

# Supercomputer Fugaku and QCD Wide SIMD Library (QWS) on Fugaku

Yoshifumi Nakamura

RIKEN

Asia-Pacific Symposium for Lattice Field Theory

Aug. 4-7, 2020

# Disclaimer

- The results obtained on the evaluation environment in the trial phase do not guarantee the performance, power and other attributes of the supercomputer Fugaku at the start of its operation.



- Supercomputer Fugaku
- QCD Wide SIMD Library (QWS)
- Tuning for Fugaku
- Benchmark on Fugaku
- Summary and outlooks

## Acknowledgements

This talk is based on discussion at 2020 with the LQCD codesign team in flagship 2020 project

RIKEN) Y.N., I. Kanamori, K. Nitadori, M. Tsuji

Fujitsu) I. Miyoshi, Y. Mukai, T. Nishiki

Hiroshima) K.-I. Ishikawa

KEK) H. Matsufuru

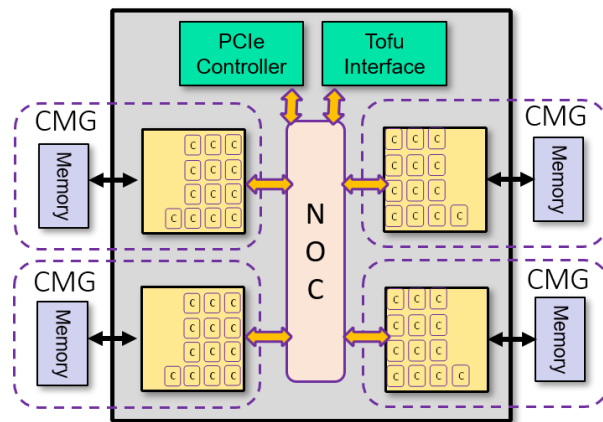
# Supercomputer Fugaku

- **158,976 nodes (A64FX)**
- **432 racks**
  - 396 full nodes (384 nodes) racks
  - 36 half nodes (192 nodes) racks
- **Performance**
  - Normal mode (2.0 GHz)
    - DP(64bit) 488 PFLOPS, SP(32bit) 977 PFLOPS, HP(16bit) 1.95 EFLOPS, Int(8bit) 3.90 EOPS
  - Boost mode (2.2 GHz)
    - DP(64bit) 537 PFLOPS, SP(32bit) 1.07 EFLOPS, HP(16bit) 2.15 EFLOPS, Int(8bit) 4.30 EOPS
- **Memory : 4.85 PiB, 163 PB/s**



# A64FX

Architecture	Armv8.2-A SVE (512 bit SIMD) Fujitsu extension : hardware barrier, sector cache, prefetch
Core	48 (+ 2 assistant cores) 4 core memory group (4 CMG)
TFLOPS	2.0 GHz) DP: 3.072, SP: 6.144, HP: 12.288 2.2 GHz) DP: 3.3792, SP: 6.7584, HP: 13.5168
L1 cache (a 2GHz)	L1D/core: 64 KiB, 4way, 256 GB/s (load), 128 GB/s (store)
L2 cache (at 2GHz)	L2/CMG: 8 MiB, 16way L2/node: 4 TB/s (load), 2 TB/s (store) L2/core: 128 GB/s (load), 64 GB/s (store)
Memory	HBM2 32 GiB, 1024 GB/s
Interconnect	Tofu Interconnect D (28 Gbps x 2 lane x 10 port) 6 Tofu network interface (RDMA engine) 40.8 GB/s (6.8 GB/s x 6)
I/O	PCIe Gen3 x16
Technology	7nm FinFET



