SciDAC
Scientific Discovery through Advanced Computing

USQCD US Lattice Quantum Chromodynamics

QUDA: QCD in CUDA for Multi-GPU Lattice Field Theory

Rich Brower (SciDAC software co-ordinator)

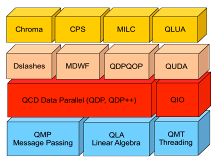
Stage 1: Basic Linear Solvers

QUDA (QCD in CUDA) library started in 2008 with NVIDIA's CUDA implementation by Kip Barros and Mike Clark at Boston University. It has expanded to a broad base of USQCD SciDAC [1] software developers and is in wide use as the GPU backend for HEP and NP SciDAC application codes: Chroma, CPS, MILC, etc.

Provides:

-- Various solvers for several discretizations, including multi-GPU support and domain-decomposed (Schwarz) preconditioners

-- Additional performance-critical routines needed for gauge-field generation

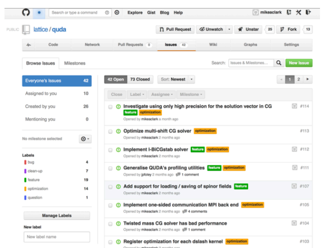


Maximize performance:

- Exploit physical symmetries
- Mixed-precision methods
- Autotuning for high performance on all CUDA-capable architectures
- Cache blocking

"QCD on CUDA" team – http://lattice.github.com/quda

- Ron Babich (NVIDIA)
- Kip Barros (LANL)
- Rich Brower (Boston University)
- Michael Cheng (Boston University)
- Mike Clark (NVIDIA)
- Justin Foley (University of Utah)
- Joel Giedt (Rensselaer Polytechnic Institute)
- Steve Gottlieb (Indiana University)
- Bálint Joó (Jlab)
- Claudio Rebbi (Boston University)
- Guochun Shi (NCSA -> Google)
- Alexei Strelchenko (Cyprus Institute -> FNAL)
- Hyung-Jin Kim (BNL)
- Frank Winter (UoE -> Jlab)



Reduce on Card Memory Traffic

- **SU(3) matrices are all unitary complex matrices with det = 1**
  - **12-number parameterization: reconstruct full matrix on the fly in registers**

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \rightarrow \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix} \quad c = (a \times b)^*$$

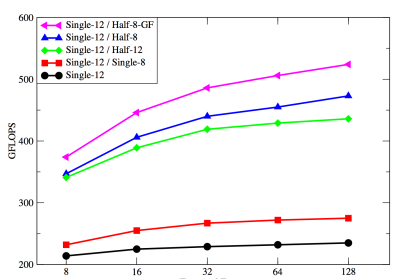
  - **Additional 384 flops per site**
  - **Also have an 8-number parameterization of SU(3) manifold (requires sin/cos and sqrt)**

$$\text{Group Manifold:} S_3 \times S_5$$

- **Impose similarity transforms to increase sparsity**
- **Still memory bound - Can further reduce memory traffic by truncating the precision**
  - **Use 16-bit fixed-point representation**
  - **No loss in precision with mixed-precision solver**
  - **Almost a free lunch (small increase in iteration count)**

K20X performance
$V = 24^3 \times T$
Wilson-Clover is ±10%
BiCGstab is -10%

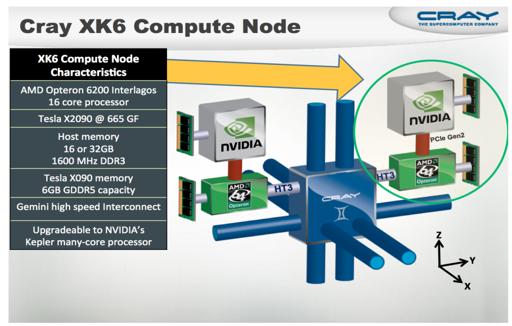
Kepler Wilson-Solver Performance



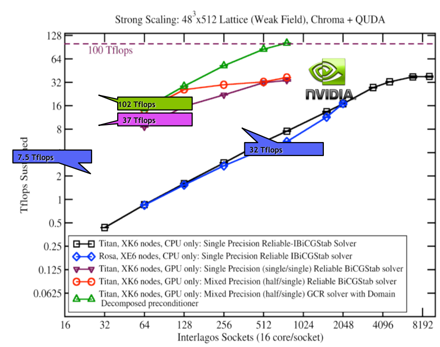
Kepler 20X with 2688 CUDA cores!

Stage 2: Scaling to Multi-GPUs

Communication Reduction between GPUs



Architecture Aware Algorithm: Domain Decomposition



Block Jacobi Domain Decomposition [2]

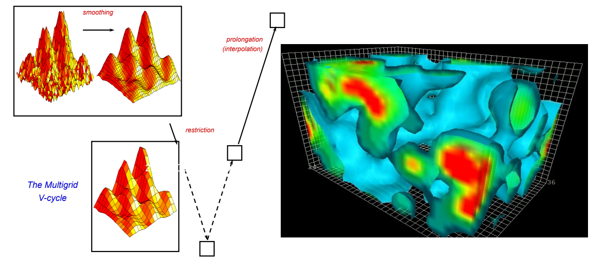
References

*[1] SciDAC-3 HEP: **Searching for Physics Beyond the Standard Model: Strongly-Coupled Field Theories at the Intensity and Energy Frontiers***

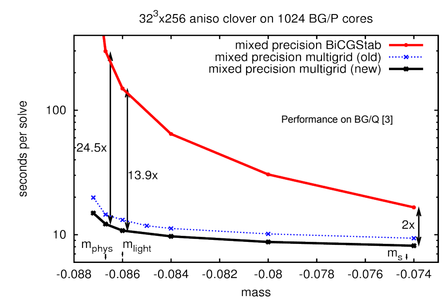
*[2] R. Babich, M. A. Clark, B. Joo, G. Shi, R. C. Brower and S. Gottlieb, **Scaling Lattice QCD beyond 100 GPUs** Super Computing 2011, arXiv:1109.2935 [hep-lat].*

*[3] R. Babich, J. Brannick, R. Brower, M. Clark, T. Manteuffel, S. McCormick, J. Osborn, C. Rebbi, **Adaptive Multigrid algorithm for the lattice Wilson-Dirac operator** Phys. Rev. Lett. 105, 201602 (2010)*

Stage 3: Multi-scale Physics



Physics Aware Algorithm: Adaptive Multigrid



GPU technology + MG → Reduce $ cost by over a factor of 1/100.
(GPU/MG project: Rich Brower, Michael Cheng and Mike Clark report at Lattice 2013)

Future: Synthesis DD + MG to satisfy Arch + Physics:

Need new research fitting multi-scale physics to hierarchical (multi-scale) computers.
Need compromise and auto tuning to bring them into a happy marriage!