# LATTICE QCD ON MODERN GPU SYSTEMS

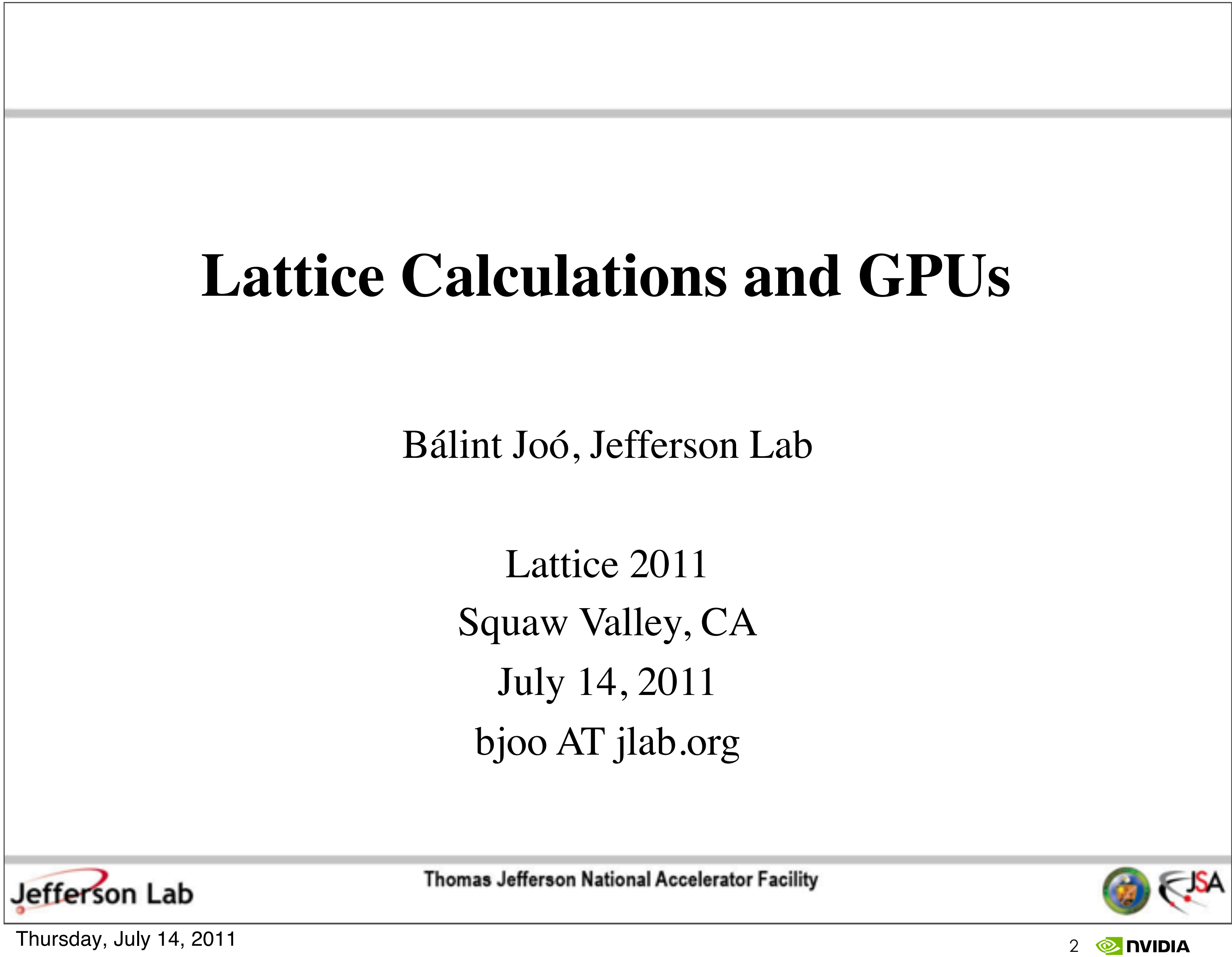
Kate Clark, Mathias Wagner

# LOOKING BACK

## QCD was an early adaptor

First use of GPUs for LQCD already over 10 years ago:

*“Lattice QCD as a video game”*, Egri et. al (2006)



**Lattice Calculations and GPUs**

Bálint Joó, Jefferson Lab

Lattice 2011

Squaw Valley, CA

July 14, 2011

bjoo AT jlab.org

Jefferson Lab

Thomas Jefferson National Accelerator Facility

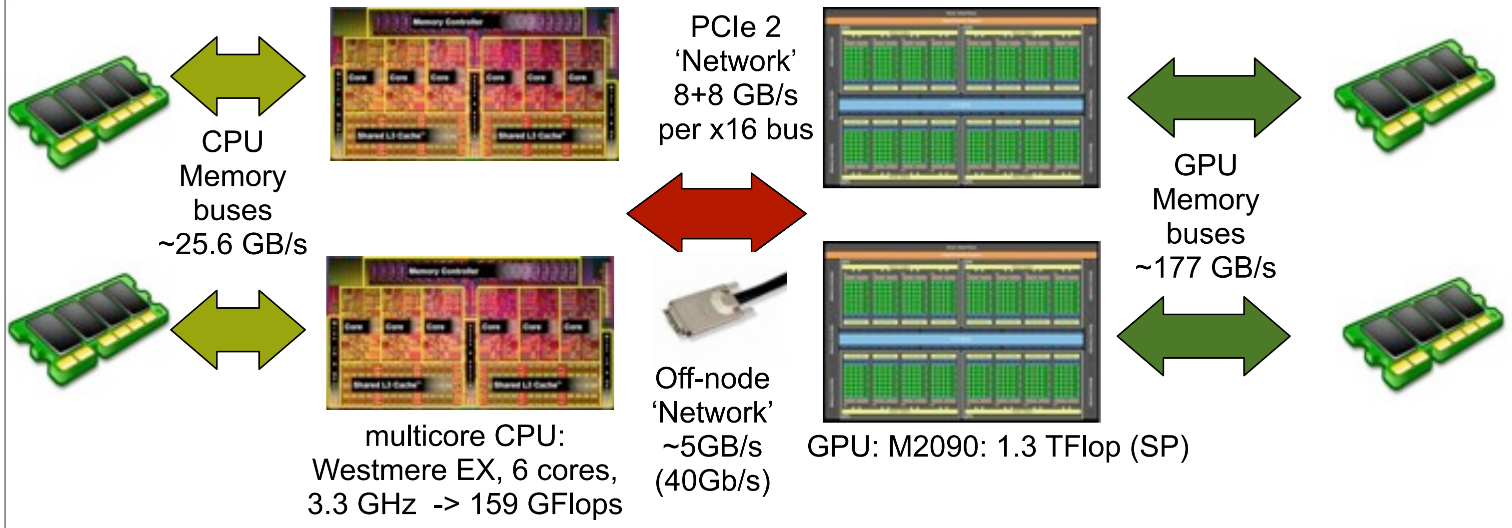
JSA

Thursday, July 14, 2011

# LOOKING BACK

## Lattice 2011

### What does a full system look like?




- GPU Mem. B/W / CPU Mem. B/W ~6.9x
- GPU Peak Flops (SP) / CPU Peak Flops(SP) ~ 8.4x
- PCIe Gen2 serious bottleneck for multi-GPU

*NB: 'Speeds and Feeds' come from comparing a 6 core Westmere EX, running at 3.33 GHz, with a Tesla M2090 - using respective datasheets.*

*JLab 10G cluster*

Jefferson Lab — Thomas Jefferson National Accelerator Facility

Thursday, July 14, 2011

# LOOKING BACK

## Bielefeld GPU Cluster 2012

Got me started on LQCD and GPUs …

… and out of academia (eventually)

… but not out of LQCD and GPUs

# AND NOW ...

## TESLA V100 32GB

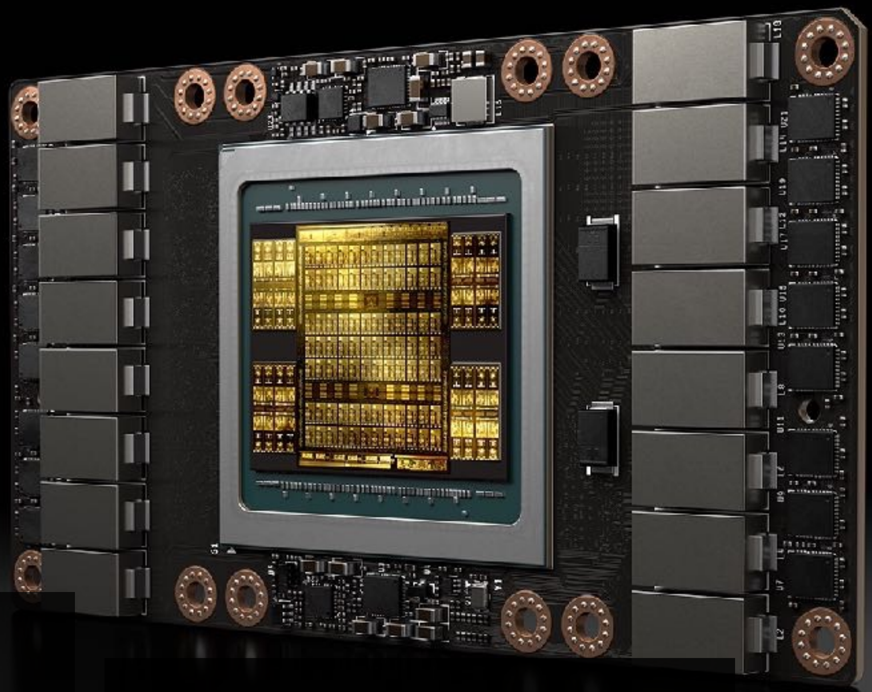
5,120 CUDA cores
**640 NEW** Tensor cores
7.8 FP64 TFLOPS | 15.7 FP32 TFLOPS | 125 Tensor TFLOPS
20MB SM RF | 16MB Cache
**32GB** HBM2 @ 900GB/s | 300GB/s NVLink

