SciDAC Software Infrastructure for Lattice Gauge Theory

Richard C. Brower
All Hands Meeting BNL, March 22-23, 2007


Code distribution see  http://www.usqcd.org/software.html
QUIZZ

THIS IS THE 50th ANNIVERSARY OF WHAT?
FORTRAN, the first high level programming language, was announced to the computing world by John Backus and his team from IBM at the Western Joint Computer Conference held in Los Angeles, California in February 1957.
John Backus' team in the late 1950s

Creating the IT Profession
An IBM 704 mainframe

Creating the IT Profession
In late 1953, John Backus sent a brief letter to his boss at IBM, asking that he be allowed to search for a "better way" of programming computers, with a project timescale of six months. He got the nod and began the research project that would eventually produce FORTRAN.

As John Backus says in the film, “project completion was always six months away”!

Creating the IT Profession
Frank Engel of Westinghouse, Pittsburg was concerned about the efficiency of the tape operations with the first FORTRAN compiler. He asked IBM if he could have a copy of the source code. They replied "IBM does not supply source code." So Frank worked his way through an octal dump of the compiler and optimised the tape operations. The improvement so impressed IBM that they asked for a copy of the code, to which Frank replied "Westinghouse does not supply source code."
**Major Participants in SciDAC Project**

<table>
<thead>
<tr>
<th>Institution</th>
<th>Participants</th>
<th>Institutions</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arizona</td>
<td>Doug Toussaint</td>
<td>MIT</td>
<td>Andrew Pochinsky</td>
</tr>
<tr>
<td></td>
<td>Dru Renner</td>
<td></td>
<td>Joy Khoriaty</td>
</tr>
<tr>
<td>BU</td>
<td>Rich Brower *</td>
<td>North Carolina</td>
<td>Rob Fowler</td>
</tr>
<tr>
<td></td>
<td>James Osborn</td>
<td></td>
<td>Ying Zhang *</td>
</tr>
<tr>
<td></td>
<td>Mike Clark</td>
<td>JLab</td>
<td>Chip Watson *</td>
</tr>
<tr>
<td>BNL</td>
<td>Chulwoo Jung</td>
<td></td>
<td>Robert Edwards *</td>
</tr>
<tr>
<td></td>
<td>Enno Schloz</td>
<td></td>
<td>Jie Chen</td>
</tr>
<tr>
<td></td>
<td>Efstratios Efstathiadis</td>
<td></td>
<td>Balint Joo</td>
</tr>
<tr>
<td>Columbia</td>
<td>Bob Mawhinney *</td>
<td>IIT</td>
<td>Xien-He Sun</td>
</tr>
<tr>
<td>DePaul</td>
<td>Massimo DiPierro</td>
<td>Indiana</td>
<td>Steve Gottlieb</td>
</tr>
<tr>
<td>FNAL</td>
<td>Don Holmgren *</td>
<td></td>
<td>Subhasish Basak</td>
</tr>
<tr>
<td></td>
<td>Jim Simone</td>
<td>Utah</td>
<td>Carleton DeTar *</td>
</tr>
<tr>
<td></td>
<td>Eric Neilsen</td>
<td></td>
<td>Ludmila Levkova</td>
</tr>
<tr>
<td></td>
<td>Amitoj Singh</td>
<td>Vanderbilt</td>
<td>Ted Bapty</td>
</tr>
</tbody>
</table>

* Software Committee: Participants funded in part by SciDAC grant
<table>
<thead>
<tr>
<th>Institutions</th>
<th>Oversight</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNL/Columbia</td>
<td>Mawhinney/ Chulwoo Jung</td>
</tr>
<tr>
<td>JLab</td>
<td>Edwards/Watson</td>
</tr>
<tr>
<td>FNAL/ITT/Vanderbuilt</td>
<td>Holmgren/Simone</td>
</tr>
<tr>
<td>BU/MIT</td>
<td>Brower/Pochinsky</td>
</tr>
<tr>
<td>DePaul/NorthCarolina</td>
<td>DiPierro/Zhang</td>
</tr>
<tr>
<td>Arizona/Indiana/Utah</td>
<td>DeTar/Gottlieb/Toussaint</td>
</tr>
</tbody>
</table>
SciDAC-1 QCD API

Level 3
- Optimized Dirac Operators, Inverters
  - Optimised for P4 and QCDOC
  - ILDG collab

