

Blue Gene/L

A Short Tour Through a New Research Computing Hardware Architecture

**Internet2 Workshop on Effective Approaches to
Campus Research Computing Cyberinfrastructure**

April 26, 2006



Blue Gene/L:
A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 1

Outline

- Blue Gene Architecture
Patrick Dreher, MIT
- Blue Gene Implementation
Curtis Hillegas, Princeton



Characteristics of Beowulf-Class Computers

- Beowulf cluster designs entered HPC community in the mid 90's and have grown and evolved in size and complexity
- Common design themes for these HPC hardware architectures
 - > Cost-effective, mass-market, commodity off-the-shelf technologies utilizing rapidly evolving commodity silicon technology microprocessor CPU and memory hardware
 - > Commodity networking facilitating the design of distributed-memory systems with tolerable bandwidths and latencies
 - > Free or inexpensive operating systems, such as Linux, distributed with reliable, supported, and complete source code
 - > Deliver the largest number of flops at the lowest possible price

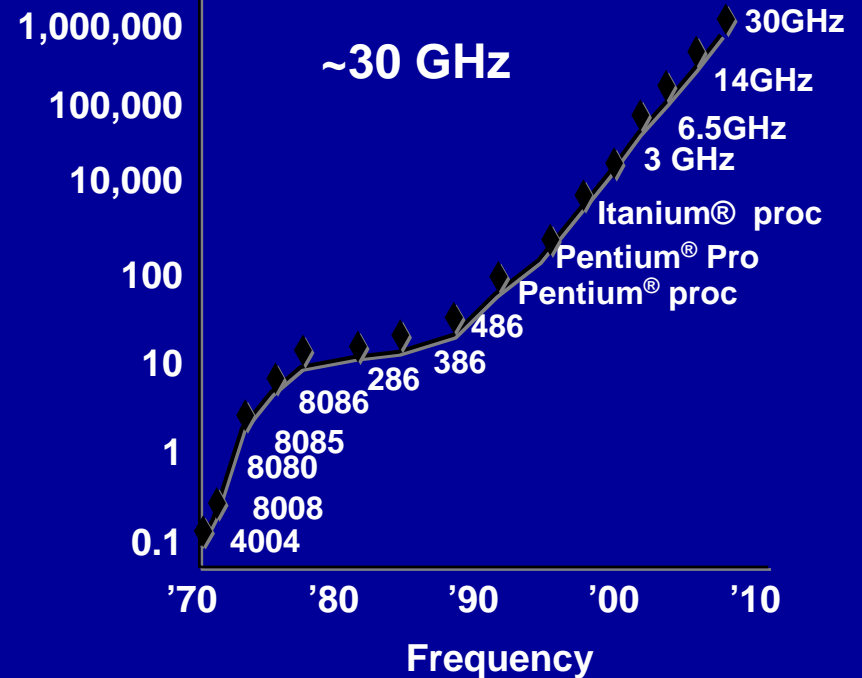
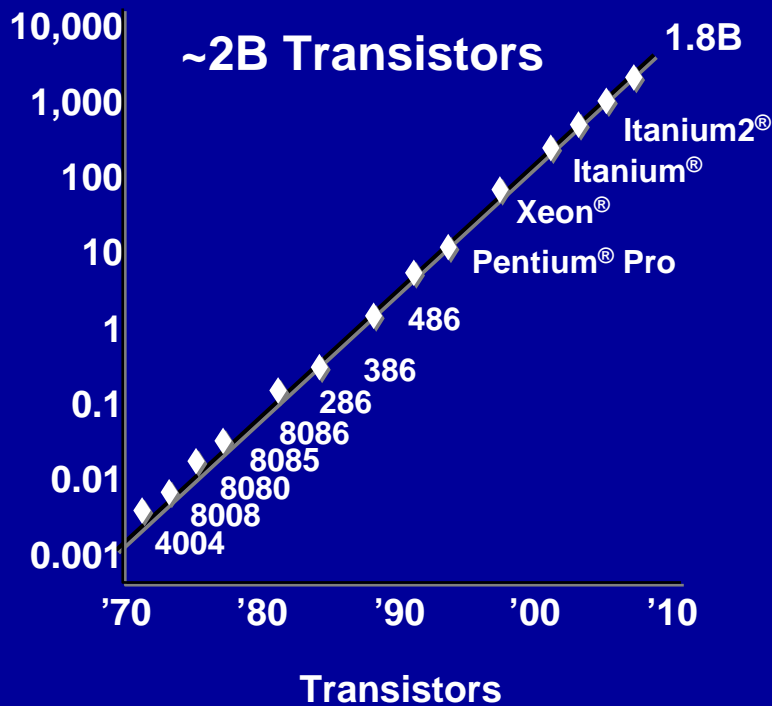


