SGI® Altix™ Hardware Architecture

Reiner Vogelsang
SGI GmbH
reiner@sgi.com

June 22, 2005

sgi®

SGI Altix 3000

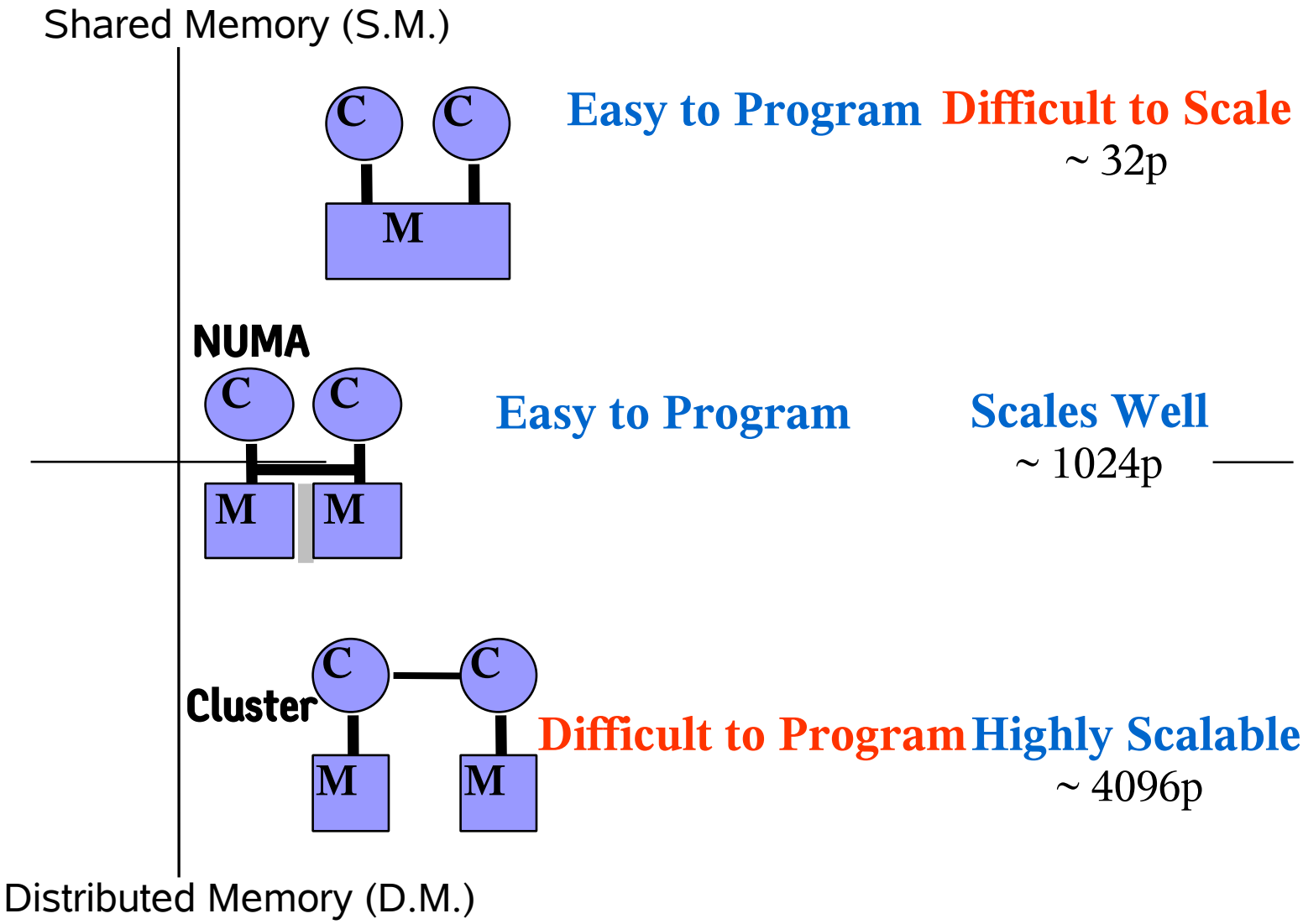
- Introduced January 2003
  - Red Hat- or Suse-Linux compatible Operating System:
  - 512 CPU SSI Linux released
  - Intel Itanium2 processors (Madison) in all variants
  - Over 55000 processors sold, systems from 2 PEs to >512 PEs
  - Huge shared memory
    - Several orders for < 100 PEs with >2 TB shared memory.
  - Two OS Variants:
    - SGI enhanced Red Hat AS 2.1 based Linux OS
    - Standard SUSE SLES 9
  - Most traditional SGI value-adds available:
    - CXFS Client/Server, DMF, MPT,SCSL
    - Multipipe GFX available

SGI ccNuma Balanced System Architecture

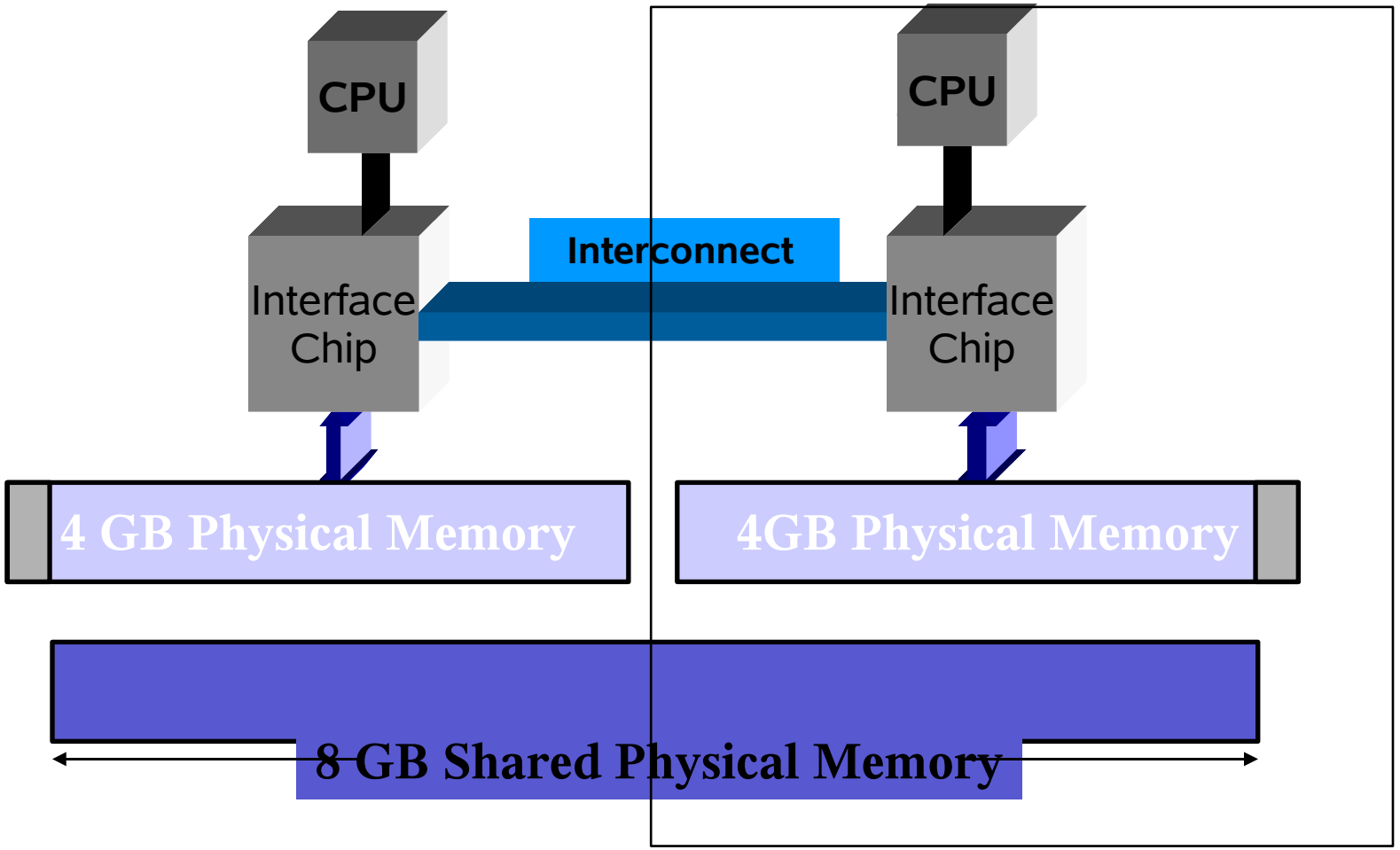
SGI Altix4000

- SGI Altix4000 computer system is characterized by:
  - (Scalable) cache coherent shared memory (SMP)
  - Intel Itanium-2 processors
  - Standard Linux operating system

Parallel Architectures



SGI Scalable ccNUMA Architecture



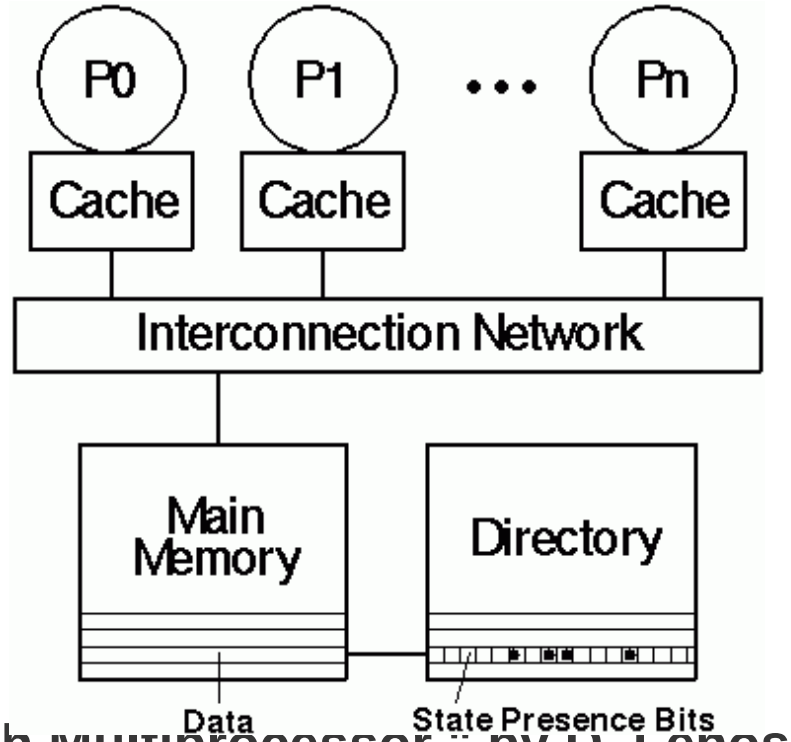
Interconnect: section of interface chip, cables and routers

ccNuma: Distributed Shared Memory

- ccNuma:
  - Memory is physically distributed but logically shared
  - Memory is kept coherent automatically by hardware
  - Coherent memory: memory is always valid (caches hold copies)
  - Granularity is L3 cacheline (128 B)
- Directory memory:
  - For each cacheline access information is stored:
  - Who has valid copies
  - Which processor has write access
  - Hardware revokes access rights automatically
- *In contrast snoopy bus protocols do not scale well*
  - *Access requests are broadcasted*
- Directory information is stored in main memory
  - Directory entry is 4 byte wide for each 128 byte cache line

ccNuma: Distributed Shared Memory

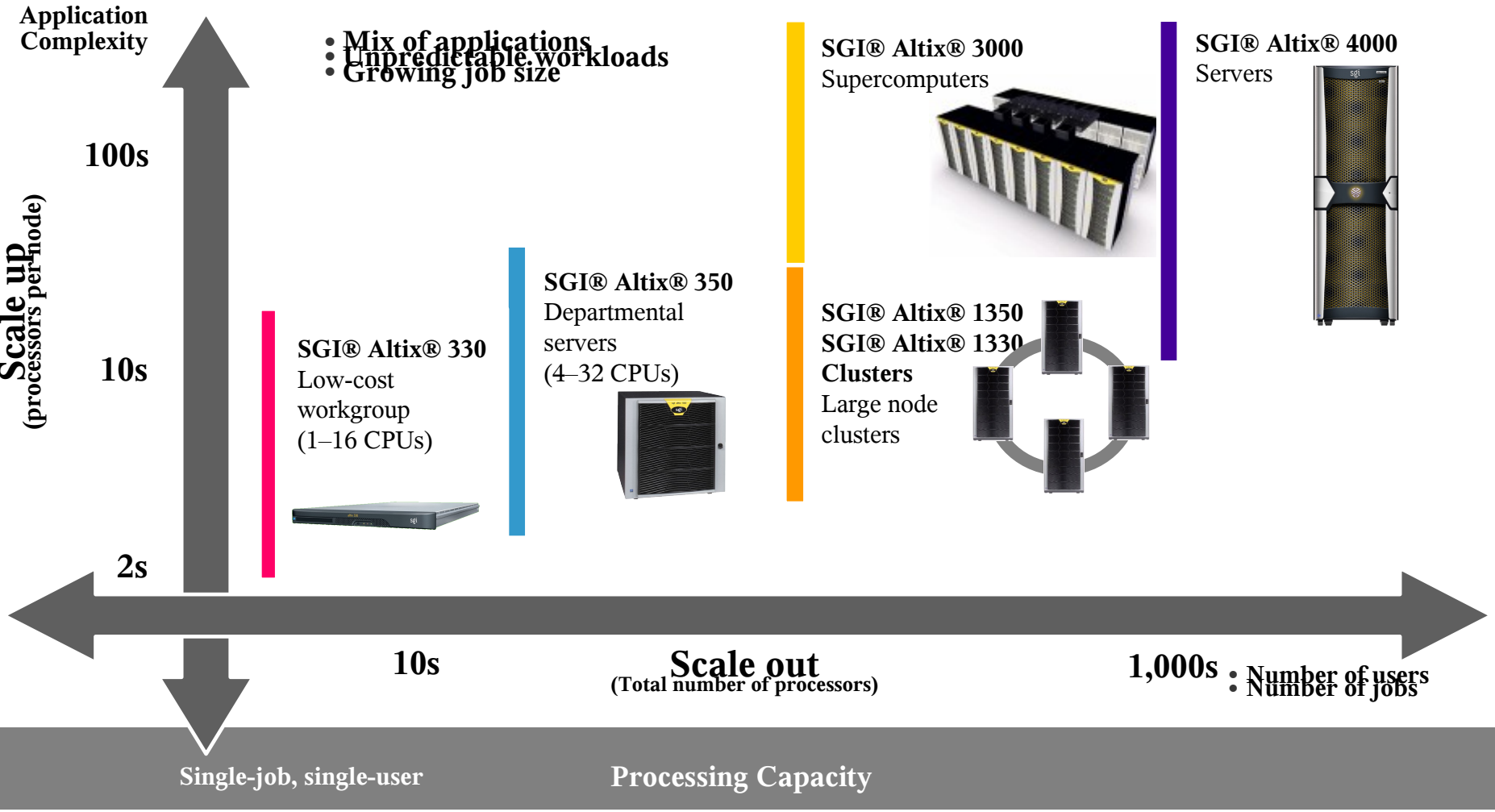
- Schematic view onto a full directory based coherence scheme



