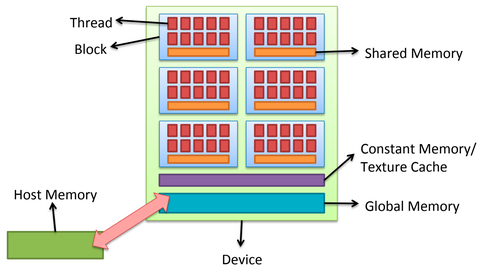
GPU and CUDA Architecture

Since 2009, researchers at National Taiwan University have successfully set up a GPU cluster which currently constitutes of 350 GPUs. This is the first GPU supercomputer in Taiwan. We have developed highly efficient CUDA codes for the most computationally challenging problems in high energy physics, condensed matter physics, and astrophysics. In 2014, our GPU cluster attains 150 Teraflops (sustained) for lattice QCD. During 2009-2014, we have developed efficient algorithms and CUDA codes for the ground-breaking simulation of lattice QCD with exact chiral symmetry. Now we are one of the three lattice QCD groups (RBC-UKQCD, JLQCD, TWQCD) around the world who can perform such a demanding large-scale lattice QCD simulation incorporating dynamical quarks with exact chiral symmetry. Remarkably, we have succeeded in performing our simulations using a GPU cluster, rather than expensive supercomputers (e.g., IBM BlueGene/Q).

One of the most crucial part in the simulation program is the multi-GPU Conjugate Gradient (CG) solver with OpenMP. In the followings, the implementation and optimization of the two main kernels in matrix-vector multiplication in our CG calculation are discussed.



CG Kernels (D_w Multiplication)

$$(D_w)_{x,x'} = \frac{-1}{2}\sum_{\hat{\mu}}\left[(1-\gamma_\mu)U_\mu(x)\delta_{x+\hat{\mu},x'} + (1+\gamma_\mu)U_{-\mu}(x)\delta_{x-\hat{\mu},x'}\right]$$

- Hopping terms
  - Texture is used for caching data
  - Internal loop is used to reuse the read-in data
  - Peer-to-Peer access is enabled to load the hopping term from other devices
- Link variables multiplication
  - For a given $\hat{\mu}$, U is the same in fifth dimension, hence the shared memory is used
- Gamma matrix multiplication
  - Only the left-handed Dirac indices are calculated.

From threads and blocks indices, calculate x, y, z, s
Iteration for t — To reuse the data; significant performance enhancement; divide loops to each device
Iteration for 8 directions — Loop expand
Load vector (texture to register)
Load link variable (texture to shared memory) — Memory bandwidth bound
Gamma matrix multiplication (one left-handed components)
Link variable multiplication — All loops here are expanded
Restore right-handed components
Write output to global memory

CG Kernels (M_5 Multiplication)

$$M_5 = \left[(4-m_0) + \omega^{\frac{-1}{2}}\left[c(1+L)(1-L)^{-1} + d\omega^{-1}\right]^{-1}\omega^{\frac{-1}{2}}\right]^{-1}$$

- **Block diagonal in chiral basis**
- **Does not depend on x, y, z, t or color-Dirac indices**
- **It is the constant matrix multiplication in the 5th space**
- **Use share memory to store source vectors**
- **Internal loop to reuse the read-in M_5 matrix**

From threads and blocks indices, calculate x, y, z, s
Iteration for t and part of z — To reuse the data; divide t loops to each device
Load M_5 matrix (texture to register)
Iteration for 24 color/Dirac/complex indices — Loops expanded
Load vector (texture to shared memory)
M_5 matrix multiplication — Loops expanded
Write output to global memory

Benchmarks

CG (mixed prec.) attains 410 GFLOPS on GTX-TITAN

	Dw(Single)	M5(Single)	Dw(Double)	M5(Double)	CG(Mixed)
GTX285	177	346	33	69	181
C1060	128	290	29	61	132
C2070	171	244	22	96	156
GTX480	293	309	37	116	252
GTX580	338	445	41	150	317
GTX TITAN	440	578	53	195	410
GTX TITAN Z	454	438	123	132	410

All numbers are in unit of GFLOPS, tested with DWF on 16^3 x 32 x 16 lattice

	1 GPU card	2 GPU cards	Speedup
GTX680	248	453	1.83
GTX690	475	942	1.98
K20c	286	535	1.87
GTX TITAN	410	781	1.90
GTX TITAN Z	410	780	1.90

