Performance of machines for lattice QCD simulations

Tilo Wettig

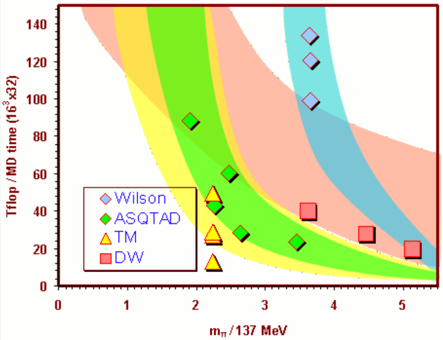
Institute for Theoretical Physics
University of Regensburg

Lattice 2005, 30 July 05

Outline

1. Introduction, definitions, terminology
2. Three choices
   - Commercial machines
   - PC clusters
   - Custom-designed machines
3. Speculations on future machines
4. Conclusions

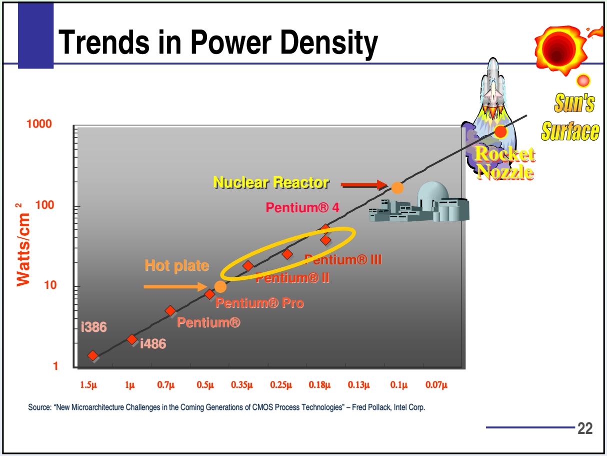
The need for speed



Tony Kennedy,
Lattice 2004

the computational needs of large-scale lattice QCD simulations can only be satisfied by massively parallel machines

Parallelism is inescapable



Source: "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp.

- power density/heat dissipation is a major issue in the industry
- clock frequency of single chips is limited

→ trend towards on-chip parallelization

Lattice QCD parallelization and design goals

- Lattice QCD is a relatively easy problem to parallelize
  - regular hypercubic grid
  - simple boundary conditions
  - uniform and predictable communication patterns

  $\longrightarrow$ divide global volume into identical local volumes (SPMD)
  e.g., $V_{\text{global}} = 32^3 \times 64$ on 8192 processors $\longrightarrow V_{\text{local}} = 4^4$

- Main workhorse in dynamical simulations: conjugate gradient algorithm
  two main ingredients that should perform well on a parallel machine
  - matrix-vector multiplication
  - global sum

- Everything else being equal, the number to maximize is

  science per € $\sim$ sustained MFlop/s per €

Peak vs sustained performance

- peak performance of a single processor

  theoretical # of Flops per clock cycle × clock frequency

  e.g., 3 GHz, 2 Flops per cycle → 6 GFlop/s peak

- sustained performance

  average # of Flops executed per clock cycle × clock frequency

  (for a parallel machine, multiply both by # of processors)

- within a given budget, we thus have four control parameters

  1. clock frequency
  2. theoretical # of Flops per cycle (e.g., vector instructions)
  3. # of processors
  4. percentage of peak sustainable (depends on # of processors)

  (depending on your institution, power/cooling/space will also be factors)

Sustained performance

percentage of peak that can be sustained depends on several factors, e.g.,

- imbalance of multiply/add in algorithm
- stalls due to memory access (if not cache-resident)
- stalls due to communication between processors
- software overhead (OS, communication calls)

→ design/select hard- and software that minimizes the dead time

Scalability

Scalability = sustained performance (in %) vs # of processors

- weak scaling: keep local volume fixed and increase global volume with # of processors
- strong scaling: keep physical problem size (global volume) fixed and decrease local volume with # of processors

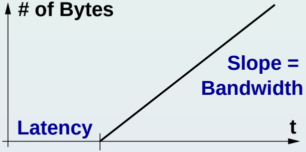
We are mainly interested in strong (= hard) scaling:
want to solve fixed physical problem in shortest possible wall-clock time

V_{local} becomes small → two competing effects

- good: data might fit into on-chip memory
- bad: surface-to-volume ratio becomes large
  → more communication per unit of computation
  can be evaded by communication latency hiding

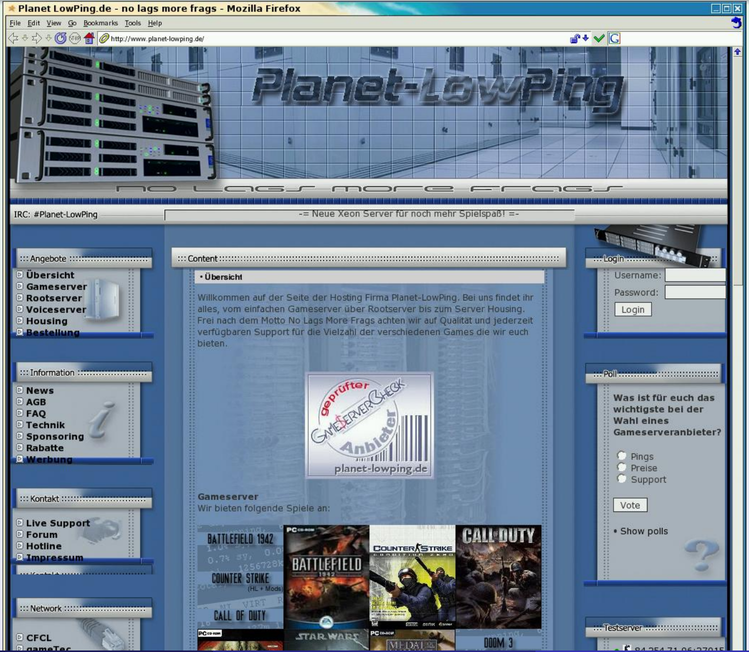
Bandwidth and Latency

- for both memory access and communication, the two main parameters are bandwidth and latency
- consider a data transfer from A to B, e.g.,
  - A = memory, B = processor
  - A and B = processors on the network



