IXPUG 2019 Annual Conference at CERN

Applying Vectorization to Lattice QCD Calculations

Shun Xu, Zhong Jin

Computer Network Information Center
Chinese Academy of Sciences

September 25,2019

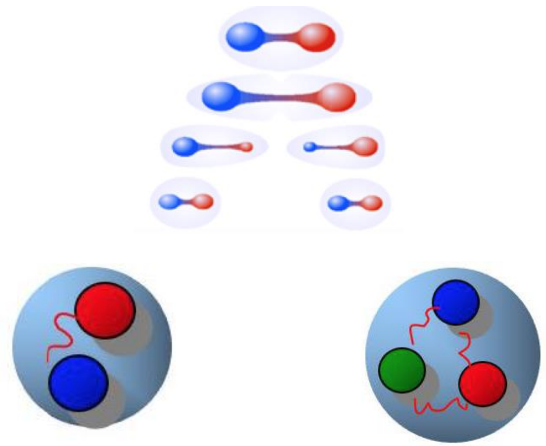
Outline

- **Introduction to lattice QCD**
  – Background in lattice QCD
  – The computational method and its challenge
- **Vectorization optimization of lattice QCD**
  - The principle of algorithm
  - On Intel & Sunway processors
- **Summary**

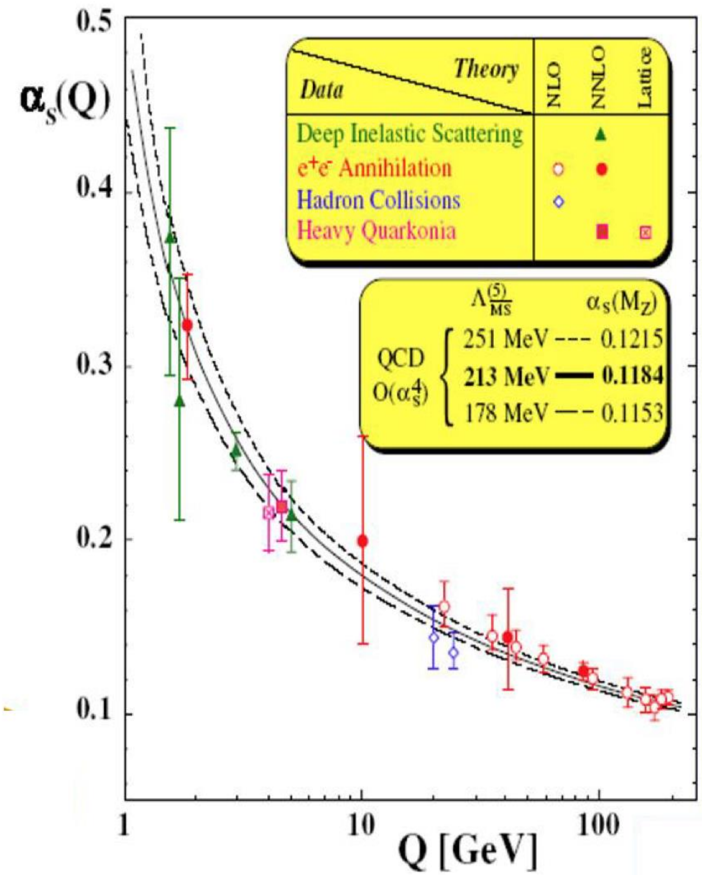
Introduction to lattice QCD

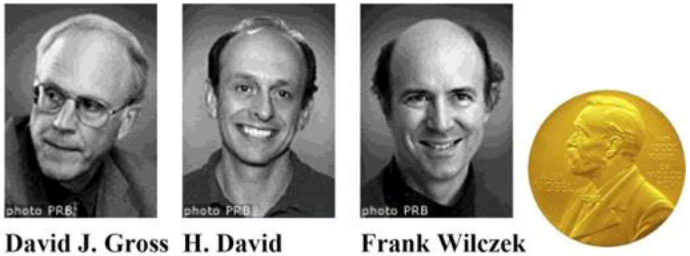
Quantum ChromoDynamics (QCD)： a fundamental theory of strong interaction

Confinement



meson hadron



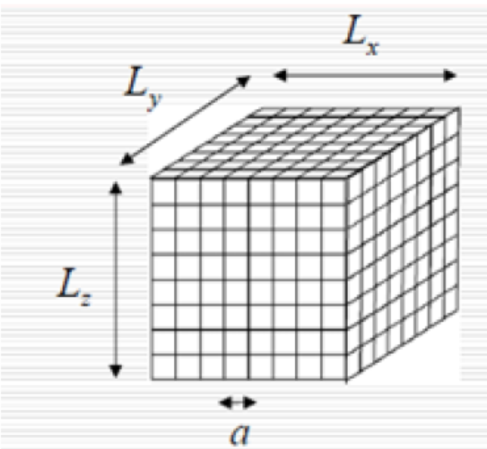


David J. Gross H. David Politzer Frank Wilczek

For the discovery of **asymptotic freedom** in the theory of the **strong interaction**.

