# Introduction to High Performance Computing

## Computer Architectures

# Classification

Flynn (1966)

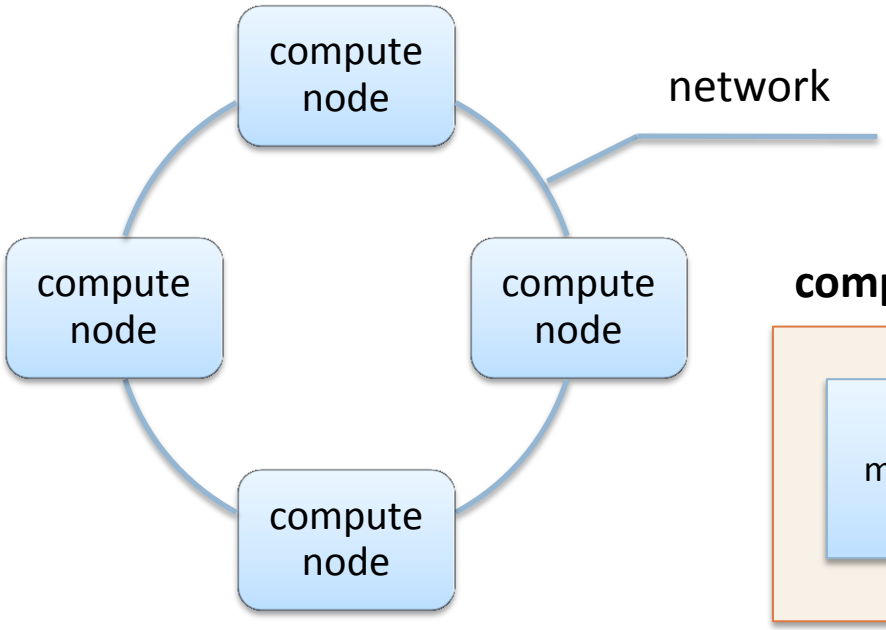
- Single Instruction / Single Data (SISD)
- Single Instruction / Multiple Data (SIMD)
- Multiple Instruction / Single Data (MISD)
- Multiple Instruction / Multiple Data (MIMD)

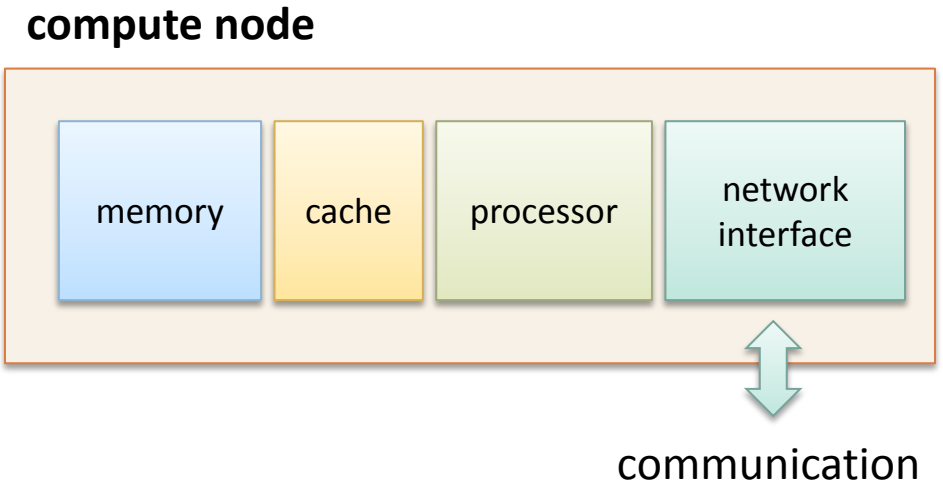
**MIMD**

Execution units are independent of each other and work asynchronously. Processors handle multiple instruction on multiple (different) data.