All numbers are in unit of GFLOPS, tested with LQCD on 24^3 x 48 x 16 lattice

	2 GPU	4 GPU	Speedup
GTX TITAN/Z	780	1350	1.73

All numbers are in unit of GFLOPS, tested with DWF on 32^3 x 64 x 16 lattice

TWQCD COLLABORATION

Large-Scale Simulation of Lattice QCD with GTX-TITAN

Ting-Wai Chiu[1,2], Yu-Chih Chen[1], Han-Yi Chou[1]
[1] Physics Department, National Taiwan University, Taipei 10617, Taiwan
[2] Center for Quantum Science and Engineering, National Taiwan University, Taipei 10617, Taiwan

Abstract

We present the state-of-the-art simulation of lattice QCD with dynamical (u,d,s,c) quarks at National Taiwan University. Using a unit of two GTX-TITAN, lattice QCD with (1+1+1+1)-flavors of domain-wall quarks can be simulated on the $32^3 \times 64$ lattice, attaining sustained 780 Gflops/s. This study is vital for understanding QCD (Quantum Chromodynamics), the fundamental theory for the interaction between quarks and gluons, which manifests as the strong interaction inside the nucleus and plays an important role in the evolution of the universe.

Lattice QCD and Domain-Wall Fermions

Quantum Chromodynamics (QCD) is the fundamental theory for the interaction between quarks and gluons. It manifests as the short-range strong interaction in the nucleus, and plays an important role in the evolution of the early universe, from the quark-gluon "plasma" phase to the hadron phase. To solve QCD is a grand challenge, since it requires the largest scale numerical simulation of the discretized action of QCD on the 4-dimensional space-time lattice[1].

For the QCD action $S = S_G(U) + \bar{\psi}D(U)\psi$, any physical observables $\mathcal{O}(\bar{\psi},\psi,U)$ can be obtained from

$$\langle \mathcal{O}(\bar{\psi},\psi,U)\rangle = \frac{\int dU d\bar{\psi} d\psi \mathcal{O}(\bar{\psi},\psi,U) e^{-S}}{\int dU d\bar{\psi} d\psi e^{-S}}$$


Kenneth G. Wilson
Nobel Prize (1982)

Then we can put this integral on the lattice and use Hybrid Monte Carlo (HMC) method to compute this integral. The most time-consuming part in HMC is to solve a linear system by the conjugate gradient algorithm(CG). By using GPU, we can boost our simulation dramatically.

Moreover, since quarks are relativistic fermions, the 5-th dimension is introduced such that massless quarks with exact chiral symmetry can be realized at finite lattice spacing, on the boundaries of the fifth dimension, the so-called domain-wall fermion (DWF)[2]. The effective action of DWF can be written as $D_m(U) = D_w(U) + M_5(m)$

where the definition of D_w and M_5 are given above.

1. K. G. Wilson, Phys. Rev. D 10, 2445 (1974).
2. D. B. Kaplan, Phys. Lett. B 288, 342 (1992)

Salient Features of the Quark Matrix $D_m(U)$

- D_m is a sparse matrix, only involving the nearest neighbor interactions.
- Iterative algorithms (conjugate gradient, Lanczos, etc.) are used, which involve the matrix-vector multiplication.
- CUDA kernels can be optimized for the matrix-vector ops. in QCD.

Exact One Flavor Algorithm(EOFA)

Recently, we have devised a novel pseudofermion action for hybrid Monte Carlo simulation of one-flavor domain-wall fermion (DWF) in lattice QCD. This pseudofermion action is exact, without taking square-root, unlike the widely-used rational hybrid Monte-Carlo algorithm (RHMC) which is inexact, requires an additional memory space which is prohibitively expensive for GPUs.

The main idea in HMC is to find a pseudo-fermion action S_{pf} such that

The integral we want $\Rightarrow$ $\int d\bar{\psi}d\psi e^{-\bar{\psi}D(U)\psi} \propto \det[D_T(U)] \propto \int d\phi^\dagger d\phi e^{-S_{pf}}$ $\Leftarrow$ The integral we calculate

where $S_{pf} \equiv \phi^\dagger \bar{D}(U)\phi$. In principle, there are infinite possibilities of S_{pf} can be used to satisfy the above equation. The crucial point here is finding a suitable $\bar{D}(U)$ which can be calculated efficiently.

For the wildly used RHMC, the S_{pf} can be written as

$$S_{pf} = \phi^\dagger \sqrt[4]{D_1^\dagger D_1}\left[\sqrt{D_m^\dagger D_m}\right]^{-1}\sqrt[4]{D_1^\dagger D_1}\phi$$

where the subscripts 1 and m are labeled for cutoff and quark mass respectively. With this action, one can use rational approximation and multi mass shift conjugate gradient (MMCG) to calculate the roots. But it needs copious memories to do the calculation efficiently, which means that it is very hard to implement RHMC on GPU.

