

# BG/Q Architecture

Carlo Cavazzoni, HPC department, CINECA





# FERMI @ CINECA

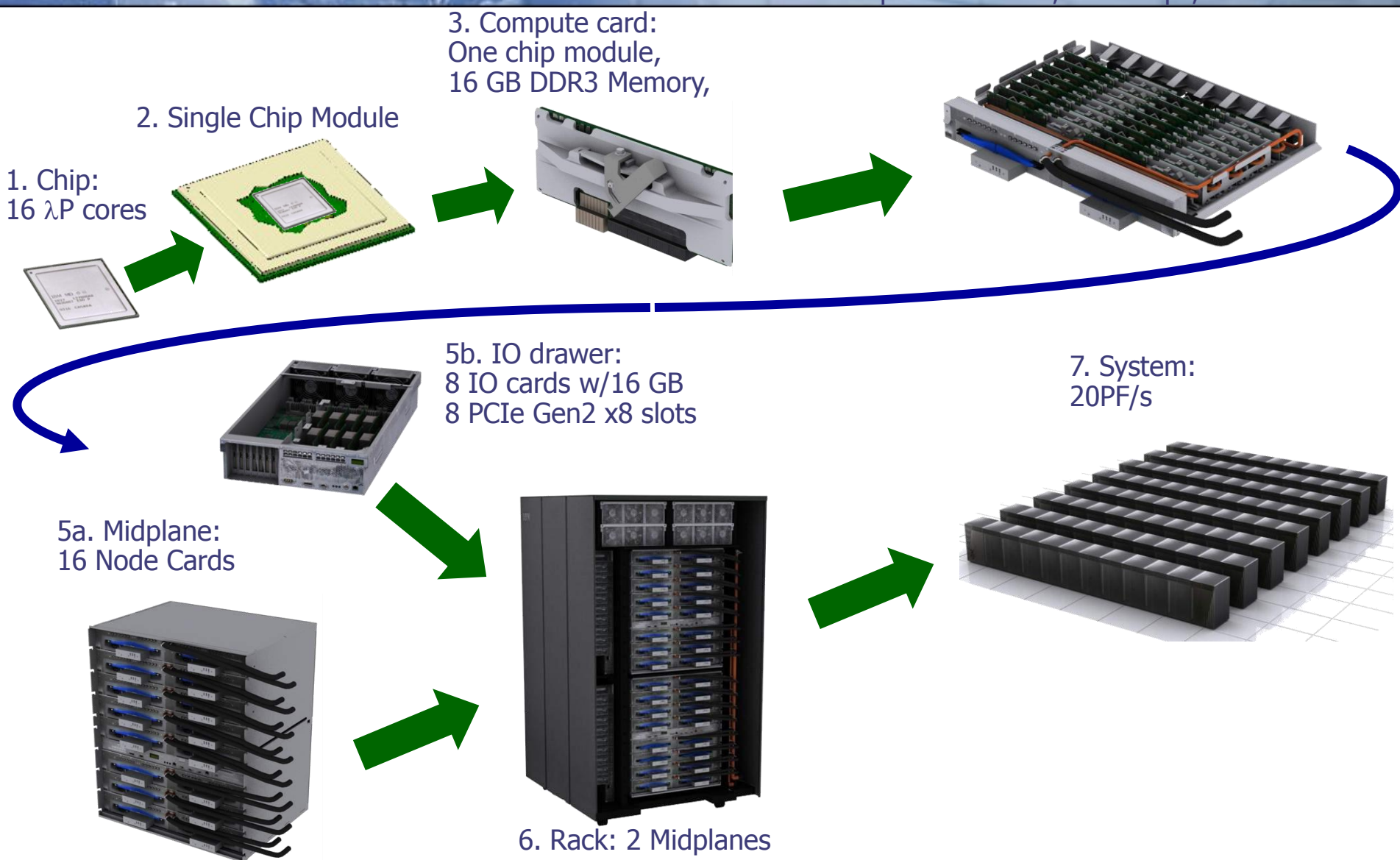
## PRACE Tier-0 System

Architecture: 10 BGQ Frame  
Model: IBM-BG/Q  
Processor Type: IBM PowerA2, 1.6 GHz  
Computing Cores: 163840  
Computing Nodes: 10240  
RAM: 1GByte / core  
Internal Network: 5D Torus  
Disk Space: 2PByte of scratch space  
Peak Performance: 2PFlop/s