# memory control

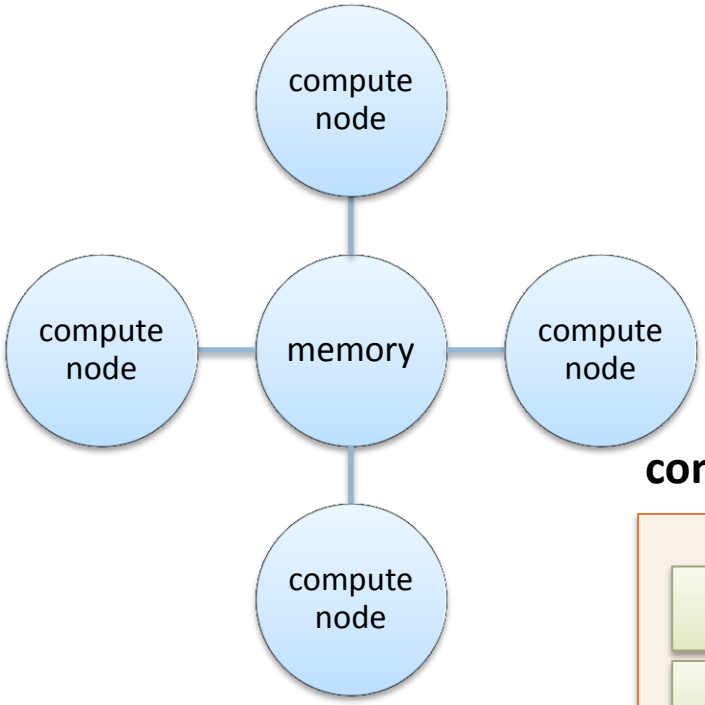
**Distributed memory**
Each processor owns it‘s own memory.



