



IBM Systems & Technology Group

The Evolution of the Blue Gene/L Supercomputer

José E. Moreira
IBM Systems and Technology Group
Rochester, MN



Blue Gene/L at a glance

- The world's fastest supercomputer since November of 2004, according to the TOP500 list (www.top500.org)
- A computational science machine that is capable of producing “new science” for tough problems that could not be tackled before
- The first in a family of supercomputers from IBM that will be leaders in performance for scientific and technical computing
- An exercise in collaboration, both within IBM and between IBM and a major United States national laboratory that produced a machine that “delivers”
- A source of many sleepless nights and (temporary) frustrations for a dedicated team of scientists and engineers that designed, developed, and made the machine work (hardware, software, and applications)

Flagship Blue Gene/L – 65,536 dual-processor compute nodes



Status of the LLNL flagship system (why do you care?)

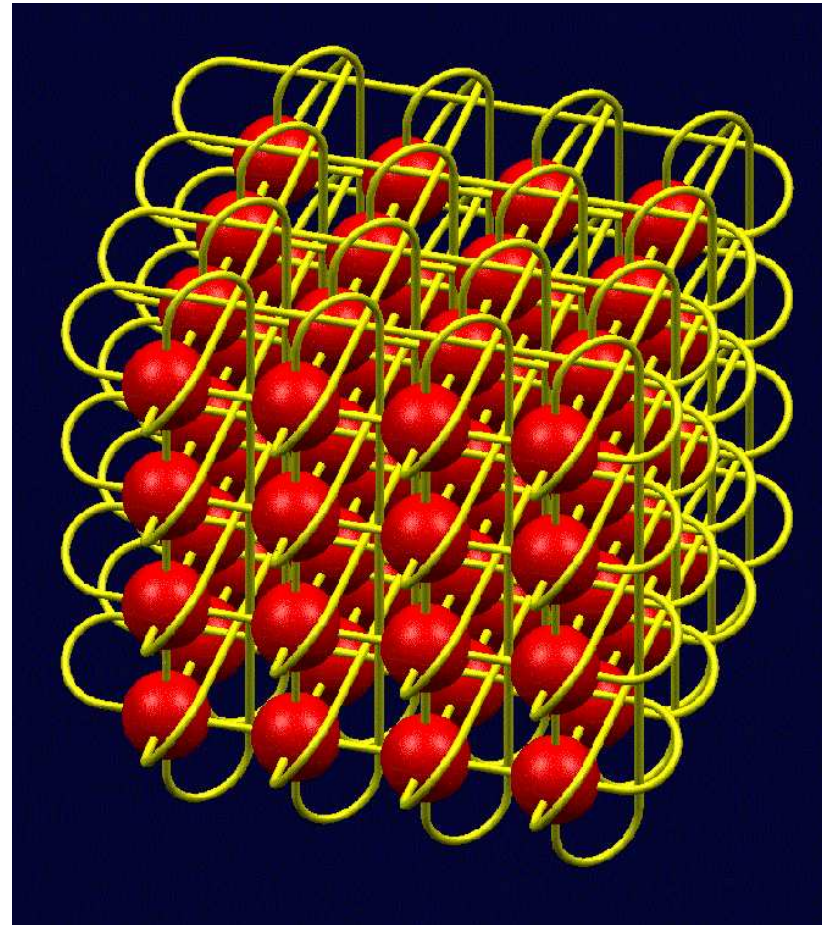
- The 64-rack flagship Blue Gene/L system completed acceptance tests at LLNL last week (just in time for end of 3Q2005)
- Even before acceptance, the system was used to produce new science results, using full 65,536-node runs in virtual node mode (over 130,000 tasks!)
- Although getting the 64-rack system running was not that difficult (both 32-rack halves had been in operation at LLNL and IBM Rochester), some new issues became more evident at 64 racks
 - *These issues required software workarounds*
 - *Although their impact is less on small systems, these issues are still relevant*
 - *Overall, we will have a better system (even small configurations) because of the extreme scaling tests done at LLNL*
 - *Win-win for everybody*

A more intimate view of Blue Gene/L

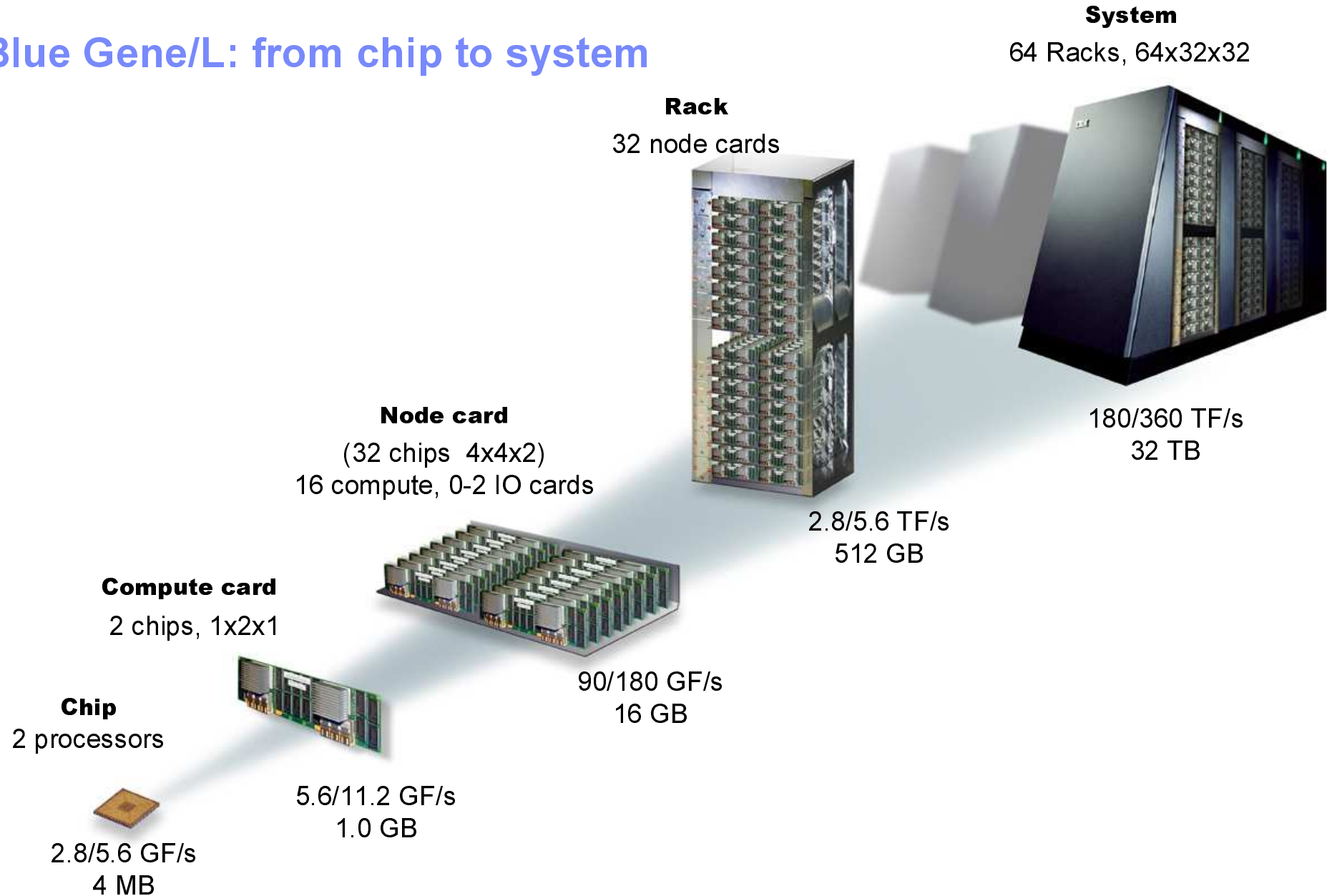


BlueGene/L fundamentals

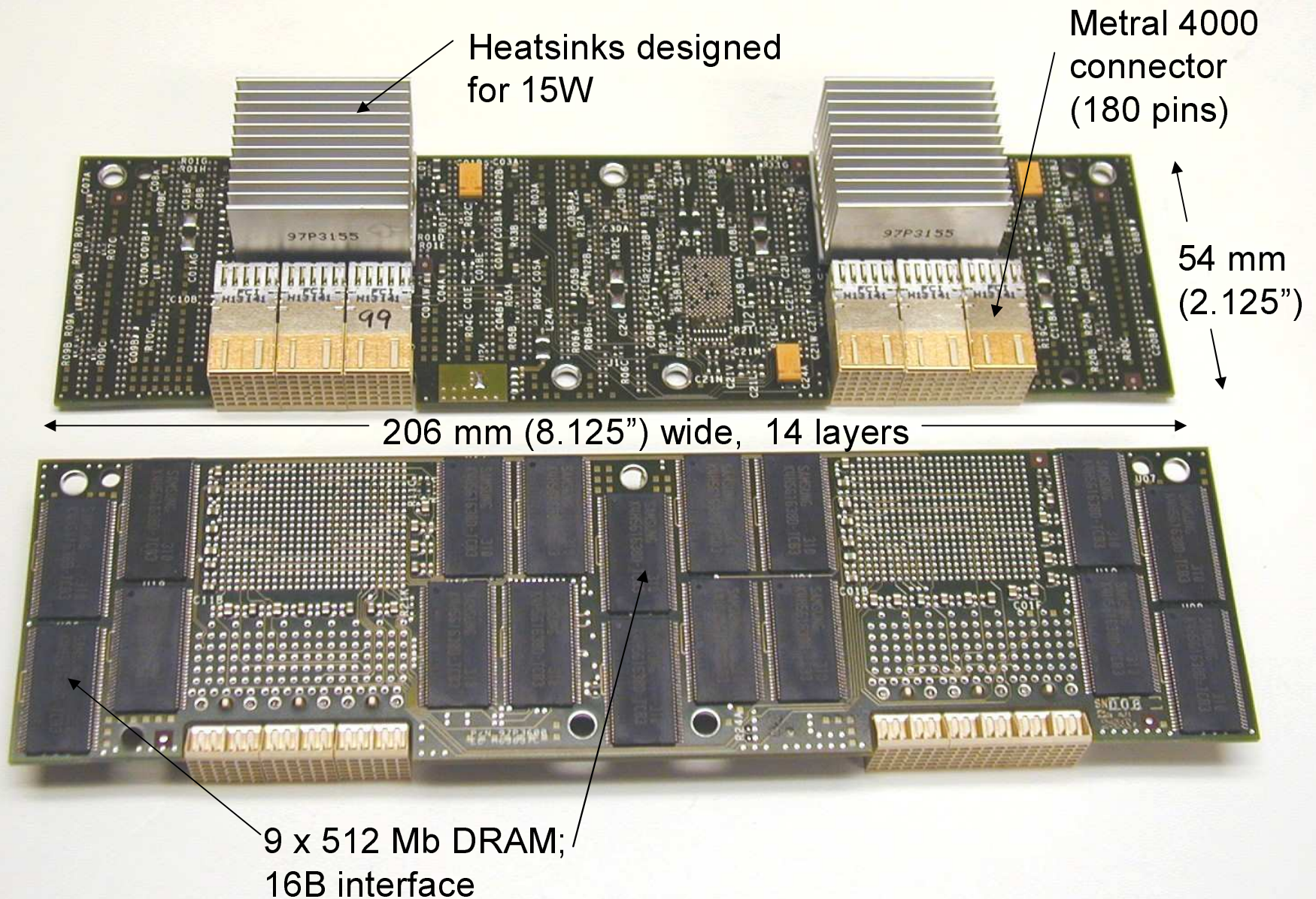
- A large number of nodes (65,536)
 - Low-power (25W) nodes for density
 - High floating-point performance
 - System-on-a-chip technology
- Nodes interconnected as 64x32x32 three-dimensional torus
 - Easy to build large systems, as each node connects only to six nearest neighbors – full routing in hardware
 - Auxiliary networks for I/O and global operations
- Applications consist of multiple processes with message passing
 - Strictly one process/node (coprocessor mode)
 - Two processes/node (virtual node mode)
 - Minimum OS involvement and overhead