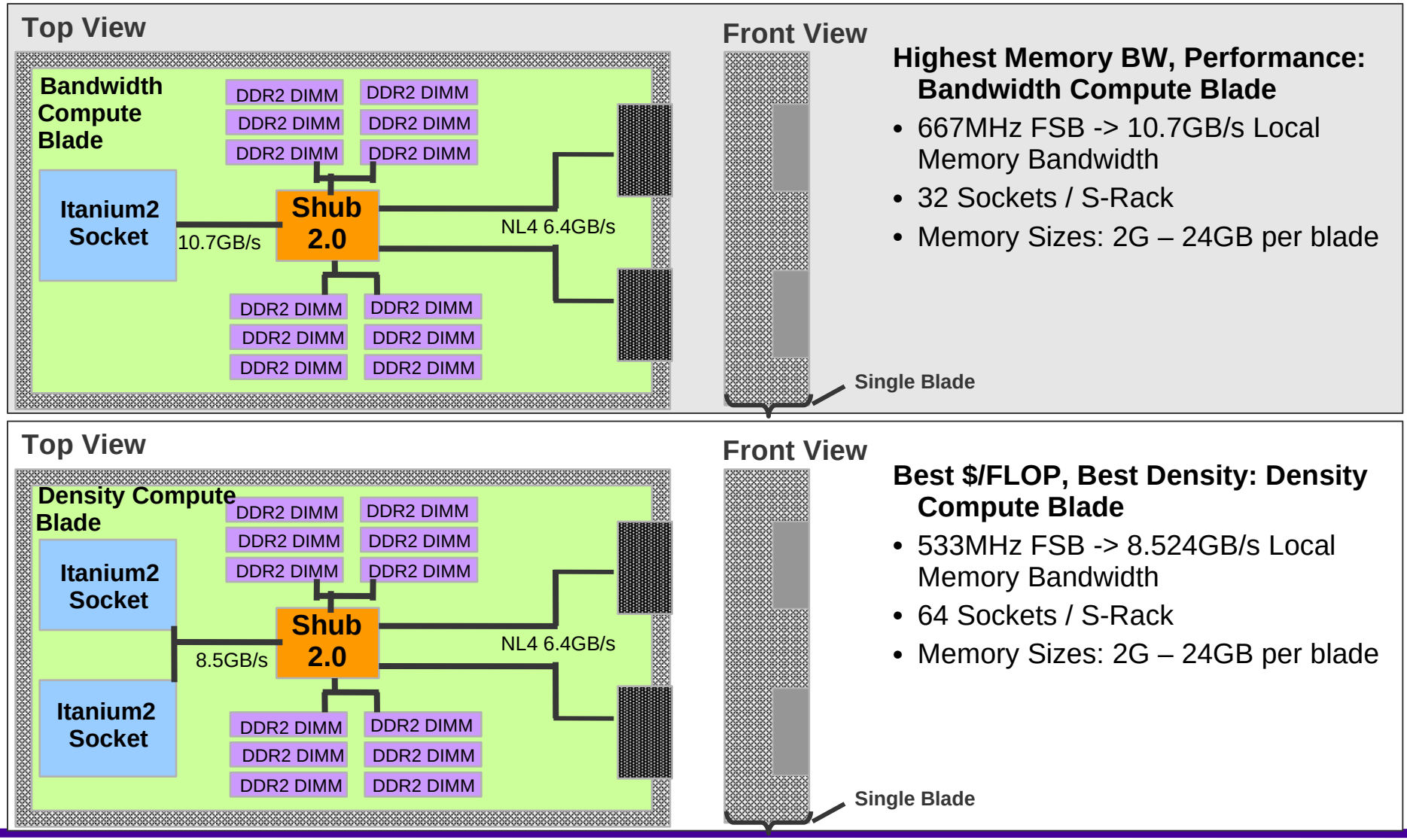
- The Stanford Dash Multiprocessor," by D. Lenoski et al., IEEE Computer, 25(3), March 1992, pp 63-79"
- http://www.cse.ucsd.edu/classes/fa00/cse240/lectures/Lecture18.html

SGI Altix 4000

High-Performance SGI® Altix® Servers and Supercomputers



SGI Altix 4700 Processor Blade

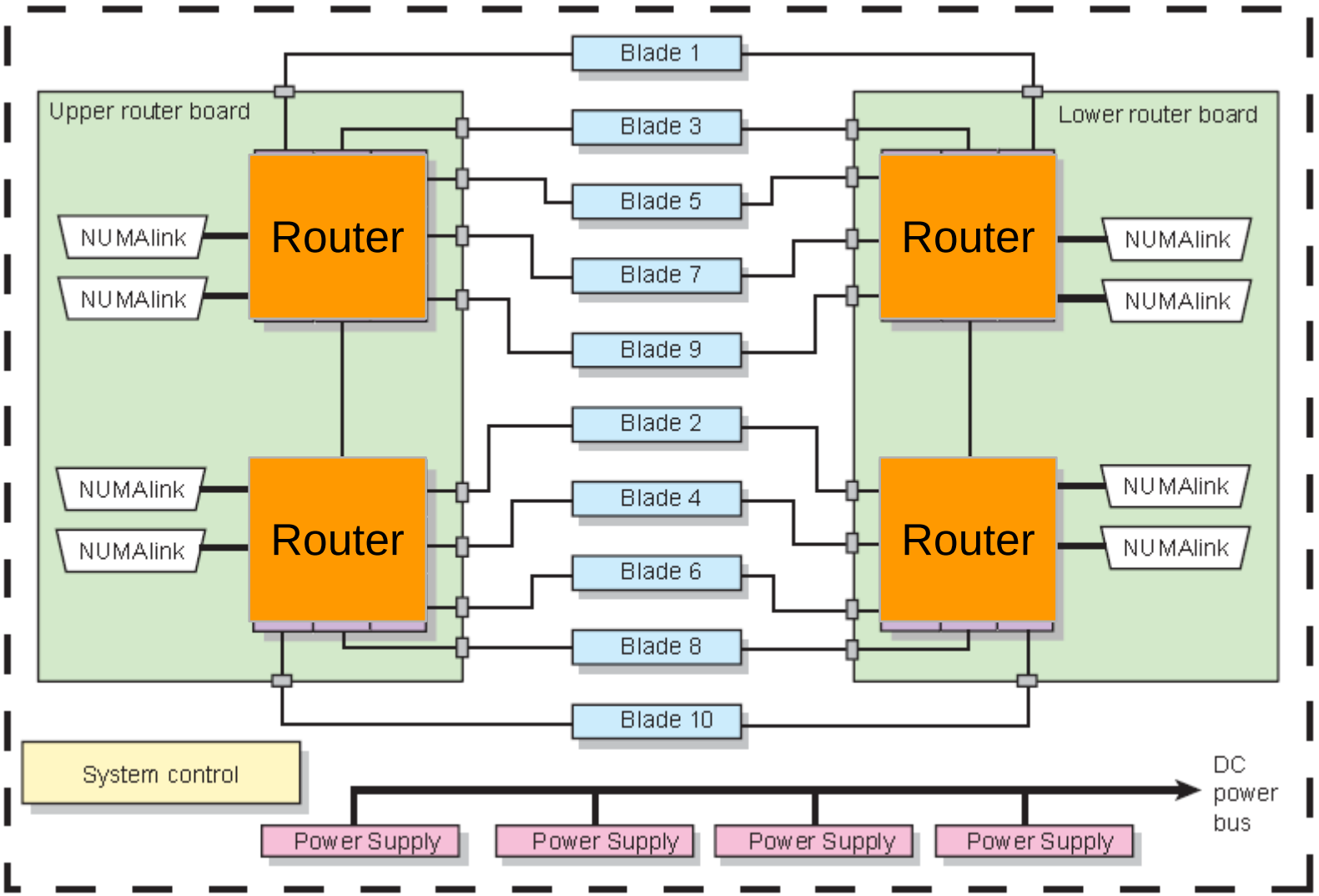


SGI® Altix™ 3000BX2 Memory

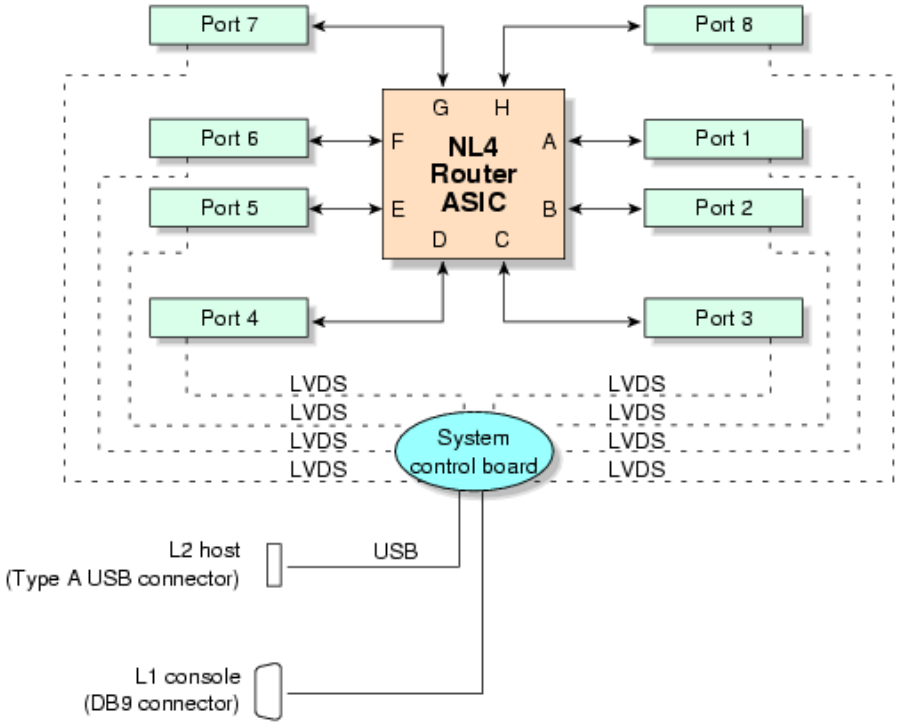
Each CPU module:

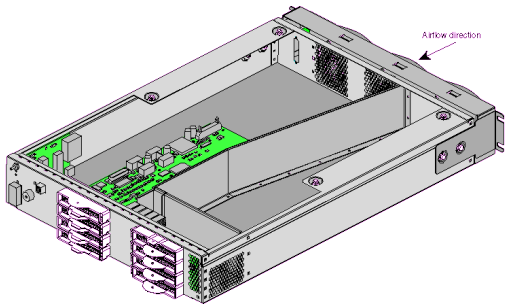
- 4 banks of up to 3 DDR-SDRAM dimms
- Dimms are 512 MB, 1GB or 2GB in size
  - PC2100 = 133MHz (DDR226) Altix BW = 8.5 GB/s - 7.5 ns
  - PC2700 = 166MHz (DDR333) Altix BW = 10.2 GB/s - 6.0 ns
  - PC3200 = 200MHz (DDR400) Altix BW = 12.8 GB/s - 5.0 ns

