Software development and performance of Fugaku and ARM architectures

Yoshifumi Nakamura (RIKEN)

30 July 2021

The 38th International Symposium on Lattice Field Theory, 26-30 July 2021(US/Eastern), ZOOM/GATHER@MIT

Supercomputer Fugaku

- **new Japanese flagship supercomputer at Kobe RIKEN**
  - Successor of supercomputer K
- **the top on the 4 major high-performance computer rankings for 3 consecutive terms (2021/6, 2020/11, 2020/6)**

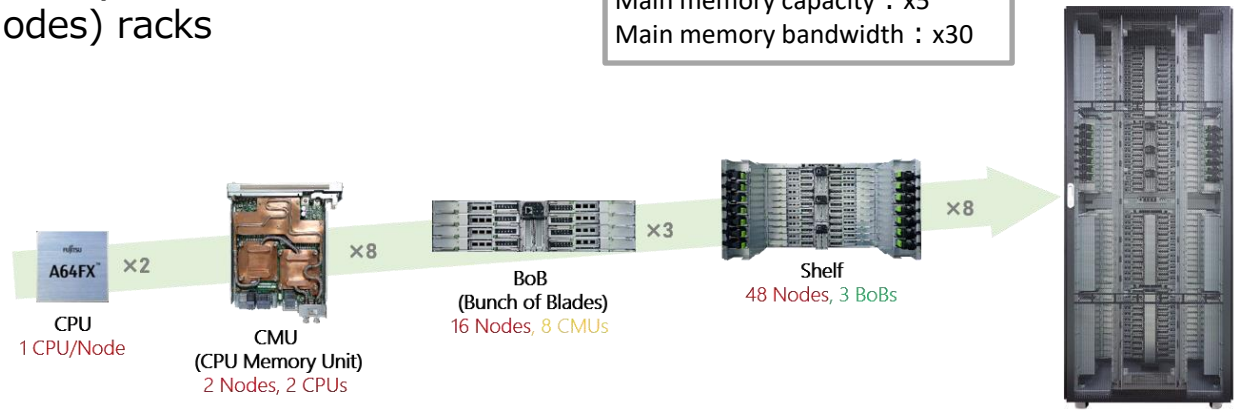
Benchmark (2021/June)	1st	score	2nd	score
TOP500	Fugaku	442.0 PFLOPS	Summit	148.6 PFLOPS
HPCG	Fugaku	16.0 PFLOPS	Summit	2.93 PFLOPS
HPL-AI	Fugaku	2.00 EFLOPS	Summit	1.15 EFLOPS
Graph500	Fugaku	102,950 GTEPS	Taihulight	23,756 GTEPS

- PFLOPS : Peta floating operations per second
- EFLOPS : Exa floating operations per second
- GTEPS : Giga traversed edges per second

Fugaku overview

- **Co-developed by RIKEN and Fujitsu in flagship 2020 project**
- **158,976 nodes (A64FX CPU)**
- **432 racks**
  - 396 (full node rack : 384 nodes) racks
  - 36 (half node rack : 192 nodes) racks
- **Performance**
  - Normal mode (2.0 GHz)
    - Double prec. 488 PFLOPS
    - Single prec. 977 PFLOPS
    - Half prec. 1.95 EFLOPS
    - INT 3.90 EOPS
  - Boost mode (2.2 GHz)
    - Double prec. 537 PFLOPS
    - Single prec. 1.07 EFLOPS
    - Half prec. 2.15 EFLOPS
    - INT 4.30 EOPS
- **Main memory : 4.85 PiB, 163 PB/s**
- **Storage**
  - 1st layer : 1.6TB SSD/16 nodes
  - 2nd layer : 150 PB Fujitsu FEFS (Lustre based file system)
  - 3rd layer : public could, and so on

Versus K (predecessor of Fugaku)
Double prec. performance：x50
Main memory capacity：x5
Main memory bandwidth：x30