- often people only care about bandwidth, but for strong scaling, latency is the dominating factor (small packets)
- this is just like your DSL connection:
  - bandwidth is important for large downloads (e.g., latest Linux distro)
  - ping times are important for online gaming

www.planet-lowping.de — no lags more frags



Planet LowPing.de - no lags more frags - Mozilla Firefox

File Edit View Go Bookmarks Tools Help
http://www.planet-lowping.de/
Planet-LowPing
no lags more frags

IRC: #Planet-LowPing

-= Neue Xeon Server für noch mehr Spielspaß! =-

::: Angebote :::
- Übersicht
- Gameserver
- Rootserver
- Voiceserver
- Housing
- Bestellung

::: Information :::
- News
- AGB
- FAQ
- Technik
- Sponsoring
- Rabatte
- Werbung

::: Kontakt :::
- Live Support
- Forum
- Hotline
- Impressum

::: Network :::
- CFCL
- gameTec

::: Content :::

• Übersicht

Willkommen auf der Seite der Hosting Firma Planet-LowPing. Bei uns findet ihr alles, vom einfachen Gameserver über Rootserver bis zum Server Housing. Frei nach dem Motto No Lags More Frags achten wir auf Qualität und jederzeit verfügbaren Support für die Vielzahl der verschiedenen Games die wir euch bieten.

Gameserver
Wir bieten folgende Spiele an:

::: Login :::
Username:
Password:
Login

::: Poll :::
Was ist für euch das wichtigste bei der Wahl eines Gameserveranbieter?
○ Pings
○ Preise
○ Support
Vote
• Show polls

::: Testserver :::

ping-ping & ping-pong

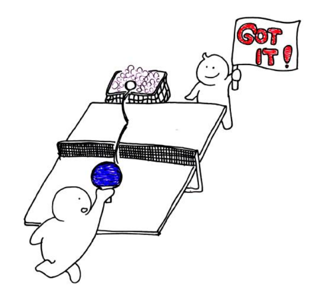
Bandwidth and latency can be measured in ping-ping and ping-pong benchmarks:

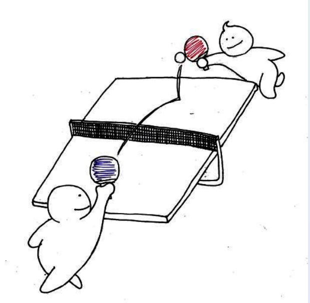
- ping-ping: unidirectional send

  a series of fixed-size packets is sent from A to B and from B to A

- ping-pong: bidirectional send

  a single packet is bounced back and forth between A and B





Capability and capacity machines

Roughly speaking:

- Capability = ability of a machine to finish a given (difficult) calculation in a certain amount of time
- Capacity = ability of a machine to carry out a given workload (typically many jobs) in a certain amount of time

In lattice QCD, both kinds are needed

- capability machines for generation of configurations (long Markov chains, small quark masses)
- capacity machines for analysis or scans of parameter space

Design parameters I: Hardware

- Processor
  - peak performance
  - amount of cache / on-chip memory
  - interfaces to memory and network
  - availability and quality of compilers
- Memory
  - latency and bandwidth (balanced with QCD requirements)
  - accessibility (shared vs distributed)
  - cache coherence
- Network
  - latency and bandwidth (balanced with QCD requirements)
  - topology (switched vs mesh)
  - DMA capabilities, hardware acceleration for typical operations
  - I/O performance
- price / power / cooling / space / packaging density

Design parameters II: Software

- Operating system
  - should provide all necessary services without hindering performance
  - ideally single user, single job
- Compilers
  - should produce correct and efficient code
  - should be free and widely available (Gnu tools)
  - assembler generator (BAGEL)
- Application code
  - code system should be easy to understand, easy to use, easy to extend
  - high performance → optimized kernels
  - low-level libraries for communication calls (provided by vendor or written by developers)
  - exemplary: USQCD/SciDAC and collaborators
    (QDP, QLA, QIO, QMP, Chroma)

Above all:

Balanced design (no bottlenecks)

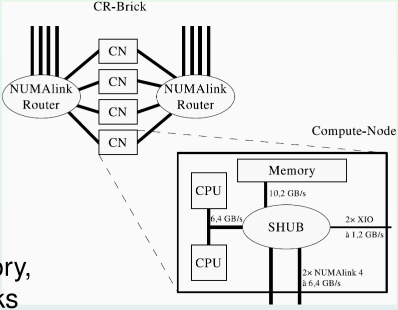
Three choices

1. commercial supercomputers (IBM, Cray, SGI, Hitachi, . . .)
   - suitable for general applications
   - typically not optimized for a particular problem
   - rather expensive
2. PC clusters
   - suitable for general applications
   - cheaper than commercial machines
   - communication latency typically rather high
     → strong scaling beyond $\mathcal{O}(100)$ nodes is a challenge
3. custom-designed machines (apeNEXT, QCDOC)
   - optimized with lattice QCD in mind → best scalability
   - best price-performance ratio, but PC clusters are close
   - high performance not guaranteed for non-QCD applications