Introduction to lattice QCD

Unfortunately, the quark **confinement** problem can not be solved directly from the perturbative method of QCD, but the discrete and numerical solution on lattice is an effective measure, i.e. lattice QCD.



$$N = L^3 \times T$$
$$\{P_l, A_l\}, l = 8 \times 4 \times N;$$
$$\{\pi_i, \phi_i\}, i = 12N;$$
$$\{\pi^*{}_i, \phi^*{}_i\}, i = 12N$$
$$M[U]: 12N \times 12N(matrix)$$
$$Q[U] = M^+[U]M[U]$$

$$H = \frac{1}{2}\sum_l P_l^2 + \sum_i \pi_i^* \pi_i + S_G[U] + S_{PF}[U, \phi, \phi^*]$$
$$S_{PF} = \sum_{i,j} \phi_i^* Q^{-1}{}_{ij} \phi_j = \phi^+ Q^{-1} \phi$$
$$\overline{O} = \frac{1}{Z}\int [DUDP] \int D\phi D\phi^* D\pi D\pi^* e^{-H} O[U]$$
$$Z = \int DUD\phi D\phi^* DPD\pi D\pi^* e^{-H}$$

Space-time discretization

Huge degree of freedom

Path integral quantification: from quantum field theory to statistical ensemble

Introduction to lattice QCD

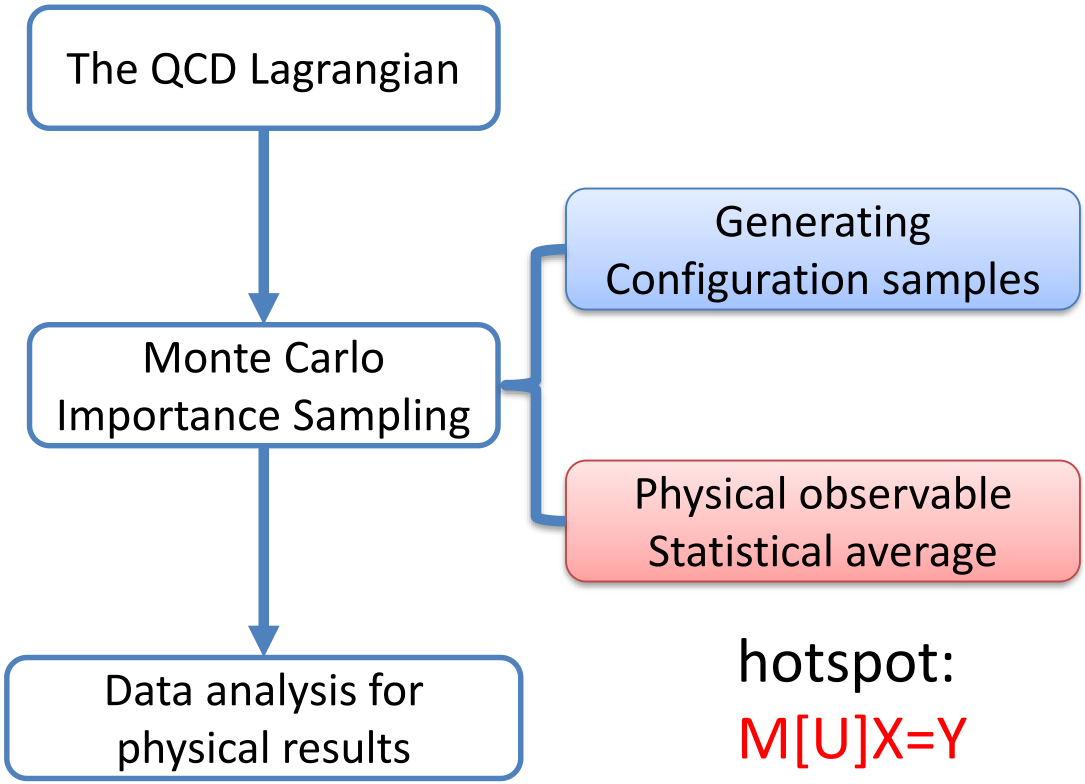
Lattice QCD: A first-principle method to study the QCD non-perturbative properties

Strong interaction property in low energy ⇔ Quark confinement ⇔ Hadron property and structure

- Lattice QCD is of great theoretical significance
  - the study of strong interaction
  - the accurate inspection of the standard model
  - the search of new physical results
- Lattice QCD numerical simulation is expensive computationally
  - Won **the Gordon Bell Prize** in 1988, 1998 and 2006 and the finalist in 2018.

