SciDAC Software Infrastructure for Lattice Gauge Theory

DOE meeting on Strategic Plan --- April 15, 2002

Software Co-ordinating Committee

- Rich Brower --- Boston University
- Carleton DeTar --- University of Utah
- Robert Edwards --- Jefferson Laboratory
- Don Holmgren --- Fermi National Laboratory
- Bob Mawhinney --- Columbia University/BNL
- Celso Mendes --- University of Illinois
- Chip Watson --- Jefferson Laboratory

SciDAC Software Infrastructure Goals

- ***Create a unified programming environment that will enable the US lattice community to achieve very high efficiency on diverse multi-terascale hardware***

Major Software Tasks

I. QCD API and Code Library

II. Optimize Network Communication

III. Optimize Lattice QCD Kernels

IV. Application Porting and Optimization

V. Data Management and Documentation

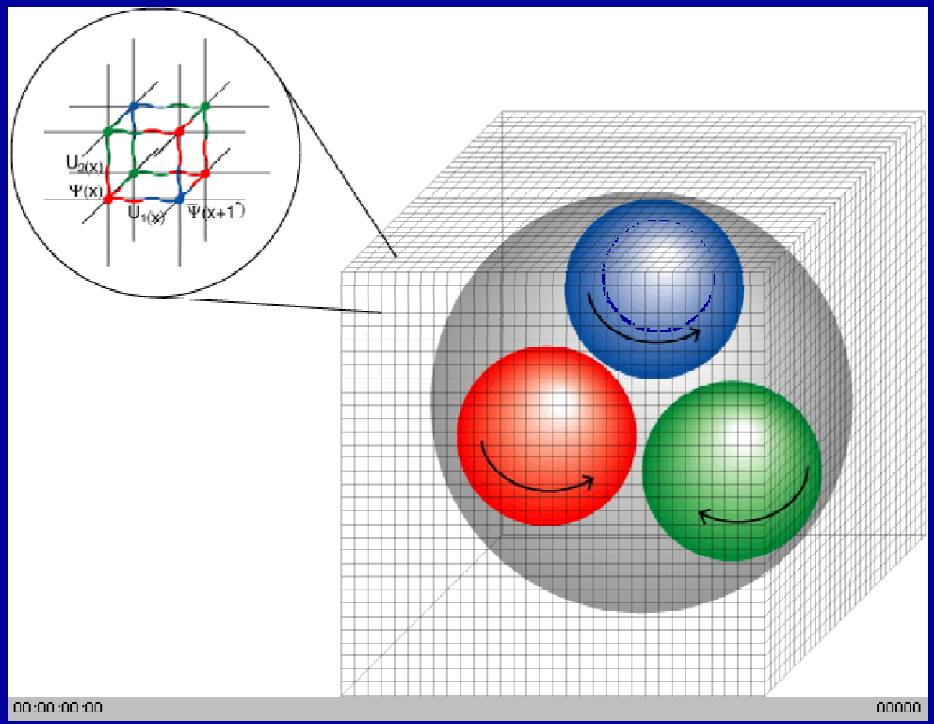
VI. Execution Environment

Participants in Software Development Project

Bob Mawhinney	Columbia	----	
Chulwoo Jung	BNL	100%	Sept 1, 2001
Chris Miller	BNL	100%	
Konstantin Petrov	BNL	100%	June 1, 2002
Don Holmgren	FNAL	40%	
Jim Simone	FNAL	35%	
Simon Epsteyn	FNAL	10%	
Amitoj Sing	FNAL	100%	Jan 22, 2002
Daniel A. Reed	Illinois	----	
Celso L. Mendes	Illinois	35%	Oct 1, 2001

Robert Edwards	Jlab	----	
Chip Watson	Jlab	33%	
Walt Akers	Jlab	100%	Jan 1, 2002
Jie Chen	Jlab	100%	Jan 1, 2002
Andrew Pochinsky	MIT	100%	
Richard Brower	BU	30%	
New Hire	BU	100%	(Oct 1, 2002)
Carleton DeTar	Utah	----	
James Osborn	Utah	50%	Sept 1, 2001
Doug Toussaint	Arizona	----	
Eric Gregory	Arizona	50%	Oct 15, 2001