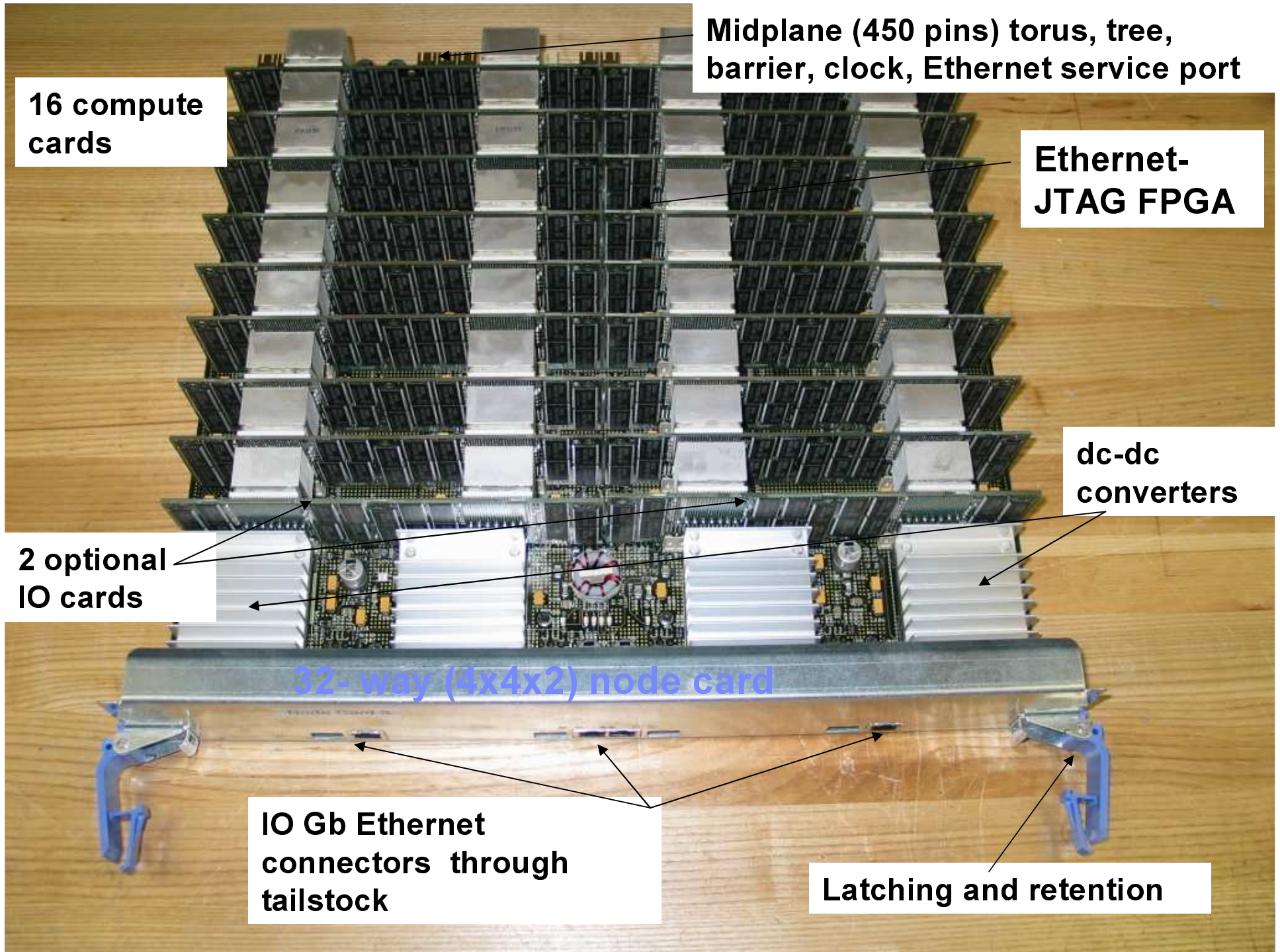


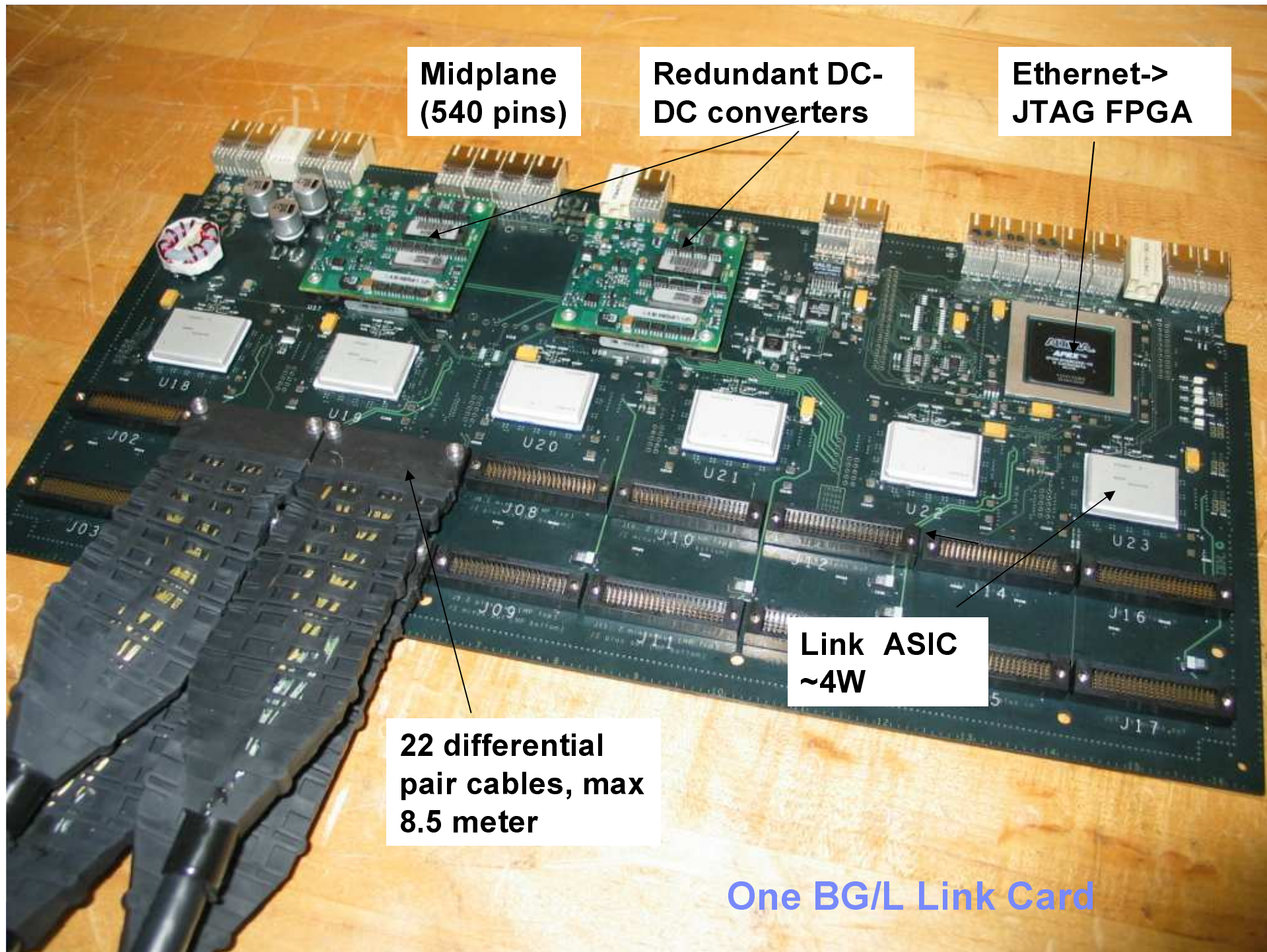
Blue Gene/L: from chip to system



One dual-node compute card







Midplane
(540 pins)

Redundant DC-
DC converters

Ethernet->
JTAG FPGA

Link ASIC
~4W

22 differential
pair cables, max
8.5 meter

One BG/L Link Card

One Blue Gene/L rack

- One Midplane:
 - 16 x 32 Node Node Boards
 - 1 Service Card
 - 4 Link Cards
- Two Midplanes / Rack