The Lattice QCD Method

The numerical computation method of lattice QCD



hotspot:

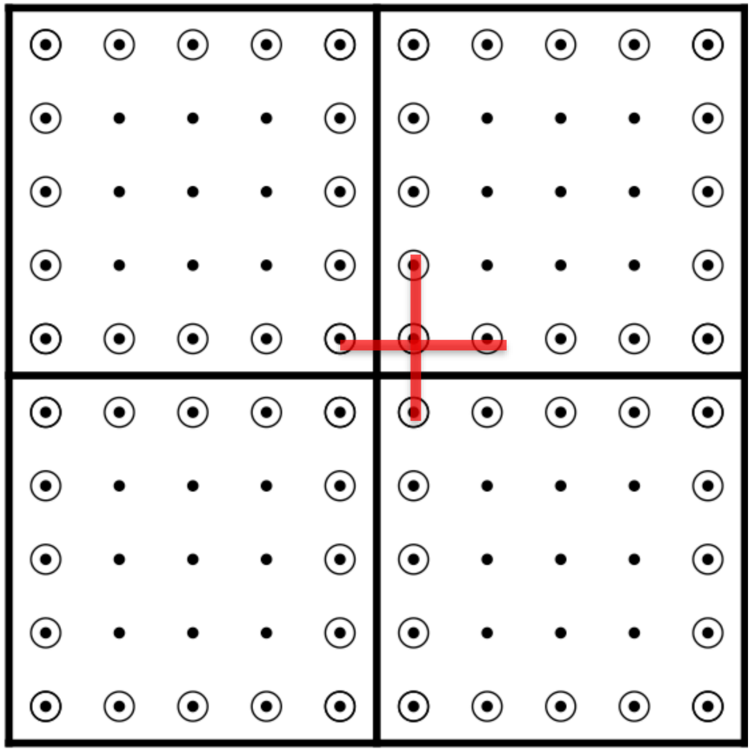
M[U]X=Y

U: 3x3- dimensional matrix

M[U]: L^3xTx12-dimensional sparse matrix

The Computational Challenges

$$
\begin{aligned}
M_{x,x'} &= -\frac{1}{2}\sum_{\mu=1}^{4}\left(P^{-\mu}\otimes U_x^{\mu}\,\delta_{x+\hat{\mu},x'} + P^{+\mu}\otimes U_{x-\hat{\mu}}^{\mu\dagger}\,\delta_{x-\hat{\mu},x'}\right)\\
&\quad + (4+m+A_x)\delta_{x,x'}\\
&\equiv -\frac{1}{2}D_{x,x'} + (4+m+A_x)\delta_{x,x'}.
\end{aligned}
$$