Cluster Infrastructure Requirements

- Large Beowulf clusters present serious hardware networking and infrastructure challenges such as increasing transistor density and clock frequency leading to more heat dissipation
- Each Beowulf type rack may require > 10 kW to power the racks plus an additional 4-5 kW for cooling (~40% beyond power that must be delivered to the racks).
- The greater the number of servers in each rack, and the more total racks, the greater the challenges for network and communications requirements for servers integrated into this hardware environment



Transistor Density and Clock Frequency by the End of the Decade



Blue Gene/L:

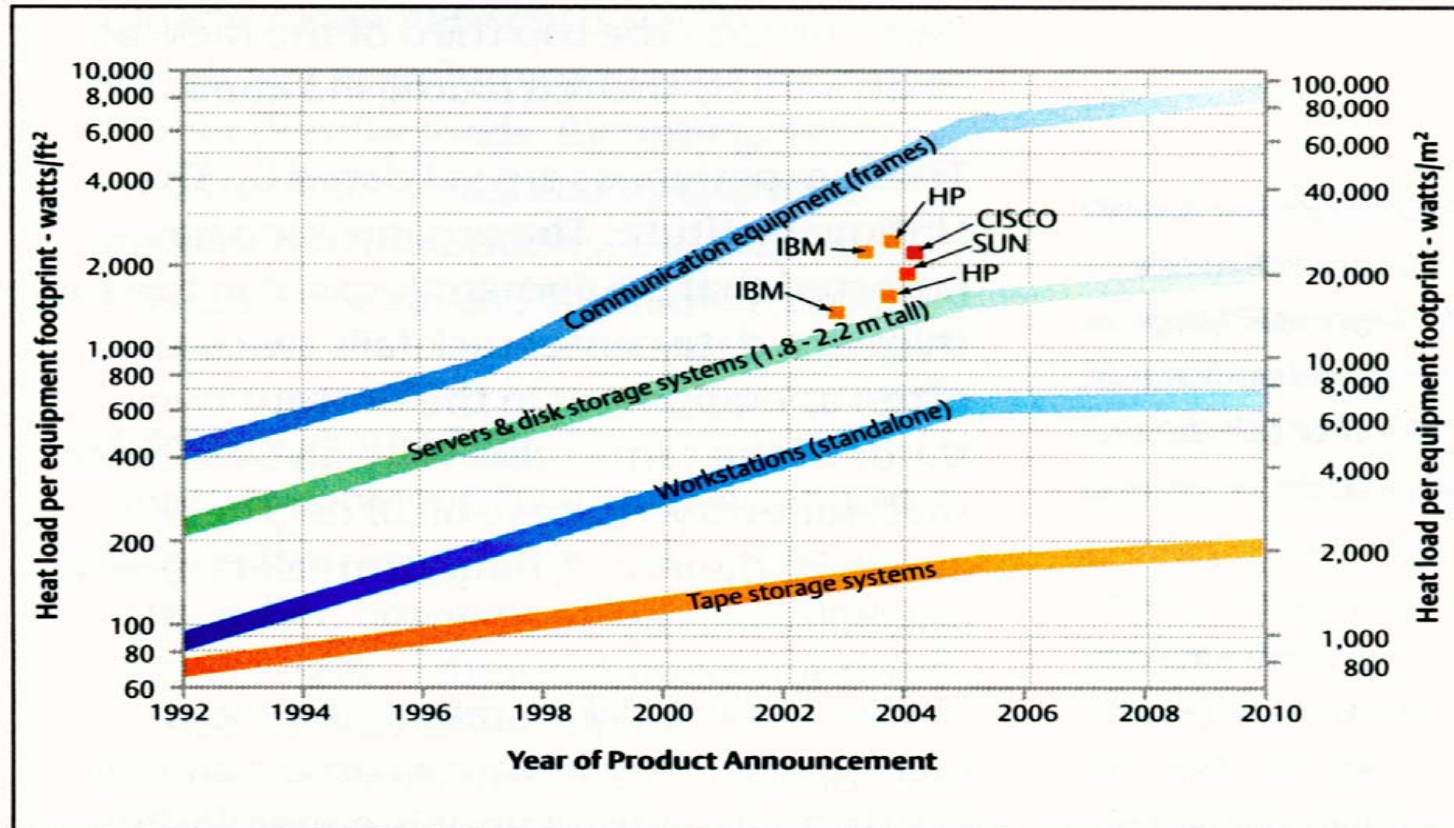
A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 5



Computer Hardware Equipment Density

Source Liebert White Paper and the Uptime Institute (2000)



Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 6



New Architecture Design

- Blue Gene was developed to increase computation capabilities through a new hardware design and to address the infrastructure issues
- Used system on a chip technology that integrated all functions of a node (except main memory) onto a single Application Specific Integrated Circuit (ASIC)



