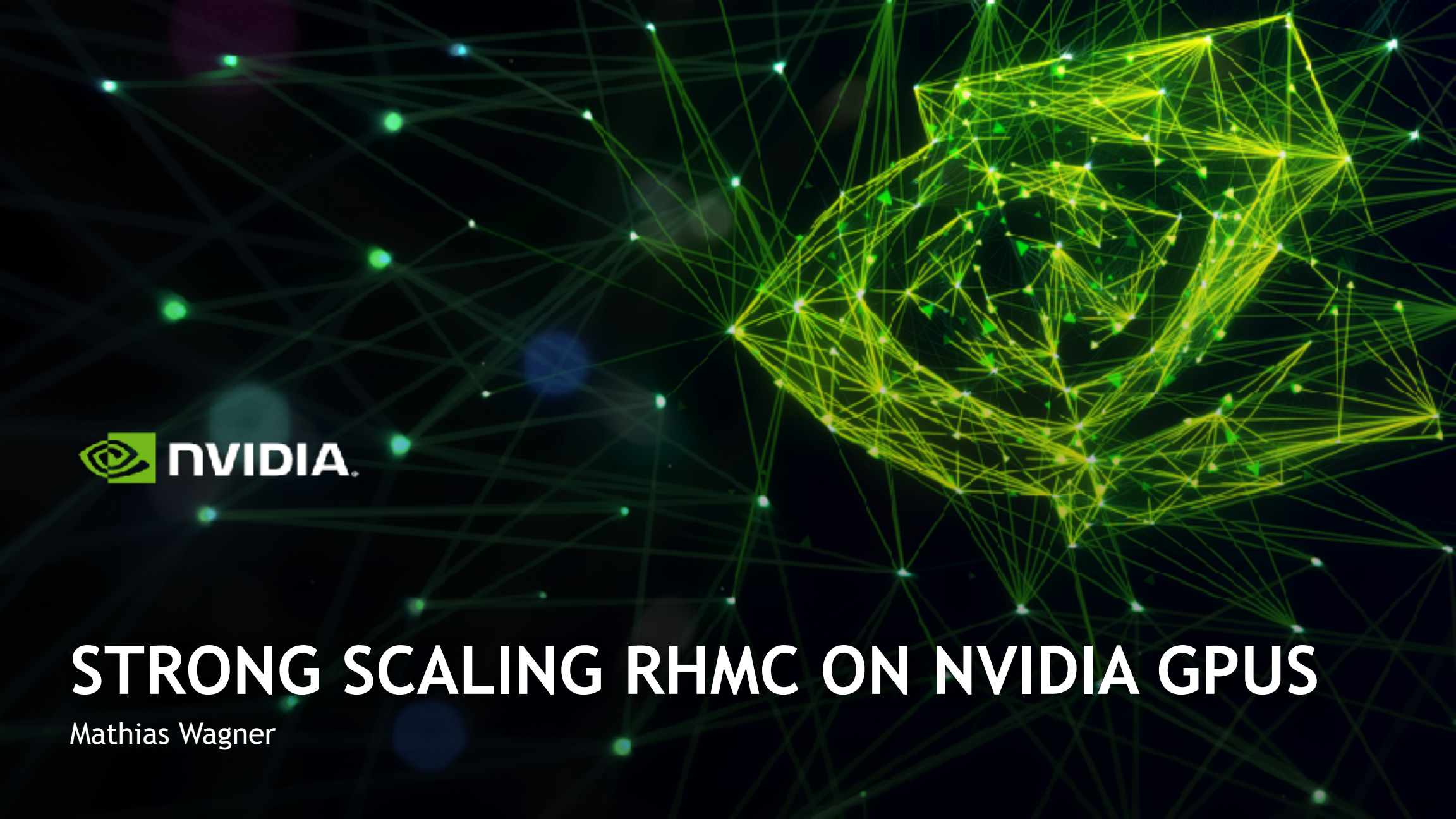


STRONG SCALING RHMC ON NVIDIA GPUS

Mathias Wagner

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 (LLNL)
Bálint Joó (Jlab -> OLRNL)
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 (NVIDIA)
Alejandro Vaquero (Utah University)
Mathias Wagner (NVIDIA)
Evan Weinberg (NVIDIA)
Frank Winter (Jlab)

