BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Tuning And Understanding MILC Performance In Cray XK6 GPU Clusters

Mike Showerman, Guochun Shi

Steven Gottlieb

NCSA

GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

CRAY

CUG 2012

Outline

- Background
  - Lattice QCD and MILC
  - GPU and Cray XK6 node architecture
- Implementation in GPU with QUDA library
  - QUDA library
  - Data layout in CPU and GPU memory
  - Multi-GPU with partition over space-time dimensions
  - CG, mixed precision and multi-shift solvers
- Performance
  - Benchmarking the PCIe communication
  - Profiling Dslash with ESS and Dirac
  - Solver performance
- Conclusion and future work

Background

- Boston University: QUDA for Wilson type quarks, arXiv:0810.5365, arXiv:0911.3191
- 8/2009: Prof. Steven Gottlieb sabbatical at NCSA
- NCSA/IU extended QUDA to staggered quarks, single GPU performance reported in LATTICE'2010 and SAAHPC'2010
- We will be focusing on multi-GPU performance

Quantum ChromoDynamics

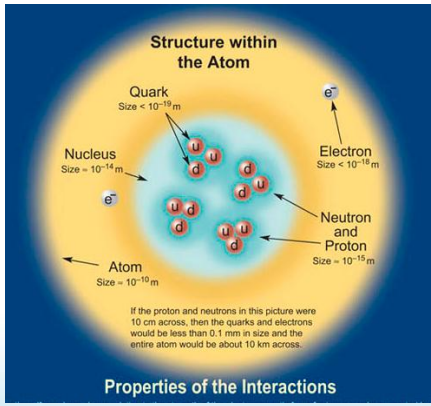
- QCD is the theory of the strong force that binds nucleons
- Impose local SU(3) symmetry on vacuum
  - Color charge analogous to electric charge of QED
- Lagrangian of the theory simple to write down

$$\mathcal{L}_{QCD} = \psi_i \left(i\gamma^\mu (D_\mu)_{ij} - m\delta_{ij} \right) \psi_j - G^a_{\mu\nu} G^{\mu\nu}_a$$

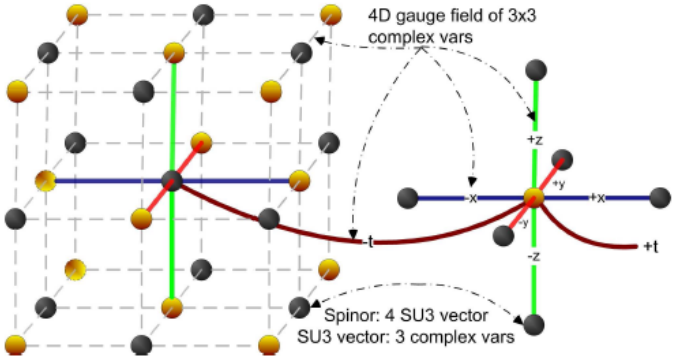
- Path integral formulation

$$\langle \Omega \rangle = \frac{1}{Z} \int [dU] e^{-\int d^4x L(U)} \Omega(U)$$

- Infinite dimensional integral
- Theory is strictly non-perturbative at low energies



Introduction: Lattice QCD



Four dimensional space-time Lattice QCD.

Diagram courtesy of K.Z.Ibrahim, IRISA/INRIA, France

- Quantum Chromodynamics (QCD) studies the strong force (color force) interaction between quarks and gluons.
- QCD is highly nonlinear problem thus lattice QCD approach is introduced. The computation is performed on 4-D space-time lattice.

Lattice QCD computation on CPU

DP HISQ run on 8192 Intrepid cores using a $64^3 \times 144$ grid ($8^3 \times 9$ per core) with $m_l = 0.1m_s$.

CG and fermion force dominate time; I/O takes about 2% of time

Activity	time(s)	per cent
CG	15696	53.2
FF	7999	27.1
GF	2575	8.7
Fat	2960	10.0
Long	17	<1
Input config.	231	≈1
total above	29480	
unaccounted	8	<1
wallclock	29488	

- Two phases in Lattice QCD computation
  - Configuration generation: compute a snapshot of state of the strong force field, Conjugate Gradient (CG), fermion force (FF), gauge force (GF) and link fattening (FAT), see above table for the time distribution in this phase.
  - Analysis: the observables of interest are computed over the configurations. Conjugate Gradient (CG) is even more dominant in this phase.
  - In short, CG is the dominant part, others are important.