IRU Blockdiagramm



Router

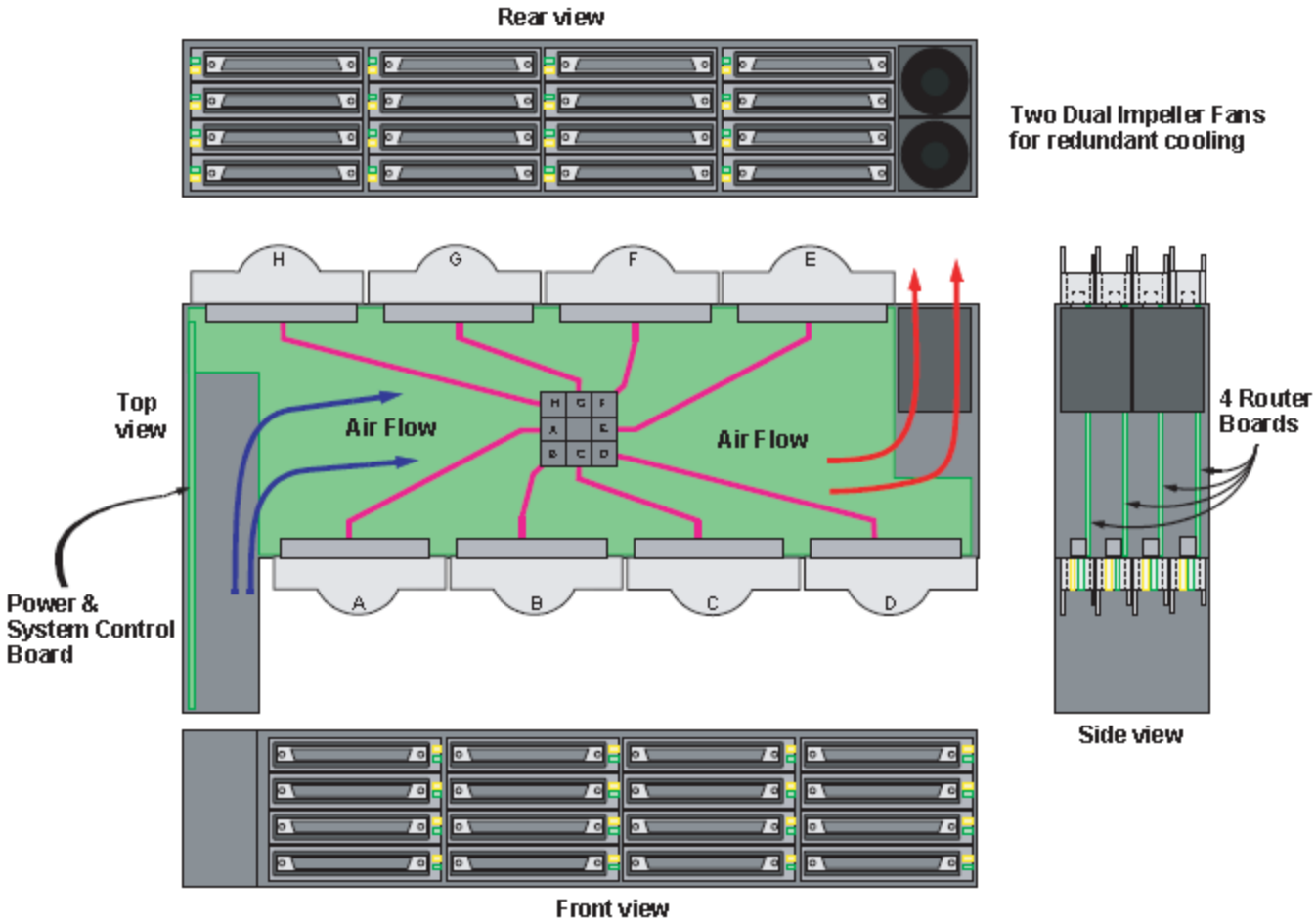




Numaflex-4 Router:

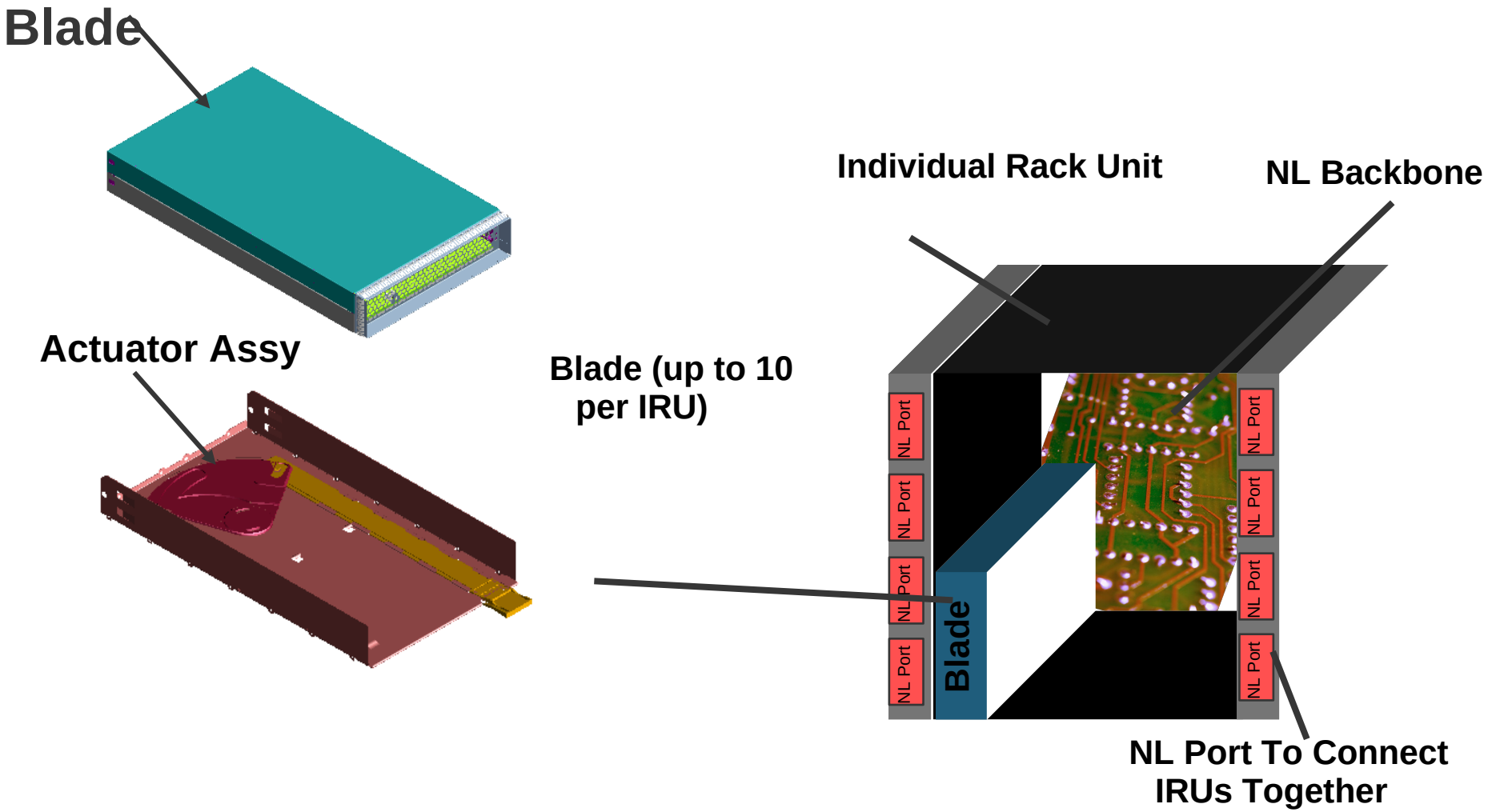
- **Microarchitecture elements of Cray T3E**
  - **Enhanced hardware support synchronization primitives**
- **8 bidirectional ports**
- **3.2 GB/s per direction per port**
- **Low latency about 50 nsec per router**
- **Dual plane configuration:**
  - **2 x 6.4GB/sec total bandwidth between C-bricks**

Quad Dense Metarouter

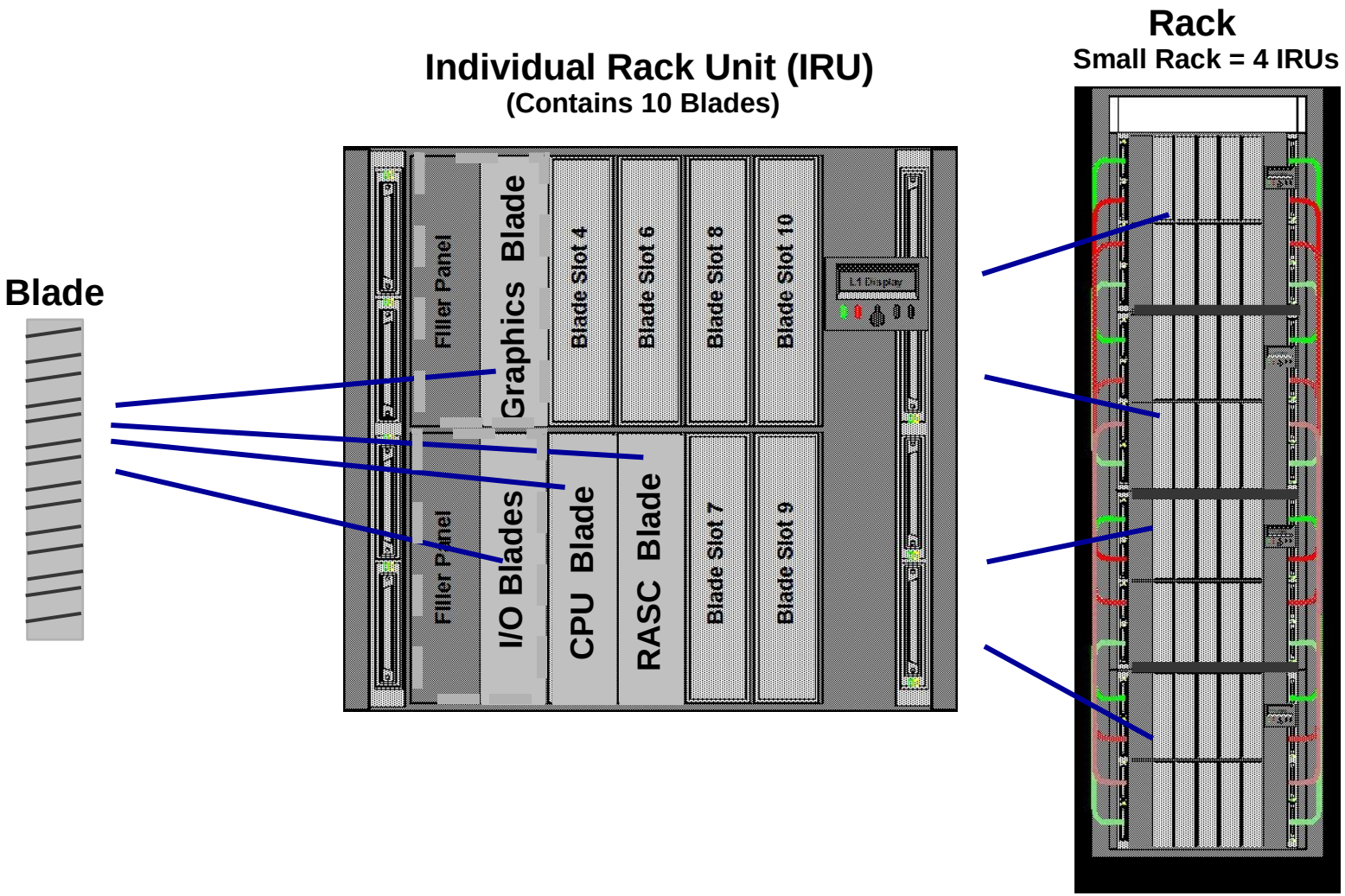


- Four 8-port routers in dense 2U package

SGI Altix 4700 – Blade Concept

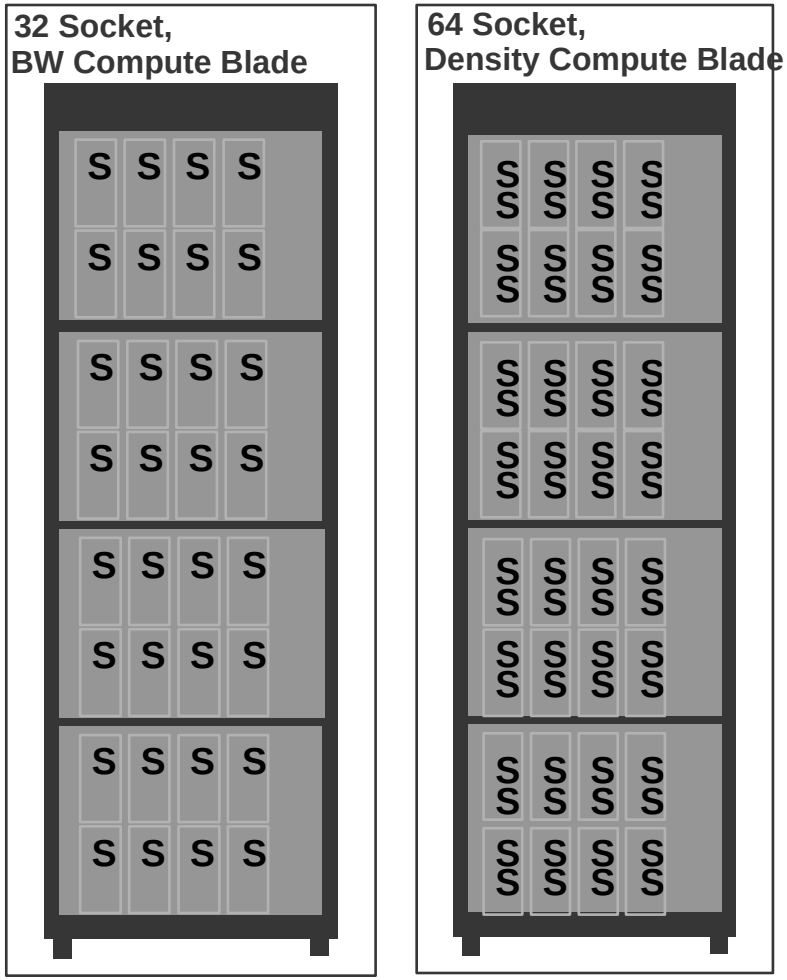


Standardized Blades, NUMAlink Backbone



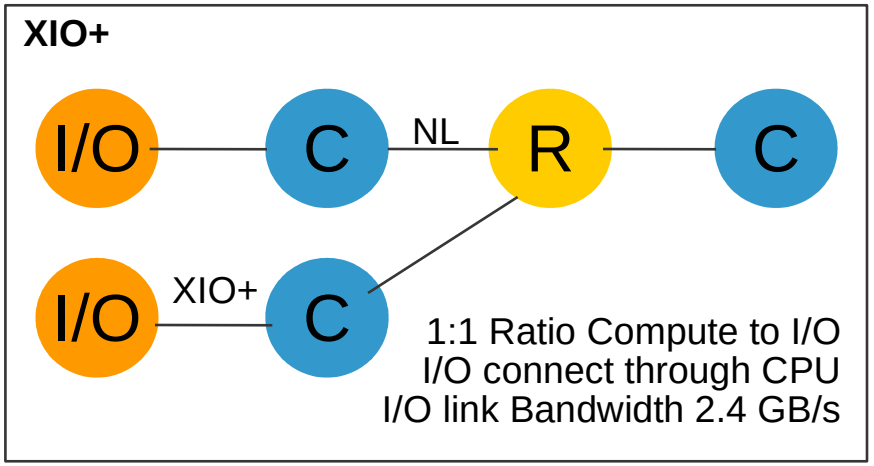
Leadership Performance Density & Versatility

