Status of HMC algorithm for Fugaku and possible problems with extreme parallelism

Ken-Ichi Ishikawa (Hiroshima U.)

For WG9 codesign

HMC algorithm (on K-computer)

- Combines
  - Luscher's Domain-decomposed HMC algorithm
  - + Multi-mass preconditioning for u/d quarks
  - UV-filtered Polynomial HMC for strange quark
  - Stout smearing
  - Full link update + Random shift of origin of lattice
- Written in
  - Fortran 90 with object oriented style for Double precision part
- Quark solver
  - U/D quark solver: Outer/inner mixed precison BiCGstab + Crhonological guess (Fortran90)
    - inner : (C/C++) DD-preconditioned BiCGStab single precision with [K computer SIMD intrinsics]
- Communication (Tofu)
  - rdma_comlib.c : wrapper to
  - MPI Fujitsu Extension (FJMPI) for One-sided comm. Put

Status HMC algorithm on Fugaku

- Combines
  - Luscher's Domain-decomposed HMC algorithm
  - + Multi-mass preconditioning for u/d quarks
  - ~~UV-filtered Polynomial~~ Rational HMC for strange quark
  - Stout smearing
  - Full link update + Random shift of origin of lattice
- Written in
  - Fortran 90 with object oriented style for Double precision part
- Quark solver
  - U/D quark solver: Outer/inner mixed precison BiCGstab + Crhonological guess (Fortran90)
    - inner : (C/C++) DD-preconditioned BiCGStab single precision with [ARM ACLE SIMD intrinsics (QWS)] will be called. (QWS status => talk by Y.Nakamura)
- Communication (TofuD)
  - rdma_comlib.c : wrapper to
  - ~~MPI Fujitsu Extension (FJMPI) for One-sided comm. Put~~ Fujitsu low level comm. API uTofu. (comlib status => talk by I. Kanamori)

Status HMC algorithm on Fugaku

- Stauts:
  - ☹ Compile without QWS and uTofu (Fortran90 part) :
    Not yet done. My duty for next year.
  - ☹ Marge QWS : Interface arrangement is needed.
  - ☹ Marge uTofu : I think it is trivial.

HMC algorithm/solver on huge lattices/ huge nodes

- Concerns
  - Experiences on clusters (KNL, KNC, PC-clusters)
    - Communication timing suffers from OS jitter (other processes) => Minimize OS/background system processes on compute nodes.
    - MPI tag overflow : do not use individual tags for each communication. Check the MPI manual by bender where various limitations are described.
  - Experiences on K-computer
    - Integer(4), int32 could overflow when lattice sites or dof on whole lattice => Use Integer(8), int64 or real(8), double to count.
    - MPI tag overflow : do not use individual tags for each communication.
- How about on Fugaku?

HMC algorithm/solver on huge lattices/ huge nodes

- Concerns
- How about on Fugaku?
  - Target Lattice size (Mile stone) : 192^4 at physical quark mass, 1/a=2—3 GeV.
  - DoF Count overflow?
    - DoF Link :

      192^4 x 4 x3x3= 3*3*5435817984 = 48922361856 > 4294967296= 2^32
    - DoF Spinor:

      192^4x3x4 = 16307453952
    - These numbers are still smaller than 2^64 = 18446744073709551616.
  - Energy conservation and accuracy of dH for the Total Hamitonian in HMC.

HMC algorithm/solver on huge lattices/ huge nodes

- Concerns
- How about on Fugaku?
  - Target Lattice size (Mile stone) : 192^4 at physical quark mass, 1/a=2—3 GeV.
  - Energy conservation and Accuracy of dH for the Total Hamitonian in HMC.
    - Total Energy H : Intensive Variable $\propto$ Volume

$$H \propto \ L^4 \ (Gauge + PseudoFermions)$$

    - Energy conservation with a high acceptance in HMC requires

$$H(\tau = 1) - H(\tau = 0) = \Delta H \simeq 1$$

    - Loss of significant digits will be a concern with a finite precision math (cancellation between large numbers)

