Lattice QCD

James C. Osborn
Boston University

Chulwoo Jung
Brookhaven National Laboratory

for
USQCD

CScADS
July 23, 2007

Project Overview

USQCD (84+ people, 40+ places)			
LHPC (Jlab, MIT, BU, ...)	MILC/ FNAL (many U's)	RBC (Riken, BNL, Columbia)	smaller groups or independent

- common interest in simulations of strong nuclear force (QCD)
- coordinate efforts and share lattice configurations, analysis typically done separately
- separate code bases; working on common & portable API's for communication, etc. (SciDAC)
- support from NSF & DOE (nuclear & high energy physics)

Science Lesson

- lattice QCD is only available theoretical tool for first-principle calculations of many quantities (particle masses, thermodynamic properties, etc.)
- discretize QCD fields on a 4D lattice
- simulation typically in two steps
  - generate field configurations (lattices) via importance sampling
  - measure quantities across all generated configurations

Parallel Programming Model

- mostly C/C++ code with some assembly; communication predominantly 4D nearest-neighbor and global sum
- uses MPI for comms, or lower level interfaces on some platforms (QCDOC, BG/L) (QMP:SciDAC comms API)
- runs anywhere with a working C/C++ compiler and MPI; also QCDOC, Ethernet Mesh clusters (with port of QMP)
- now investigating use of OpenMP/pthreads/shmem for multicore chips
- planning port of QMP to BG/P

Computational Methods

- field configurations generated via Monte Carlo process (Hybrid Monte Carlo, MD)
- requires repeated solution of large sparse system (Dirac matrix)
- typically use CGNE or sometimes BiCGStab
- solving for multiple sources becoming more important
- low eigenmode projection has been used in some cases
- investigating use of multigrid methods

I/O Patterns and Strategy

- configuration generation
  - write one file <10 GB every few hours
- analysis
  - involves many solutions of Dirac matrix
  - sometimes want to save solutions (propagator) for later use
  - each propagator is 4 times larger than configuration
- can either stream through node 0 to single file or have multiple files with one file per group of processors

Visualization and Analysis

- mostly generate data for 2D plots
- some more advanced visualization has been used in past using density or isosurface plots of 3D slices of 4D quantities
  - energy density (action)
  - topological charge
- SciDAC project to develop tools suited for 4D lattice QCD data