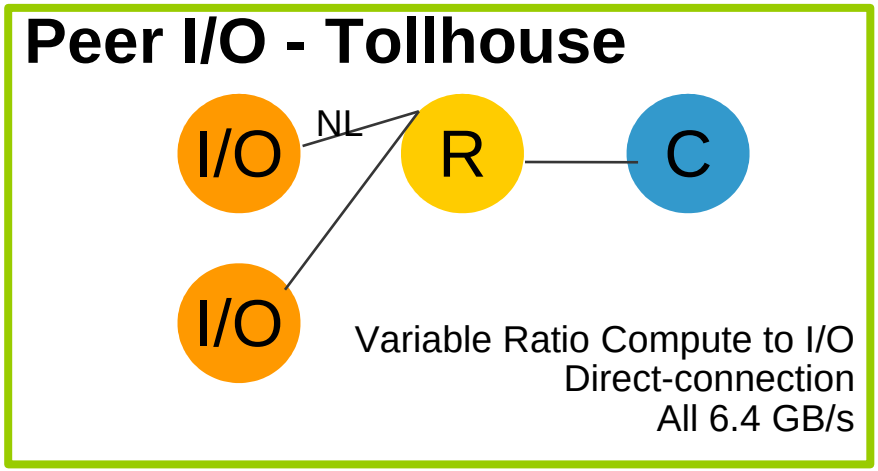
- **Configuration Flexibility: Design for Density or Performance**
  - **32-sockets per rack**
  - **64-sockets per rack**
  - **64-cores per rack**
  - **128-cores per rack**
  - **Best Memory BW, Performance (Bandwidth Compute Blade)**



Peer I/O: Enabling Increased I/O Flexibility & Performance

- Direct connection of I/O into NUMAlink memory fabric
- Increased I/O link bandwidth 2.4 → 6.4GB/sec
- Memory, Compute, I/O are universally accessible
- Total flexibility of compute to I/O ratio
- Allows I/O channel performance to scale concurrently with NUMAlink improvements





Excursion on PCI

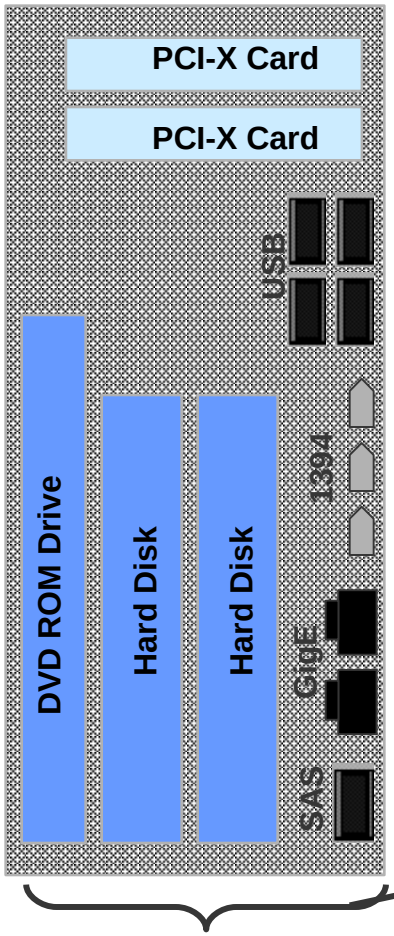
- Peripheral Component Interconnect
  - Invented by Intel
  - Started as 32-bit bus
  - Bus is buffered and works asynchronously
  - Supports Plug and Play configuration (PnP)
- PCIX, extension to width of 64 bits, up to 133 Mhz
- Some performance data

PCI		PCI-X		
33 MHz	66 MHz	66 MHz	100 MHz	133 MHz
132 MB/s	256 MB/s	512 MB/s	800 MB/s	1000 MB/s

- http://www.pcisig.com/specifications/
- http://arstechnica.com/articles/paedia/hardware/pcie.ars/1

SGI Altix 4700 I/O Blades (PCI-X Based)

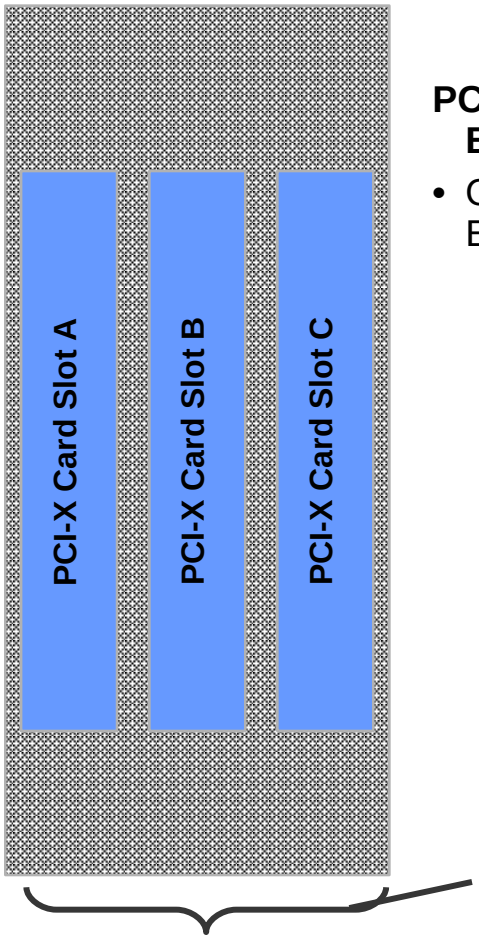
Front View



Base I/O Blade:

- Minimum of 1 Blade Required for Every SSI, Partition
  - Supports 2 SAS Drives
  - Low Profile PCI-X Slots
  - SAS, GigE, 1394, USB Capable

Front View

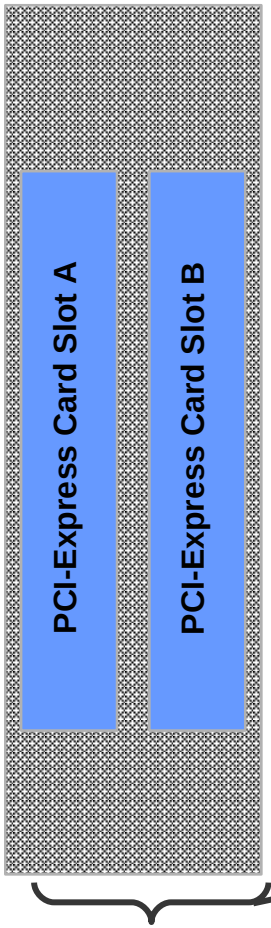


PCI-X Expansion Blade:

- Optional PCI-X Expansion
  - 3 Full PCI-X Slots, Hot Plug Capable
  - Slot A: 133MHz Bus
  - Slots B, C: 133MHz Each, 100MHz if Both Populated

SGI Altix 4700 Graphics or I/O Expansion (PCI-Express Based)

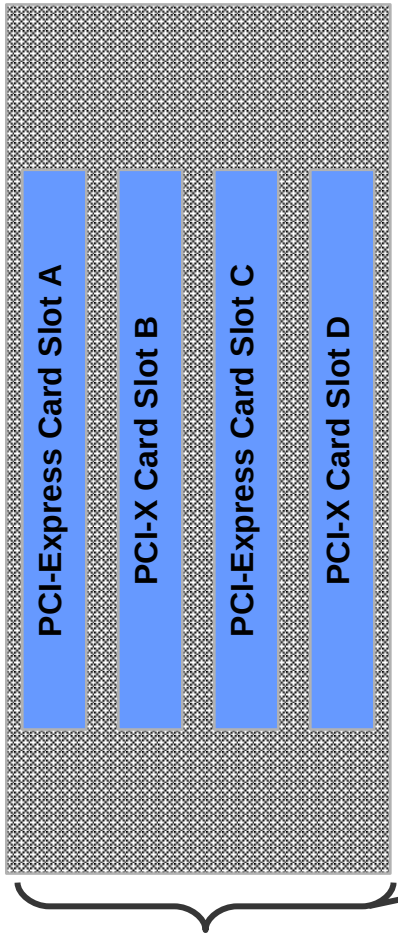
Front View



PCI-Express I/O & Graphics Expansion Blade:

- Optional PCI-Express Expansion for Graphics, I/O
  - 2 Full PCI-Express Slots
  - 1 PCI-Express Slot Per Channel with 16X PCI-Express Connector
  - Supports up to 90W per card for 2 Graphics Pipes, 150W per card for 1

Front View

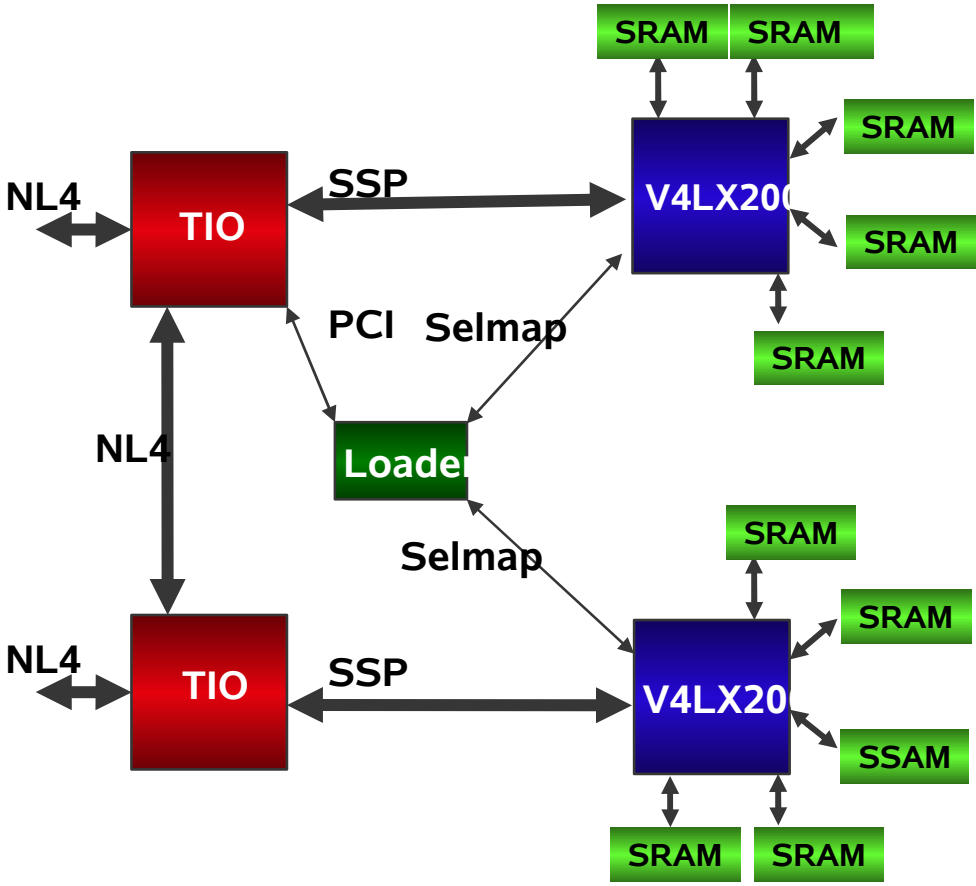


PCI-X + PCI-Express I/O & Graphics Expansion Blade:

- Optional PCI-X/PCI-Express Expansion for Graphics, I/O
  - 2 Full PCI-X, 2 Full PCI-Express Slots
  - Slots A, C: 16X PCI-Express
  - Slots B, D: 133MHz Bus Each
  - Supports Max of 150W Graphics Pipes (With B, D Unpopulated)