# ISCRA & PRACE call for projects now open!

4. Node Card:  
32 Compute Cards,  
Optical Modules, Link Chips, Torus



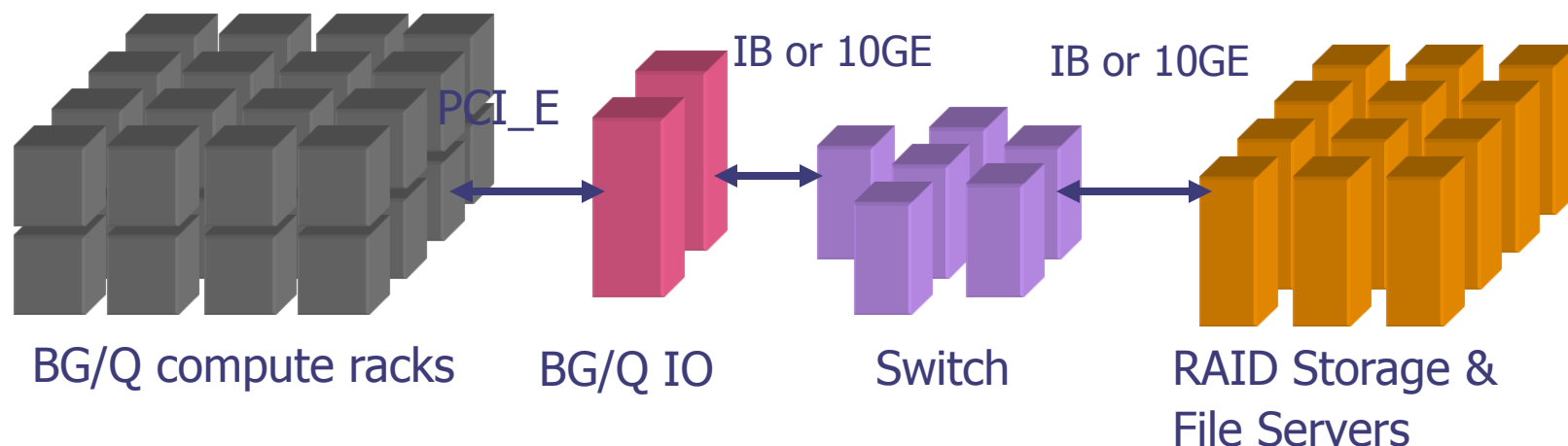
Point-to-point fiber cables,  
attaching the 8 I/O nodes  
(on top of rack)  
to compute nodes  
(on 8 node cards)



4D torus fiber cables,  
connecting the  
midplane to  
other midplanes  
(in same and other racks)



# BG/Q I/O architecture



## External, independent and dynamic I/O system

- I/O nodes in separate drawers/rack with private interconnections and full Linux support
- PCI-Express Gen 2 on every node with full sized PCI slot
- Two I/O configurations (one traditional, one conceptual)

## BlueGene Classic I/O with GPFS clients on the logical I/O nodes

Similar to BG/L and BG/P

Uses InfiniBand switch

Uses DDN RAID controllers and File Servers

BG/Q I/O Nodes are not shared between compute partitions

- **IO Nodes are bridge data from function-shipped I/O calls to parallel file system client**

Components balanced to allow a specified minimum compute partition size to saturate entire storage array I/O bandwidth

### **I/O Network to/from Compute rack**

- 2 links (4GB/s in 4GB/s out) feed an I/O PCI-e port
- Every node card has up to 4 ports (8 links)
- Typical configurations
  - ✓ 8 ports (32GB/s/rack)
  - ✓ 16 ports (64 GB/s/rack)
  - ✓ 32 ports (128 GB/s/rack)
- Extreme configuration 128 ports (512 GB/s/rack)

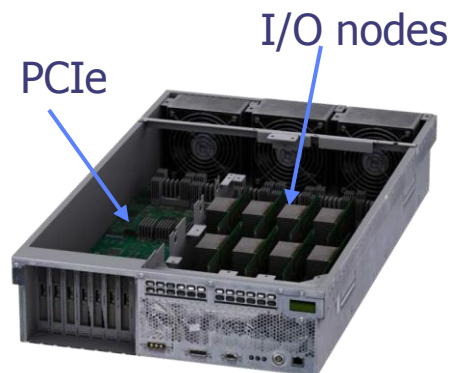
### **I/O Drawers**

- 8 I/O nodes/drawer with 8 ports (16 links) to compute rack
- 8 PCI-e gen2 x8 slots (32 GB/s aggregate)
- 4 I/O drawers per compute rack
- Optional installation of I/O drawers in external racks for extreme bandwidth configurations