QUDA 1.1

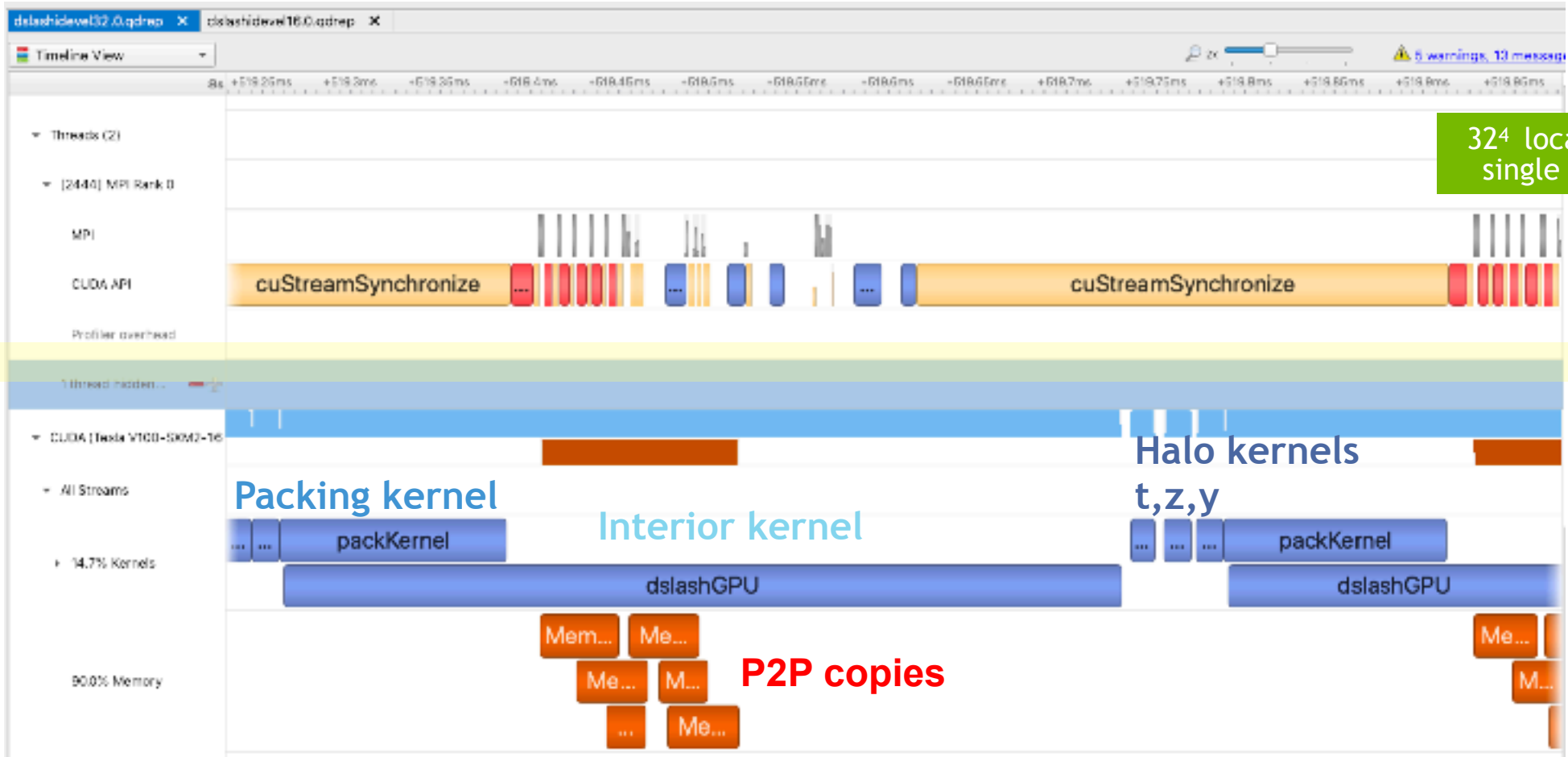
Summer 2021

- **Add support for NVSHMEM communication for the Dslash operators, for significantly improved strong scaling**.
- Addition of the MSPCG preconditioned CG solver for Möbius fermions.
- Addition of the Exact One Flavor Algorithm (EOFA) for Möbius fermions.
- Addition of a fully GPU native Implicitly Restarted Arnoldi eigensolver (as opposed to partially relying on ARPACK).
- Significantly reduced latency for reduction kernels through the use of heterogeneous atomics. Requires CUDA 11.0+.
- Addition of support for a split-grid multi-RHS solver.
- Continued work on enhancing and refining the staggered multigrid algorithm. The MILC interface can now drive the staggered multigrid solver.
- Multigrid setup can now use tensor cores on Volta, Turing and Ampere GPUs to accelerate the calculation.
- Improved support of managed memory through the addition of a prefetch API.
- Improved the performance of using MILC RHMC with QUDA
- Add support for a new internal data order FLOAT8.
- Remove of the singularity from the reconstruct-8 and reconstruct-9 compressed gauge field ordering.
- The clover parameter convention has been codified.
- QUDA now includes fast-compilation options which enable much faster build times for development at the expense of reduced performance.
- Add support for compiling QUDA using clang for both the host and device compiler.
- Significant cleanup of the tests directory to reduce boiler plate and extend ctest.
- General improvements to the cmake build system using modern cmake features. We now require cmake 3.15.
- Fix a long-standing issue with multi-node Kepler GPU and Intel dual socket systems.
- Improved ASAN integration: SANITIZE builds now work out of the box
- Multiple bug fixes and clean up to the library.

GPU-CENTRIC COMMUNICATION

MULTI-GPU PROFILE

overlapping comms and compute



32^4 local volume, single precision

DGX-1, 1x2x2x2 partitioning

STRONG SCALING

Reduce time to solution

Same problem size

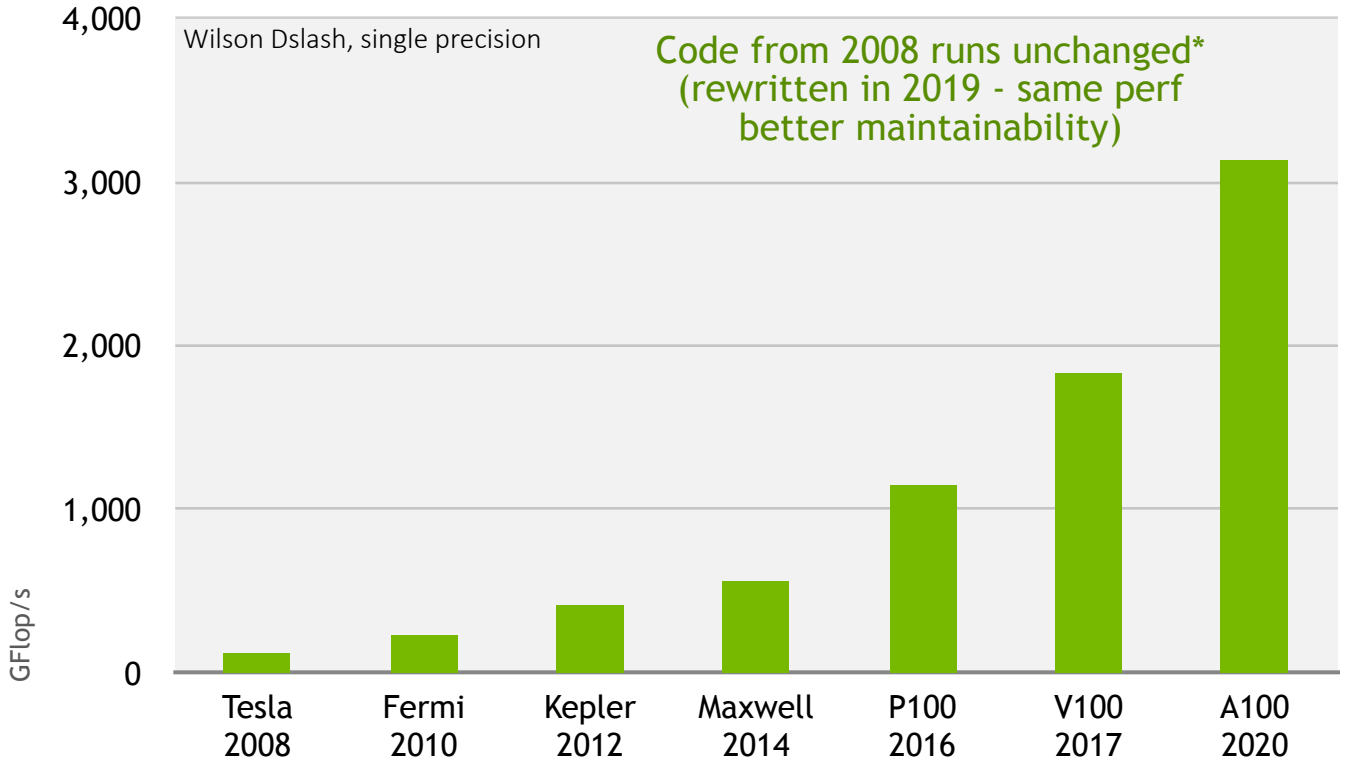
more resources

- more nodes (GPUs)
- faster GPUs (next generation)

