BlueGene/L Architecture Motivations and Decisions

Alan Gara
Outline

- Historical View of Project Motivations/Origins
- Major Architectural Decisions
- Brief Architecture Overview
- A Few comments on the Future
BlueGene/L Pedigree

QCDSP (600GF based on Texas Instruments DSP C31)
  Gordon Bell Prize for Most Cost Effective Supercomputer in '98
  Columbia University Designed and Built
  Optimized for Quantum Chromodynamics (QCD)
  12,000 50MF Processors
  Commodity 2MB DRAM

QCDOC (20TF based on IBM System-on-a-Chip)
  Collaboration between Columbia University and IBM Research
  Optimized for QCD
  IBM 7SF Technology (ASIC Foundry Technology)
  20,000 1GF processors (nominal)
  4MB Embedded DRAM + External Commodity DDR/SDR SDRAM

Blue Gene/L (180/360 TF based on IBM System-on-a-Chip)
  Designed by IBM Research in IBM CU-11 Technology
  64,000 2.8GF dual processors (nominal)
  4MB Embedded DRAM + External Commodity DDR SDRAM
BlueGene/L Project Motivations

- In 1999 IBM announces a $100M plan to achieve a PF
  - Simulation of the folding of a protein as a “killer application” driver for new architectures

- Clear that the dependence on supercomputers in many areas of science is accelerating
  - Dominance of big metal is beginning to fade. Power efficiency is becoming critical
  - Linux clusters and “white” boxes on the rise.
Supercomputer Power Efficiencies

Year

GFLOPS/Watt

QCDSP
Columbia

QCDOC
Columbia/IBM

Blue Gene/L

ASCI White
Power 3

ASCI Q

NCSA, Xeon

ECMWF, p690

Power 4+

Earth Simulator

LLNL, Itanium 2
Power Efficiency?

Steam Iron
5W/cm²

Opportunity
BlueGene/L Project Research Target

- Address as broad as possible a set of applications while maintaining the cost/performance and power/performance of special purpose machines.
  - Many special purpose machines had been very successful despite the incredible technical barriers.
- Applications running on supercomputers do scale fairly well.
  - Growing volume of such applications
  - Physics is local
  - Darwinian selection of applications/algorithms is strong for supercomputers
- Complexity and power are major driver for cost and reliability
  - Simplicity was our mantra from the beginning
  - Choose the right areas to innovate (risk management is paramount)
  - Strong focus on RAS
  - Integration is key (SOC is enabling technology here)
- Software architecture must enable users to exploit hardware.
  - Scaling will be a challenge and users at high end are very good at understanding hardware architectures and exploiting but they must be enabled.
  - Minimal system interference in application code path is best.
BlueGene/L Project Definition Team (First 12 months)

- Core Architecture team of ~ 12 people
  - 3 previous Gordon Bell recipients
  - Most had extensive supercomputer application experience
  - Majority of team members had extensive hardware, software and applications experience (very broad skills)
  - World class packaging
  - Tremendous management support
  - Members of the BlueGene science team were integral part of architecture decisions and discussions

- Nearly immediate engagement with some outside partners.
  - LLNL was very supportive both financially and technically
  - SDSC played a very important role early on and has continued.

- Team Followed project through all stages
  - Architecture, Design, Verification and Bringup of Prototype
# BlueGene/L Project History

<table>
<thead>
<tr>
<th>Jan’00</th>
<th>Jan’01</th>
<th>Jan’02</th>
<th>Jan’03</th>
<th>Jan’04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLL design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLL verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLL FAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLC design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLC verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLC FAB (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-way bring-up</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128-way bring-up</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512-way bring-up</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLC FAB (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Reviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**First chips**
BlueGene/L Major Architectural Decisions

- Networks
- Processor
- Memory System
- Software
- Packaging
BlueGene/L Major Architectural Decisions (Networks)

- The machine was defined around the network capabilities
  - First 3 months were focused on networks and packaging
- To achieve good application performance we needed to offset moderate single process performance with exceptional network performance.
  - Application experience played a crucial role in defining networks.
    - Torus – workhorse, general purpose
    - Combining – Global operations and broadcasts
    - Interrupt – Initially for fast system halt, also useful for user barrier
    - Ethernet – commodity file system connection
    - Jtag – used to configure, boot and monitor
- Unprecedented scaling requirement resulted in architectural evaluation tools with unprecedented scalability and resolution
  - Full 64k node torus network was simulated with all control and data flow accurate to approximately a byte-clock time.
  - Simulations utilized extensively to make architectural design choices.
BlueGene/L - Five Independent Networks