Locations of IO enclosures can be:

- Qxx-Iy (in an IO rack, y is 0 - B)
- Rxx-Iy (in a compute rack, y is C - F)

I/O drawers

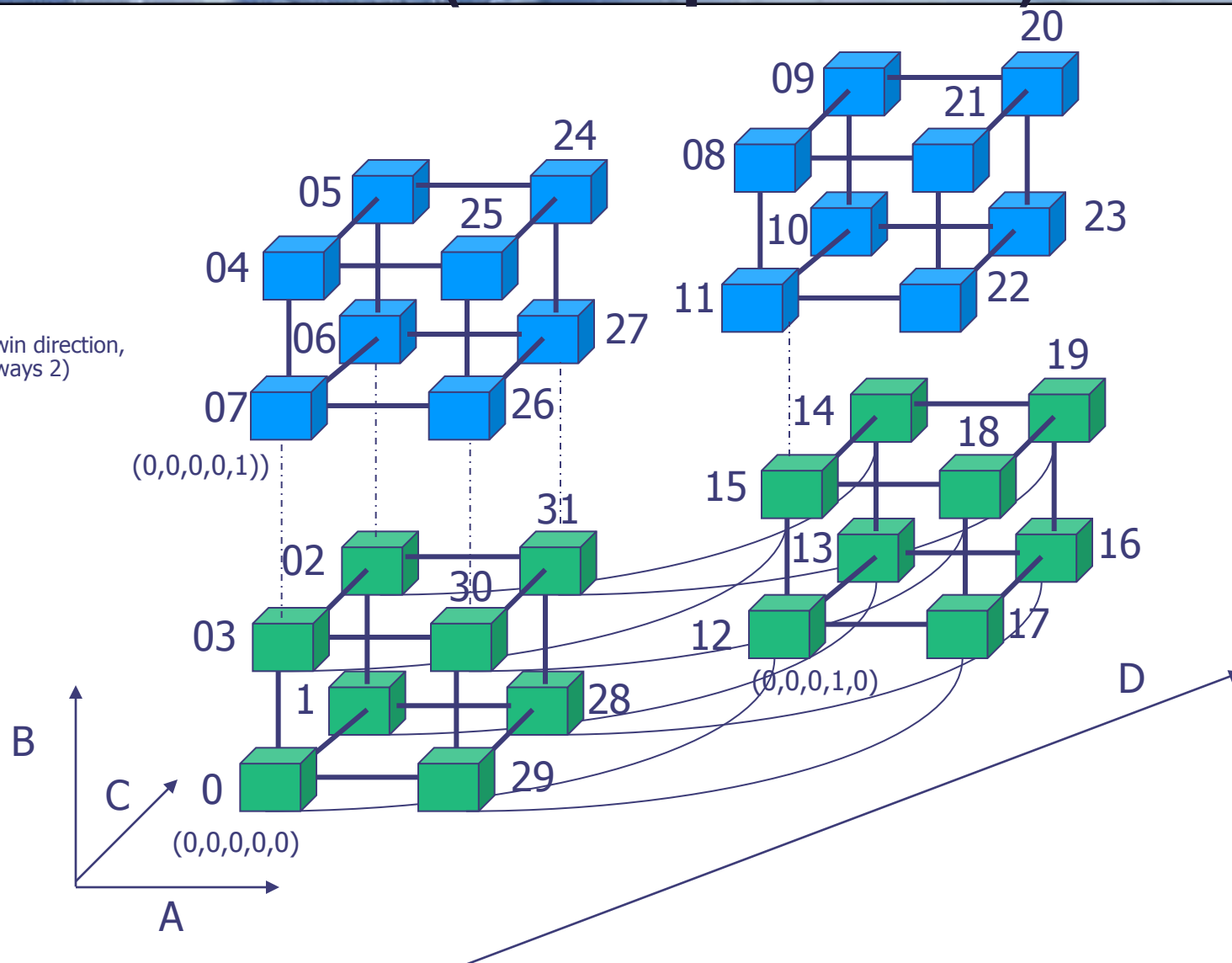


# New Network architecture:

- 5 D torus architecture sharing several embedded Virtual Network/topologies
  - ✓ 5D topology for point-to-point communication
    - ❖ 2 GB/s bidirectional bandwidth on all (10+1) links
    - ❖ Bisection bandwidth of 65TB/s (26PF/s) / 49 TB/s (20 PF/s) BGL at LLNL is 0.7 TB/s
  - ✓ Collective and barrier networks embedded in 5-D torus network.
- Floating point addition support in collective network
- 11<sup>th</sup> port for auto-routing to IO fabric