# NVIDIA POWERS WORLD'S FASTEST SUPERCOMPUTER

Summit Becomes First System To Scale The 100 Petaflops Milestone


**122 PF**
HPC

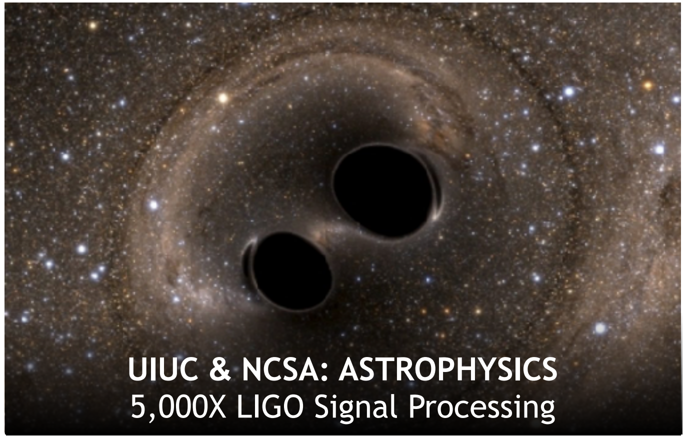
**3 EF**
AI



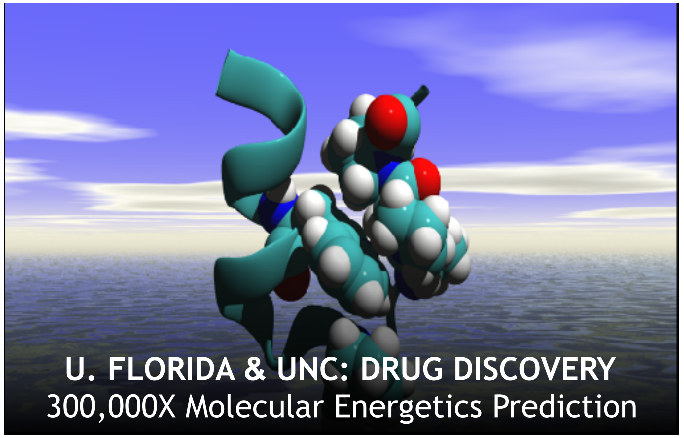
**27,648**
Volta Tensor Core GPUs

# DEEP LEARNING COMES TO HPC

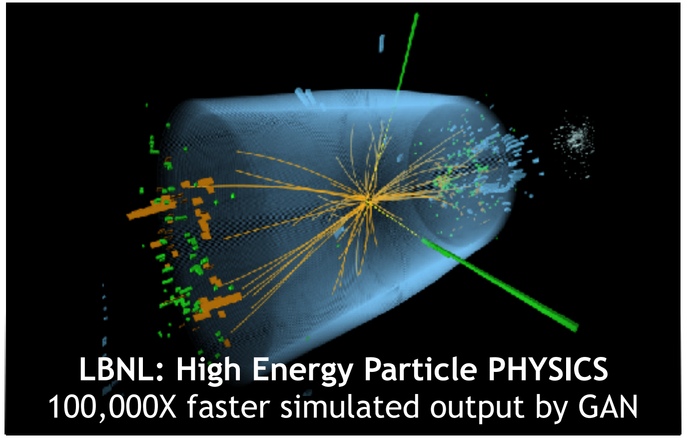
## Accelerates Scientific Discovery



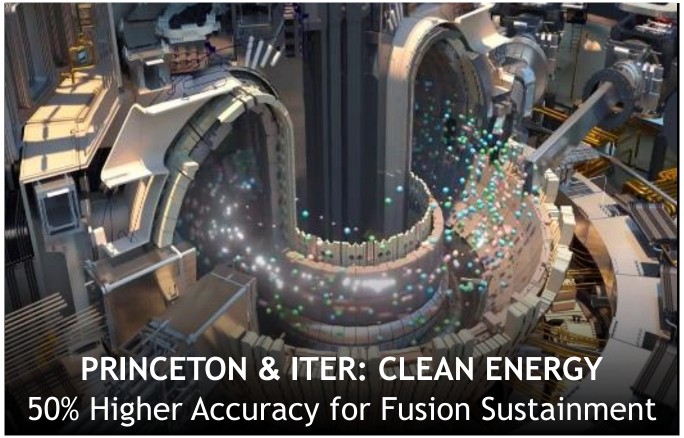
**UIUC & NCSA: ASTROPHYSICS**
5,000X LIGO Signal Processing



**U. FLORIDA & UNC: DRUG DISCOVERY**
300,000X Molecular Energetics Prediction

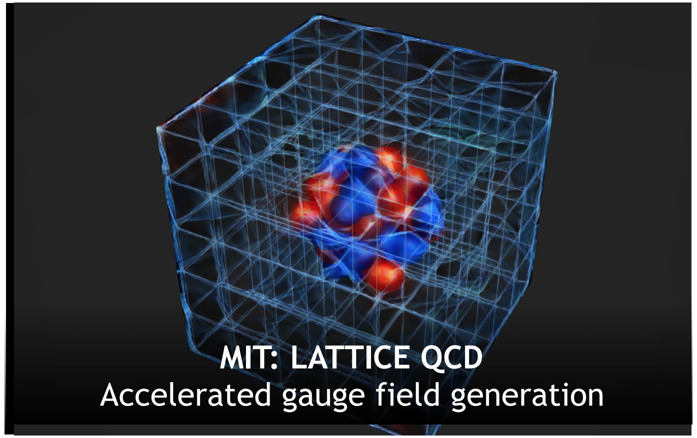


**LBNL: High Energy Particle PHYSICS**
100,000X faster simulated output by GAN



**PRINCETON & ITER: CLEAN ENERGY**
50% Higher Accuracy for Fusion Sustainment

**U.S. DoE: PARTICLE PHYSICS**
33% More Accurate Neutrino Detection



**MIT: LATTICE QCD**
Accelerated gauge field generation

DOMAIN EXPERTISE

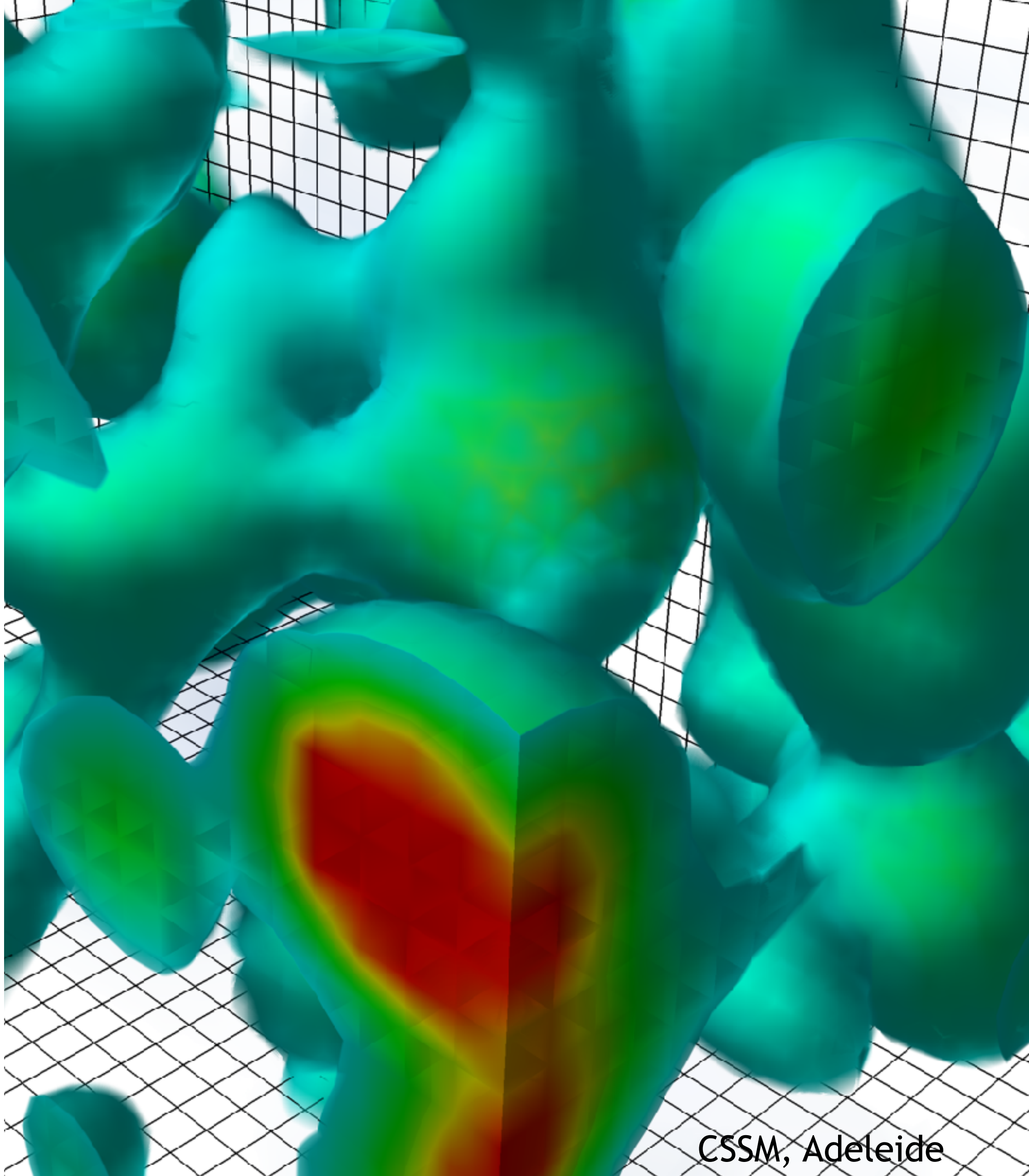
\+ VISUALIZATION TECH

\+ ART

\--------------------------------

SCIENCE OUTREACH

Talk to us about collaborations!