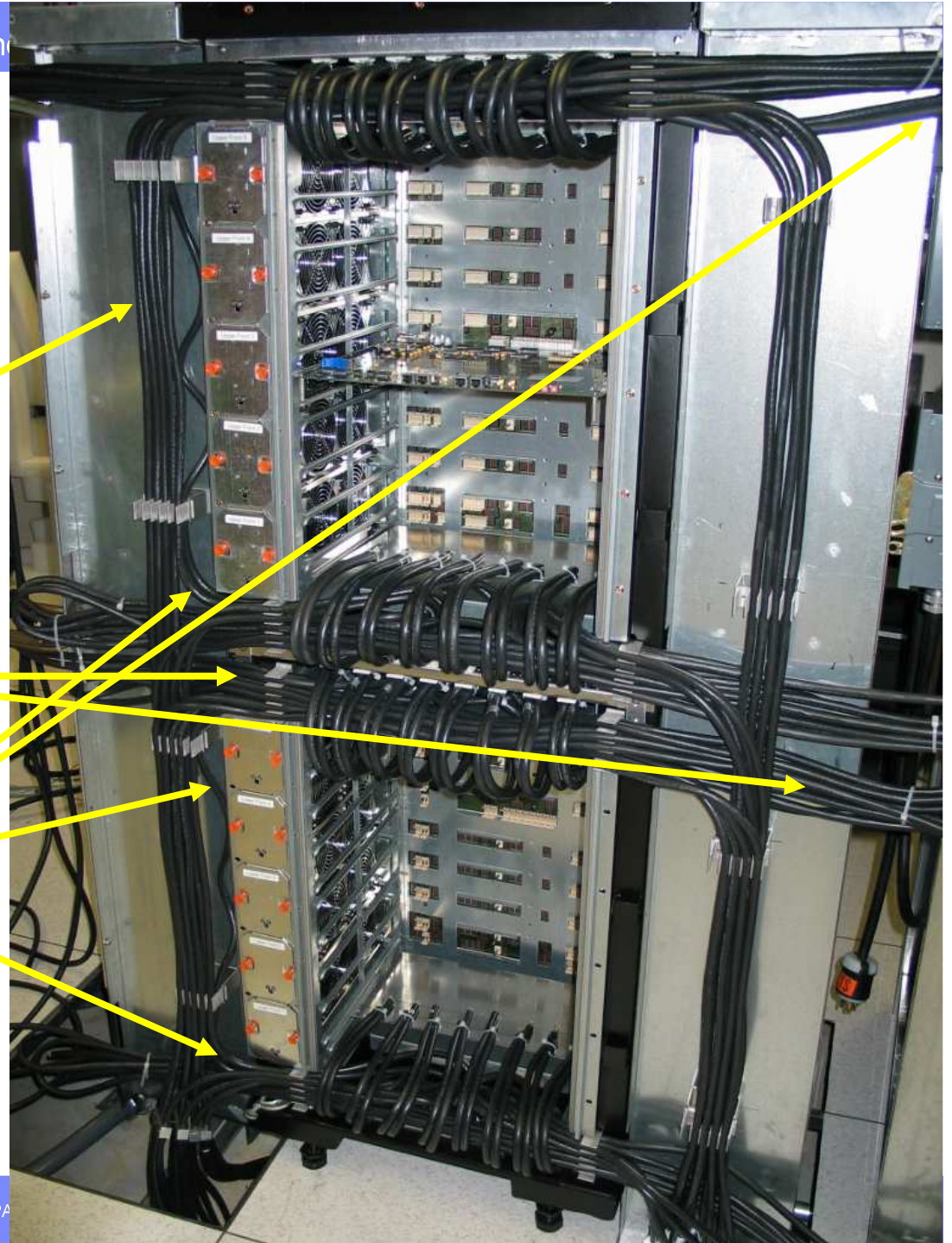


The cables

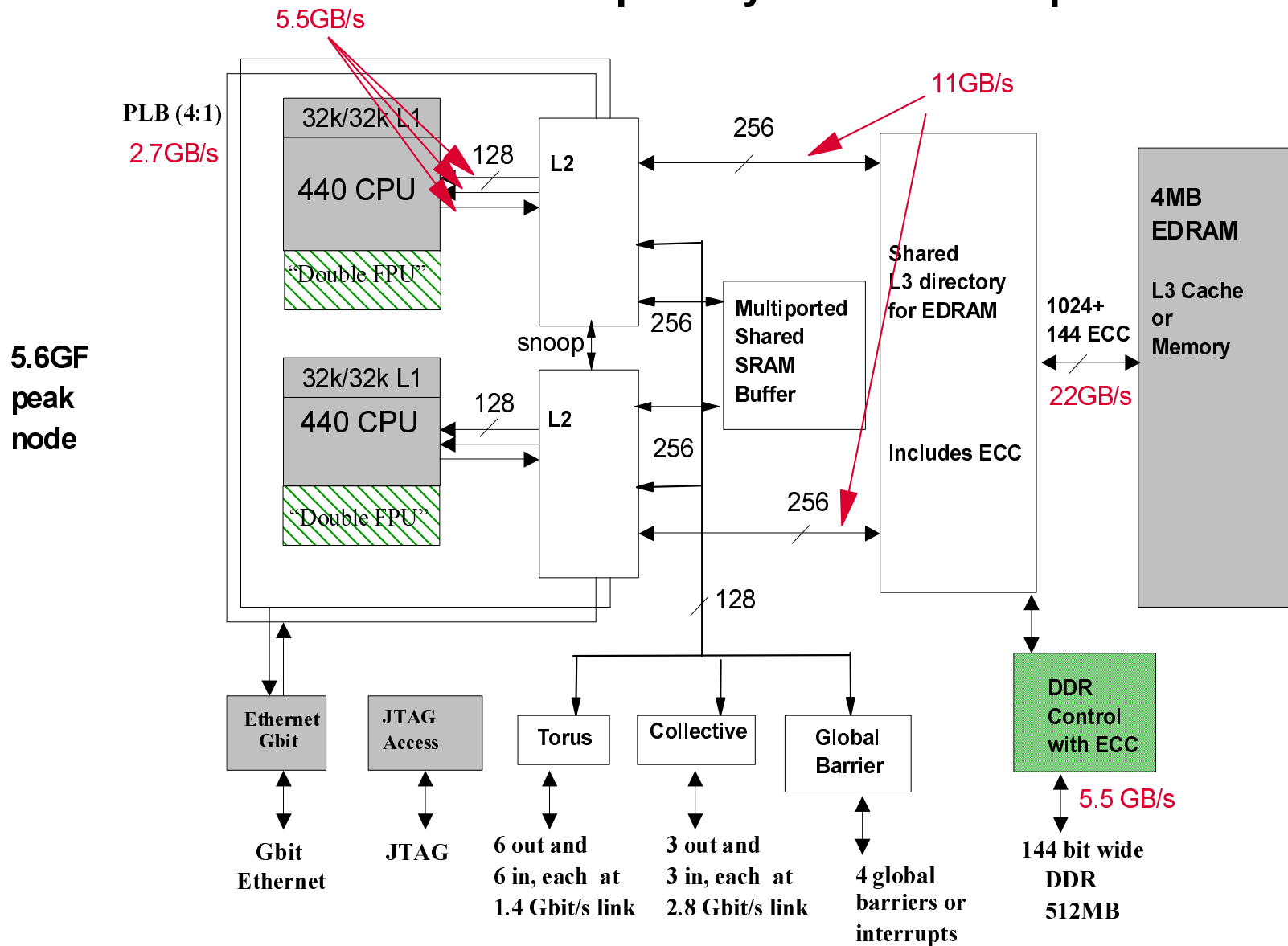
X Cables

Y Cables

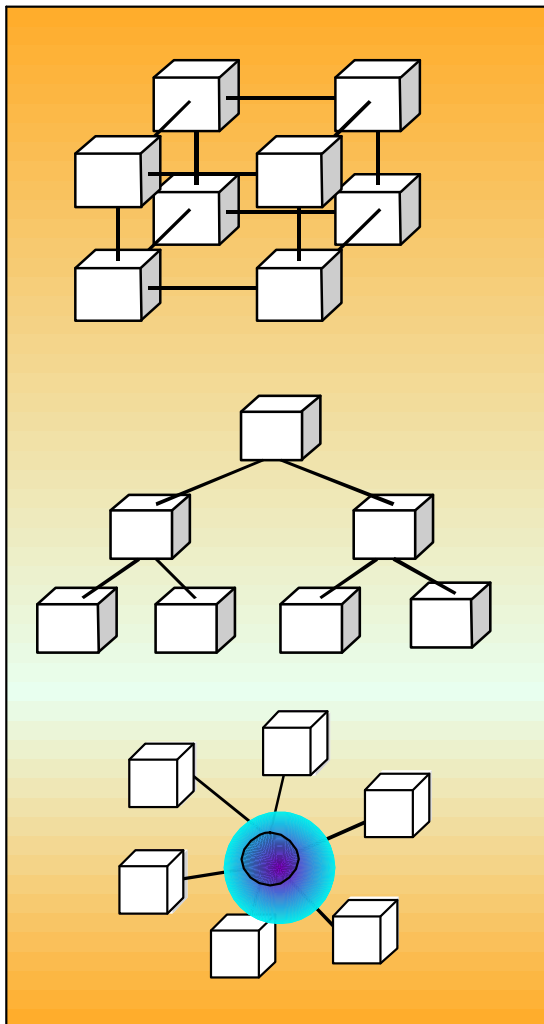
Z Cables



BlueGene/L Compute System-on-a-Chip ASIC



Blue Gene/L interconnection networks



3 Dimensional Torus

- Interconnects all compute nodes (65,536)
- Virtual cut-through hardware routing
- 1.4Gb/s on all 12 node links (2.1 GB/s per node)
- Communications backbone for computations
- 0.7/1.4 TB/s bisection bandwidth, 67TB/s total bandwidth

Global Collective Network

- One-to-all broadcast functionality
- Reduction operations functionality
- 2.8 Gb/s of bandwidth per link; Latency of tree traversal 2.5 μ s
- ~23TB/s total binary tree bandwidth (64k machine)
- Interconnects all compute and I/O nodes (1024)