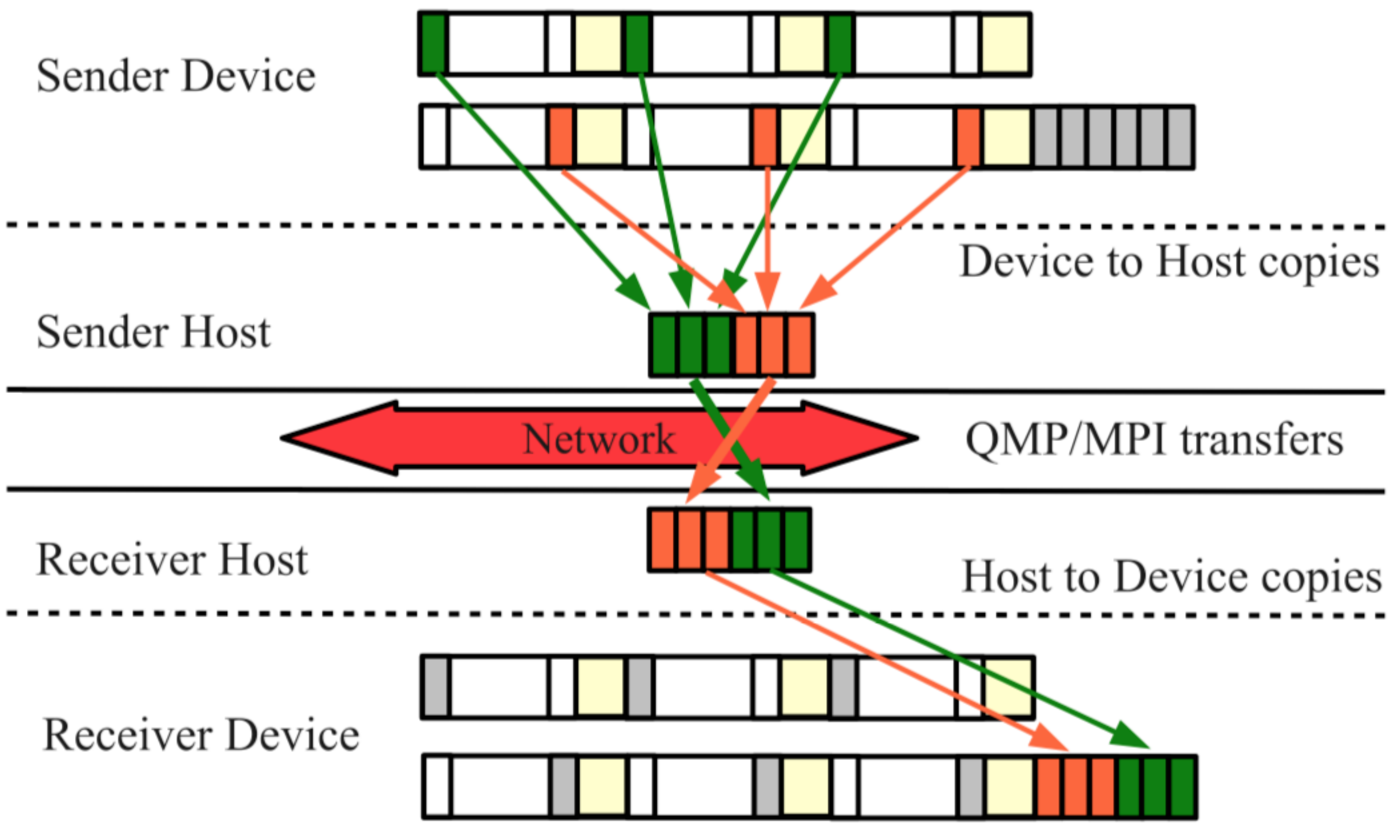


The computational challenges

1. a statistical problem for a multi-degree-of-freedom system in neighbor interaction.
2. a computation-intensive task
   - high parallelism
   - high scalability

Methods to improve performance

1. **Task:** Overlap between calculation and communication
2. **Instruction:** Vectorization optimization of high-dimensional data



Ref.: R. Babich, M. Clark, B. Joo, SC10, arXiv:1011.0024

Vectorization optimization

中国科学院 计算机网络信息中心 Computer Network Information Center, Chinese Academy of Sciences

To cut the space-time of the periodic boundary condition into a four-dimensional lattice array

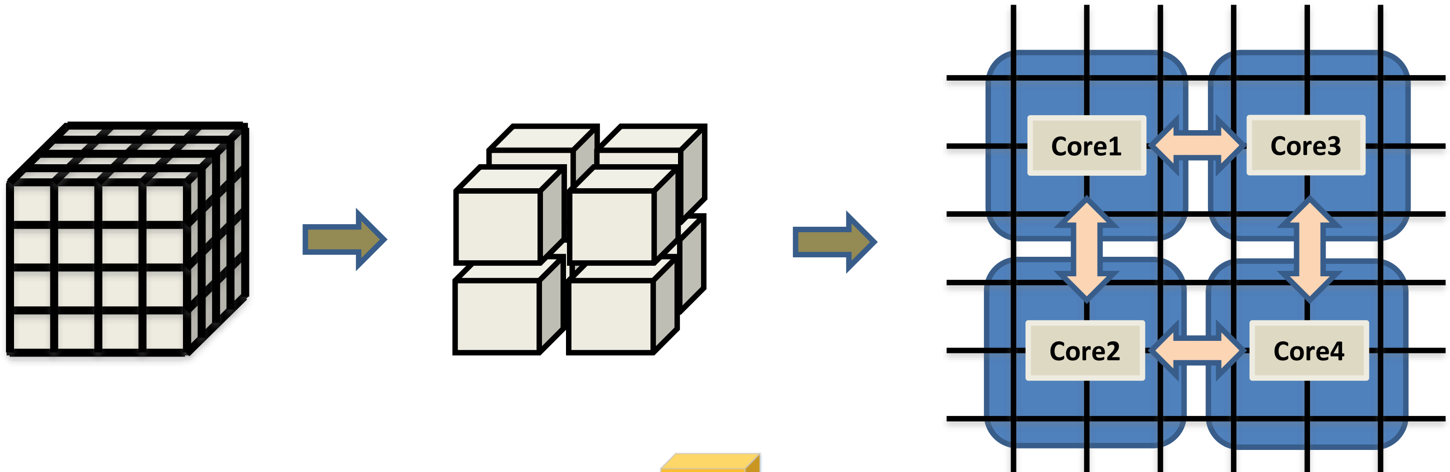
The fermion field quantity representing quark is placed at each lattice site, which is the Glasman number with three chromatic components and four spinor components (12 complex vectors).