Weak scaling: scale problem size with resources

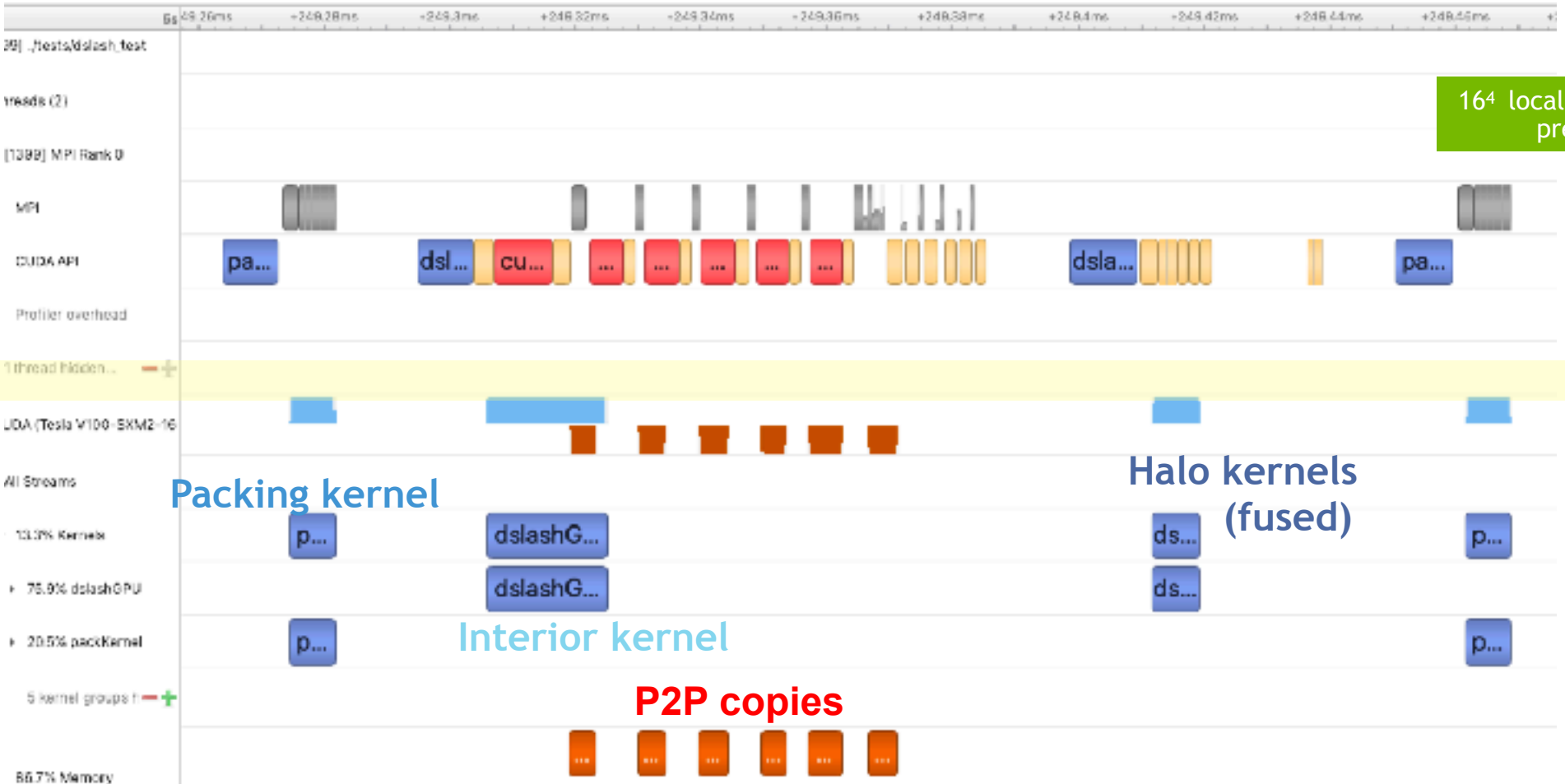
SINGLE GPU PERFORMANCE

Wilson Dslash Kernel



STRONG SCALING PROFILE

~~overlapping~~ comms and compute



NVIDIA

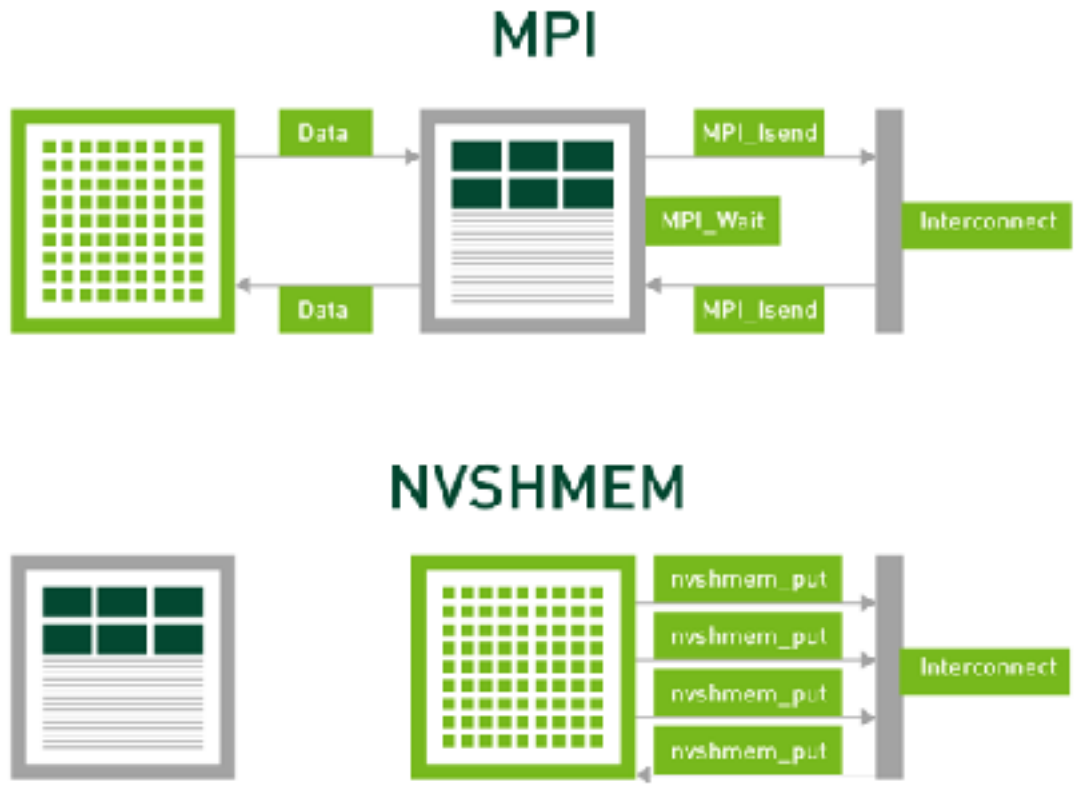
NVSHMEM: OpenSHMEM FOR CLUSTERS OF NVIDIA GPUS

- Compute on GPU
- Communication from GPU

Benefits:

- Eliminates offload latencies
- Improves overlap of computation and communication
- Hides latencies using multithreading
- Easier to express scalable algorithms with inline communication

