

RIKEN Miniapps for Computational Science Codesign

Naoya Maruyama

nmaruyama at riken.jp

RIKEN Advanced Institute for Computational Science

2014 Blue Waters Symposium

May 15, 2014

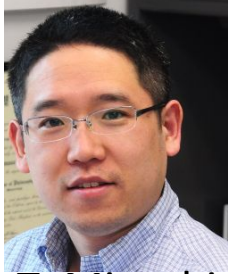
Overview of RIKEN AICS



- RIKEN (pronounced as ree-ken)
 - One of the largest government-funded research laboratories in Japan
 - AICS is one institute within RIKEN, established in 2010 to operate the K computer
- Director: Kimihiko Hirao
- Operations Division
 - Responsible for operations and management of the K computer
- Research Division
 - Division Head: Akinori Yonezawa
 - 19 research groups
 - Approximately 100 research members
- Exascale Supercomputing Project
 - Newly formed in April 2014



Computational Science Groups



T. Miyoshi
(Data Assimilation)



Y. Sugita
(Biophysics)



N. Ito
(Discrete Event Sim.)



J. Makino
(Astrophysics)



Y. Kuramashi
(Field Theory)



F. Tama
(Comp. Biology)



M. Hori
(Disaster Simulation)



S. Yunoki
(Material Science)



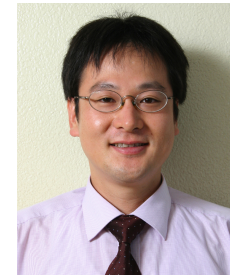
K. Hirao
(Comp. Chem.)



H. Tomita
(Climate)

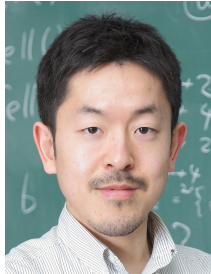


T. Nakajima
(Molecular Science)

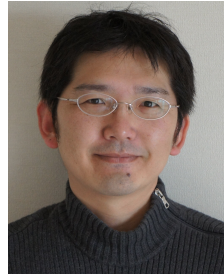


M. Tsubokura
(Fluid Simulation)

Computer Science Groups



N. Maruyama
(Prog. Framework)



T. Imamura
(Numerical Comp.)



K. Ono
(Viz.)



T. Maeda
(Usability)



Y. Ishikawa
(System Software)



M. Sato
(Prog. Environment)



M. Taiji
(Processor)

Japanese Exascale Activities

- Q3 2012 -- Q1 2014: Preliminary feasibility study projects (MEXT HPCI FS)
 - Explored enabling technologies for systems in the 2018-2020 timeframe
 - Identified key application requirements for the same timeframe
- Q2 2014 --: Exascale Computing Project launched at RIKEN AICS
 - Plan to replace the K computer in 2018-2020
 - Focus science, not Top500
 - More close collaboration between architecture, system software, and applications

Miniapp?

A tool for application and architecture codesign

- *Small* applications
 - Lines of code < 10,000
 - Simplified program organization
- *Not too small* applications
 - Retain essential characteristics of the original applications
 - What is essential?
- Simple process to “Download → Compile → Execute”
 - Open-source licenses
 - Easy compilation steps
 - Documentation of program execution input and parameters
 - Packaging of necessary input files

FiBER Mini-App Suite

- A suite of miniapps derived from the full-scale applications for the future computational science challenges
 - Originally supported by the feasibility study project
 - Currently supported by the RIKEN exascale project
- Originally developed and used on high-end machines such as the K computer
- Mainly developed at AICS with collaboration with the full-app developers
- <http://fiber-miniapp.github.io/>

Development & Usage

- Call for applications to computational science communities in Japan
 - Required agreement on open-source licensing
 - 17 full-scale apps submitted to RIKEN, being converted to mini-apps at RIKEN
 - 8 mini-app-like small apps submitted to RIKEN, packaged as mini-apps at RIKEN
- Provided to the pre-exa feasibility study projects for evaluating their proposed systems
- Used to design and evaluate future architectures in the exascale project