Next Generation Reconfigurable Compute Technology

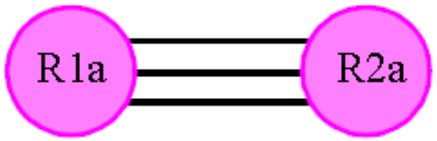
	SGI® RASC™ RC100 Blade
FPGA	Xilinx Virtex-4 LX200
No. of FPGAs	Two per blade
Host System	SGI® Altix® 4000 SGI® Altix® 3700 Bx2 or 350 * Silicon Graphics Prism™*+
Memory	80MB QDR SRAM **OR** 20GB DDR2 SDRAM
I/O	Dual NUMAlink™ 4 ports
Max Config	Up to 8 RC100 blades per system More available with custom config
Dimensions	Blade Form Factor ▪10-U Altix® 4000 IRU ▪Up to 8 RC100 blades per IRU Rack-Mountable Form Factor ▪2 blade slot chassis ▪3U (5.25" H x 19"W x 26"D)
O/S	Linux® OS (on host server)

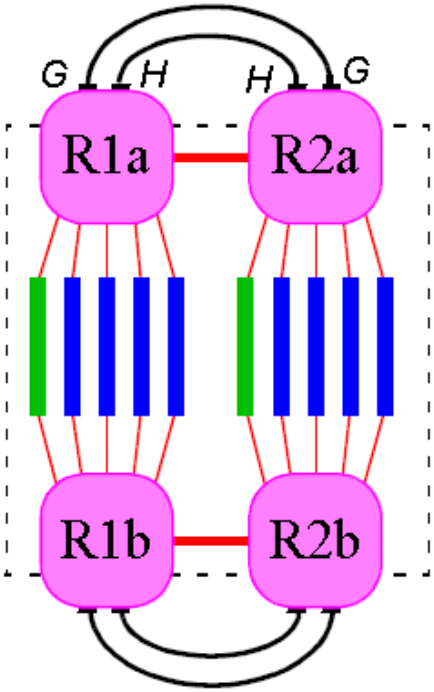


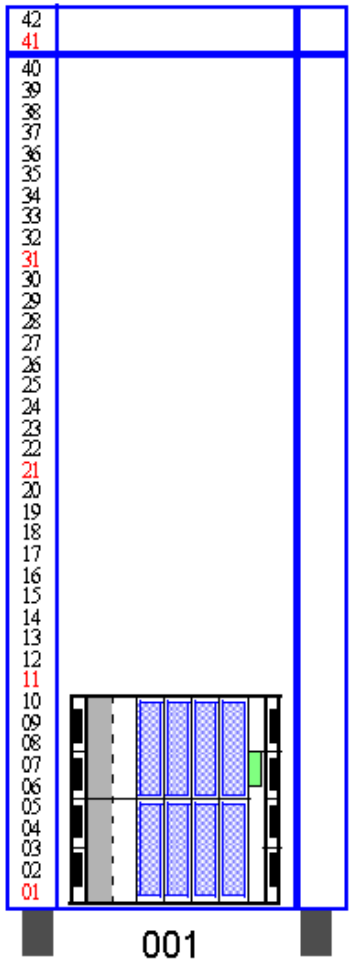
* with available 2 blade slot upgrade chassis
+ rack mounted version only