Architecture Information  
<https://github.com/fujitsu/A64FX>

# QCD Wide SIMD Library (QWS)

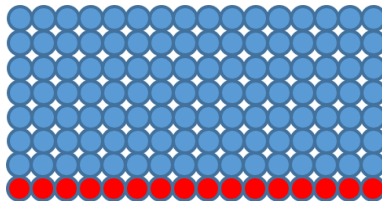
- **Lattice quantum chromodynamics simulation library for Fugaku and computers with wide SIMD, in C and C++**
- **Development**
  - Has been started by Y.N. since 2014 for a benchmark program for “Post-K” supercomputer in Flagship 2020 project
  - Y. Mukai (Fujitsu) joined at 2015
  - K.-I. Ishikawa (Hiroshima) joined at 2015
  - I. Kanamori (Hiroshima -> RIKEN) joined at 2018
- **High performance on Fugaku (Post-K)**
- **Download**
  - <https://github.com/RIKEN-LQCD/qws> (BSD License)
- **Should be used from other QCD applications to get high performance on Fugaku**
  - LDDHMC (K.-I. Ishikawa (Hiroshima), used on K)
  - Grid (P. Boyle (Brookhaven), et al. )
  - Bridge++ (H. Matsufuru (KEK), et al.)
  - BQCD (H. Stüben (Hamburg), T. Haar, Y.N.)

# Tuning for Fugaku

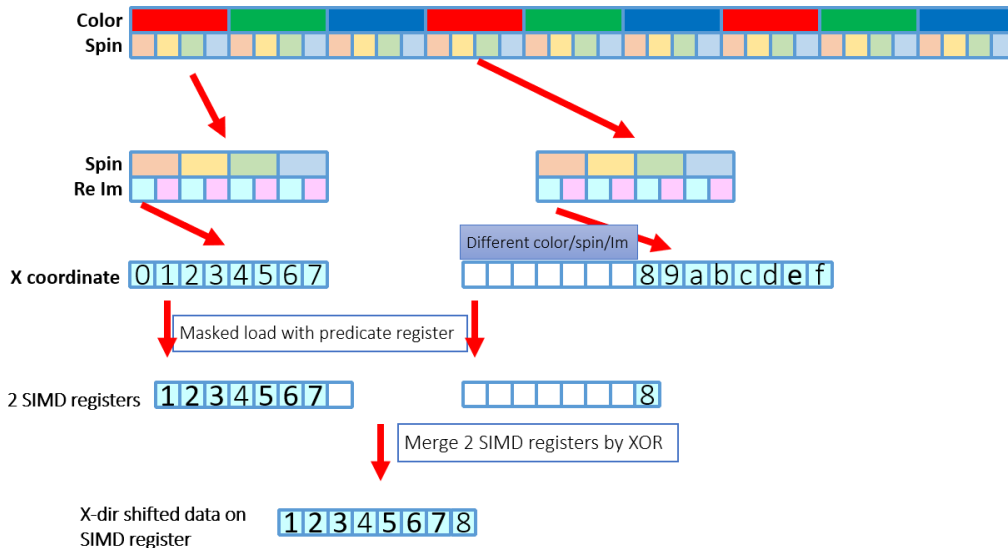
- **SIMD vectorization for 512 bits SIMD**
  - Changing data layout and tuning with Arm C Language Extensions (ACLE) for stencil calculation (**next page**)
- **Removing temporal arrays (unnecessary copy)**
- **Prefetching explicitly 256 Bytes for all arrays by hands**
- **OMP Parallel region expansion : Making omp parallel region is costly, must put “omp parallel” on higher level caller routines (important on many core architecture)**
- **Avoiding narrow memory bandwidth**
  - Increasing reuse of data in low level cache by loop blocking
  - Possible precision changes : double -> single, single -> half
  - Block Krylov subspace method
- **Minimizing communication overhead**
  - Process mapping and double buffering (**Next talk by I. Kanamori**)

- Continuous access except for x-direction (for example)

- Fugaku(FP64):[nt][nz][ny][nx/8][3][4][2][8]
- Fugaku(FP32):[nt][nz][ny][nx/16][3][4][2][16]
- Fugaku(FP16):[nt][nz][ny][nx/32][3][4][2][32]
- cf. K:[nt][nz][ny][nx][3][4][2]