Submitted Applications (1/3)

feram	Nishimatsu	Ferroelectrics MD	OpenMP (with MPI for parameter survey), 3D FFT
MARBLE	Ikekuguchi	MD(PME)	MPI+OpenMP, 3D FFT
SMASH (para-TCCI)	Ishimura	Hartree-Fock	MPI+OpenMP, Sequential diagonalization of dense matrices
FFVC	Ono	Thermal-Fluid Analysis	MPI+OpenMP, SOR or GMRES
pSpatioocyte	Iwamoto	Signal propagation	MPI+OpenMP
NEURON_K+	Kazawa	Neural circuit simulation (Modified NEURON)	MPI+OpenMP, many ALL_GATHER, Translated to C from modeling language
GT5D	Idomura	5D plasma turbulence (5D FDM+2D FEM)	MPI+OpenMP, CG, 1D FFT
MODYLAS	Ando	MD(FMM)	MPI+OpenMP
STATE	Inagaki	First-Principles MD (DFT)	MPI+OpenMP (Replica parallel, k-point parallel, band or plane-wave parallel), FFT, eigenvalue problem (RMM)
FrontFlow/blue	Yamade	Thermal-Fluid Analysis (Irregular mesh, FEM)	MPI+auto parallelization, BiCGSTAB
SiGN-L1	Tamada	Neural network (L1 regularization)	MPI+OpenMP, Bottleneck at file output
NTChem/RI-MP2	Katouda	Electron correlation	MPI+OpenMP, DGEMM, Some sequential computation, Memory usage $O(N^3)$

Submitted Applications (2/3)

OpenFMO	Inadomi	Hartree-Fock FMO	MPI+OpenMP, Dynamic load balancing
CONQUEST	Miyazaki	First-principles MD, O(n) method	MPI+OpenMP SpMV, FFT
NGS Analyzer	Tamada	Genome sequence analysis	MPI I/O bound
DCPAM	Nishizawa	Climate model with spectral method	MPI+OpenMP
RSGDX	Hyodo	Earthquake simulation	

Submitted Applications (3/3)

Simplified versions submitted to the RIKEN FS

fft_check, fft_check_mpi	Nishimatsu	3D FFT benchmarking
ALPS/looper	Todo	Quantum monte carlo, linked lists, integer ops MPI+OpenMP
CCS QCD Solver Benchmark test program	Ishikawa	Lattice QCD benchmark, BiCGStab, MPI+OpenMP
ZZ-EFSI	Sugiyama	Fluid Structure Integration MPI+OpenMP
rmcsm bench nocore	Shimizu	Monte Carlo Nuclear Shell Model MPI+OpenMP
GCEED	Nobusada	DFT
NICAM-DC	Yashiro	Climate model (NICAM), dynamics, FVM MPI
mVMC	Imada	Strongly correlated matter

FiBER Mini-App Current Status

CCS QCD	Lattice QCD	→Tokyo FS, Tsukuba FS
MARBLE	MD(PME)	
MODYLAS	MD(FMM)	→Tokyo FS, Tsukuba FS
FFVC	Thermal-Fluid Analysis (Cartesian, FDM)	→Tohoku FS
NGS Analyzer	Genome Sequence Analysis	→Tokyo FS, Tsukuba FS
ALPS/looper	Quantum Monte Carlo	(Almost done)
CONQUEST	First-Principles MD ($O(N)$)	(Almost done)
NICAM-DC	Climate	(Almost done)
FrontFlow/ blue	Thermal-Fluid Analysis (Irregular mesh, FEM)	(Under development)
mVMC	Variational Monte Carlo	(Under development)

Molecular Dynamics

- Two alternative algorithms for solving equivalent problems
 - Particle Mesh Ewald
 - Bottlenecked by all-to-all communications at scale
 - Example implementation: MARBLE (Ikeguchi et al.)
 - Fast Multipole Method
 - Tree-based problem formulation with no all-to-all communications
 - Example implementation: MODYLAS (Okazaki et al.)
- Allows algorithmic comparisons