Low Latency Global Barrier and Interrupt

- Round trip latency 1.3 μ s

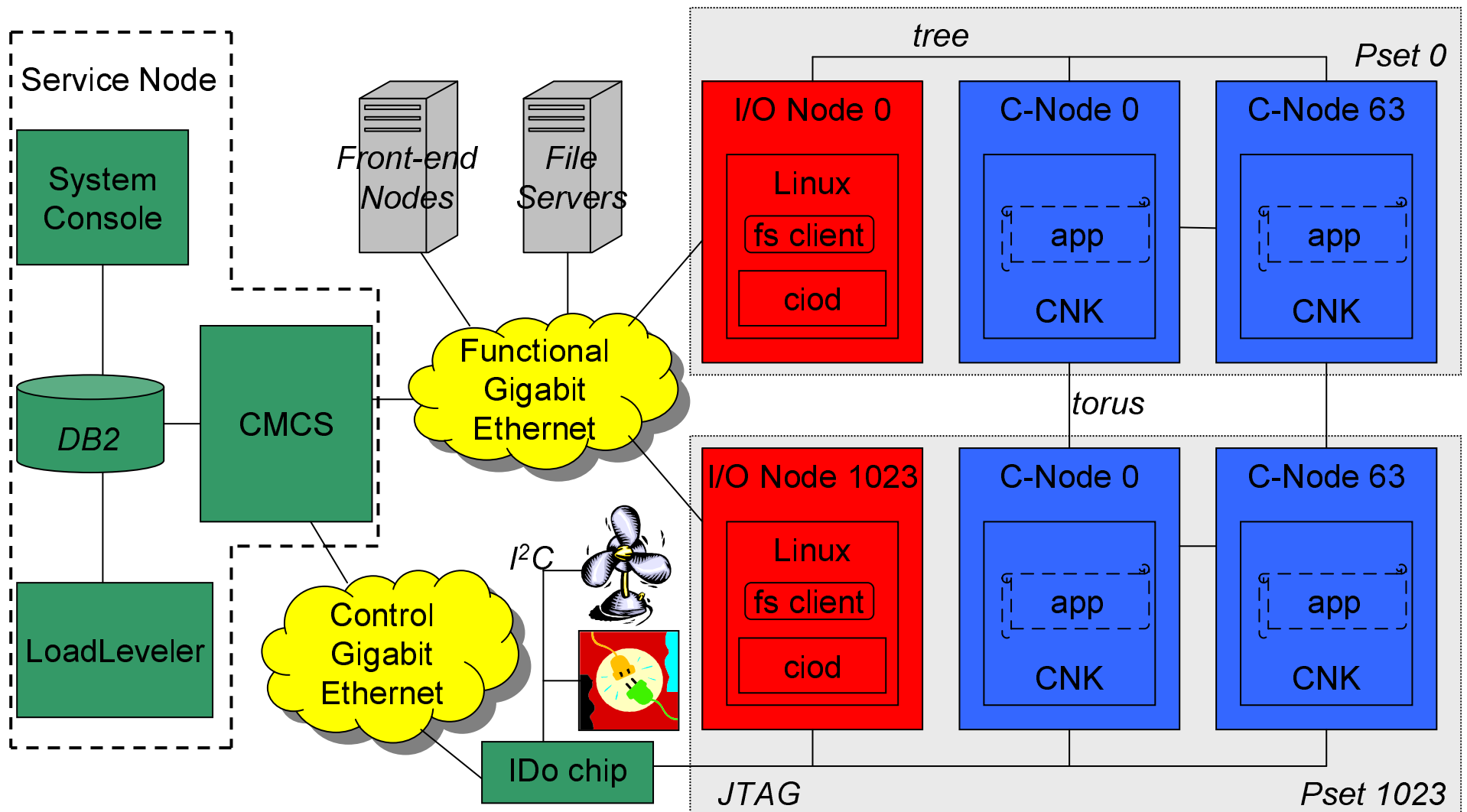
Control Network

- Boot, monitoring and diagnostics

Ethernet

- Incorporated into every node ASIC
- Active in the I/O nodes (1:64)
- All external comm. (file I/O, control, user interaction, etc.)

Blue Gene/L system architecture



How did Blue Gene/L came to be?

- Status quo in 1999:
 - *Universities and research labs in the US, Europe, and Japan had demonstrated that massively parallel computers, with a dedicated architecture, could efficiently and effectively scale to ~10,000 processors when performing QCD computations – these machines were based on a simple and regular nearest-neighbor interconnect (torus or mesh).*
 - *The “accepted” approach for building more “general purpose” was to use large SMP nodes (4 to 16 processors), interconnected by a very flexible network (usually a fat-tree).*
- What happened in 1999:
 - *Researchers at IBM Thomas J. Watson Research Center proposed to Lawrence Livermore National Laboratory to build a supercomputer with the basic architecture of the QCD machines, but that would be more “general purpose”.*
 - *The basic concept was not “crazy”, since machines like ASCI Red and the Cray T3D/T3E had been quite successful in the past.*
 - *The scale, however, was “crazy”: 65,536 compute nodes!*
 - *LLNL got very interested and we started a partnership to develop the machine.*
- The rest is history ...

Blue Gene/L – the science

- What is the theory behind Blue Gene/L?
- That is easy: $\lim_{N \rightarrow \infty} \alpha N = \infty, \forall \alpha > 0$
- In English: We can make up for a slow processor by using many replicas of that processor
- But, but, but, ... why would I want to use a slow processor?
- That is easy: Slower processors are more efficient, they have higher flop/watt rate than fast processors – this is a consequence of the technology that makes getting more performance out of a processor increasingly more difficult
- So, if you are (electrical) power constrained (and all modern systems are), then you get more (compute) power by using slower processors ...
- OK, so let us use 100 billion 1 kHz processors (oops, sorry, we all do that already ☺)

Blue Gene/L – the engineering

- In the end, it is all about making the right engineering decisions
- Computer design is all about making compromises – Turing proved that all computers are equivalent a while ago!
- So what are the engineering decisions to be made:
 - *First, you have to choose your building blocks, at several levels from small to large*
 - *You need to make sure that you can power and cool each of those building blocks*
 - *Those building blocks need to be interconnected together – How?*
 - *The resulting system must be reliable enough to be usable and useful*
 - *What about the system “manufacturability” and “serviceability”?*
 - *How are you going to test the system after manufacture or service?*
 - *What system software are you going to run on this system?*
 - *At the end of all this, you can ask the “interesting question”: do I have a system that can solve problems at a price and performance that is competitive?*

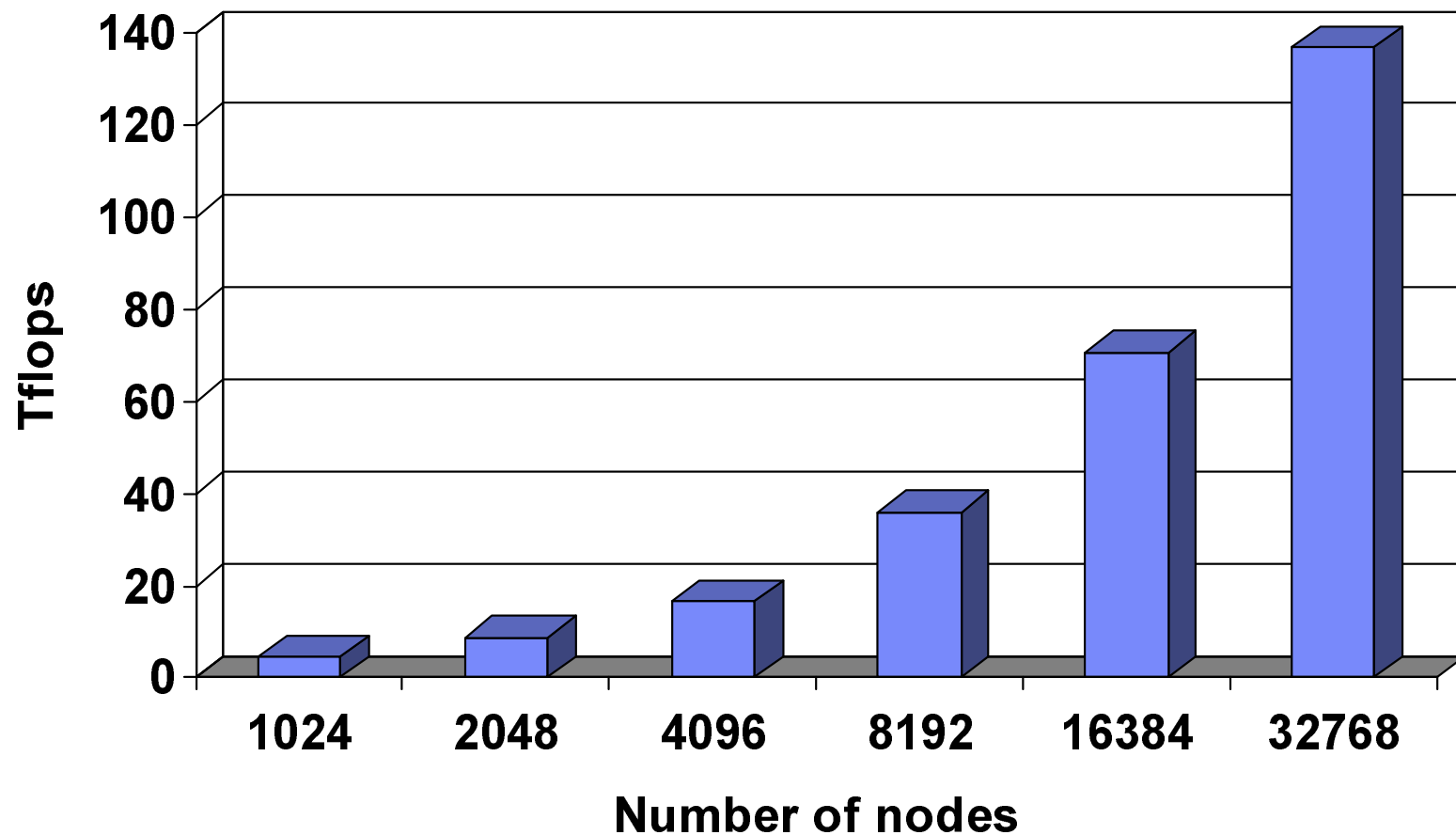
So, how were the decisions made?

