A performance evaluation of CCS QCD Benchmark on Intel Xeon Phi (KNC) systems

Ken-Ichi Ishikawa (Hiroshima U.)

In collaboration with

Taisuke Boku (CCS, U. of Tsukuba)
Yoshinobu Kuramashi (CCS, U. of Tsukuba)
Lawrence Meadows(Intel Corporation)
Michael D'Mello(Intel Corporation)
Maurice Troute(Intel Corporation)
Ravi Vemuri (Intel Corporation)

This work is supported by the Intel Parallel Computing Center at Center for Computational Sciences (CCS), University of Tsukuba

Plan of My Talk

1. Introduction
   - COMA (Intel Xeon Phi(KNC) System at Center for Computational Sciences (CCS), University of Tsukuba
   - CCS QCD Solver Benchmark (CCS-QCD) program
2. Tuning CCS-QCD for COMA system
   - MPI communications and reverse offloading
3. Results
4. Summary

1. Introduction

- Lattice QCD simulations require a lot of computational resources.
- Important that collaborative reserch among Lattice Field theory ,HPC system scientists and developers.

- Center for Computational Sciences (CCS), University of Tsukuba is one of the place where such a collaboration takes place in Japan.
  - CP-PACS(1996), PACS-CS(2005), HA-PACS(2011), ...

- CCS installed PACS-IX system named "COMA" 2014.
  and CCS is one of Intel Parallel Computing Center program.

- In this talk, I will show
  - Performance tuning and evaluation of the QCD quark solver for COMA system.

1. Introduction

http://www.ccs.tsukuba.ac.jp/eng/research-activities/supercomputers/

- **COMA (PACS-IX) at CCS**
  - **Computational node**
    - CPU x 2 : Intel Xeon E5-2670v2
    - MIC x 2 : Intel Xeon Phi 7110P (KNC)
    - MEM : CPU=64GB, MIC=16GB (8GB x 2)
    - NET : IB FDR Full-bisection b/w Fat Tree
  - **Total Nodes : 393 nodes**
  - **Peak Perf. :**
    - CPU=157.2 TFlops, MIC=843.8 TFlops
    - TOT = 1.001 PFLOPS
  - **Typical KNC system.**
- ***Equipped with Xeon Phi (KNC) accelerators.***

	Year	Name	Performance
I	1978	PACS-9	7 KFLOPS
II	1980	PACS-32	500 KFLOPS
III	1983	PAX-128	4 MFLOPS
IV	1984	PAX-32J	3 MFLOPS
V	1989	QCDPAX	14 GFLOPS
VI	1996	CP-PACS	614 GFLOPS
VII	2006	PACS-CS	14.3 TFLOPS
VIII	2012	HA-PACS	802 TFLOPS
IX	2014	COMA	1.001 PFLOPS

- **To derive a high efficiency for Lattice applications, tuning for KNC system is required.**

1. Introduction

http://www.ccs.tsukuba.ac.jp/eng/research-activities/supercomputers/

- To derive a high efficiency for Lattice applications, tuning for KNC system is required.
- Lattice QCD on KNC systems?
  - A lot of work has been done on the tuning for KNC systems

B. Joó et al, ISC 2013.
Simon Heybrock et al, Supercomputing 2014, SC '14 Proceedings [arXive:1412.2629].
Hwancheol Jeong et al., PoS (LATTICE 2013) 423 [arXive:1311.0590].
Paul Arts et al., PoS(LATTICE2014) 021 [arXiVe:1502.04025].
Denis Barthou et al, CCP2013.
O. Kaczmarek et al., PoS(LATTICE2014)044 [arXive:1409.1510].
Yida Wang et al, SC '15 Proceedings.
Ruizi Li and Steven Gottlieb, PoS(LATTICE2014)034 [arXive:1411.2087]
Peter Boyle, Azusa Yamaguchi, Guido Cossu, Antonin Portelli, Lattice 2015 [arXive:1512.03487]
and many...

We also tune the QCD quark solver for COMA system.

1. Introduction

- **CCS QCD Solver (CCS-QCD) Benchmark Program**

  http://www.ccs.tsukuba.ac.jp/eng/research-activities/published-codes/qcd/

  - **A simple Wilson-Clover Fermion solver.**
  - **Written in Fortran90.**

$$Dx = b$$

  - **is solved with the even/odd-site preconditioned BiCGStab solver.**
  - **Timing and FLOPS are counted to benchmark.**
  - **Can be used**
    - **to analyze a new HPC architecture.**
    - **to develop a new algorithm.**

We use CCS-QCD as the tuning target for COMA system

  - ***How to tune CCS-QCD for COMA (KNC) system?***
  - **In this talk, I will focus on the tuning of CCS-QCD for COMA system.**

2. Tuning CCS-QCD for COMA system

- **Algorithm**
  - **We extend CCS-QCD to the mixed precision solver by adding single precision BiCGStab.**
    - **Outer (DP) Flexible-BiCGStab**
    - **Inner (SP) Even/odd-site & RAS-DD Preconditioned BiCGStab**
  - ***Benefit from mixed precision solver***
    - **Memory and Cache usage /2 and bandwidth x 2**
    - **SP performance = DP perf x 2.**
    - **Minimum change in the DP (Fortarn90) part source program.**
  - ***Offload* the entire SP solver to Xeon Phi (KNC)**
    - **SP part: Intel C/C++ compiler with SIMD intrinsics (_mm512_*_ps).**
    - **MIC : SIMD 16 (SP), FMA**
- **Performance Issues :**
  - ***SIMD vectorization, Cache and Prefetching, OpenMP threading, MPI communications, Offloading.***