NVSHMEM's Partitioned Global Address Space (PGAS) model improves performance while making it easier to program



FINE-GRAINED SYNCHRONIZATION

libcu++ gives us std::atomic in CUDA

Need replacement for kernel boundaries

Packing is independent of interior

interior and exterior update on boundary → possible race condition

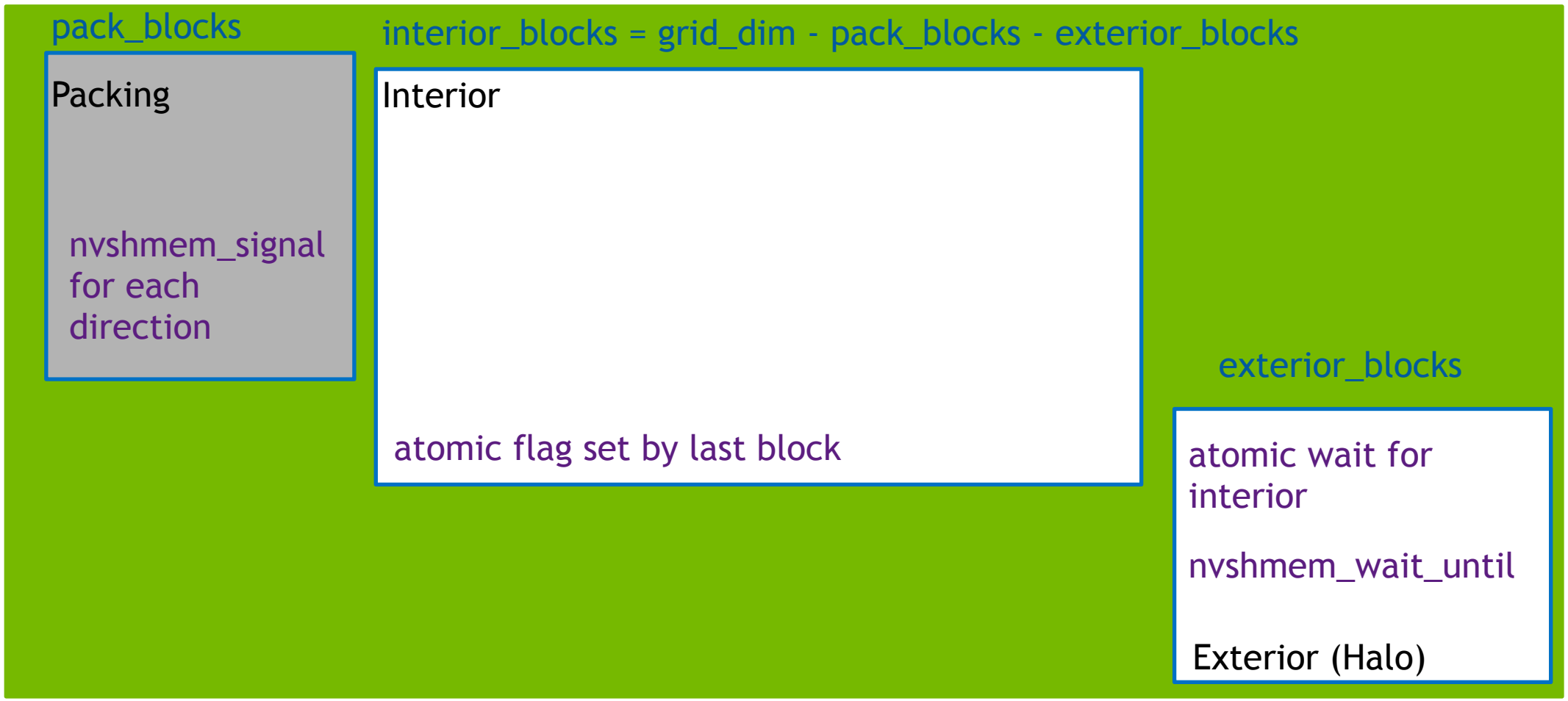
use cuda::atomic from libcu++

```
#include <atomic>
std::atomic<int> x;

#include <cuda/std/atomic>
cuda::std::atomic<int> x;

#include <cuda/atomic>
cuda::atomic<int, cuda::thread_scope_block> x;
```

