

MILC with SciDAC C

Carleton DeTar
HackLatt 2008

SciDAC Software Team

ANL	James Osborn	MIT	Andrew Pochinsky
Arizona	Doug Toussaint		Joy Khoriaty
	Dru Renner	U North Carolina	Rob Fowler
BU	Rich Brower*		Ying Zhang*
	Mike Clark	JLab	Chip Watson*
BNL	Chulwoo Jung		Robert Edwards*
	Enno Scholz		Jie Chen
	Efstratios Efstathiadis		Balint Joo
Columbia U	Bob Mawhinney*	IIT	Xien-He Sun
DePaul U	Massimo DiPierro	Indiana	Steve Gottlieb*
FNAL	Don Holmgren*		Subhasish Basak
	Jim Simone	Utah	Carleton DeTar*
	Jim Kowalkowski		Tommy Burch
	Amitoj Singh	Vanderbilt	Ted Bapty

Application Codes:

[MILC](#) / [CPS](#) / [Chroma](#) / Roll Your Own

SciDAC-2 QCD API

Level 4

QCD Physics Toolbox

Shared Alg,Building Blocks, Visualization,Performance Tools

Workflow

and Data Analysis tools

Level 3

QOP (Optimized in asm)

Dirac Operator, Inverters, Force etc

Uniform User Env

Runtime, accounting, grid,

Level 2

QDP (QCD Data Parallel)

Lattice Wide Operations, Data shifts

QIO

Binary / [XML files & ILDG](#)

Level 1

QLA

(QCD Linear Algebra)

QMP

(QCD Message Passing)

QMC

(QCD Multi-core interface)



Nine Mile Canyon, Utah (ca 1200)

SciDAC/C Tutorial 2007

- James Osborn and Andrew Pochinski
<http://web.mit.edu/~bgl/scidac-2007/>

USQCD Software Download

- USQCD software page
<http://www.usqcd.org/software.html>

New: Installation qinstall

- `CVSROOT =`
`:pserver:anonymous@cvs.jlab.org:/group/lattice/cvsroot`
- `cvs -d $CVSROOT co -P qinstall`
- Profile for each processor type

```
qinstall p4 qmp 2.2.0  
qinstall p4 qio 2.3.4
```

Level 1

- QMP (message passing)
 - MPI
 - QCDOC
- QLA (single-processor linear algebra)
 - Some 25,000 Perl-generated routines in a couple dozen libraries

`QLA_F3_M_veq_M_times_M`

Level 2

- QIO (file I/O)
 - USQCD file formats
 - ILDG file formats
 - Create your own format
 - LIME (message encapsulation – tar-like)
- QDP (data parallel operations)

`QDP_F3_M_eq_sM_times_M`

Example of QDP coding

- Plaquette calculation

```

#include <qdp.h>
QLA_Real plaquette(QDP_ColorMatrix *link[])
{
    int mu, nu;
    QLA_Real plaq, total;
    QDP_ColorMatrix *tmp1, *tmp2, *tmp3, *tmp4;

    total = 0;

    tmp1 = QDP_create_M(); tmp2 = QDP_create_M();
    tmp3 = QDP_create_M(); tmp4 = QDP_create_M();
    for (mu = 0; mu < NDIM; mu++) {
        for (nu = mu + 1; nu < NDIM; nu++) {
            QDP_M_eq_sM(tmp1, link[nu], QDP_neighbor[mu], QDP_forward, QDP_all);
            QDP_M_eq_sM(tmp2, link[mu], QDP_neighbor[nu], QDP_forward, QDP_all);
            QDP_M_eq_Ma_times_M(tmp3, link[nu], link[mu], QDP_all);
            QDP_M_eq_M_times_M(tmp4, tmp3, tmp1, QDP_all);
            QDP_r_eq_re_M_dot_M(&plaq, tmp2, tmp4, QDP_all);
            total += plaq;
        }
    }
    QDP_destroy_M(tmp1); QDP_destroy_M(tmp2);
    QDP_destroy_M(tmp3); QDP_destroy_M(tmp4);
    return total;
}

```

color matrix field

subset

shifted field

QDP/C vs QDP++

- Procedure calls straightforward but cumbersome
- Optimization inherited from QLA
- Proliferation of precompiled procedures
- Expressions powerful but require learning
- Selective optimization
- Expression templates compiled for each instance in code

Level 3

- QOP (inverters, Dslash, fermion force)
 - QCDOC implementation
 - QDP implementation
- Can be implemented in assembly

QOP API: Asqtad solver example

```
QOP_F3_ColorVector *  
QOP_F3_create_V_from_raw(float *src, QOP_evenodd_t evenodd);
```

```
QOP_F3_FermionLinksAsqtad *  
QOP_F3_asqtad_create_L_from_raw(  
    float *fatlinks[],  
    float *longlinks[],  
    QOP_evenodd_t evenodd);
```

```
void  
QOP_F3_asqtad_invert(QOP_info_t *info,  
    QOP_F3_FermionLinksAsqtad *asqtad,  
    QOP_invert_arg_t *inv_arg,  
    QOP_resid_arg_t *res_arg,  
    float mass,  
    QOP_F3_ColorVector *out_pt,  
    QOP_F3_ColorVector *in_pt);
```

MILC Makefile Options for SciDAC

```
PRECISION = 1      or      2
```

```
WANTQOP = QDP      or      QCDOC  
WANTQDP = true  
WANTQIO = true  
WANTQMP = true
```

```
# Parallel version  
QMPPAR = ${SCIDAC}/qmp  
QIOPAR = $(SCIDAC)/qio  
QLA     = ${SCIDAC}/qla  
QDP     = ${SCIDAC}/qdp  
QOP     = ${SCIDAC}/qopqdp  
# Single processor version  
QMPSNG = ${SCIDAC}/qmp-single  
QIOSNG = $(SCIDAC)/qio-single
```

Summary

- SciDAC/C is supported by the MILC code
- Makefile options select the desired level of integration with SciDAC/C

Tutorial 2 goals

- Build MILC code with SciDAC support
- Create a gauge file in SciDAC format
- Learn how to use qinstall