Multigrid Solver on Fugaku

Hideo Matufuru (KEK), Issaku Kanamori (RIKEN), Ken-Ichi Ishikawa (Hiroshima U.)

July 28th/29th, 2021
The 38th International Symposium on Lattice Field Theory (Lattice 2021)

Outline

1. Introduction supercomputer Fugaku and QWS
2. Algorithm and Implementation fermion: Clover
3. Performance
4. Summary and Outlooks

Fugaku and QWS

cf. Y.Nakamura's plenary talk, 30th July

Hardware specification https://www.r-ccs.riken.jp/en/fugaku/about/

- at RIKEN Center for Computational Science, Kobe, Japan
- theoretical peak: 488 PFlops(double), 977 PFlops(single), 1.94 EFlops(half)
- memory: 4.85 PiB
- network: Tofu Interconnect D (28 Gbps x 2 lane x 10 port)
- CPU: a64fx (Armv8.2-A SVE 512bit + Fujitsu extension)
  - 48 cores: 12 cores $\times$ 4 core memory group (CMG)
  - 3,072 GFlops (2.0 GHz, double) ($\sim$ 1 KNL)
  - HBM2 32 GiB, 1024 GB/s ($\sim \times 2$ KNL in BW)

Software

Lattice QCD is one of targets of the co-design with the hardware design

$\Rightarrow$outcome: QCD Wide SIMD library (QWS) https://github.com/RIKEN-LQCD/qws

- nested BiCGstab solver with domain decomposed preconditioner
- +100 PFlops on the almost whole Fugaku Y.Nakamura, APLAT2020
- each part (e.g. preconditioner, mult, etc.) can be called

We can use QWS as a building block of MG solver

we use a domain-decomposed preconditioner in QWS as a smoother

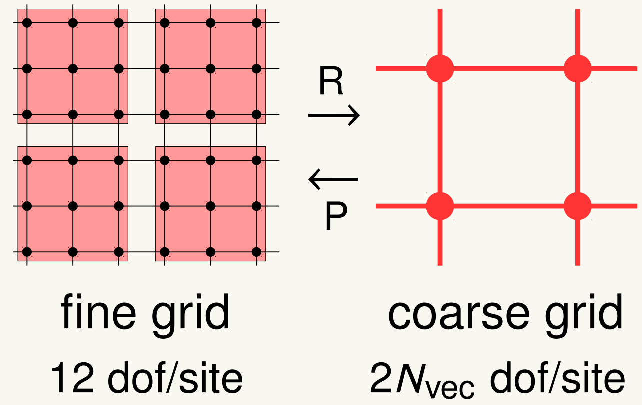
Algorithm and Implementation

application to QCD: R. Babich et al. PRL 105 (2010) 201602
our implementation is based on DDαAMG: A. Frommer et al. , SIAM J. Sci. Comput. 36 (2014) A1581

Multigrid solver is very efficient when the quark mass is small

Multigrid steps

- used as a preconditioner
- Restriction (R):
  fine grid (original lattice) $\to$ coarse grid
- Coarse grid solver: solve the coarse system
- Prolongation (P): coarse grid $\to$ fine grid
- Smoother: improve the solution in the fine grid



Our implementation

- 2-level multigrid, single prec.
  - coarse solver: BiCGStab
  - post smoother: multiplicative Schwarz Alternating Procedure (SAP)
    inner: Jacobi iteration
  - outer solver: Flexible BiCGStab
- setup: generate N_{vec} null space vectors
  initial SAP + 4 times adaptive MG preconditioner

cf. DDαAMG : uses (e/o)[F]GMRes instead of [F]BiCGStab

Algorithm and Implementation: Domain size

domain size for restriction/prolongation $\neq$ domain size for SAP

- domain decomposed algebraic restriction/prolongation
  O(10)–O(100) domains (=coarse sites) in each process

  domain size: $8 \times 4 \times 4 \times 4$ some constraints due to SIMD usage [next slide]

- SAP: 2 domains [fixed] in each process
  implementation in QWS:
  - 2 domains in x-direction
  - x-extent of each domain must be multiple of 16 a constraint due to SIMD usage
    [x-extent of the local volume must be multiple of 32]