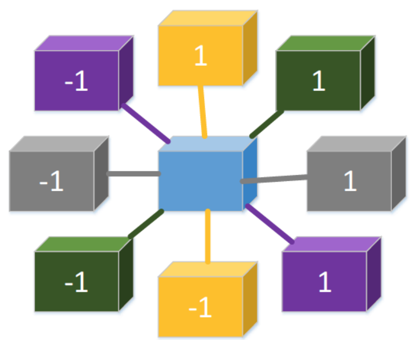
$$\phi(x, y, z, t) = \begin{pmatrix} \begin{pmatrix} d_{11} \\ d_{12} \\ d_{13} \end{pmatrix} \\ \begin{pmatrix} d_{21} \\ d_{22} \\ d_{23} \end{pmatrix} \\ \begin{pmatrix} d_{31} \\ d_{32} \\ d_{33} \end{pmatrix} \\ \begin{pmatrix} d_{41} \\ d_{42} \\ d_{43} \end{pmatrix} \end{pmatrix}$$

The gauge field of gluon is placed on the connection between adjacent sites, which can be written as a 3x3 complex matrix with unitary mode on color space. Each lattice has eight such matrices, each connected to eight neighbors.

$$U_\mu(x, y, z, t) = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Data parallelism in different levels





4D stencil computation in Lattice QCD

$$\begin{pmatrix} a+bi, & c+di, & e+fi \\ g+hi, & j+ki, & l+mi \\ n+pi, & q+ri, & s+ti \end{pmatrix} \begin{pmatrix} u+vi \\ w+xi \\ y+zi \end{pmatrix} = \begin{pmatrix} (au+cw+ey-bv-dx-fz)+(av+cx+ez+bu+dw+fy)i \\ (gu+jw+ly-hv-kx-mz)+(gv+jx+lz+hu+kw+my)i \\ (nu+qw+sy-pv-rx-tz)+(nv+qx+sz+pu+rw+ty)i \end{pmatrix}$$

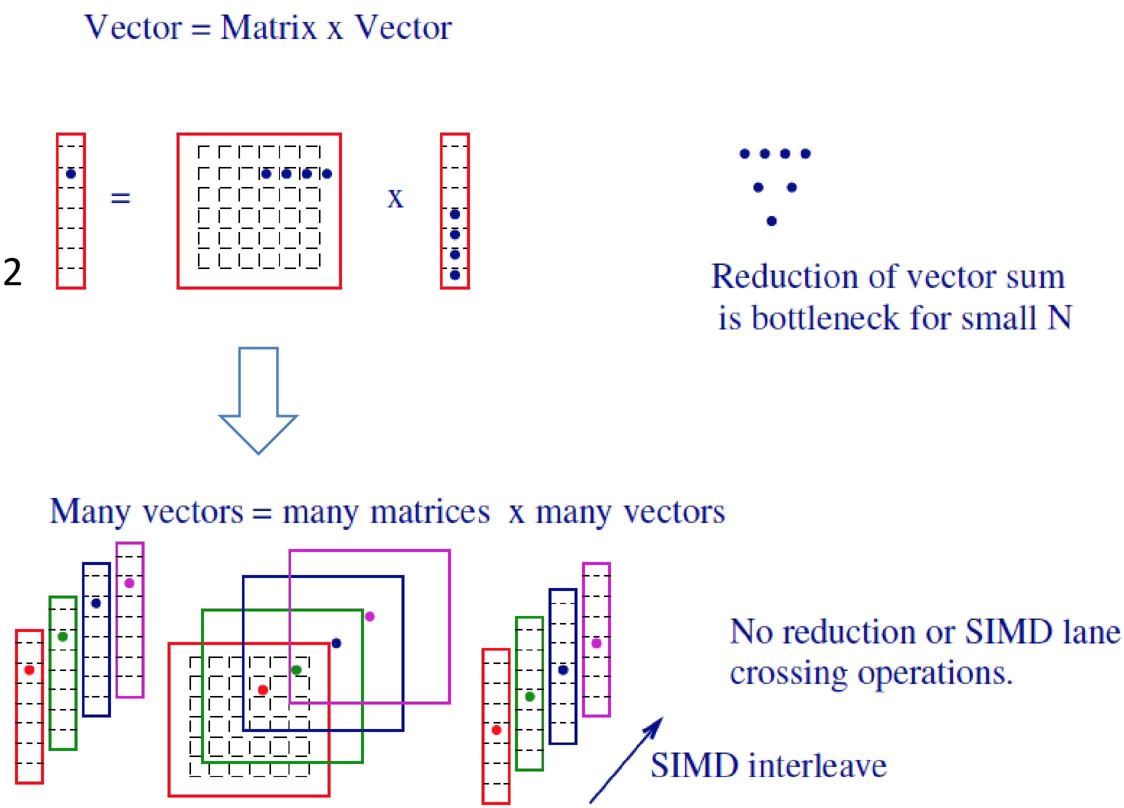
Vectorization on Intel Processors

- Programming of vectorization
  - SSE, AVX, AVX2 and AVX512
  - ASM and Intrinsic function
  - OpenMP simd directive
- Packages available
  - QDP++
    - Scalarsite sse library (for blas and linalg)
  - MILC
    - Single precision SSE routines for MILC
  - **Grid**
    - C++ 11 template classes for SIMD vectors