Lattice QCD – extremely uniform



Dirac operator:

$$D\psi(x) = \sum_{\mu} \left(\partial_{\mu} + igA_{\mu}(x) \right) \psi(x)$$

Lattice Operator:

$$D\psi(x) = \frac{1}{2a} \sum_{\mu} \left[U(x)\psi(x+\hat{\mu}) - U^{\dagger}(x-\hat{\mu})\psi(x-\hat{\mu}) \right]$$

- Periodic or very simple boundary conditions
- SPMD: Identical sublattices per processor

QCD-API Level Structure

Level 3
Dirac Operators, CG Routines etc. C, C++, etc. (Organized by MILC or SZIN or CPS etc.)

QDP_XXX Level 2	
Data Parallel QCD Lattice Operations (overlapping Algebra and Messaging) A = SHIFT(B, mu) * C; Global sums, etc	
Lattice Wide Linear Algebra (No Communication) e.g. A = B * C	Lattice Wide Data Movement (Pure Communication, non-blocking) e.g Atemp = SHIFT(A, mu)

QLA_XXX Level 1	QMP_XXX
Single Site Linear Algebra API SU(3), gamma algebra etc.	Message Passing API (Know about mapping of Lattice onto Network Geometry)

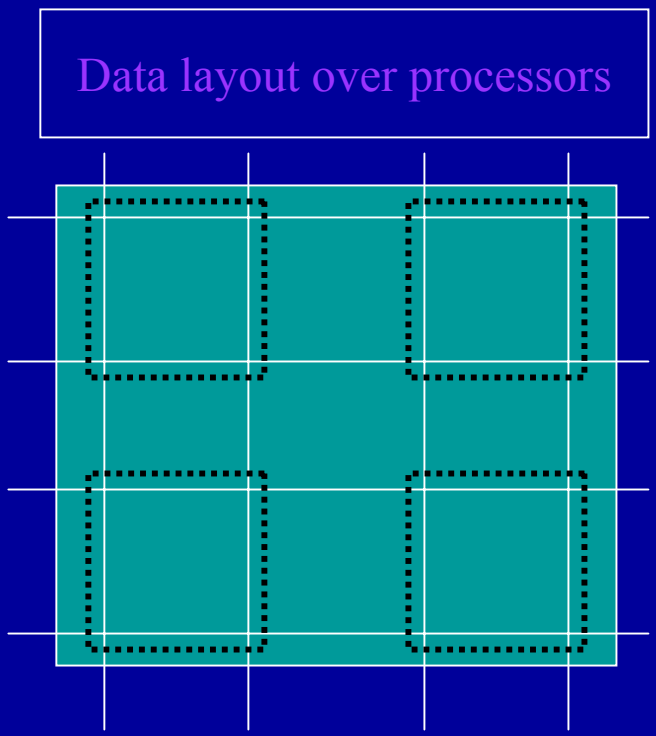
I. Design & Documentation of QCD-API

- Major Focus of Software Co-ordinating Committee
  - Working documents on http://physics.bu.edu/~brower/SciDAC
  - Published documents to appear on http://www.lqcd.org

- Design workshops: Jlab: Nov. 8-9, 2001, Feb 2, 2002
  - Next Workshop: MIT/BU: June, 2002 after Lattice 2002

- Goal:
  - C and C++ implementation for community review by Lattice 2002 in Boston, MA.
  - Foster "Linux style" contributions to level 3 API library functions.

Data Parallel paradigm on top of Message Passing



- Basic uniform operations across lattice: C(x) = A(x)*B(x)
- Map grid onto virtual machine grid.
- API should hide subgrid layout and subgrid faces communicated between nodes.
- Implement API without writing a compiler.

