Code Optimization for Lattice QCD on Kepler GPU and Xeon Phi

Hwancheol Jeong

Seoul National University
Lattice Gauge Theory Research Center

Joint Winter Conference
High1, Jan. 28, 2015

Lattice QCD

- Lattice QCD (LQCD) : non-perturbative approach for QCD
- continuous space-time
  $\rightarrow$ discrete 4-dim. Euclidean space-time (lattice)
- infinite dimensional path integral $\rightarrow$ finite

$$\int D\psi D\overline{\psi} \int DA \;\rightarrow\; \prod_{n_\mu} \int d\psi(an_\mu) d\overline{\psi}(an_\mu) \int dU(an_\mu)$$

- when lattice spacing $a \rightarrow 0$, continuum QCD is recovered

Ex1: Conjugate Gradient Method

- solve lattice Dirac equation $D\chi = h$ where

$$D_{x,y} = m\delta_{x,y} + \frac{1}{2}\sum_{\mu}\eta_{\mu}(x)\left[U_{\mu}(x)\delta_{x+\widehat{\mu},y} - U_{\mu}^{\dagger}(x-\widehat{\mu})\delta_{x-\widehat{\mu},y}\right]$$

  where $\eta_{\mu}(x) = (-1)^{\sum_{\nu<\mu} x^{\nu}}$.
- lattice Dirac operator D is sparse, but has very large dimension : $10^{6\sim 8} \times 10^{6\sim 8}$ complex square matrix
- we use Conjugate Gradient algorithm and its variants to solve this equation.

Conjugate Gradient Method

- Conjugate Gradient (CG)
  : minimizing residual $\|b - Ax\|$ for hermitian matrix A
- convergence is guaranteed and quick

- basic algorithm

$$
\begin{aligned}
& x_0 = 0, \ r_0 = b, \ p_0 = r_0 \\
& \texttt{for} \ n = 1, 2, 3, \cdots \\
& \quad \alpha_n = (r_{n-1}^\dagger r_{n-1}) / (p_{n-1}^\dagger A p_{n-1}) \\
& \quad x_n = x_{n-1} + \alpha_n p_{n-1} \\
& \quad r_n = r_{n-1} - \alpha_n A p_{n-1} \\
& \quad \beta_n = (r_n^\dagger r_n) / (r_{n-1}^\dagger r_{n-1}) \\
& \quad p_n = r_n + \beta_n p_{n-1}
\end{aligned}
$$

: A is lattice Dirac operator D.

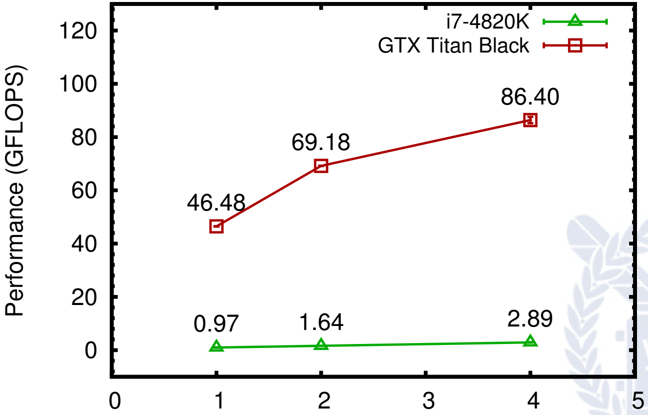
: Ap_{n-1} is dominant calculation

CPU vs GPU

	i7-4820K (1 core)	GTX Titan Black
single precision performance (TFLOPS)	0.03	5.1
double precision performance (TFLOPS)	0.015	1.3
memory bandwidth (GB/sec)	25.6	336

Conjugate Gradient Method - Performance

- CG performance on CPU (i7-4820K) and GPU (GTX Titan Blakc) (for $20^3 \times 64$ MILC asqtad ensemble)

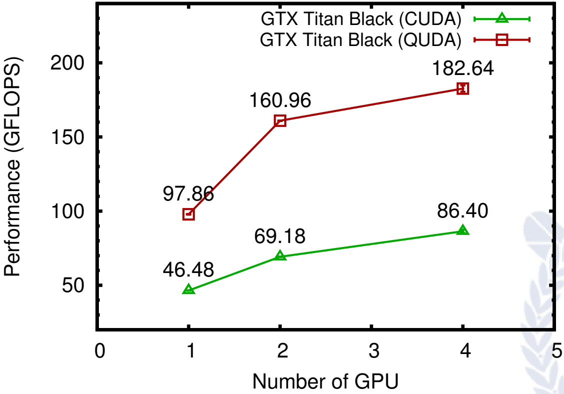


CPS & QUDA for staggered fermion

- CPS : Columbia Physics System. one of the most popular LQCD library
- QUDA : library for lattice QCD based on CUDA
- QUDA provides high performance CG and BiCG inverters of mixed precision
- We adopted QUDA staggered fermion inverter to CPS

CG Performance with New CPS & QUDA

- performance of CUDA(old code) and QUDA(new code) CG inverters (for $20^3 \times 64$ MILC asqtad ensemble)



