Lattice QCD on GPU Clusters, using the QUDA library and the Chroma Software System

Bálint Joó (Jefferson Lab)

Mike Clark (Harvard Smithsonian)

ICCS 2011, Berkeley CA

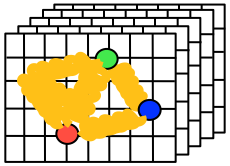
Friday, January 28, 2011

A very brief introduction to QCD

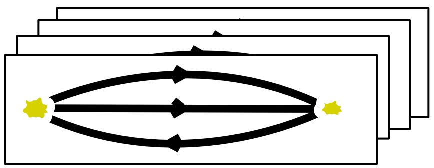
- Quantum Chromo-Dynamics (QCD) is a theory of quarks, gluons and their interaction via the 'strong nuclear force'
- Lattice QCD (LQCD) is a version of QCD amenable to computing
  - discretize space-time as a finite lattice (4D array)
- LQCD underpins many current calculations in nuclear and high energy physics
  - Spectrum of Hadrons (resonances, hybrids)
  - Nuclear Interactions (binding of quarks into light nuclei)
  - Test of Standard Model (e.g CKM matrix elements)
  - Beyond the Standard Model (BSM) physics

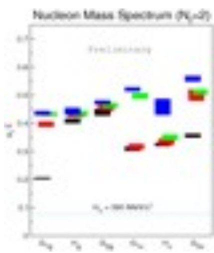
Large Scale LQCD Simulations Today

- Stage 1: Generate Configurations
  - via Markov Chain Monte Carlo
  - single chain
  - requires large capability machine



- Stage 2: Analysis of Configurations
  - soon/now more FLOPS than gauge generation
  - **BUT** task parallelizable (per configuration)
  - each task still numerically intensive
  - on large capacity clusters or multiple smaller partitions of capability machine
  - on clusters of GPUs





- Stage 3: Extract Physics
  - on workstations, small cluster partitions

QCD on GPUs

- First report: “Lattice QCD as a Video Game”, (Egri et. al.) in 2006
  - Coded in OpenGL
- In the U.S.
  - Wuppertal Group used for BSM Studies (Kuti et. al.)
  - QUDA Library (focus of this talk)
  - Alexandru et. al. @ GWU
- Worldwide (not in any particular order, not necessarily complete)
  - Wuppertal Group (Z. Fodor et. al.)
  - IRISA - France (F. Bodin et. al.)
  - Pisa - Italy (A. Di Giacomo et. al.)
  - Japan (KEK) (Hayakawa et. al.)
  - Taiwan (T-w Chiu et. al.)

The Main Problem

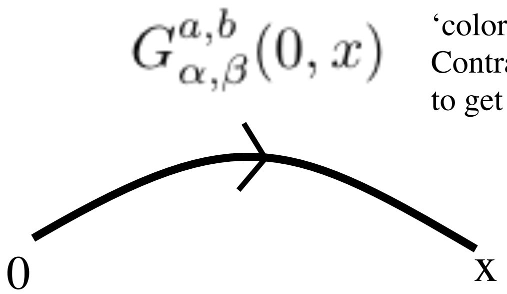
- Propagation of quarks is described by the matrix equation:

color indices — Fermion Matrix

$$G^{a,b}_{\alpha,\beta}(x,y) = [M^{-1}]^{a,b}_{\alpha,\beta}(x,y)$$

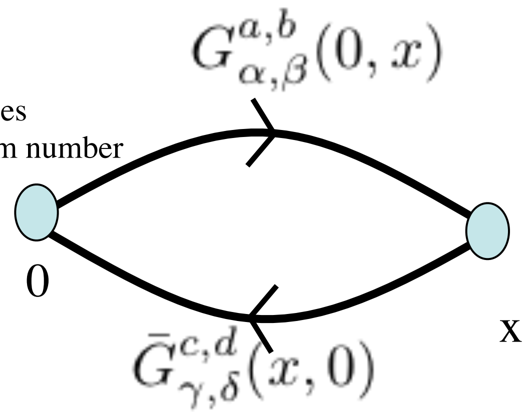
spin indices — 4d lattice sites (linearized to 1d)

Quark:



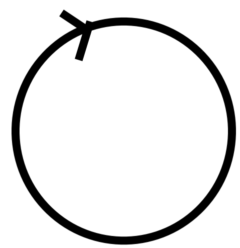
Meson:

Contract color indices to leave 'colorless' objects
Contract spin indices to get right quantum number



Contract color indices to leave 'colorless' objects
Contract spin indices to get right quantum number

Vacuum 'bubble':



The Fermion Matrix

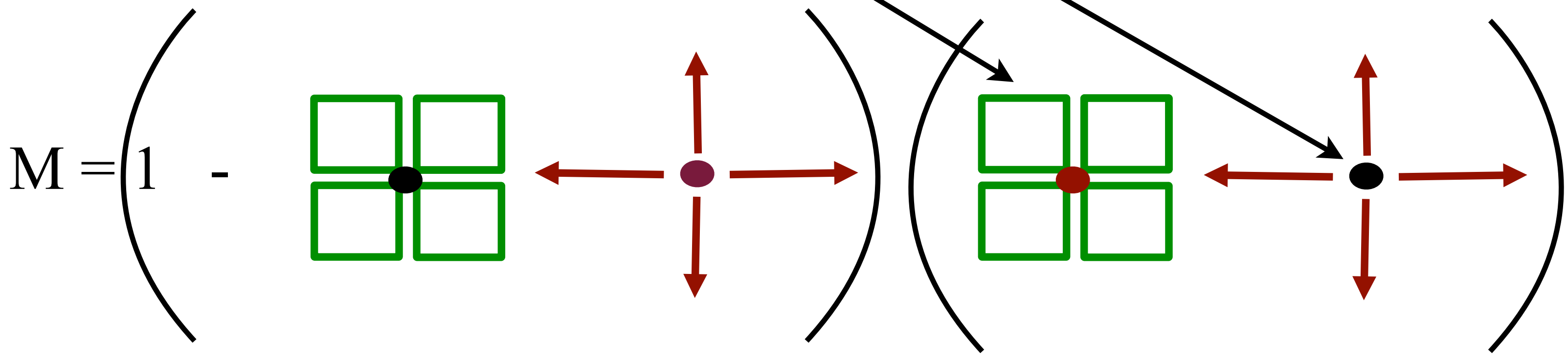
- Several formulations of the Fermion Matrix (sacrifice different symmetry)
- Concentrate on the so called 'Wilson-Clover' version in this talk
- Wilson Clover matrix M is complex with properties:
  - Dimension=12 V: $V=32^3 \times 256$ sites that's ~100M
  - γ_5-Hermiticity (aka J-Hermiticity); $\gamma_5 M = M^\dagger \gamma_5$
  - Sparse with regular structure (nearest neighbor)
    - Store only fields which occur in M
    - Compute Mv
  - Condition increases as $(1/a)^\alpha (1/m_q)^\beta$, $\alpha, \beta > 0$
  - Typically solved by CG or BiCGStab
- Need to solve:
  - Propagators: $Mx = b$
  - Force terms in gauge generation:

$$(M^\dagger M)x = b, \quad (M^\dagger M + c_i)x = b$$

The Fermion Matrix

$$M = 1 - A_{oo}^{-1} D_{oe} A_{ee}^{-1} D_{eo}$$

total: 1824 flops,
408 words in + 24 words out
FLOP/Byte: 1.06 (SP), 0.53 (DP)



SU(3) matrix

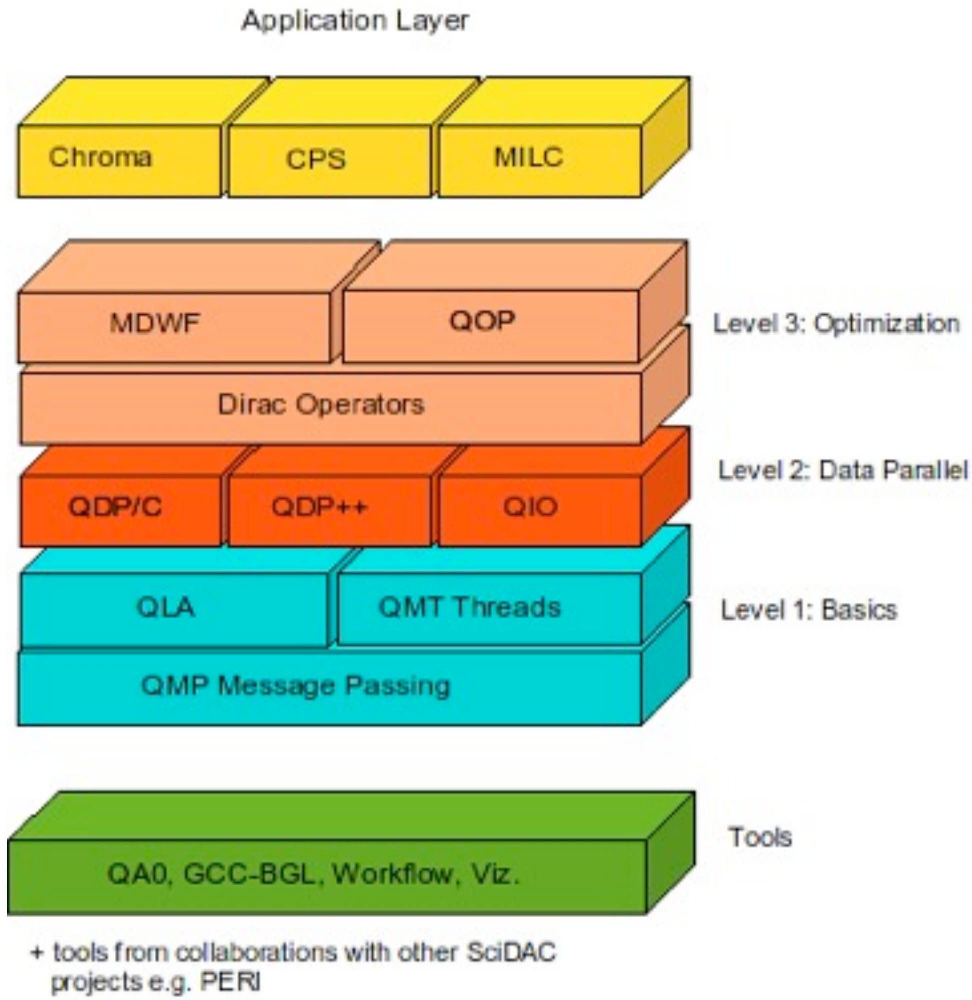
permutes spin components, flips signs

'get nearest neighbour' from forward μ direction

$$D_{x,y} = \frac{1}{2} \sum_{\mu=0}^{4} U_\mu(x) \otimes (1 - \gamma_\mu) \otimes \delta_{x+\hat{\mu},y} + U_\mu^\dagger(x - \hat{\mu}) \otimes (1 + \gamma_\mu) \otimes \delta_{x-\hat{\mu},y}$$