- Designing a computer system, specially a brand new one, is part science and part art
- The hardware architects and engineers were very aggressive: They wanted to implement an entire compute node (except for memory) in a single chip
 - *With everything integrated on a single chip, the rest of the system is much easier*
 - *And cheaper!*
 - *And more reliable!*
 - *And uses less power!*
- Based on the power budget of the technology, they knew they could get approximate 1000 of these compute nodes in one rack of electronics
- So, they worked very hard to solve the power, packaging, cooling, and cabling issues in order to achieve those goals
- They succeeded!

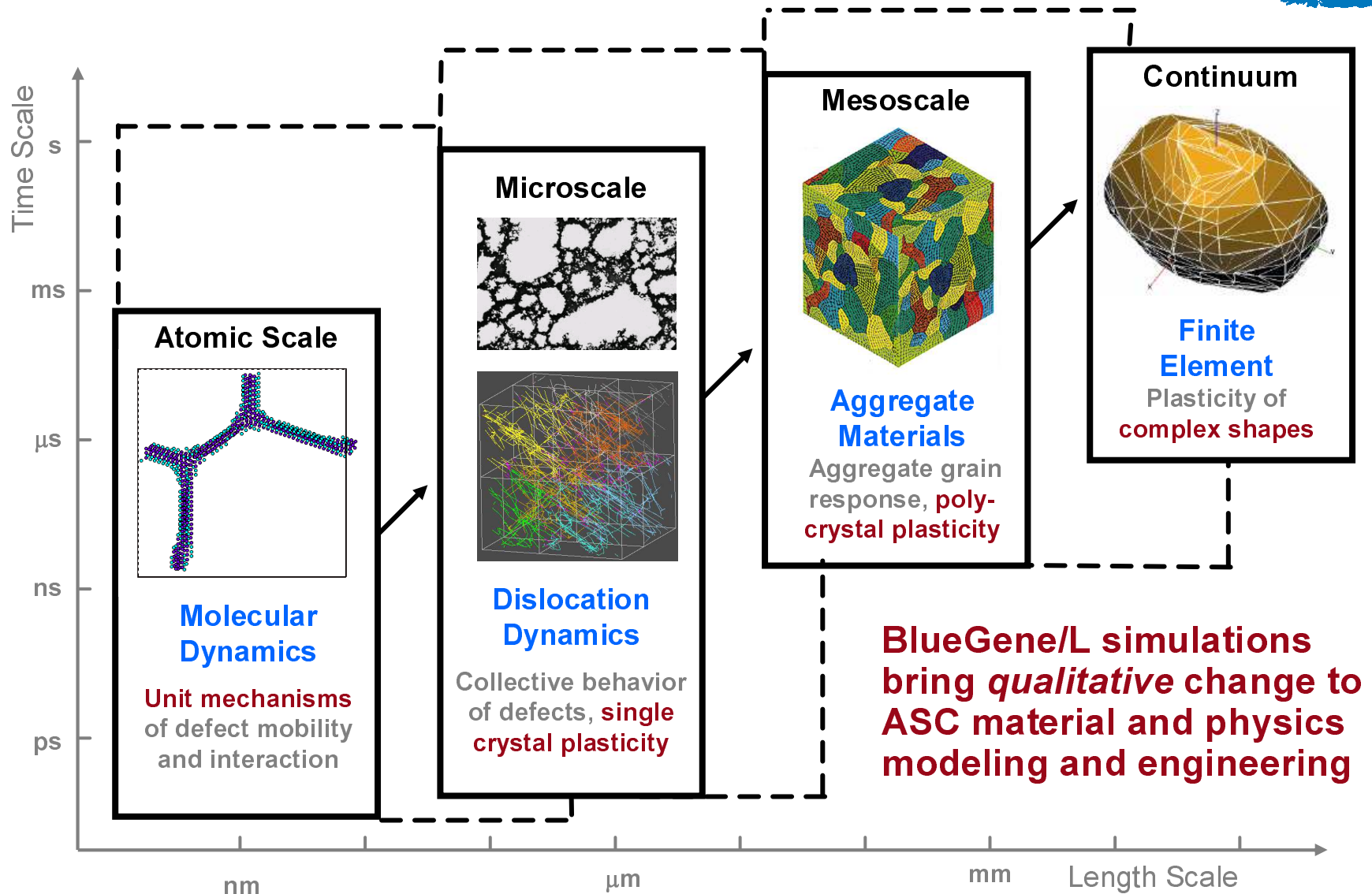
What about software?

- The software architects and engineers had one major goal: To allow “out-of-the-box” applications to efficiently exploit the hardware and achieve high performance on the machine
- That this goal was realistic is the result of 20+ years of hard work around the world on parallelization of applications – when Blue Gene/L arrived, there were applications ready to use it!
- Approach of the software team: Stick to proven languages and libraries, which means Fortran/C/C++ and MPI, but reimplement where necessary to achieve high performance
- They succeeded!

Linpack – the benchmark for TOP500



BlueGene/L will allow overlapping evaluation of models for first time

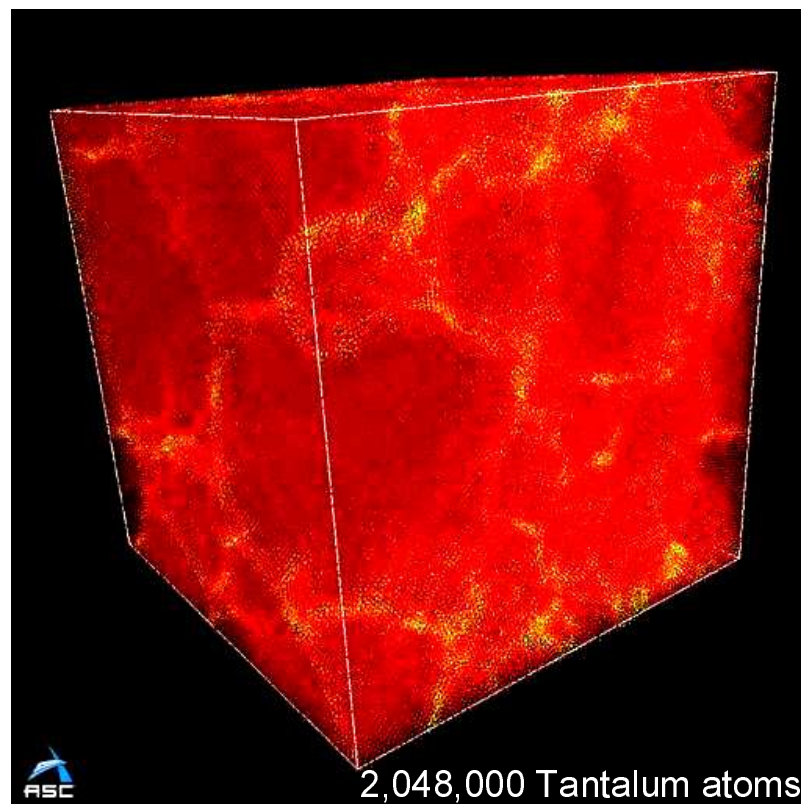




Classical MD – ddcMD

- Scalable, general purpose code for performing classical molecular dynamics (MD) simulations using highly accurate MGPT potentials
- MGPT semi-empirical potentials, based on a rigorous expansion of many body terms in the total energy, are needed in to quantitatively investigate dynamic behavior of transition metals and actinides under extreme conditions

64K and 256K atom simulations on 2K nodes are order of magnitude larger than previously attempted; based on 16M atom simulation on 32K nodes, *close to perfect scaling* expected for full machine (“very impressive machine” says PI...)

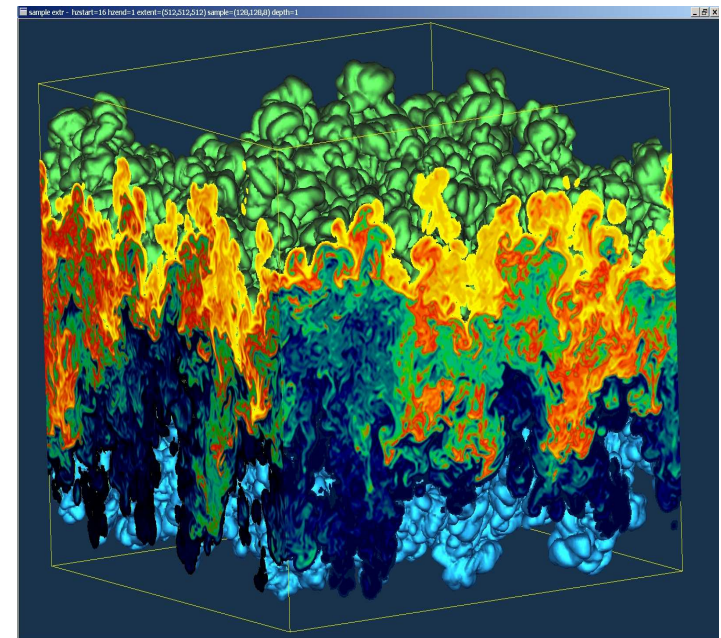


Visualization of important new scientific findings already achieved on BG/L: Molten Ta at 5000K demonstrates solidification during isothermal compression to 250 GPa

Instability and Turbulence - Miranda



- High order hydrodynamics code for computing fluid instabilities and turbulent mix
- Employs FFTs and band-diagonal matrix solvers to compute spectrally-accurate derivatives, combined with high-order integration methods for time advancement
- Contains solvers for both compressible and incompressible flows
- Has been used primarily for studying Rayleigh-Taylor (R-T) and Richtmyer-Meshkov (R-M) instabilities, which occur in supernovae and Inertial Confinement Fusion (ICF)