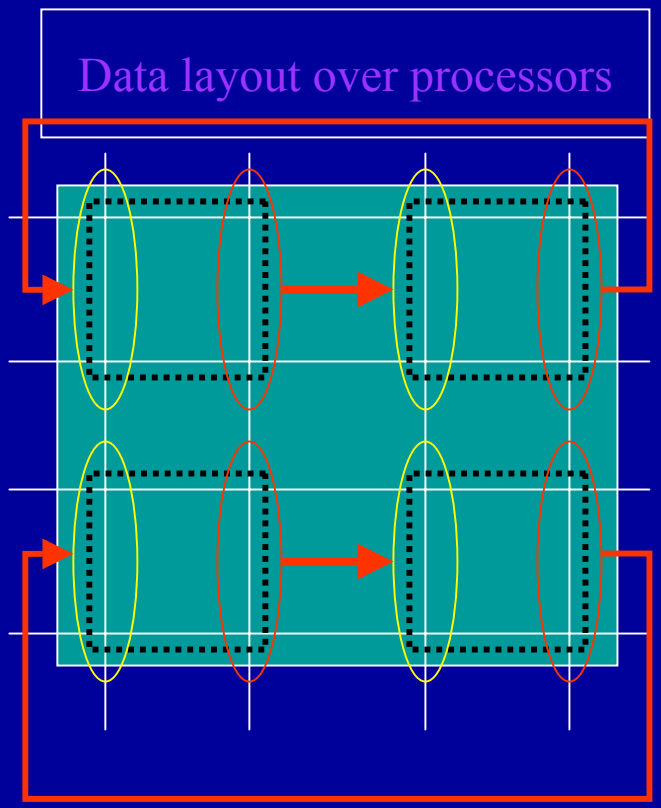
API Design Criteria

- Routines are extern C functions callable from C and Fortran: extern functions <==> C++ methods.
- Overlapping of computation and communications.
- Hide data layout: Constructor, destructors. Query routines to support limited number of deftypes.
- Support for multi-process or multi-threaded computations hidden from user control.
- Functions do not (by default) make conversions of arguments from one layout into another layout. An error is generated if arguments are in incompatible.

II. Level 1 MP-API implementation

- Definition of MP interface (Edwards, Watson)
  - Bindings for C, C++ and eventually Fortran.
  - see doc http://www.jlab.org/~watson/lqcd/MessageAPI.html
- Implementation of MP-API over MPI subset (Edwards)
- Implementation of C++ MP-API for QCDOC (Jung)
- Myrinet optimization using GM (Jie Chen)
- Port of MILC code to level 1 MP-API (DeTar, Osborn)

Performance Considerations for Level 2



- Overlapping communications and computations:
- C(x)=A(x)*shift(B,mu):
  - The face of a subgrid is sent non-blocking to a neighboring node, e.g. in the forward direction.
  - The neighboring node, in the backward direction, sends its face into a pre-allocated buffer.
  - While this is going on, the operation is performed on the interior sites.
  - A “wait” is issued and the operation is performed on the face.

Lazy Evaluation for Overlapping Comm/Comp

Consider the equation dest(x) = src1(x)*src2(x+nu) ; (for all x)

or decomposed as tmp(x) = src2(x+mu);
dest(x) = src1(x)*tmp(x)

Implementation 1: As two functions:

```
Shift(tmp, src2, mu,plus);
Multiply(dest, src1, tmp);
```

Implementation 2: Shift also return its result:

```
Multiply(dest, src1, Shift(src2, mu,plus));
```

Data Types

- Fields have various types (indices):

  Color: $U^{ij}(x)$, Spin: $\Gamma_{\alpha\beta}$, $\psi^{i}_{\alpha}(x)$, $Q^{ij}_{\alpha\beta}(x)$

- Index type (i.e the "fiber" over "base" lattice site)
  - Gauge : Product(Matrix(Nc),Scalar)
  - Dirac: Product(Vector(Nc),Vector(Ns))
  - Scalars: Scalar
  - Propagators: Product(Matrix(Nc),Matrix(Ns))?

- Support Red/Black sublattices & other subsets (Mask ?)