The Chroma Library

- Developed as part of USQCD SciDAC project
- Free to download, distribute, modify
- Built on USQCD layered structure
- Object oriented design:
  - Can integrate Level 3 libraries by wrapping up as Chroma objects (e.g. QUDA solvers)
  - Users see Chroma 'look and feel'
  - Allows integration of solvers with LARGE library of analysis tasks
  - Chroma Unit tests help debug external libraries & vice versa
- Bring Benefits of GPU to large worldwide user base

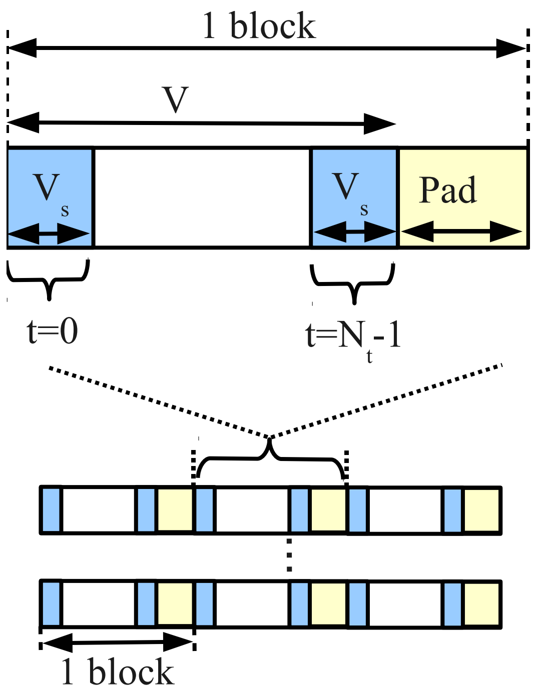


The QUDA Library

- Highly Successful GPU Library
- Started off at Boston University:
  - Kipton Barros, Ron Babich, Rich Brower, Mike Clark, Claudio Rebbi
- Attracted developer community
  - parallelized over multiple GPUs
  - QUDA is now on GitHub (lattice/quda)
- Current QUDA development team:
  - Now: Mike Clark (Harvard), Ron Babich (BU) - lead developers
  - Guochun Shi (NCSA) - staggered quark development & MILC
  - Bálint Joó (JLab) - Chroma integration
  - Rich Brower, Claudio Rebbi
  - others: Joel Gieldt (domain wall), Will Detmold, ...

QUDA Optimizations

- Data Layout tuned for Memory Coalescing
  - 1 thread / lattice site,
  - break up data for site data into chunks (e.g. float4 for SP)



Single Precision Example:

- blocks of V float4s
- Vs surface float4s
- #of blocks depends on data types
  - spinor: 24 floats -> 6 blocks
  - 8 real storage SU(3) matrix: 8 floats -> 2 blocks
  - 2 rows of SU(3) matrix: 12 floats -> 3 blocks
  - full SU(3) matrix: 18 floats -> 5 blocks
- Add Pad to avoid 'partition camping'

QUDA Tricks: Compression

- Bandwidth reduction through compression
  - Store 3x3 SU(3) matrix as 6 complex numbers, or 8 reals
  - spend ‘free’ flops to uncompress
    - For DP no compression is best - not enough free flops

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{pmatrix} \quad \begin{aligned} \mathbf{a} &= (a_1, a_2, a_3) \\ \mathbf{b} &= (b_1, b_2, b_3) \\ \mathbf{c} &= (\mathbf{a} \times \mathbf{b})^* \end{aligned} \longrightarrow \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix}$$

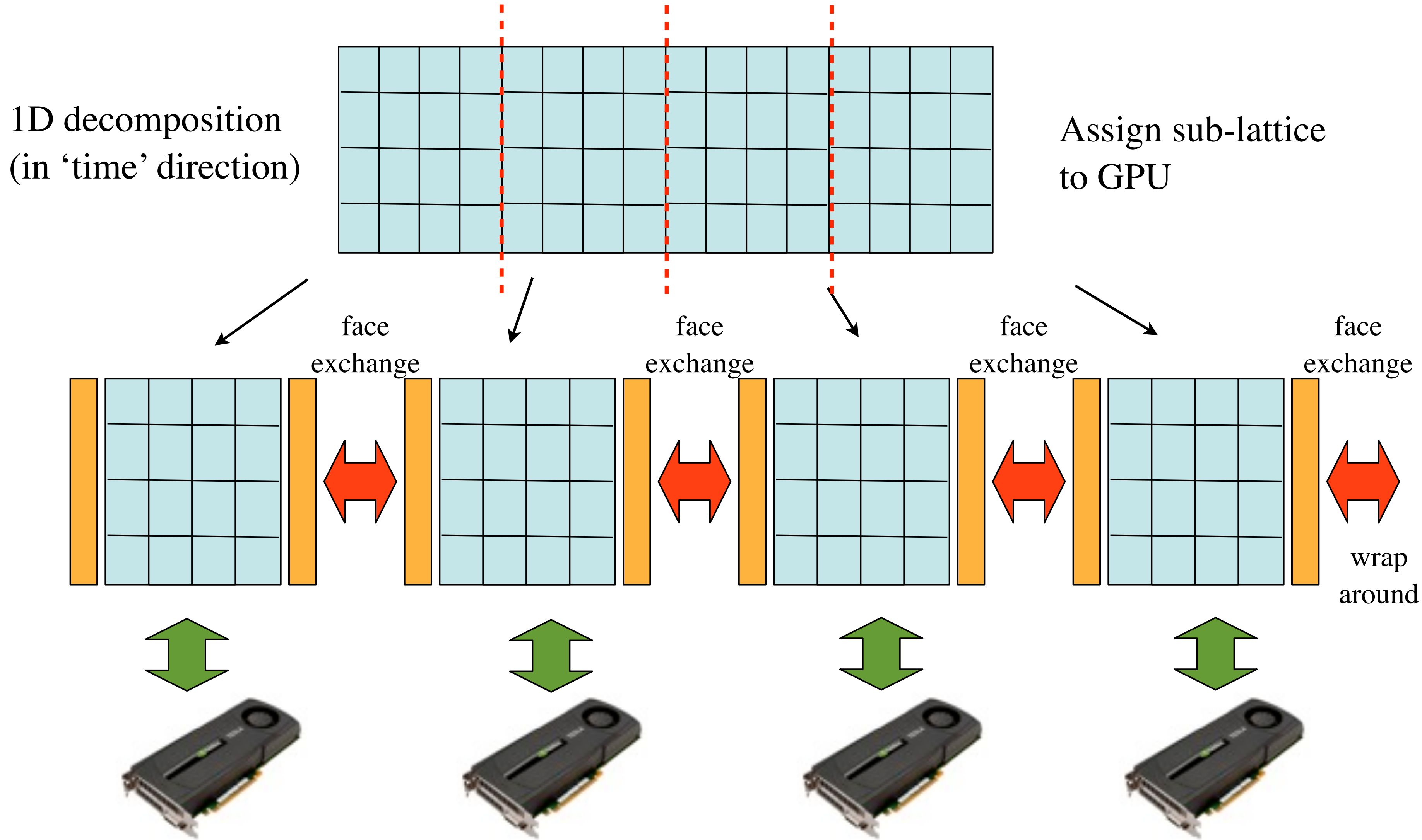
Other QUDA Tricks

- Choice of basis: save loading of 24 words (6 complex forw. + back)
- Axial Gauge Fixing:
  - LQCD has a symmetry: Invariance under local rotations in color space (gauge invariance)
  - Rotate temporal links to be unity (Axial Gauge)
    - saves the loading remaining 24 words
    - use for double precision (rounding errors from rotations)
- Mixed precision solver (details to follow)
  - using 16-bit fixed point precision
  - Gauge Links are SU(3) so elements ~1...
- Fusion of BLAS operations (eg: y = ax + y; norm2(y))
- Autotuning: BLAS via 'make tune', Dslash at runtime

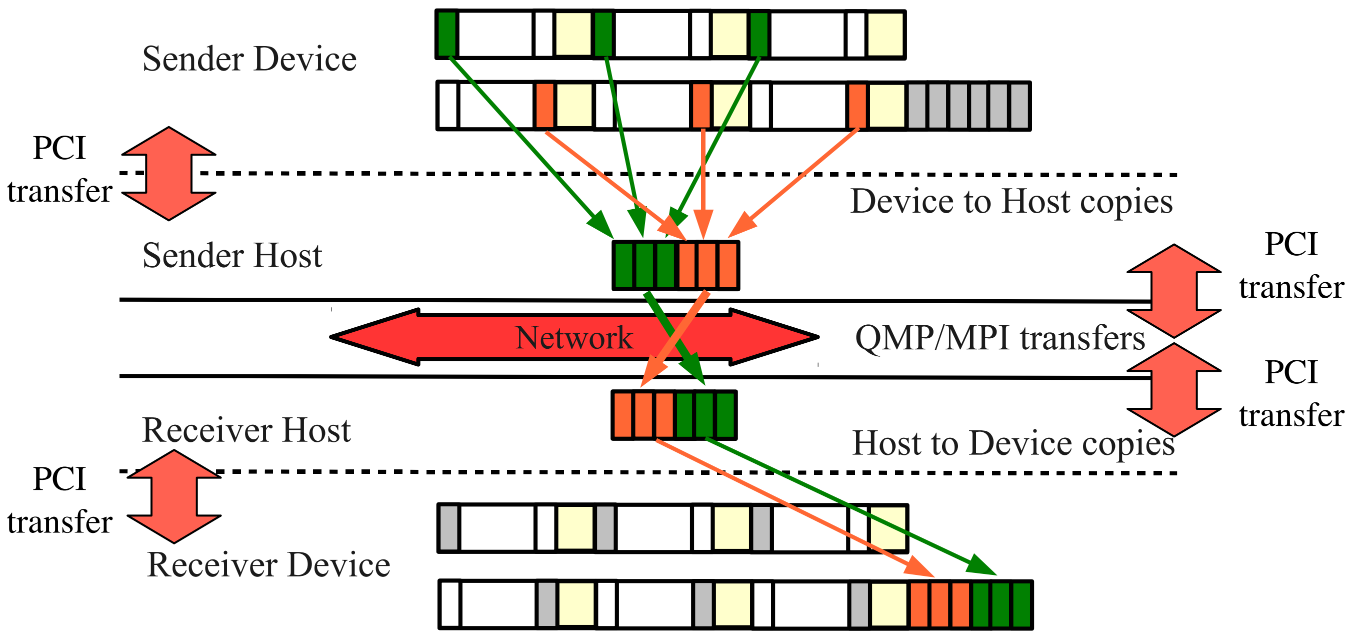
Multi-Precision Solvers

- Multi-precision solvers (Conjugate Gradients/BiCGStab)
  - 'reliable updates'
  - Originally designed to control stagnation in BiCGStab
    - convert to scale invariant form:
      - $M\,x = b \;\;\Rightarrow\;\; M\,e = r_0,\;\; e = x - x_0,\;\; r_0 = b - M\,x_0$
    - drop residuum of new system by factor γ in reduced precision
    - group update solution by current e (x += e)
    - 'flying restart':
      - recompute r_0 in full precision,
      - re-convert to scale invariant form
      - recursion coefficients not affected by 'flying restart'
      - no need to throw away built up Krylov space