Lattice QCD: Conjugate Gradient

$$M\phi = b$$

where $\phi_{i,x}$ and $b_{i,x}$ are complex vectors carrying a color index $i = 1, 2, 3$ and a four-dimensional lattice coordinate x. The matrix M is given by

$$M = 2maI + D$$

where I is the identity matrix, $2ma$ is a constant, and the matrix D (called "D slash") is given by

$$\mathrm{D_{x,i;y,j}} = \sum_{\mu=1}^{4}\left(\mathrm{U}^{F\,i,j}_{x,\mu}\delta_{\mathrm{y,x+\hat{\mu}}} - \mathrm{U}^{F\,\dagger\,i,j}_{x-\hat{\mu},\mu}\delta_{\mathrm{y,x-\hat{\mu}}}\right) + \sum_{\mu=1}^{4}\left(\mathrm{U}^{L\,i,j}_{x,\mu}\delta_{\mathrm{y,x+3\hat{\mu}}} - \mathrm{U}^{L\,\dagger\,i,j}_{x-3\hat{\mu},\mu}\delta_{\mathrm{y,x-3\hat{\mu}}}\right)$$

The linear system (3) is solved using a conjugate gradient method after recasting it in the positive definite form

$$M^\dagger M\phi = M^\dagger b.$$

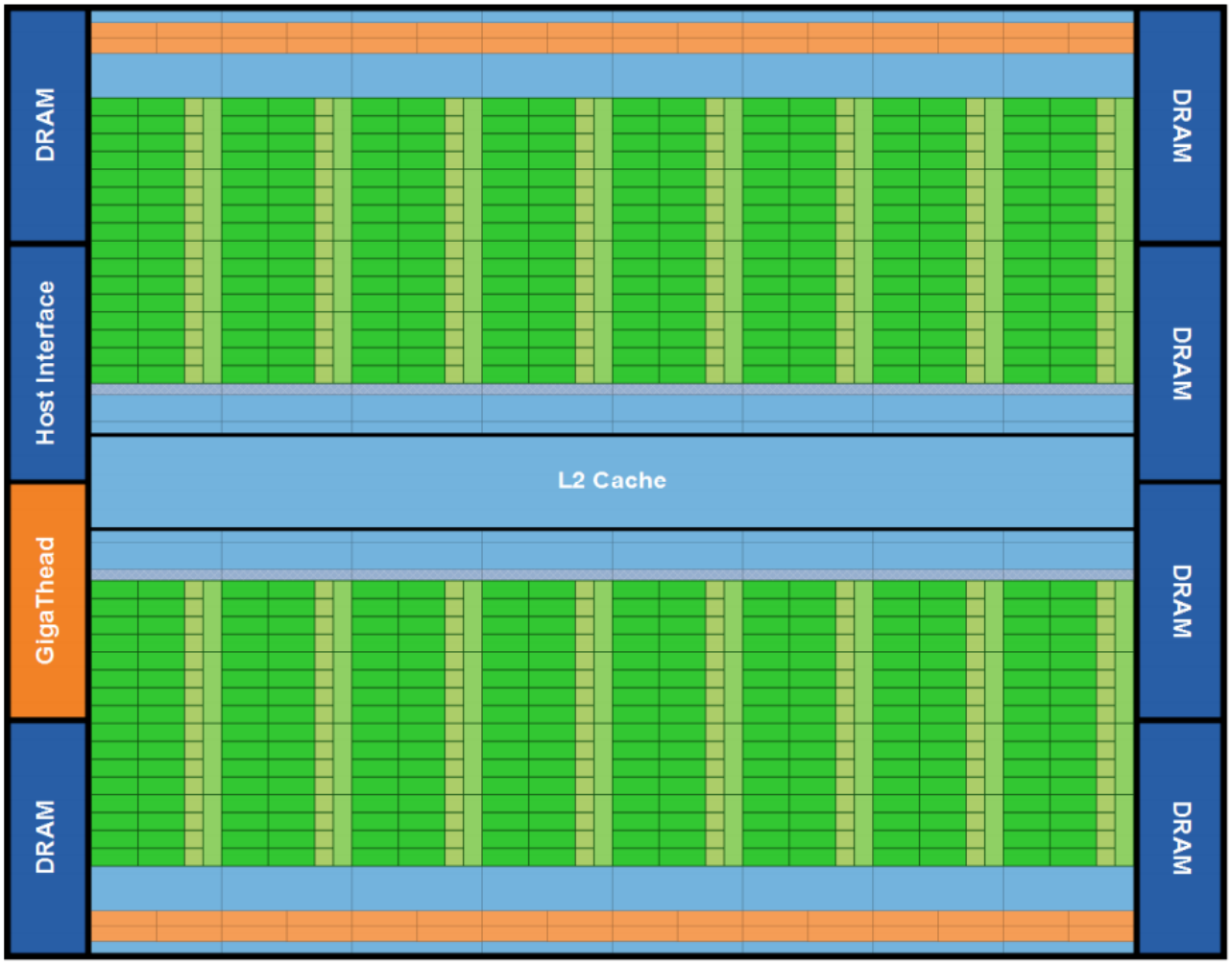
where

$$M^\dagger M = (2ma)^2 I + D^\dagger D$$

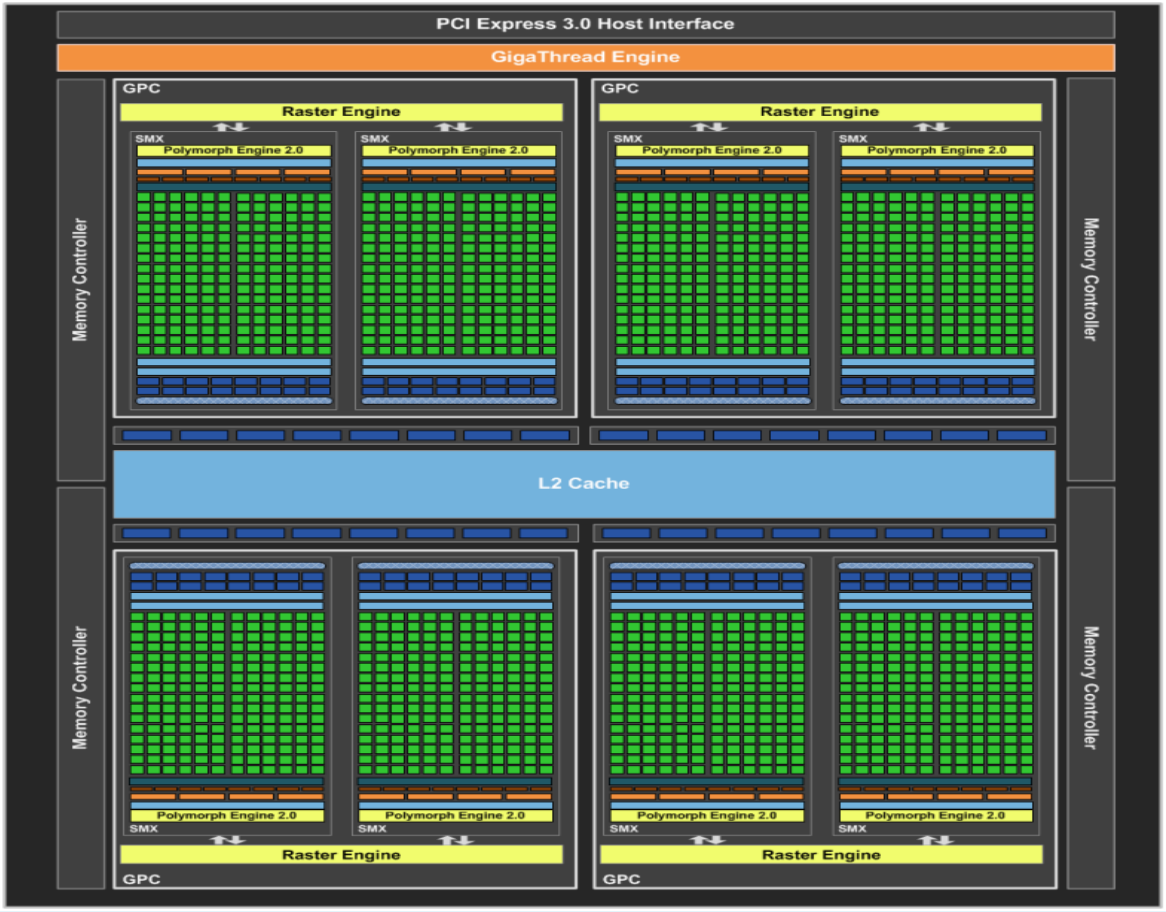
QUDA library

- A generic lattice QCD library using Nvidia's GPU written in CUDA, supporting various fermion actions.
- First initialized by Boston University
- Collaboration among multiple institutions. Developers include Ron Babich(Boston University), Mike Clark(Harvard University), Balint Joo(Jefferson Lab), Guochun Shi(NCSA, UIUC), etc
- Our work focus on Improved Staggered action.
- Open Source. Available in http://lattice.github.com/quda/