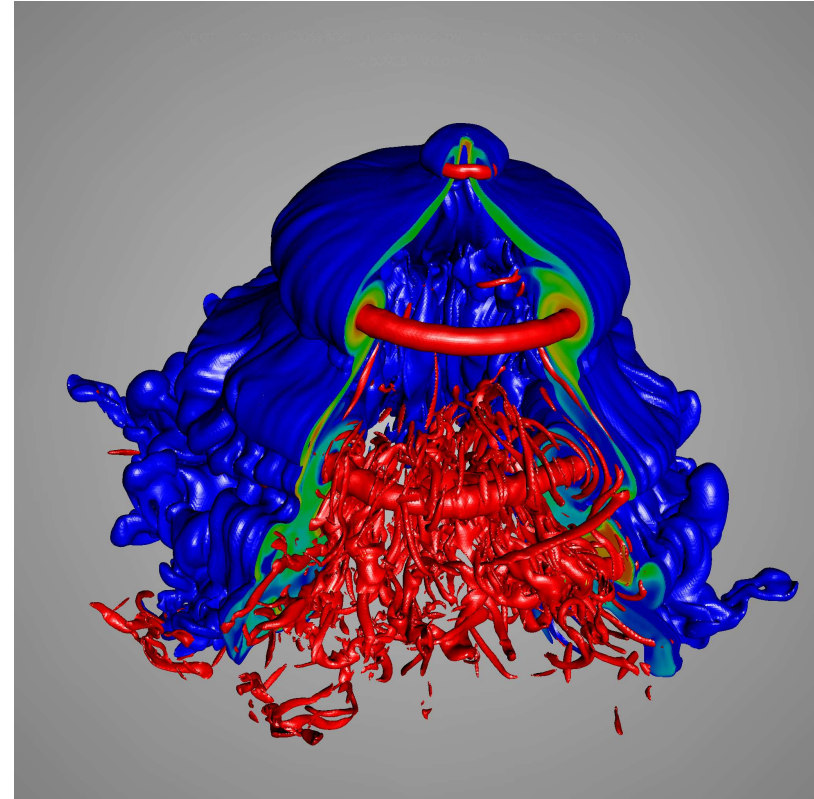
Miranda has successfully run on 32,768 nodes on BG/L and also on 32,768 processors in “virtual node” mode. BG/L enables wide range of scales in space and time necessary to represent turbulent flows of interest. Good time-to-solution improvement from MCR to BG/L .

Instability and Turbulence - Raptor



- Multi-physics Eulerian Adaptive Mesh Refinement (AMR) code used for applications at LLNL including astrophysics, Inertial Confinement Fusion (ICF) and shock-driven instabilities and turbulence
- Can be used to simulate purely fluid dynamics systems and more complex physical systems where the fluids are coupled to the radiation field, such as in ICF or astrophysics

Simulations at full scale on BG/L will offer the computational power to gain an order of magnitude more resolution in simulations of three-dimensional shock-driven systems.

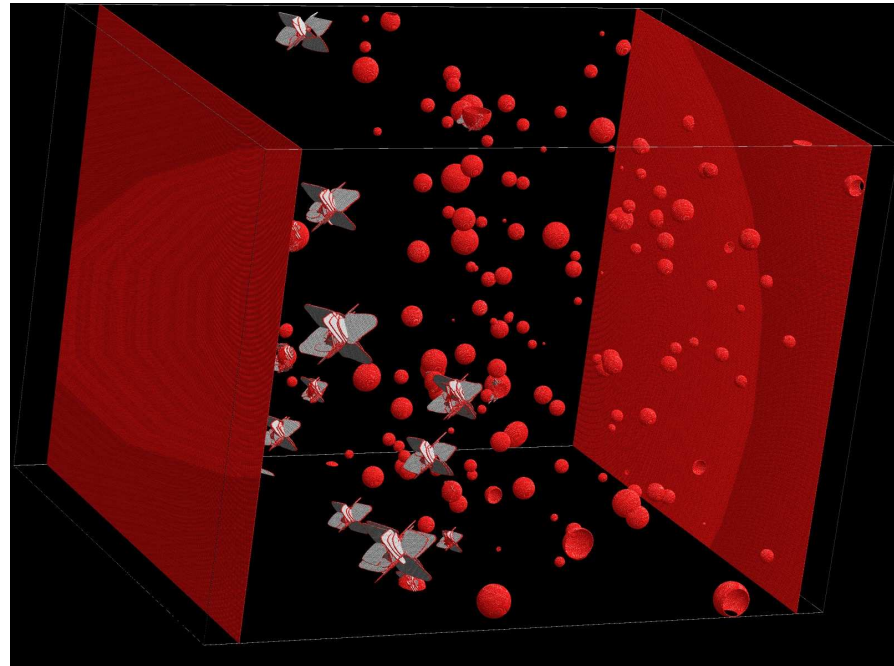


Dense spherical argon bubble, initially contained in a thin spherical soap film, suspended in nitrogen, subjected to a strong planar shock wave about 509 microseconds after shock-bubble interaction.



Classical MD - SPaSM (LANL)

- A high performance (1993 and 1998 Gordon Bell prizes) code for Scalable Parallel Short-range Molecular dynamics simulations
- A variety of finite-range empirical potentials are implemented, including EAM and MEAM for metals, Stillinger-Weber Si/Ge, and a reactive empirical bond-order (REBO) potential for detonation studies.



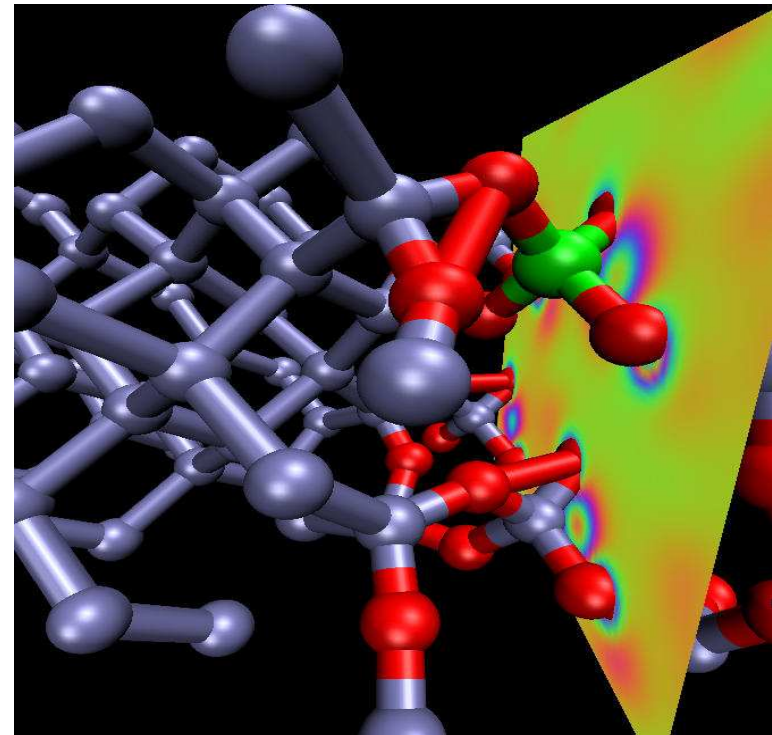
SPaSM has exhibited excellent scaling for up to 100 billion atoms on 16,384 nodes, and an initial production run on 8k nodes simulated the shock loading of a 2.1 billion atom EAM copper crystal with 0.41% (by volume) voids. BG/L will enable the exploration of an entirely new class of (previously intractable) problems such as this.

First-Principles MD - CPMD

(IBM Zurich, LLNL collaboration)



- Computation and communication intensive application developed by Alessandro Curioni and colleagues at IBM Zurich, now beginning science runs at LLNL
- Uses plane wave basis functions, 3D-FFT and parallel linear algebra routines to study the electronics and structural properties of complex materials from first principles



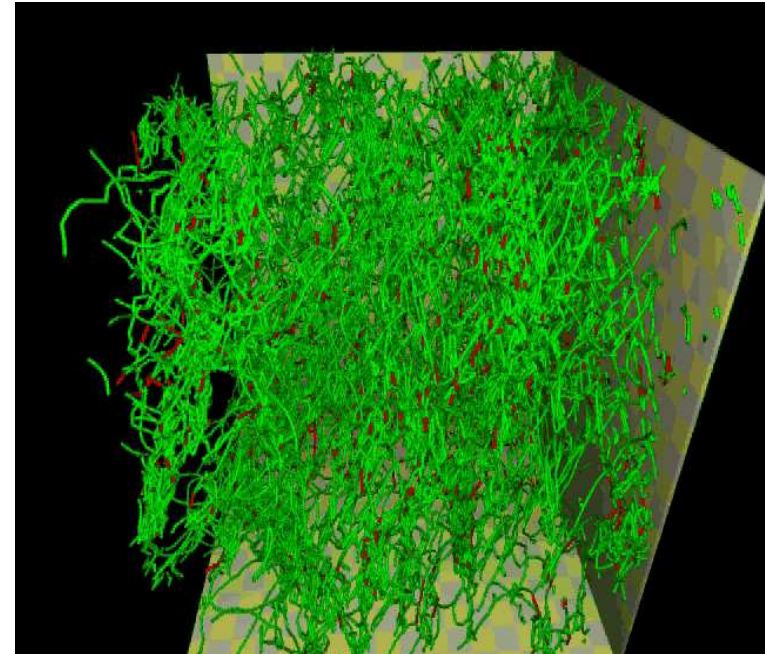
CPMD is used worldwide, and has been identified as an important tool for several critical material research areas and efforts at LLNL.

Dislocation Dynamics - ParaDiS (Parallel Dislocation Simulator)



- New LLNL code for direct computation of plastic strength of materials
- Tracks simultaneous motion of millions of dislocation lines
- Promises to close the computational performance gap that prevents scientists from understanding the fundamental nature of material strengthening (or hardening)

ParaDiS has run on 16,384 nodes of BG/L, and is currently investigating scaling and dynamic load balancing issues to achieve higher efficiencies.