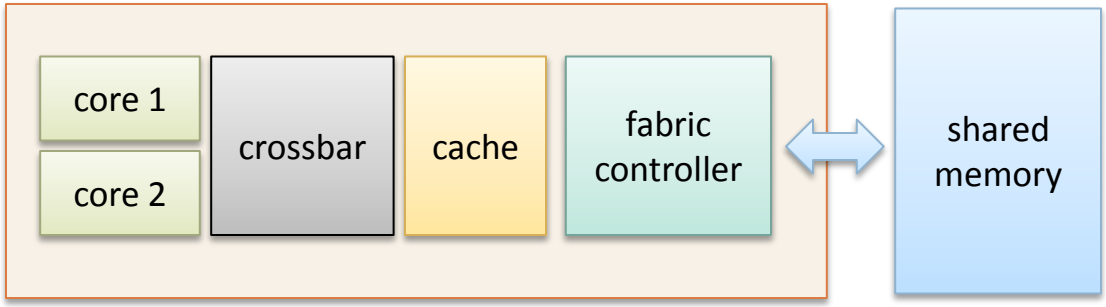



# memory control

**Shared memory**
Processor share a common memory.



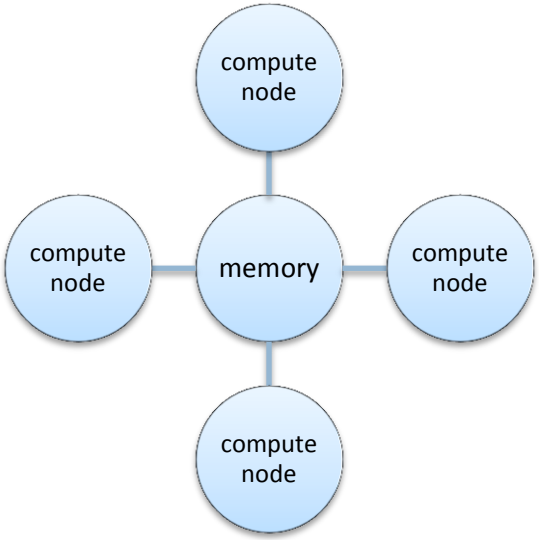

**compute node**

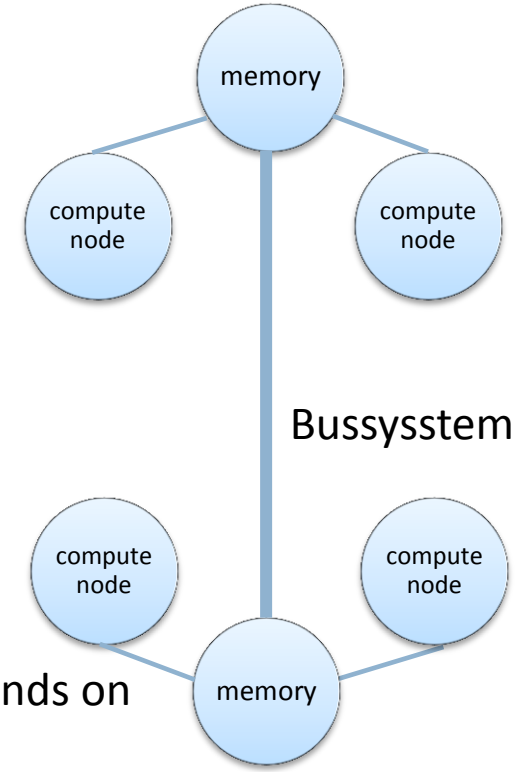


# shared memory

**multicore** processors : Traditional CPUs with several cores.
**manycore** processors: CPUs with many cores running in SIMD mode, accelerators.

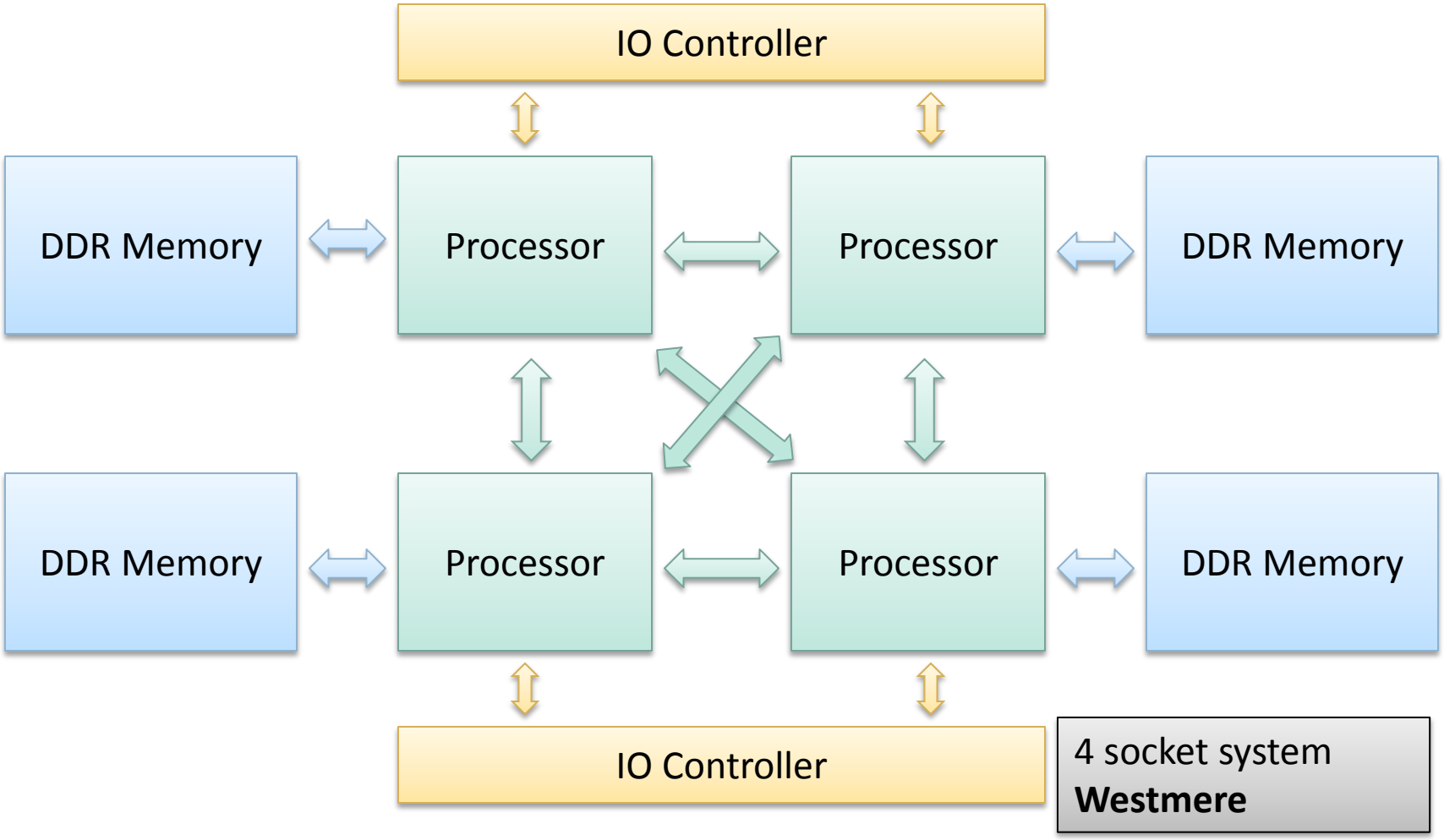
**symmetric design**: Uniform memory access time for all cores