Level 2
- QDP (QCD Data Parallel)
  - Lattice Wide Operations, Data shifts
- QIO
  - Binary/ XML Metadata Files

Level 1
- QMP (QCD Message Passing)
  - Exists in C/C++
- QLA (QCD Linear Algebra)

C/C++, implemented over MPI, native QCDOC, M-via GigE mesh
SciDAC-2 QCD API

**Application Codes:**
- QOP (Optimized in asm)
- Dirac Operator, Inverters, Force etc
- QDP (QCD Data Parallel)
  Lattice Wide Operations, Data shifts
- QMP (QCD Message Passing)
- QIO
  Binary / XML files & ILDG
- QMC
  (QCD Multi-core interface)

**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

**Level 2**
- QDP (QCD Data Parallel)
  Lattice Wide Operations, Data shifts

**Level 1**
- QLA
  (QCD Linear Algebra)
- QMP
  (QCD Message Passing)

SciDAC-1/SciDAC-2 = Gold/Blue

Application Codes:
- MILC / CPS / Chroma / RoleYourOwn
Some current activities & Priorities

- Fuller use of API in application code.

  Round table: Software vs software

- Porting API to new Machines

  BG/L & BG/P: QMP and QLA using XLC & Perl script

  Cray XT3 & XT4: Opteron, 32 bit SSE, etc.

- Common Runtime Env. “Practical Meta-facility”

  File transfer, Batch scripts, Compile targets
Workflow and Data Analysis

Automate campaign to combine lattices, propagators to extract physical parameters. (FNAL Jim Simone & ITT)

Tool Box (shared algorithms / building blocks)

RHMC, eigenvector solvers, etc
Visualization and Performance Analysis

Exploitation of Multi-core

Plans for a QMC API (JLab Jie Chen/ Edwards)
Status of QMP on BG/L

- based on QMP/MPI code base
- added `--with-qmp-comms-type=BGL` option
- native BG/L point-to-point (send/receive)
- uses MPI for everything else (collectives)
- requires barriers (MPI_Barrier) around some collectives (broadcast, binary_reduction)
- mostly done -- still needs cleanup & testing & (more)optimization
Performance of QMP on BG/L
(contiguous quad-aligned buffers)

Ping pong test

<table>
<thead>
<tr>
<th>bytes</th>
<th>round trip time / 2 (microseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e1</td>
<td>1000</td>
</tr>
<tr>
<td>1e2</td>
<td>1000</td>
</tr>
<tr>
<td>1e3</td>
<td>1000</td>
</tr>
<tr>
<td>1e4</td>
<td>1000</td>
</tr>
<tr>
<td>1e5</td>
<td>1000</td>
</tr>
<tr>
<td>1e6</td>
<td>1000</td>
</tr>
</tbody>
</table>

- 2 nodes-MPI
- 2 nodes-native
- 8 nodes-MPI
- 8 nodes-native
- 64 nodes-MPI
- 64 nodes-native
Status of QLA on BG/L

- previous version had a single 440 asm routine
- now has a 440d asm version of same routine
- development version now uses XLC v8 and C99 complex types (along with necessary alignment and disjoint hints) to make use of 440d
- has passed full testsuite running on BG/L
- BAGEL routines may still be useful

James Osborn, Joy Khoriaty & Andrew Pochinsky
Performance of QLA on BG/L (QOPQDP – asqtad inverter)
Performance of QLA on BG/L
(QOPQDQP – Wilson inverter)
Performance of QMP+QLA on BG/L
(QOPQDP – asqtad inverter)
Performance of QMP+QLA on BG/L (QOPQDP – Wilson inverter)

64 nodes

- old - float
- new QLA - float
- new QMP+QLA - float
- old - double
- new QLA - double
- new QMP+QLA - double
# Software Committee

- **Rich Brower** (chair)  
  brower@bu.edu
- **Carleton DeTar**  
  detar@physics.utah.edu
- **Robert Edwards**  
  edwards@jlab.org
- **Don Holmgren**  
  djholm@fnal.gov
- **Bob Mawhinney**  
  rdm@phys.columbia.edu
- **Chip Watson**  
  watsone@jlab.org
- **Ying Zhang**  
  zhang@cs.uiuc.edu
QLA on Opterons (kaon)

staggered matrix-vector product

- pion - C
- pion - SSE
- kaon - C
- kaon - SSE