QUDA Parallelization



Message Passing



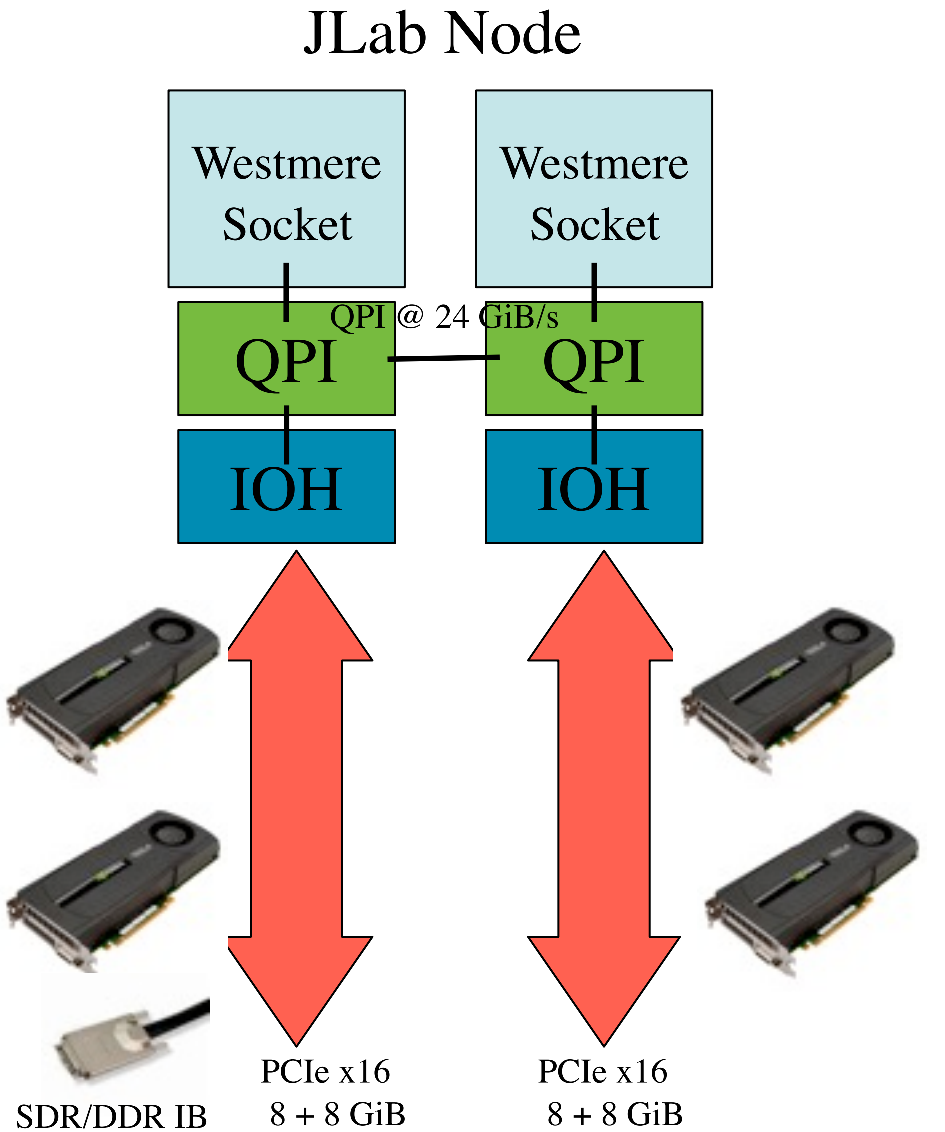
- Gather block surfaces into send buffer on host
  - # blocks cudaMemcpyAsync-s per face
- Exchange faces: 1 message for each face
- Scatter face into 'tail' receive buffers
  - 1 cudaMemcpyAsync per face

Systems

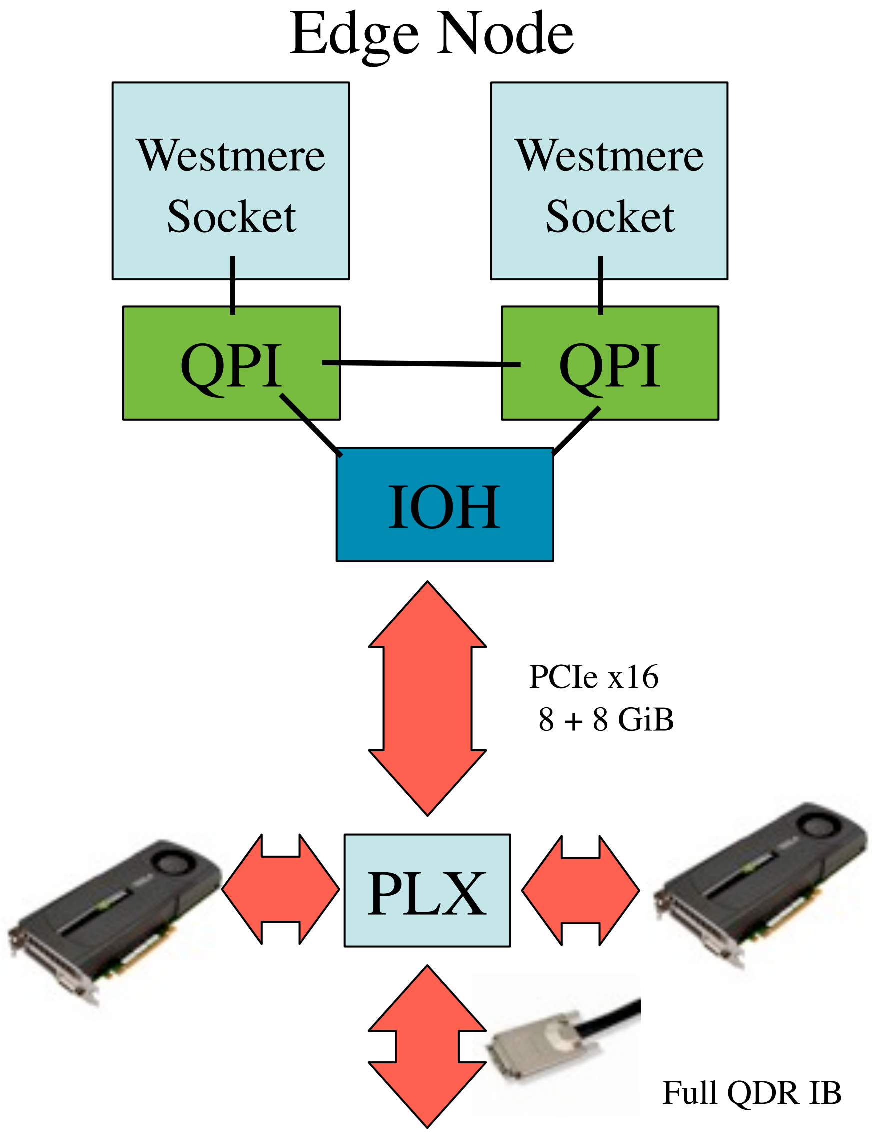
- JLAB ARRA GPU Cluster
  - 2x4 core Intel E5630, 2.53 GHz (Westmere)
  - CentOS 5.3 Linux
    - 2.6.18-128.7.1.el5
  - CUDA Toolkit v3.0
  - CUDA Driver: 195.36.24
  - 4 GPU Units / node
  - IB Card in 1/2 PCI Slot
- 2 IOH Chipsets, 2 PCIe Buses
- 32x4 C2050s (128)
- 18x4+24x4 GTX480 (168)
- 156 x GTX285

- LLNL Edge Cluster
  - 2x6 core Intel X5660, 2.8 GHz (Westmere)
  - CHAOS Linux (CentOS 5 like)
    - 2.6.18-103chaos
  - CUDA Toolkit v3.0 (and 3.2rc12)
  - CUDA Driver: 260.19.12
  - 2 GPU Units / node
  - IB at full B/W
- Single PCIe bus, PLX Switch
- 206 x 2 M2050

Buses...



JLab Node
Westmere Socket
Westmere Socket
QPI @ 24 GiB/s
QPI
QPI
IOH
IOH
SDR/DDR IB
PCIe x16
8 + 8 GiB
PCIe x16
8 + 8 GiB