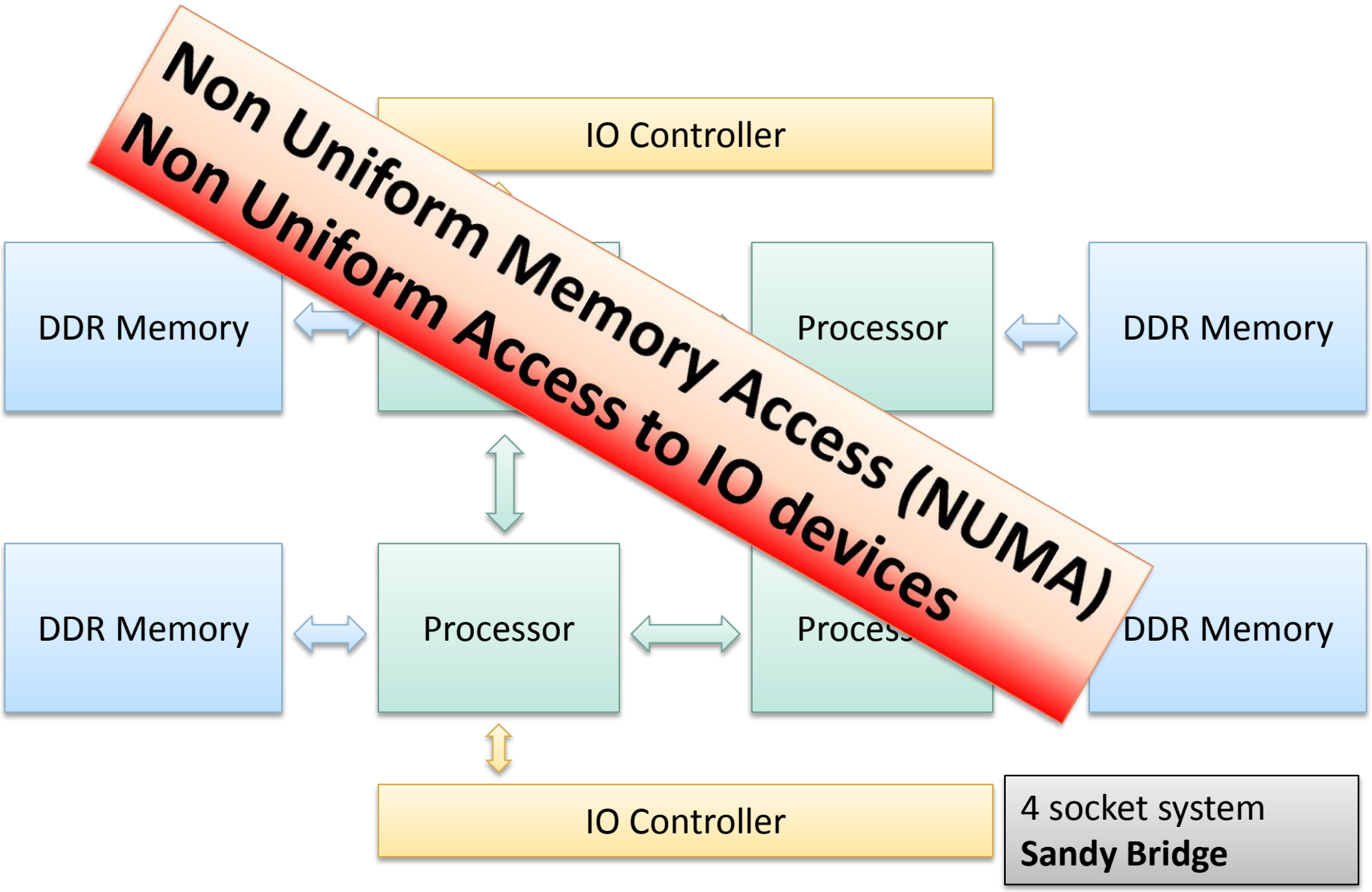



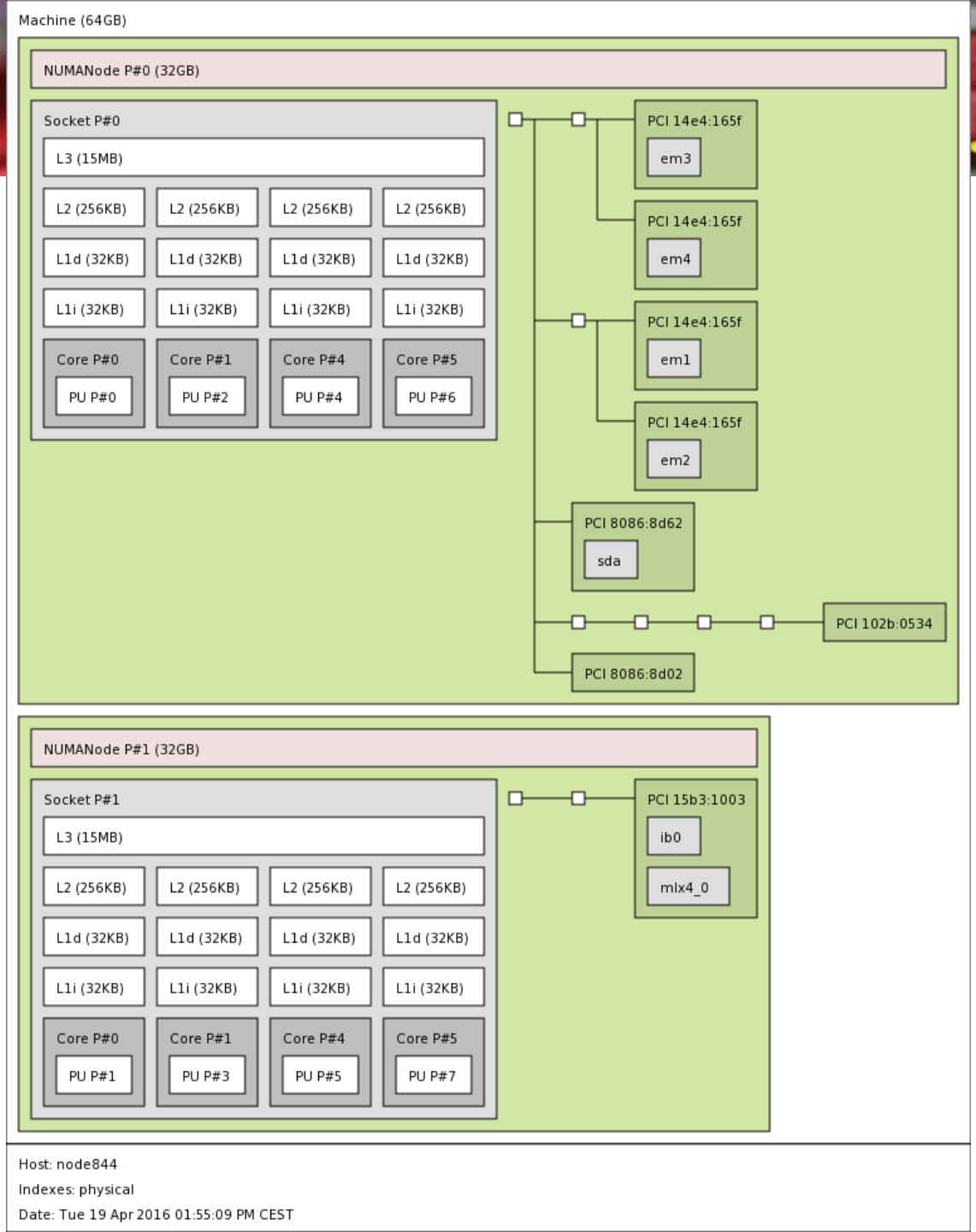

**assymmetric design**: memory