# Node Board (32 Compute Nodes): 2x2x2x2x2

E  
(twin direction,  
always 2)



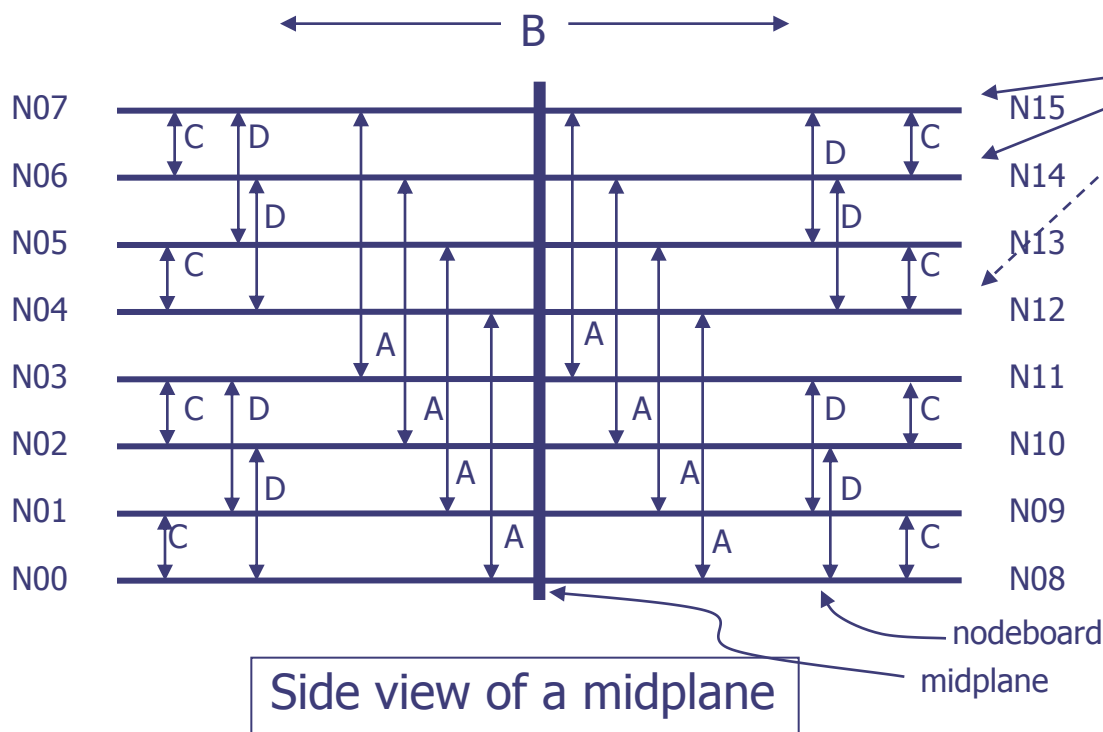


# Network topology | Mesh versus torus

# Node Boards	# Nodes	Dimensions	Torus (ABCDE)
1	32	2x2x2x2x2	00001
2 (adjacent pairs)	64	2x2x4x2x2	00101
4 (quadrants)	128	2x2x4x4x2	00111
8 (halves)	256	4x2x4x4x2	10111

# 5-D torus wiring in a Midplane

The 5 dimensions are denoted by the letters A, B, C, D, and E. The latest dimension E is always 2, and is contained entirely within a midplane.



Each nodeboard is 2x2x2x2x2  
Arrows show how dimensions A,B,C,D span across nodeboards  
Dimension E does not extend across nodeboards

The nodeboards combine to form a 4x4x4x4x2 torus  
Note that nodeboards are paired in dimensions A,B,C and D as indicated by the arrows

# BGQ PowerA2 processor

Carlo Cavazzoni, HPC department, CINECA



# Power A2

64bit

Power instruction set (Power1...Power7, PowerPC)

RISC processors

Superscalar

Multiple Floating Point units

SMT

Multicore

# PowerA2 chip, basic info

16 cores + 1 + 1 (17th Processor core for system functions)

1.6GHz

32MByte cache

system-on-a-chip design

16GByte of RAM at 1.33GHz

Peak Perf 204.8 gigaflops

power draw of 55 watts

45 nanometer copper/SOI process (same as Power7)

Water Cooled



4-way SMT

SIMD floating point unit (8 flop/clock) with alignment support: QPX

Speculative multithreading and transactional memory support with  
32 MB of speculative state