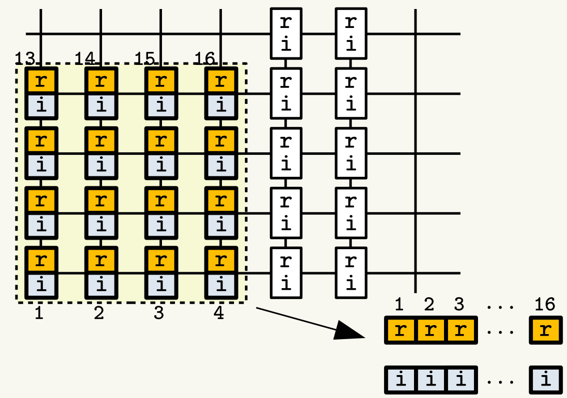
Algorithm and Implementation: Implementation details

code base: Bridge++ 

https://bridge.kek.jp/Lattice-code/

SIMD variables (scalable vector extension, SVE) width: 512 bits

- real and imaginary parts: assigned to different vector variables
- site degrees of freedom: packed in SIMD vector with 2-dim tiling in x-y (SAP in QWS: 1-dim tiling in x)
- the SIMD tiling divides the domain size, and the coarse lattice uses the same tiling
  $\Rightarrow$on the coarse grid, O(# site/SIMD) = O(# thread)



Communication

- MPI persistent communication with Fujitsu extension: uses assistant cores to accelerate the communication

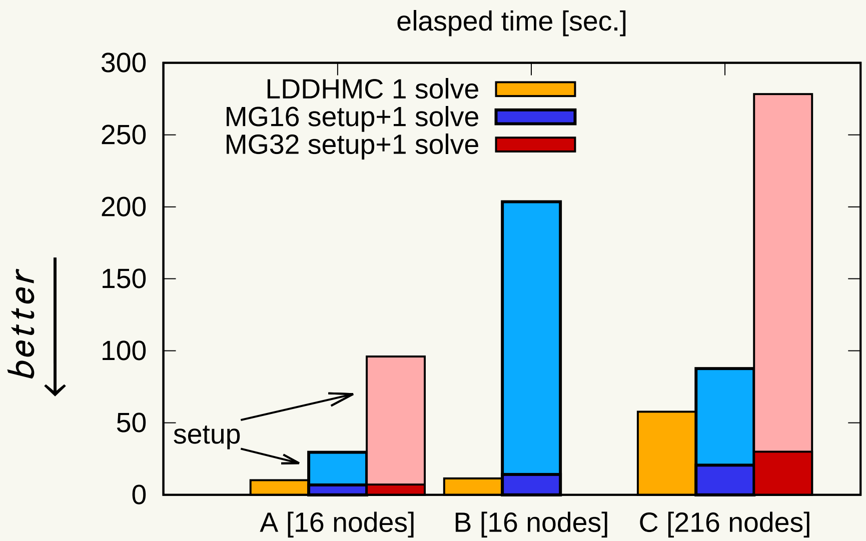
Smoother employs QWS https://github.com/RIKEN-LQCD/qws

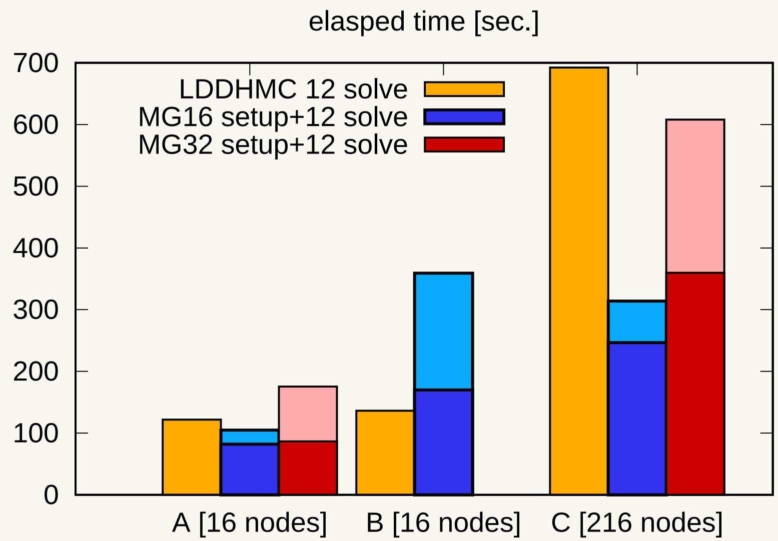
- multiplicative SAP (inner: Jacobi iterations)
- communication: uses low level API (uTofu), double buffering

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)