SGI Altix Configurations

Basic System – Single IRU





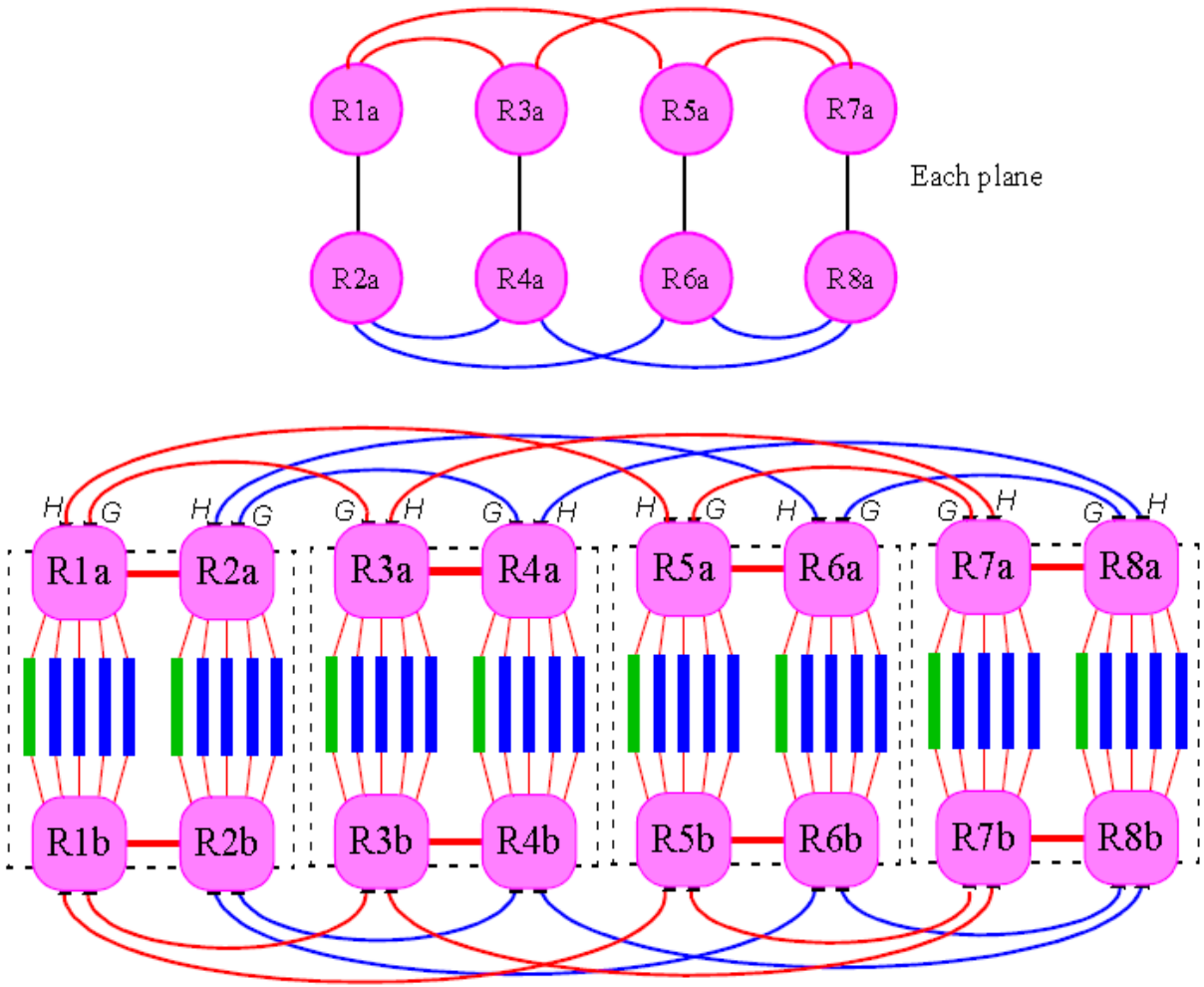


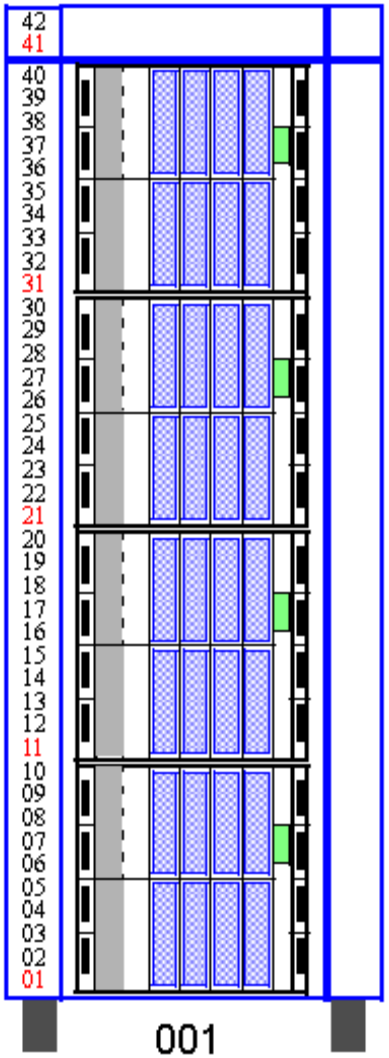
79.5 in. H × 25.8 in. W × 45

6 x 6.4 = 38.4 GB/s = 3.84 GB/s/blade

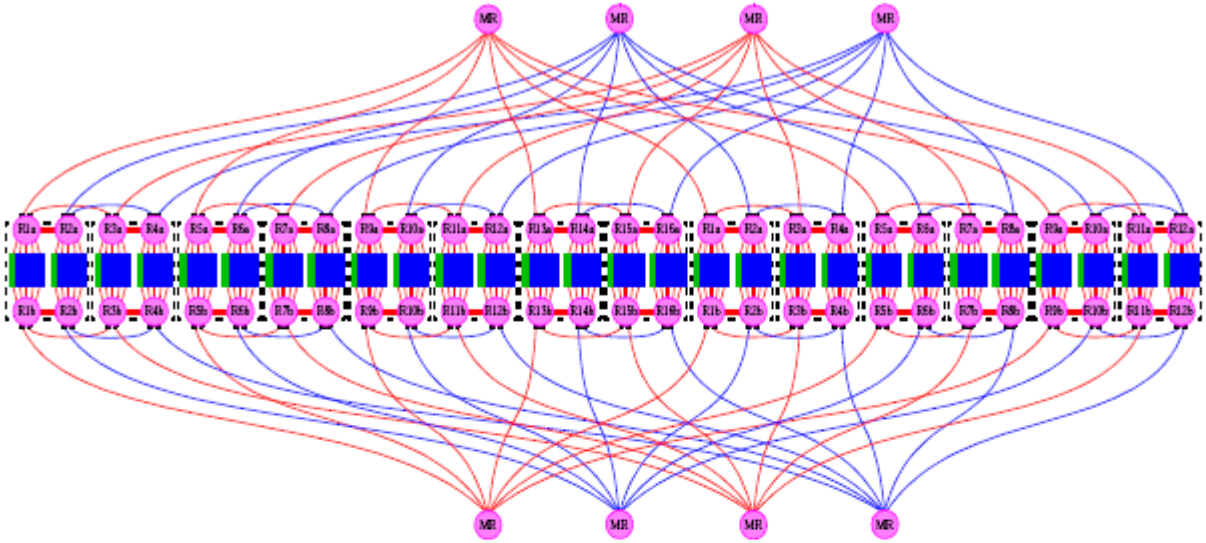
Single Rack

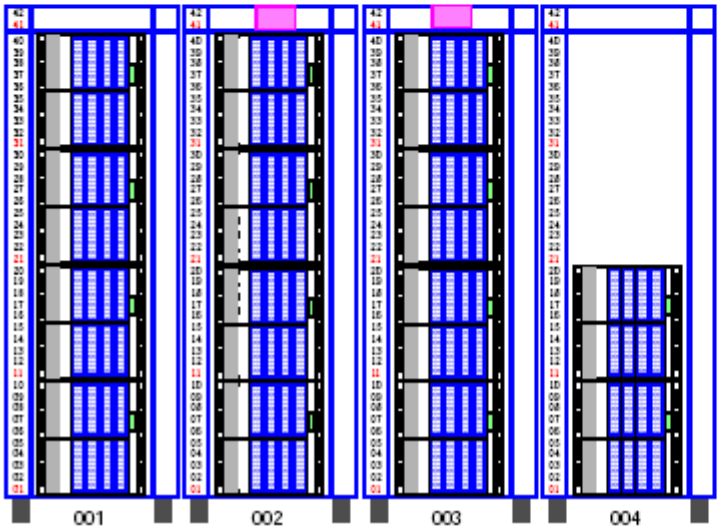
Hypercube topology within rack





128 Compute Blades

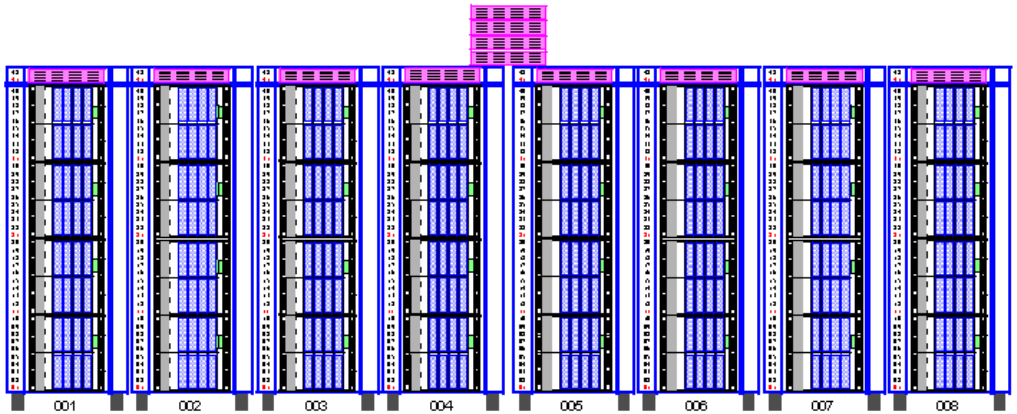




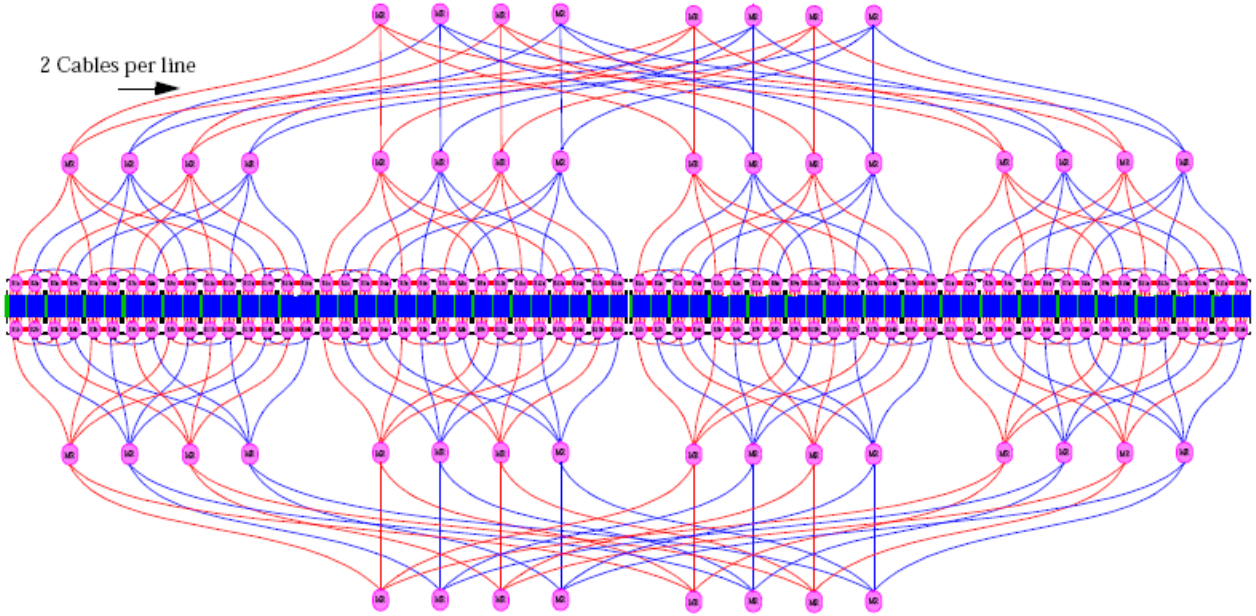
28 x 6.4 = 179.2 GB/s = 1.28 GB/s/blade

256 Processor blade system.

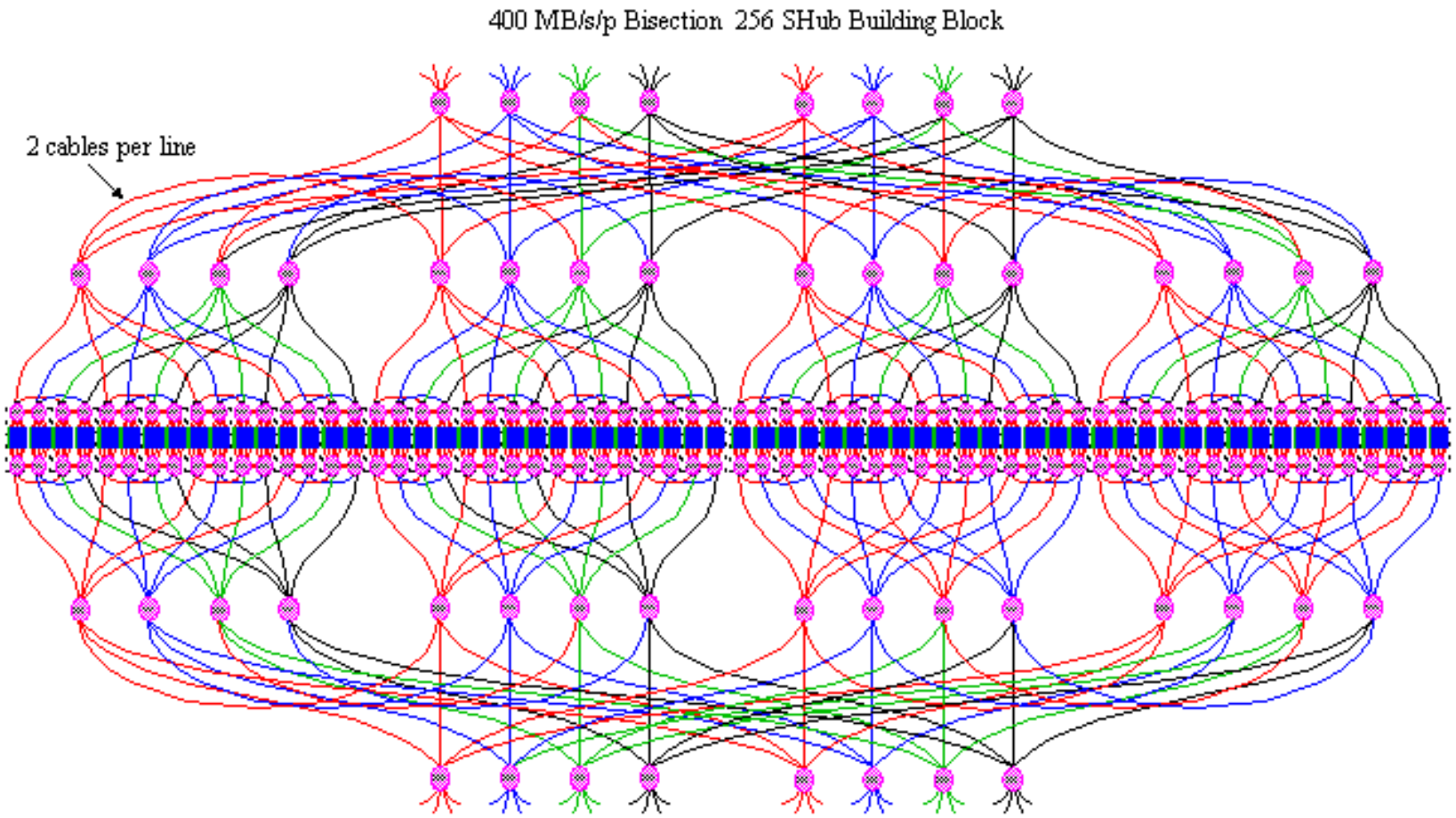
Fat-Tree Topology for multiple racks



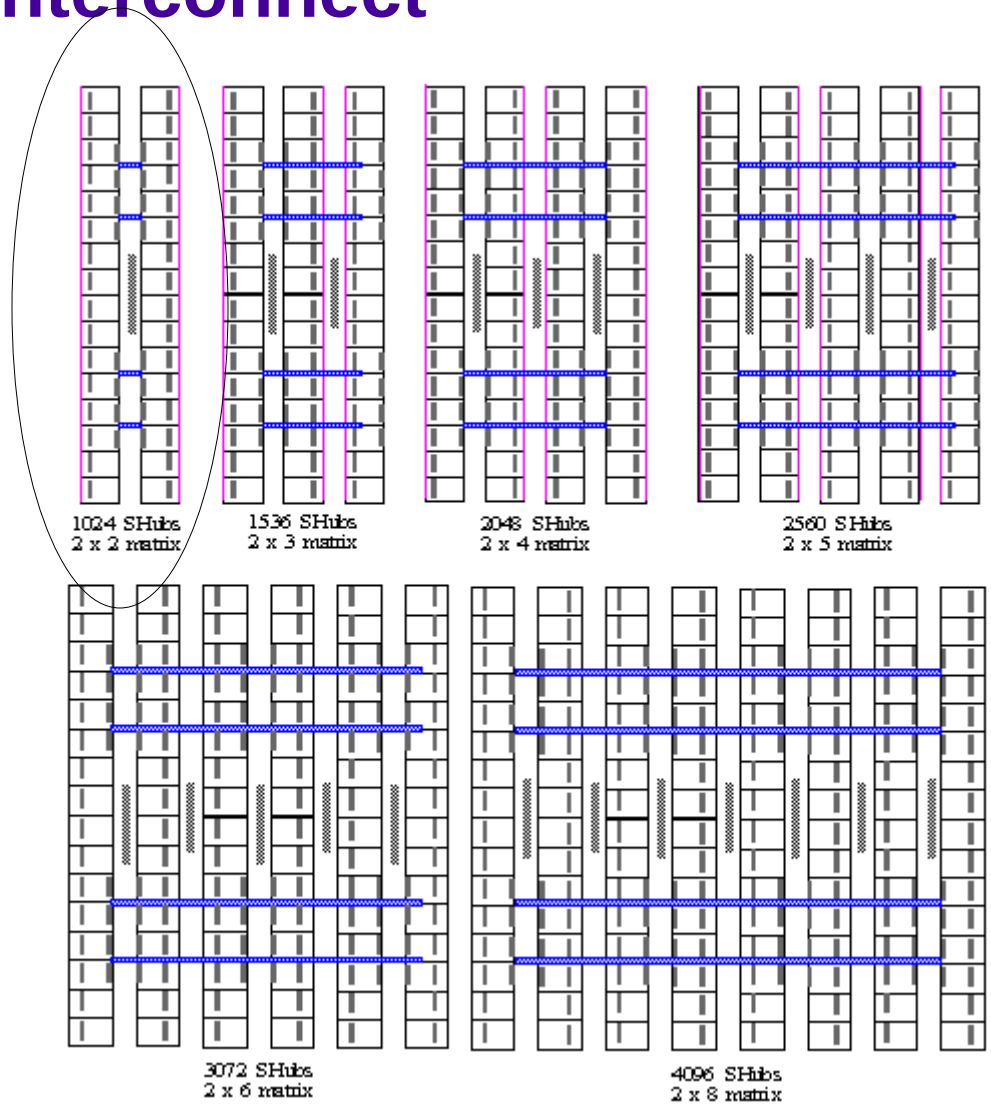
64 x 6.4 = 409.6 GB/s = 1.28 GB/s/blade



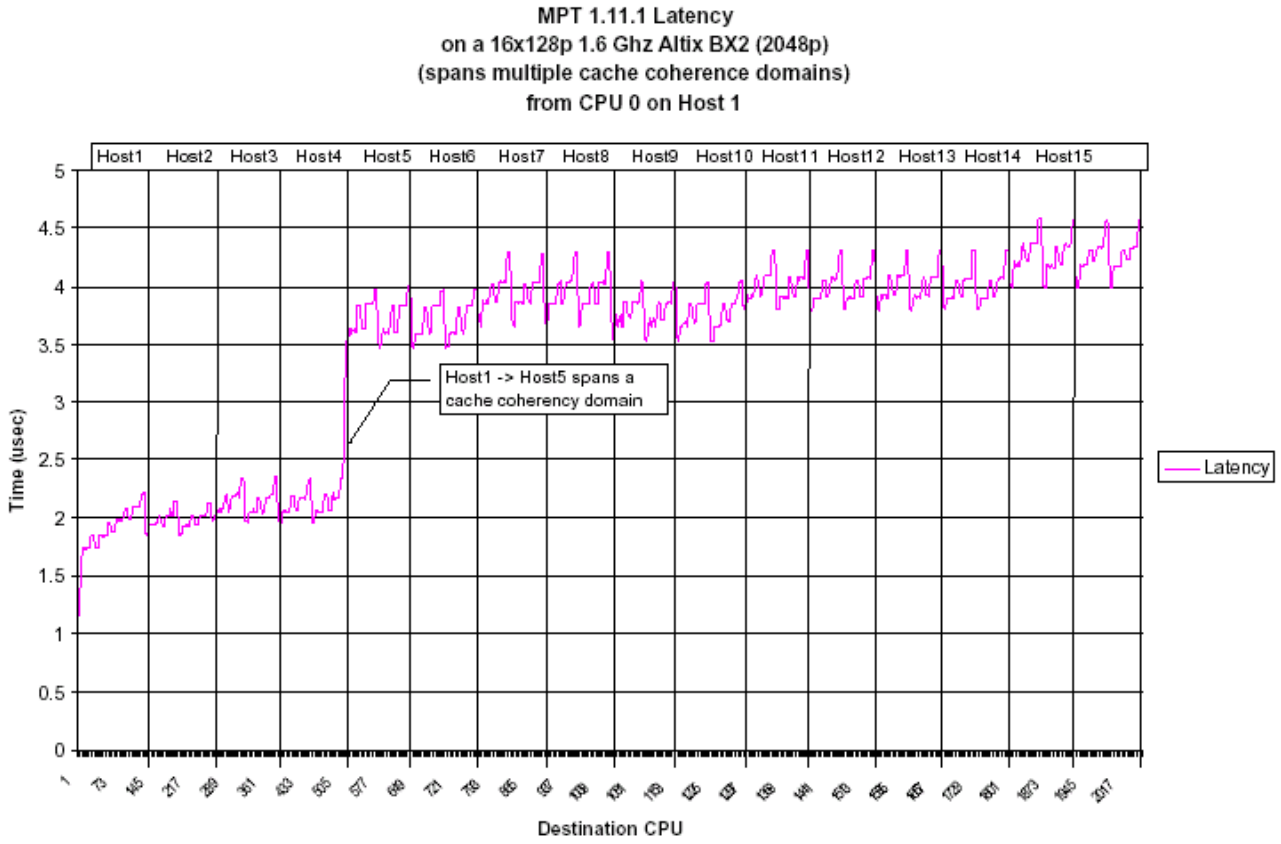
Building Block Beyond 256 Blades



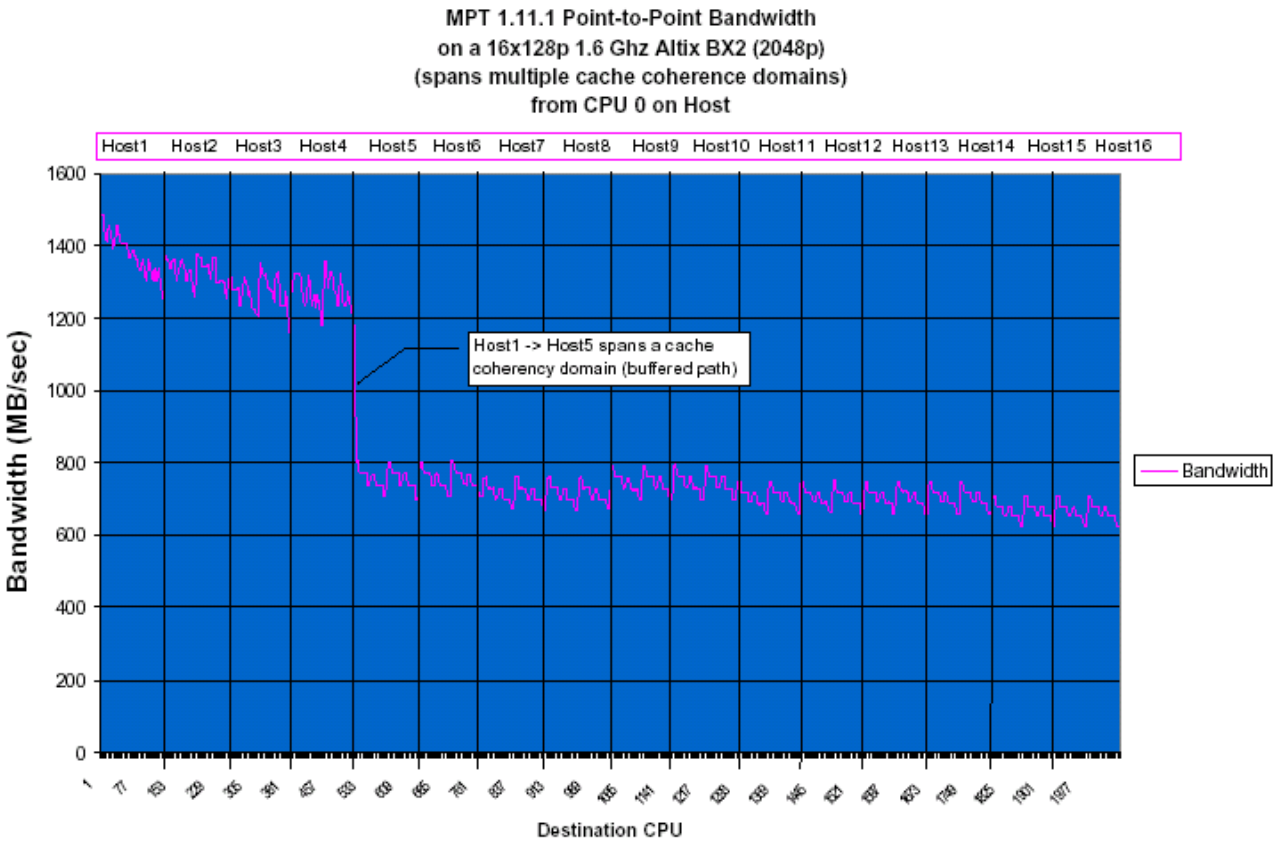
2d Matrix Interconnect



MPI Latencies



MPI Bandwidth




intel inside
ITANIUM 2

Intel® Itanium® 2 - Why it is important?