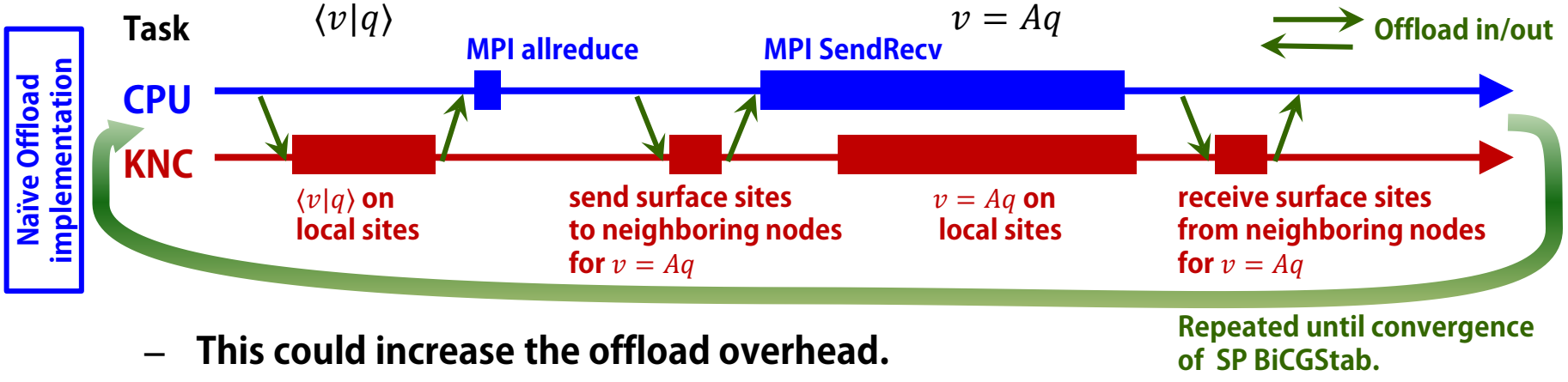
2. Tuning CCS-QCD for COMA system

- **Hopping matrix tuning**
  - **SIMD vectorization**
    - SP SIMD 16
    - Intel Compiler (C/C++) intrinsics (__m512 zmm, _mm512_*_ps)
  - **Loop tiling and Thread manager**
    - 4D loop in HT,Z,Y,X loop (HT=T/2 even/odd-prec.) is tiled
    - Tile size : (8,4,2,2)
    - These tiles are distributed the Xeon Phi cores (60 cores x 4 HT).
    - The works (Tiles) are assigned to OpenMP threads with a thread scheduler to reduce the load imbalance.
  - **Prefetch insertion**
    - Prefetch intrinsics (_mm_prefetch) are inserted in the loop in the tile.
    - The addresses to be prefetched are prepared dynamically in list vectors in each tile.
    - The performance does not suffer from the use of the list vector. List vector simplifies the address prediction in the 4-D site loop.
    - Prefetch is very important on KNC system.

Importance of SIMD vectorization, cache utilization with prefetching, and thread assignment including HT to each core.
These issues have been well discussed in the literature. I skip the details of these issues.

Instead I will focus on the MPI communications and reverse offloading.

- **MPI communications and reverse offloading**
  - There are three system modes for KNC system.
    1. Native mode : Use only Xeon Phi (KNC) codes. Can run a program complied for KNC native mode.
    2. Offload mode : Use both of CPU and Phi. The CPU code launces KNC regions indicated by compiler directives.
    3. Mixed mode : Use both of CPU and Phi. CPU codes and native codes are running on the system. CPU runs CPU codes, Phi runs KNC codes.
  - We employ Offload mode to separate DP(Fortran90) part and SP(C/C++) part. This simplifies the code structure.
  - We cannot use MPI functions within the offload region.