Courtesy of FUJITSU LIMITED

A64FX

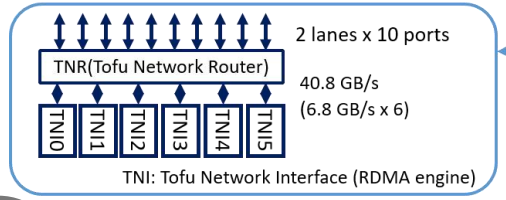
Architecture	Armv8.2-A SVE (512-bit SIMD) Fujitsu extension : hardware barrier, sector cache, prefetch
Core	48 (+ 2 or 4 assistant cores) 4 core memory group (4 CMG)
TFLOPS	2.0 GHz) DP: 3.072, SP: 6.144, HP: 12.288 2.2 GHz) DP: 3.3792, SP: 6.7584, HP: 13.5168
L1 cache (at 2GHz)	L1D/core: 64 KiB, 4way 256 GB/s (load), 128 GB/s (store) B/F=4
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) B/F=2
Memory	HBM2 32 GiB, 1024 GB/s B/F=0.3
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

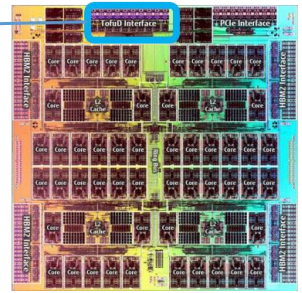
Architecture Information https://github.com/fujitsu/A64FX

Interconnect of Fugaku

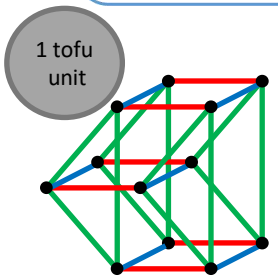
- **Tofu interconnect D**
- **Topology (X,Y,Z,a,b,c) : (24,23,24,2,3,2)**
  - 3D torus by combining XYZabc
- **Fast Allreduce using Tofu barrier**
  - up to 3 floating point, up to 8 integer
- **Link bandwidth : 6.8 GB/s**
- **Injection bandwidth : 40.8 GB/s**
- **Concurrent communications with 6 RDMA engines**
- **Cache injection**
  - Function to write received data directly to L2$, which may reduce L2$ misses

Outcome of co-design activity for LQCD





Courtesy of Fujitsu



QCD Wide SIMD Library (QWS)

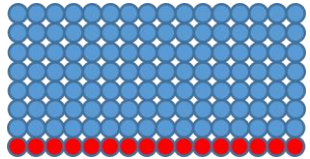
- **Lattice QCD simulation library for Fugaku and computers with wide SIMD, in C/C++**
- **Development with Fujitsu**
  - Has been started by Y. Nakamura since 2014
  - Y. Mukai (Fujitsu) joined since 2015
  - K.-I. Ishikawa (Hiroshima) joined since 2015
  - I. Kanamori (Hiroshima -> RIKEN) joined since 2018
- **Download : https://github.com/RIKEN-LQCD/qws**
- **High performance on Fugaku for Wilson-clover**
  - **SIMD vectorization for 512 bits SIMD**
    - FMA computational latency (Real : 9, Complex : 15 or 16)
    - It is easier to achieve performance with a data layout that separates real and imaginary parts (**RRII layout**)

    Re Re Re Re Re Re Re Re 512 bits SIMD register
    Im Im Im Im Im Im Im Im 512 bits SIMD register

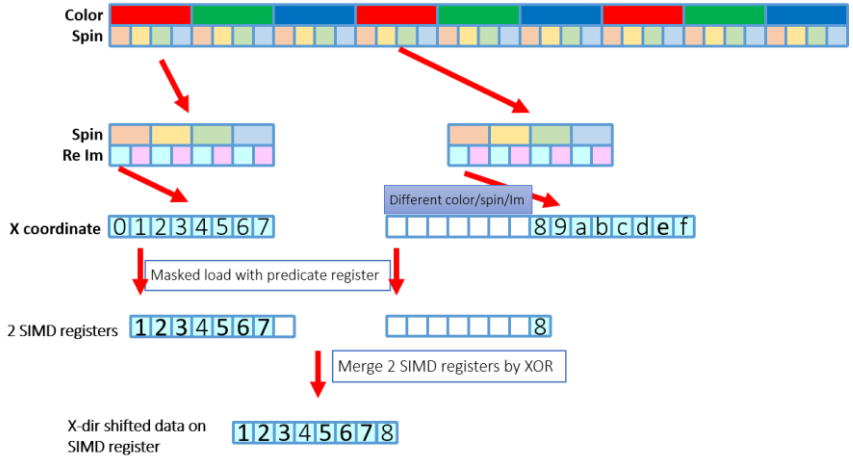
  - Removing temporal arrays, unnecessary copies
  - Manual prefetch for all arrays
  - OMP Parallel region expansion
  - Mixed precision Krylov subspace method
  - Minimizing communication overhead by process mapping and double buffering