Commercial supercomputers

- large computing centers like to buy commercial machines
- a number of vendors (IBM, SGI, Cray, Hitachi, NEC, Fujitsu, HP, Dell)
- typically capacity machines (clusters of SMPs)
- users don't have complete control over machine
  - only get fraction of the time
  - hard to get large partitions
  - cannot use privileged instructions (TLB)
  - administrative overhead
- will concentrate on two machines:
  1. SGI Altix (LRZ Munich)
     - 33 TFlop/s peak 2006-07
     - 69 TFlop/s peak 2007-10
  2. BlueGene/L — capability machine
     - 11.2/5.6 TFlop/s peak at Jülich
     - 5.6/2.8 TFlop/s peak each at Edinburgh, BU, MIT

SGI Altix



- based on Itanium-2 processor
- compute node: 2 CPUs, 8 GB memory, S-HUB, ccNUMA links
- connected by fat tree (shmem up to 512 CPUs)
- 3.2 μs SGI-MPT latency
- weak scaling results for Wilson-Dslash ($V_{\text{local}} = 4^4$, fits in L3 cache)

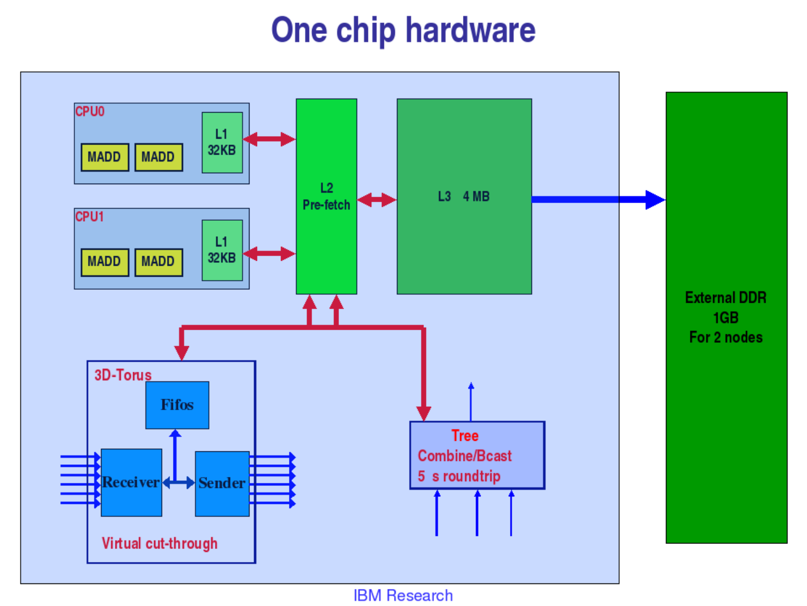
# CPUs	V_{global}	sustained perf.
8	$8^3 \times 4$	31%
16	8^4	26%
32	$8^3 \times 16$	30%
64	$8^3 \times 32$	28%

source:
Thomas Streuer

BlueGene/L overview

- currently #1 and #2 on the Top 500 list
  (183 TFlop/s peak at LLNL, 115 TFlop/s peak at IBM Watson)
- grew out of the QCDOC project
- system-on-a-chip design (ASIC)
  - 2 PowerPC 440 cores and 4 FPUs at 700 MHz → 5.6 GFlop/s peak
  - 32+32 kB L1 cache (I/D) per core, 2 kB L2 cache per core (prefetch), 4 MB shared L3 cache
- distributed memory (512 MB DDR per chip)
- network:
  - 3-d torus with nearest-neighbor links and virtual cut-through routing
  - global tree network for global operations
  - no DMA
- two modes of operation:
  - co-processor mode: one CPU for computation, one for communication
    → peak performance 2.8 GFlop/s per chip
  - virtual-node mode: both CPUs for computation and communication
    → 5.6 GFlop/s, but communication cannot overlap with computation