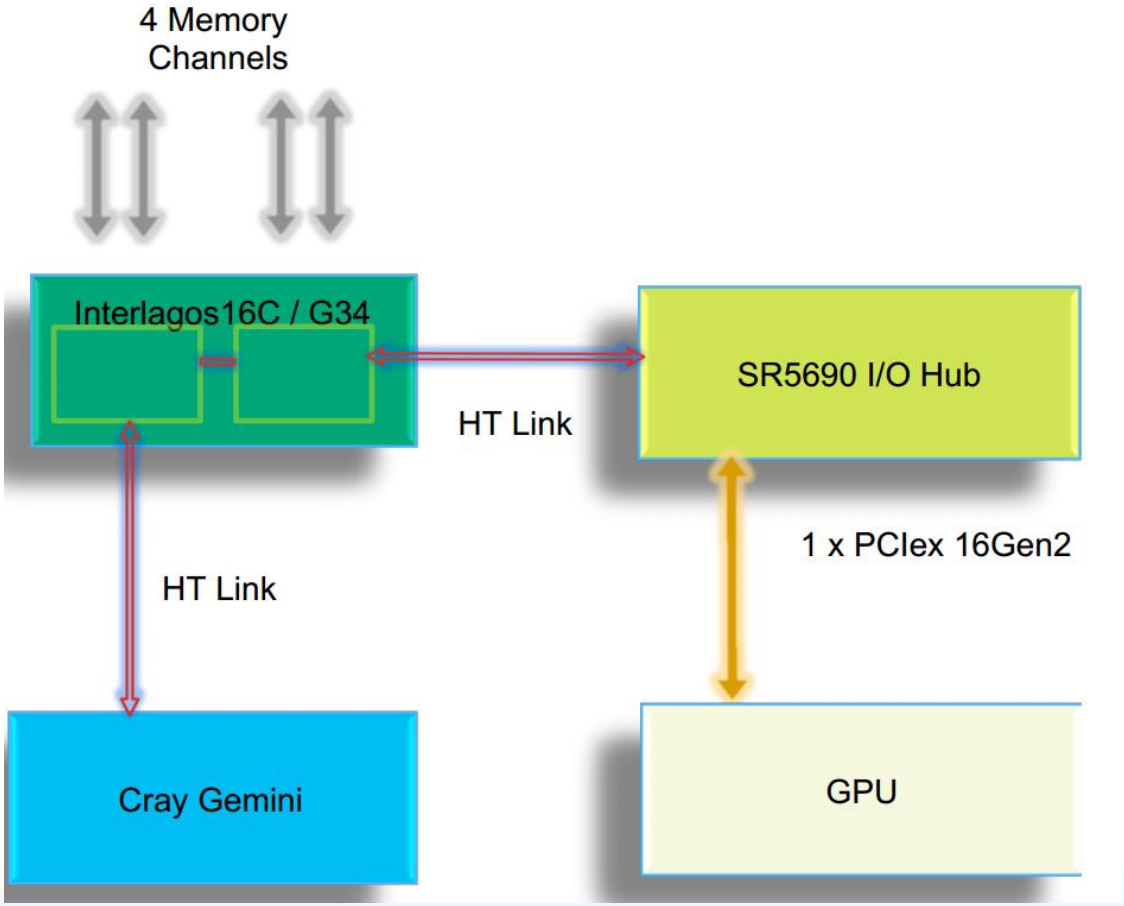
Fermi Architecture



Kepler-I architecture



Cray XK6 node

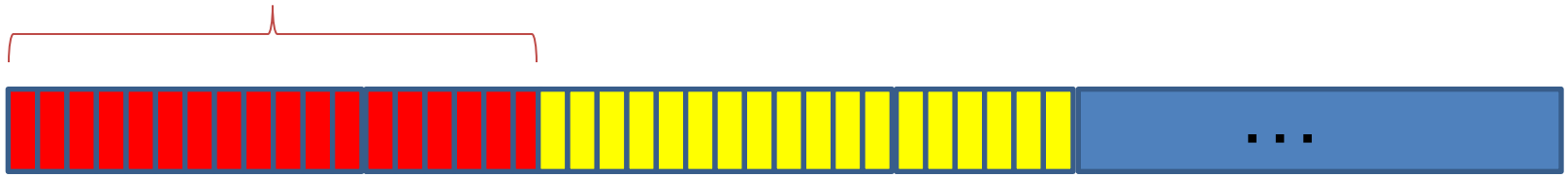


ESS system

48 cabinet
16 XK nodes
with fermi
In current use

Gauge field layout in host memory

one gauge field



Gauge field layout in GPU memory:

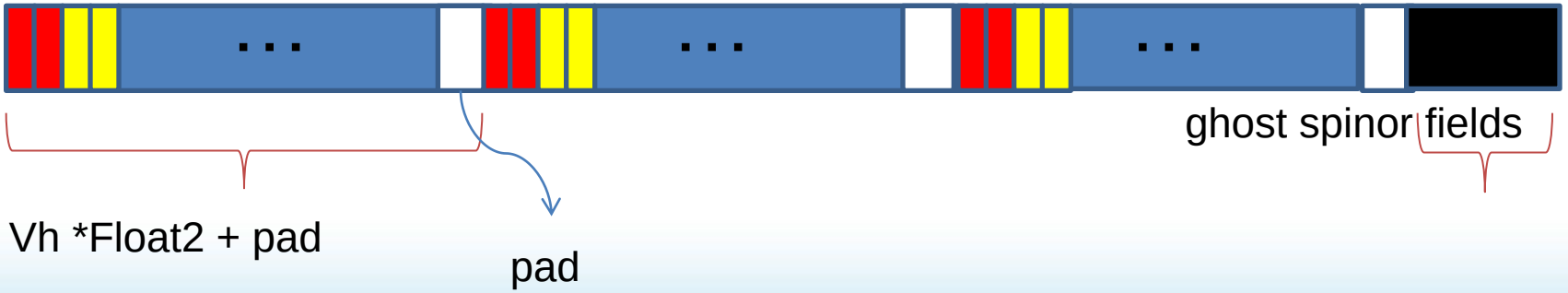
Vh *Float2 + pad

pad/ghost gauge fields

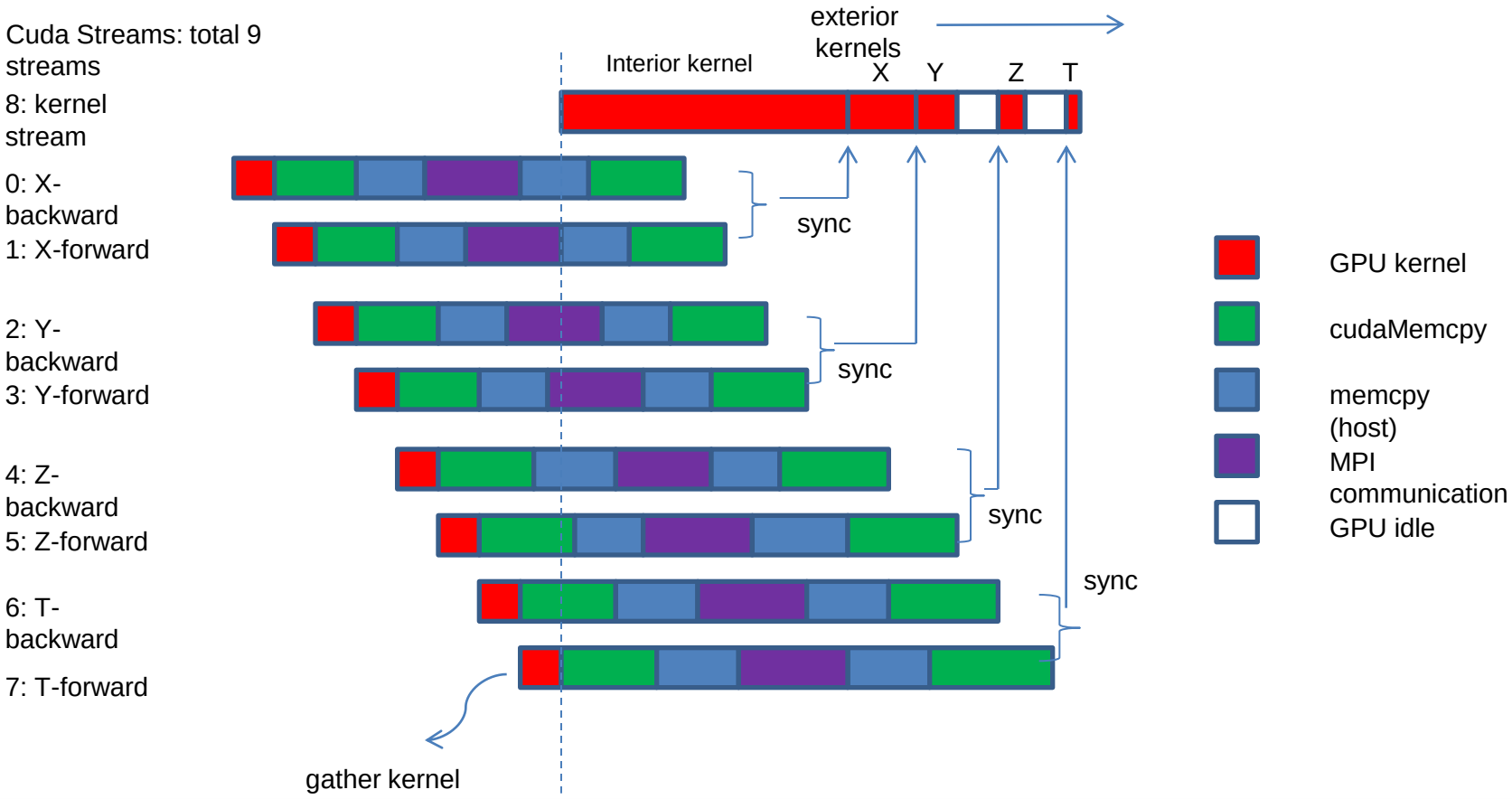
Spinor field layout in host memory:

one spinor field

Spinor field layout in GPU memory:



Computation/Communication overlap with multi-dimension partitioning