- Tuning with for stencil calculation

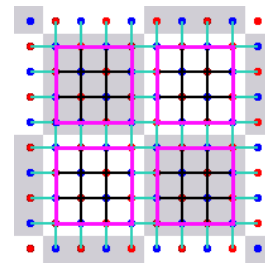
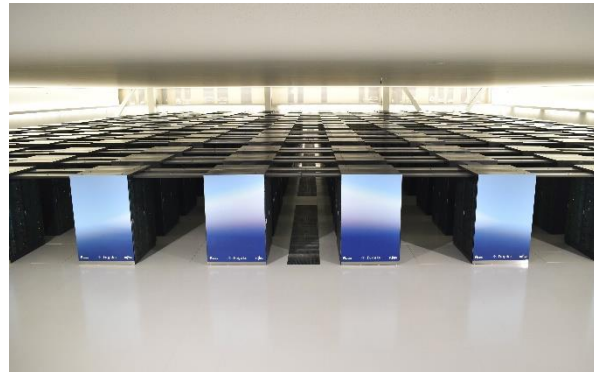


- Useful for general stencil calculation to obtain high SIMD vectorization ratio



# LQCD benchmark test with QWS on Fugaku

- Target problem is  $192^4$  using full system of Fugaku
  - Nodes : 147456
  - MPI processes : 589824 (4 proc / node)
  - OMP threads : 12 / proc
  - Local lattice size :  $32 \times 6 \times 4 \times 3$  / proc
- **Single precision BiCGstab solver ( $Dx=b$ )**
  - Evaluation region in FS2020 project
  - Clover Wilson Dirac operator (D)
  - 5 iteration Schwarz Alternating Procedure (SAP) preconditioning
    - 2 iteration Jacobi solver for inside domain Dirac operator



# 1 node performance (single precision, 2.2GHz)

- Performance (TFLOPS, single peak ratio)
- 4 proc/node
- 2 domains/proc of domain decomposition

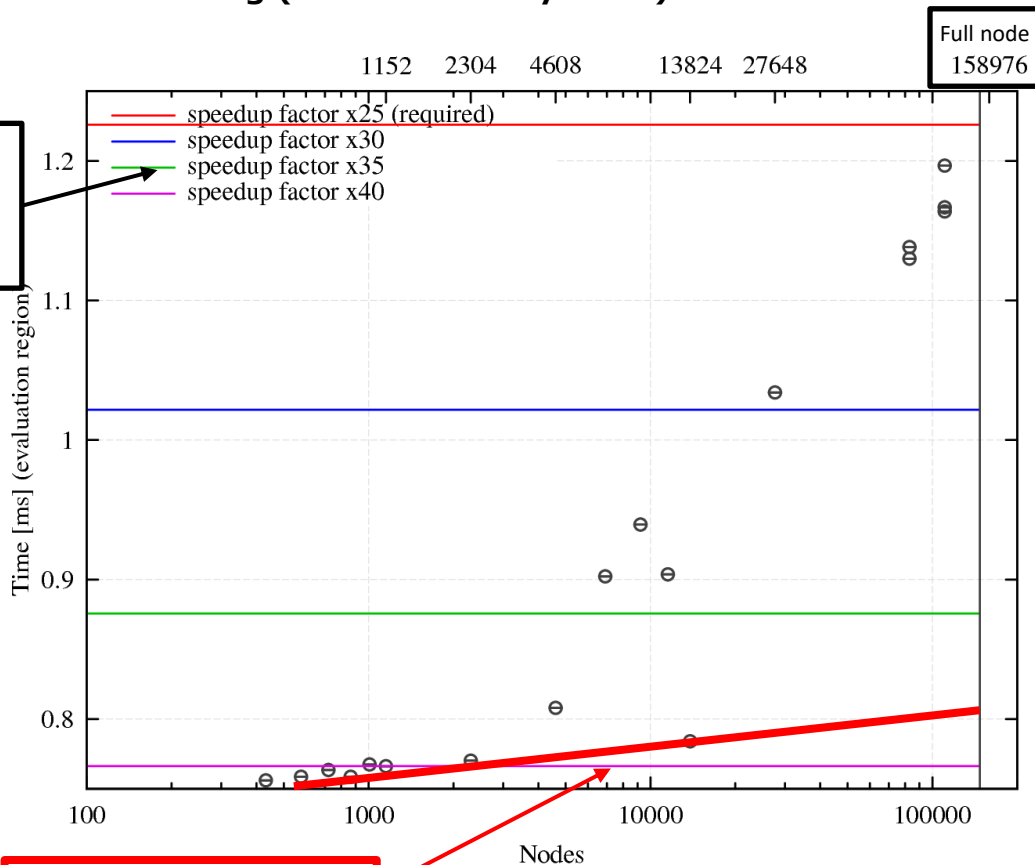
Size / proc	Din	BiCGStab
<b>32x6x4x3 (target size)</b>	<b>1.15 TFLOPS, 17%</b>	<b>0.88 TFLOPS, 13%</b>
<b>32x6x4x6</b>	<b>1.35 TFLOPS, 20%</b>	<b>1.16 TFLOPS, 17%</b>
<b>32x6x8x6</b>	<b>1.51 TFLOPS, 22%</b>	<b>1.05 TFLOPS, 16%</b>
<b>32x6x8x12</b>	<b>1.20 TFLOPS, 18%</b>	<b>0.93 TFLOPS, 14%</b>
<b>32x12x8x12</b>	<b>1.14 TFLOPS, 17%</b>	<b>0.85 TFLOPS, 13%</b>