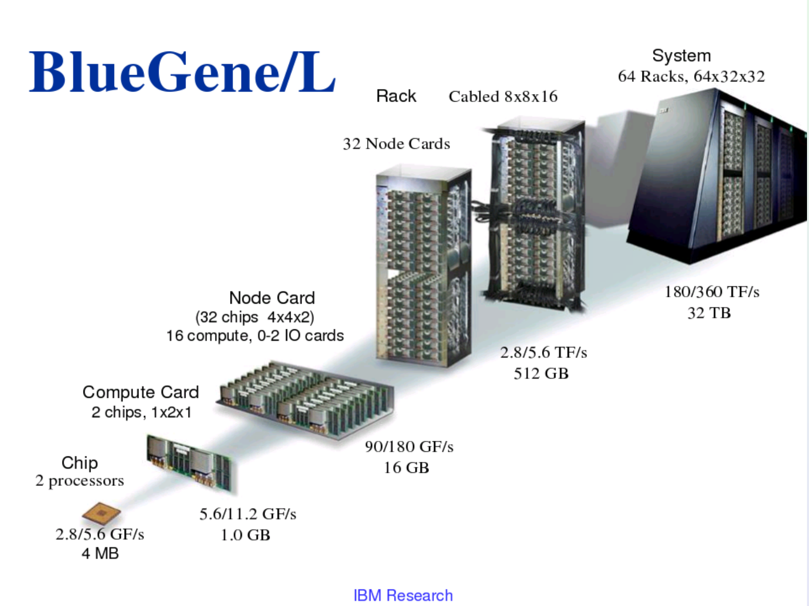
BlueGene/L ASIC



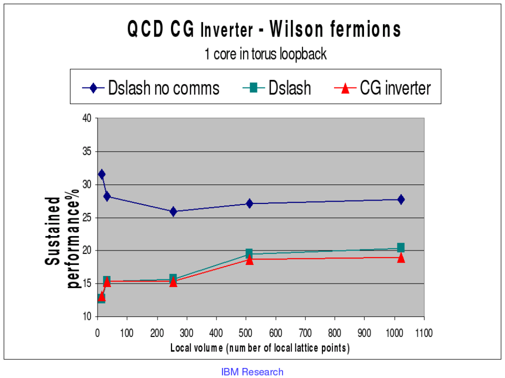
One chip hardware

IBM Research

BlueGene/L System



BlueGene/L strong scaling



Source:
Pavlos Vranas

- virtual-node mode (V_{local} refers to one core)
- one chip only, but using torus network (loopback)
- Dslash hand-coded in assembler
  network communications coded specifically for QCD
  L1 attributes set by hand [not (yet) part of standard OS]

PC Clusters

- high-volume market
  - many choices
  - low cost
  - increasingly driven by gaming industry (vector extensions)
- price-performance ratio competitive with custom-designed machines
- sensible choice for many groups (lots of clusters on Top 500 list)
- very much a moving target!
  - by the time you've done your benchmarks, new hardware is on the market
  - benchmarks often hard to compare because details matter

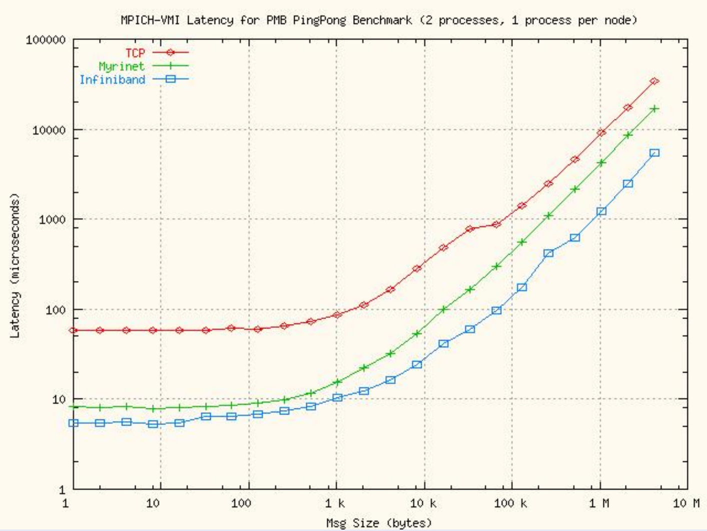
for more detailed information:

- poster by Don Holmgren
- http://online.kitp.ucsb.edu/online/lattice_c05/edwards (Robert Edwards)
- http://lqcd.fnal.gov/allhands_holmgren.pdf (Don Holmgren)

PC cluster design considerations

- Hardware
  - CPU: Pentium 4, Xeon, Opteron, G5, Itanium
    (FSB, memory controller on or off-chip, HT, HT, . . .)
  - Memory: DDR, DDR2, Rambus
  - Network: Gig-E, Myrinet, Infiniband, Quadrics
    topology (switched vs mesh)
  - Motherboard: PCI-X, PCI Express, chipsets
- Software
  - high-performance kernels use SSE instructions
  - efficient implementation of communication calls essential
  - USQCD/SciDAC software runs on clusters and QCDOC
    (low-level routines invisible to user)
- single-node performance usually very good
  (P4 most cost-effective right now, Opteron likely to take over)
  main challenge is network performance (both latency and bandwidth)

NCSA latency benchmarks



dual 3 GHz Xeon, 64-bit PCI-X

PC cluster networks

- switched clusters

	bandwidth	latency	cost
Gig-E	modest	high	low
Myrinet	good	low	moderate
Infiniband	good	low	moderate
Quadrics	good	very low	very high

- Gig-E meshes
  - no need for switches
  - high aggregate bandwidth
  - high latency ($15 \sim 25\mu s$) is the main problem
- very important: lean communication libraries to decrease latency
  - TCP/IP has too much overhead
  - QMP over M-VIA for Gig-E (latency $12.5\mu s$)
  - QMP over MPI for Myrinet (latency $10 \to 5\mu s$)
  - QMP over MPI/VAPI for Infiniband (latency $8 \to 3.5\mu s$)
- future: Infinipath/Hypertransport (Latency $0.8\ \mu s$)

Large PC cluster installations

existing lattice QCD clusters with more than 1 TFlop/s peak

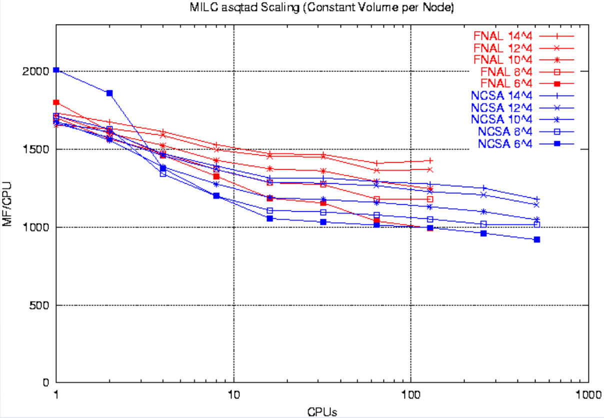
	CPU	Network	Peak (TFlop/s)	Name
Wuppertal	1024 Opteron	Gig-E (2d)	3.7	ALICEnext
JLAB	384 Xeon	Gig-E (5d)	2.2	4G
JLAB	256 Xeon	Gig-E (3d)	1.4	3G
Fermilab	260 (520) P4	Infiniband	1.7 (3.4)	Pion
Fermilab	256 Xeon	Myrinet	1.2	W