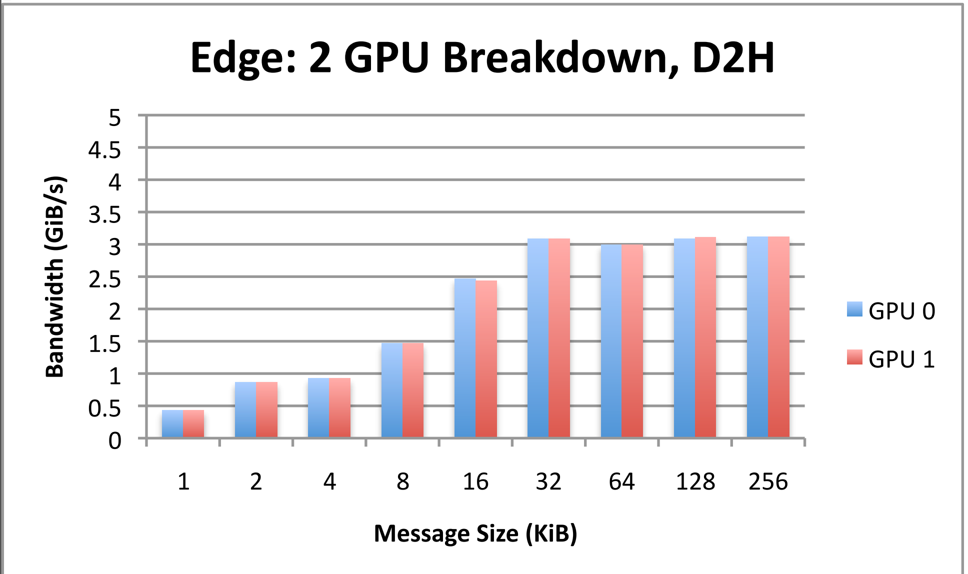


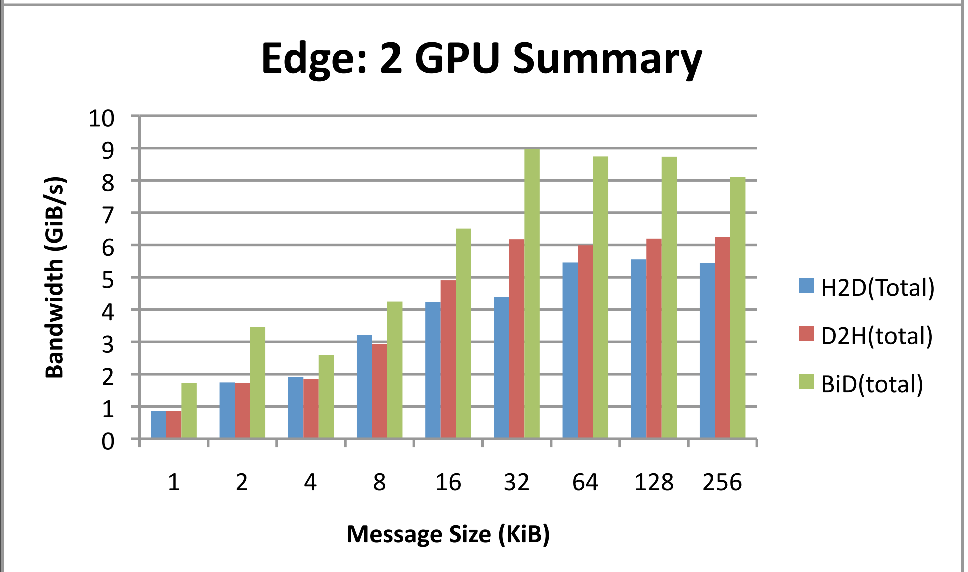
Edge Node
Westmere Socket
Westmere Socket
QPI
QPI
IOH
PCIe x16
8 + 8 GiB
PLX
Full QDR IB

Memory B/W tests.

- Host-to-device (H2D), Device-to-Host (D2H) and Bi-Directional (BiD) tests.
- Bandwidth Test Kernels use zero-copy
  - Use device kernel to copy between host/device
  - BiD test uses single stream/kernel
- Multiple MPI processes run concurrently
  - Edge: up to 2 MPI Processes
  - JLab: up to 4 MPI Processes
- Barrier AFTER each process finished its own timing loop
- Gather individual timings and sum bandwidths.
- Measure latency around 2.8usec for both systems but caveat
  - This may not be a good way of measuring latency

Edge: 2 GPU Bandwidths

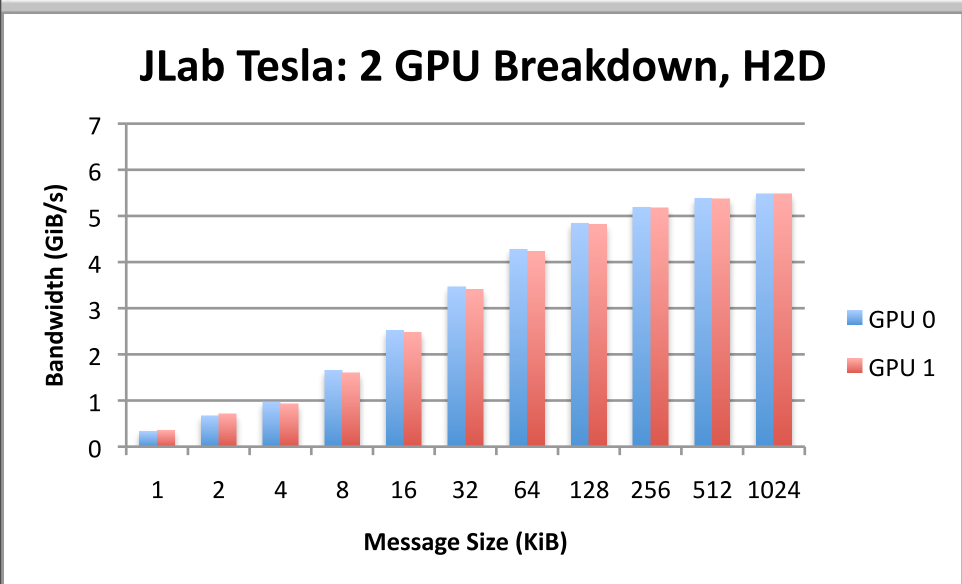


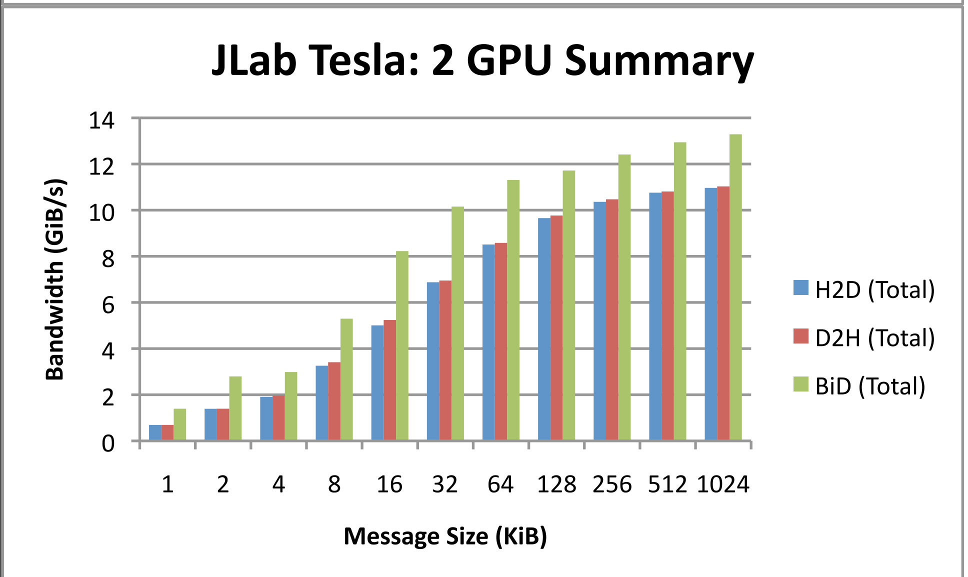


- Single PCIe x16 bus
  - max 8 GiB/s each way
  - expect maybe 5-6 GiB/s each way. 10-12GiB/sec BiD
- B/W distributed evenly between GPUs

- Slight asymmetry between H2D & D2H
- BiD ~1.3x D2H - only 8.5 GiB

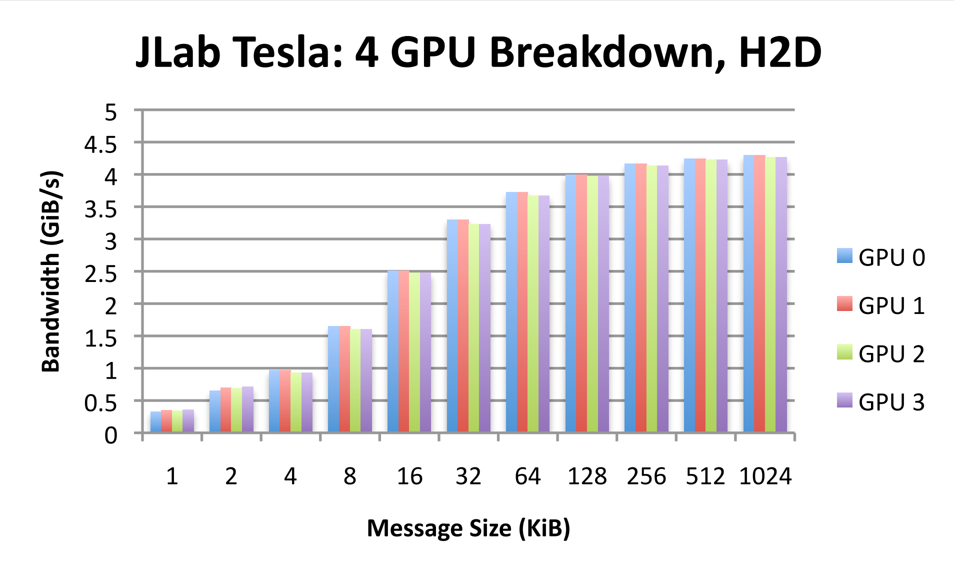
JLab Tesla: 2 GPUs

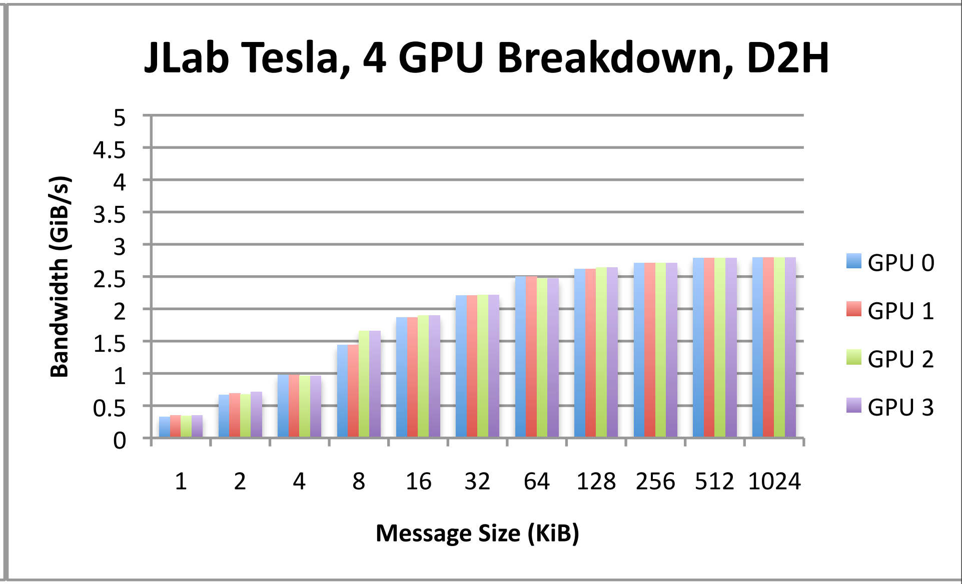




- 2 PCIe x16 buses
- Best case scenario
  - 1 GPU on each bus (other 2 GPUs in box are idle)
  - B/W evenly distributed between both GPUs
  - requires careful pinning of MPI process to socket with corresponding GPU
  - D2H breakdown similar
  - Smaller relative gain from BiD (~1.2x) than for Edge