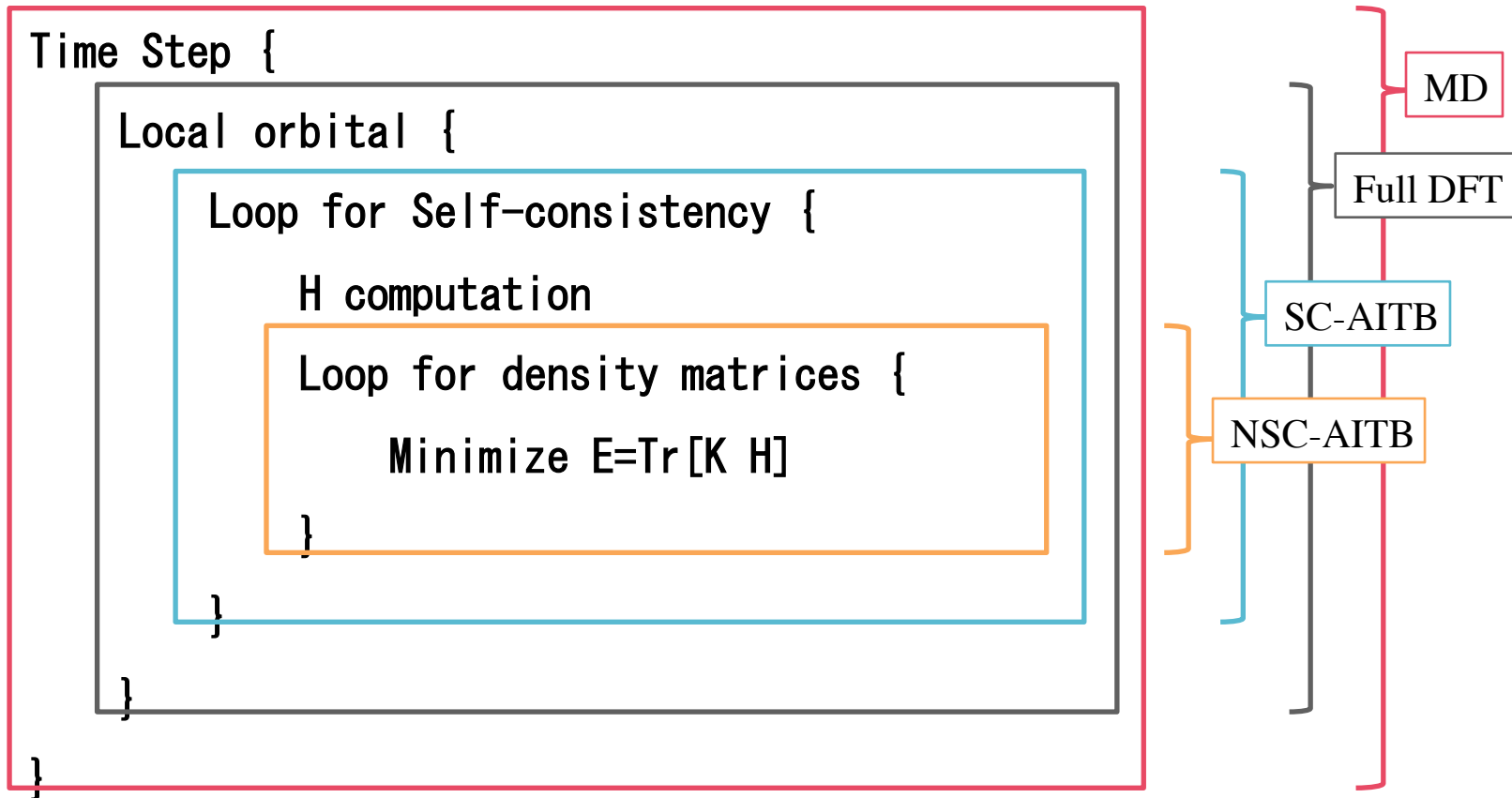
Molecular Dynamics

- Simplified problem settings
 - Only simulates water molecules in the NVE setting
 - Can reduce the codebase significantly
 - Easier to create input data sets of different scales
 - Whether it's sufficient is still under discussions
- Kernels: Pairwise force calculation + Long-range updates (FFT or FMM)
- Two reference implementations to study performance implications by algorithmic differences
 - MARBLE (12K SLOC)
 - MODYLAS (11K SLOC)