FULLY FUSED DSLASH KERNEL



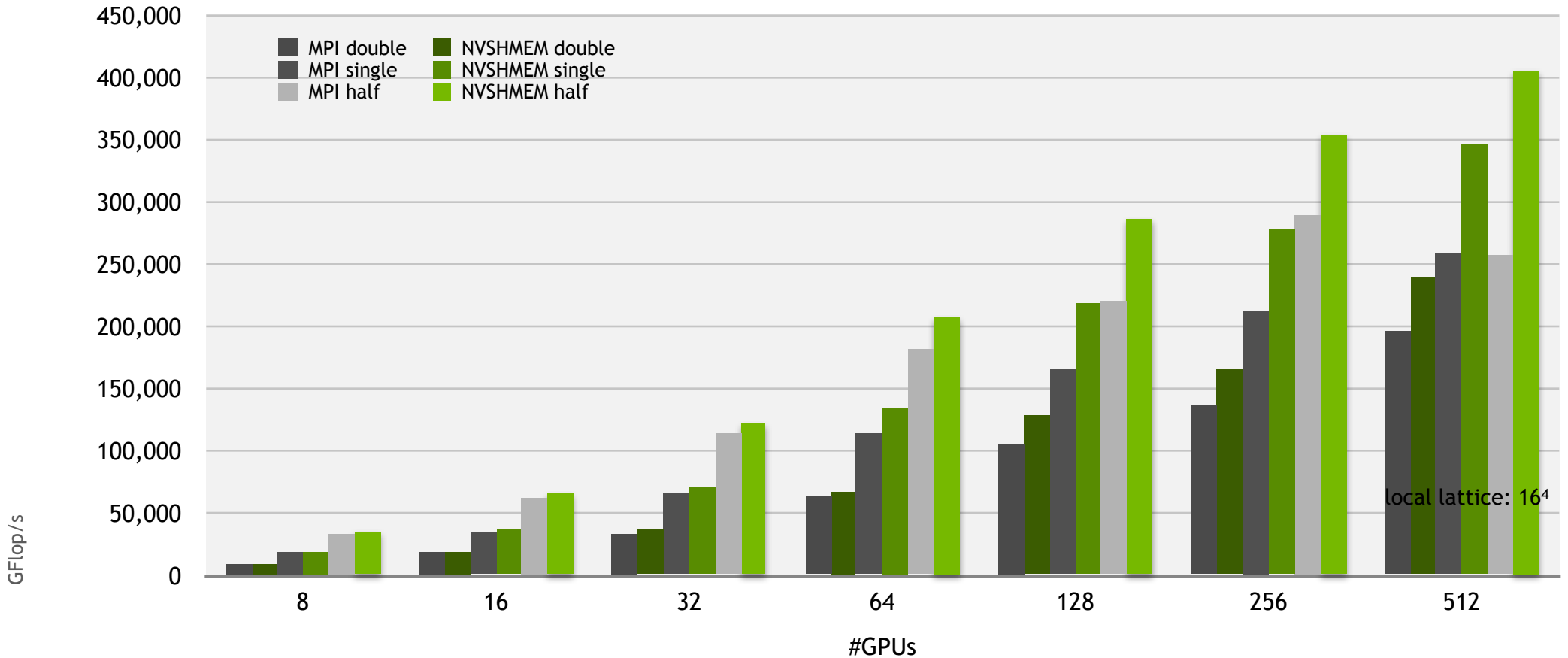
FULLY FUSED KERNEL

16^4 local volume, half precision

DGX-1, 1x2x2x2 partitioning

SELENE STRONG SCALING

Global Volume $64^3 \times 128$, Wilson-Dslash



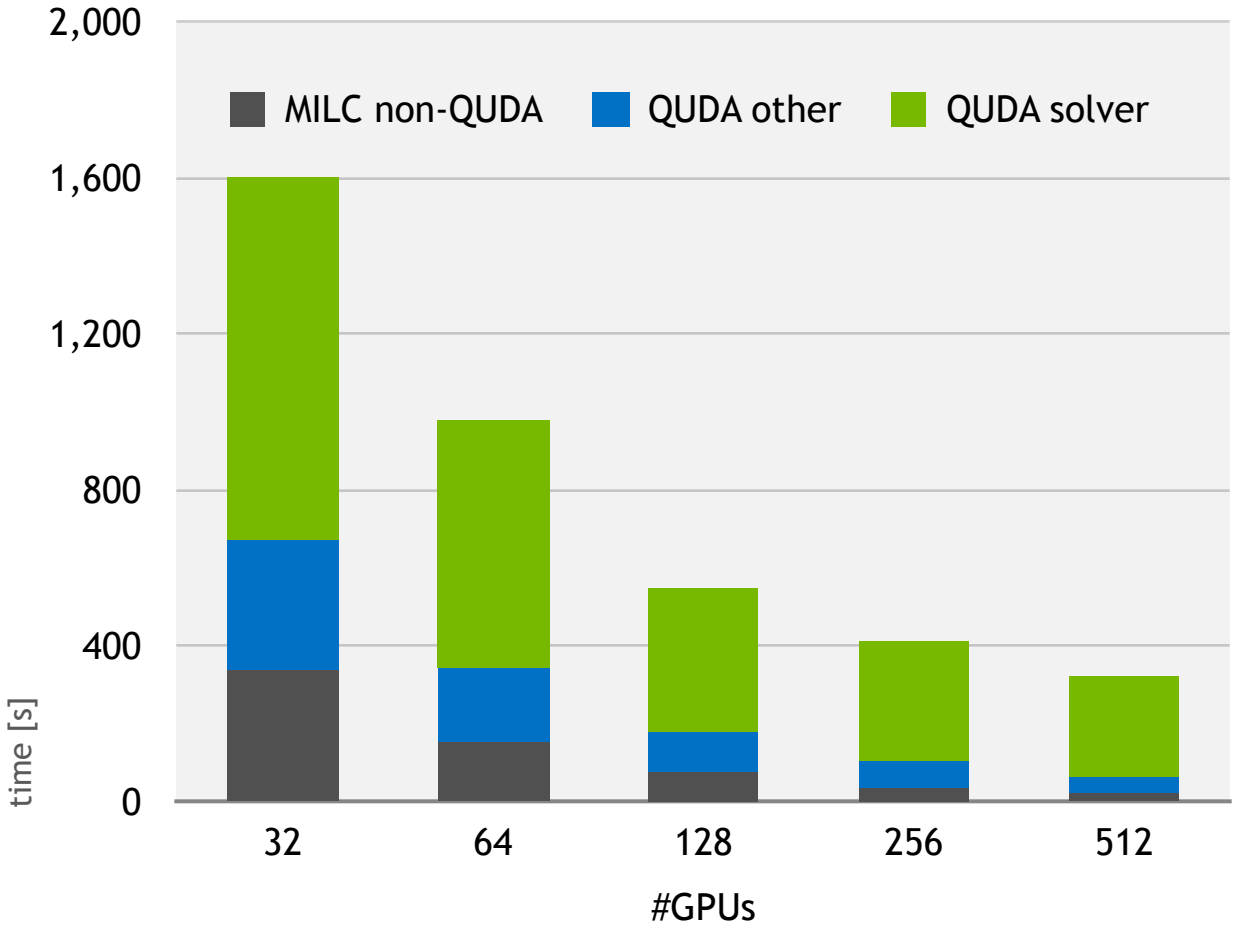
RHMC SCALING

MILC NERSC BENCHMARK OVERVIEW

- MILC NERSC Benchmark comes in 4 lattice sizes
  - small 18^3x36, medium **36^3x72**, large 72^3x144, x-large 144^3x288
- Benchmark runs the RHMC algorithm
  - Dominated by the multi-shift CG sparse linear solver (stencil operator)
  - Also have auxiliary “Force” and “Link” computations
- Since 2012 MILC has built-in QUDA support
  - Enabled through a Makefile option
  - All time-critical functions off loaded to QUDA