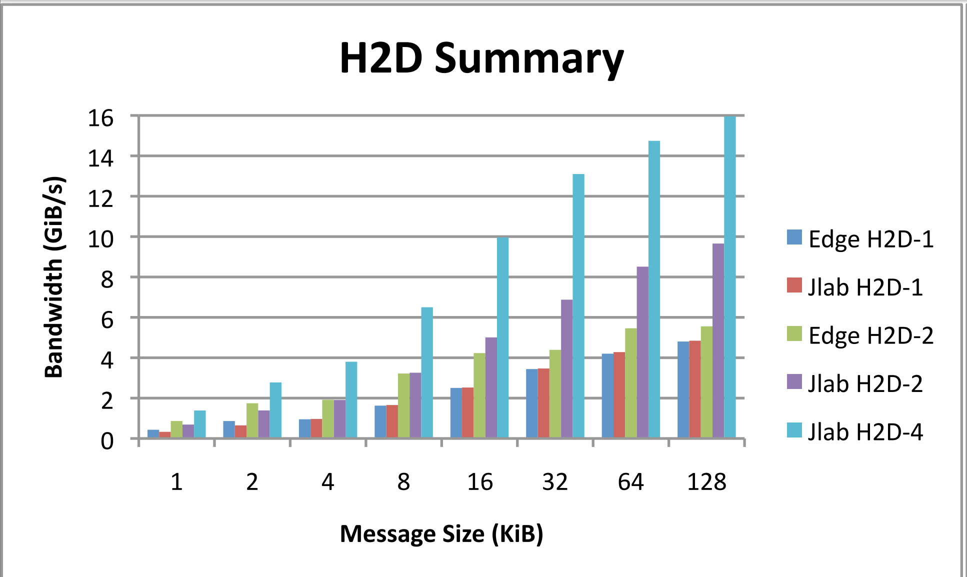
JLab Tesla: 4 GPUs

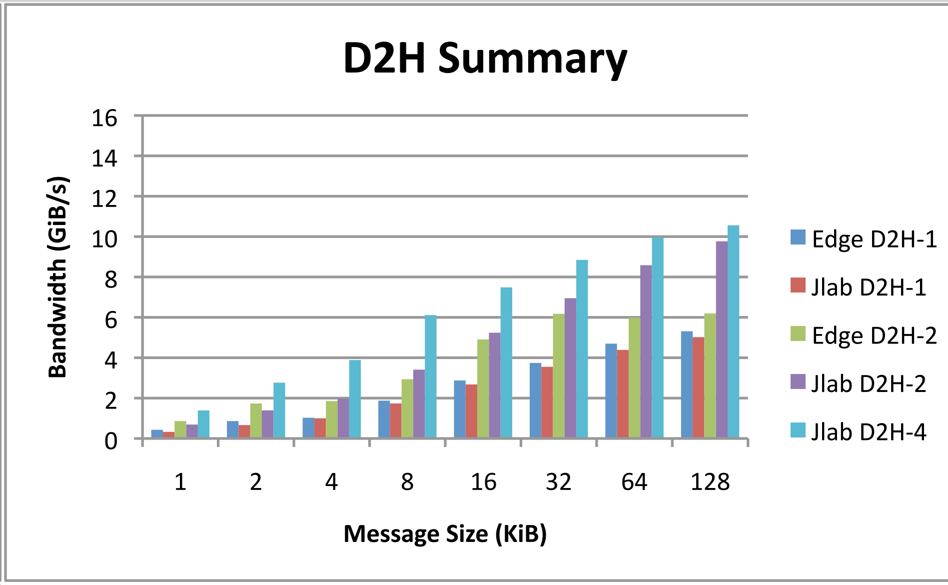


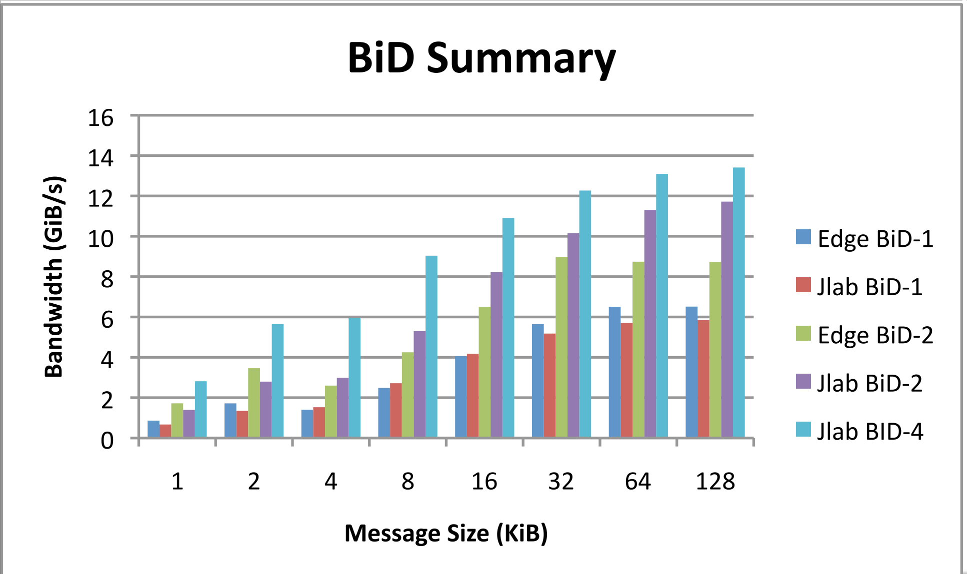


- Puzzling Asymmetry between D2H and H2D B/W for 4 GPU
- Can see this also for 2 GPU if both GPUs bound to same socket
  - i.e.: GPU 0 and GPU 1 bound to socket 1
- B/W is divided equally between all GPUs for both H2D and D2H

Comparison

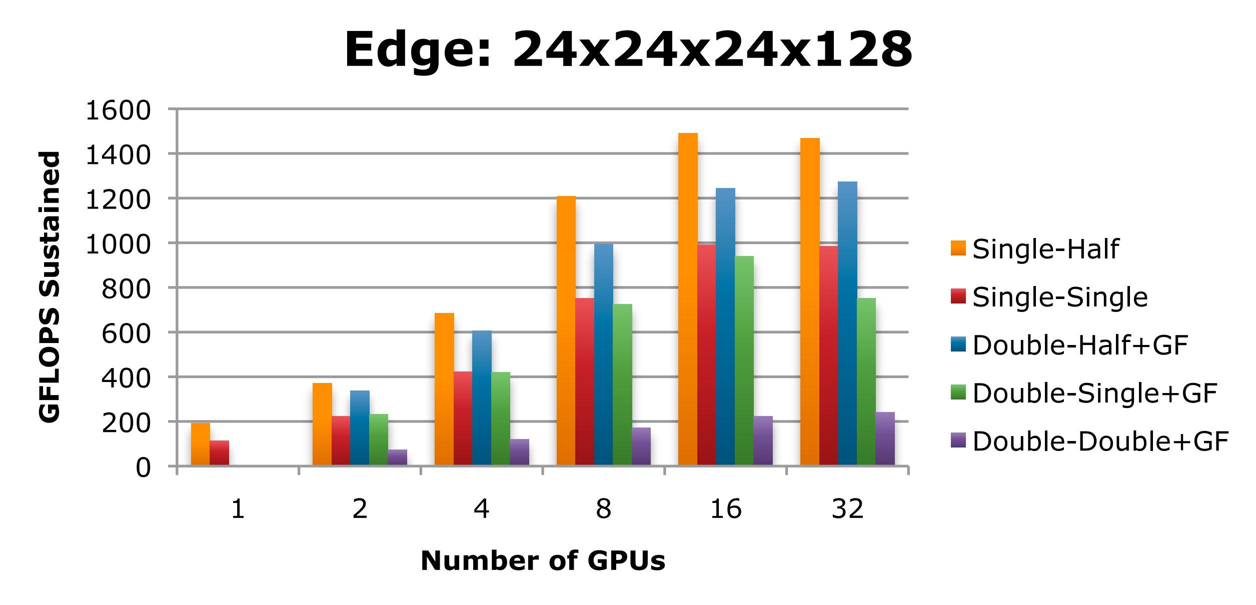


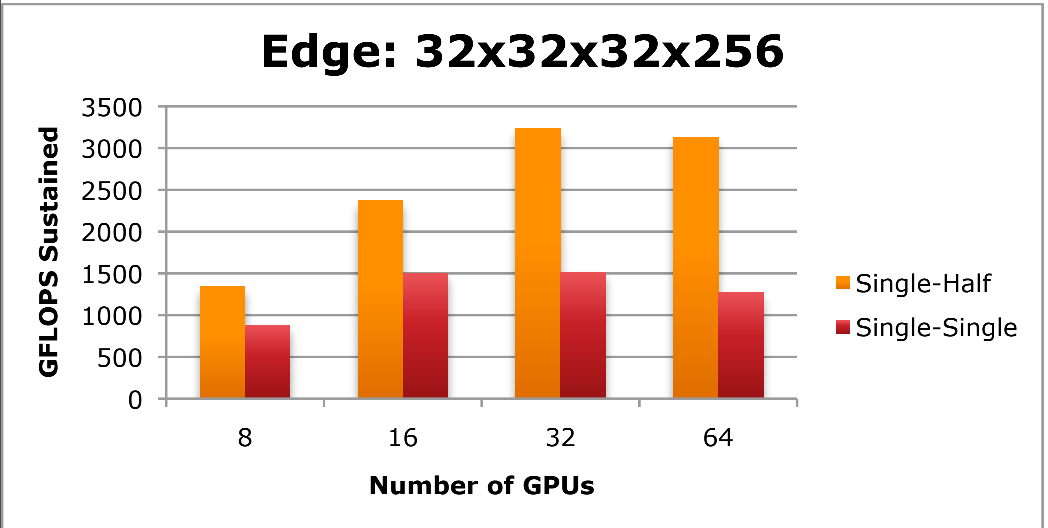




- 1 Edge GPU ~ 1 JLab Tesla GPU
- JLab 2 GPU gets significantly more B/W for larger messages
- JLab 4 GPU dragged down by D2H…