Performance

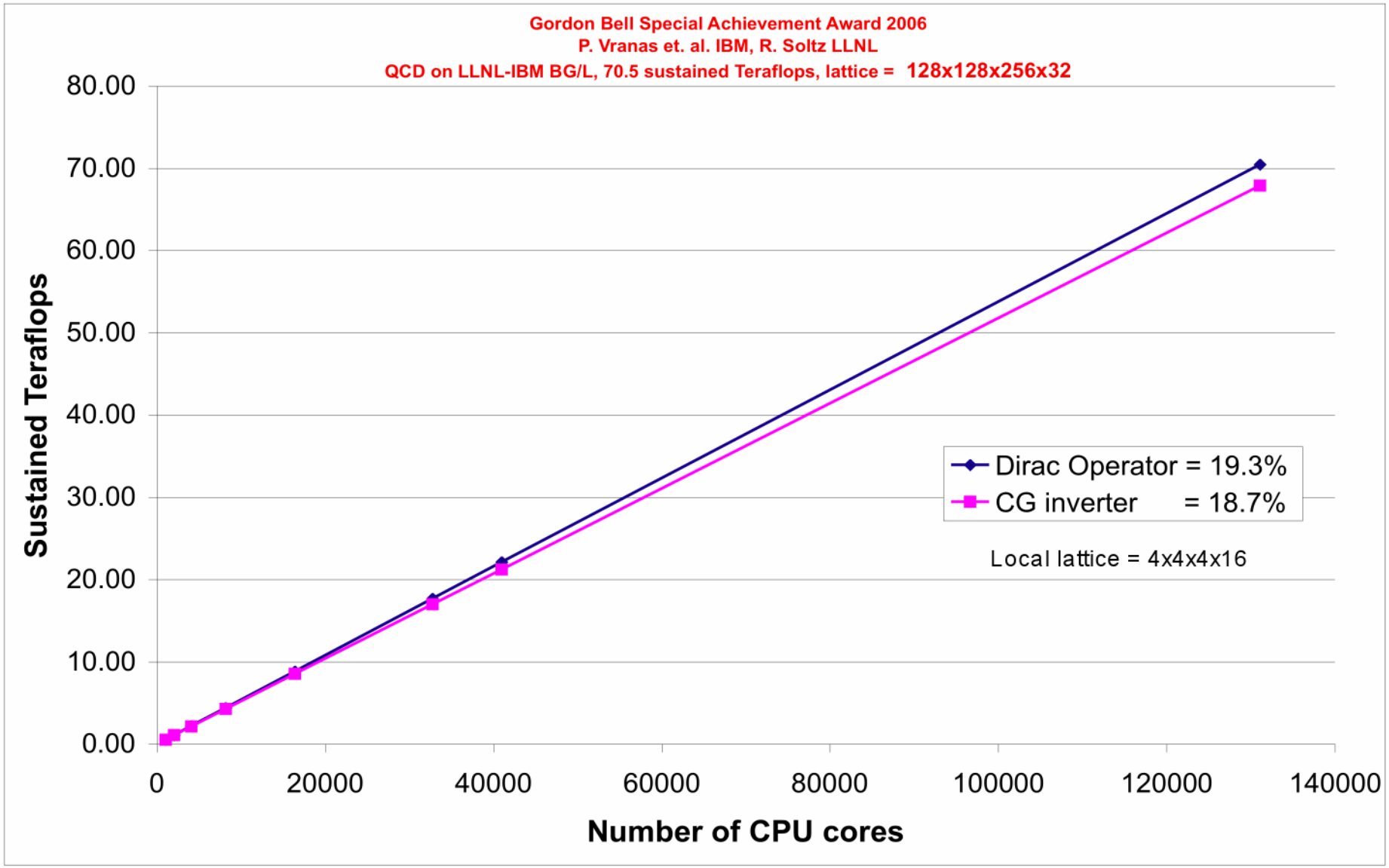
- mostly rely on built in timers and flop counts for core routines (Dirac matrix, etc.)
- sparse linear algebra – usually memory and/or communications bandwidth limited
- maps very nicely onto 4D torus, may take some performance hit for other topologies
- performance analysis tools can be useful, as long as it does not affect the performance itself
- SciDAC project to investigate performance in more detail using tools like SvPablo

Tools

- generally lattice QCD has been run as “early users” - often running on machines before sophisticated tools becomes available
- sometimes even used to test hardware (QCDOC)
- codes often have built in checks of results (solvers, etc.) or periodically rerun from last saved configuration as a check

Status and Scalability

- weak scaling is usually very good (depending on network architecture)
- strong scaling is also good (sometimes even see super-scaling from better cache/EDRAM utilization) up to a point (around 4^4 local sites) – network latency usually an issue


Gordon Bell Special Achievement Award 2006
P. Vranas et. al. IBM, R. Soltz LLNL
QCD on LLNL-IBM BG/L, 70.5 sustained Teraflops, lattice = 128x128x256x32
Sustained Teraflops
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
0
20000
40000
60000
80000
100000
120000
140000
Number of CPU cores
Dirac Operator = 19.3%
CG inverter = 18.7%
Local lattice = 4x4x4x16

Roadmap

- bigger lattices & lighter quark masses (worse conditioned Dirac matrix)
  - want 84^3x144 lattice (~250M dim. matrix)
- there have been major progress in configuration generation algorithms -- measurements and I/O becoming a bigger issue
- in constant search for more computing power
- ongoing efforts for improved algorithms and optimized libraries