Hardware mechanisms to help with multithreading

Dual SDRAM-DDR3 memory controllers with up to 16 GB/node

# PowerA2 chip, more info

Contains a 800MHz crossbar switch

links the cores and L2 cache memory together

peak bisection bandwidth of 563GB/sec

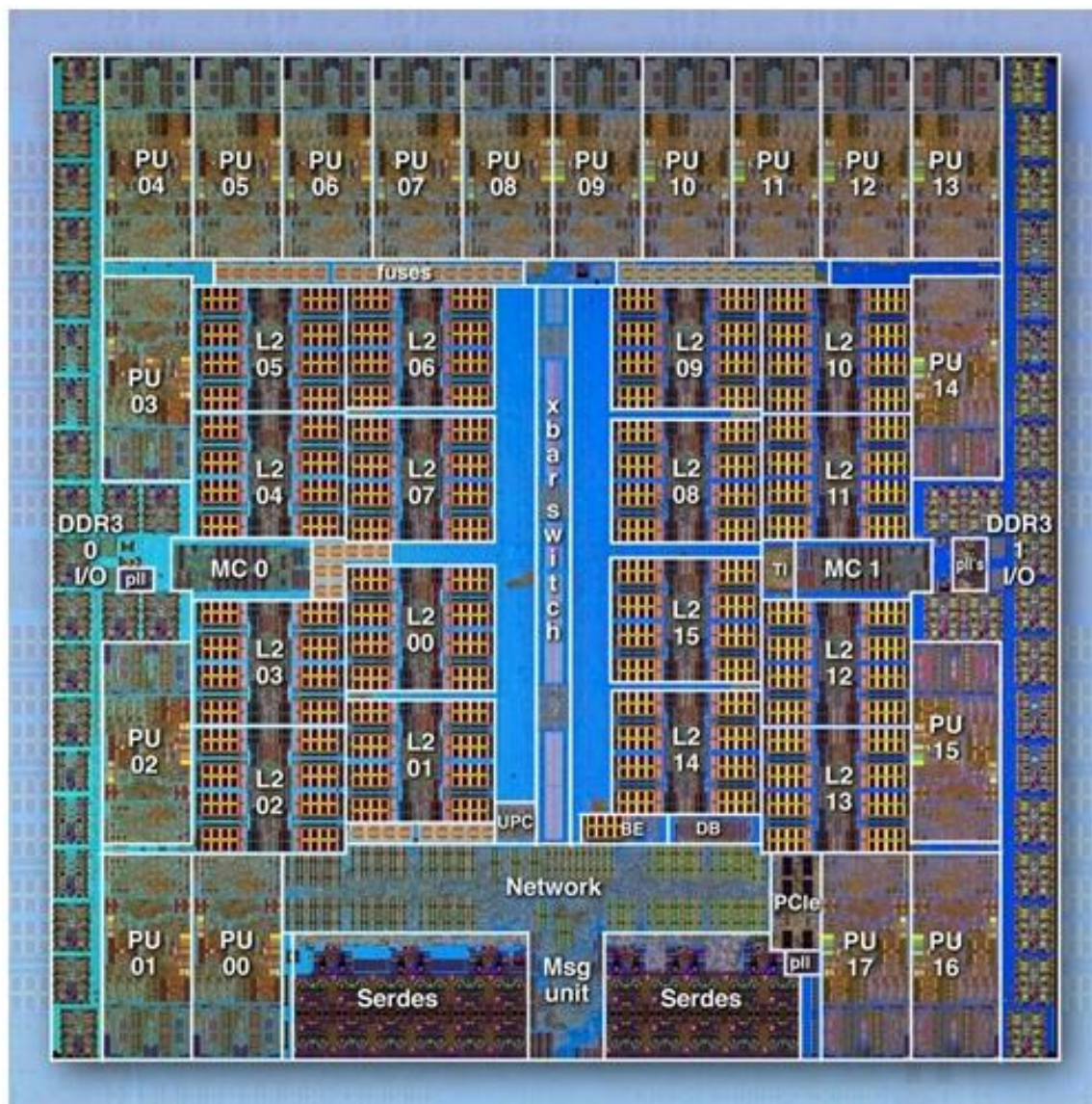
connects the processors, the L2, the networking

5D torus interconnect is also embedded on the chips

Two of these can be used for PCI-Express 2.0 x8 peripheral slots.

supports point-to-point, collective, and barrier messages and also implements direct memory access between nodes.