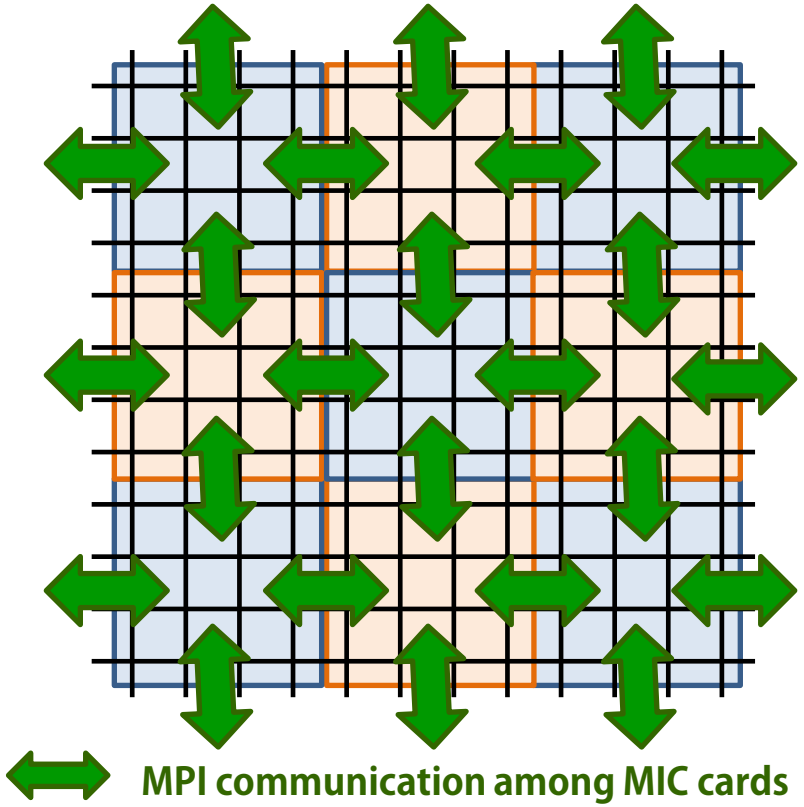


  - This could increase the offload overhead.

Reverse offloading technique removes the offload overhead almost completely.

2. Tuning CCS-QCD for COMA system

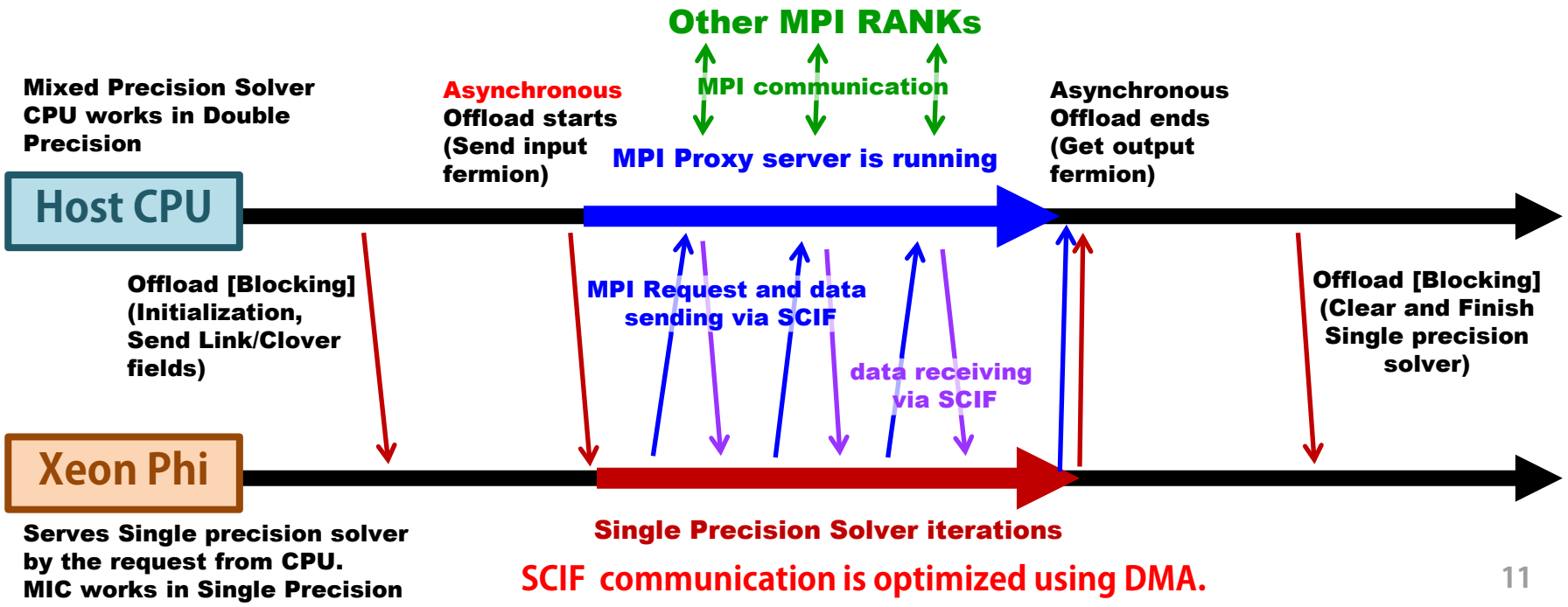
- **Reverse offloading**
  - offload MPI tasks to CPU form KNC.
  - MPI cannot be used in the offload region (on KNC).
  - We can use *SCIF* interface among CPU and KNC.
    - *SCIF* is a low level communication API provided by Intel.
  - CPU hosts the proxy of MPI.
  - KNC send the MPI comm. requests to CPU via SCIF interface.
  - We can put the single precision solver entirely in a single offload region.
  - As a byproduct, communication and computation overlapping is also efficiently implemented. CPU works on MPI communication, KNC on computation.