- Support compatible operations on types:

$U^{ij}(x) * \Gamma_{\alpha\beta} * \psi^{i}_{\alpha}(x)$ ⟹ Matrix(color)*Matrix(spin)*Vector(color,spin)

C Naming Convention for Level 2

- void QCDF_mult_**T3T1T2**_*op3*(**Type3** *r, const **Type1** *a,const**Type2***b)

- **T3**, **T1**, **T2** are short for the type **Type1, Type2** and **Type3 :**
  LatticeGaugeF, LatticeDiracFermionF,
  LatticeHalfFermionF, LatticePropagatorF

- *op3* are options like

nnr	r = a*b	nnn	r = -a*b
ncr	r = a*conj(b)	ncn	r = -a*conj(b)
cnr	r = conj(a)*b	cnn	r = -conj(a)*b
ccr	r = conj(a)*conj(b)	ccn	r = -conj(a)*conj(b)
nna	r = r + a*b	nns	r = r – a*b
nca	r = r + a*conj(b)	ncs	r = r – a*conj(b)
cna	r = r + conj(a)*b	cna	r = r – conj(a)*b
cca	r = r + conj(a)*conj(b)	ccs	r = r - conj(a)*conj(b)

Data Parallel Interface for Level 2

Unary operations: *operate on one source into a target*

```
Lattice_Field Shift(Lattice_field source, enum sign, int direction);
void Copy(Lattice_Field dest, Lattice_Field source, enum option);
void Trace(double dest, Lattice_Field source, enum option);
```

Binary operations: *operate on two sources into a target*

```
void Multiply(Lattice_Field dest, Lattice_Field src1, Lattice_Field src2, enum option);
void Compare(Lattice_Bool dest, Lattice_Field src1, Lattice_Field src2, enum compare_func);
```

Broadcasts: *broadcast throughout lattice*

```
void Fill(Lattice_Field dest, float val);
```

Reductions: *reduce through the lattice*

```
void Sum(double dest, Lattice_Field source);
```

III. Linear Algebra: QCD Kernels

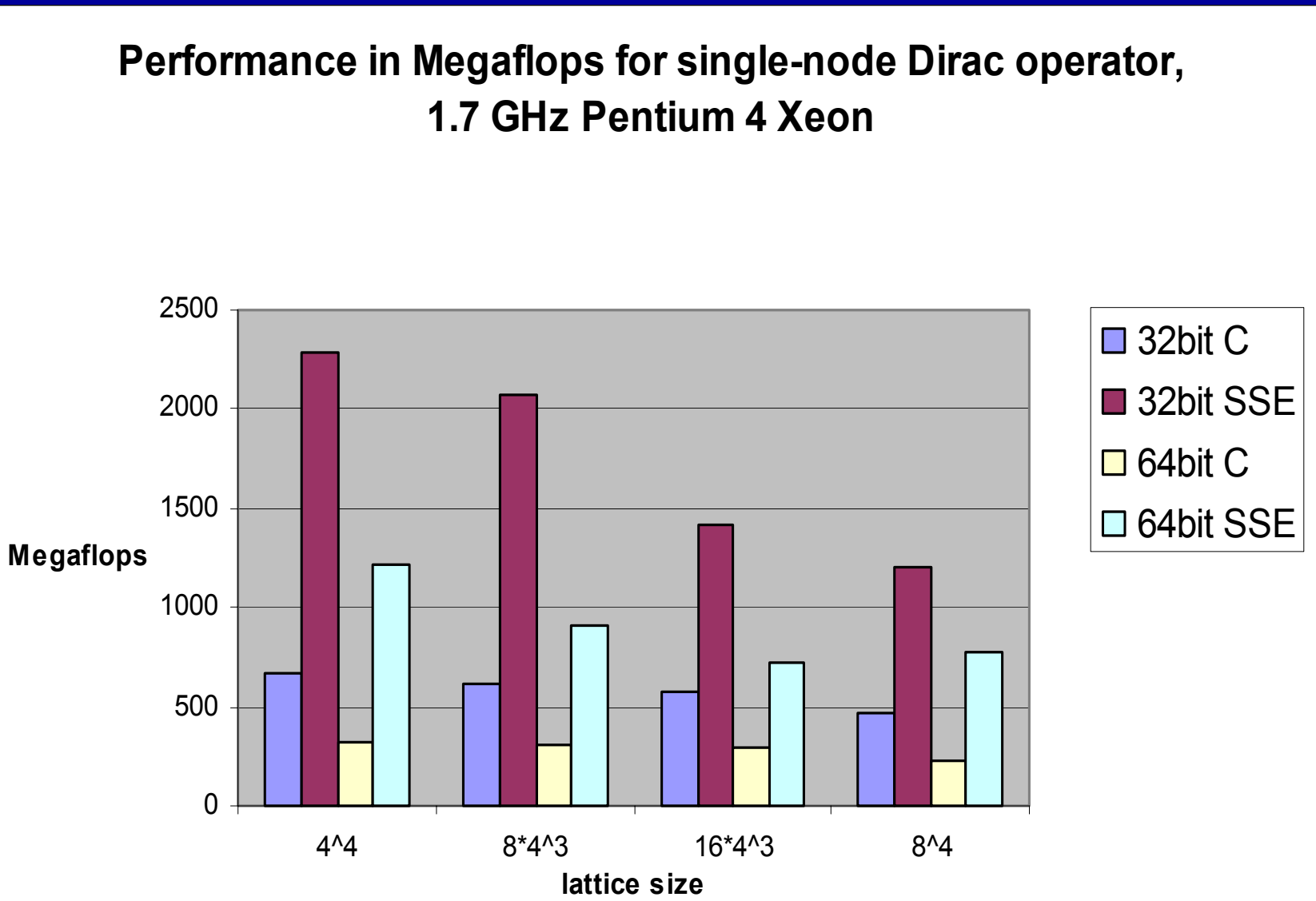
- First draft of Level 1 Linear Algebra API (DeTar, Edwards, Pochinksy)