Blue Gene ASIC

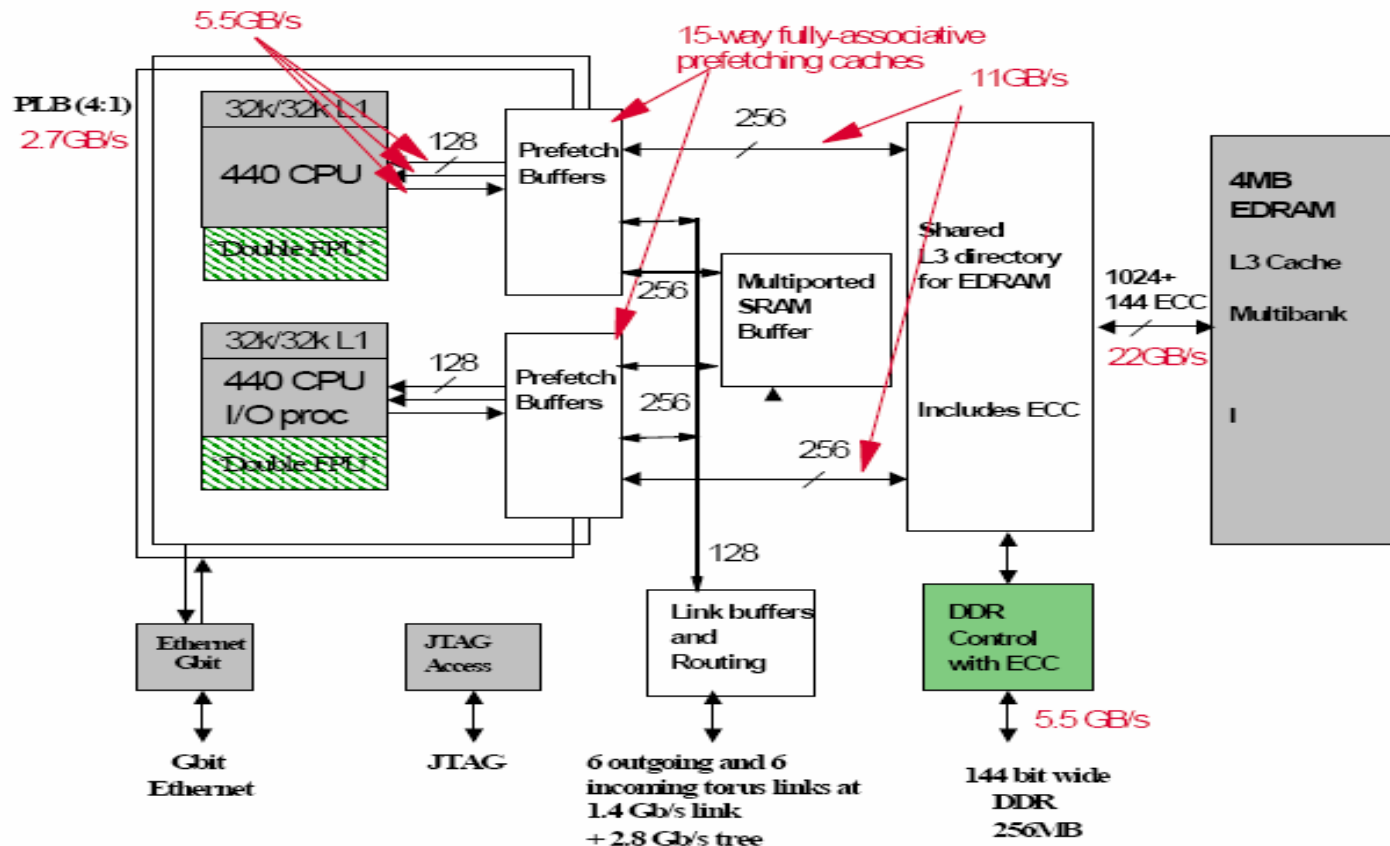
- Two PowerPC cores each with 64-bit “double” FPU that can operate in SIMD mode
- Each FPU
 - > Can execute two multiply-adds per cycle
 - > Has its own instruction and data cache (32KB)
 - > Has a small L2 pre-fetch buffer and 4 megabit shared L3 built from embedded DRAM*
 - > Has a DDR memory controller
- Clock rate of the Blue Gene/L is 700 MHz

* Each core lacks Session Management Protocol and so the two cores are not L1 cache coherent but the L2 and L3 caches are coherent between the two cores



Blue Gene/L Node Diagram

An Overview of the Blue Gene/L Supercomputer SC2002



Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 9



Blue Gene/L Power and Cooling

- Each node has 5.6 gigaflops peak
- Each ASIC consumes 12 watts of power
- These individual nodes can be assembled to construct a standard rack footprint on the machine room floor with 5.6 TFlops CPU peak that dissipates ~ 24 kw of power.