2. Tuning CCS-QCD for COMA system

- **Reverse offloading**
  - At the beginning of the DP solver
    - Start up KNC card , initialize the SP solver,
    - prepare SP link/clover data and send them to KNC card
  - At the SP solver in the outer DP solver
    - Send SP fermion data and receive resulting SP fermion data (using offload pragma).
    - Asynchronously offload the SP solver.
    - Run MPI proxy on CPU. The proxy can handle; MPI_Allreduce, MPI_Sendrecv for NNB comm.
  - At the end of the DP solver
    - Finalize the SP solver and KNC card

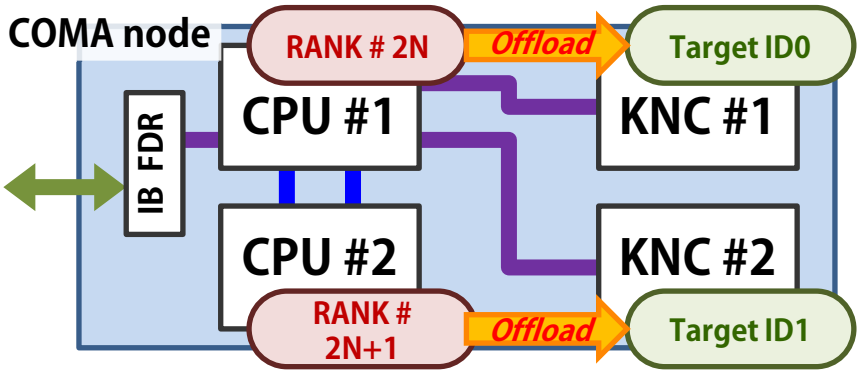
The effects of the reverse offloading and communication overlapping are benchmarked.



SCIF communication is optimized using DMA.

3. Results

- **We measure the performance (Timing, GFLOPS) of**
  - **Mult : the timing consists of**
    - **Mult_pre : surface data packing and send to the MPI proxy on running CPU**
    - **Mult_in : Wilson-Clover Dirac matrix multiplication on interior sites**
    - **Mult_pst : receive surface data from the MPI proxy and do matrix multiplication on surface sites**
    - **MPI comm.: overlapped on Mult_in**
  - **Mult_RAS :**
    - **C.-W. D. matrix multiplication on extended domain. No MPI comm. is needed.**
  - **Entire SP solver One MPI process uses one KNC card.**
  - **COMA : Two KNC (Xeon Phi 7110P) cards / node.**
  - **Assign two MPI processes on a COM node.**
  - **Assign offload target ID from the MPI RANK to couple one CPU process and one KNC card on the same node statically.**