# PROGRAMMABILITY

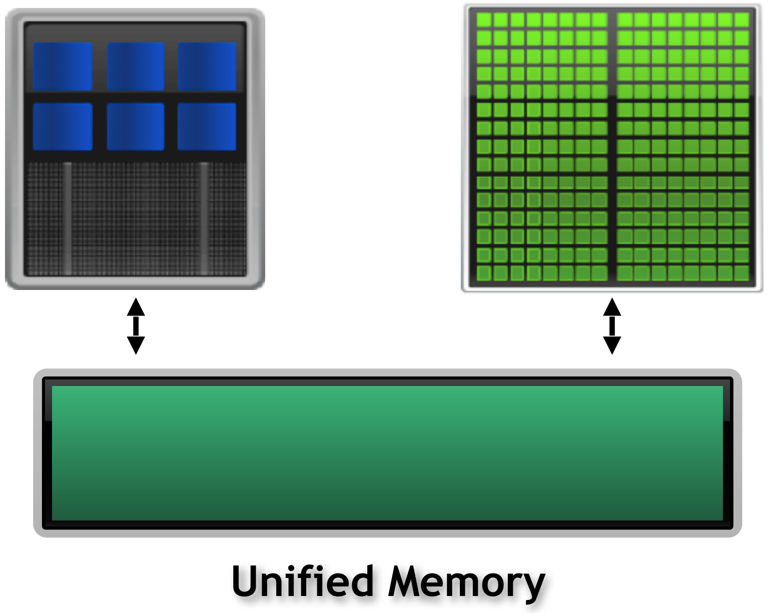
## MULTIPLE OPTIONS

**Libraries** — AmgX, cuDNN, cuBLAS, OpenCV, Thrust

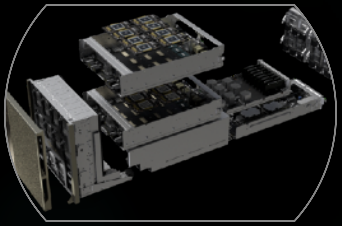
**Compiler Directives** — OpenACC

**Programming Languages** — C/C++, Fortran, python, Java

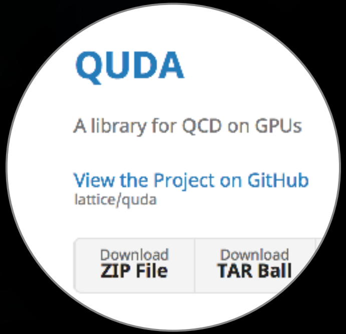
### Automatic data migration



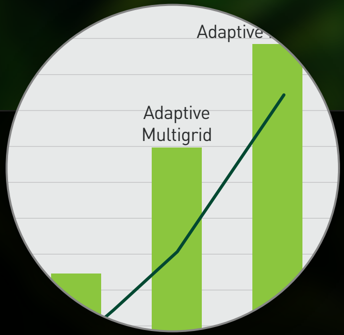
# $S^4$



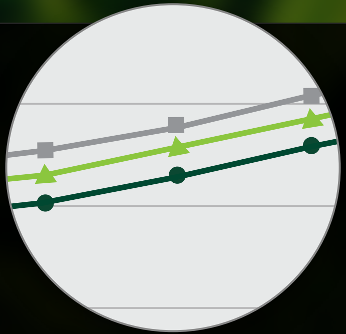
SYSTEMS




SOFTWARE




SPEEDUP



SCALING

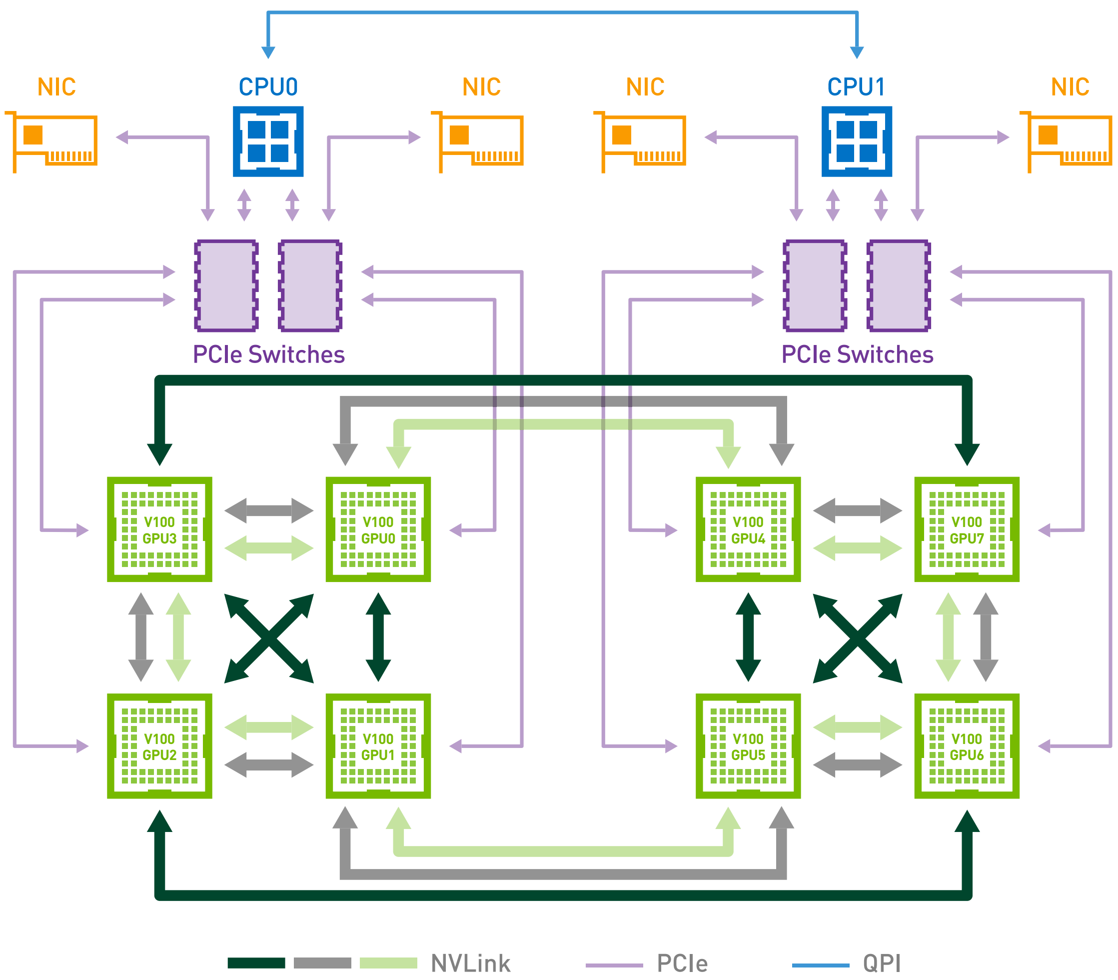
# MODERN GPU SYSTEMS

# NVIDIA DGX-1

8 V100 GPUs (16/32 GB)

Hypercube-Mesh NVLink

4 EDR IB

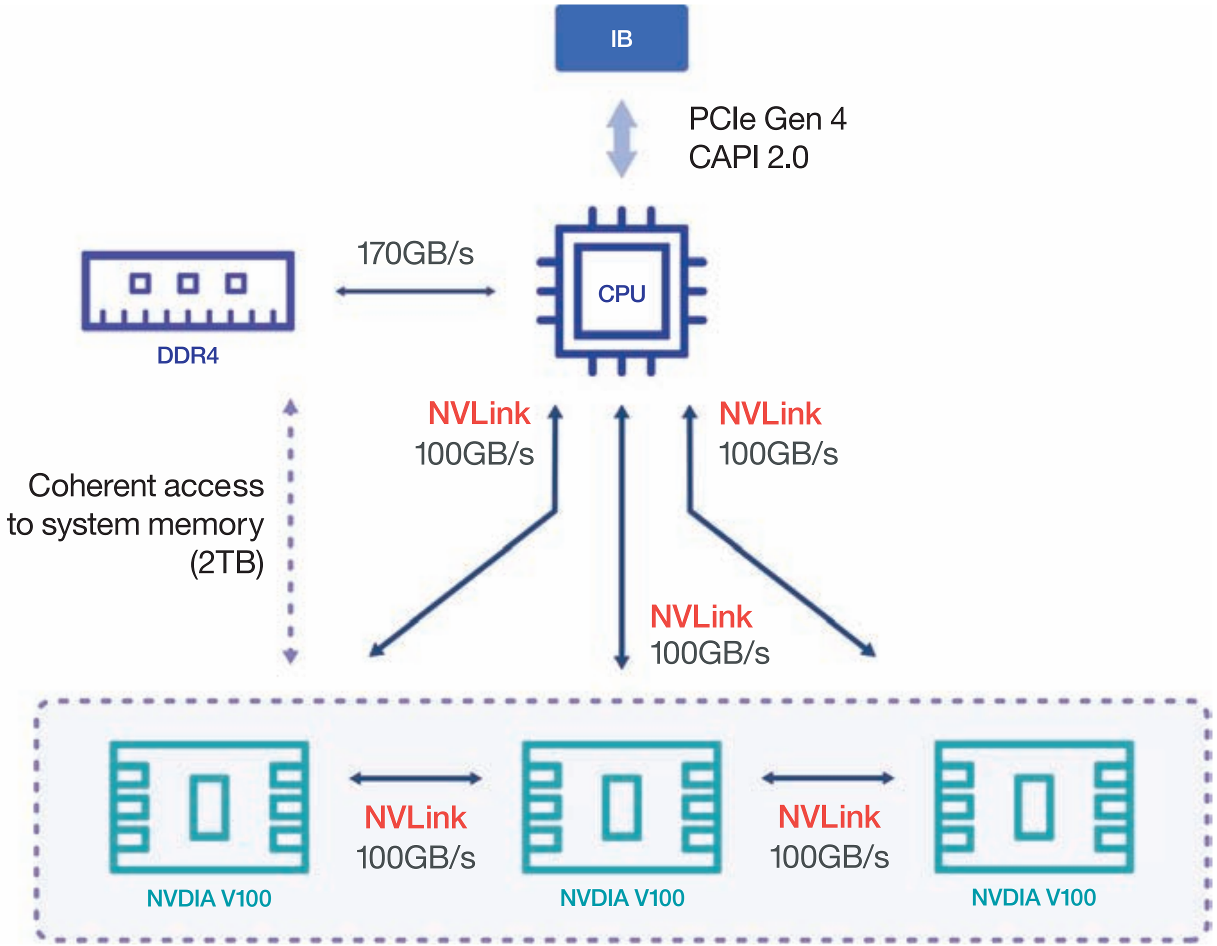


# IBM AC 922

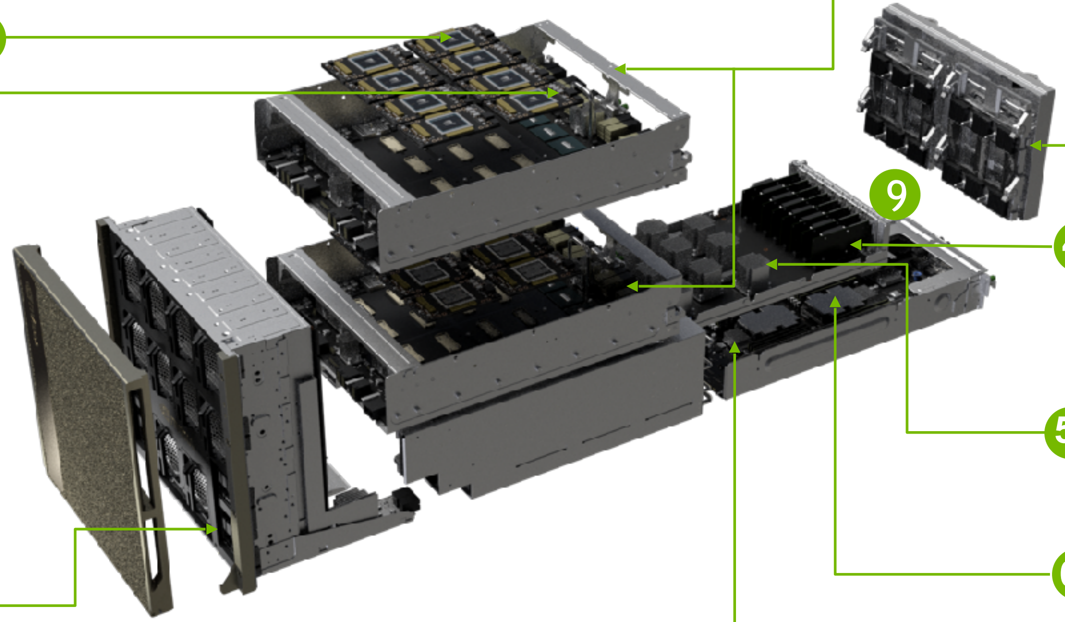
4/6 V100 GPUs

NVLink to GPU and P9

2 EDR IB



# NVIDIA DGX-2



NVIDIA Tesla V100 32GB 1

**2** **2 GPU Boards**
8 V100 32GB GPUs per board
6 NVSwitches per board
512GB Total HBM2 Memory
interconnected by
**Plane Card**