Killer applications:

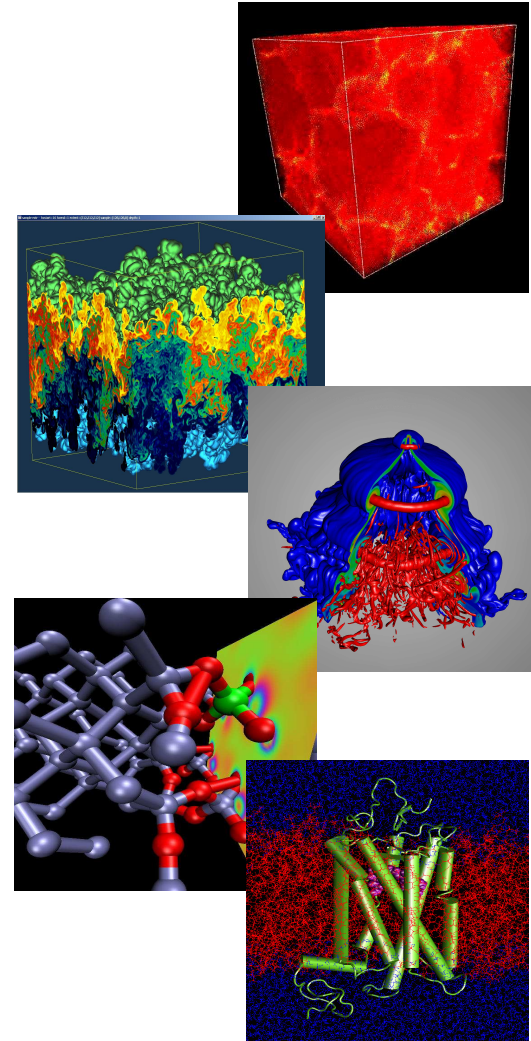
- full simulation of poly-crystal solidification from melt
- alloy microstructure evolution during plastic deformation
- science of ultra-fast polymer crystallization

Application “time to solution” to be improved by an order of magnitude



Application results to date on BG/L are pointing to a *qualitative change in the way science can be done*

- Scientist can do a new run every day and can afford to make investigations and then explore alternatives
- An entire scientific study can be now be done in the same time as just one run only a year ago



System installations (1 rack = 1024 compute nodes)

1. LLNL: 64 racks (+1-rack development system)
2. San Diego Supercomputing Center: 1 rack
3. Argonne: 1 rack
4. AIST (Japan): 4 racks
5. University of Edinburgh: 1 rack
6. ASTRON (Netherlands): 6 racks
7. National Center for Atmospheric Research: 1 rack
8. NIWS (Japan): 1 rack
9. IBM Almaden: 2 racks
10. IBM Yorktown Heights: 20 racks (+2-rack development system)
11. Boston University: 1 rack
12. EPFL (Switzerland): 4 racks
13. IBM On Demand Center (Rochester): 2 racks
14. MIT: 1 rack
15. Research Centre Juelich (Germany): 1 rack
16. KEK: 10 racks (won bid – need to install)
17. Princeton University: 1 rack (under installation)

Over 500 Tflops deployed!

Recent software accomplishments

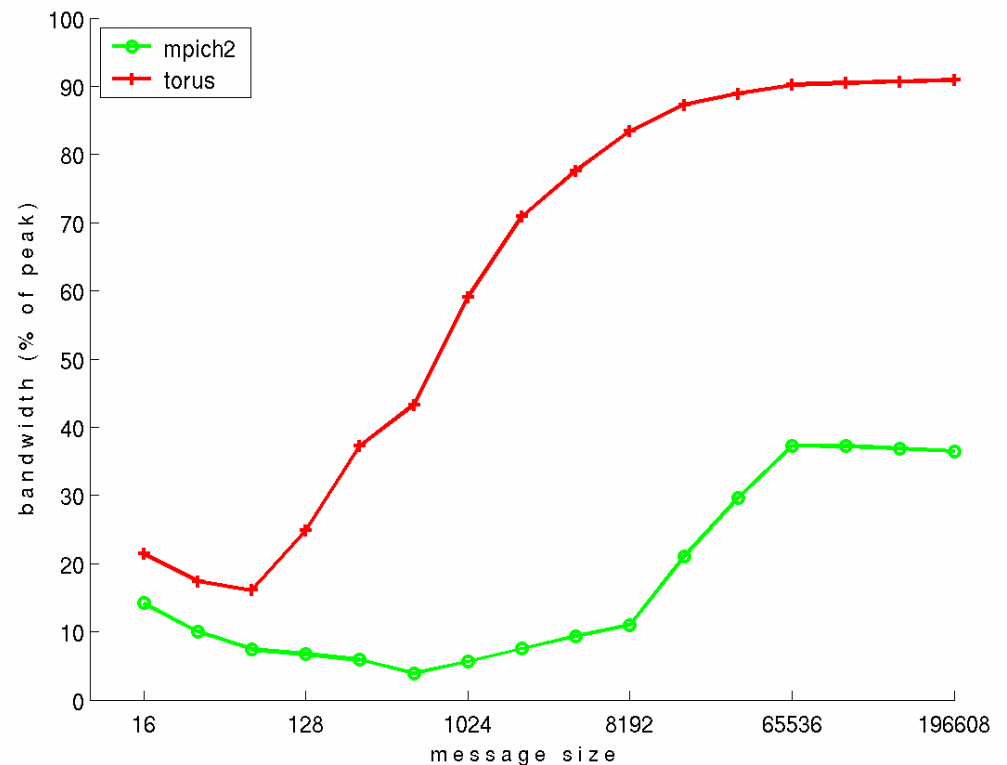
- General availability (GA) of system software stack in August/2005, which required significant testing and hardening effort (Release 2 in November)
- Scaling to 64 racks at LLNL (fixes were required)
- Significant hardening of XL compilers (particularly for Double FPU)
 - *Much more robust code generation*
 - *Much fewer cases of performance degradation from 440 to 440d*
- Started deploying batch scheduling system
 - *Based on LoadLeveler for p-Series*
 - *Uses a special-purpose external scheduler developed for Blue Gene/L*
- Significant enhancements to MPI
 - *Better exploitation of the coprocessor in coprocessor mode (more offloading)*
 - *Better optimizations for virtual node mode (division of labor)*
 - *Enhanced collectives that leverage Blue Gene/L networks (torus, collectives, GIs): broadcast, barrier, alltoall, alltoallv, allgather, allgatherv, reduce, allreduce*

Why does Blue Gene/L scale?

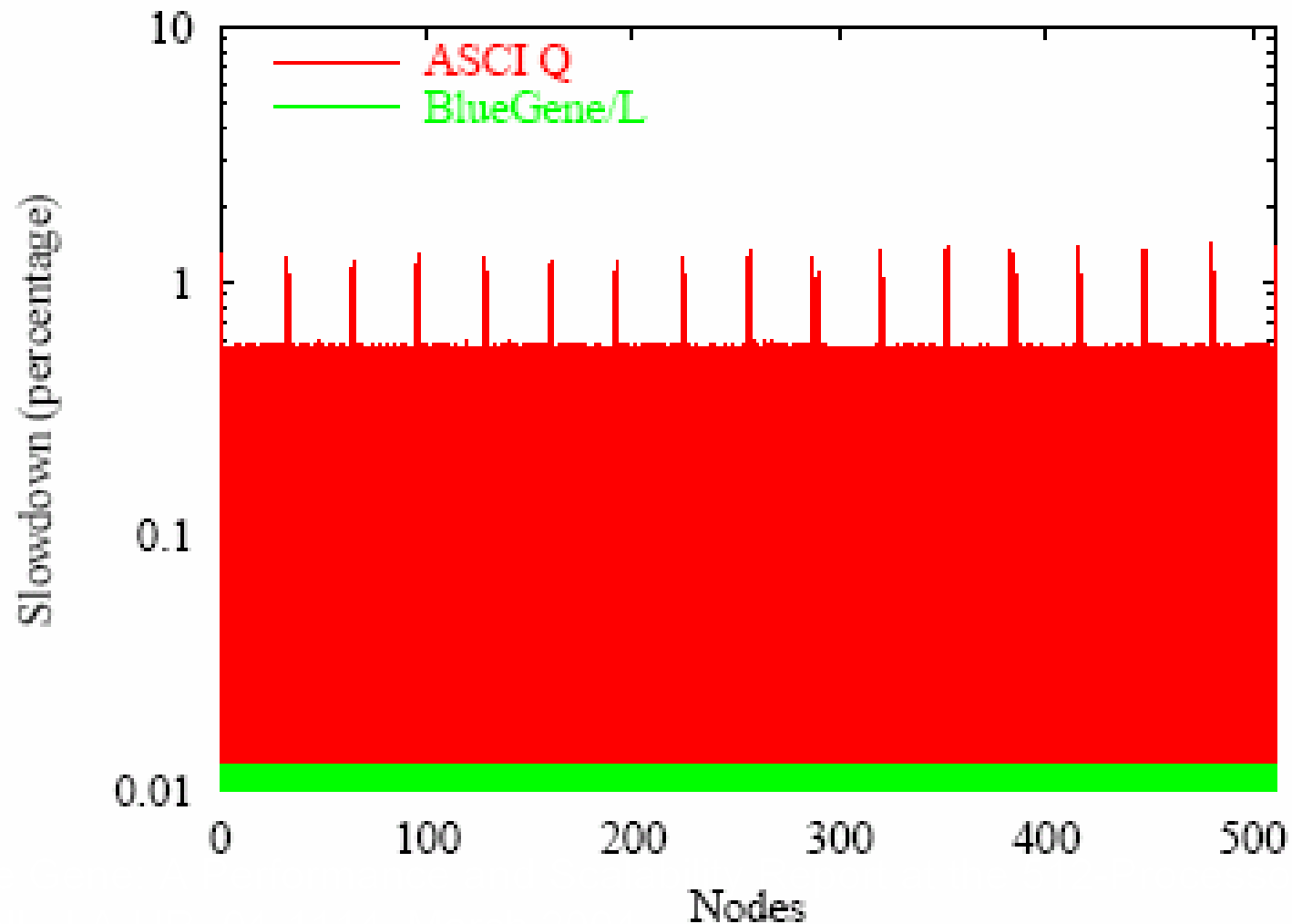
- Two types of scalability:
 - *Weak scaling – keep the problem size/processor constant (easier)*
 - *Strong scaling – keep the problem size/job constant (harder)*
- Blue Gene/L was designed for strong scalability:
 - *High nearest neighbor bandwidth – 175 MB/s each direction for 6 nearest neighbors*
 - *Tight integration results in low nearest neighbor latency*
 - ~3 us or 2300 cycles nearest neighbor latency
 - Also results in low $n_{1/2}$ – problems in BlueGene/L don't have to be huge
 - *Architectural features for fast collectives*
 - Collectives network for reductions/broadcasts on entire partition
 - Torus deposit bit for rectangular collectives
 - *Key software features for scalability*
 - MPI library that exploits the hardware features to deliver high communication performance
 - Operating environment that dedicates compute nodes to application processes, resulting in less interference
 - *Judicious use of redundancy*
 - ***20+ years of application development for parallel systems***