MILC HMC SCALING ON SELENE

NERSC LARGE BENCHMARK 72^3x144



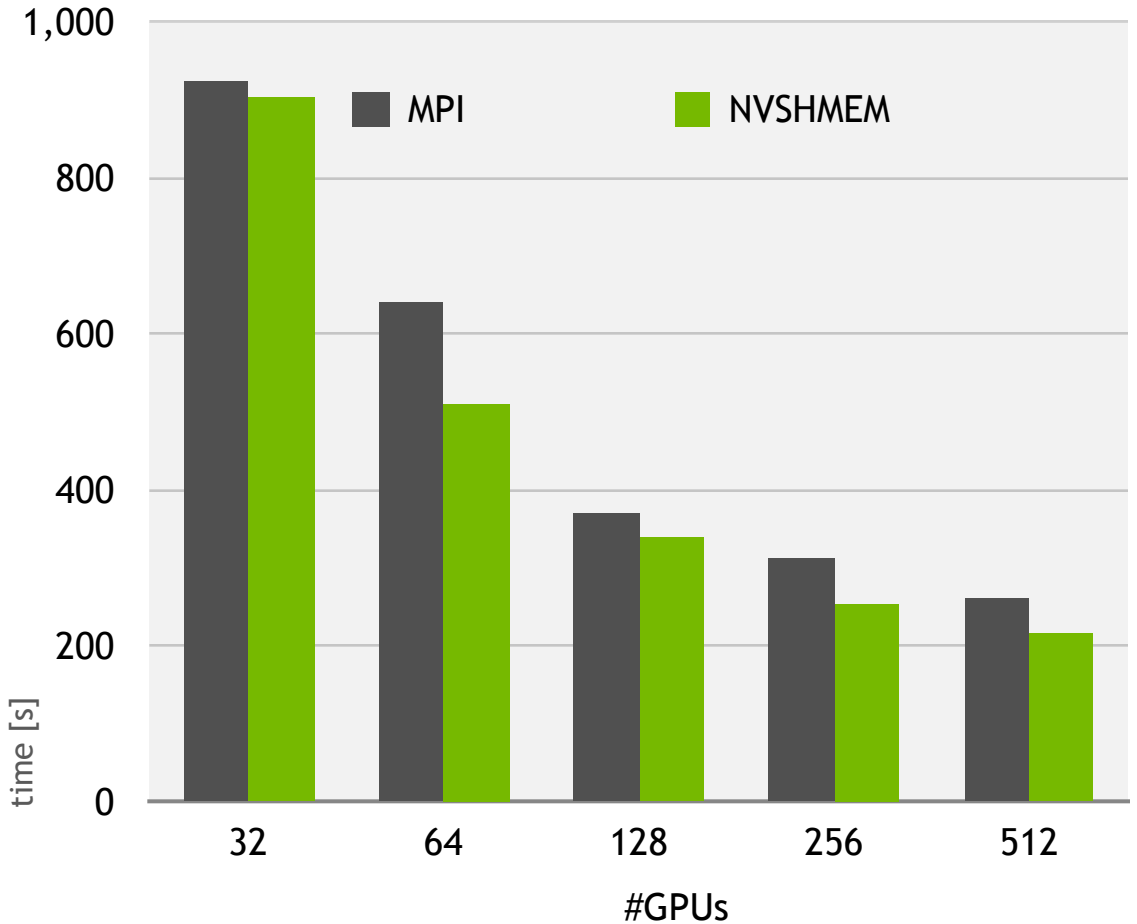
Running with MPI

other part scales reasonably
(not limited by communication)

solver part needs improvements

MILC SOLVER SCALING ON SELENE

NERSC LARGE BENCHMARK $72^3 \times 144$



mixed precision methods:

- lower precisions harder to scale
- NVSHMEM crucial for mixed precision

QUDA solver in MILC

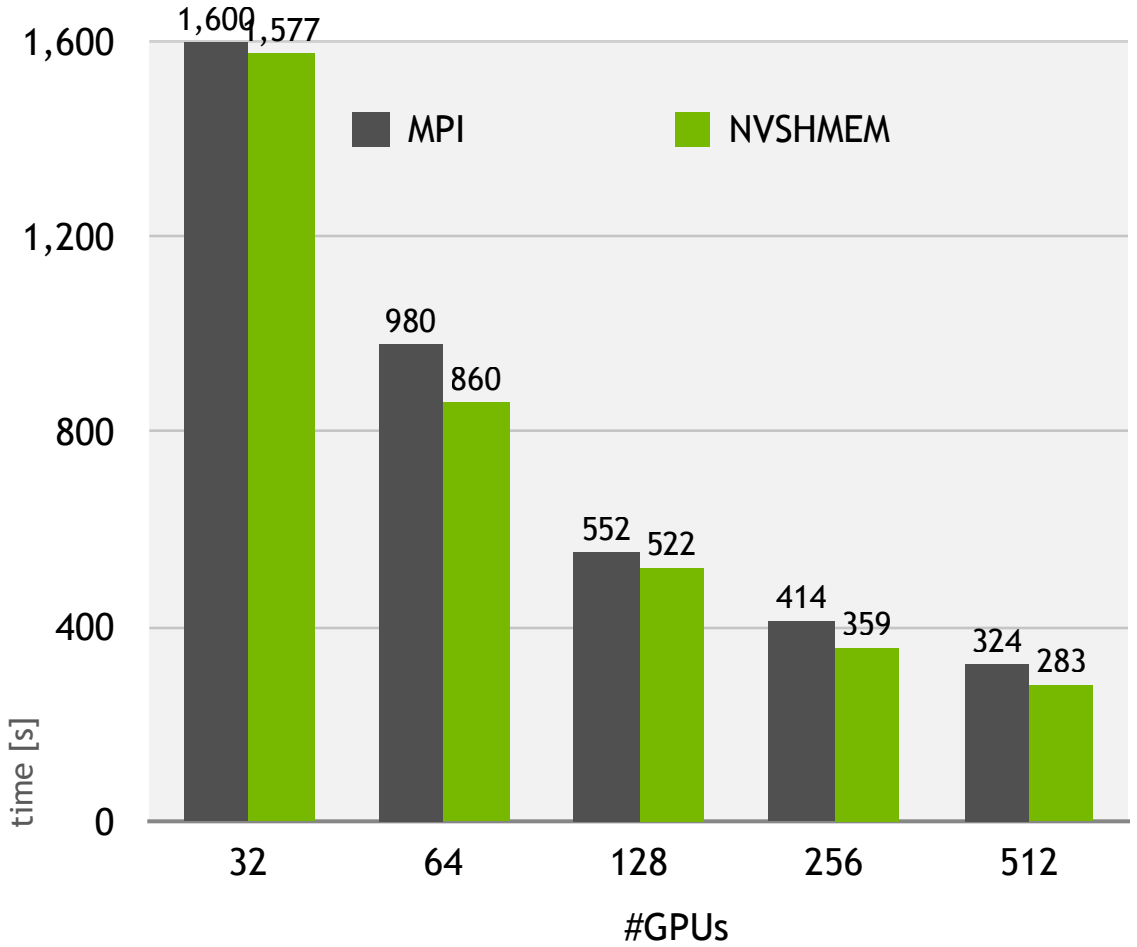
- mixed precision multishift: double-single
- mixed precision refinement: double-half

sweet spot at 256 GPUs:

- ~20% less time in solver

MILC SOLVER SCALING ON SELENE

NERSC LARGE BENCHMARK $72^3 x144$

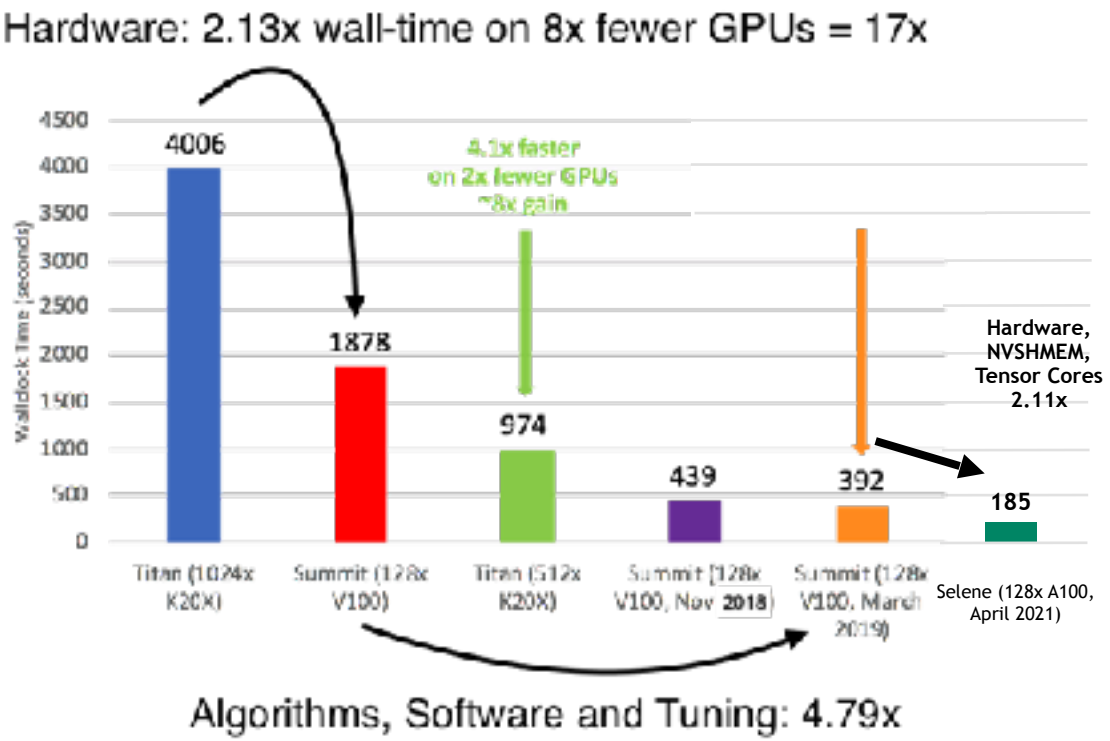


Full benchmark

>10% gains for runtime

CHROMA WILSON-CLOVER HMC

- Dominated by QUDA Multigrid
- Few solves per gauge configuration, can be bound by “heavy” (well-conditioned) solves
  - Evolve and refresh coarse space as the gauge field evolves
- Mixed precision an important piece of the puzzle
  - Double – outer solve precision
  - Single – GCR preconditioner
  - Half – Coarse operator precision
  - Int32 – deterministic parallel coarsening
- Wilson-clover MG implemented in **QUDA**, hooked into **Chroma** ; support in **Grid**
- **Latest and greatest:** Wilson-clover NVSHMEM plus tensor-core-accelerated setup

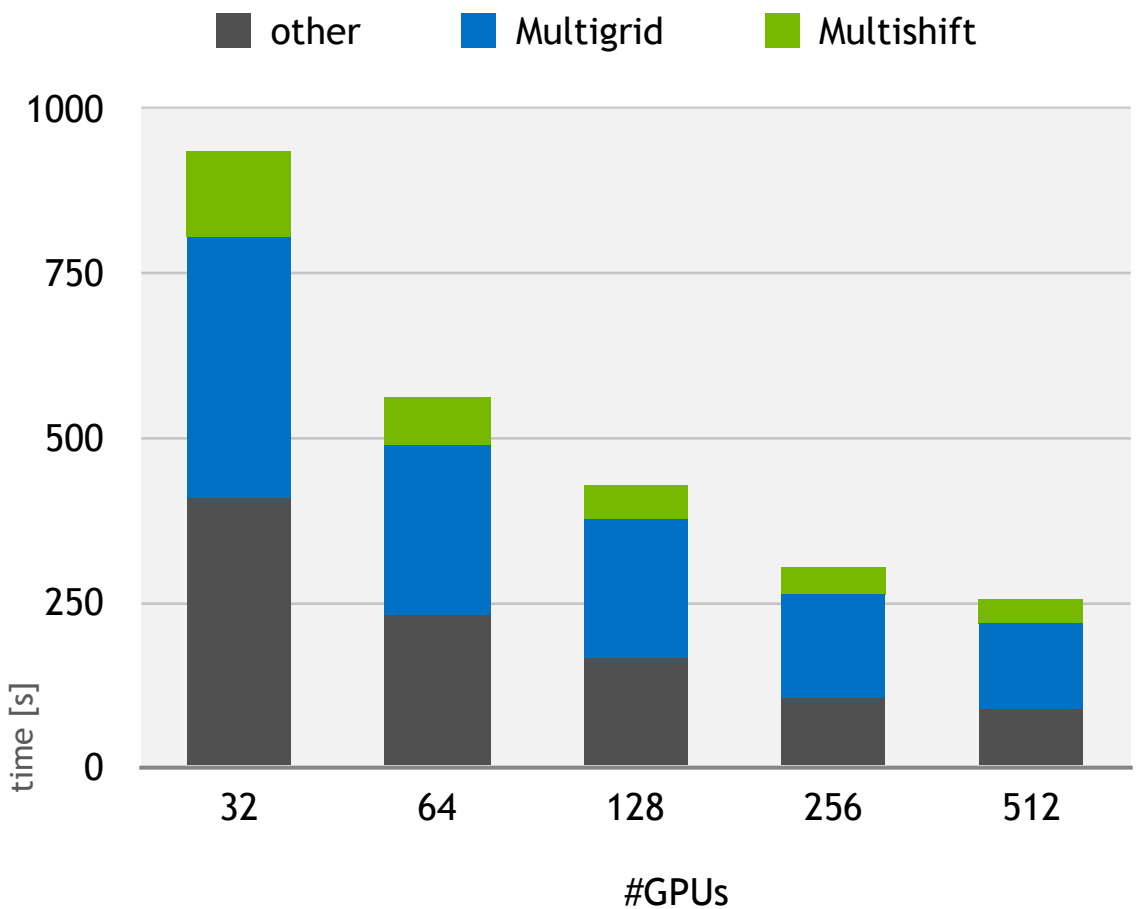


Chroma w/ QDP-JIT and QUDA, ECP FOM data, $V=64^3x128$ sites, m_π ~172 MeV, (QDP-JIT by F. Winter, Jefferson Lab)