**12 NVSwitches** **3**
2.4 TB/sec bi-section
bandwidth

9

**4** **8 EDR Infiniband/100 GigE**
1600 Gb/sec Total
Bi-directional Bandwidth

**5** **PCIe Switch Complex**

**6** **2 Intel Xeon Platinum CPUs**

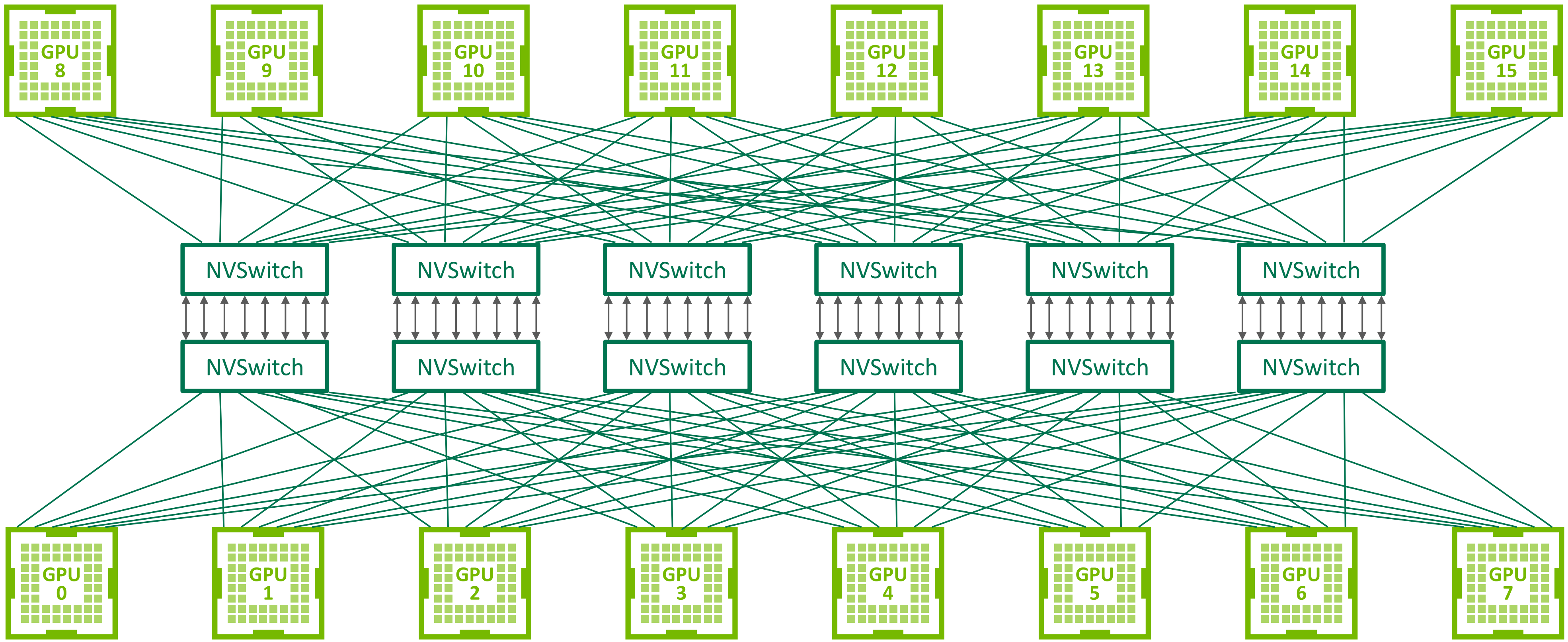
**7** **1.5 TB System Memory**

**30 TB NVME SSDs** **8**
Internal Storage

**Dual 10/25 Gb/sec** **9**
**Ethernet**

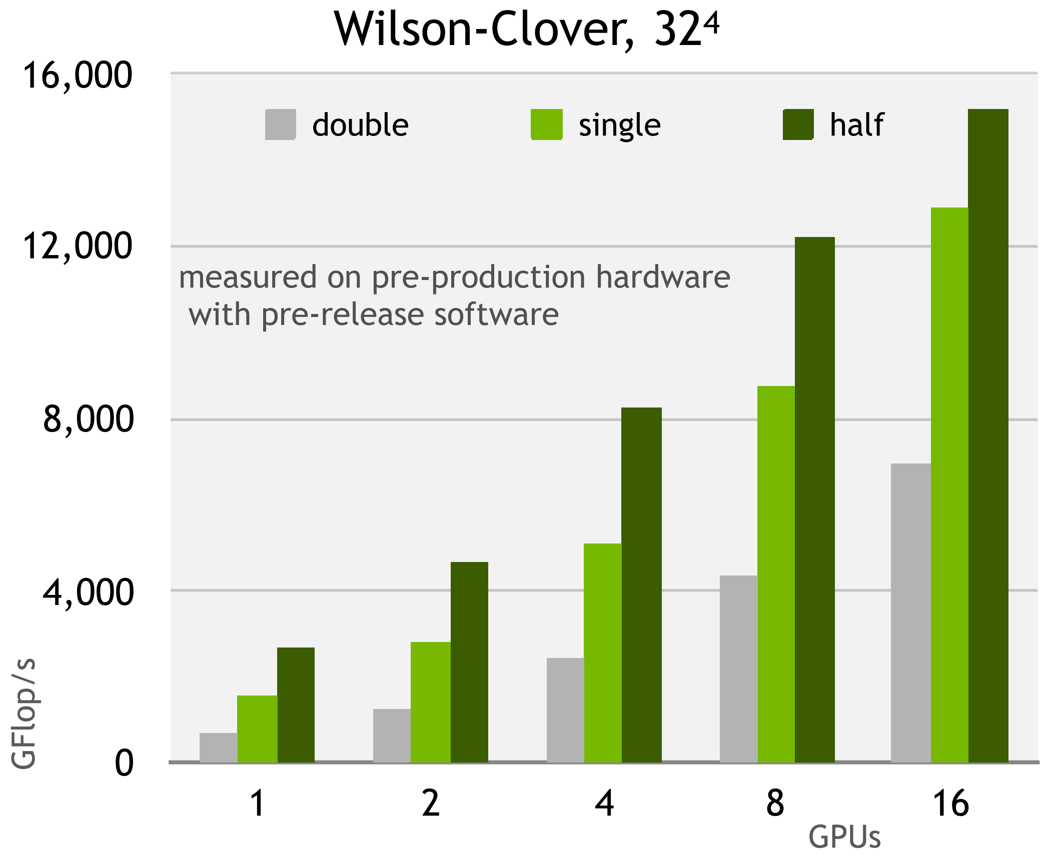
# DGX-2: FULL NON-BLOCKING BANDWIDTH

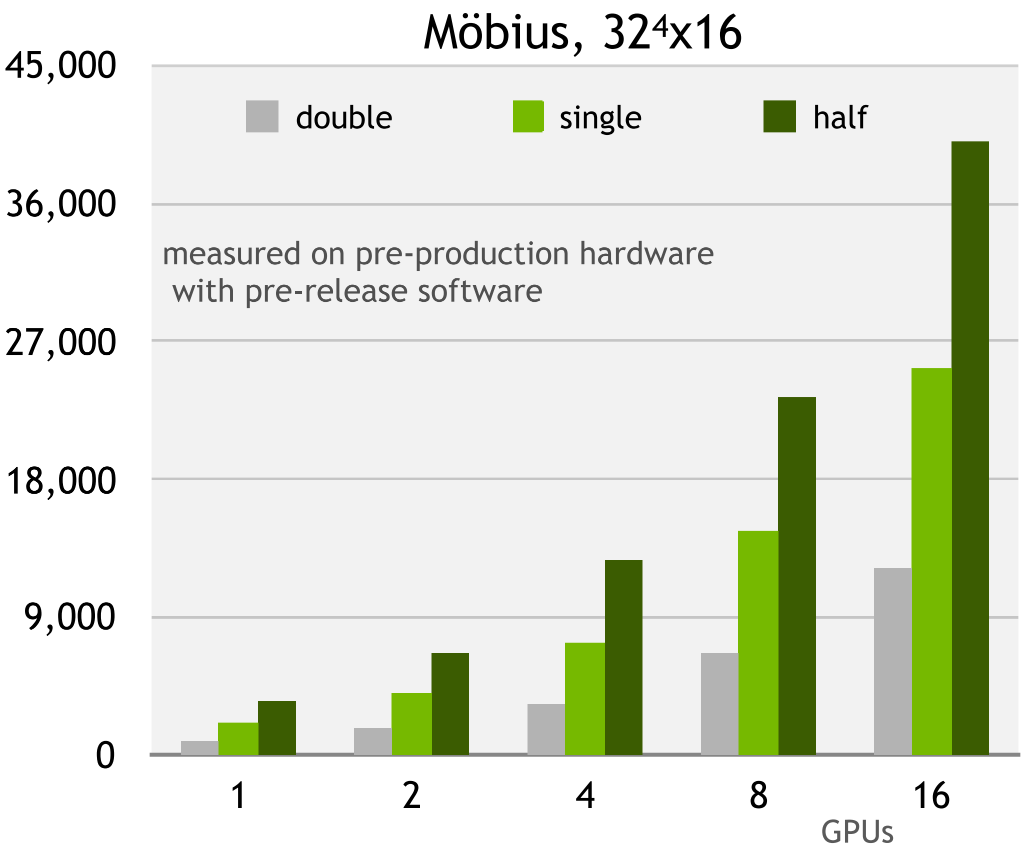
2.4 TB/s bisection bandwidth



# NVIDIA DGX-2

## QUDA Dslash strong scaling

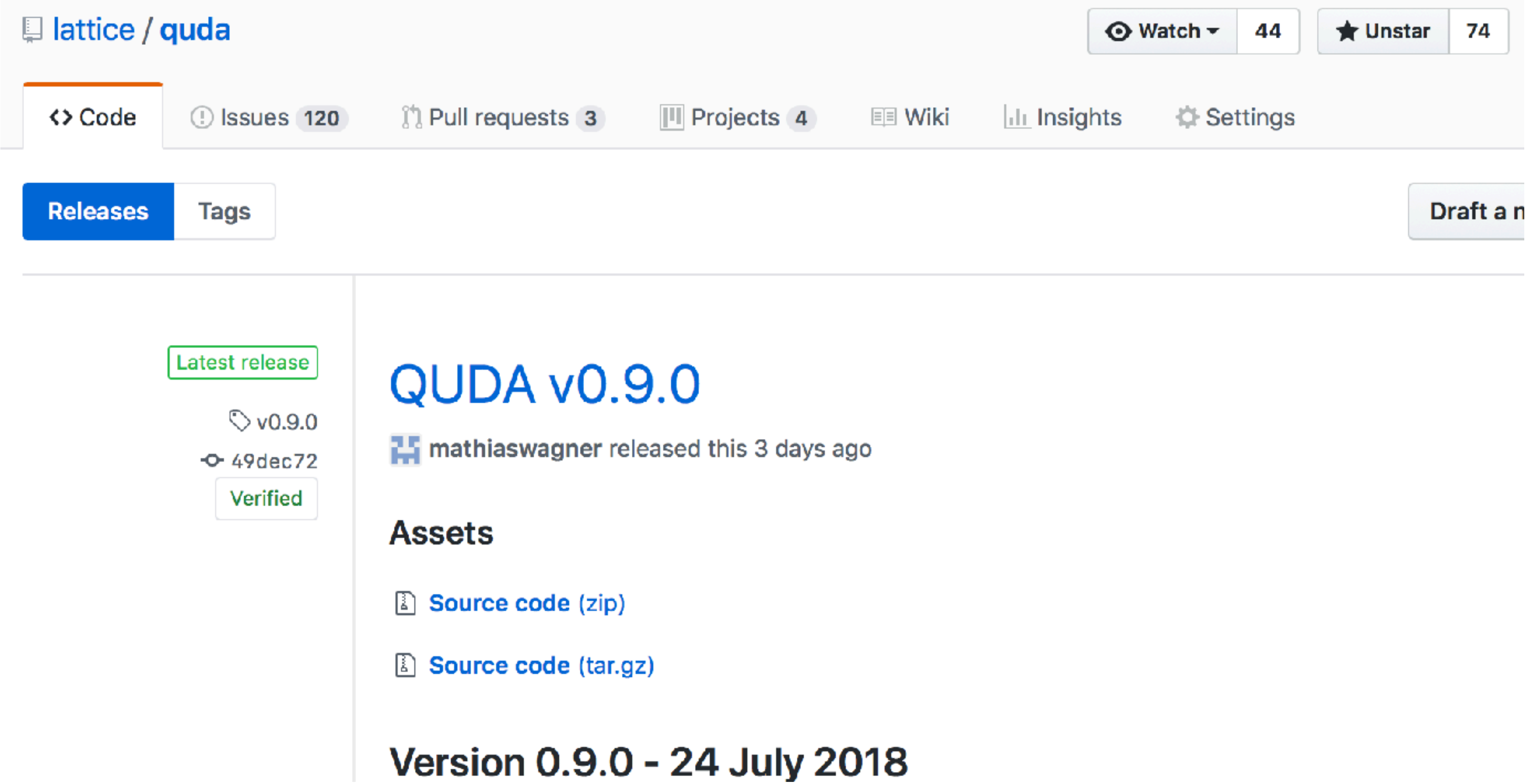


# SOFTWARE

# QUDA - LATTICE QCD ON GPUS

http://lattice.github.com/quda, BSD license



# QUDA CONTRIBUTORS

## 10 years - lots of contributors

Ron Babich (NVIDIA)
Simone Bacchio (Cyprus)
Michael Baldhauf (Regensburg)
Kip Barros (LANL)
Rich Brower (Boston University)
Nuno Cardoso (NCSA)
Kate Clark (NVIDIA)
Michael Cheng (Boston University)
Carleton DeTar (Utah University)
Justin Foley (Utah -> NIH)
Joel Giedt (Rensselaer Polytechnic Institute)
Arjun Gambhir (William and Mary)
Steve Gottlieb (Indiana University)
Kyriakos Hadjiyiannakou (Cyprus)

Dean Howarth (BU)
Bálint Joó (Jlab)
Hyung-Jin Kim (BNL -> Samsung)