Data layout example for Fugaku

Continuous access and 100% SIMD vectorization rate, except for X-direction difference calculation
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]

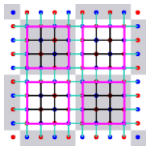


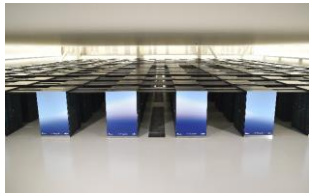
X-direction shift with Arm C Language Extensions (ACLE)



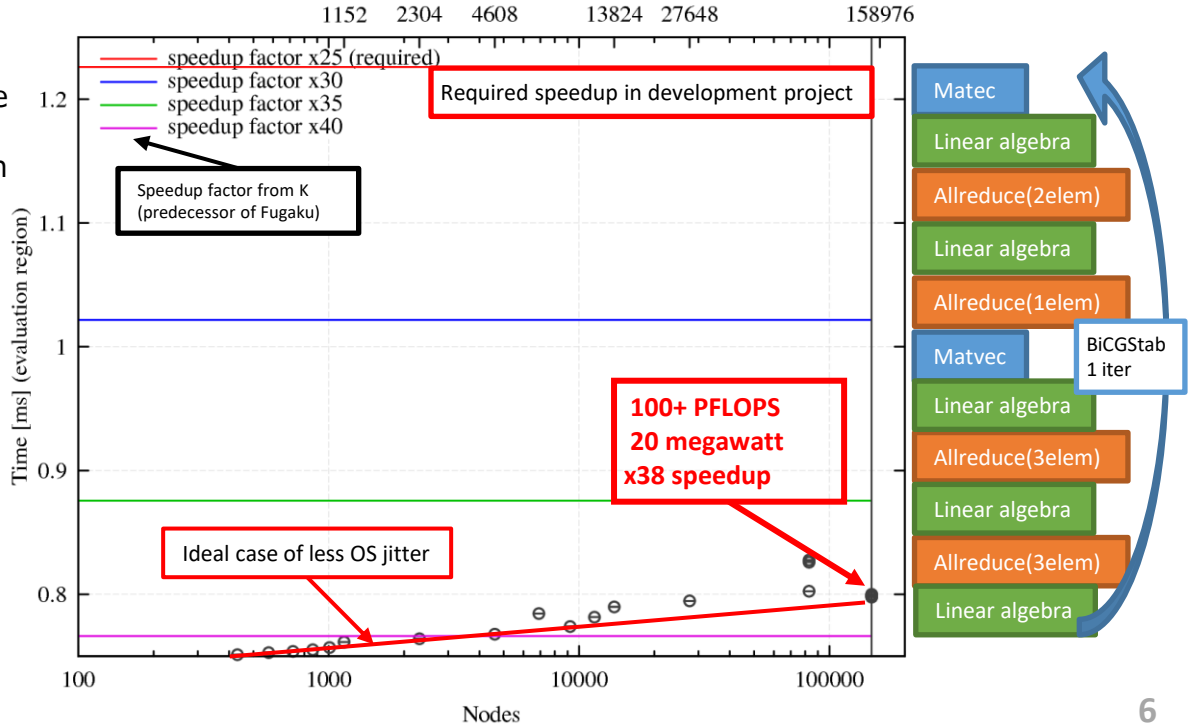
100 PFLOPS single precision BiCGstab on Fugaku with QWS

- **Boost mode (2.2 GHz) and non-Eco mode**
- **147456 nodes (almost full system)**
  - 589824 MPI processes (4 proc/node)
  - 12 OMP threads/proc
- **Target problem size : 192^4**
- **Local lattice size : 32x6x4x3/proc**
- **Single precision BiCGstab solver**
  - Clover Wilson Dirac equation
  - 5 iteration Schwarz Alternating Procedure preconditioning
  - 2 iteration Jacobi solver for inside domain Dirac operator