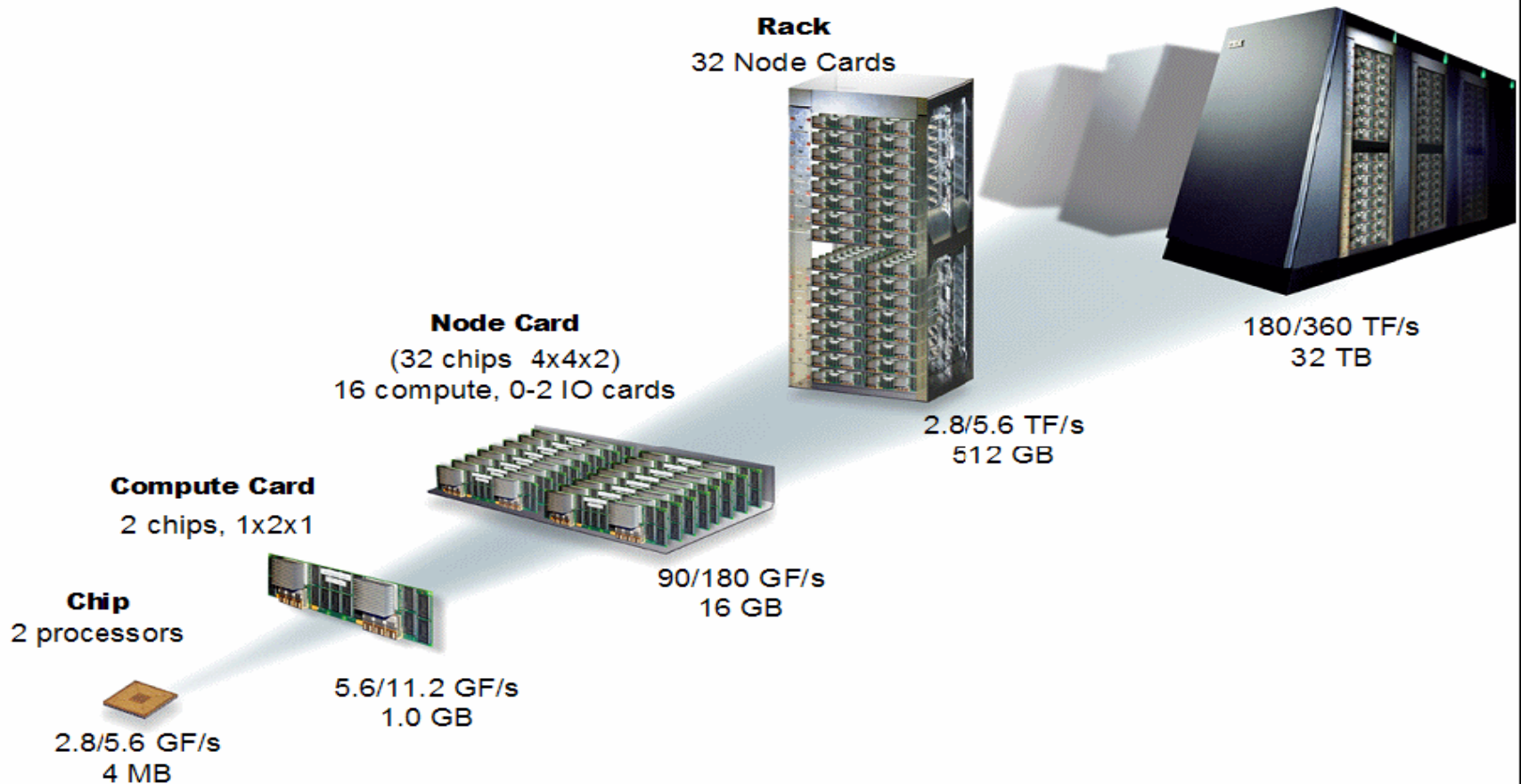


Blue Gene Building Blocks

- Each Blue Gene chip has
 - > 2 processor embedded cores – 5.6 GF/s – 4 MB cache
 - > Integrated embedded DDR memory controller
 - > Gigabit ethernet adapter
 - > Network cut through buffer and controls
- Each compute card has
 - > 2 chips 1x2x1 – 11.2 GF/s – 1.0 GB
- Each node card has
 - > 16 compute cards w/ 0-2 I/O cards (32 chips)
 - > 180 GF/s – 16 GB
- Each rack has
 - > 32 node cards w/ 5.6 TF/s
- Maximum of 64 racks delivers 360 TF/s w/32 TB



Blue Gene



Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 12



Getting Around Inside the BG/L

- The primary network topology of the Blue Gene/L is a 3 dimensional torus
- Different topology from classic switched network
- Torus network topology offers
 - > higher bandwidth
 - > lower latency
- But these gains come at the cost of optimizing a complicated communications problem



I/O Node Hardware Configuration

- Similar hardware to compute nodes with the exceptions that
 - > I/O nodes do not attach to the 3-d torus
 - > No interchip wires for the torus network
- I/O nodes only run system software
- Compute nodes exist under the I/O nodes
- Routing -- (class and collective network)
 - > Sender places recipient address into packet
 - > Data is sent up/down the collective network until address in packet from compute/I-O node matches the I-O/compute node



Understanding the Network(s)

- BG/L has 5 networks in the machine
 - > 3-d torus point-to-point messaging network between compute nodes
 - > A collective network for global operations
 - > A global barrier/interrupt network
 - > A GigE ethernet JTAG (Joint Task Action Group) for machine control
 - > Another GigE network for external connectivity to other systems and servers