SIMD in Grid

A **SIMD** vector parallelism **framework** offered by Grid package

1. The abstraction of vector operations using modern C++ 11
2. SSE, AVX, AVX2, FMA4, IMCI and AVX512

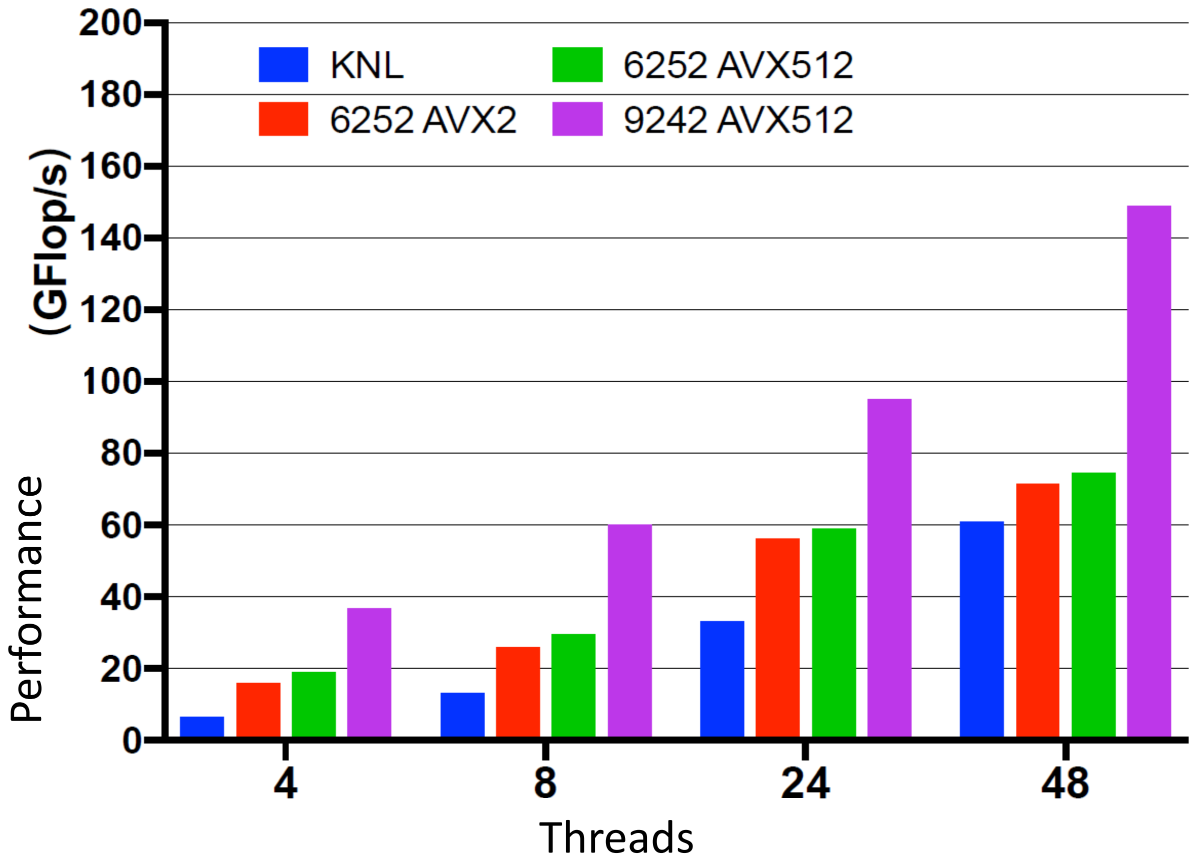



```
#if defined (SSE2)
 typedef __m128 zvec;
#endif
#if defined (AVX1) || defined (AVX2)
 typedef __m256 zvec;
#endif
#if defined (AVX512)
 typedef __m512 zvec;
#endif
 class vComplexD  {
   zvec v;
   // Define arithmetic operators
   friend inline vComplexD operator + (vComplexD a, vComplexD b);
   friend inline vComplexD operator - (vComplexD a, vComplexD b);
   friend inline vComplexD operator * (vComplexD a, vComplexD b);
   friend inline vComplexD operator / (vComplexD a, vComplexD b);
   static int Nsimd(void);
 }
```

Ref.: *Grid: A next generation data parallel C++ QCD library.* Boyle, Peter & Cossu, Guido & Yamaguchi, Azusa & Portelli, Antonin. (2016).