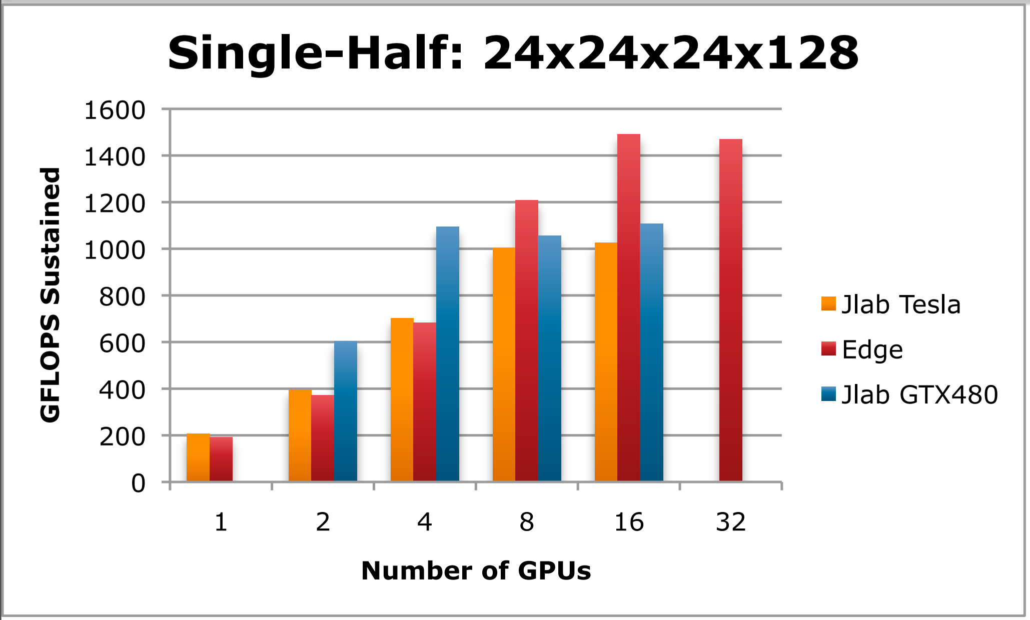
Edge Inverter Performance

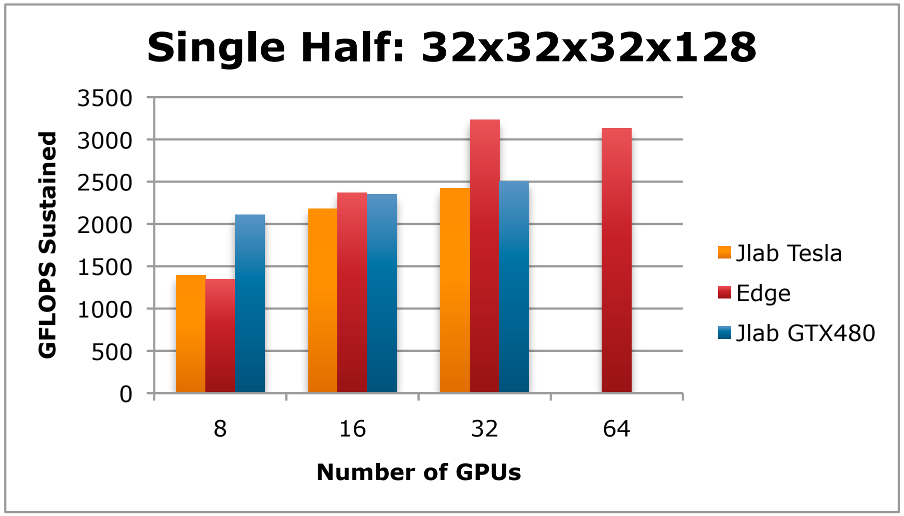




- Single-Half mixed precision is fastest
- 24^3 levels off at 16 GPUs
- 32^3 scales out to 32GPUs
  - S-S to 16 only
- 3 TFlops
- No DP 32^3 results :(

Comparison

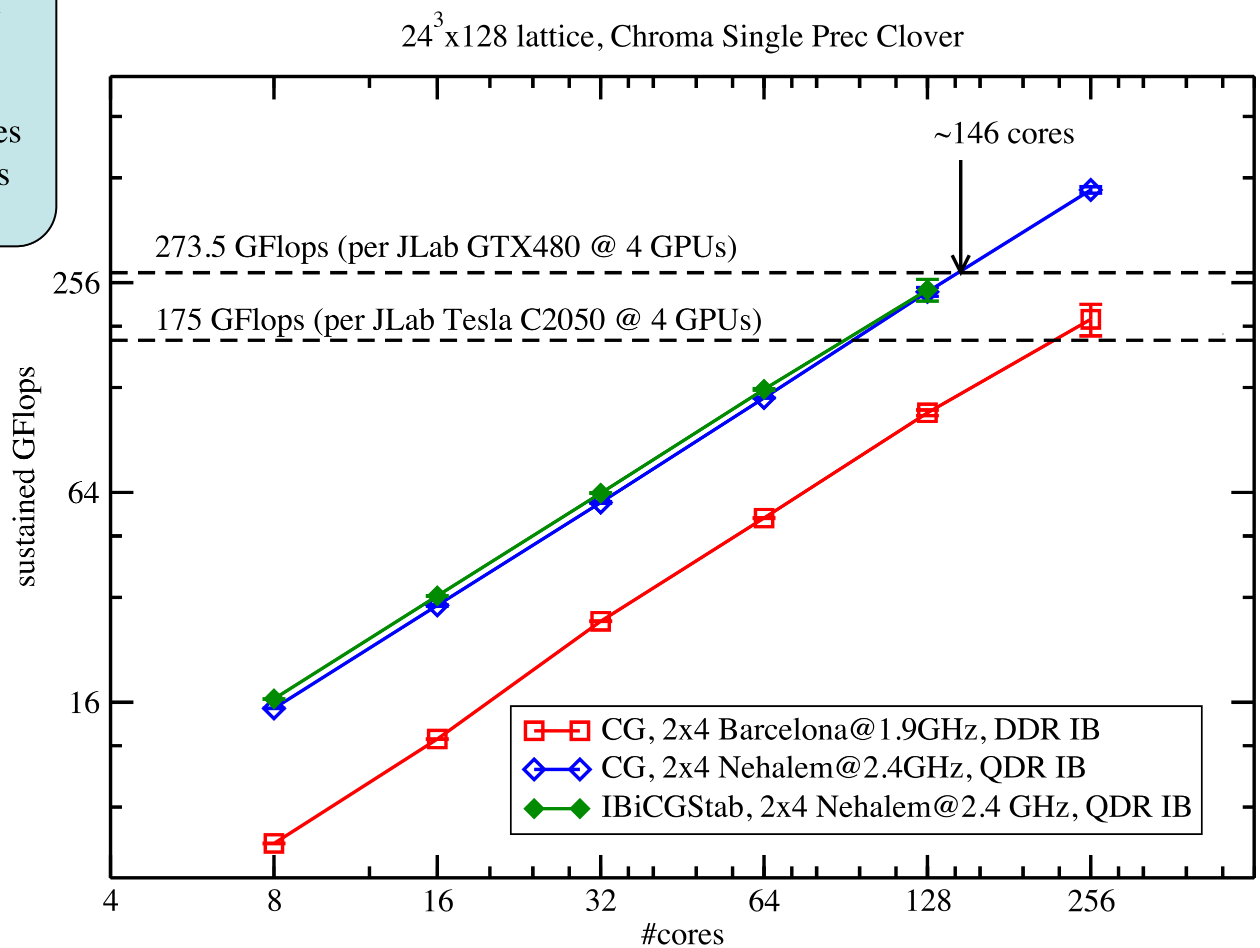




- Edge scales best for inversions
- But JLab nodes has more internal PCI Bandwidth
- Difference due to interconnect speed(?)
  - caveat funny 4 GPU PCI behavior
- Teslas 'catch up' to 480s
  - Teslas DDR IB
  - GTX480-s SDR IB ?
- 8x GTX480 => 2 Tflops

Compare to Multi-Core cluster

Modulo David Bailey caveats regarding comparing apples with non-apples



Vector isoscalar (I=0) spectrum

light-strange (ls) basis

N_f=2+1, $m_{1/4}$ ~ 400MeV, L~2fm

$$\mathcal{O}_l^{\Gamma} = \frac{1}{\sqrt{2}}\left(\bar{u}\boldsymbol{\Gamma}\mathbf{u} + \bar{\mathbf{d}}\boldsymbol{\Gamma}\mathbf{d}\right)$$

$$\mathcal{O}_s^{\Gamma} = \bar{s}\boldsymbol{\Gamma}\mathbf{s}$$

Figure courtesy of Robert Edwards

I = 0: Must include all disconnected diagrams

- ‘Distillation’ technique, 16^3x128 lattice
- N_{ev} x N_s x N_t x #quark x #cfg solves
    - N_{ev}=64, N_s=4, N_t=128, #quark=2, #cfg=479
    - 31 Million solves.
    - for 24^3x128, will need N_{ev}=128 - 162
    - Too expensive to do this without GPUs
- omega + 7 excited states, ~1% statistical error.

Vector isoscalar (I=0) spectrum

light-strange (ls) basis

$$\mathcal{O}_l^{\Gamma} = \frac{1}{\sqrt{2}}(\bar{u}\Gamma u + \bar{d}\Gamma d)$$

$$\mathcal{O}_s^{\Gamma} = \bar{s}\Gamma s$$

I = 0: Must include all disconnected diagrams

- 'Distillation' technique, 16^3x128 lattice
- N_{ev} x N_s x N_t x #quark x #cfg solves
  - N_{ev}=64, N_s=4, N_t=128, #quark=2, #cfg=479
  - 31 Million solves.
  - for 24^3x128, will need N_{ev}=128 - 162
  - Too expensive to do this without GPUs
- omega + 7 excited states, ~1% statistical error.

N_f=2+1, m_{π} ~ 400MeV, L~2fm

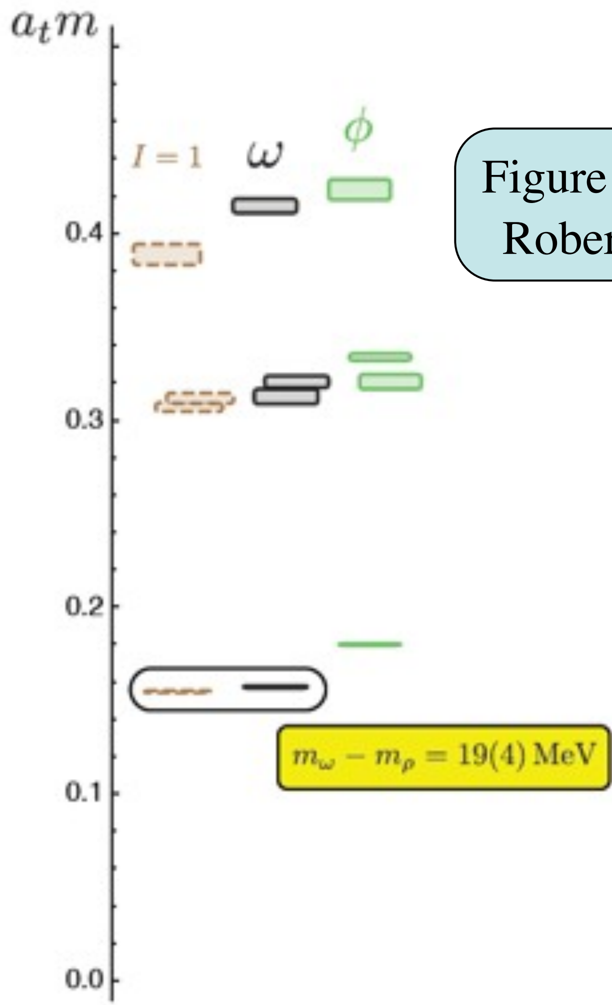


Figure courtesy of Robert Edwards

Future Work

- 1D decomposition: useful for long lattices (Anisotropic Clover: T ~ 5L to 8L)
  – but limiting for other actions where T ~ 2L
  – decompose in more dimensions
    - but need to improve strong scaling...
    - system improvements (direct GPU-GPU transfer over IB?)
- Algorithmic improvements:
  – decouple sub-blocks with domain-decomposed solvers?
  – multi-grid ?
- Gauge Generation using Multiple-GPUs
  – (Quick and Dirty) evolution vs. (Too Slow) revolution
  – GPU based capability machines may force timetable
- Analysis: Lots of correlation fn. contractions... should run well on GPUs
- Would like a high level, portable, standard and usable programing model (at least for non performance critical code) as we head towards the exascale