Bartek Kostrzewa (Bonn)
Claudio Rebbi (Boston University)
Hauke Sandmeyer (Bielefeld)
Guochun Shi (NCSA -> Google)
Mario Schröck (INFN)
Alexei Strelchenko (FNAL)
Jiqun Tu (Columbia)
Alejandro Vaquero (Utah University)
Mathias Wagner (NVIDIA)
Evan Weinberg (NVIDIA)
Frank Winter (Jlab)

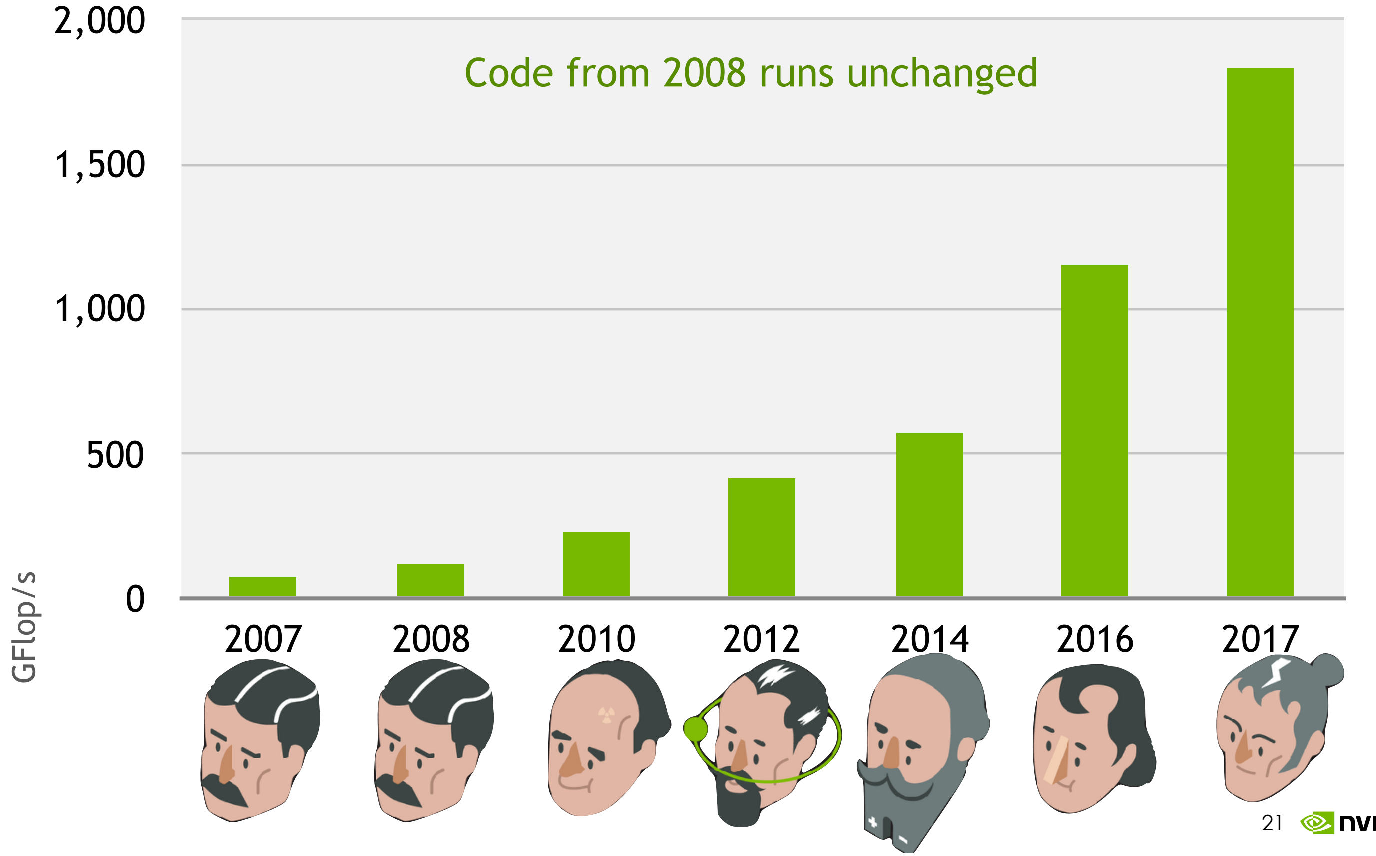
# TEN YEARS OF QUDA

## in use as GPU backend for BQCD, Chroma, CPS, MILC, TIFR, etc.

- Solvers for all major fermionic discretizations
- Routines needed for gauge-field generation
- Maximize performance
    - Exploit symmetries to minimize memory traffic
    - Mixed-precision methods (16 bit / 8 bit)
    - Domain-decomposed (Schwarz) preconditioners for strong scaling
    - Eigenvector and deflated solvers (Lanczos, EigCG, GMRES-DR)
    - Multi-source solvers
    - Multigrid solvers for optimal convergence


# RECOMPILE AND RUN

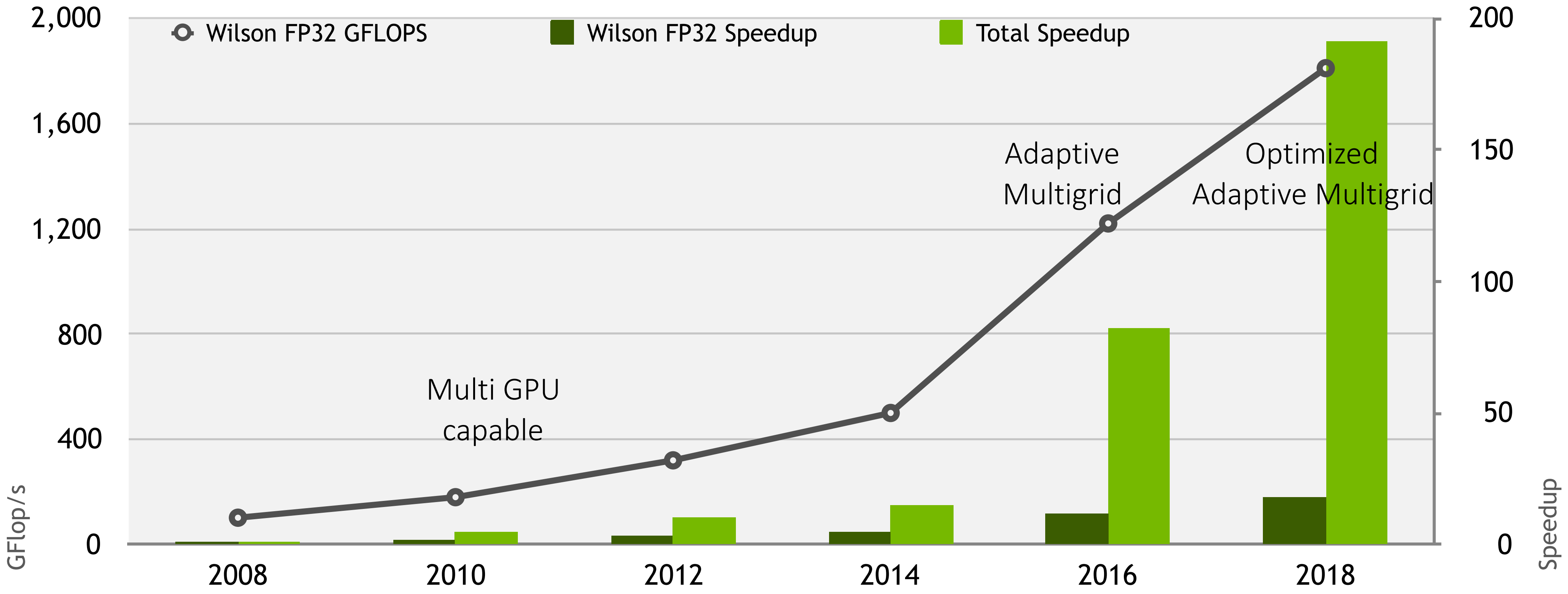
## Autotuning provides performance portability



# SPEEDUP

# NODE PERFORMANCE OVER TIME

## Multiplicative speedup through software and hardware




Time to solution is measured time to solution for solving the Wilson operator against a random source on a 24x24x24x64 lattice, β=5.5, $M_{\pi}$= 416 MeV. One node is defined to be 3 GPUs