Performance results of Grid

计算机网络信息中心 Computer Network Information Center, Chinese Academy of Sciences



KNL: Intel Xeon Phi 7210

6252: Intel Xeon Gold 6252

9242: Intel Xeon Platinum 9242

Grid tests of vectorization on different processors

Vectorization on Sunway Processor

SU(3) Vectorization multiplication

BEHIN

for i in (a_1,a_2,a_3)

  tmp1 = simd_load(i,i,i,i)

  for j in (α, γ, μ)

    tmp2 = simd_load(j_1, j_2, j_3, j_4)

    result_re += simd_mul(tmp1,tmp2)

  for j in (β, δ, ν)

    tmp2 = simd_load(j_1, j_2, j_3, j_4)

    result_im += simd_mul(tmp1,tmp2)

for i in (b_1,b_2,b_3)

  for j in (α, γ, μ)

    tmp2 = simd_load(j_1, j_2, j_3, j_4)

    result_im += simd_mul(tmp1,tmp2)

  for j in (β, δ, ν)

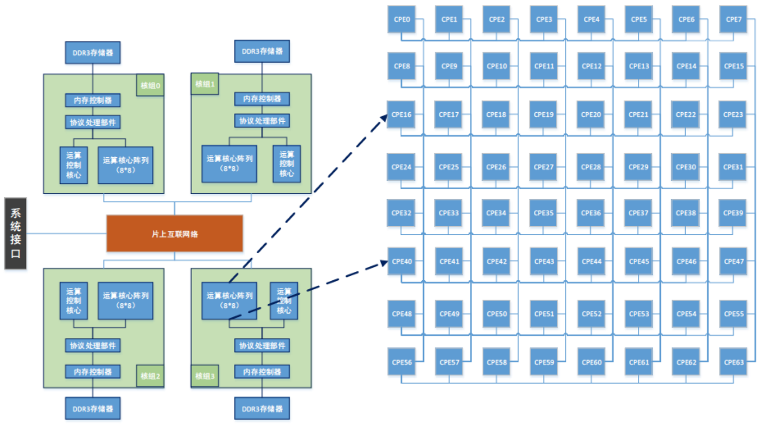
    tmp2 = simd_load(j_1, j_2, j_3, j_4)

    result_re += simd_mul(tmp1,tmp2)

END

Algorithm of vectorization

1. Partition data by the Z and T axis.
2. 256-bit SIMD vectorization
3. Register communication

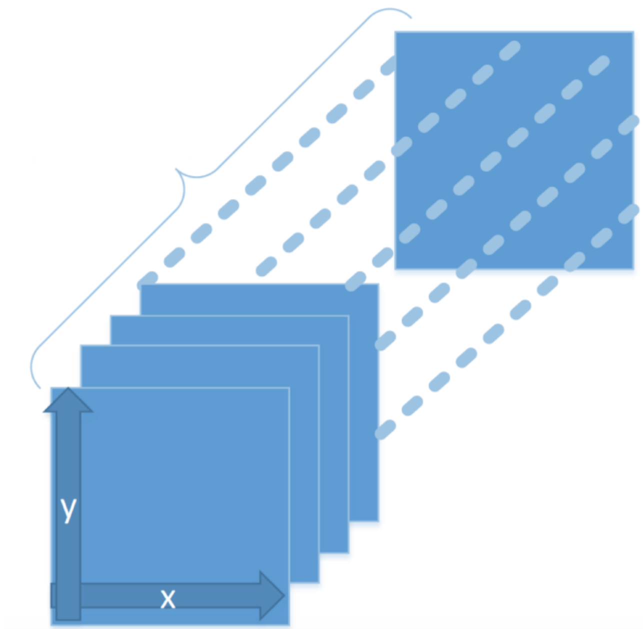


The architecture of Sunway many-core processor

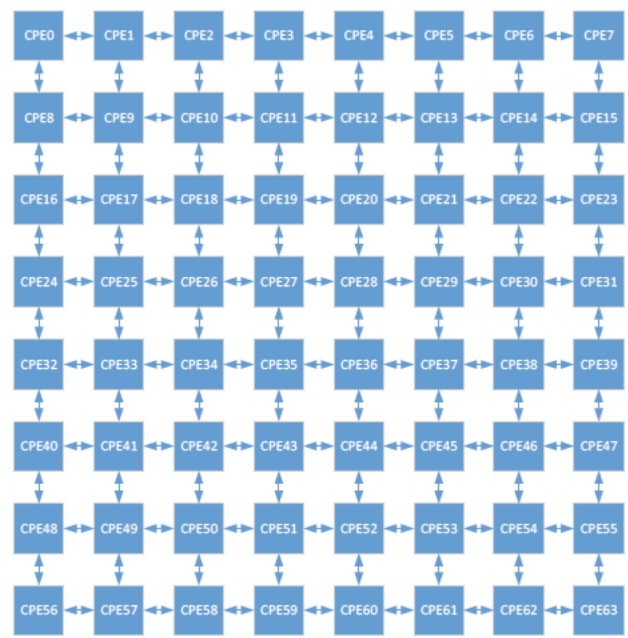
Sunway TaihuLight Supercomputer

Data organization for vectorization

To fit 64KB size of LDM(Local Data Memory) in computing processing element (CPE)



