# Qlua

Stefan Krieg (BUW)

Andrew Pochinsky (MIT)

Christopher Schroeder (LLNL)

Sergey Syritsyn (BNL)

Oliver Witzel (BU)

+ HDF5 gang

Massachusetts
Institute of
Technology

# QLUA

Evolution of Qlua continues to be physics driven. The software is employed in production by several class A and class B USQCD programs, as well as exploration and development efforts.

Active users include groups outside USQCD.

# QLUA

Evolution of Qlua continues to be physics driven. The software is employed in production by several class A and class B USQCD programs, as well as exploration and development efforts.

Active users include groups outside USQCD.

- BG/Q inverters
- HDF5
- HMC
- Hypre
- Multigrid

# BG/Q Inverters

BlueGene/Q backend has been created for qa0. It allows one to convert low level inverter routines into BG/Q quad intrinsics. As a result, performance of MDWF and Clover inverters improved by 40%.

BG/Q is targeted without changes to .qa0 sources.

# BG/Q Inverters

BlueGene/Q backend has been created for qa0. It allows one to convert low level inverter routines into BG/Q quad intrinsics. As a result, performance of MDWF and Clover inverters improved by 40%.

BG/Q is targeted without changes to .qa0 sources.

Both MDWF and Clover Level III inverters have been extended with exact Lanczos EigCG.

# HDF5 for USQCD

Massachusetts
Institute of
Technology

# HDF5 for USQCD

## File drivers

- `posix` single-node write, serial data only
- `phdf5` multi-node write, serial and parallel data
- `mpiposix` multi-node write, serial and parallel data

High-level control of file organization is provided for optimizing storage throughput (chunking, alignment, GPFS hints, transfer modes)

# HDF5 for USQCD

## Object attributes

- `kind` standard string describing object's kind
- `time` 64 bit signed int time (μs since UNIX epoch)
- `sha256` SHA-256 checksum of the dataset
- other attributes ignored by readers

# HDF5 for USQCD

## Serial data types

Storage is compatible with SciPy conventions.

Serial data are written in HDF5 scalar dataspaces. Floating point data can be written in either single or double precision. The following types are currently provided:

```
String, Real, Complex, VectorInt(M), VectorReal(M),
VectorComplex(M), MatrixReal(N,M), MatrixComplex(N,M),
ColorVector(N), ColorMatrix(N), DiracFermion(N),
DiracPropagator(N)
```

# HDF5 for USQCD

## Lattice data types

Lattice data are written in HDF5 simple dataspaces. Each object has its own lattice geometry. Floating point data can be written in either single or double precision. The following types are currently provided:

```
LatticeInt, LatticeReal, LatticeComplex,
LatticeColorVector(N), LatticeColorMatrix(N),
LatticeDiracFermion(N), LatticeDiracPropagator(N)
```

# HDF5 for USQCD

## Example

```
hf = qcd.hdf5.Reader(“prop-sample.h5”);
p_forward = hf:read(“/u1750/forward/G24.2/x4y16z7t0/prop.61”)
p_backward = hf:read(“/u1750/backward-61/P/t19/px0py0pz-1/prop”)
hf:close()
```

# HMC

# HMC

- Design plug-and-play HMC environment

# HMC

- Design plug-and-play HMC environment
- Qlua HMC interface

# HMC

- Design plug-and-play HMC environment
- Qlua HMC interface
- Krylov solvers

# HMC

- Design plug-and-play HMC environment
- Qlua HMC interface
- Krylov solvers
- Qlua HMC prototype implementation

# HMC

- Design plug-and-play HMC environment
- Qlua HMC interface
- Krylov solvers
- Qlua HMC prototype implementation
- Extend Level III solvers interfaces

# HMC

- [x] Design plug-and-play HMC environment
- [x] Qlua HMC interface
- [x] Krylov solvers
- [x] Qlua HMC prototype implementation
- [ ] Extend Level III solvers interfaces

# HYPRE and Qlua

- Define abstraction layer (HQL) between HYPRE and USQCD
  - Wilson-Clover operator in Qlua for test and validation
  - Up-interface to QDP
  - Down-interface to HYPRE
- Access to HQL for Qlua (avp)
  - Wilson-Clover operator in Qlua for test and validation
  - Domain Wall operators
- HQL to HYPRE interface (Chris)
- HYPRE changes (Rob)
  - complex numbers.
  - arbitrary, user-defined dimension.

# Multigrid

# Multigrid

- Collaboration with FASTMath to access Hypre algorithms from Qlua

# Multigrid

- Collaboration with FASTMath to access Hypre algorithms from Qlua
- Exploring the universe of coarsening for MDWF

# Multigrid

- Collaboration with FASTMath to access Hypre algorithms from Qlua
- Exploring the universe of coarsening for MDWF
- HMC plug-in

# Pointers

- https://usqcd.lns.mit.edu/
  - …/redmine/projects/qlua
  - …/w/index.php/QLUA_Tutorials:HDF5
  - …/w/index.php/QLUA_Tutorials:Eigenspace_deflation_interface
  - …/redmine/projects/qa0
- https://www.hdfgroup.org/HDF5/