3 Dimensional Torus
- Point-to-point

Collective Network
- Global Operations

Global Barriers and Interrupts
- Low Latency Barriers and Interrupts

Gbit Ethernet
- File I/O and Host Interface

Control Network
- Boot, Monitoring and Diagnostics
Injection Control

**Single(k):** Inject when link is available and VC has at least k tokens

**All(k):** Inject when link is available and all VCs have at least k tokens

**None:** Inject when link is available and VC has tokens (=Single(8))

64K BlueLight (64x32x32): 20 KB/msg to 6 Neighbors
Sparse Solver with Random Mapping onto Torus
Use of selective by-pass path does not significantly affect throughput with enhanced routing under random overload and hot region traffic.
BlueGene/L Major Architectural Decisions (Processor)

- Processor must have strong floating point performance and have efficient interface for communicating to other nodes.
  - This resulted in the double floating point unit and the quad-load capability.
  - Double floating point unit primarily added to allow quad load capability.

- Users will aggressively tune both single node and multi-node performance.
  - Early efforts to develop a DMA engine for messaging resulted in a large area impact and increasingly complex design. A second processor was added to allow for the overlapping of communications and computation.

- System-on-a-chip has a menu of processor options.
  - Embedded processors have many of the same attributes needed for supercomputing.
    - Highly power efficient
    - Flexible “accelerator “APU port
    - Small silicon footprint
BG/L performance history

- Peak
- Linpack
- Climate Code

- 442 dies
- double 440s
- original bandwidth
- conceived
- double bandwidth
BlueGene/L Major Architectural Decisions (Memory System)

- Embedded DRAM allows for clear differentiation.
  - On Chip DRAM enables large bandwidth with relatively low power
  - Large DRAM also allows for some applications working set to live entirely on chip.
- Memory system must support aggressive prefetching of data
  - All cache levels support prefetching allowing one to entirely hide the memory system latency for streaming data.
- Streaming data is a property of some applications.
  - Bandwidth to main store must be balanced
- Networks must be memory mapped for fast user space access.
- Coherent memory outside of 440 cores is important for allowing hand off for messaging to second processor.
Sequential Read Bandwidth

Bytes/pclk vs. Block Size

- L2 enabled, L3 enabled
- L2 disabled, L3 enabled
- L2 enabled, L3 disabled
- L2 disabled, L3 disabled
BlueGene/L Major Architectural Decisions (Software)

- Operating system must not get in the way of applications
  - Simple kernel is all that is needed/wanted on compute nodes.
  - Function ship complexity to I/O nodes where OS can be more complex and decoupled from computation.
- OS must allow for efficient use of all hardware resources via simple, efficient APIs
  - Both APIs and documentation are needed
- Must have highly tuned efficient MPI
  - MPI is the clear leader in supercomputer application space.
- Libraries and Intrinsic functions will be most effective way to leverage double floating point unit.
  - Identify common libraries and exploit double floating point unit. Floating point unit was optimized for linear algebra.
- Performance monitoring and evaluation tools are critical to enabling users to exploit machine
  - Without the necessary monitoring tools users will likely be frustrated angry and disappointed.
BlueGene/L Major Architectural Decisions (Packaging)

- Packaging constraints are severe.
  - Power and space are precious
  - BlueGene/L density allows for 7/8 of the torus links to be contained inside a midplane avoiding many cables.
- Signaling technology is one of the enabling technologies for BlueGene/L
  - BlueGene/L team developed their own high speed links which exceeded all available links in performance/watt and performance/mm^2
- Cooling innovations were required to allow for 1000 nodes in a single rack.
  - Tilted plenums
- Electrical noise environment is enormously challenging
BlueGene/L Architectural Summary
BlueGene/L

System
(64 cabinets, 64x32x32)

Cabinet
(32 Node boards, 8x8x16)

Node Board
(32 chips, 4x4x2)
16 Compute Cards

Compute Card
(2 chips, 2x1x1)

Chip
(2 processors)

2.8/5.6 GFLOPS
4 MB

180 GFLOPS
16 GB DDR

11.2 GFLOPS
1.0 GB DDR

5.7 TFLOPS
512 GB DDR

360 TFLOPS
32 TB DDR

5.6 GFLOPS
4 MB

1.0 GB DDR

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

5.6GF peak node

PLB (4:1) 2.7GB/s

32k/32k L1
440 CPU
“Double FPU”