access time depends on data locality

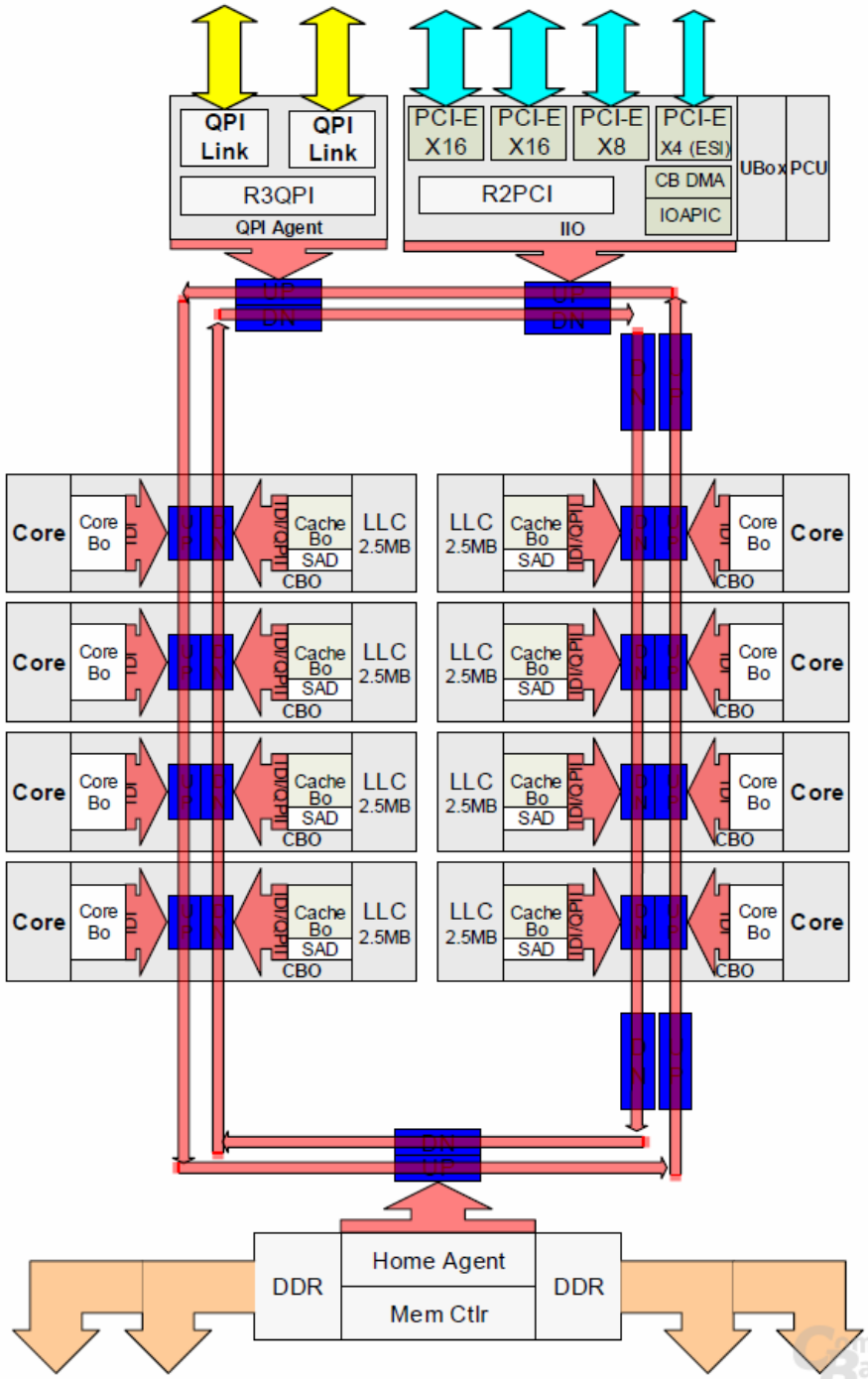
# assymetric


IO Controller
DDR Memory
Processor
Processor
DDR Memory
DDR Memory
Processor
Processor
DDR Memory
IO Controller
4 socket system
**Westmere**