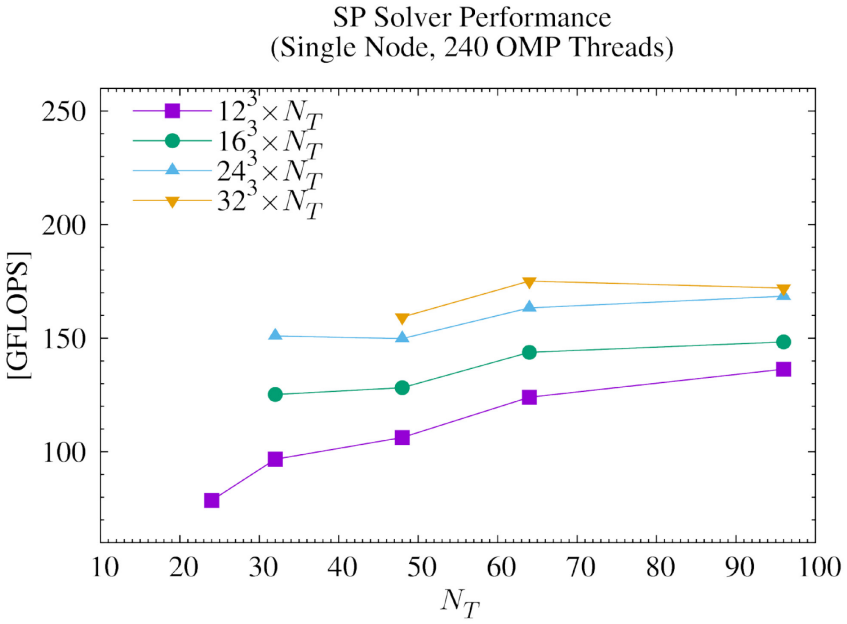


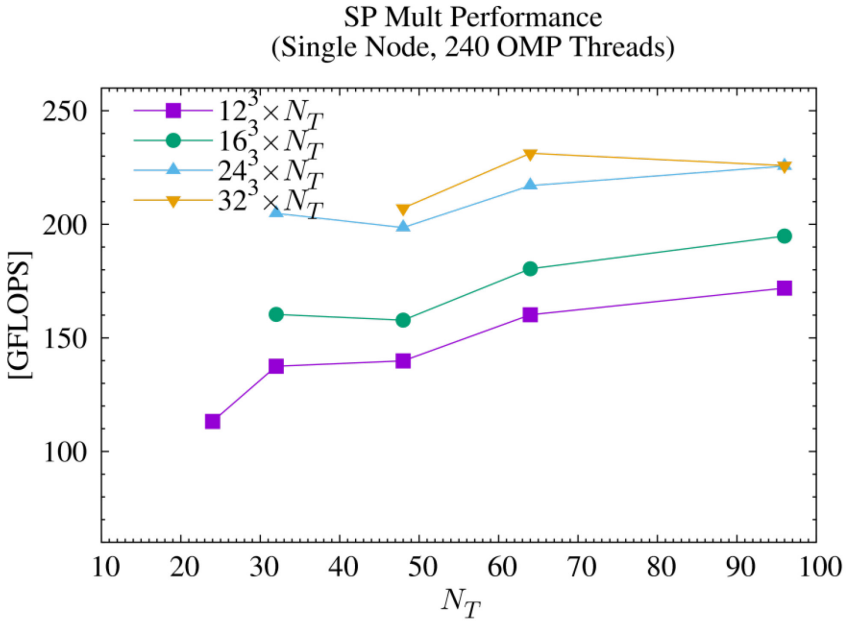
3. Results

- **We measure the performance (Timing, GFLOPS) of**
  - **SP Solver : BiCGStab preconditioned with**
    - **even/odd-site + DD-RAS(NRAS=8,NMR=2)**
  - **Mult : hopping mult**

$$\mathrm{Mult}(n,m) = F(n)\sum_{\mu=1}^{4}\left[(1-\gamma_\mu)U_\mu(n)\delta_{n+\hat{\mu},m} + (1+\gamma_\mu)U_\mu^H(n-\hat{\mu})\delta_{n-\hat{\mu},m}\right]$$

- **Strong Scaling Benchmark (Single Process)**

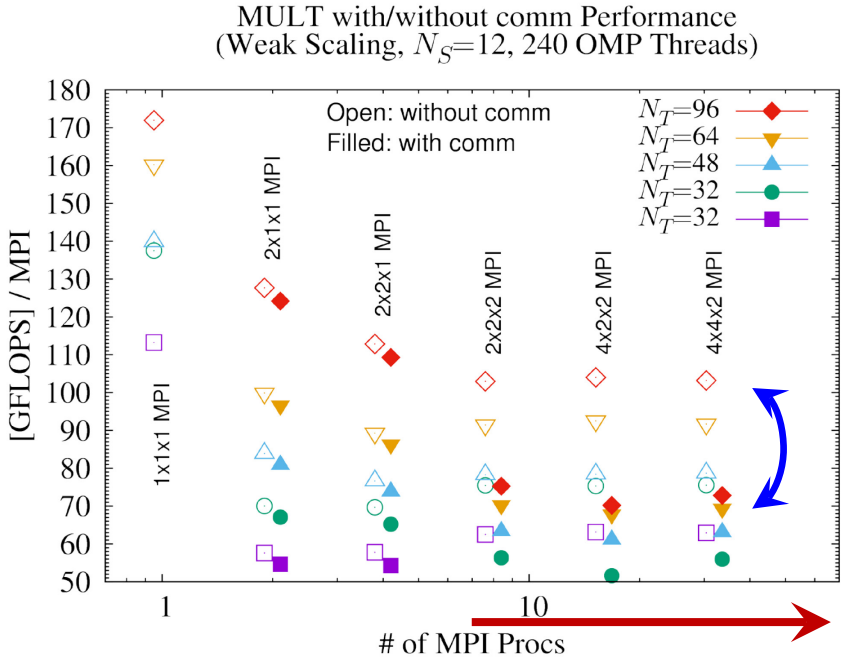


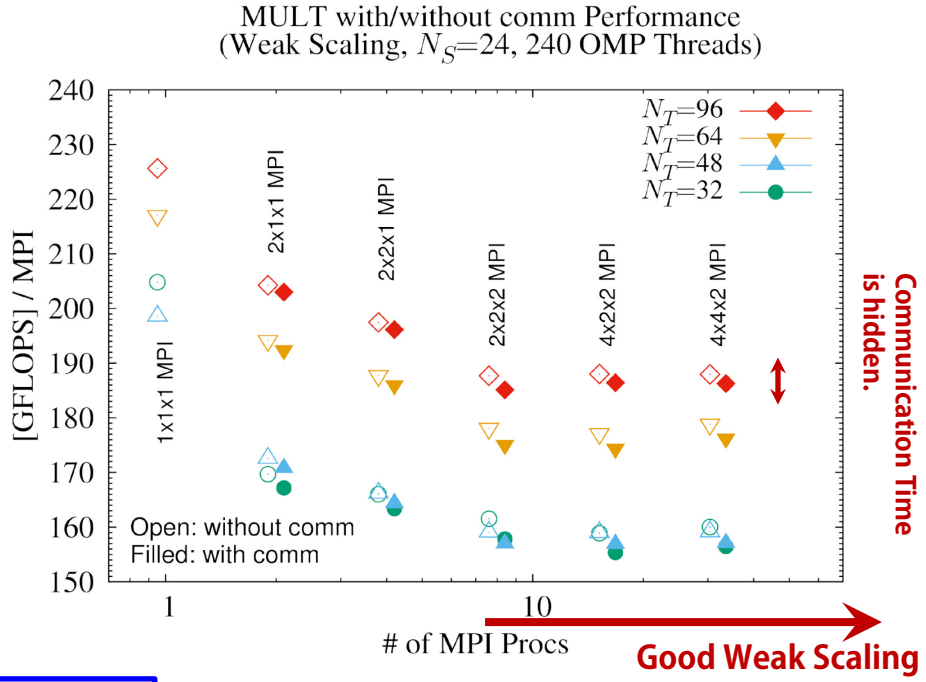


- **Mult achieves ≈ 200GF or more for lattices larger than 40000 sites.**
- **SP Solver performance is ~70%~80% of that of Mult.**