# PowerA2 chip, layout



# PowerA2 core

## 4 FPU

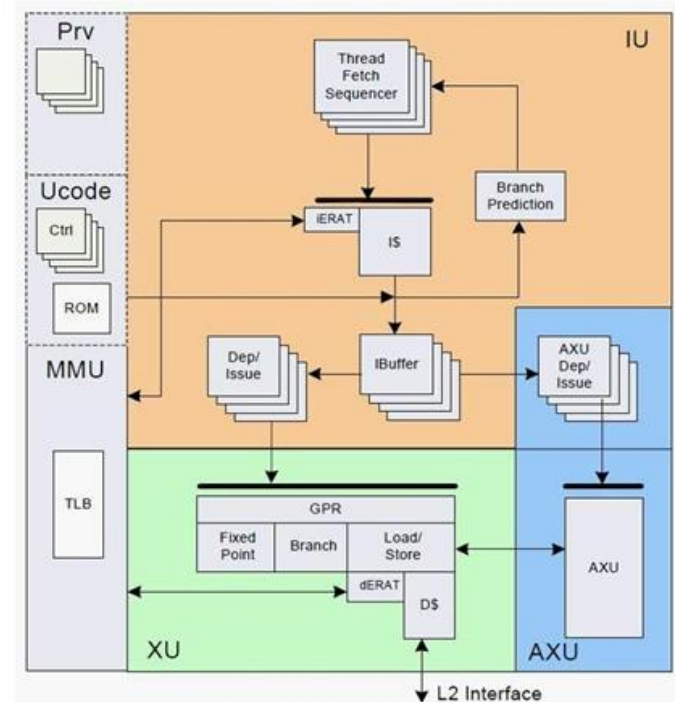
## 4 way SMT

## 64-bit instruction set

## in-order dispatch, execution, and completion

16KB of L1 data cache

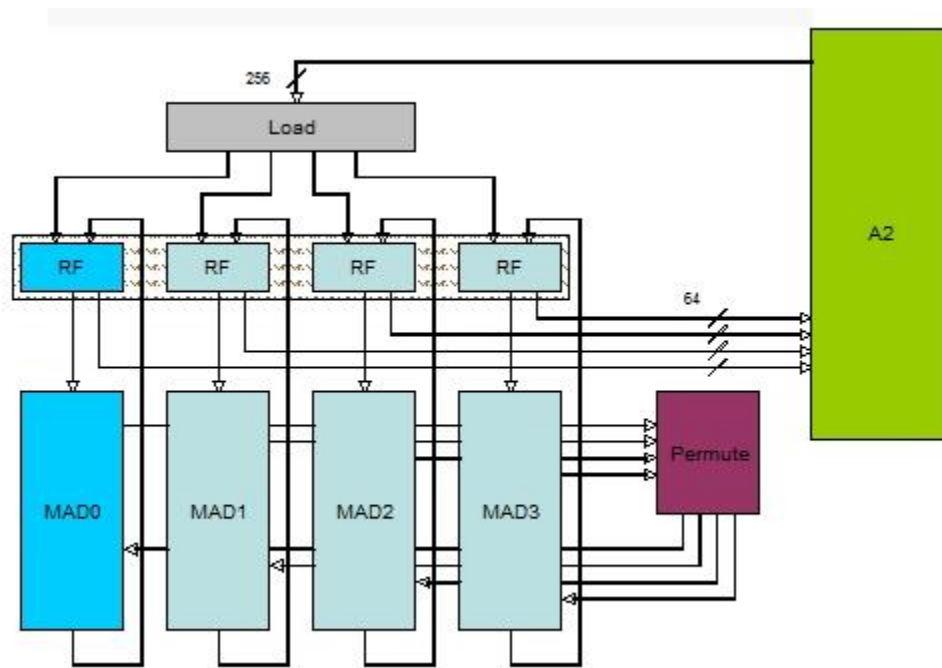
## 16KB of L1 instructions cache





# PowerA2 FPU

Each FPU on each core has four pipelines  
execute scalar floating point instructions  
Quad pumped  
four-wide SIMD instructions  
two-wide complex arithmetic SIMD inst.  
six-stage pipeline  
permute instructions  
maximum of eight concurrent  
floating point operations  
per clock plus a load and a store.





