Accelerating Quantum Chromodynamics calculations on GPU based systems with an Adaptive Multi-Grid Algorithm

K. Clark (NVIDIA), R. Brower (BU) M. Cheng (BU), A. Gambhir (W&M), B. Joó (Jefferson Lab), A. Strelchenko (FNAL), E. Weinberg (BU)
Introduction

• It is believed that the fundamental building blocks of matter are **quarks** bound together by **gluons**, via the **strong nuclear force**.

• Quantum Chromodynamics (QCD) is the theory which describes the strong interactions.

• Understanding how QCD makes up matter and how quarks and gluons behave is a subject of intense experimental scrutiny:
  - only ~5% of the mass of a proton comes from mass of the quarks, rest comes from binding
  - gluon self-coupling and gluon excitations can create **exotic forms of matter**

- meson: 2 quarks
- baryon: 3 quarks
- glueball: 0 quarks only gluons

GlueX in the new Hall-D of Jefferson Lab@12 GeV. Hunting for exotics!
LQCD Calculation Workflow

- **Gauge Generation**: Capability Computing on Leadership Facilities
  - configurations generated in sequence using Markov Chain Monte Carlo technique
  - focus the power of leadership computing onto single task exploiting data parallelism

- **Analysis**: Capacity computing, cost effective on Clusters
  - task parallelize over gauge configurations in addition to data parallelism
  - can use clusters, but also LCFs in throughput (ensemble) mode.
The Wilson-Clover Fermion Matrix

\[ M_{x,y} = \left( N_d + M - \frac{ic_{SW}}{8} \sum_{\mu<\nu} [\gamma_\mu, \gamma_\nu] F^{\mu\nu}(x) \right) \delta_{x,y} - \frac{1}{2} \sum_\mu \left( 1 - \gamma_\mu \right) U_\mu(x) \delta_{x,x+\hat{\mu}} + \left( 1 + \gamma_\mu \right) U_\mu^+(x-\hat{\mu}) \delta_{x,x-\hat{\mu}} \]

- "Dslash" Term is a nearest neighbor stencil like term - very sparse
- M is J-Hermitian with J=\( \gamma_5 \):  \( \gamma_5 M = M^\dagger \gamma_5 \)  \( \gamma_5 = \gamma_1 \gamma_2 \gamma_3 \gamma_4 \)  \( \gamma_5^2 = 1 \)  \( \gamma_5 = \gamma_5^\dagger \)
- \( \gamma_5 \) is maximally indefinite (ev-s are 1,-1)
Gauge Configuration Generation

- Gauge Generation proceeds via Hybrid Molecular Dynamics Monte Carlo (e.g. HMC)
- Momentum Update Step needs ‘Force’ term:
  \[ \pi_\mu(x) \leftarrow \pi_\mu(x) + F_\mu(x) \delta \tau \]
- Computing F needs to solve linear system:
  \[ M^\dagger M x = b \]
- For Wilson-Clover we can use two step solve:
  \[ M^\dagger y = b \quad M x = y \]
Analysis

- Quark Line Diagrams describe Physical Processes
- Each line is a Quark Propagator, solution of:

\[ Mq = s \]

- Many solves needed for each field configuration
  - e.g. 256 values of t x 386 sources x 4 values of spin
  - x 2 (light and strange quarks) = \( 790,528 \) isolves
  - Typically 200-500 configurations are used

- Single precision is good enough
- Same Matrix, Multiple Right Hand sides
Chroma Software Stack

- Layered Software
  - Algorithms in Chroma
  - Chroma coded in terms of QDP++
  - Fast Solvers Come from Libraries
    - QUDA on NVIDIA GPUs
- Different QDP++ Implementations provide ‘performance portability’ for Chroma
  - Chroma is 99% coded in terms of QDP++ constructs
  - QDP-JIT/PTX and QDP-JIT/LLVM using NVVM for GPUs
- Chroma wraps performance optimized libraries
  - can give e.g. QUDA solvers a ‘Chroma look & feel’

```
LatticeFermion psi, chi;
gaussian(psi); // gaussian RNG fill

// shift sites from forward 0 dir.
// nearest neighbor communication
chi = shift(psi, FORWARD, 0);

// Arithmetic expressions on lattice
// subsets
chi[rb[0]] += psi;

// Global reduction
Double n2 = norm2(chi);
```

Example QDP++ Code
Adaptive Multigrid in LQCD

- Critical Slowing down is caused by ‘near zero’ modes of M
- Multi-Grid (MG) method
  - separate (project) low lying and high lying modes
  - reduce error from high lying modes with “smoother”
  - reduce error from low modes on coarse grid
  - Gauge field is ‘stochastic’, so no geometric smoothness on low modes => algebraic multigrid
  - Setting up restriction/prolongation operators has a cost
  - Easily amortized in Analysis with O(100,000) solves
QUDA Implementation

- Outer Flexible Krylov Method: GCR
- MG V-cycle used as a Preconditioner.