NB:

- peak numbers are double precision
- all benchmarks are single precision

ALICEnext at Wuppertal

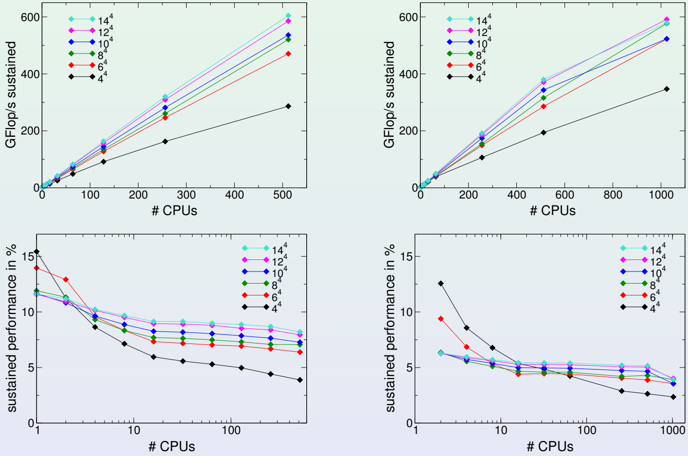
PC cluster scaling, asqtad inverter



- blue (t2): 3.6 GHz Xeon, Infiniband, PCI-X, MILC v6
- red (pion): 3.2 GHz Pentium 640, Infiniband, PCI-E, MILC using QDP optimized code by James Osborn

Cluster scalability: Dual Xeon 3.6 GHz, Infiniband, VMI

asqtad conjugate gradient (source: Steve Gottlieb)

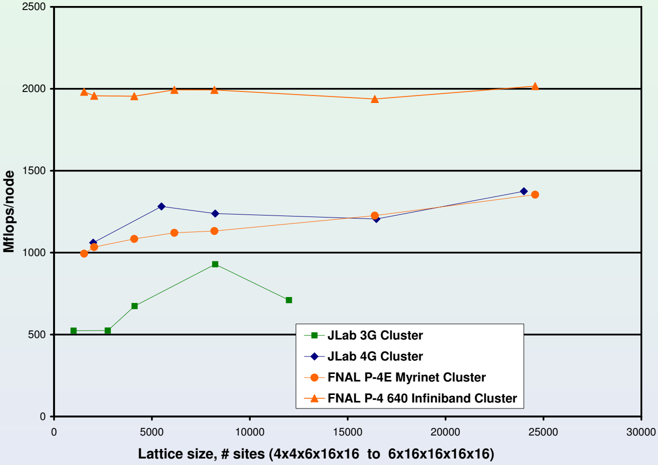


single CPU

both CPUs

second Xeon essentially useless here (memory bottleneck)
Opteron does not have this problem

PC cluster weak scaling, DWF inverter



peak per node:

3G:	5300
4G:	5600
P-4E:	4800
P-4 640:	6400

- inverter in assembler (Andrew Pochinsky)

PC cluster price and performance

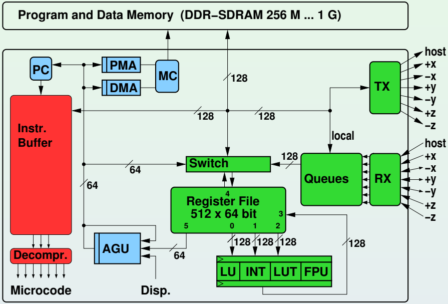
- 384-node cluster at JLAB (2.8 GHz Xeon, 800 MHz FSB, 3d Gig-E mesh) currently sustains 650 GFlop/s (DWF inverter, $V_{\text{local}} = 8^4 \times 16$)
  $\rightarrow$ \$1.10 per sustained MFlop/s in single precision
  (twice that for double precision)
- in the future, will be able to sustain $1 \sim 2$ TFlop/s on $\mathcal{O}(1000)$ nodes with price-performance ratio of about \$1 per sustained MFlop/s (single precision)
- 5% of cost per year for power and cooling

Custom-designed machines

- two machines: apeNEXT and QCDOC
- hardware optimized for typical lattice QCD algorithms
  → superior scalability
- custom OS
  standard compilers + assembler kernels
- developed by small collaborations
- clearly capability machines
  → workhorses for gauge field generation (with small quark masses)

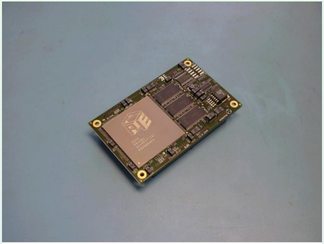
NB: all of the following benchmarks are double precision