For our exact one flavor algorithm (EOFA) [3], the S_{pf} is

$$S_{pf} = (0,{\phi_1}^\dagger)\left[I - k{\Omega_-}^T\frac{1}{H_1}\Omega_-\right]\begin{pmatrix}0\\ \phi_1\end{pmatrix} + ({\phi_2}^\dagger,0)\left[I + k{\Omega_+}^T\frac{1}{H_2}\Omega_+\right]\begin{pmatrix}\phi_2\\ 0\end{pmatrix}$$

where H_1 and H_2 are composed of D_m and M_5 matrix, k is a constant, and $\Omega_\pm$ are 5th dimensional constant matrices. This action possesses all good properties required for the HMC simulation and is more efficient than RHMC. Moreover, the memory usage of EOFA is much less than RHMC. Thus we can perform HMC simulation with this exact action on GPU.

3. Y. C. Chen and T. W. Chiu [TWQCD Collaboration], Phys. Lett. B 738, 55 (2014)

EOFA vs. RHMC

We compare the performance of the exact one-flavor algorithm (EOFA) with RHMC, and find that EOFA outperforms RHMC, no matter in terms of the efficiency or the memory consumption[3].

$$\frac{M_{RHMC}}{M_{EOFA}} = \frac{20 + 3(2+2N_p)N_s}{32 + 10.5N_s}$$

is the ratio of the memory requirement between RHMC and EOFA, where N_s is the number of lattice sites in the fifth dimension of DWF, and N_p is the number of poles used in the rational approximation for RHMC.

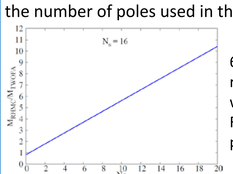


Fig. 2 For N_p = 12 and N_s = 16, the ratio is 6.58 for any 4D lattices. In other words, if EOFA requires 12 GB to perform HMC of lattice QCD with DWF on the $32^3 \times 64 \times 16$ lattice, then RHMC with 12 poles needs at least 79 GB to perform the simulation.

The memory-saving feature of EOFA is crucial for large-scale simulations of lattice QCD with GPUs, in view of each GPU having enormous floating-point computing power but limited device memory.

For example, using EOFA, two GPUs (each of 6 GB device memory, e.g., Nvidia GTX-TITAN) working together with OpenMP is capable to simulate lattice QCD with (u, d, s, c) DWF quarks on the $32^3 \times 64 \times 16$ lattice, attaining sustained 780 Gflops for two GTX-TITANs.

We did the tests of $N_f = 1$ and $N_f = (2+1)$ QCD on the $16^3 \times 32 \times 16$ lattice, for the conventional DWF . The details of the simulation of 2-flavors of DWF have been presented in Ref. [4]. After the initial thermalization of 300 trajectories (done with a GPU), we pick one configuration and use 4 cores CPU of i7-4820K CPU@3.70GHz to continue the HMC simulation with EOFA and RHMC respectively, and accumulate 5 trajectories.

4. T. W. Chiu [TWQCD Collaboration], J. Phys. Conf. Ser. 454, 012044 (2013)

EOFA vs. RHMC and 1+1 Flavor simulation

With the statistics of five trajectories (all accepted), the average time (seconds) for generating one HMC trajectory (after thermalization) is listed below

	EOFA	RHMC
$N_f = 1$	93241(290)	119445(408)
$N_f = 2+1$	143099(833)	172569(588)

To demonstrate the practicality of EOFA, we perform the first dynamical simulation of the (1+1)-flavors QCD with DWF, which also provides gauge ensembles for studying the isospin symmetry breaking effects in the hadron spectrum as well as other physical quantities.

We compute the valence quark propagator with the point source at the origin, and with parameters exactly the same as those of the sea-quarks. In Fig. 3, we plot the time-correlation function C(t) and the effective mass of the charged pion [3].

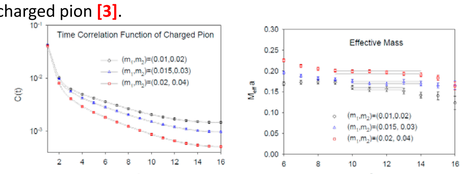


Fig. 3 (a) The time-correlation function of charged pion. (b) The effective mass of charged pion

Summary

To summarize, in the last 6 years (2009-2014), TWQCD Collaboration has devised novel algorithms and developed highly efficient CUDA codes for solving lattice QCD with domain-wall fermion. This not only asserts that GPU is the most cost-effective device for large-scale simulation of lattice QCD, but also provides ground-breaking results in the zero temperature and the finite temperature lattice QCD with exact chiral symmetry.