CONQUEST

- First-Principles computation with $O(n)$ method
 - Structural optimization, molecular dynamics
- Originally developed by Tsuyoshi Miyazaki (National Institute for Materials Science), et al.
 - URL: <http://www.order-n.org/>
- Production application with very large codebase
 - 115K lines of Fortran code
- Target problem size:
 - Ensemble simulations with 100K to 1M atoms
 - Simulation of 100M atoms

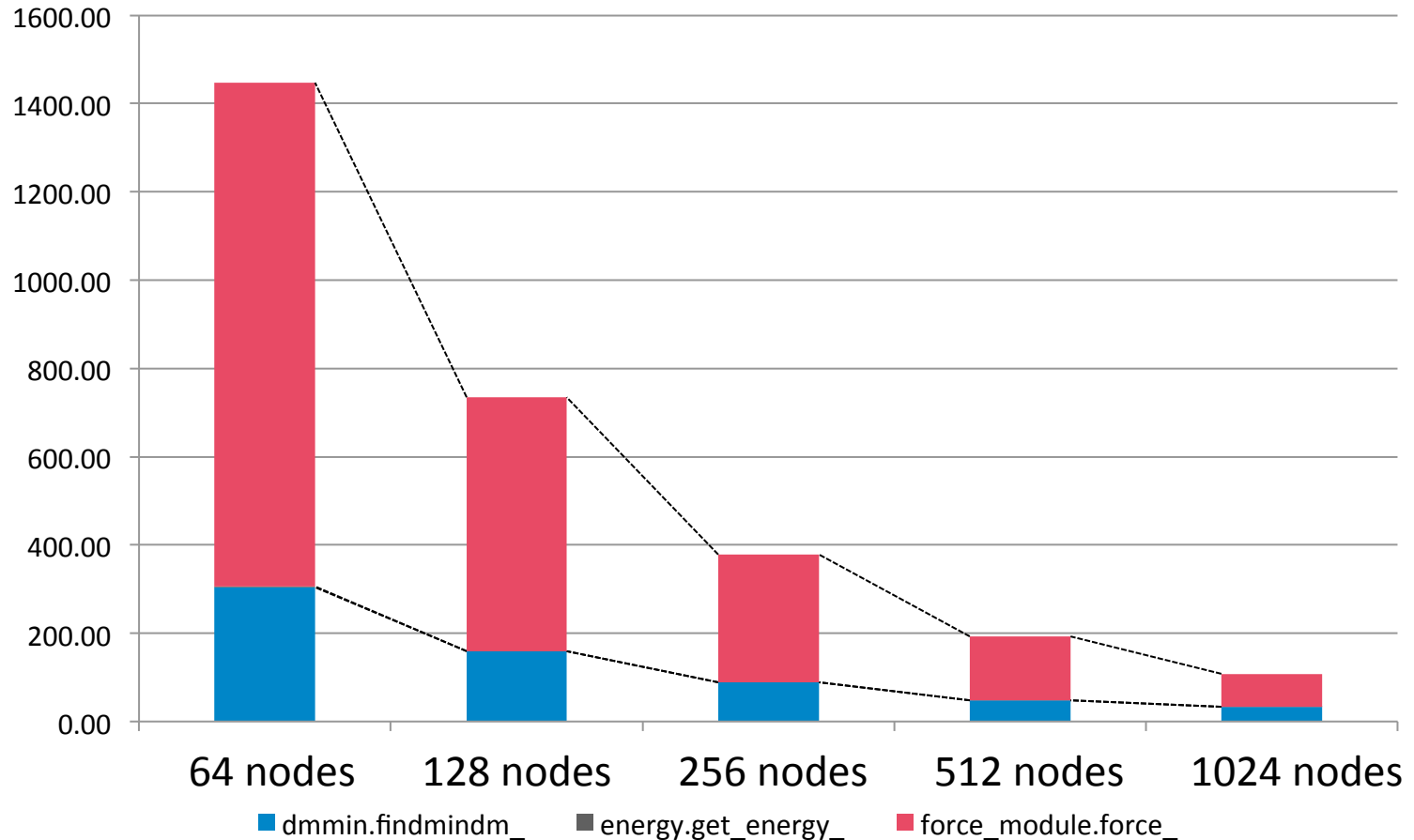
Program Loop Structure and Mini-App



- Compute NSC-AITB only as benchmark mode
 - The time for full DFT can be easily estimated from the time for NSC-AITB

Strong Scaling (Atom Si32768)

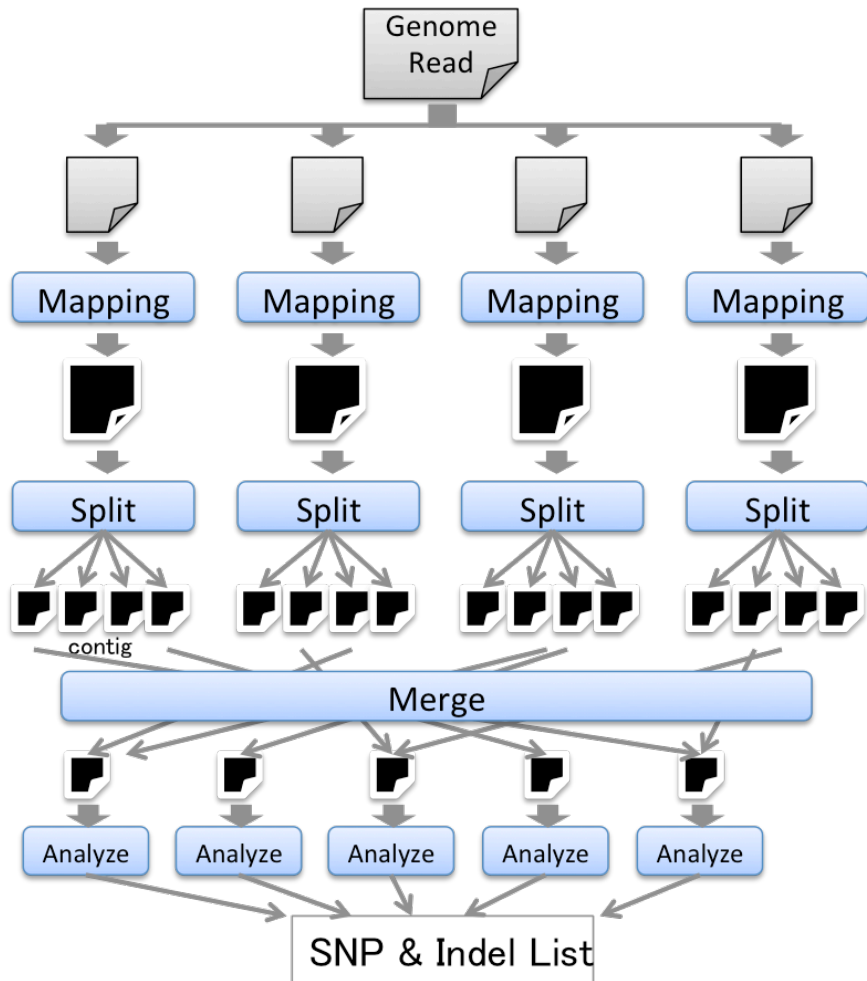
- FX10 (Oakleaf @ U-Tokyo)