  - Null space:
    - Solve $M \mathbf{x} = 0$ for $N_{\text{vec}}$ random $\mathbf{x}$ with BiCGStab
    - Construct $R$, $P$, $M_c$
  
  - Smoother: fixed number of iterations with MR
  
  - ‘Bottom Solver’: GCR
    - May be deflated (e.g. FGMRES-DR) later
    - Is recursively preconditioned by next MG level

- Coarsest levels may have very few sites
  - Turn to other ‘fine grained’ sources of parallelism
Benefits of Multigrid: Speed

- Algorithmic Speed Improvements
  - 5x-10x compared to BiCGStab
- BiCGStab running in optimal configuration:
  - Mostly low precision with ‘Reliable Update’ flying restarts
  - Mixed Precision (16-bit/64 bit)
  - ‘Gauge Field’ Compression
- MG is a preconditioner
  - Can run in reduced precision with flexible outer GCR solver.

from K. Clark et. al. SC’16 - sneak preview
Benfits of Multigrid: Optimality

- MG minimizes error, rather than residuum
- Solver is better behaved than BiCGStab
- number of iterations is stable
- \|error\|/\|residuum\| is more stable
- Important for t-to-same-t propagators
  - single precision is good enough
  - BUT:
  - want precision guarantee from solve to solve

from Clark et. al. SC’16 - sneak preview
Benefits of Multigrid: Power Efficiency

- Power Draw of a GPU node during BiCGStab and Multigrid running.
  - GPU Power only (nvidia-smi)
- Once setup is complete, integrated power for 12 solves is much less than for BiGCStab
- Ongoing optimizations
  - smarter setup
  - move more work to GPU

from Clark et. al. SC’16 - sneak preview
In Praise of Hackathons

• Hackathons bring together members of a distributed group for a burst of concentrated activity, to accomplish a concrete development goal.

• Hackathons
  - clear the calendar
  - are focused — no distractions
  - Interfaces developers from different sides of interfaces
  - teach new things (me@OLCF Hack: NVProf, NVIDIA Visual Profiler, Allinea MAP)

• Hackathons for QUDA
  - JLAB: Multi-Node QUDA (way back in 2011?), QUDA Multigrid & Chroma (Jan’16)
  - Fermilab: QUDA Deflation algorithms
  - OLCFHack’16: Multigrid in Chroma Gauge Generation, BiCGStab-L, Staggered Multigrid (Oct’16)

Team Hybrid Titans at OLCFHack. Photograph Courtesy of Sherry Ray, OLCF
Summary

- Taking advantage of modern architectures needs development both in the algorithmic space and in the ‘software’ space
  - algorithmic optimality, performance optimization, integration with existing codes
- Recent QUDA improvements provide Chroma code (and other users) with improved capability.
  - Multi-Grid solver now in production for propagator calculations on Titan and GPU clusters
  - Multi-Grid solver has been integrated into Chroma for Gauge Generation projects
- Hackathons (a.k.a Code-Fests) are a great way to make rapid advances
  - We love Hackathons!
- Please go and see Kate Clark’s Technical Paper Presentation: 2:30pm, Rm 355-E
Thanks and Acknowledgements

• Thanks for Organizing Hackathons:
  - Chip Watson (Jefferson Lab) for January Multi-Grid Mini-Hackathon at Jefferson Lab
  - OLCF for the October OLCFHack GPU Hackathon in Knoxville

• Results in this talk were generated on the OLCF Titan system (Cray XK7) utilizing USQCD INCITE (LGT003) allocations.

• This work is supported by the U.S. Department of Energy, Office of Science, Offices of Nuclear Physics, High Energy Physics, and Advanced Scientific Computing Research.

• This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under contract DE-AC05-06OR23177.