3. Results

- **Weak Scaling Benchmark**
  - **Parallelized in X,Y,Z directions (3D partitionig)**
  - **Fixed Local Volume** ($(N_X/N_{PX}) \times (N_Y/N_{PY}) \times (N_Z/P_Z) = N_S^3$)
  - **Varying the MPI size** ($N_{PX} \times N_{PY} \times N_{PZ}$) **: (1x1x1, 2x1x1,2x2x1,2x2x2,4x2x2,4x4x2)**
  - **Verifying the effect of the Communication and Computation Hiding.**
    - **MULT + communication timing VS MULT computing timing**





- **N_S=12 : Communication time is still visible.**

- **N_S=24 : Communication time is completely hidden by the computation. (N_S>24)**
- **Good Weak Scaling is observed for the cases with 3D MPI partitioning.**

4. Summary

- **We have optimized and tuned the QCD Solver Benchmark program for COMA (KNC) system @ CCS U. of Tsukuba.**
  - **by adding**
  - **the Single Precision BiCGStab Solver in a single offload region**
  - **with**
  - **vectorization with SIMD intrinsics, loop tiling, prefetching, reverse offloading technique, and communication and computation overlapping.**
- **In this talk, we focused on**
  - **Strong Scaling performance of Single Process.**
    - **>~ 200GF for MULT with sizes $> 24^3 \times 32, 16^3 \times 96$ lattices.**
  - **Weak Scaling performance at some fixed local spatial volumes.**
    - **Good Weak scaling is seen for 3D-X,Y,Z MPI partitioning.**
    - **With local Lattice sizes larger than $24^3 \times 32$, the communication in MULT is completely hidden by the computation of MULT.**

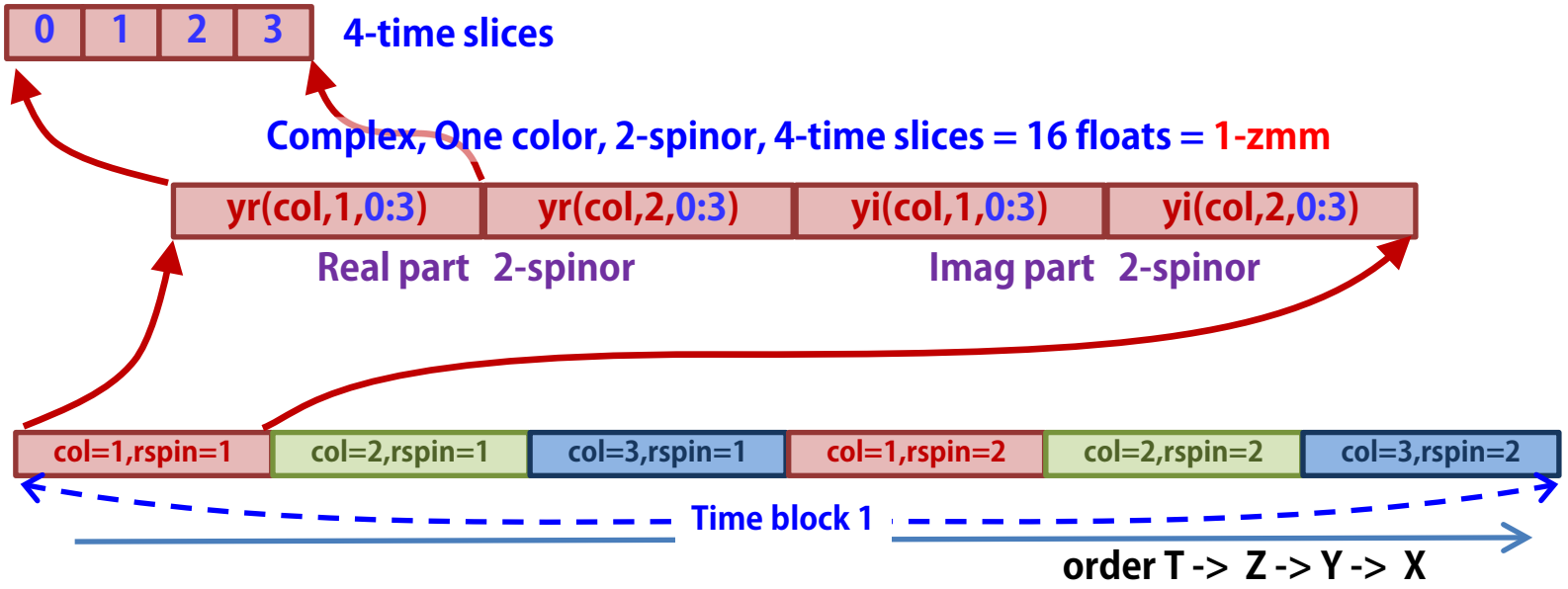
CCS QCD Solver Benchmark tuned for COMA (KNC) system is also available at

http://www.ccs.tsukuba.ac.jp/eng/research-activities/published-codes/qcd/

Backups

Backups 2. Tuning CCS-QCD for COMA system

- **Data Layout**
  - SIMD 16 (SP, __m512 zmm) vectorization
    - Quark fields
      - T, Z, Y, X : T-loop is inner most.
      - 4-time slices of a two-spinor (complex) of one color fit the SIMD16.
      - Spin projections are done with Shuffle/Permute ops.
      - Link multiplications are done among the SIMD16 (__m512 zmm) data.