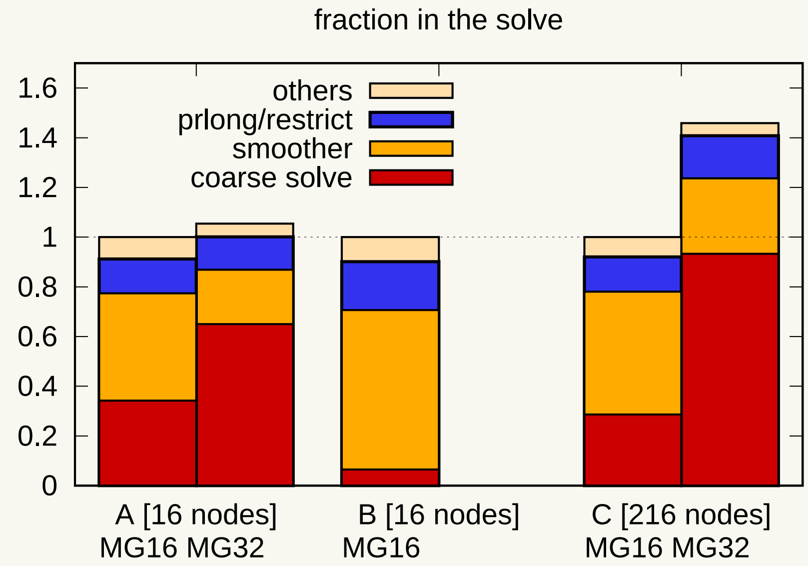


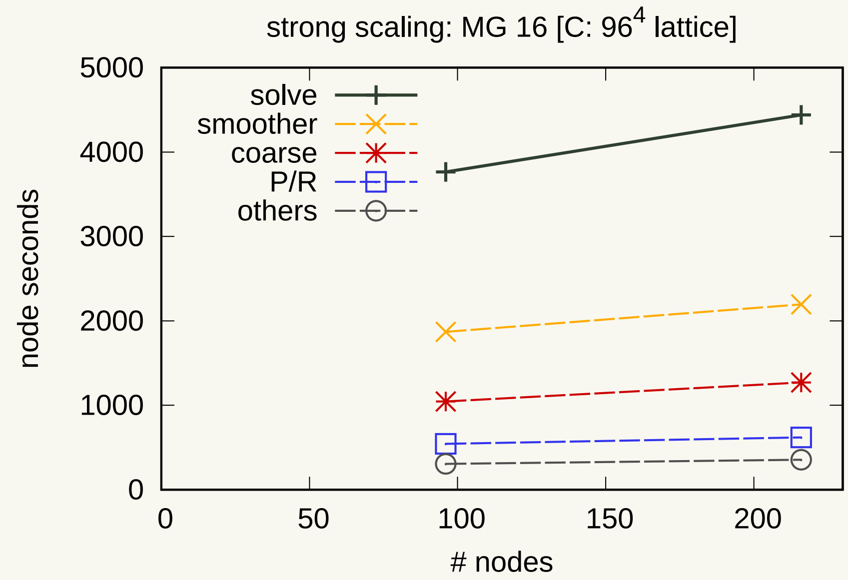
1 solve: setup overhead is large

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 $n_{\text{vec}} = 16, 32$ MG32 for B did not run due to insufficient memory size

Performance: timing fraction





relative fraction normalized by solving time of MG16

reasonable strong scaling

Flops (single prec.): Config. C on 216 nodes, MG16

smoother	790 GFlops/node	SAP from QWS is very efficient
coarse solver	91 GFlops/node	
restrict	82 GFlops/node	
prolong	125 GFlops/node	

Fugaku 2.0 GHz: 6,144 GFlops/node (single prec)

Summary and Outlooks

- Efficient implementation of Multgrid solver on Fugaku is ready

  will be publicly available in Bridge++
- Example of using QWS, an outcome of co-design activity for Fugaku
- Practically, 2-level algorithm is enough
- [further improvement in the coarse solver and the setup is underway]

Summary and Outlooks

- Efficient implementation of Multgrid solver on Fugaku is ready

  will be publicly available in Bridge++
- Example of using QWS, an outcome of co-design activity for Fugaku
- Practically, 2-level algorithm is enough
- [further improvement in the coarse solver and the setup is underway]

Acknowledgments

- Computational resource: Fugaku (RIKEN Center for Computational Science), Flow (Nagoya U.)
- Grants: JSPS KAKENHI(20K03961, 19K03837), MEXT as “Program for Promoting Researches on the Supercomputer Fugaku” (Simulation for basic science: from fundamental laws of particles to creation of nuclei), “Priority Issue 9 to be Tackled by Using Post K Computer” and Joint Institute for Computational Fundamental Science
- Configurations: Japan Lattice Data Grid
- Bridge++ members and LQCD co-design team in flagship 2020 project