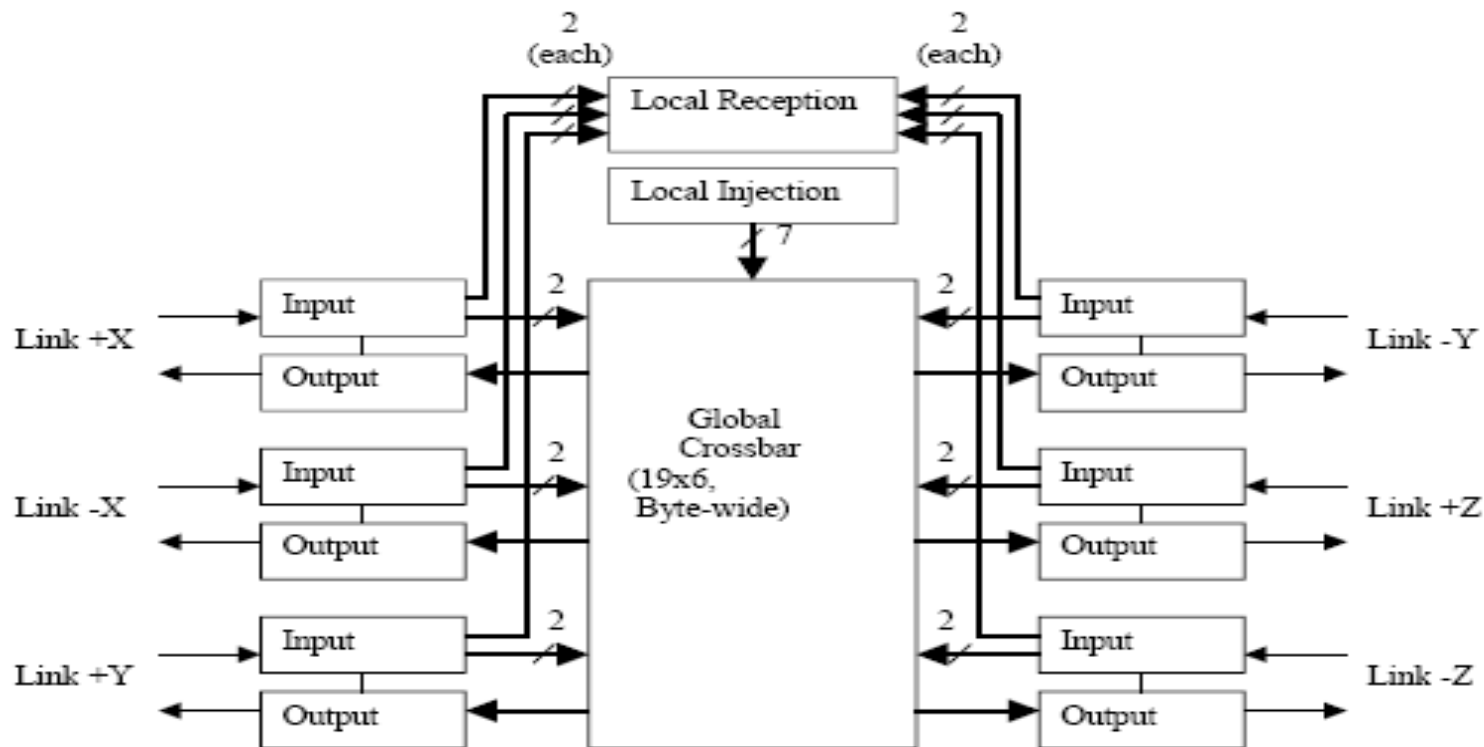
Communications on the Torus Interface fifos

- CPUs access the torus via the memory mapped torus fifos (1 kbyte of SRAM each)
- 6 normal – 2 hi-priority injection fifos (not associated with network directions)
- 2 groups of normal priority 6 reception fifos (one for each direction)
- Packet header indicates where the packet should be received
- All fifos have a readable status bit that can be accessed from specific hardware addresses and which indicates fifo capacity status at any given time



Architecture for Torus Route

An Overview of the Blue Gene/L Supercomputer SC2002



Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 17



Torus Communications Code Injection Recipe

- Build a complete packet in memory with 8 byte HW header and content
- Align on a 16 byte boundary of memory (constrained from 32 to 256 bytes)
- Select a torus fifo and continue reading status bits until there is enough space for the packet
- Load first Quad (16 bytes) in a double FPU register and store using the Quadstore
- Continue loading and storing into the selected torus fifo until all bytes are stored



Torus Communications Code Reception Recipe

- Query reception fifo for data to be read
- Use double FPU Quadload to receive first 16 bytes (packet header and size)
- Use double FPU Quadstore to store the 16 bytes in specific memory location
- Continue reading and storing 16 byte increments in memory until finished