# CHROMA HMC MULTIGRID

HMC typically dominated by solving the Dirac equation, but

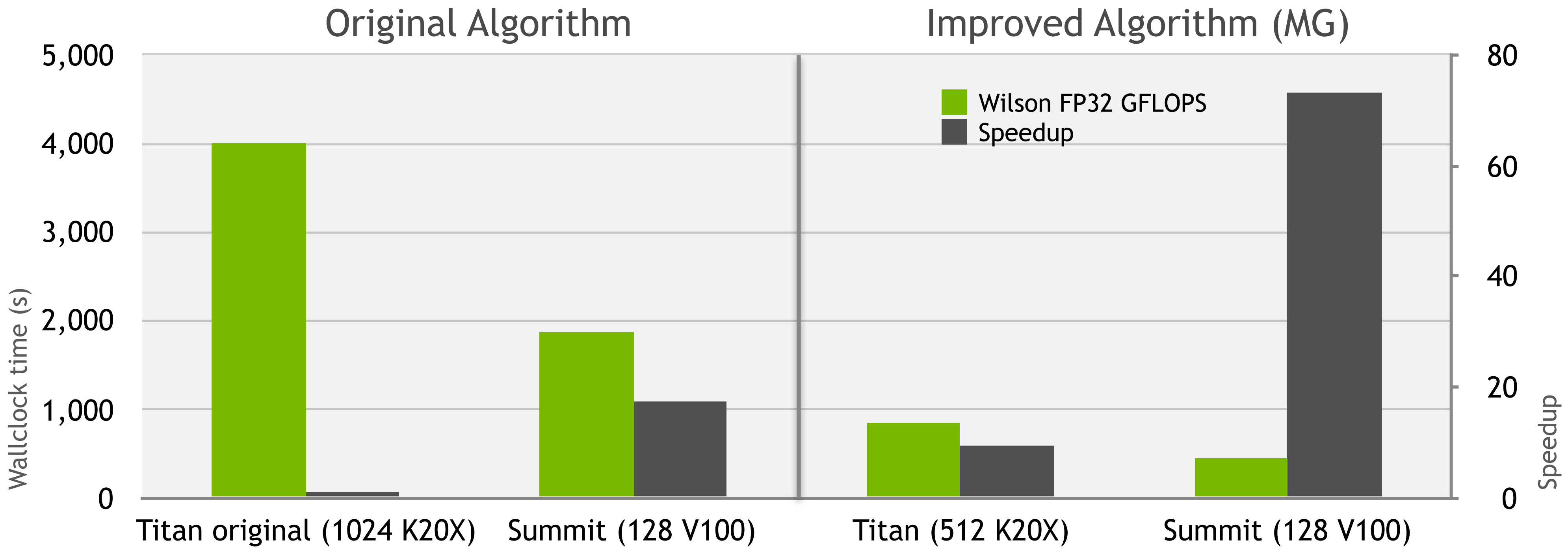
- Few solves per linear system
- Can be bound by heavy solves (c.f. Hasenbusch mass preconditioning)

Multigrid setup must run at speed of light

- Reuse and evolve multigrid setup where possible
- Use the same null space for all masses (setup run on lightest mass)
- Evolve null space vectors as the gauge field evolves (Lüscher 2007)
- Update null space when the preconditioner degrades too much on lightest mass

# MULTI-GRID ON SUMMIT

## Full Chroma Hybrid Monte Carlo




Data from B. Joo (Jefferson Lab). Chroma w/ QDP-JIT (F. Winter, Jefferson Lab) and QUDA.

B. Joo gratefully acknowledges funding through the US DOE SciDAC program (DE-AC05-06OR23177)

# HPC BEYOND MOORE'S LAW

## going wide

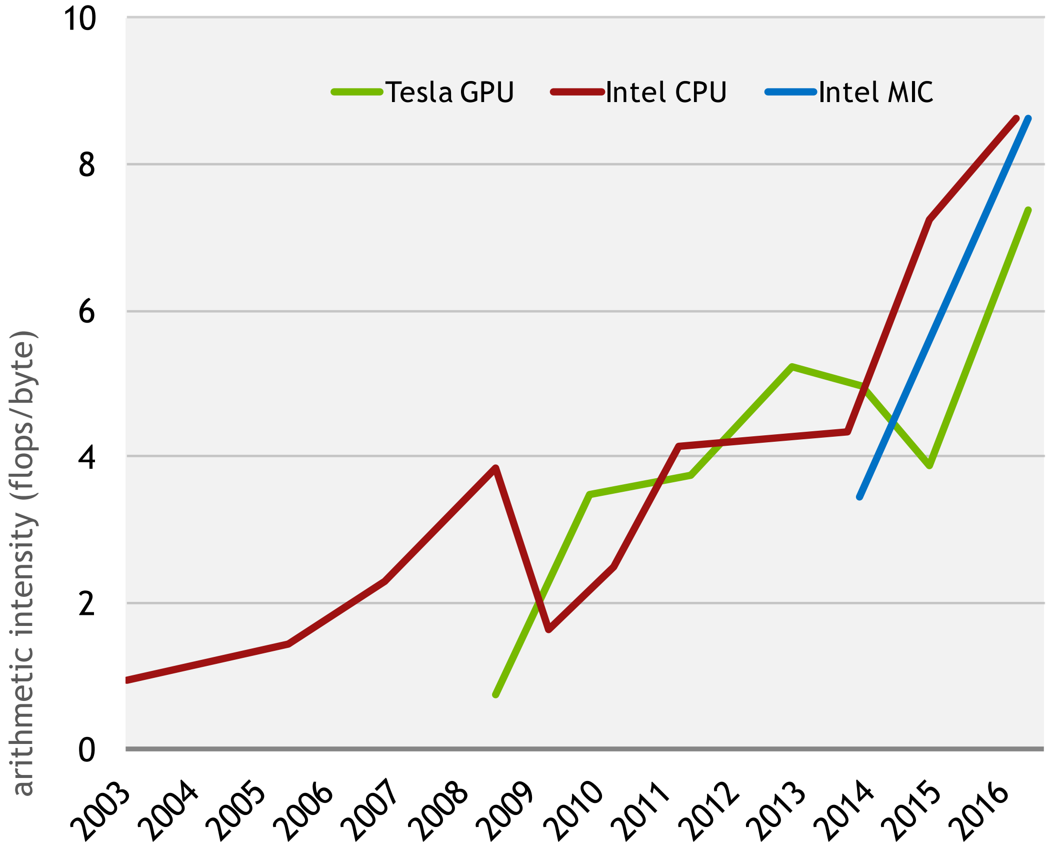
CPUs and GPUs becoming wider

increase in flops is driven by more cores

also applies to CPUs (server to mobile)

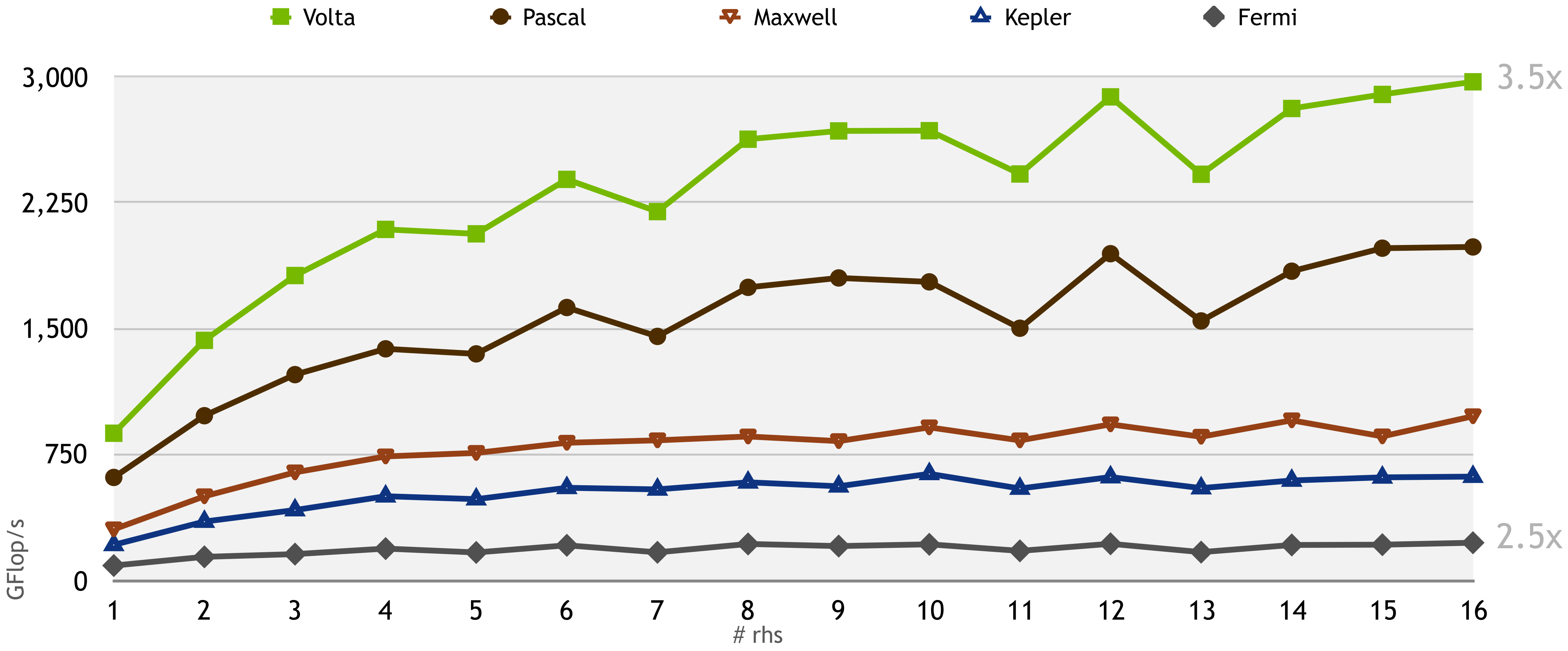
need sufficient amount of parallelism to fill architectures