## Standards-based programming environment

- Linux™ development environment
- Familiar GNU toolchain with GLIBC, pthreads, gdb
- XL Compilers providing C, C++, Fortran with OpenMP
- Totalview debugger

## Message Passing

- Optimized MPICH2 providing MPI 2.2
- Intermediate and low-level message libraries available, documented, and open source
- GA/ARMCI, Berkeley UPC, etc, ported to this optimized layer

## Compute Node Kernel (CNK) eliminates OS noise

- File I/O offloaded to I/O nodes running full Linux
- GLIBC environment with few restrictions for scaling

## Flexible and fast Job Control

- MPMD (4Q 2012) and sub-block jobs supported

# Toolchain and Tools

## BGQ GNU toolchain

- gcc is currently at 4.4.4. Will update again before we ship.
- glibc is 2.12.2 (optimized QPX memset/memcopy)
- binutils is at 2.21.1
- gdb is 7.1 with QPX registers
- gmon/gprof thread support
  - ✓ Can turn profiling on/off on a per thread basis

## Python

- Running both Python 2.6 and 3.1.1.
- NUMPY, pynumeric, UMT all working
- Python is now an RPM

Toronto compiler test harness is running on BGQ LNs

Backup slides

# PowerPC A2 processor

Simple core, designed for excellent power efficiency and small footprint.

Embedded **64-bit PowerPC compliant**

1.60 GHz @ 0.74V.

AXU port allows for unique BGQ style floating point

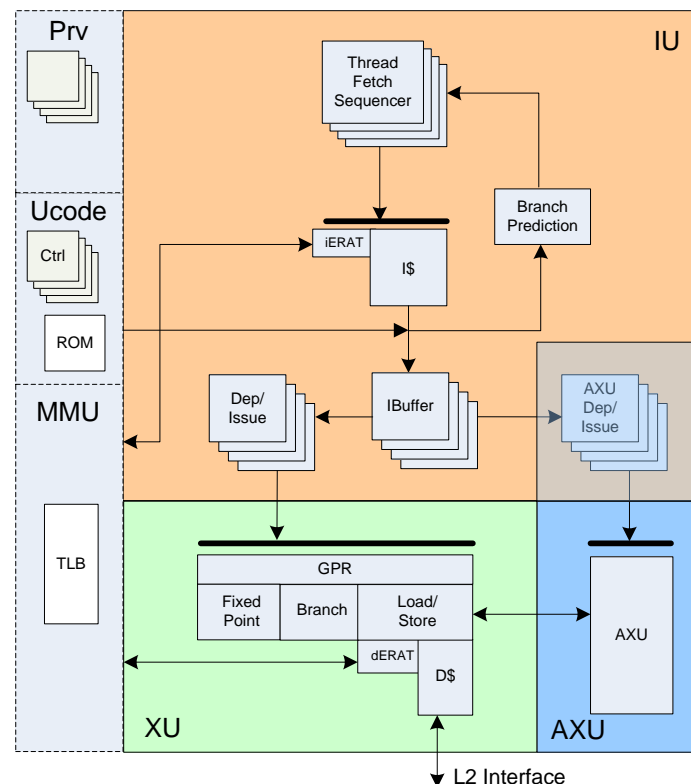
Up to 2 instructions issued per cycle

- **One FPU instruction (AXU)**
- **One Integer/Load/Store/Control instruction**

4 SMT threads issuing to two pipelines

- **Impact of memory access latency reduced**
- **At most one instruction issued per thread**
- **4 x 32 register sets**

In-order execution



- Four threads issuing to two pipelines
- Impact of memory access latency reduced