Summary

- Lattice QCD calculations require the solution of the Dirac Equation to compute the propagation of quarks
- Modern calculations need millions of solves for constructing correlation functions
- The QUDA library provides a fast solver
- The Chroma software system packages it up for use in calculations
- The combination of QUDA, Chroma + JLab ARRA Cluster has enabled hitherto prohibitively expensive calculations
- QUDA + Chroma are now enabling GPU based LQCD worldwide
- Lots of work out there:
  - Algorithms for gauge generation & capability GPU based machines, programming models for exascale
  - Correlation function contractions for LQCD Analysis

Acknowledgements

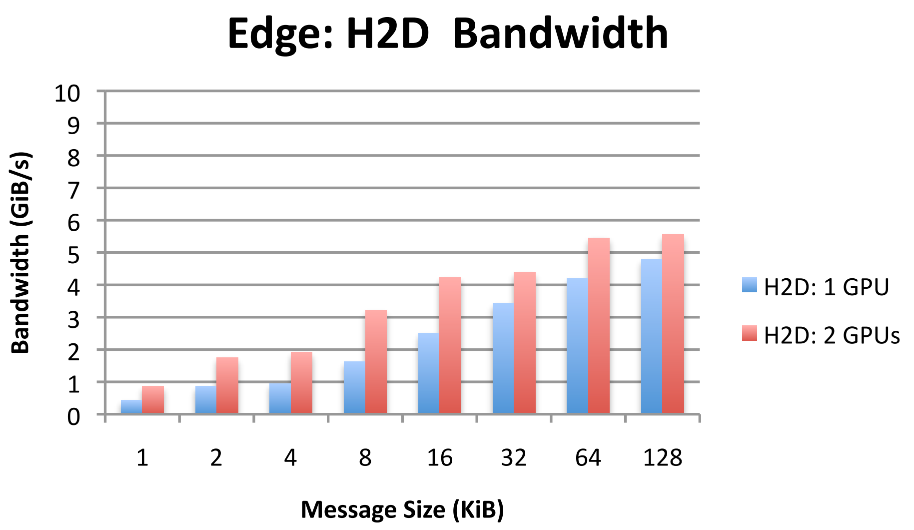
- B. Joó is funded through the USQCD SciDAC project grants: US D.O.E Grants DE-FC02- 06ER41440, DE-FC02-06ER41449 (SciDAC) and DE-AC05-060R23177 under which Jefferson Science Associates LLC manages and operates the Jefferson Lab.
- We thank Lawrence Livermore National Laboratory, NERSC and Jefferson Lab for time to carry out the benchmarking activities on their clusters.

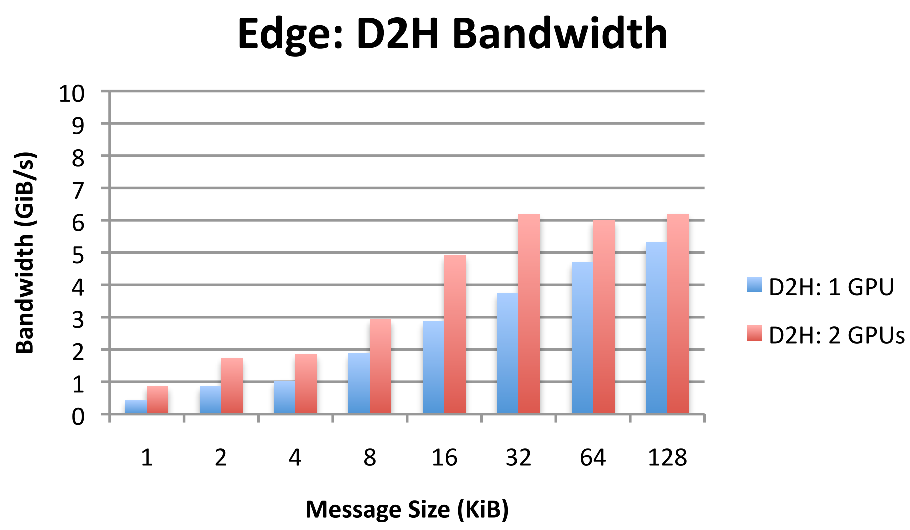
References

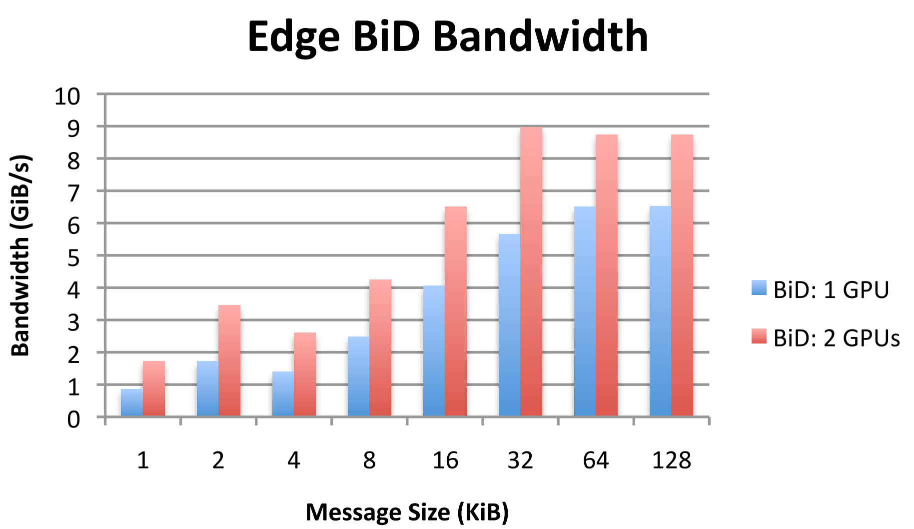
- **Lattice QCD as a video game.**
Gyozo I. Egri, (Eotvos U.) , Zoltan Fodor, (Eotvos U. & Wuppertal U. & UC, San Diego) , Christian Hoelbling, (Wuppertal U.) , Sandor D. Katz, (Eotvos U. & Wuppertal U.) , Daniel Nogradi, Kalman K. Szabo, (Wuppertal U.) . Nov 2006. 10pp. Comput.Phys.Commun.177:631-639,2007. e-Print: hep-lat/0611022
- **SciDAC-2 Software Infrastructure for Lattice QCD**. Balint Joo, In Journal of Physics: Conference Series 78 (2007).
- **The Chroma software system for lattice QCD.**
By SciDAC Collaboration and LHPC Collaboration and UKQCD Collaboration (Robert G. Edwards et al.). JLAB-THY-04-54, Sep 2004. 3pp. Presented at 22nd International Symposium on Lattice Field Theory (Lattice 2004), Batavia, Illinois, 21-26 Jun 2004. Published in Nucl.Phys.Proc.Suppl.140:832,2005. Also in *Batavia 2004, Lattice field theory* 832-834, e-Print: hep-lat/0409003
- **Blasting through lattice calculations using CUDA.**
Kipton Barros, Ronald Babich, Richard Brower, (Boston U.) , Michael A. Clark, (Boston U., Ctr. Comp. Sci.) , Claudio Rebbi, (Boston U.) . Oct 2008. 7pp. Presented at 26th International Symposium on Lattice Field Theory (Lattice 2008), Williamsburg, Virginia, 14-20 Jul 2008. Published in PoS LATTICE2008:045,2008. e-Print: arXiv:0810.5365 [hep-lat]
- **Solving Lattice QCD systems of equations using mixed precision solvers on GPUs.**
M.A. Clark, (Harvard-Smithsonian Ctr. Astrophys. & Harvard U.) , R. Babich, (Boston U., Ctr. Comp. Sci. & Boston U.) , K. Barros, (Northwestern U.) , R.C. Brower, C. Rebbi, (Boston U., Ctr. Comp. Sci. & Boston U.) . Nov 2009. 30pp. Published in Comput.Phys.Commun.181:1517-1528,2010. e-Print: arXiv:0911.3191 [hep-lat]
- **Parallelizing the QUDA Library for Multi-GPU Calculations in Lattice Quantum Chromodynamics.**
Ronald Babich, Michael A. Clark, Balint Joo, . JLAB-IT-10-01, Nov 2010. 11pp. Proceedings of ACM / IEEE SC2010 - International Conference for High Performance Computing, Networking, Storage and Analysis held November 13-19, 2010 in New Orleans, Louisiana., e-Print: arXiv:1011.0024 [hep-lat]