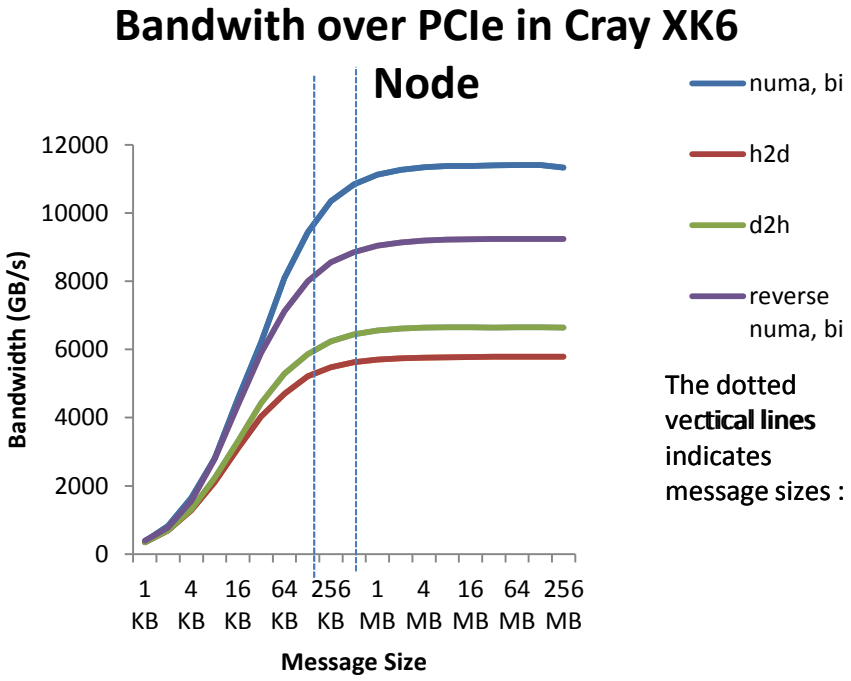
Mixed Precision CG Solver

- Both source vector *b* and the guess solution *x* are in high precision
- Majority of the work is done in lower precision
- The final solution is as accurate as high precision