High Bandwidth

Many functional units

Large onchip caches

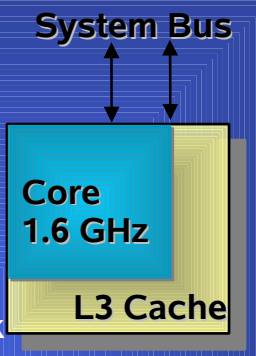
Large physical address space

System Bus
128 bits wide
200 MHz/400 MT/sec
6.4GB/sec

Width
2 bundles per clock
6 integer units
2 loads and 2 stores per clock
11 issue ports
4 FPMultiply Adds per Clock

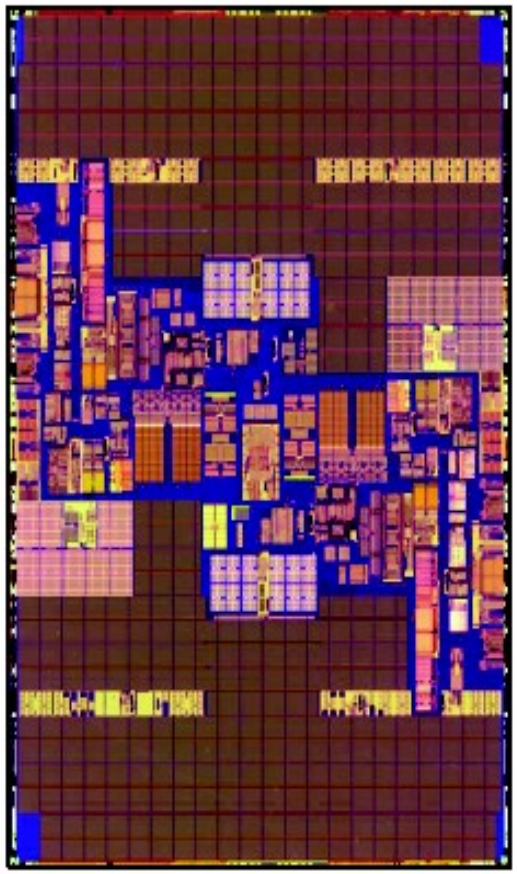
Caches
L1: 2X16KB—1 clock latency
L2: 256K—5 clock latency
L3: 3-9MB—12 clk
32GB/sec bandwidth

Addressing
50-bit physical addressing
64-bit virtual addressing
Maximum page size of 4GB



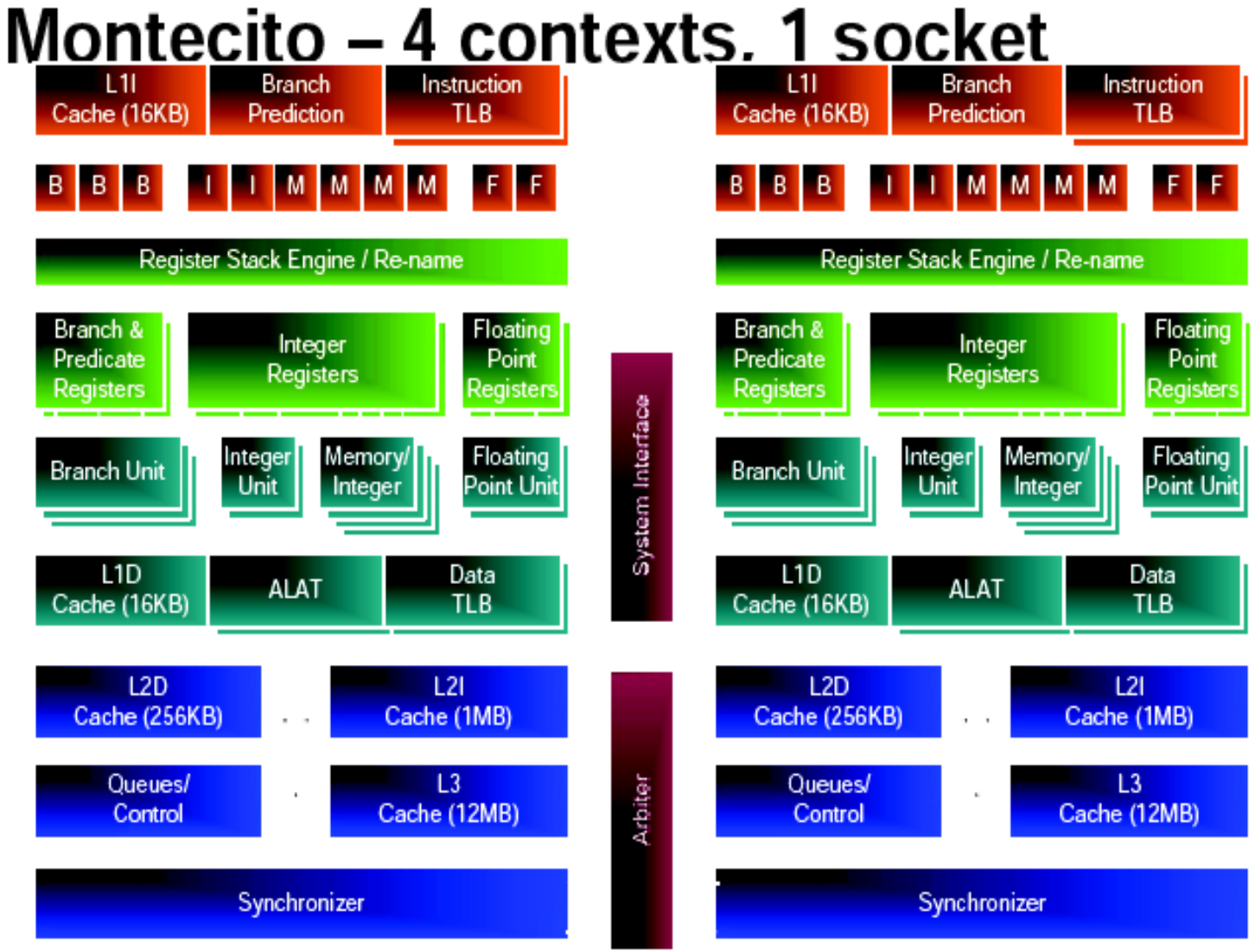
Montecito, Intel P9000

Itanium Dual Core: Montecito

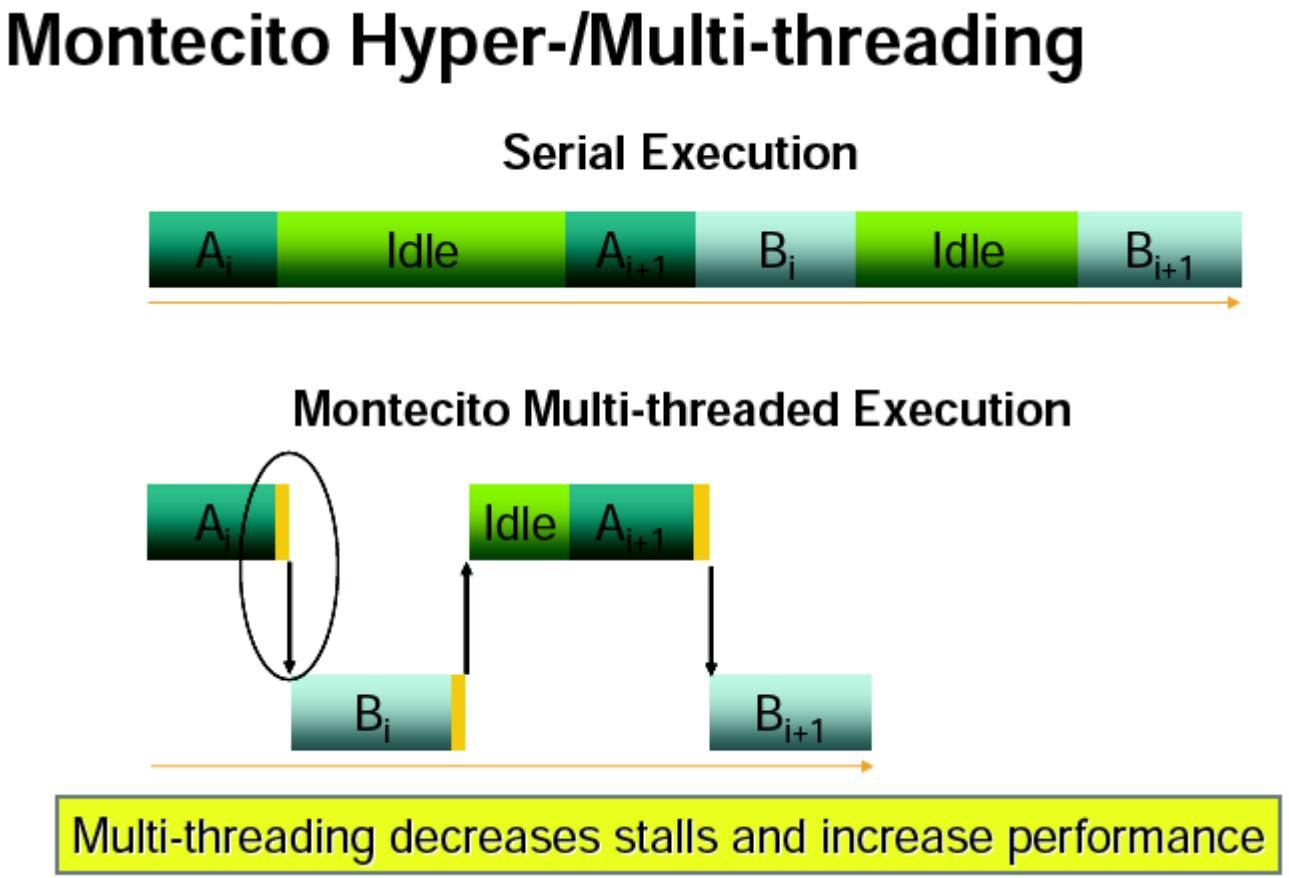


Montecito Feature Summary	
Simultaneous Threads	4
Process technology	90nm
L1 Cache	2 x (D16K + I16K)
L2 Data	2 x 256K
L2 Instruction Cache	2 x 1 MB
L3 Cache (Unified)	2 x 12 MB
Transistors	1,720,000,000
Availability Target	2006