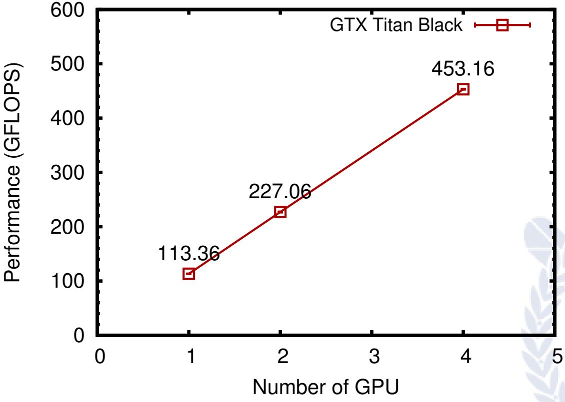
Ex2: Non Perturbative Renormalization

- Non Perturbative Renormalization (NPR) : a non-perturbative approach, but has very many calculations
- one-color four fermion operator $O^{f_1 f_2 f_3 f_4}_{i;I}$ has very many contractions, which can not be split

$$
\begin{aligned}
O^{f_1 f_2 f_3 f_4}_{i;I}(z) \quad = \quad & \overline{\chi}^{f_1}_{i;c_1}(z_A)\overline{(\gamma_{S_1} \otimes \xi_{F_1})}_{AB}\chi^{f_2}_{i;c_2}(z_B) \\
& \times\overline{\chi}^{f_3}_{i;c_3}(z_C)\overline{(\gamma_{S_2} \otimes \xi_{F_2})}_{CD}\chi^{f_4}_{i;c_4}(z_D) \\
& \times[U_{i;AD}]_{c_1 c_4}(z)[U_{i;CB}]_{c_3 c_2}(z)
\end{aligned}
$$

NPR code performance

- performance of one-color four fermion operator calculation (for $20^3 \times 64$ MILC asqtad ensemble)



Ex3: Finite Volume Correction

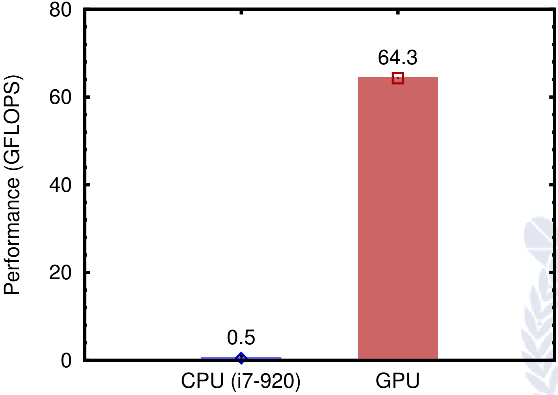
- error induced by the finiteness of lattice
- correction terms $\delta_1^{FV}(X)$ and $\delta_3^{FV}(X)$ are given by

$$\delta_1^{FV}(M^2) = \frac{4}{ML}\sum_{n\neq 0}\frac{K_1(\|n\|ML)}{\|n\|}\ , \quad \delta_3^{FV}(M^2) = 2\sum_{n\neq 0}K_0(\|n\|ML)$$

  - M : pion mass
  - $n = (n_1, n_2, n_3, n_4)$: vector of integers labeling image position on lattice
  - K_1, K_0 : standard modified Bessel functions of second kind
- need to calculate Bessel functions for a number of norms $\|n\|$

Finite Volume Correction - Performance

- performance of finite volume correction calculation on CPU(i7-920) and GPU(GTX 480) (for all gauge ensembles)



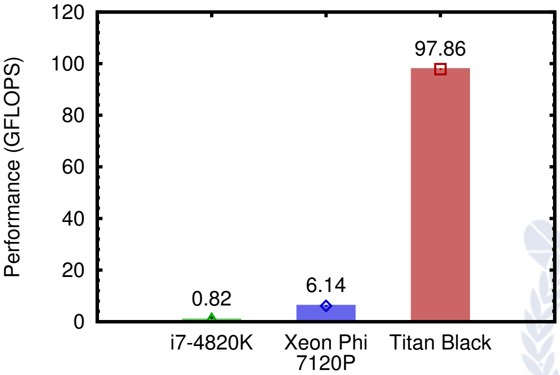
Xeon Phi

- Many CPU cores are put on a PCIe card as GPU's CUDA cores
- Easier to import a usual C code than GPU, which requires CUDA
  - OpenMP, OpenCL, MPI are available
- Provides 512 bit SIMD register
  - Simultaneous 8 double precision operations
  - Vectorization is very important.

	Intel Xeon Phi		NVIDIA GPU	
	7110X	5110P	Tesla K20X	GTX Titan black
SP TFLOPS	2.44	2.02	3.95	5.1
DP TFLOPS	1.22	1.01	1.31	1.3
Memory Size (GB)	16	8	6	6.1
Mem. Bandwidth (GB/s)	352	320	250	336
Price (USD)	4130	2650	3800	1100

CG performance

- CG performance for $20^3 \times 64$ MILC asqtad ensemble
  (For Xeon Phi, only MPI is used with 200 Xeon Phi processors)



Thank you!