need to be able to feed the cores



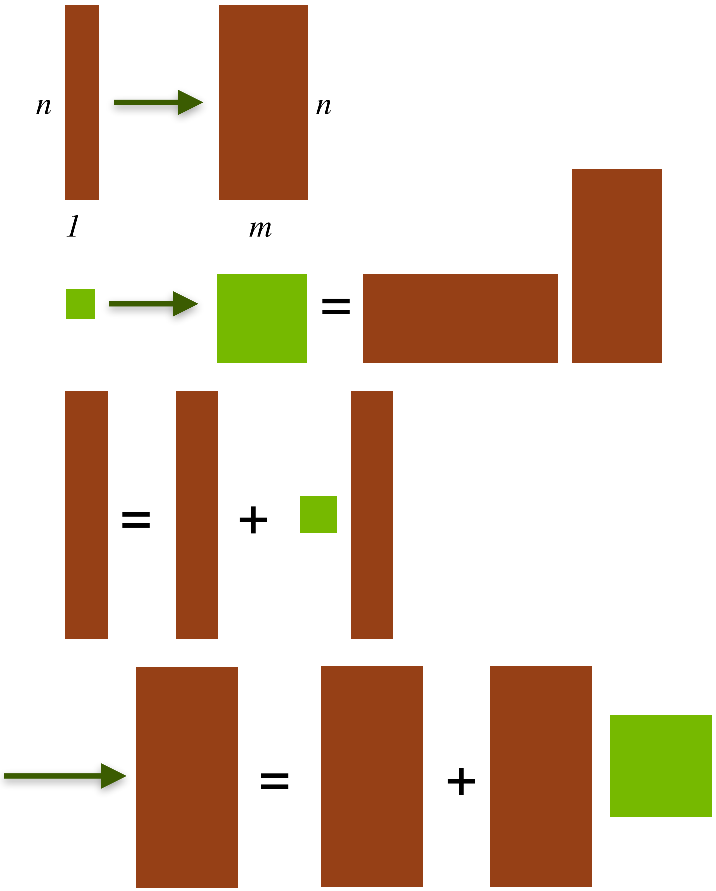
# MULTIPLE RIGHT-HAND SIDES

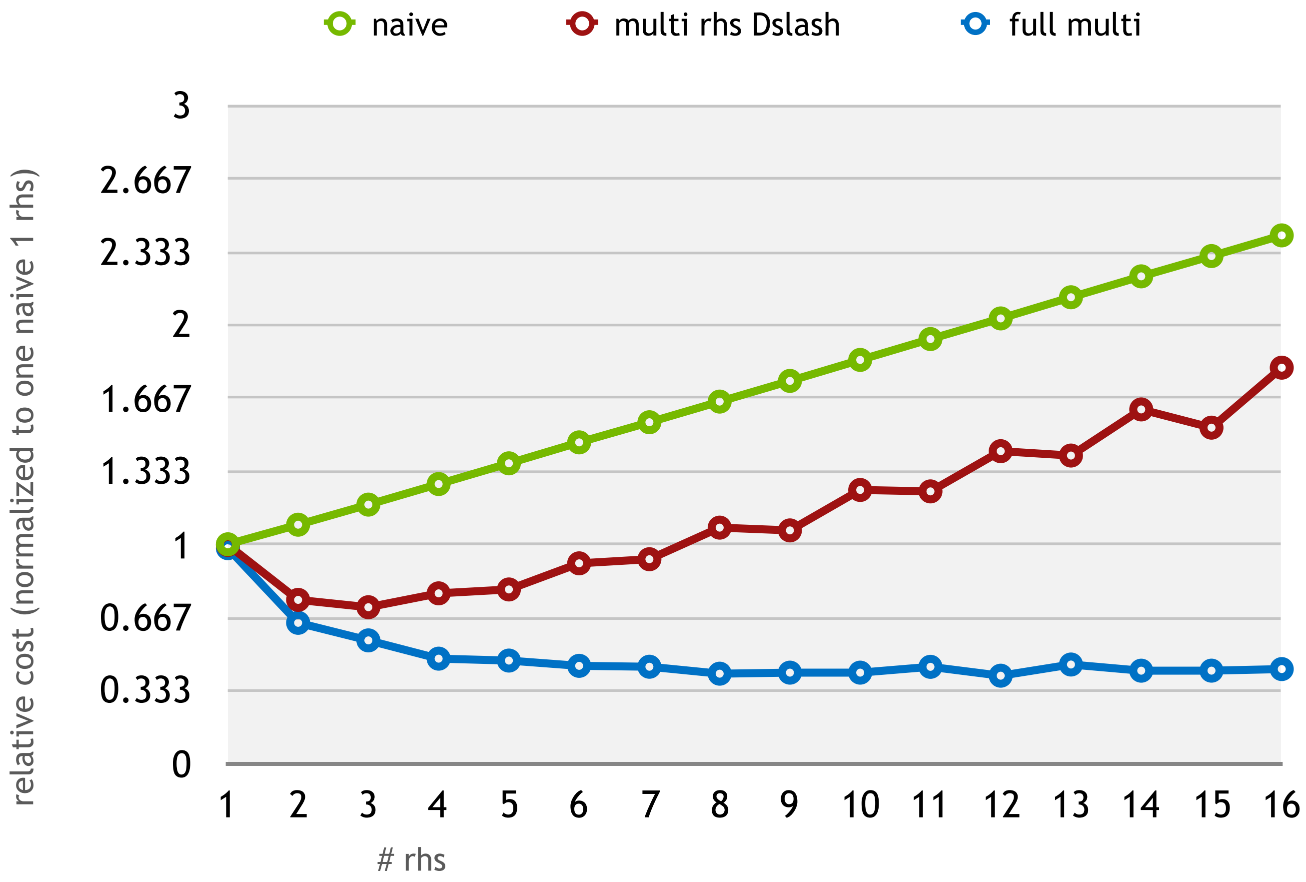
## $48^3$x12, HISQ, single precision, one code