Sub-millisecond including 4 Allreduces



Node performance (single precision, 2.2GHz)

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

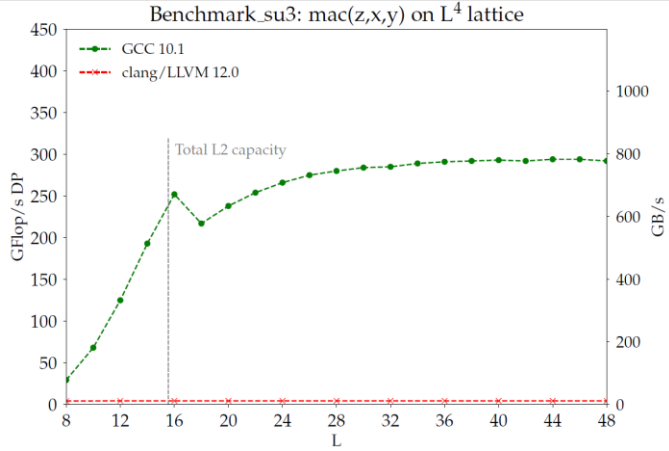
Din : bulk Clover Wilson Dirac operator multiplication, no nearest neighbor communication

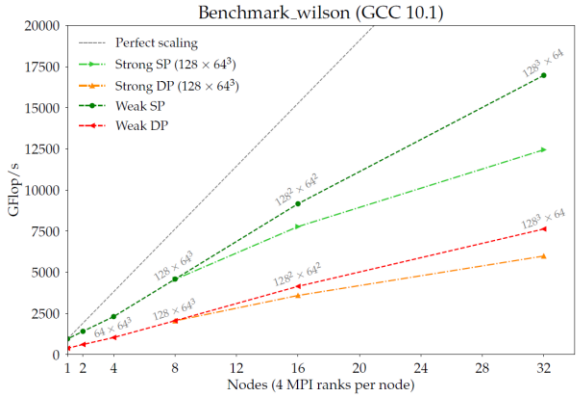
LQCD code development for A64FX, Fugaku

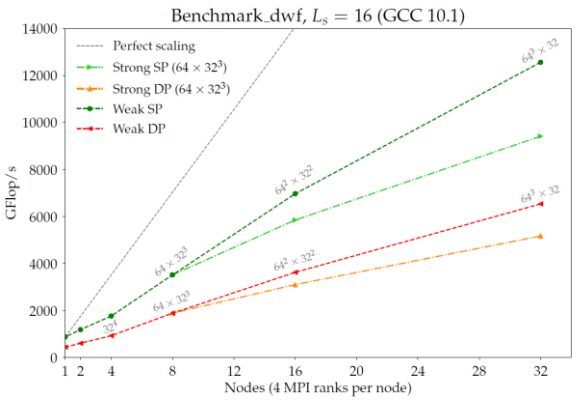
- **Grid : https://github.com/paboyle/Grid**
  - Porting to A64FX by ACLE for Grid's low-level functions, hand optimized Wilson and Domain wall using ACLE and **RIRI layout**, available in upstream Grid develop branch

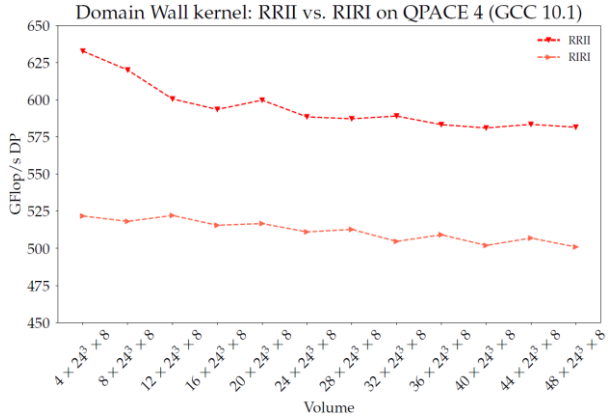
    Re Im Re Im Re Im Re Im
    512 bits SIMD register