↑  
On L2\$

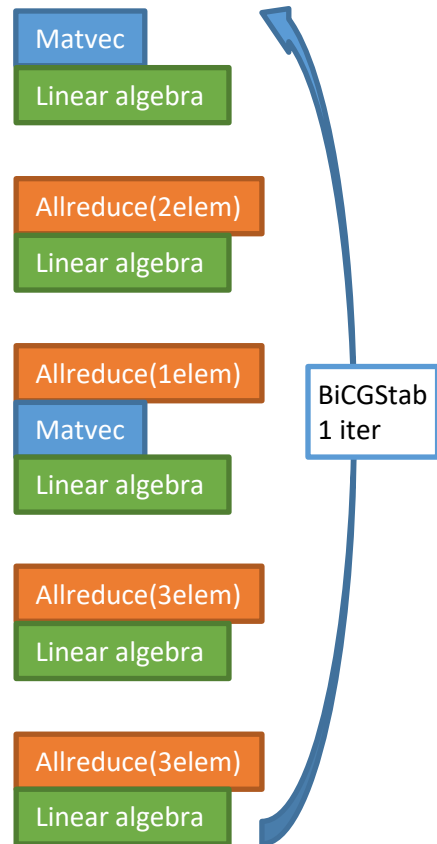
- **Din**
  - bulk Clover-Dirac operator multiplication
  - No nearest neighbor communication