Data segmentation for CPE vectorization

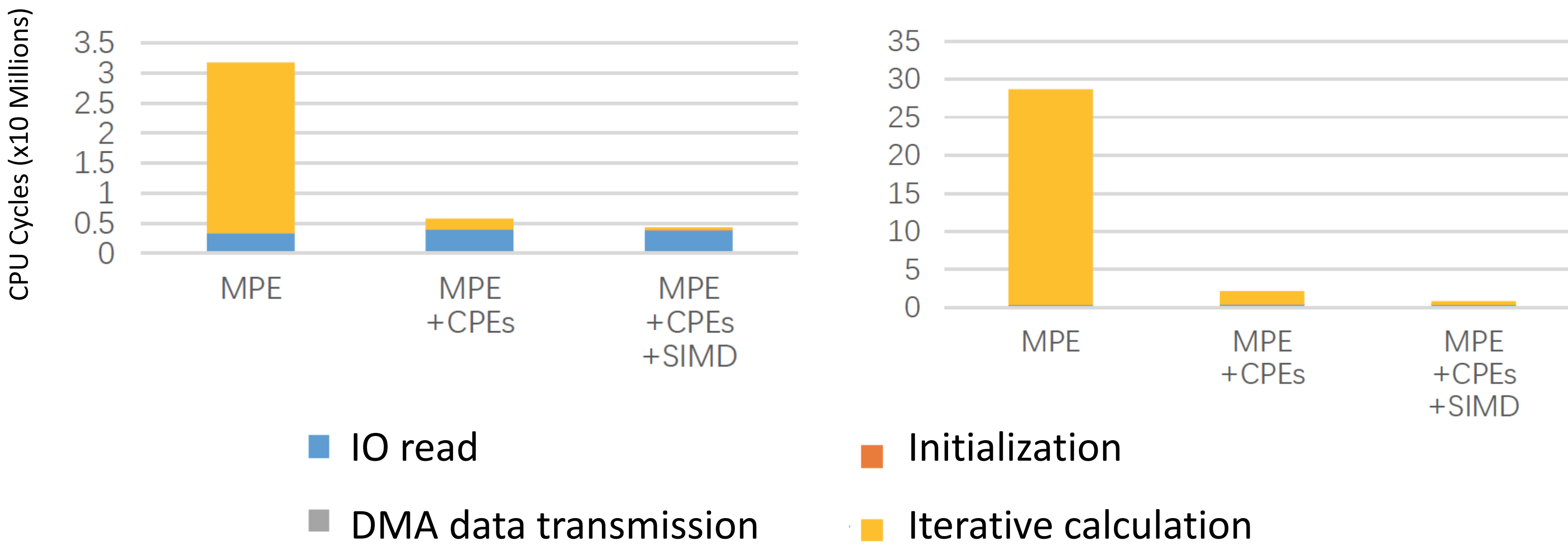


Register communication between CPEs

Performance results

The running time ratio of each part in 10 times iterative calculation

The running time ratio of each part in 100 times iterative calculation



Ref.: *Accelerating Lattice QCD on Sunway Many-Core Processor*
Zengxiao Zhang, Zhongzhi Luan, Chongyang Xu, Ming Gong and **Xu Shun**
DOI: 10.1109/BDCloud.2018.00094

A Unified Data Parallel Programming Model

- We are suffering from
  - The poor portability of vectorization code
  - The complexity hardware details overexposed to user
- We are looking forward to
  - A unified programming model with data parallelism
  - Library available on diverse architectures (e.g. SSE, AVX, ASIMD, and NEON).
  - A uniform API for domainal applications

Summary

- Lattice QCD is a useful non-perturbative theoretical calculation method, by defining field variables at discrete time-space points.
- High-performance computational methods are required to optimize the data computation and communication in large-scale lattice QCD.
- **My view:** *A Unified data parallel programming model* (e.g. One API by Intel) is expected by lattice QCD application.

Acknowledgments

- The helpful discussions with Prof. Ying Chen, Prof. Zhaofeng Liu, Dr. Ming Gong and Dr. Yujiang Bi from IHEP CAS, and Dr. Yibo Yang from ITP CAS.
- The support by IPCC-CAS program.
- The support by the National Key Research and Development Program of China (2017YFB0203200).

Thank you!

xushun@sccas.cn