  - RRII layout is better than RIRI for DWF
  - MPI scaling improved since APLAT2020









Poster by Nils Meyer (28 Jul 2021, 15:00)

LQCD code development for A64FX, Fugaku

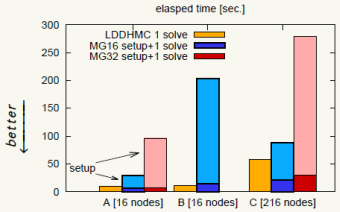
- **Bridge++ : https://bridge.kek.jp/Lattice-code/**
  - Wilson and Domain wall are working
  - QWS is available from Wilson_eo
  - Tuning with ACLE in progress

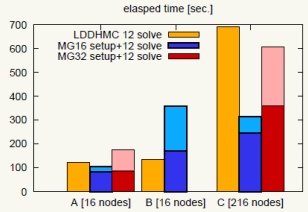
- Multigrid (MG) solver
  - Using QWS as a building block of multigrid solver, a domain decomposition preconditioner as a smoother
  - MG is faster for 12 solves than LDDHMC (reimplementation of QWS) in cases “A” and ”C”
    - Performance (SP) of MG16 in case “C”
    - Smoother 790 GFLOPS/node (from QWS)
    - Coarse solver 91 GFLOPS/node
    - Restrict 82 GFLOPS/node
    - Prolong 125 GFLOPS/node

Performance: elapsed time of 1 or 12 solves

Configurations:

- A: $32^3 \times 64$ lattice, $M_\pi = 156$ MeV S.Aoki et al. [PACS-CS] Phys. Rev. D 79, 034503 (2009)
- B: $64^3 \times 64$ lattice, $M_\pi = 512$ MeV T.Yamazaki et al. Phys. Rev. D 86, 074514 (2012)
- C: $96^3 \times 96$ lattice, $M_\pi = 145$ MeV K.-I.Ishikawa et al. [PACS] LATTICE2015, 075 (2016)





1 solve: setup overhead is large

12 solves: MG solver is faster for light enough quarks (A,C)

- LDDHMC: reimplementation of QWS in Domain Decomposed HMC (nested BiCGStab with Domain Decomposition + SAP + mixed precision)
- MG16, MG32: Multigrid solver with $n_{\mathrm{vec}} = 16, 32$ MG32 for B did not run due to insufficient memory size

I. Kanamori: Lattice 2021, Jul. 2021 7/9

Talk by Issaku Kanamori (28 Jul 2021, 14:30)

Summary

- **I presented the supercomputer Fugaku, code development, and the results of large-scale benchmark tests of single precision solver**
- **Achieved high performance the single precision BiCGstab with QWS**
  - Performance : 100 PFLOPS
  - Efficiency : 10% (single-precision peak ratio)
  - Power consumption : 20 MW
  - Power efficiency : 5 GFLOPS/W

Reference : previous best performance
Sierra@LLNL < ~20 PFLOPS (2018GB finalist)

Reference : power efficiency of LINPACK
Fugaku => 15 GFLOPS/W

- **LQCD code development for A64FX, Fugaku**
  - **Bridge++** : talk by Issaku Kanamori (28 Jul 2021, 14:30)
  - **Grid** : poster by Nils Meyer (28 Jul 2021, 15:00)
  - **BQCD** : https://www.rrz.uni-hamburg.de/services/hpc/bqcd.html by Yoshifumi Nakamura
  - **LDDHMC** : by Ken-Ichi Ishikawa
- **Future work for QWS**
  - **improving the functionality, acceleration by half-precision, McKernel** (lightweight OS without OS jitter), promoting use from other codes, deploying optimization techniques to other codes

- **Information**
  - **Meetings for application code tuning on A64FX computer systems**
    - https://www.hpci-office.jp/pages/e_meetings_A64FX
    - Tuning example for LQCD : gauge field updates by Nakamura (23 December 2020)
      - https://www.hpci-office.jp/pages/e_meeting_A64FX_201223/
  - **Ongoing calls for Japanese supercomputer systems including Fugaku**
    - https://www.hpci-office.jp/pages/e_proposal_submission

Acknowledgments : Fugaku, LQCD co-design team, others involved in the flagship 2020 project