# assymetric

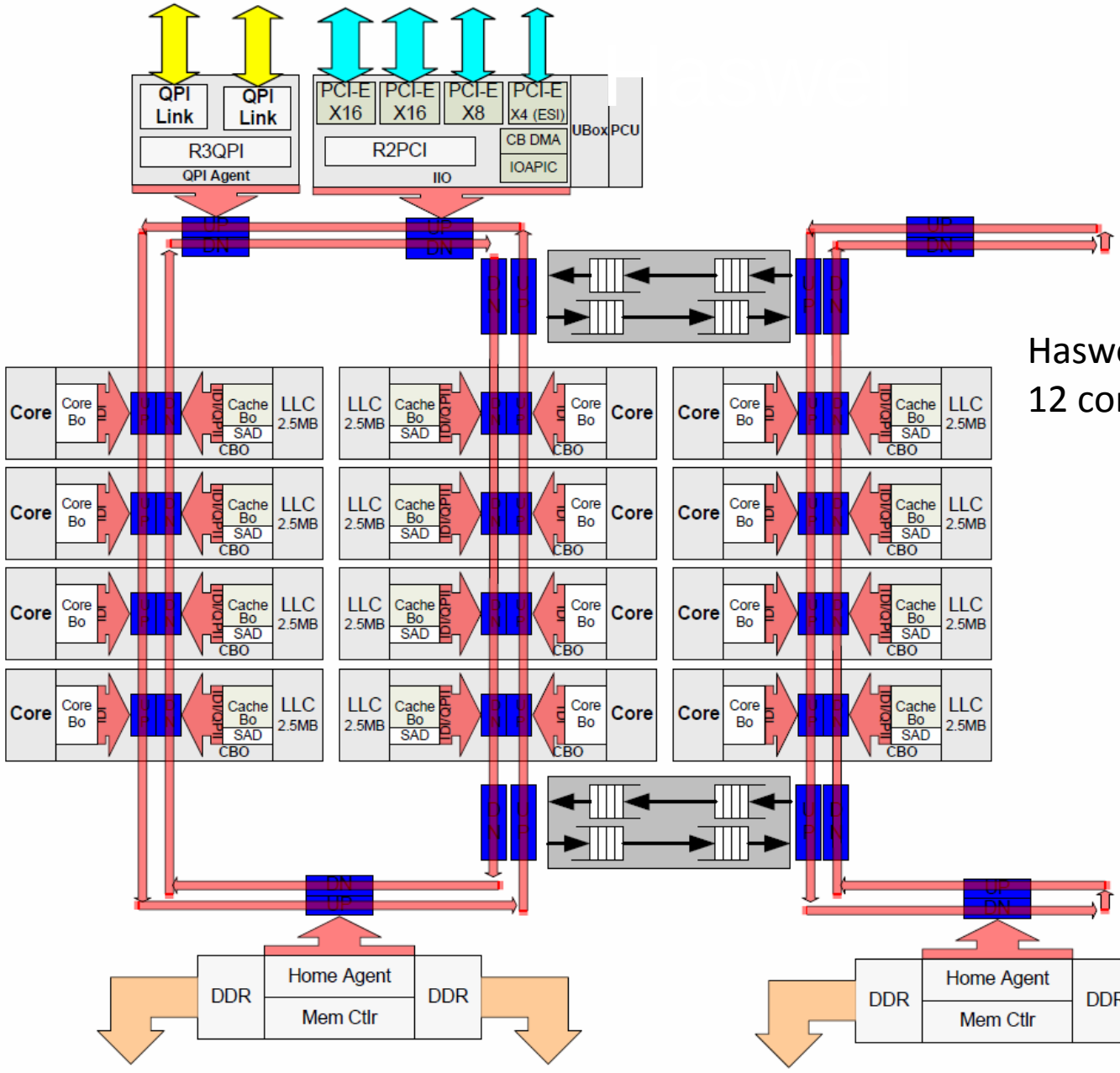

Non Uniform Memory Access (NUMA)
Non Uniform Access to IO devices
IO Controller
DDR Memory
Processor
DDR Memory
DDR Memory
Processor
DDR Memory
IO Controller
4 socket system
Sandy Bridge


Machine (64GB)
NUMANode P#0 (32GB)
Socket P#0
L3 (15MB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core P#0
PU P#0
Core P#1
PU P#2
Core P#4
PU P#4
Core P#5
PU P#6
PCI 14e4:165f
em3
PCI 14e4:165f
em4
PCI 14e4:165f
em1
PCI 14e4:165f
em2
PCI 8086:8d62
sda
PCI 102b:0534
PCI 8086:8d02
NUMANode P#1 (32GB)
Socket P#1
L3 (15MB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core P#0
PU P#1
Core P#1
PU P#3
Core P#4
PU P#5
Core P#5
PU P#7
PCI 15b3:1003
ib0
mlx4_0
Host: node844
Indexes: physical
Date: Tue 19 Apr 2016 01:55:09 PM CEST