MPI exploitation of hardware features

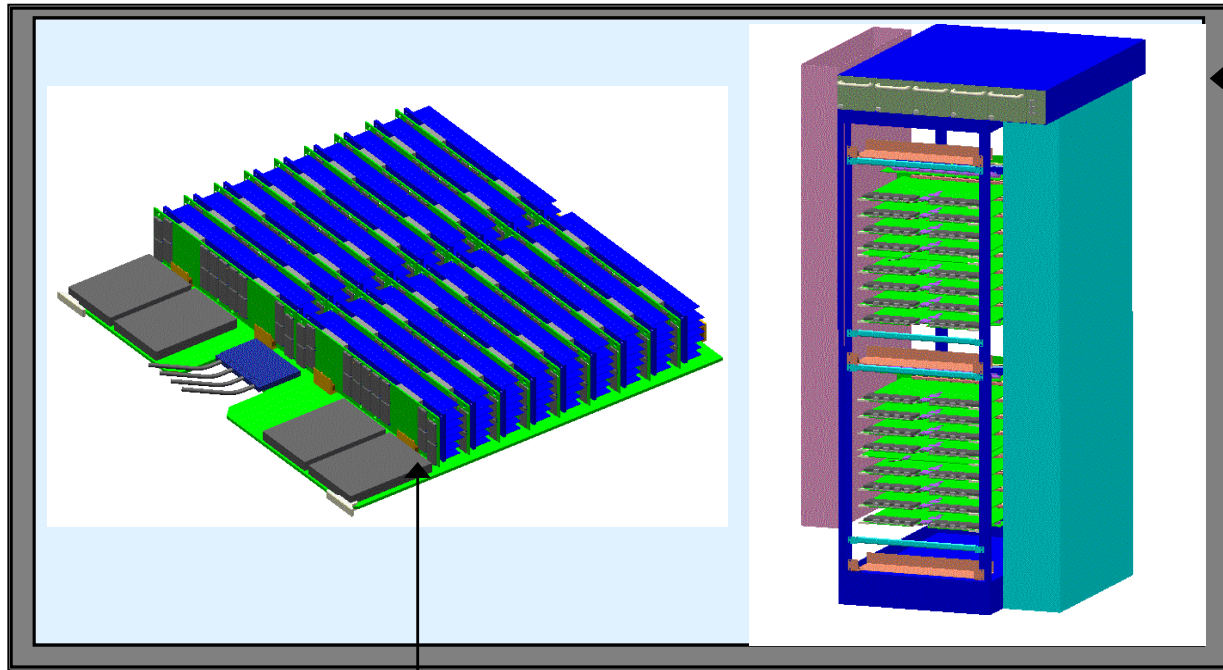
- We have been able to exploit all Blue Gene/L networks (torus, collectives, barrier) from within MPI
- Decision of which network (and protocol) to use is made inside MPI based on data type, message size, and communicator
- Effective use of the networks has required new implementations of important collective operations
- Graph compares all-to-all bandwidth for standard MPICH2 algorithm and a torus-optimized version



Noise measurements (from Adolfy Hoisie – LANL)



Judicious choice of redundancy

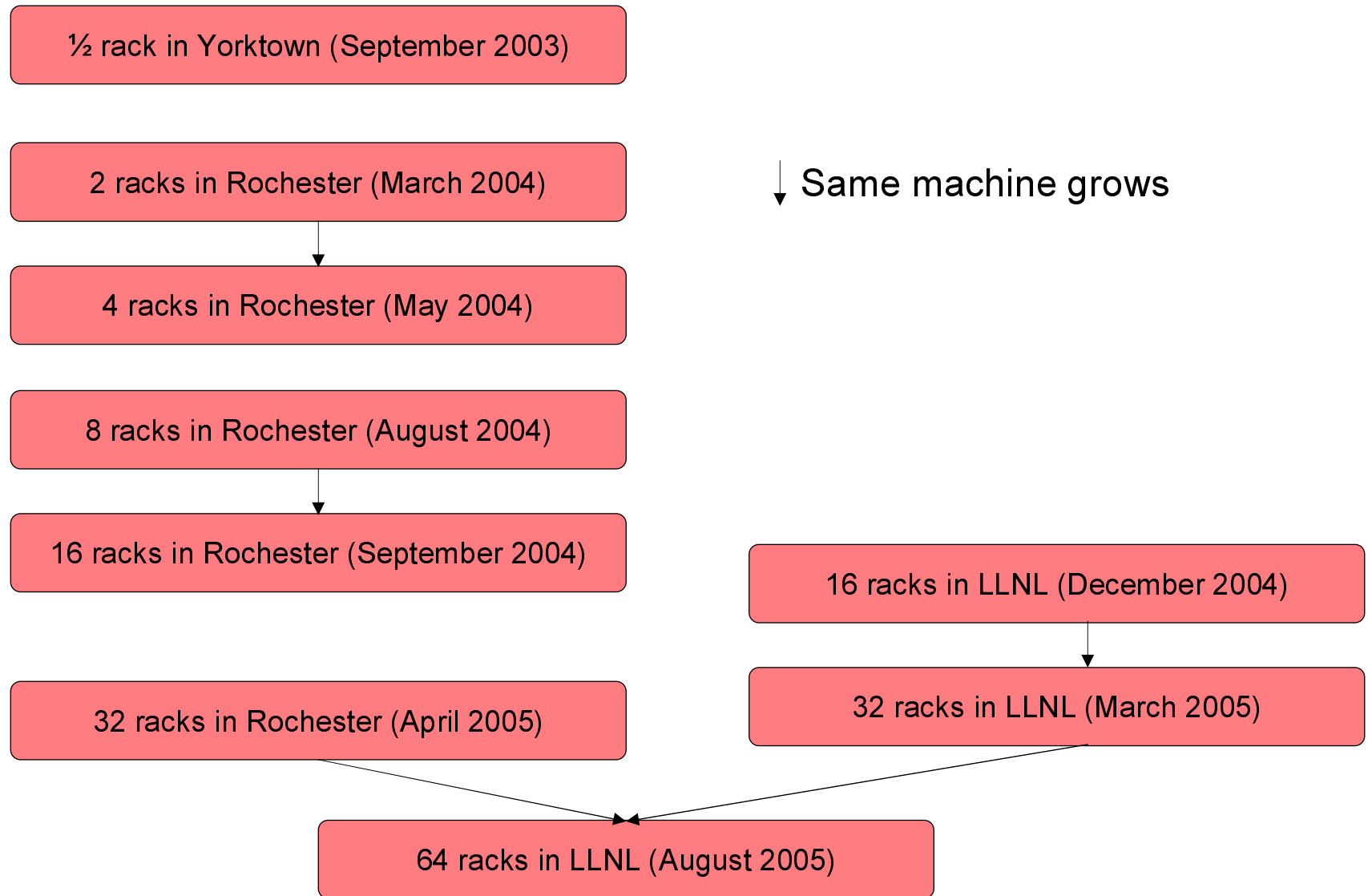


7 bulk power supplies
(AC/DC) – 6 are enough

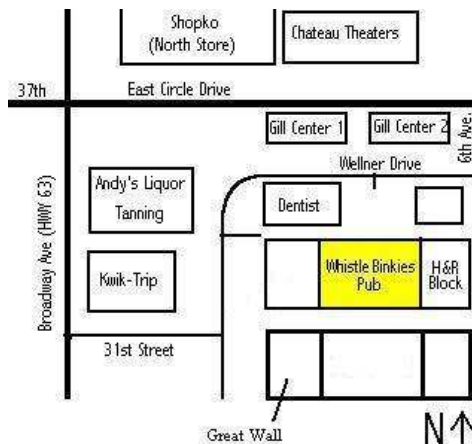
4 DC/DC converts per node card – 3 are enough

- Similar redundancy for fans
- DRAM is 144-bit wide for 128-bit of data (single-bit correct/double-bit detect)
- Torus and collectives network links are protected by two-level CRC
- Partitions can be created that by-pass bad hardware

Blue Gene/L – an exercise in doubling (1 rack = 1024 nodes)



Doubling parties at Whistle Binkies – Rochester, MN



Phone: 507.289.9200
Address: 3210 Wellner Drive NE
Rochester, MN 55906



What are the weaknesses of Blue Gene/L?

- From a spectrum of applications that we can run, small node memory has been an issue:
 - *Our current systems have 512 MB/node or 256 MB/processor*
 - *Most current systems have GBs/processor*
 - *Many times the problem is more perception than reality, but it is real*
- Compilers still have difficulties in exploiting the double floating-point-unit
 - *Expectations are high because hand-tuned code has done well (2x performance)*
 - *Aliasing, alignment make it difficult to exploit in compiler*
 - *We have made progress, but we need to set expectations right*
- Reduced functionality of the lightweight compute node kernel
 - *Necessary to get us where we are*
 - *But, there are codes (and tools) that need features we don't have*

What does the future hold?

- Software Release 2 is on its way (November/2005)
 - *Even more MPI optimizations (particularly regarding selection of algorithms)*
 - *Upgrade of I/O node Linux to 2.6*
 - *Message-queue debugging support (for TotalView)*
 - *Better security*
- IBM HPC software stack for Blue Gene/L (November/2005)
 - *GPFS (native), LoadLeveler (native), CSM, ESSL*
- New IBM XL compilers for Blue Gene/L (early 2006)
- Other improvements as we work with our customers

Conclusions

- Blue Gene/L is an exciting project – we feel that every day we are pushing the limits of scalability
- It was a real “systems” project – all the pieces had to fit together for it to work (architecture, packaging, chips, software, manufacturing, applications)
- The most rewarding factor (at least for me) has been to see scientists and engineers getting their job done by using the machine
- Customer and community excitement has been very high
- We have a mission to continue improving Blue Gene/L, particularly as required by customers