NGS Analyzer

- Genome analysis tool for cancer cell's mutation detection
 - Read the output genome data generated by a next-generation genome sequencer
 - Current: 500GB/human
 - In 2020: 100TB/human
 - The analysis pipeline consists of widely used genome analysis software, and performs sequence mapping, genotyping, etc.
 - BWA sequence mapping
 - SAM/BAM formatting software
 - Runs on K computer in parallel

Work Flow of NGS Analyzer

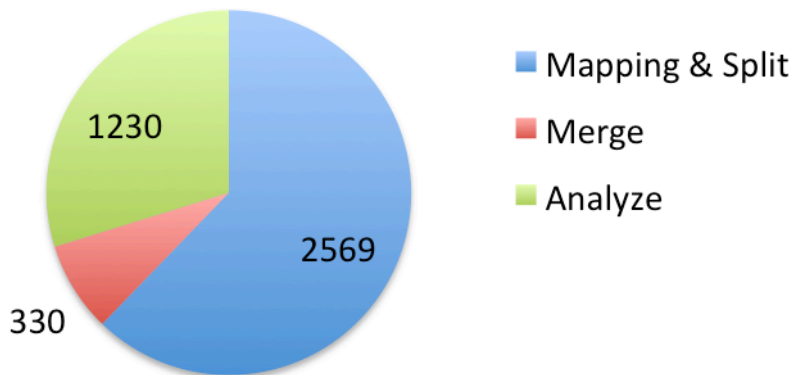


- Workflow
 - Split the input genome read, perform mapping on each of them and then split the results based on contigs
 - Merge mapping results of each contig and run analysis process on each merged data
- The original NGS Analyzer runs this workflow by executing five separated jobs in turn

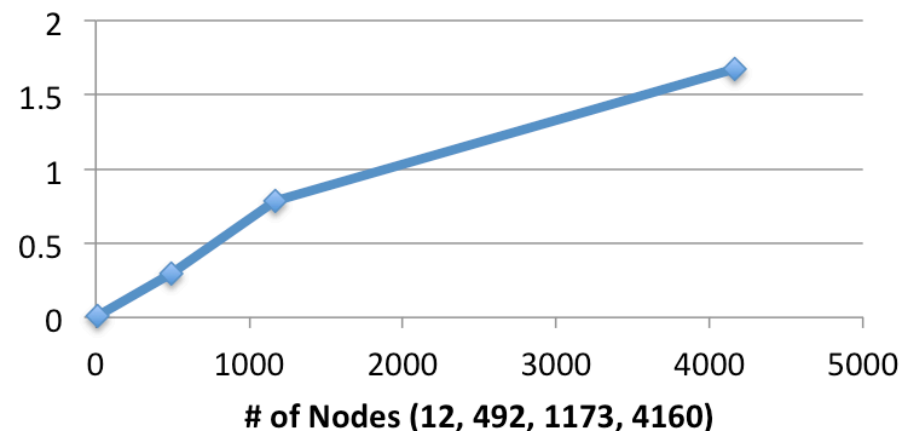
NGS Analyzer Performance

- Execution time for analyzing the whole genome of a Japanese individual : 4,129 sec
 - IO throughput is 1.6 GB/s
(490GB + 874MB + 6TB) / 4129
 - Measured on the K computer using 4,160 nodes

Time in seconds of the whole genome analysis



IO Throughput(GB/s) under various number of nodes (weak scale)



NGS Analyzer Mini App

- Three serial programs that run individual steps of the workflow to measure both computational and IO performance of each step
 - Sequence mapping
 - Duplicate removal of the mapping results
 - Mutation detection
- Single program that runs the entire workflow in parallel to measure the overall performance
- The only miniapp in Fiber that exercises storage systems

Future Plan

- First release
 - <http://github.com/fiber-miniapp>
 - miniapp at riken.jp
- Wider coverage of application domains
 - Miniapps for big-data domains
- Optimization, porting, performance modeling
- Validation of miniapps not changing much of original application behavior
- Will be used to evaluate and design the next-generation exascale machine
- Collaboration among mini-/proxy- app projects