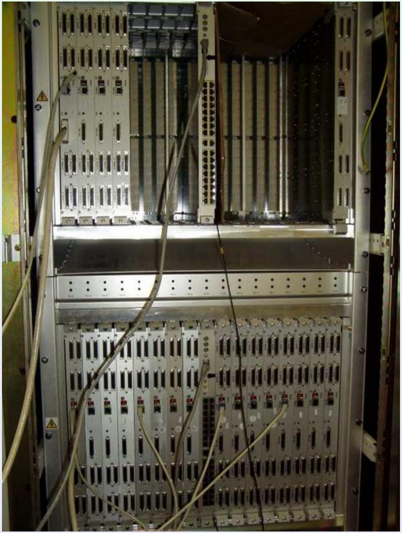
apeNEXT overview



- successor to APEmille
- collaboration of INFN/DESY/Orsay
- custom-designed processor (J&T)
- 8 Flops per cycle (complex $a \times b + c$) at 160 MHz → 1.3 GFlop/s peak
- 4 kB on-chip register file
- memory controller and communications hardware on chip
- 3-d torus network, DMA

apeNEXT system





apeNEXT performance

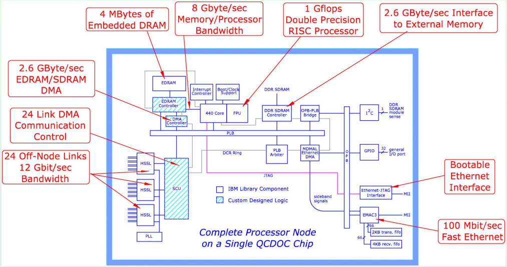
- single node performance bounds:
  - 54% for hand-coded Wilson-Dslash
  - 37% for TAO-based Clover-CG (to be optimized)
- ping-pong latency $\gtrsim$ 500 ns
  (can be hidden, except for global sums)
- global sum takes $N_x + N_y + N_z - 3$ steps of $\sim$ 60 cycles each on an $N_x \times N_y \times N_z$ processor mesh
  e.g., 11 μs on 1024 nodes
- no strong-scaling numbers yet
  expect delay of 4% due to communication overhead on $V_{\text{local}} = 2^3 \times 16$
  → close to ideal scaling

source: Hubert Simma, Lele Tripiccione

apeNEXT status

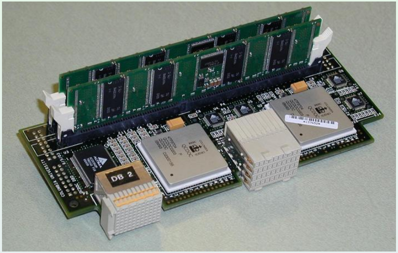
- 512 node prototype rack running stable
- version B of chip produced and tested (aiming at 160 MHz)
- TAO and C compilers stable (ongoing work to improve code-efficiency)
- physics production codes running with almost no modifications w.r.t. APEmille, but further optimization needed to reach efficiency of benchmark kernels
- planned installations:
  (1 rack = 512 nodes = 0.66 TFlop/s peak at 160 MHz)
  - 12 racks INFN
  - 6 racks Bielefeld
  - 3 racks DESY
  - 1 rack Orsay
- price is €0.60 per peak MFlop/s

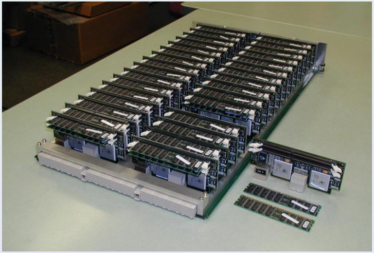
QCDOC overview



- successor to QCDSP
- collaboration of Columbia/UKQCD/RBRC/IBM
- custom-designed ASIC
- PowerPC 440 core + 64-bit FPU
  2 Flops per cycle at 400 MHz → 0.8 GFlop/s peak
- 4 MB on-chip memory
- memory controller and communications hardware on chip
- 6-d torus network, DMA

QCDOC system





QCDOC at BNL

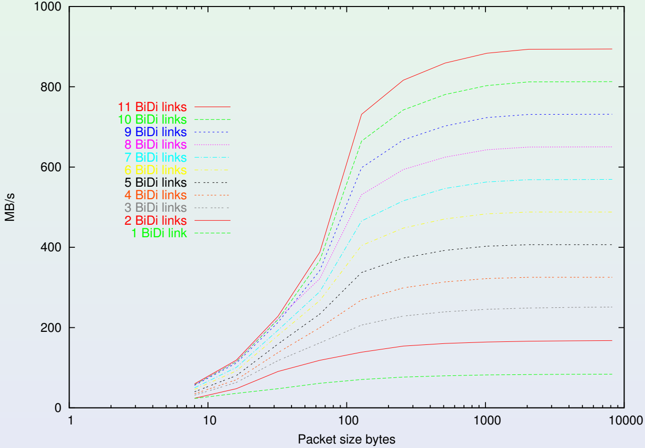
QCDOC installations

UKQCD	14,720 nodes
DOE	14,140 nodes
RIKEN-BNL	13,308 nodes
Columbia	2,432 nodes
Regensburg	448 nodes

- 12 racks = 12,288 nodes = 10 TFlop/s peak at 400 MHz
- price is \$0.45 per peak MFlop/s
- $\lesssim$ 2% of cost per year for power and cooling