Communications on the BG/L

- Define an application to consist of a number of domains
- A domain may be an MPI task, a single variable or a group of variables
- The definition of domain depends on how the communication data is collected
- Computations are performed in one domain before information is accessed from other domains
- Communication traffic is the amount of data exchanged between domains.
- One or more domains are mapped to a single BG/L node

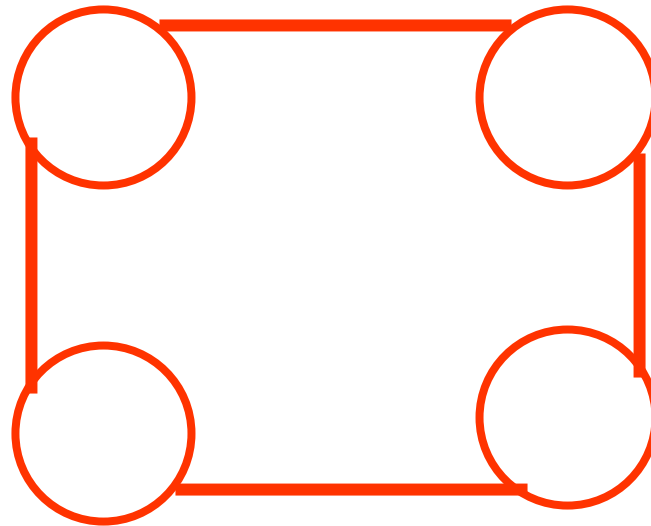


Communications on the BG/L (cont'd)

- The task layout on the machine is an important consideration to computation efficiency
- Challenge is to optimally map an arbitrary parallel application into domains that are assigned to BG/L nodes such that the communications time is minimized
- Basically this is a mathematical mapping problem to assign tasks to processors in a parallel processing computer with the constraint that the application achieves optimal load balance and minimizes inter-processor communications costs



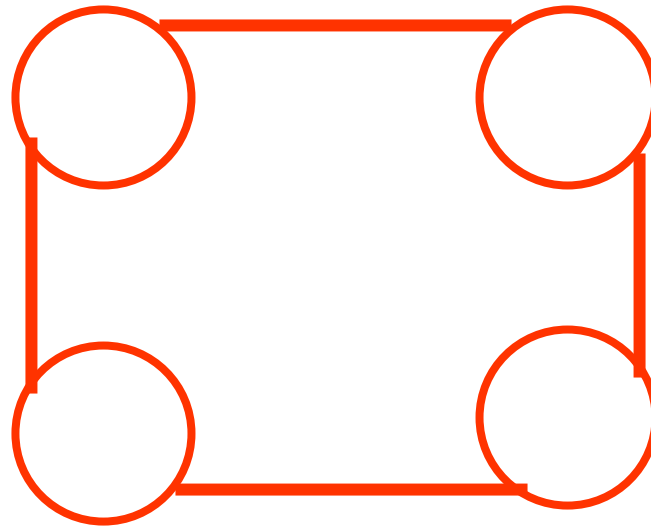
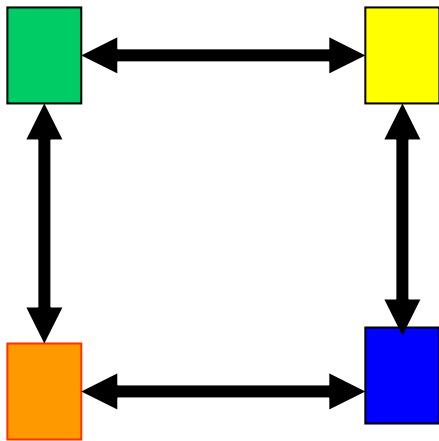
Example: 2-D Nodes on the Torus



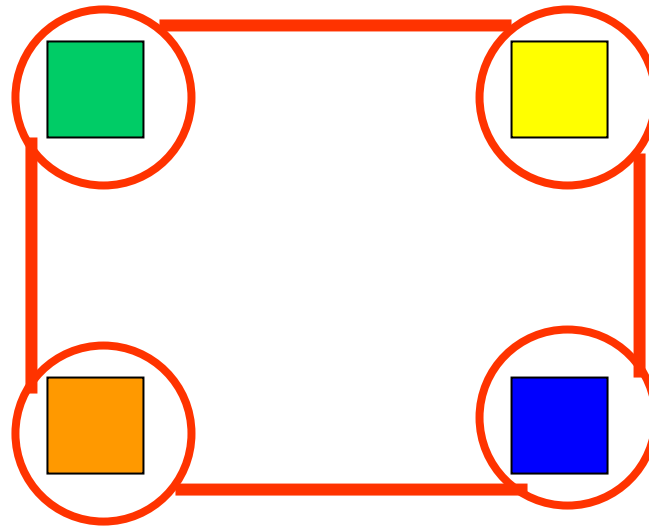
Blue Gene/L:
A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 22