# BLOCK CG

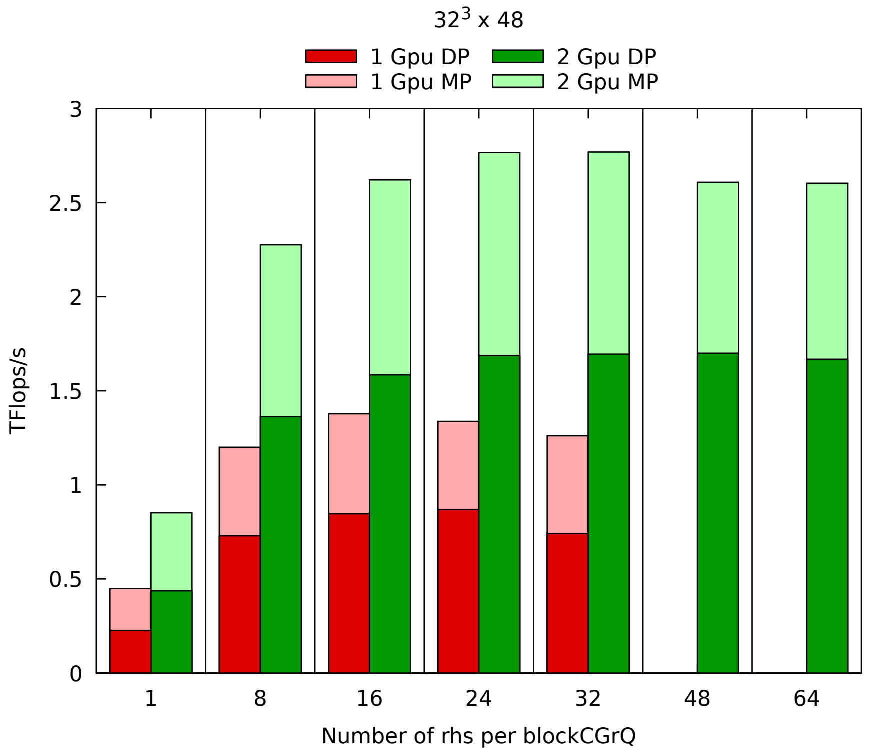
## turn vectors into matrices

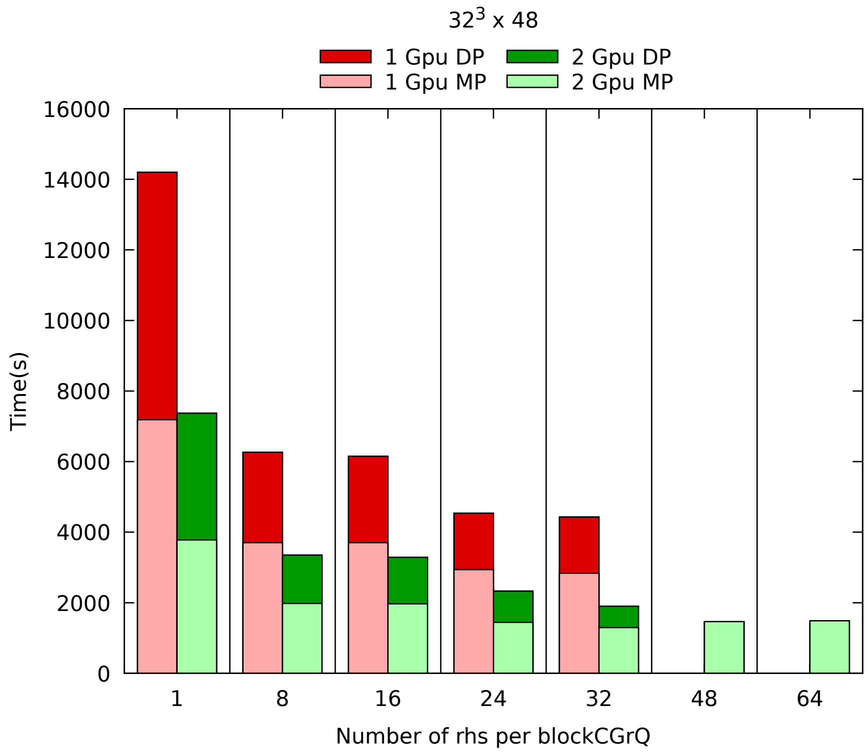


# MULTI-SRC SOLVERS ARE MULTI-PLICATIVE

## (More Flops) x (Less Iterations) -> Lower Time to solution



# SCALING

# BENCHMARKING TESTBED

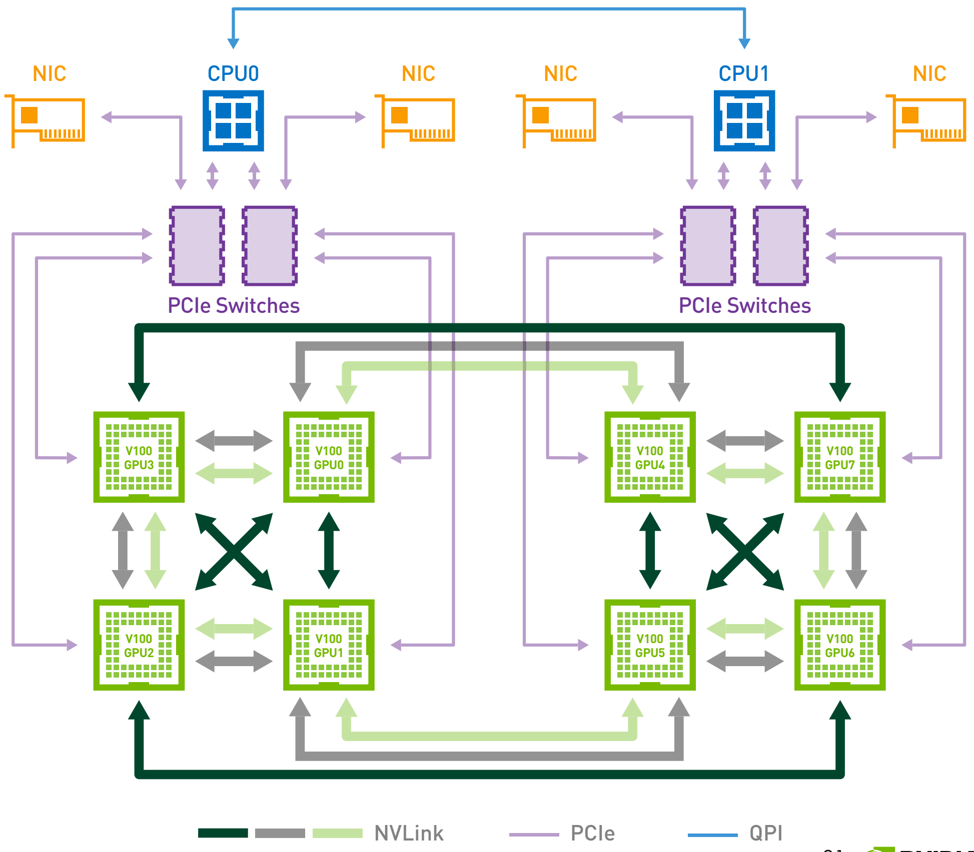
## NVIDIA Prometheus Cluster

36x DGX-1 nodes

DGX-1

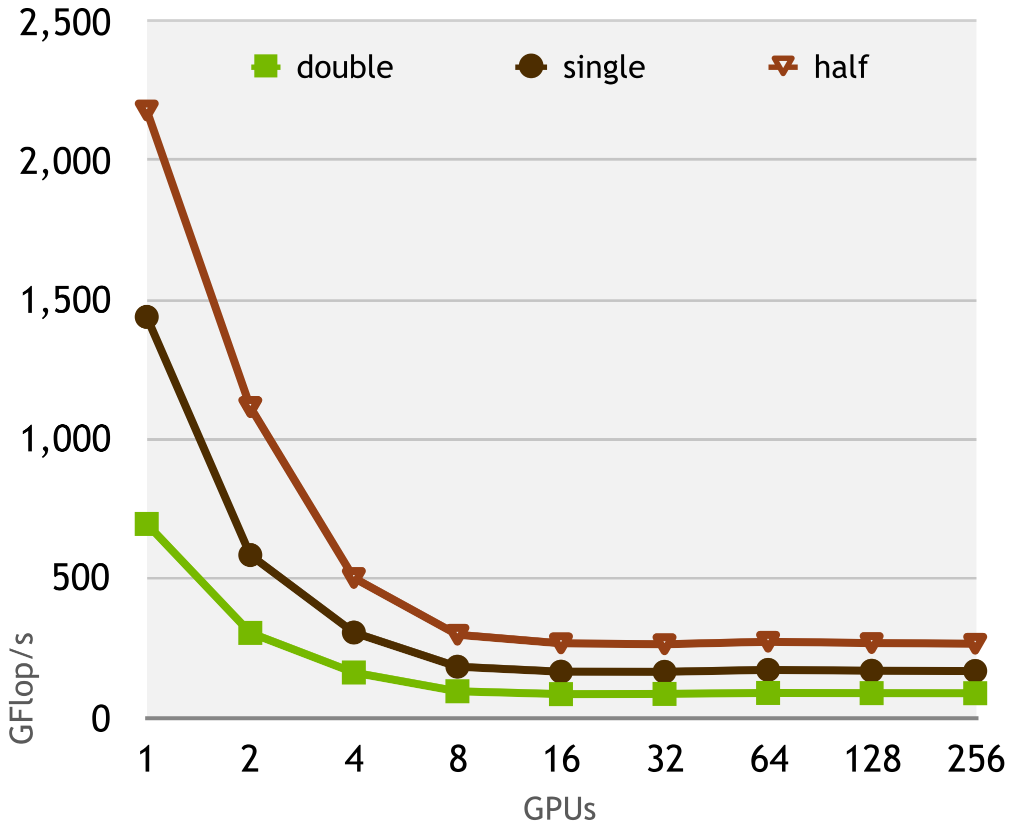
- 8x V100 GPUs connected through NVLink
- 4x EDR for inter-node communication
- Optimal placement of GPUs and NIC

Balanced GPU / IB configuration



# BASELINE PERFORMANCE

## $24^4$x16 local volume, domain wall Shamir, mixed precision CG




Original style (SC'10, SC'11 papers)

All MPI messages routed through CPU

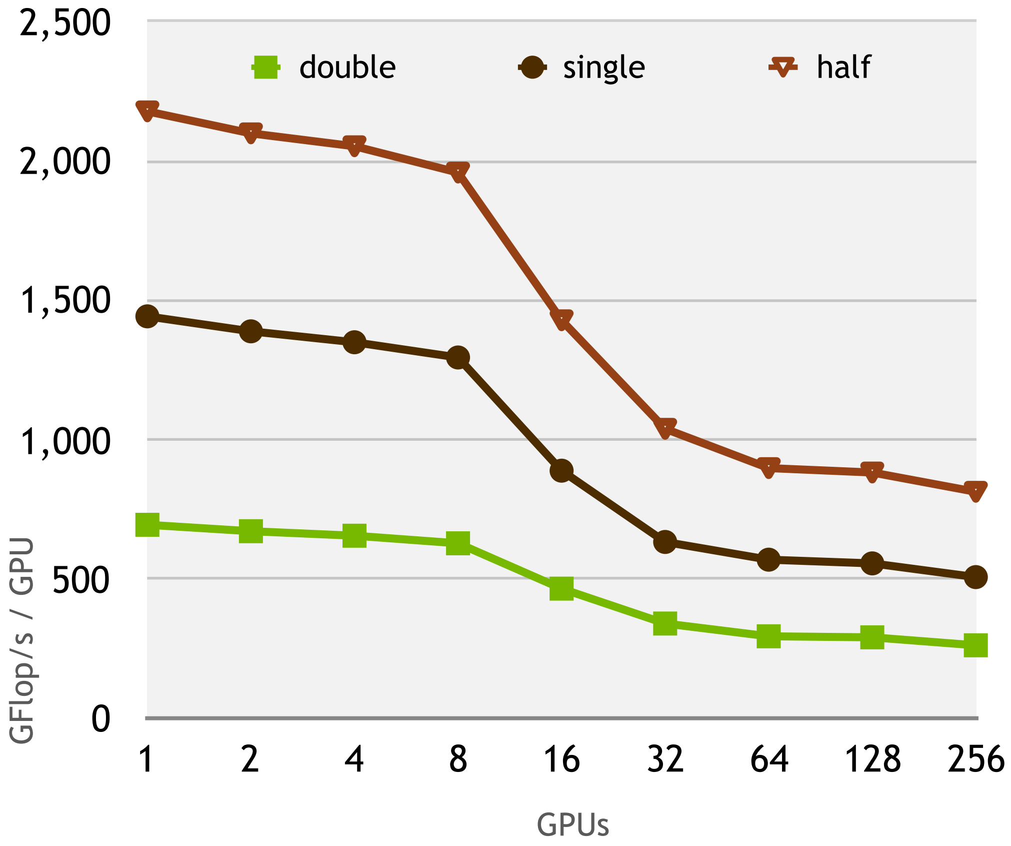
Reasonable scaling on Cray XK/XC (Titan)

Multi-dimensional pipelining works well

Disaster on dense node system

# GPUDIRECT RDMA

## with Peer-to-Peer intranode




Intra-node communication over NVLink

GPUDirect RDMA for inter-node comms

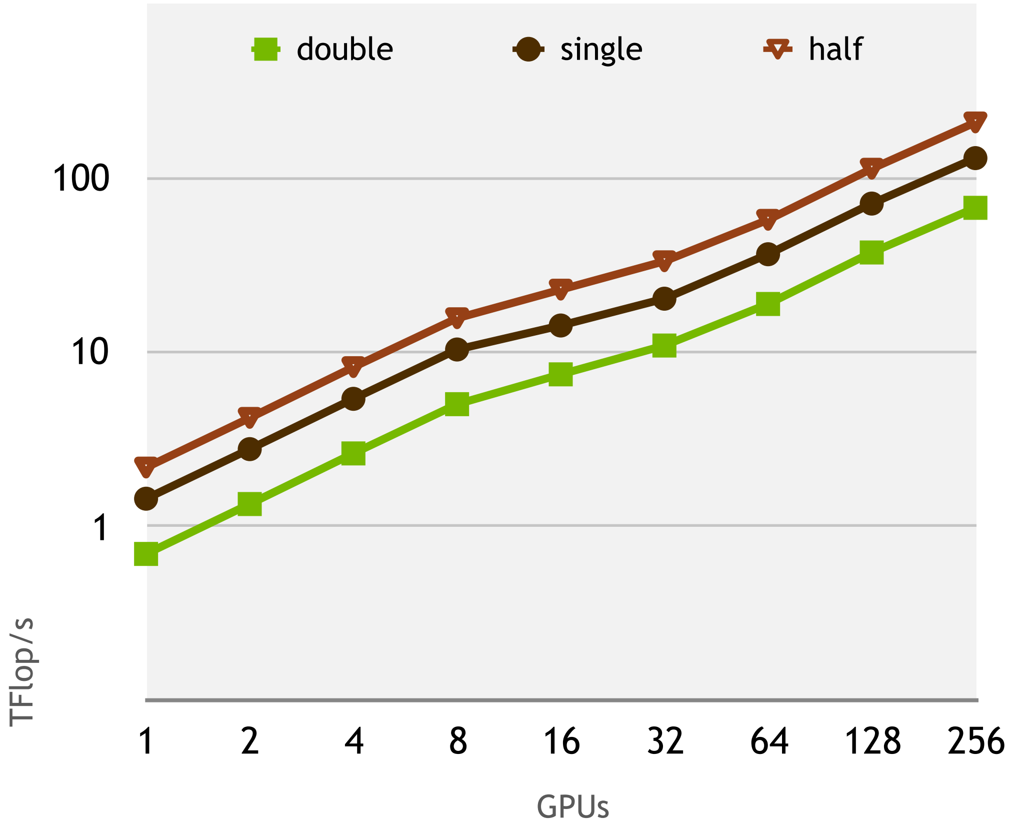
direct transfer between NIC / GPU

Still far from ideal weak scaling

Requires balanced system

# GPUDIRECT RDMA

## with Peer-to-Peer intranode




Intra-node communication over NVLink

GPUDirect RDMA for inter-node comms
direct transfer between NIC / GPU

Still far from ideal weak scaling

Requires balanced system

# SCALING FURTHER

## NVSHMEM gets the CPU out of the way

Increasingly latency limited as GPUs get faster

- Overhead from calling CUDA API / MPI routines
- Halo-region updates do not saturate the GPU

NVSHMEM: Implementation of OpenSHMEM, a Partitioned Global Address Space (PGAS) library

- Removing reliance on CPU for communication avoids overheads
- Parallelism for implicit compute – communication overlap

Improving performance while making it easier to program

Currently in early-access (limited to single node): Infiniband support soon

# SUMMARY

# QUDA - LATTICE QCD ON GPUS

## Breaking the barriers for 10 years: Exascale, take cover!

Volta, NVLink, NVSwitch are a big step up for LQCD: GPUs are here to stay

Multi-Source exploits locality and increases parallelism: more compute / bandwidth

Multigrid is in production HMC code for Summit: towards 100x more throughput

Scaling improvements through P2P, GDR and topology awareness

GPU centric communication for the Exascale

# RECOMPILE AND RUN FASTER: QUDA GETS YOU READY FOR EXASCALE SYSTEMS WITH EXASCALE ALGORITHMS

http://lattice.github.io/quda/

NVIDIA