Input:
D: dslash operator
A_h: $D^+D + 4ma^2$ in high precision
A_l: $D^+D + 4ma^2$ in low precision
b: source vector
x: guess solution vector
Output:
y: solution vector

$y \leftarrow 0$
$i \leftarrow 0$
$r \leftarrow b - A_h x$ D+K7
$d \leftarrow r$
$\delta new \leftarrow r^T r$ K1
$\delta_0 \leftarrow b^T b$ K1
while $i < i_{max}$ and $\delta > \delta_0 \varepsilon^2$ do
  $q \leftarrow A_l d$ D
  $\alpha \leftarrow \delta new / (d^T q)$ K2
  $\delta_{old} \leftarrow \delta_{new}$
  $r \leftarrow r - \alpha q$ K3
  $\delta_{new} \leftarrow r^T r$
  $\beta \leftarrow \delta_{new} / \delta_{old}$
  if (high-precision update is needed)
    $x \leftarrow x + \alpha d$ K4
    $y \leftarrow x + y$ K5
    $r \leftarrow b - A_h y$ D+K7
    $\delta_{new} \leftarrow r^T r$ K1
    $x \leftarrow 0$
    $\beta \leftarrow \delta_{new} / \delta_{old}$
    $d \leftarrow r + \beta d$ K8
  else
    $x \leftarrow x + \alpha d$ K6
    $d \leftarrow r + \beta d$
  $i \leftarrow i + 1$