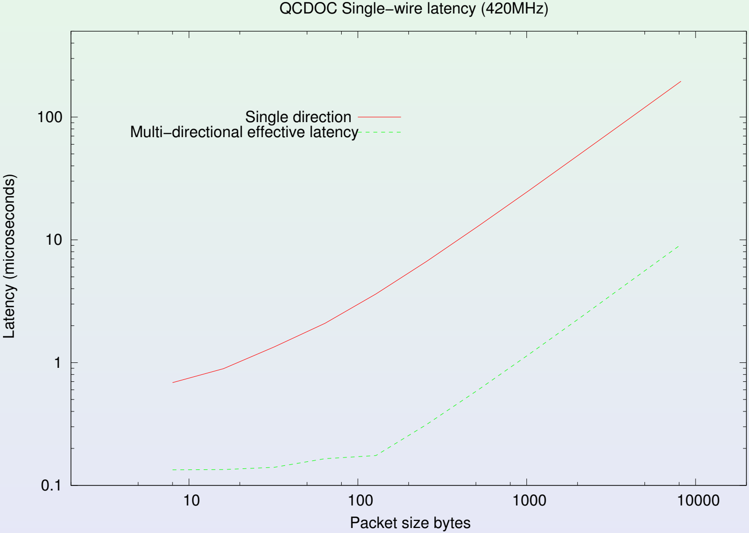
QCDOC bandwidth

(all benchmarks by Peter Boyle)

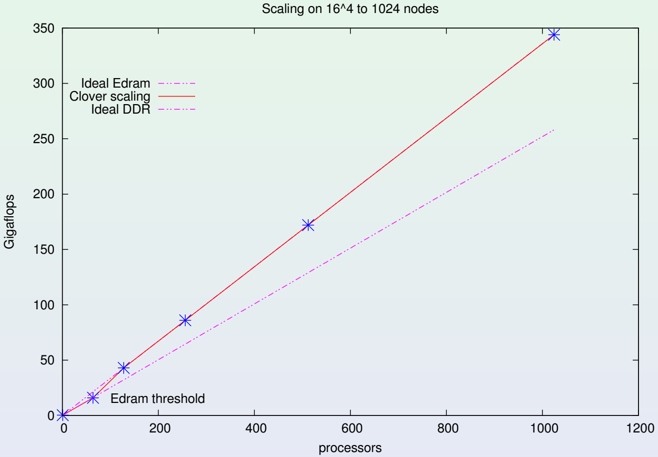


- multi-link bandwidth as good as memory bandwidth
- single link obtains 50% max bandwidth on 32-byte packets

QCDOC latency



QCDOC strong scaling



- $V_{\text{global}} = 16^4$ on up to 1024 nodes (equivalent to 32^4 on 16k nodes)
- corresponds to $V_{\text{local}} = 2^2 4^2$ on 1024 nodes

QCDOC application code performance

benchmarks presented at SC 2004
various discretizations, $V_{\text{local}} = 4^4$

Action	Nodes	Sparse matrix	CG performance
Wilson	512	44%	39%
Asqtad	128	42%	40%
DWF	512	46%	42%
Clover	512	54%	47%

(optimized code by Peter Boyle and Chulwoo Jung)

QCDOC application code performance

could not get further benchmarks because QCDOC users too busy with large jobs, e.g.,

- 3 RHMC jobs on 4096 nodes each (UK and US, different masses)
- problem doesn't fit in EDRAM
  - local volume is relatively large ($6^3 \times 2 \times 8$)
  - linear algebra required for multi-shift solver is running from DDR

  $\rightarrow$ sustained performance 32% ($>$ 1 TFlop/s sustained)
- 35% for 2-flavor DWF conjugate gradient (part of RHMC) should go up to 40% when running from EDRAM
- superlinear scaling observed when going from 1024 to 4096 nodes at fixed V_{global}
  - larger portion of problem moves into EDRAM
  - no noticeable degradation from communication overhead

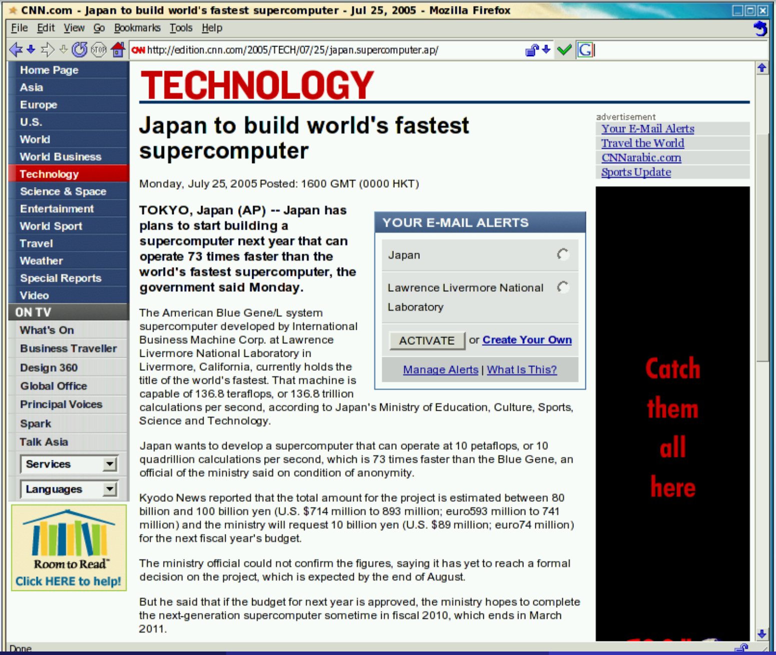
Speculations on future machines

- clear trend towards multi-core chips
- on-chip parallelization necessary for several reasons
  - not enough memory/network bandwidth for independent jobs on the cores
  - not enough memory per chip to run independent jobs on the cores (e.g., 32 cores with 1 GB memory each would require 32 GB per chip)
- automatic parallelization in hardware likely to remain a dream
- programming models likely to change
  - more fine-grained parallelism on chip
  - $V_{\text{local}} < 1$ per core !
  - mixture of pthreads, OpenMP and MPI?
- lean software essential
- very high NRE costs for custom ASIC
- FPGA-based developments (poster by Owen Callanan)
- improvements in cluster hardware
  - memory bus, chipsets, network ASICs, . . .
  - APENet project (poster by Roberto Ammendola)