Extra Slides

Edge: Bandwidths

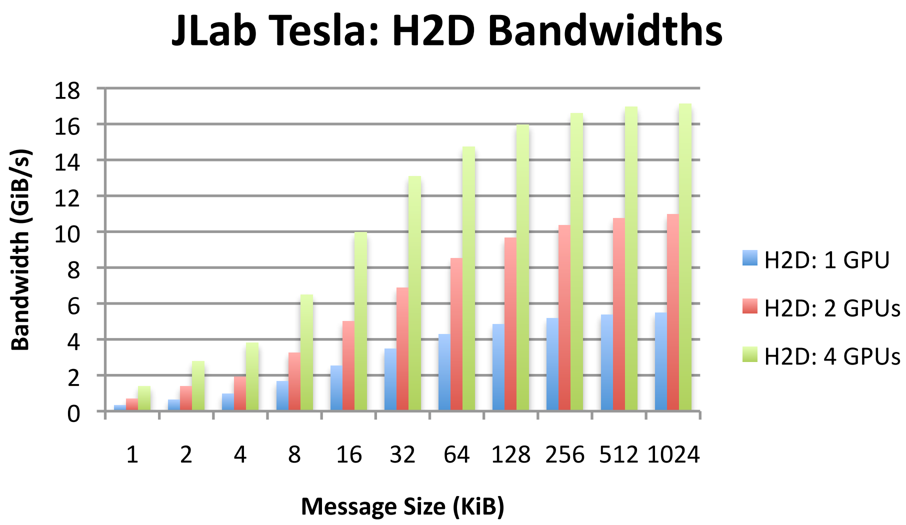


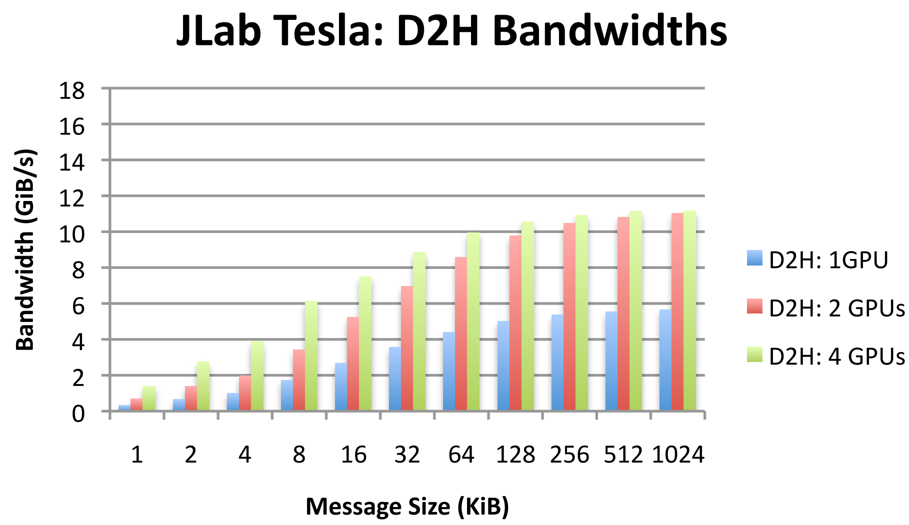


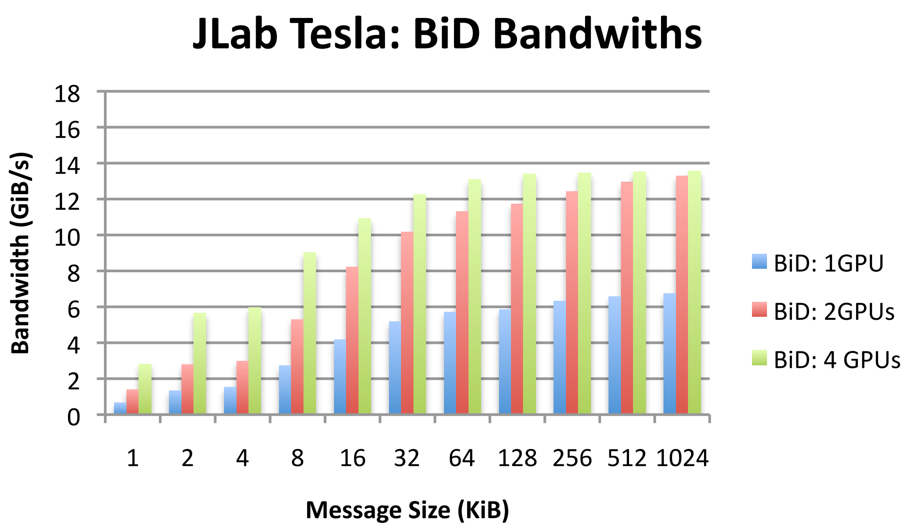


- Single PCIe x16
- 2 GPUs cannot draw much more B/W than 1 GPU for large messages
- ~1.4x for BiD

JLab Tesla: Bandwidths

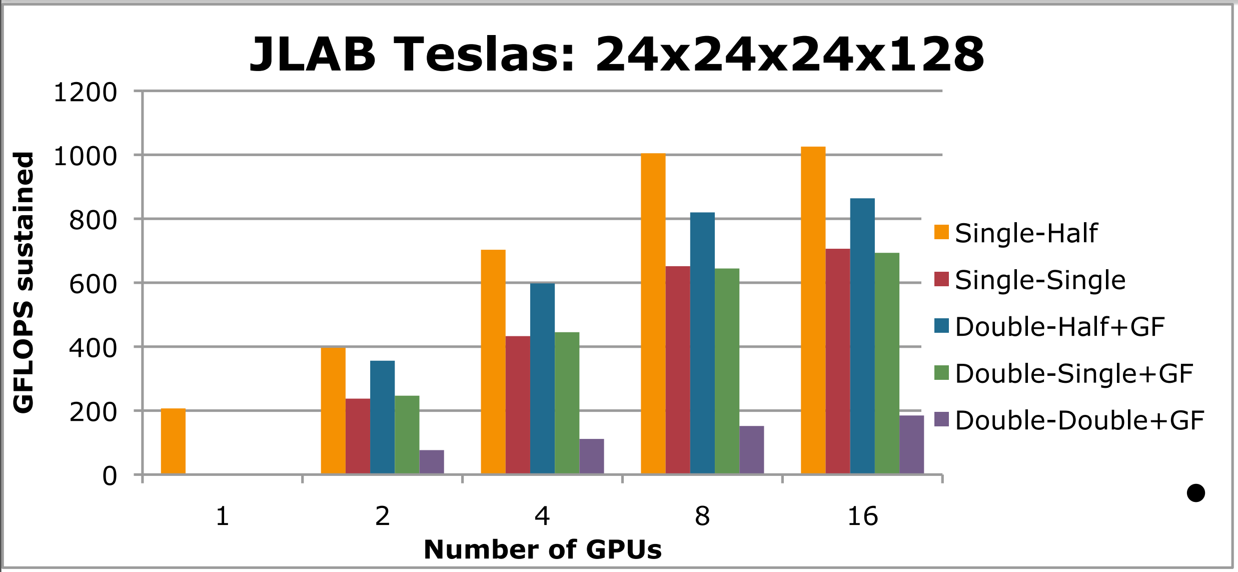


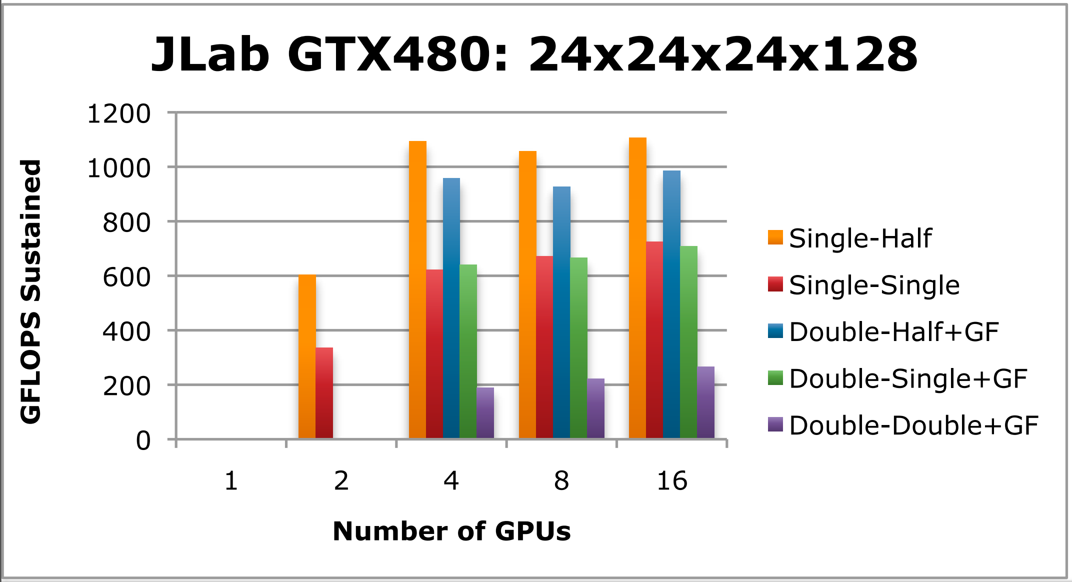




- 4 GPU D2H B/W really low (same as 2 GPU)
- 2 GPU B/W ~ 2x 1 GPU

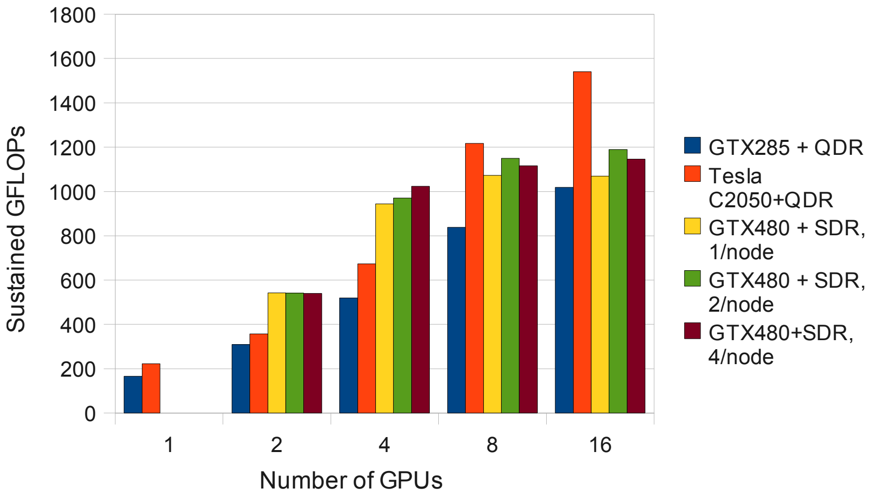
JLab Inverter Performance

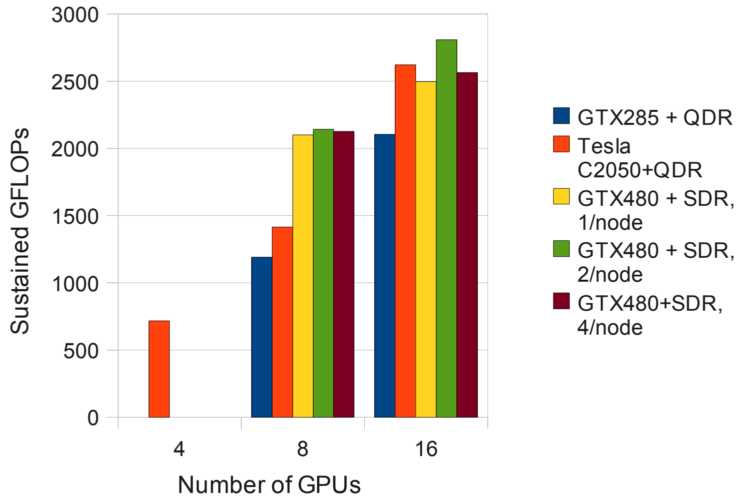




- Single Half levels off at 8 GPUs (c.f. 16 on Edge) for Tesla
- Single Half levels off at 4 GPUs on GTX480
- 4 GTX480 ~ 8 Tesla (for Single Half)

NERSC Dirac Inverter Performance





- CAVEAT: This is older data (last summer) - previous version of QUDA.
- JLab GTX480 1/2/4 per node results come from running only 1, 2 or 4 GPUs in 4-GPU system, leaving others idle:
  - 16 GPU @ 1 per node = 16 nodes (48 idle GPUs)