HMC algorithm/solver on huge lattices/ huge nodes

- Energy conservation and Accuracy of dH for the Total Hamitonian in HMC.
  - Total Energy H : Intensive Variable $\propto$ Volume

$$H \propto L^4 \, (Gauge + PseudoFermions)$$

  - Energy conservation with a high acceptance in HMC requires

$$H(\tau = 1) - H(\tau = 0) = \Delta H \simeq 1$$

  - Loss of significant digits will be a concern with a finite precision math (cancellation between large numbers)
    - L^4 = 192^4 = 1358954496 = 1.35 e^9
    - Gauge , PS dof/lattice = 3x3x4x2=72 reals , 3x4x2 = 24 reals, several PS fields 3– 4 PS fields. => Total DOF/lattice 100—200 reals.
    - H = 1.35*10^9 * 200 = O(10^11) – O(10^12)
    - In D.P. ΔH will have 4 – 5 significant digits. It still seems to be OK with double precision.

HMC algorithm/solver on huge lattices/ huge nodes

- OS / System process jitter
  - Algorithmic workarounds
    - Communication Avoiding algorithms + overlap between computation
      - Unrolled + Reordered CG algorithm, Unrolled + Reordered BiCG algorithm :
        - B. Krasnopolsky, “The reordered BiCGStab method for distributed memory computer systems”, In International Conference on Computational Science, ICCS 2010, Procedia Computer Science 001 (2012) 213–218.
        - Reduces synchronization points (global reduction, largely suffered from the jitter)
        - Tradeoff: Increased arithmetic.
      - Halo/Ghost sites extension (For Stencil like Wilson-Dirac op)
        - This also reduce synchronization between nearest neighbor nodes
        - Tradeoff: Increased arithmetic and memory size
      - Double buffering for nearest neighbor comm.
        - Reduce the completion of send/recv wait time ← We will apply this with uTofu
  - System software
    - Use Dedicated OS on compute nodes/cores => McKernel on Fugaku

HMC algorithm/solver on huge lattices/ huge nodes

- How to use high parallelism
  - Process Parallel + Core Parallel + SIMD Parallel
  - We will put 4 MPI procs on a node (1 node = 4 CMG , 1 MPI Proc / CMG)

  - MPI Parallel : Used for domain decomposition, regular grid decompsition, commonly used in LQCD apps.
  - Core Parallel : OpenMP : Used for site loops on local lattice sites.
  - SIMD Parallel, Vector :
    - Ancient Vector machines : Used for site loops on local lattice sites. But this is replaced with OpenMP.
    - K-Comuter : SIMD = 2 elements DP. => One SIMD vec is mapped to a Complex Real and Imag.
    - PC Cluster (Intel) : SIMD 4 elements SP => Color, complex Real, Imag or Spin Real,Imag
    - KNL,KNC(Intel) : AVX512, SIMD 16 elements SP => Spin Real,Imag + Sites
    - Grid : Sites (Talk by P.Boyle)
    - Fugaku : ARM SVE SIMD 16 elements SP => Real,Imag+Sites (QWS)

HMC algorithm/solver on huge lattices/ huge nodes

- How to use high parallelism
  - Process Parallel + Core Parallel + SIMD Parallel
  - We will put 4 MPI procs on a node (1 node = 4 CMG , 1 MPI Proc / CMG)

  - MPI Parallel : Used for domain decomposition, regular grid decompsition, commonly used in LQCD apps.
  - Core Parallel : OpenMP : Used for site loops on local lattice sites.
  - SIMD Parallel, Vector :
    - Fugaku : ARM SVE SIMD 16 elements SP => Real,Imag+Sites (QWS)
    - Fugaku : ARM SVE SIMD 32 elements HP => ?? (QWS)
  - Long SIMD length ⇔ OpenMP local lattice loop parallelism
  - Lack of parallelism in a MPI process.
  - How to extract the parallelism? Any Ideas? (Aiming for strong scaling)

Thank you!