Tasks and Communications Rules to be Mapped to the Nodes on the Torus



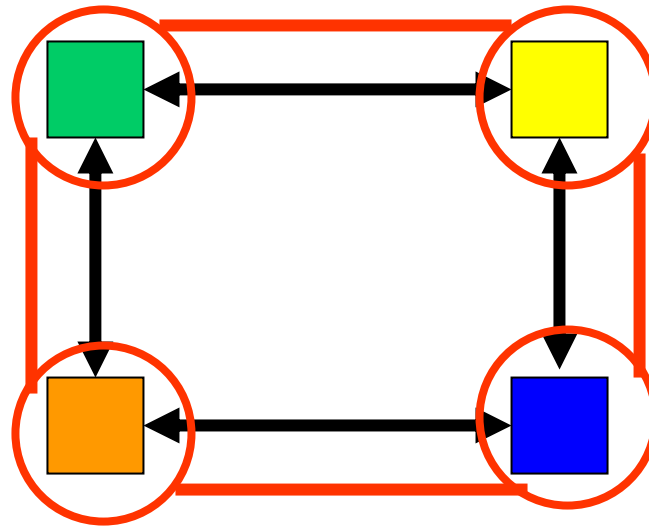
Node to Node Communications by Task Layout



Blue Gene/L:
A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 24

Node to Node Communications by Task Layout



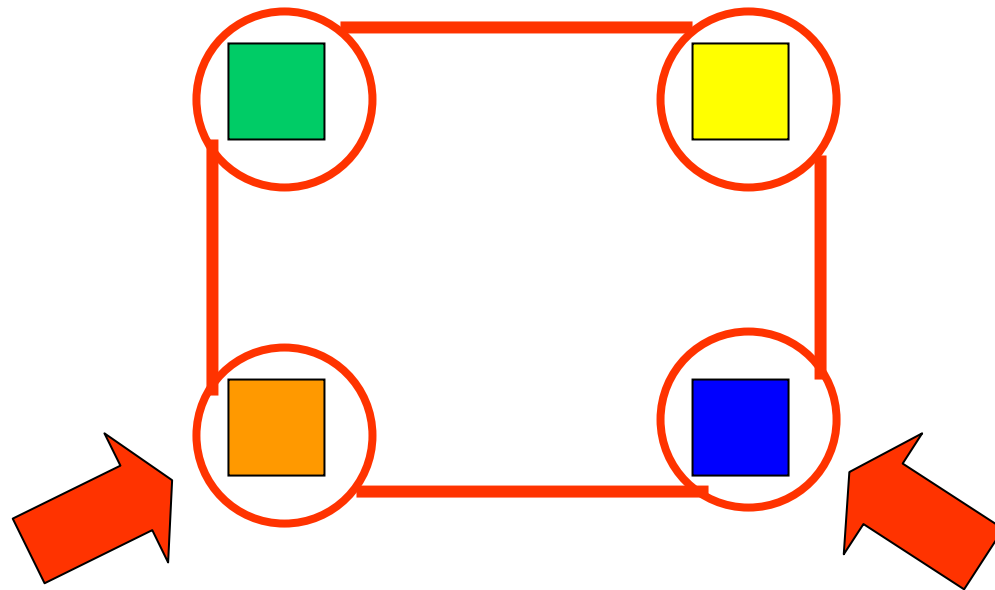
Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 25



Node to Node Communications by Task Layout



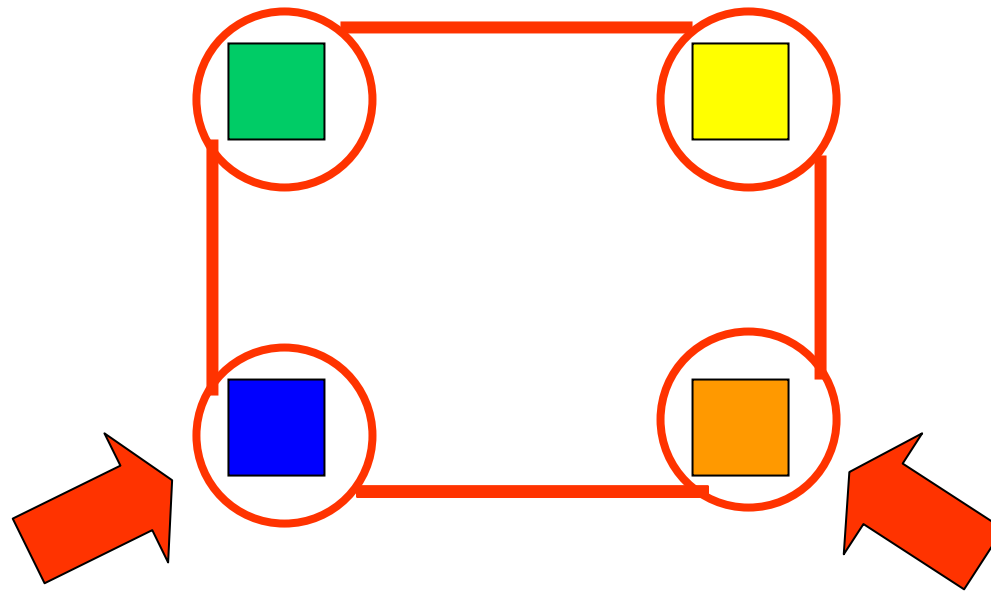
Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