Up- and coming machines

- PC cluster upgrades at Fermilab and JLAB
- PACS-CS (Tsukuba): June 2006 (talk by Akira Ukawa)
    - 14.3 TFlop/s peak
    - 2,560 2.8 GHz Intel Xeon processors
    - 3-d Gigabit Ethernet network (hyper-crossbar)
    - custom motherboard
- KEK is collecting bids for a $>$ 24 TFlop/s peak machine to be operational March 2006
- BlueGene/P
    - upgrade of BlueGene/L
    - not allowed to disclose details → IBM web site
- successor to QCDOC: under consideration
- Fujitsu
    - 3 PFlop/s by 2010 (or 2011?)
    - optical switching technology (to be developed)

Life Simulator?



TECHNOLOGY

Japan to build world's fastest supercomputer

Monday, July 25, 2005 Posted: 1600 GMT (0000 HKT)

TOKYO, Japan (AP) -- Japan has plans to start building a supercomputer next year that can operate 73 times faster than the world's fastest supercomputer, the government said Monday.

The American Blue Gene/L system supercomputer developed by International Business Machine Corp. at Lawrence Livermore National Laboratory in Livermore, California, currently holds the title of the world's fastest. That machine is capable of 136.8 teraflops, or 136.8 trillion calculations per second, according to Japan's Ministry of Education, Culture, Sports, Science and Technology.

Japan wants to develop a supercomputer that can operate at 10 petaflops, or 10 quadrillion calculations per second, which is 73 times faster than the Blue Gene, an official of the ministry said on condition of anonymity.

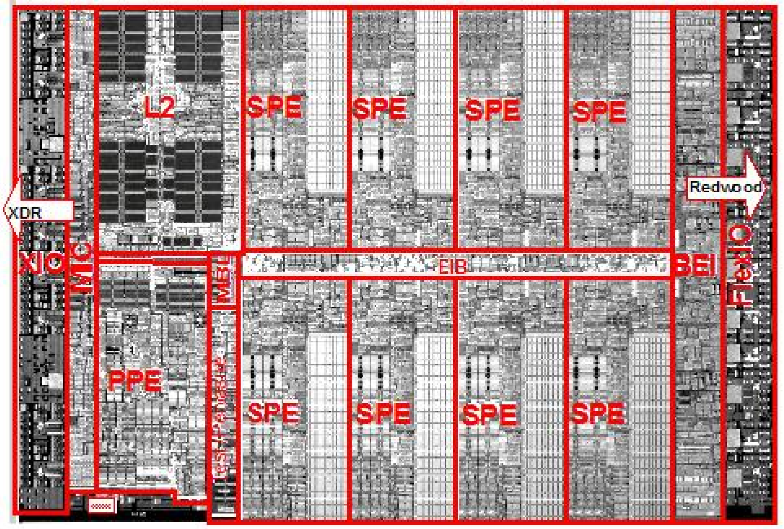
Kyodo News reported that the total amount for the project is estimated between 80 billion and 100 billion yen (U.S. $714 million to 893 million; euro593 million to 741 million) and the ministry will request 10 billion yen (U.S. $89 million; euro74 million) for the next fiscal year's budget.

The ministry official could not confirm the figures, saying it has yet to reach a formal decision on the project, which is expected by the end of August.

But he said that if the budget for next year is approved, the ministry hopes to complete the next-generation supercomputer sometime in fiscal 2010, which ends in March 2011.

The Cell

in your Playstation 3 from spring 2006



The Cell

- 300 engineers, full custom ASIC
- 0.09/0.065 μm process
- 50 $\sim$ 80 W at 4 GHz
- 1 (new) PowerPC CPU with 32 kB L1 caches (D/I)
- 8 FPUs with 256 kB of private memory
- each FPU can do 4 FMAs per cycle
  $\rightarrow$ 256 GFlop/s at 4 GHz (single precision, always rounds down)
- double precision $\sim 10\times$ slower
- 512 kB on-chip shared L2 cache
- 25 GB/s memory bandwidth (Rambus XDR)
- 76.8 GB/s I/O bandwidth (44.8 in, 32 out, Rambus FlexIO)
- Can memory subsystem keep the FPUs fed?
- Programming model?

Thanks

- SGI Altix: Thomas Streuer
- BlueGene/L: Pavlos Vranas
- PC clusters: Robert Edwards, Zoltan Fodor, Steve Gottlieb, Don Holmgren
- apeNEXT: Hubert Simma, Lele Tripiccione
- QCDOC: Peter Boyle, Mike Clark

Conclusions

- QCDOC and apeNEXT are the leading capability machines for QCD
- PC clusters competitive as capacity machines; scalability improving
- typical price-performance ratios close to $1 per sustained MFlop/s
- BlueGene/L is an interesting alternative, if you can get it cheaply (break-even point with QCDOC is $\sim \$10^6$ per rack, rental is $6.7 million/year/rack)
- commercial supercomputers typically are expensive and have limited scalability, but use them if your country/state/lab owns them
- lean software essential to get decent performance
- future developments look interesting
  - clear trend towards on-chip parallelization
  - programming models likely to change
  - both clusters and custom-designed machines will remain important

Welcome to Lattice 2007

Welcome to Lattice 2007