440 CPU
I/O proc
“Double FPU”

32k/32k L1

L2
snoop
128

256

L2

Shared L3 directory for EDRAM
Includes ECC

4MB EDRAM
L3 Cache or Memory

1024+ 144 ECC
22GB/s

256

DDR Control with ECC

6 out and 6 in, each at 1.4 Gbit/s link

Collective

3 out and 3 in, each at 2.8 Gbit/s link

Global Barrier

4 global barriers or interrupts

5.5 GB/s

5.5GB/s

11GB/s

DDR
512MB

144 bit wide

Ethernet Gbit

JTAG Access

Torus

Gbit Ethernet

JTAG

6 out and 6 in, each at 1.4 Gbit/s link

32k/32k L1

128

256

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

© 2004 IBM Corporation

Alan Gara IBM Research
3-D Torus Network

- 32x32x64 connectivity
- Backbone for one-to-one and one-to-some communications
- 1.4 Gb/s bi-directional bandwidth in all 6 directions (Total 2.1 GB/s/node)
- 64k * 6 * 1.4Gb/s = 68 TB/s total torus bandwidth
- 4 * 32 * 32 * 1.4Gb/s = 5.6 Tb/s Bisectional Bandwidth
- Worst case hardware latency through node ~ 69nsec
- Virtual cut-through routing with multipacket buffering on collision
  - Minimal
  - Adaptive
  - Deadlock Free
- Class Routing Capability (Deadlock-free Hardware Multicast)
  - Packets can be deposited along route to specified destination.
  - Allows for efficient one to many in some instances
- Active messages allows for fast transposes as required in FFTs.
- Independent on-chip network interfaces enable concurrent access.
Collective Network

- **High Bandwidth one-to-all**
  - 2.8 Gb/s to all 64k nodes
  - 68 TB/s aggregate bandwidth
- Arithmetic operations implemented in tree
  - Integer/ Floating Point Maximum/ Minimum
  - Integer addition/subtract, bitwise logical operations
- Latency of tree less than 2.5 usec to top, additional 2.5 usec to broadcast to all
- Global sum over 64k in less than 2.5 usec (to top of tree)
- Used for disk/host funnel in/out of I/O nodes.
- Minimal impact on cabling
- Partitioned with Torus boundaries
- Flexible local routing table
- Used as Point-to-point for File I/O and Host communications
Fast Barrier Network

- **Four Independent Barrier or Interrupt Channels**
  - Independently Configurable as "or" or "and"
- **Asynchronous Propagation**
  - Halt operation quickly (current estimate is 1.3 usec worst case round trip)
    - > 3/4 of this delay is time-of-flight.
- **Sticky bit operation**
  - Allows global barriers with a single channel.
- **User Space Accessible**
  - System selectable
- **Partitions along same boundaries as Tree, and Torus**
  - Each user partition contains it's own set of barrier/ interrupt signals
Control Network

**JTAG interface to 100Mb Ethernet**
- direct access to all nodes.
- boot, system debug availability.
- runtime noninvasive RAS support.
- non-invasive access to performance counters.
- Direct access to shared SRAM in every node.
Ethernet Disk/Host I/O Network

Gb Ethernet on all I/O nodes
- Gbit Ethernet integrated in all node ASICs but only used on I/O nodes.
- Funnel via global tree.
- I/O nodes use same ASIC but are dedicated to I/O Tasks.
- I/O nodes can utilize larger memory.

Dedicated DMA controller for transfer to/from Memory
Configurable ratio of Compute to I/O nodes
- I/O nodes are leaves on the tree network
Future Directions
Technology/Architecture Redirection

- It is real and happening now.
- Aggressively exploiting parallelism allows for mitigation of some power issues. Special purpose machines may strongly differentiate.
- Single thread performance constraints and expectations will drive much of the commercial direction. Will add a severe constraint to commercial system evolution.
- Innovation at the architectural and technology levels is critical.
How do we move forward

- Close collaboration on high end systems with select partners.
  - Parallelism is available. Less emphasis on single thread performance. (It is still very important)
  - Users are accustomed to leveraging “unique” hardware if there is value.
  - Allows for solutions that can influence commercial direction.
  - Detailed analysis based on real applications

- Node architecture must be accompanied by a balanced network solution
  - Latency promises to be the biggest network challenge for the future.
Conclusion

- BlueGene/L architecture has succeeded in large part due to a solid initial direction set forth by a small dedicated team.
  - This type of environment is very difficult to orchestrate and/or repeat through management or money.
- The application input from collaborators was critical to this machine being applicable to real problems.
  - Together we must build something ultimately of value to both partners.
- The next 10x improvement is going to be much harder than the previous.