Montecito, Intel P9000



Montecito, Intel P9000



Montecito, Intel P9000

Montecito Core Extensions

Slightly extended Itanium® 2 processor core:

- Larger atomic ops: 16-byte ld16/st16/cmp8xchg16
  - Support non-blocking synchronization in database apps
  - Improves performance scalability of database applications on large SMP
- Instruction(s) to support virtualization
- Cash flush extensions (fc and fc.i)
- hint@pause for thread switching
  - Is a NOP on older architecture, will not fault
- Additional integer shifter and popcount
  - Allows scheduling two variable shifts per cycle
  - Enhanced processing performance in cryptographic codes
- Faster chk.a/chk.s resteer
- Support in compiler 9.0 via
  - intrinsics (not documented yet)
  - Introduction of new machine model (KNOBs file for Montecito)

Explicitly Parallel Instruction Computing (EPIC)

- EPIC
  - New instruction set (with IA-32™compatibility)
  - 3 predicated instructions into 1 bundle (128bit)
  - 2 bundles per cycle
  - 128 general (integer) registers; up to 96 rotating
  - 128 floating-point registers; up to 96 rotating
  - 64 1-bit predicate registers; up to 48 rotating
  - 8 branch registers
  - 128 application registers (e.g., loop or epilog counters for pipelining)
  - Performance Monitor Unit (PMU) (> 100 Performance Counters)
  - Advanced Load Address Table (ALAT)
  - 6 integer units
  - 2 loads and 2 stores per clock cycle, speculative loads
  - 11 issue ports
  - Special instructions (multimedia, popcnt)

IA-64™ Instruction Bundles

1 instruction coded on 41 bits
3 instructions grouped into 1 bundle (128 bits)

Bundle type is specified through 5-bit template :

```
{       .mfi                                    // template (mem-fp-int)
   (p16) ldfd f39=[r2],16                       // load fp, post-increment
   (p19) fnma.d.s0 f49=f42,f6,f45               // multiply Add
   (p16) adds r32=16,r33        };              // integer add immediate
{          .mib                                 // template (mem-fp-br)
   (p16) ldfd f42=[r33]                         // load fp, post-increment
   (p16) adds r40=8,r33
   br.ctop.dptk.few .BB13_mp_ortho2_ ;;      };// counted loop branch
```

IA-64™ Branch Optimization

Predication allows to remove (small) branches:

```
if ( i == j) {
    k += l;
   x = y + a * b;
} else {
    k = m - 3;
    y = * p_fp ++ ;
}
```

```
cmp.eq p1,p2=r32,r33 ;;
(p1) add r1 = r1, r3
(p1) fpma.d f31 = f3, f4, f2
(p2) sub r1 = 3, r4
(p2) ldfd f31=[r34], 8
```

cycle 0

cycle 1

cycle 1

cycle 1

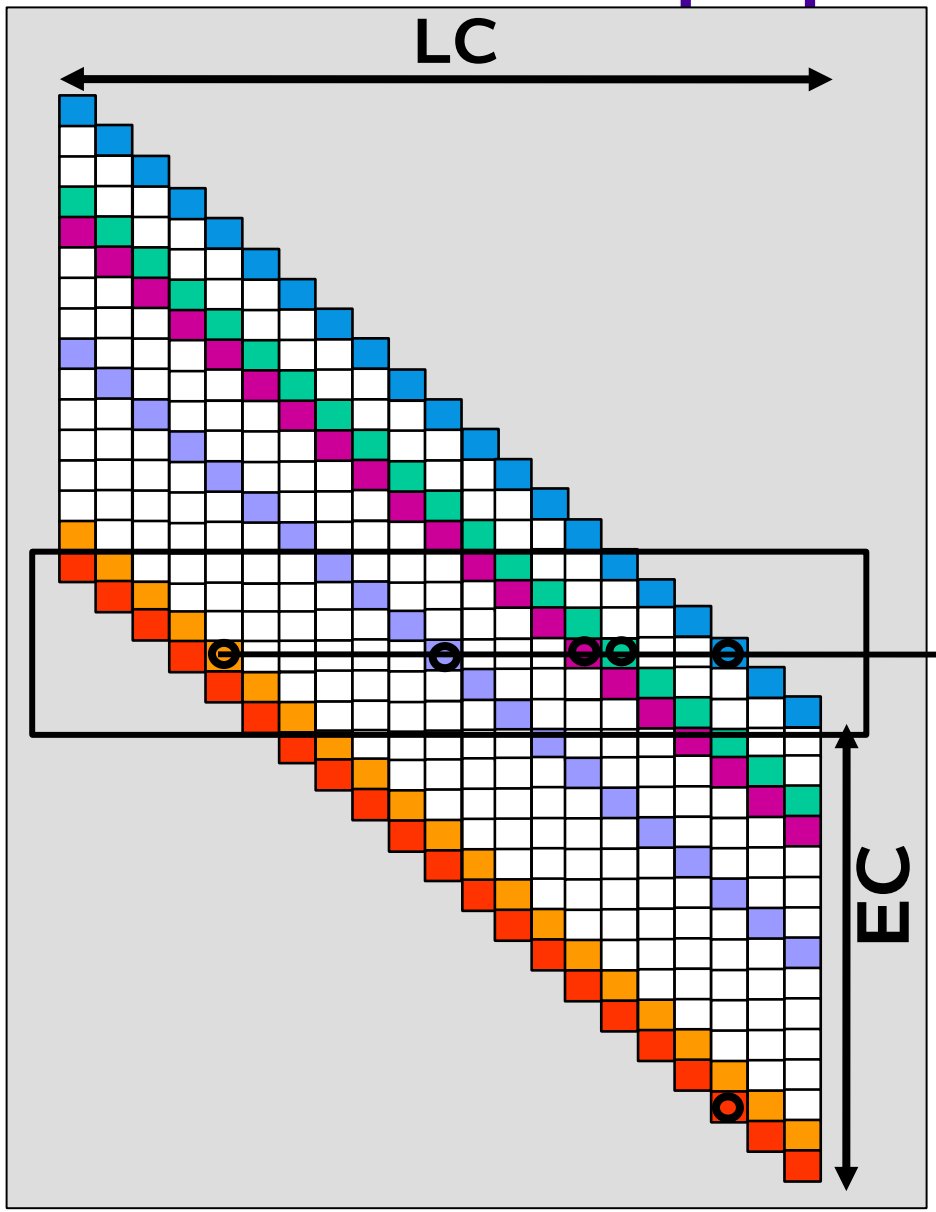
cycle 1

IA-64™ HW for Loop Optimization

Counted loops are optimized with HW support:

- **Loop counter**
- **Epilog counter**
- **Predication registers for each instruction**
- **Rotation of registers**

IA-64™ HW for Loop Optimization




```
{ .mfi
  (P16)
  (P24)
  (P19)      } ;

{ .mib
  (P30)
  (P20)
             br.ctop ;; } ;
```

Itanium™2 - Execution Units

- 6 ALU ALU0-5
- 2 Integer I0,I1
- 1 ISHIFT
- 4 Port Data Cache Unit (2ld[fp]+2st or 4ldf)
- 6 Multimedia PALU0-5
- 2 Parallel shift PSMU0,1
- 1 Parallel Multiply PMUL
- 1 POPCNT
- 2 FP multiply-add FMAC
- 2 FP other operations FMISC
- 3 Branch

Itanium™2 - Instructions Latency

– Integer Instructions	1 cycle
– Floating Point Instructions	4 cycles
– MultiMedia	2 cycles
– FP Multiply-Add/sub *fma/fnma/fms*	4 cycles
– FP Multiply or Add (*fma* x*y+0 or x*1+y)	4 cycles
– no FP Div, use approx[256] *frcpa*	4 cycles
– no FP RSQRT, use approx[256] *frsqra*	4 cycles
– no integer mult, use *setf/xma/getf*	6/4/5 cycles
– no integer Mod, Div use *setf/frcpa/…/getf*	6/4/5 cycles

Itanium™2 - FP Macros Latency

x/y, 1/sqrt(x), sqrt(x) do not translate into HW instructions.

Instead the compiler combines fma/frcpa/frsqra (Newton iterations).

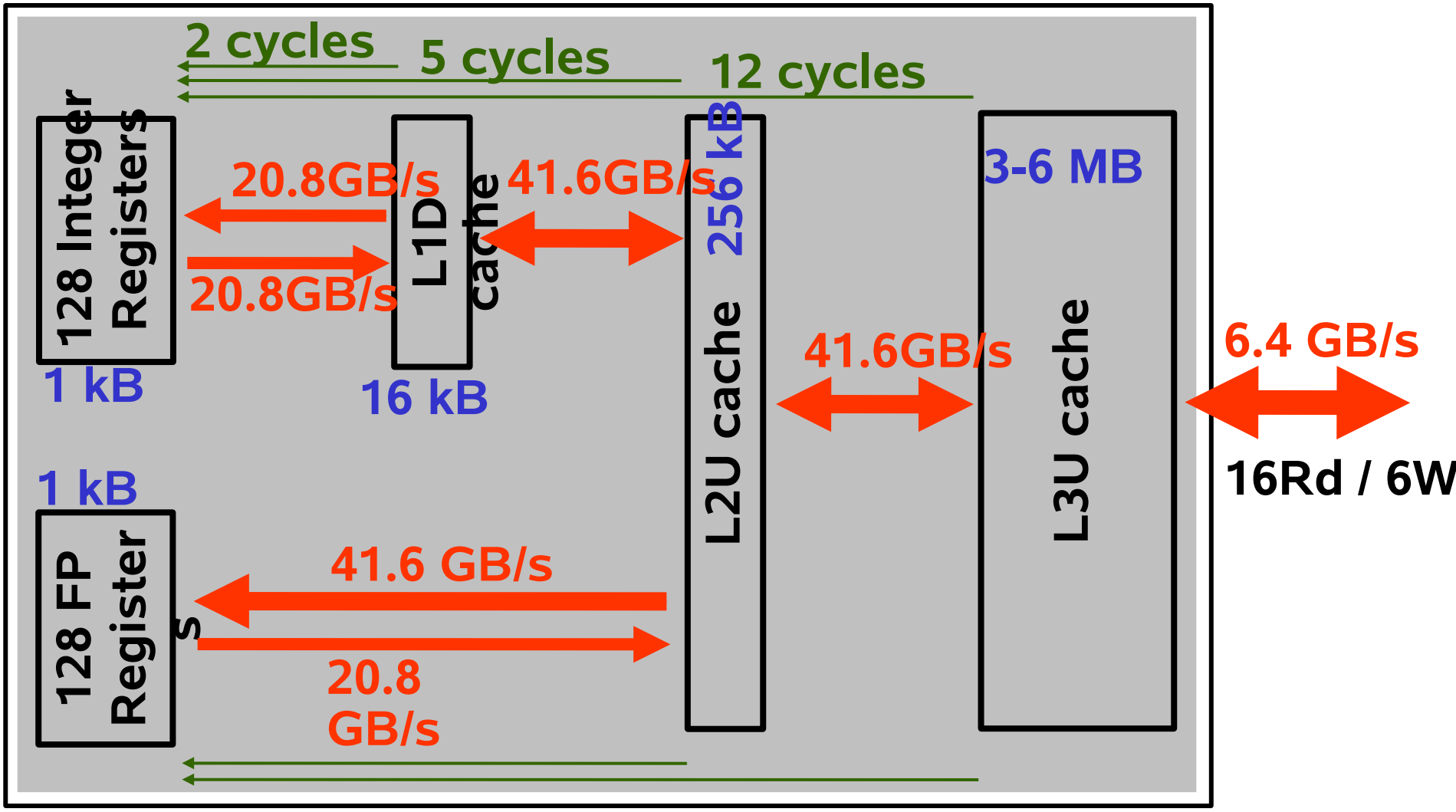
Similarly integer *, /, %(modulo) are expanded through macros.

Latency will vary depending with compiler efficiency :

FP cycles	y = a+y	y = a*y	y = a+b*y	y = b+a/y	y = a/sqrt(y	y = sqrt(y	y =y/sqrt(y)
Single	4	4	4	28	36	43	36
Double	4	4	4	32	37	55	37

Int cycles	i = i + c	i = a*i	i = a+b*i	i = b+ a / i	i = b + a % i
Single	1	15	16	37	42
Double	1	15	16	56	61

Itanium™2 Data Flow



Itanium™2 L1/L2 Data Cache

L1D is 16kByte, 64Byte/line, 4way, WriteThrough, GRegisters only:

- 1 cycle latency (2 for load, pointer chasing), no FP cached in L1D
- Store uses 8x8 bytes array. Updates L1D only if hit.
- 8 (unique) outstanding misses

L2U is 256kByte, 128Byte/line, 8way, WriteBack, NotRecentlyUsed

- 5,7,9.../6,8,10… latency for Int/FP
- 16 banks - 16bytes/bank (??? 256Byte stride/alignment ???)
- 16 (unique) outstanding misses
- L2 is not inclusive of L1D and L1I

Itanium™2 L3U Cache/Memory

L3U: 1.5/3MByte, 128Byte/line, 6/12way, WriteBack, LeastRecentlyUsed

- 12,16.../13,17… latency for Int/FP
- 16 (unique) read misses
- 6 write

Local/remote memory is accessed through SHub/NUMAflex:

Local latency	132 ns
Same brick / other node	180 ns
NL4 router	~50 ns
1 Meter cable	~10 ns


sgi®

SGI Scalable ccNUMA Architecture