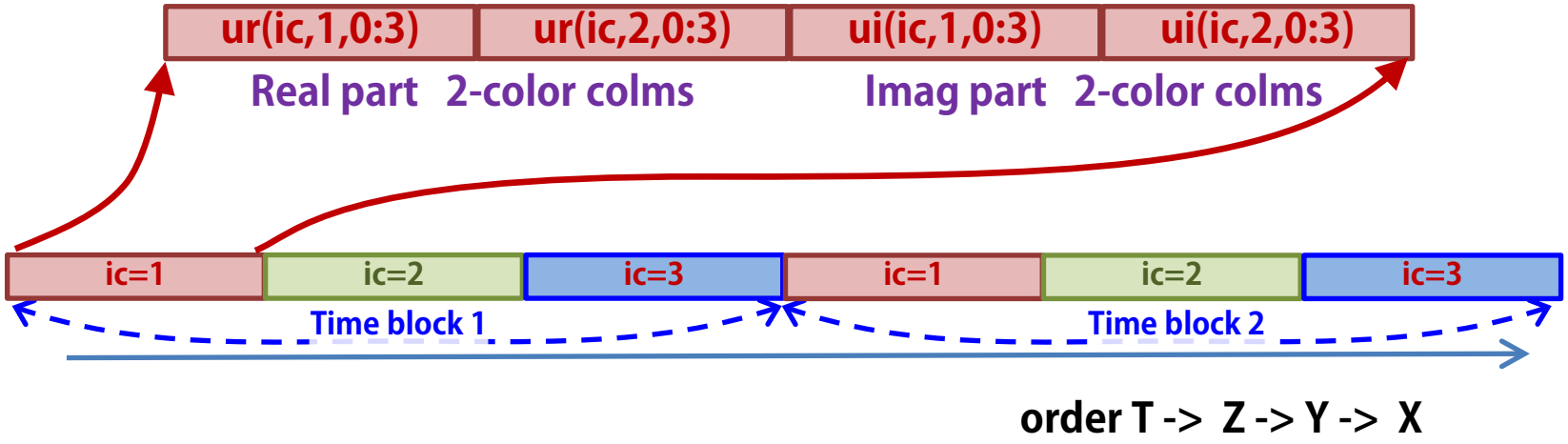


Backups 2. Tuning CCS-QCD for COMA system

- **Data Layout**
  - SIMD 16 (SP, __m512 zmm) vectorization
    - Gluon (Link) fields
      - T, Z, Y, X : T-loop is inner most.
      - 4-time slices of a two color-column (complex) of one color-row fit the SIMD16.
      - SU(3) reconstruction is used.
      - Link multiplications are done among the SIMD16 (__m512 zmm) data.

