP (33 QM atoms)

QM Level: B3LYP-D3/cc-pVDZ
NVT, 300K, 1,000 MD steps

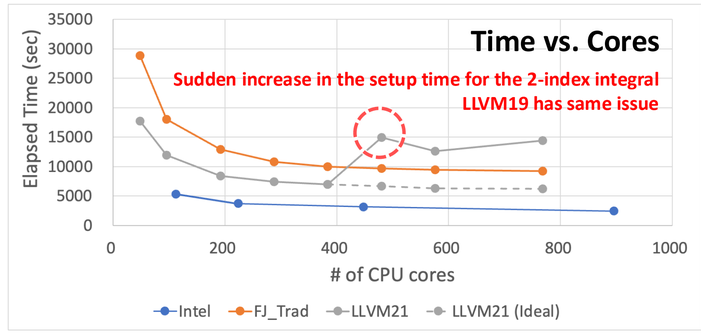


Efficiency Comparison with an Intel-Based Machine

Machine	Compiler	CPU core	SCF (sec)	Grad (sec)	Elapsed Time (sec)	Runtime Ratio
HBW2	Intel	896	1101.60	336.04	1937.14	x 1.00
Fugaku	FJ (Traditional)	768	1587.06	1230.21	5027.28	x 2.60
Fugaku	LLVM21	384	2050.27	592.72	**3918.86**	**x 2.02**
Fugaku	LLVM21 (Ideal)	768	1735.47	434.51	3413.26	x 1.76

TIM (37,226 Atoms: 37 QM)

QM Level: B3LYP-D3/cc-pVDZ
NVT, 300K, 1,000 MD steps



Efficiency Comparison with an Intel-Based Machine

Machine	Compiler	CPU core	SCF (sec)	Grad (sec)	Elapsed Time (sec)	Runtime Ratio
HBW2	Intel	896	1155.34	633.71	2462.81	x 1.00
Fugaku	FJ (Traditional)	768	2098.65	4132.13	9269.55	x 3.76
Fugaku	LLVM21	384	2715.94	2508.10	6982.46	**x 2.84**
Fugaku	LLVM21 (Ideal)	768	2278.13	2285.26	6250.00	x 2.54

Conclusion & Future work

This work demonstrates the feasibility of performing long-time, high-accuracy QM/MM-MD simulations on the Fugaku using the recent LLVM/Clang++ compiler optimized for C++ code. The optimized code achieved performance close to that of programs running on Intel CPUs with the Intel compiler. Further optimization of the code for the recent LLVM/Clang++ compiler is expected to yield additional performance improvements.