Speedup factor from K with same conditions

- full system use
- problem size
- algorithm

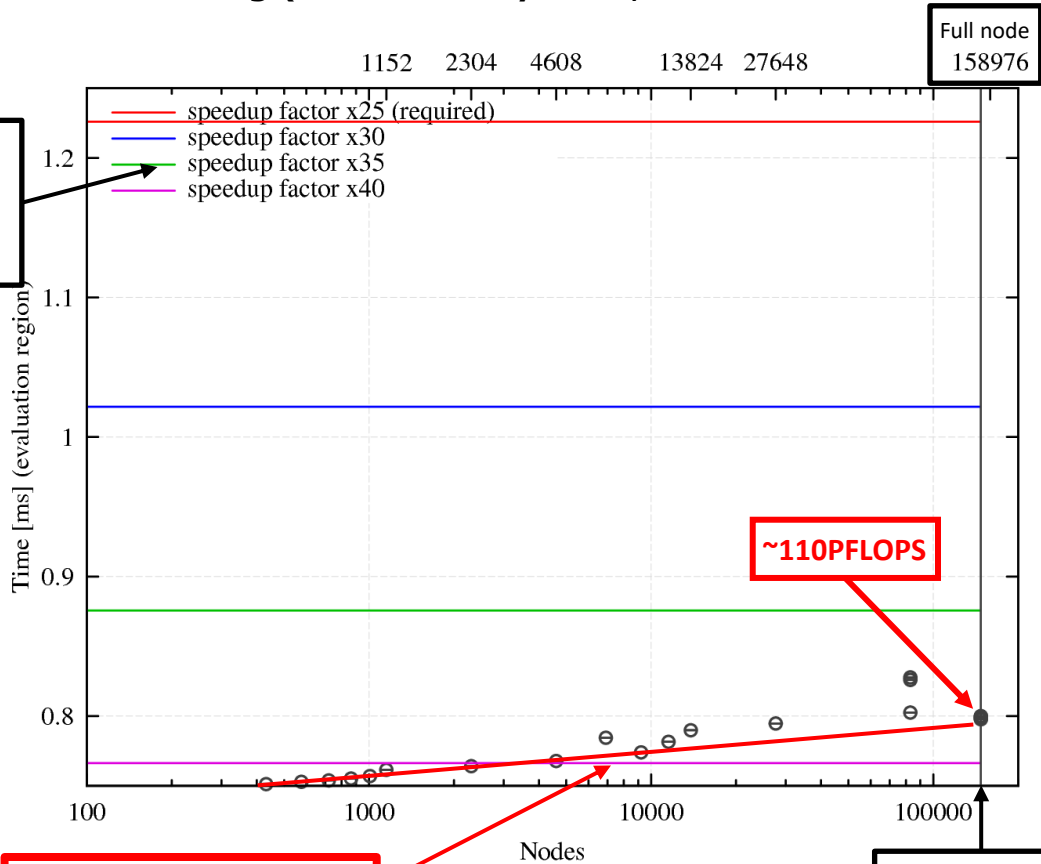


Ideal case of less OS jitter



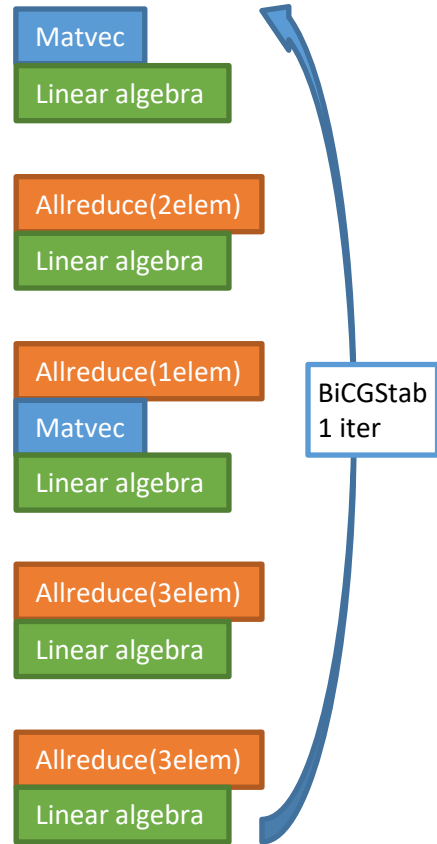
Speedup factor from K with same conditions

- full system use
- problem size
- algorithm



Ideal case of less OS jitter

Maximum nodes (147456 nodes) for 192^4



# Summary and outlooks

- **Supercomputer Fugaku**
  - Installation was completed
  - Tuning system software
- **QWS**
  - Publicly available
  - Confirming performance on Fugaku
- **LQCD Benchmark results on 192<sup>4</sup>**
  - 35+ times faster than the K computer
  - 100+ PFLOPS single precision BiCGStab
- **Plan**
  - Testing with McKernel (Light-Weight Kernel, No OS jitter)