Internet2 Workshop
Apr. 26, 2006 – slide 26



Node to Node Communications by Task Layout



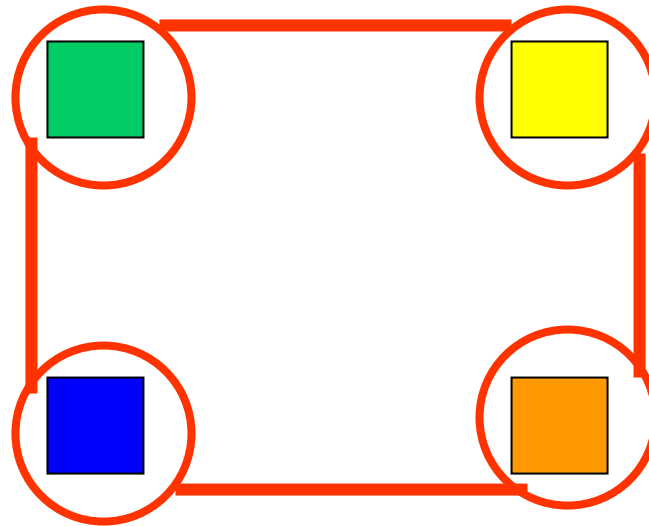
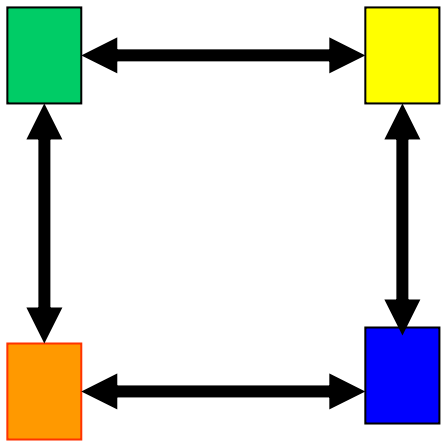
Blue Gene/L:

A Short Tour through A New Research
Computing Hardware Architecture
Patrick Dreher

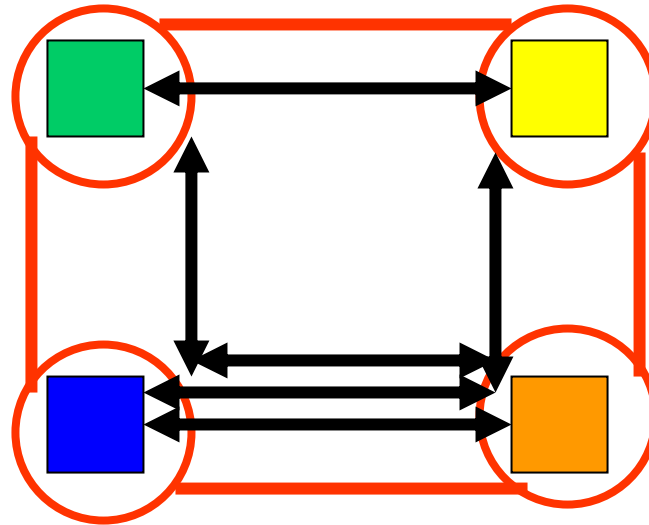
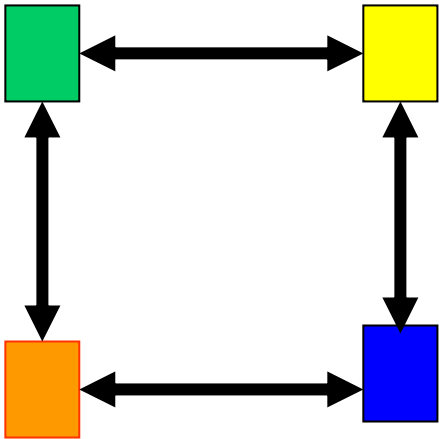
Internet2 Workshop
Apr. 26, 2006 – slide 27



Node to Node Communications by Task Layout



Node to Node Communications by Task Layout



Mathematical Complexity

- Number of maps for n tasks is $n!$
- Example – in 1968 an $n=30$ task problem proposed that took an equivalent of 7 years of computation time on an HP9000 to solve
- Blue Gene has 65536 tasks
- Need a Markov type process to do the optimizations for an application



Comments and Observations

- Potential hardware roadmap toward a petaflop machine (5 Tflops/rack with just 24 kw power dissipation on a standard rack footprint on the machine room floor)
- From a business model perspective for research computing, this machine may not seamlessly interface into campus research computing “condo models” for aggregating clusters
- Source codes and optimizations will likely differ between BG/L machines and Beowulf clusters
- Will require an investment by the researchers in time and effort to capitalize on the potential of this HPC hardware architecture