Measuring the PCIe bandwidth with different message size

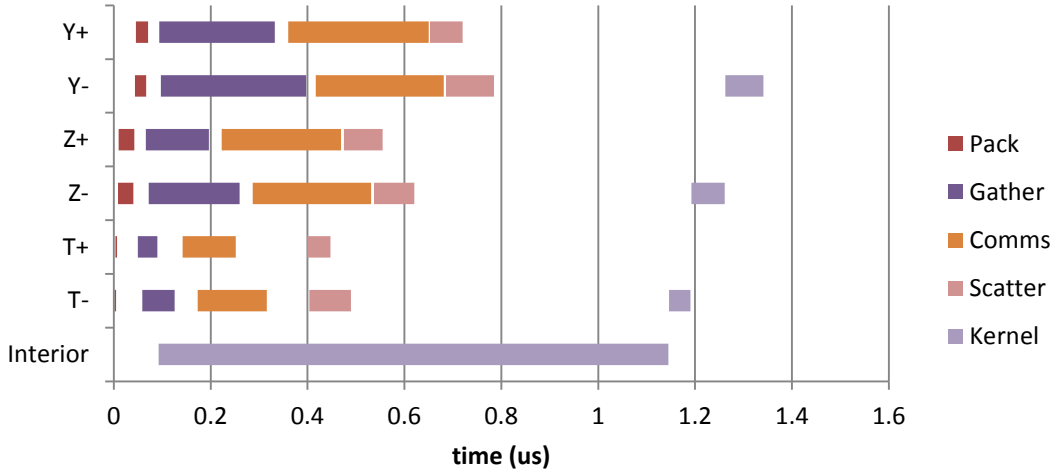


	Measured Peak (GB/s)
h2d	5.7
D2h	6.5
Bi-dir numa	11.3
Bi-dir reverse numa	9.1

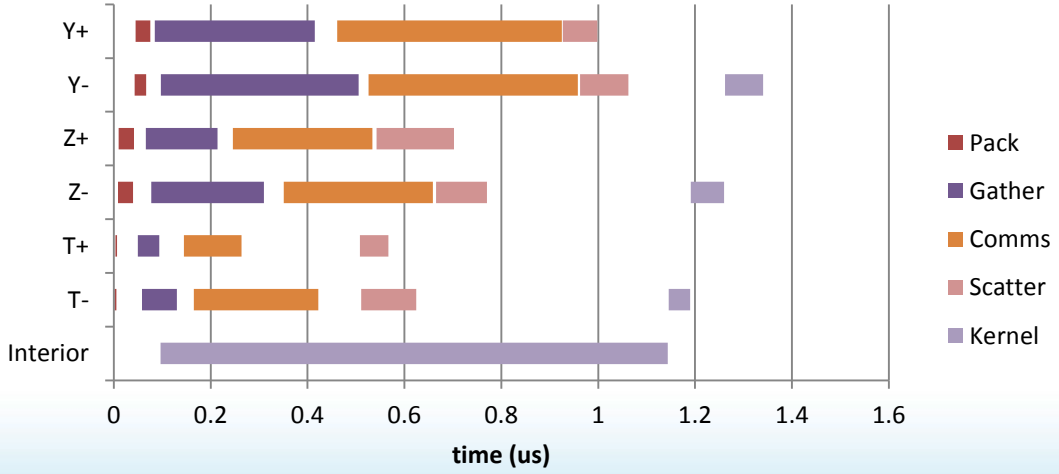
- NUMA effects are not seen in d2h or h2d measurement alone
- However it is obvious in bi-directional measurement

Dslash timeline difference in ESS

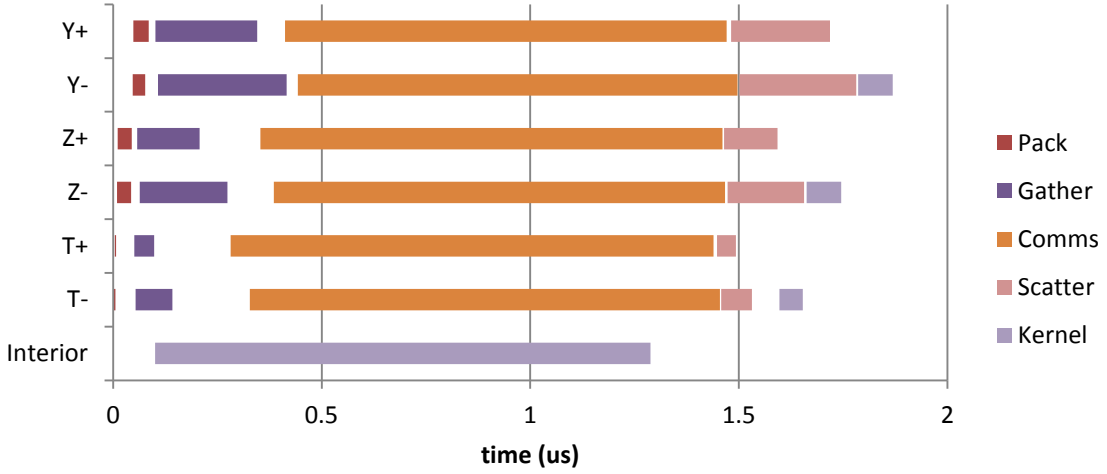
Optimal NUMA binding



Sub-optimal NUMA binding



Dslash time line in Dirac cluster from NERSC



- The communication time exceeds that of the interior kernel
- The MPI communication is much large than that of ESS
- All MPI communication finished roughly at the same time, creating contention for the scatter phase

The achieved bandwidth for different communication stages

Numa node	Comm Stage	Gather (GB/s)	Inter-process Comms (GB/s)	Scatter (GB/s)
Correct NUMA	T-	2.9	2.8	2.3
	T+	4.9	3.6	4.2
	Z-	1.8	2.8	4.1
	Z+	2.6	2.8	4.2
	Y-	1.1	2.6	3.4
	Y+	1.4	2.3	5.0
Sub-optimal NUMA	T-	2.8	1.5	1.7
	T+	4.5	3.3	3.4
	Z-	1.5	2.2	3.2
	Z+	2.3	2.4	2.1
	Y-	0.8	1.6	3.4
	Y+	1.0	1.5	4.8

The final mixed precision multi-mass solver performance

Machine	Gflops/GPU
ESS with correct NUMA	51.1
ESS with sub-optimal NUMA	50.3
Dirac	36.1

Conclusion and future work

- XK6 shows good performance due to its GPUs and its Gemini inter-connect
- NUMA is important in understanding and optimizing the performance
  - More understanding necessary for memory layout
- We are actively working on porting and optimizing the gauge generation code into multi-GPUs