### Issue

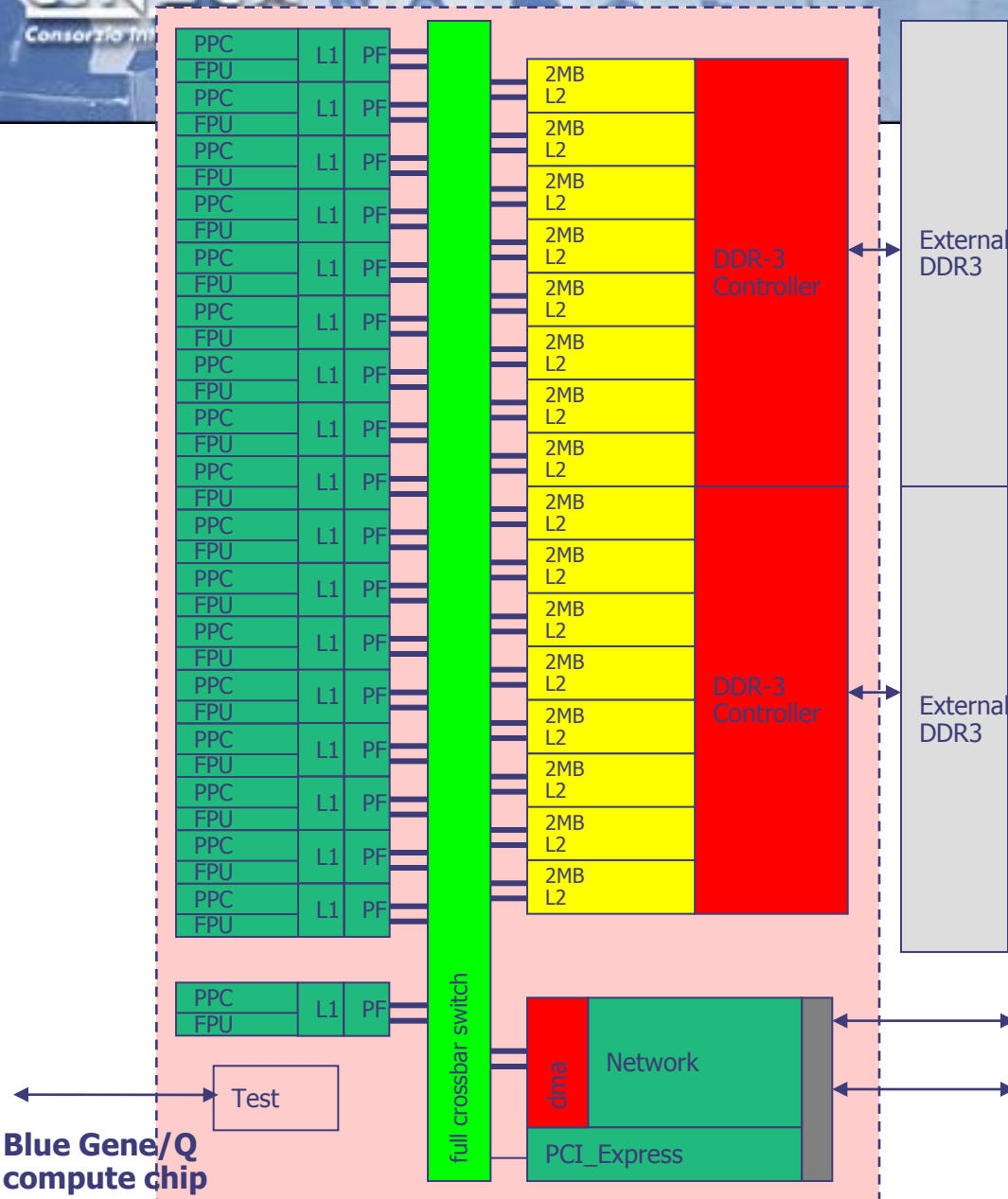
- Up to two instructions issued per cycle
  - ✓ One Integer/Load/Store/Control instruction issue per cycle
  - ✓ One FPU instruction issue per cycle
- **At most one instruction issued per thread**
  - ✓ **Only 1 FPU and 1 FXU per cycle**

### Flush

- Pipeline is not stalled on conflict
- Instead,
  - ✓ Instructions of conflicting thread are invalidated
  - ✓ Thread is restarted at conflicting instruction
- Guarantees progress of other threads



## Blue Gene/Q chip architecture



- 16+1 core SMP
    - Each core 4-way hardware threaded
  - Frequency: 1.60 GHz
  - Transactional memory and thread level speculation
  - Quad floating point unit on each core
    - 204.8 GF peak / node
  - 563 GB/s bisection bandwidth to shared L2
  - 32 MB shared L2 cache
  - 16 GB memory/node
    - 1,333 MHz DDR3
    - 2 channels each with chip kill protection
    - 42.6 GB/s bandwidth
  - 10 intra-rack interprocessor links each at 2.0GB/s
  - 1 I/O link at 2.0 GB/s
  - ~60 watts max DD1 chip power
- 2 GB/s I/O link (to I/O subsystem)
- 10\*2GB/s intra-rack & inter-rack (5-D torus)
- note: chip I/O shares function with PCI\_Express*

# PowerA2 transactional memory

Performance optimization for critical regions

1. Software threads enter “transactional memory” mode
  - Memory accesses are tracked.
  - Writes are not visible outside of the thread until committed.
2. Perform calculation without locking
3. Hardware automatically detects memory contention conflicts between threads
  - If conflict:
    - ✓ TM hardware detects conflict
    - ✓ Kernel decides whether to rollback transaction or let the thread continue
    - ✓ If rollback, the compiler runtime decides whether to serialize or retry
  - If no conflicts, threads can commit their memory

Threads can commit out-of-order.

XL Compiler only