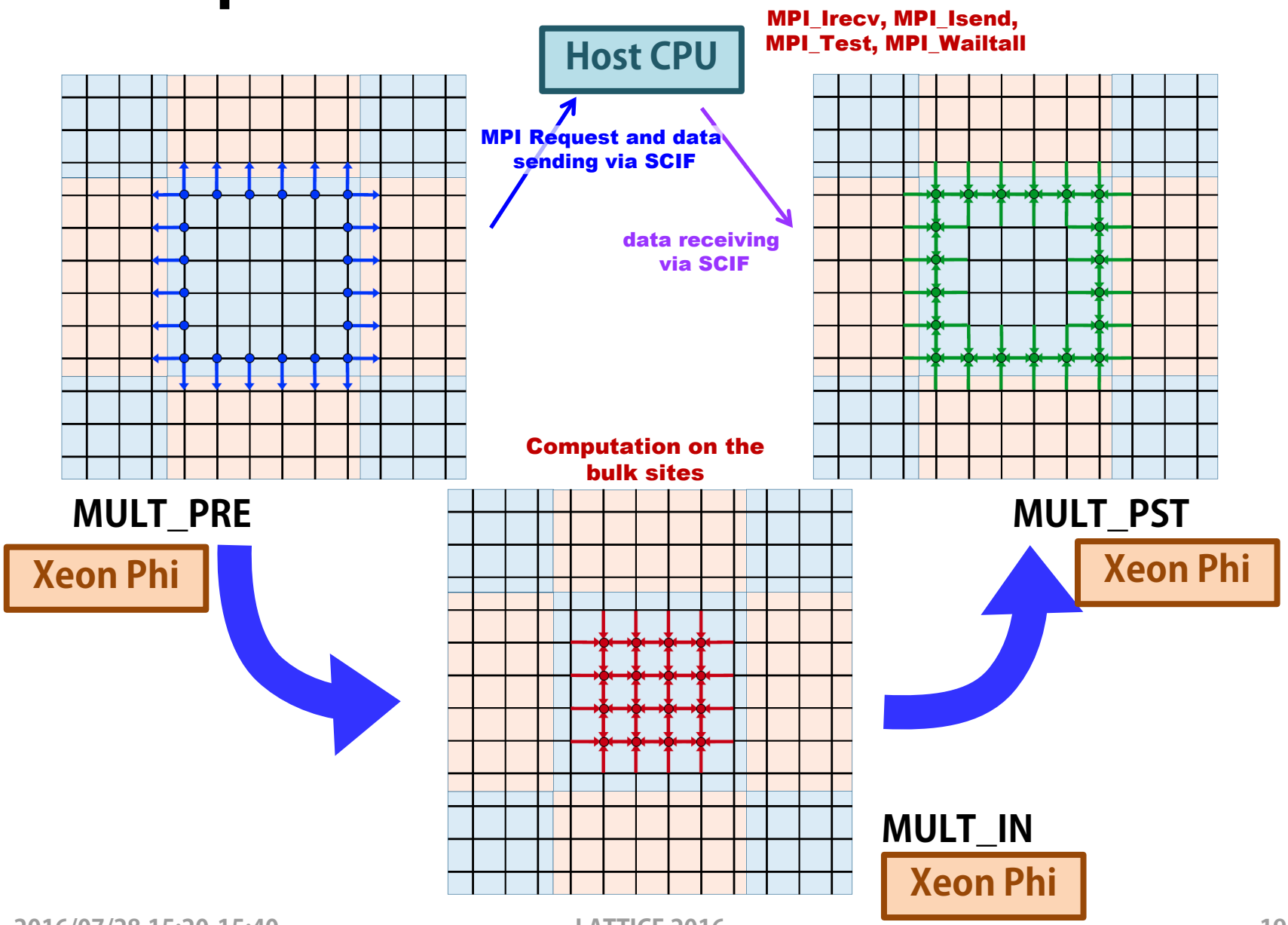
$$\begin{pmatrix} u11 & u12 & u13 \\ u21 & u22 & u23 \\ u31 & u32 & u33 \end{pmatrix} = (u1, u2, u3), u3 = (u1 \times u2)^* \Rightarrow \begin{pmatrix} u11 & u12 \\ u21 & u22 \\ u31 & u32 \end{pmatrix}$$

Complex, 1 row-2 columns, 4-time slices = 16 reals = 1-zmm



Backups

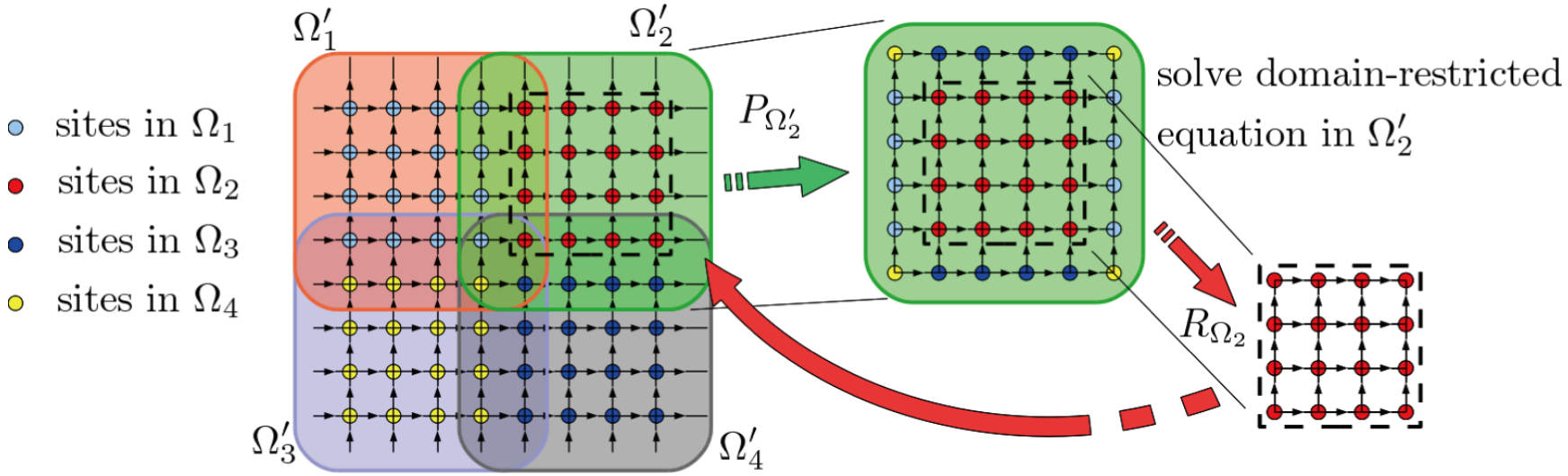
- Communication and Computation overlapping



Backups

– Overlapped Domain-decomposed preconditioner (RAS)

X.-C. Cai and M. Sarkis, *SIAM J. Sci. Comput.* 21, 792 (1999).
Yusuke Osaki and Ken-Ichi Ishikawa, PoS(Lattice 2010)036



- **Extends the local domain volume and solve the problem independently on each node.**
- **Improves the convergence rate of the iterative solver.**
- **Larger volume serves a high efficiency for MIC many core architecture.**
- **MULT_RAS : Mult on extended domain.**