Original figure credit Balint Joo

CHROMA WILSON-CLOVER HMC SCALING

MPI timing breakdown on Selene



2 trajectories

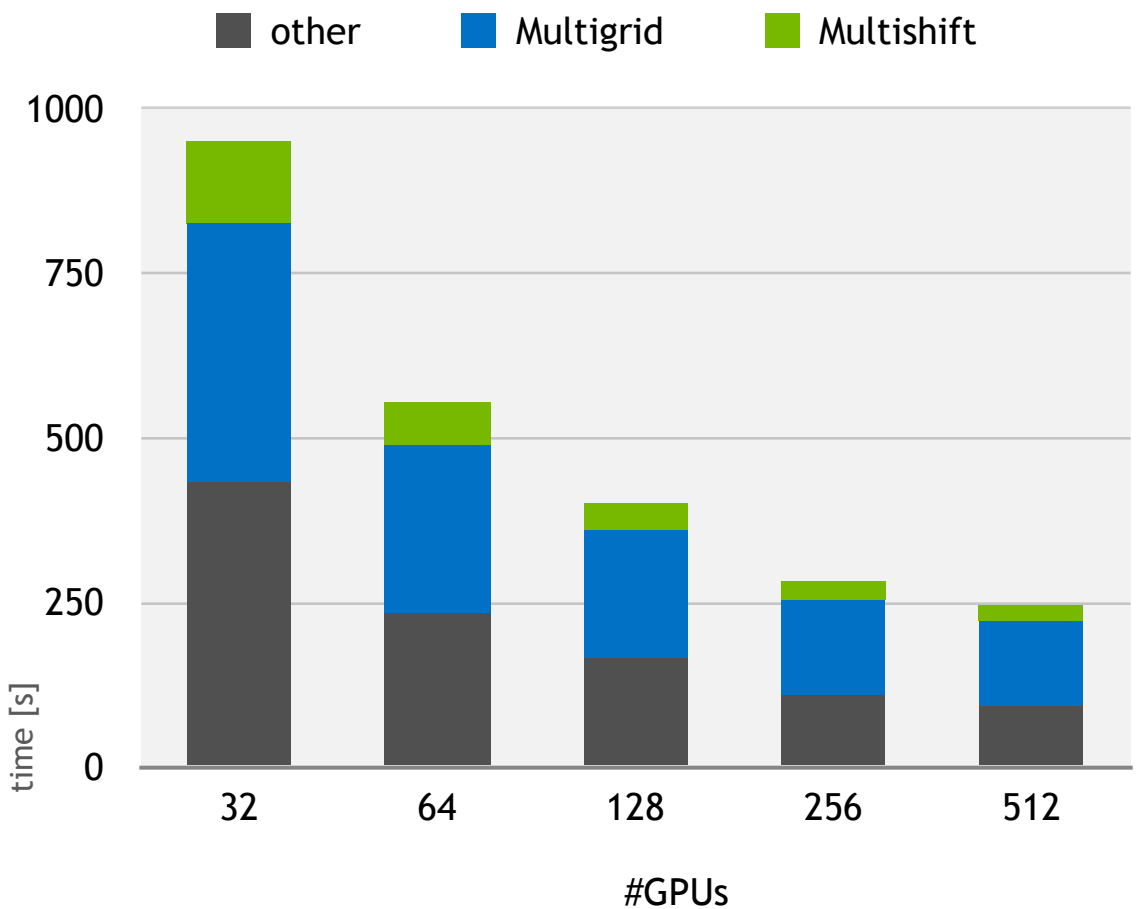
other (non QUDA)

QUDA multigrid (MPI/QMP)

QUDA multishift (MPI/QMP)

CHROMA WILSON-CLOVER HMC SCALING

NVSHMEM timing breakdown on Selene



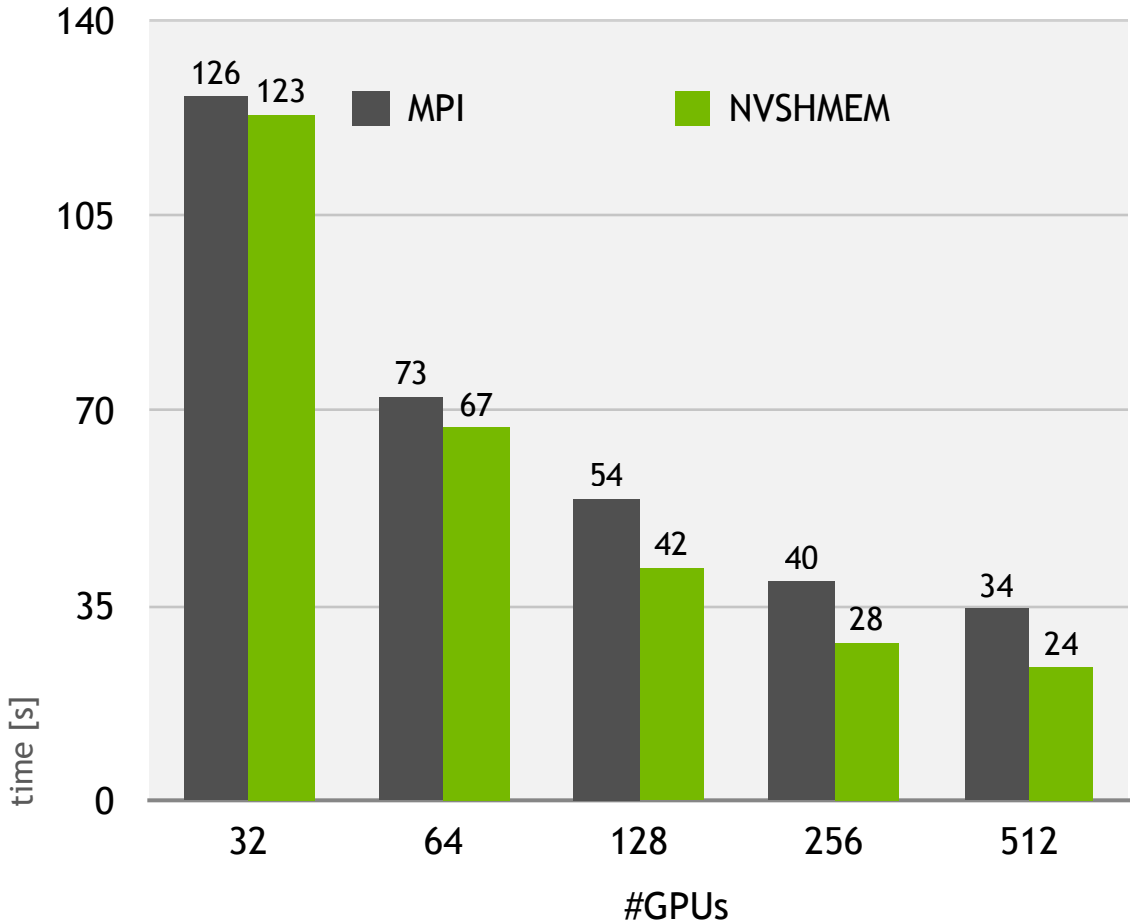
2 trajectories

other (non QUDA)
QUDA multigrid (MPI/QMP)
QUDA multishift (NVSHMEM)

Next step:
add NVSHMEM for multigrid

CHROMA WILSON-CLOVER HMC SCALING

MULTISHIFT COMPARISON



Using double-single multishift and double-half refinement step

OUTLOOK NVSHMEM WITH QUDA

NVSHMEM policies available in QUDA 1.1

RHMC scaling benefits in MILC and Chroma

- MILC dominated by Multishift: Full NVSHMEM benefit
- Chroma dominated by Multigrid, but Multishift part looks good

Further improve performance over IB

- nvshmem_putmem_signal allows signaling and data transfer in 1 IB transaction
- improved support for DWF/multi-rhs (pipelined transfers)

NVSHMEM everywhere:

- Coarse Dslashes for MultiGrid
- Further kernel fusion (multiple Dslash applications, solvers)