http://www.jlab.org/~edwards/qcdapi/LinAlg1API_0_1.htm

- Vertical slice for QCD API (Pochinsky)
  - API conformant example of Dirac CG
  - MILC implementation (Osborn)

- Optimize on Pentium 4 SSE & SSE2 code:
  - for MILC (Holgrem, Simone, Gottlieb)
  - for SZIN l (Edwards, McClendon)

Single Node


Performance in Megaflops for single-node Dirac operator,
1.7 GHz Pentium 4 Xeon
Megaflops
2500
2000
1500
1000
500
0
4^4
8*4^3
16*4^3
8^4
lattice size
32bit C
32bit SSE
64bit C
64bit SSE

IV. Application Porting & Optimization

- MILC: (revision version 6_15oct01)
  - QCDOC ASIC simulation of MILC (Calin, Christan, Toussaint, Gregory)
  - Prefetching Strategies (Holgren, Simone, Gottlieb)

- SZIN: (new documentation and revision) (Edwards)
  - Implementation on top of QDP++ (Edwards, Pochinsky)
  - Goal: efficient code for P4 by Summer 2002

- CPS (Columbia Physics System)
  - Software Testing environment running on QCDSP (Miller)
  - Native OS & fabric for MP-API (Jung)

V. Data Archives and Data Grid

- File formats and header
  - Build on successful example of NERSC QCD archive
  - Extend to include lattice sets, propagators, etc.
- Consider XML for ascii headers
  - Control I/O for data files
  - Search user data using SQL to find locations.
- Lattice Portal
  - Replicate data (multi-site), global tree structure.
  - SQL-like data base for storing data and retrieving
- Web based computing
  - batch system and uniform scripting tool.

VI. Performance and Exec. Environment

- Performance Analysis Tool:
  - SvPABLO instrumentation of MILC (Celso)
  - Extension through PAPI interface to P4 architecture (Dongarra)
- FNAL Tools:
  - Trace Tools extension to Pentium 4 and instrumentation of MILC (Rechenmacher, Holmgen, Matsumura)
  - FNAL “rgang” (parallel command dispatcher)
  - FermiQCD (DiPierro)
- Cluster Tools: (Holmgren, Watson)

  Building, operations, monitoring, BIOS update, etc

SvPablo Instrumentation of MILC