Haswell architecture with 8 cores per CPU

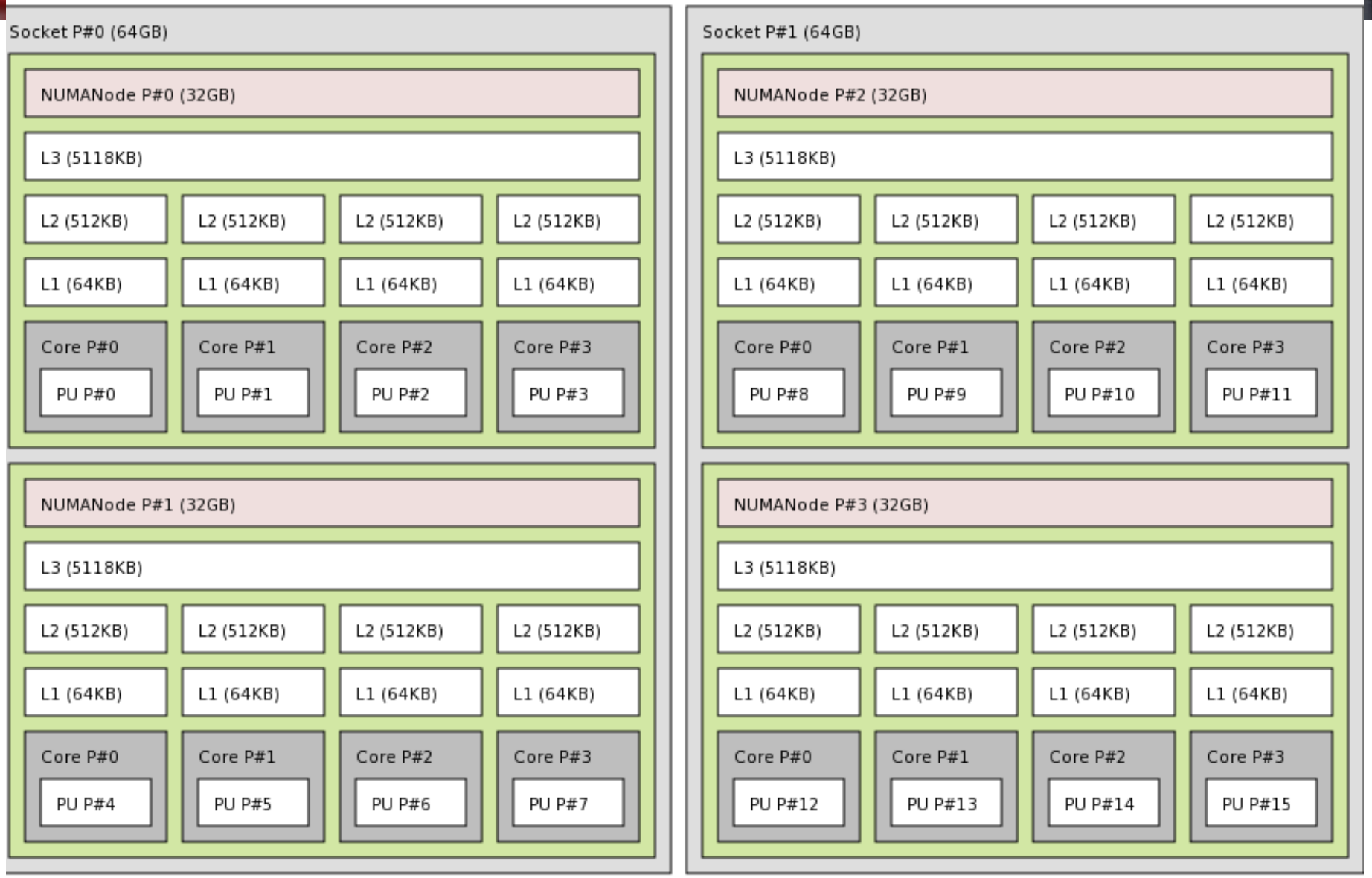
Source: pics.computerbase.de



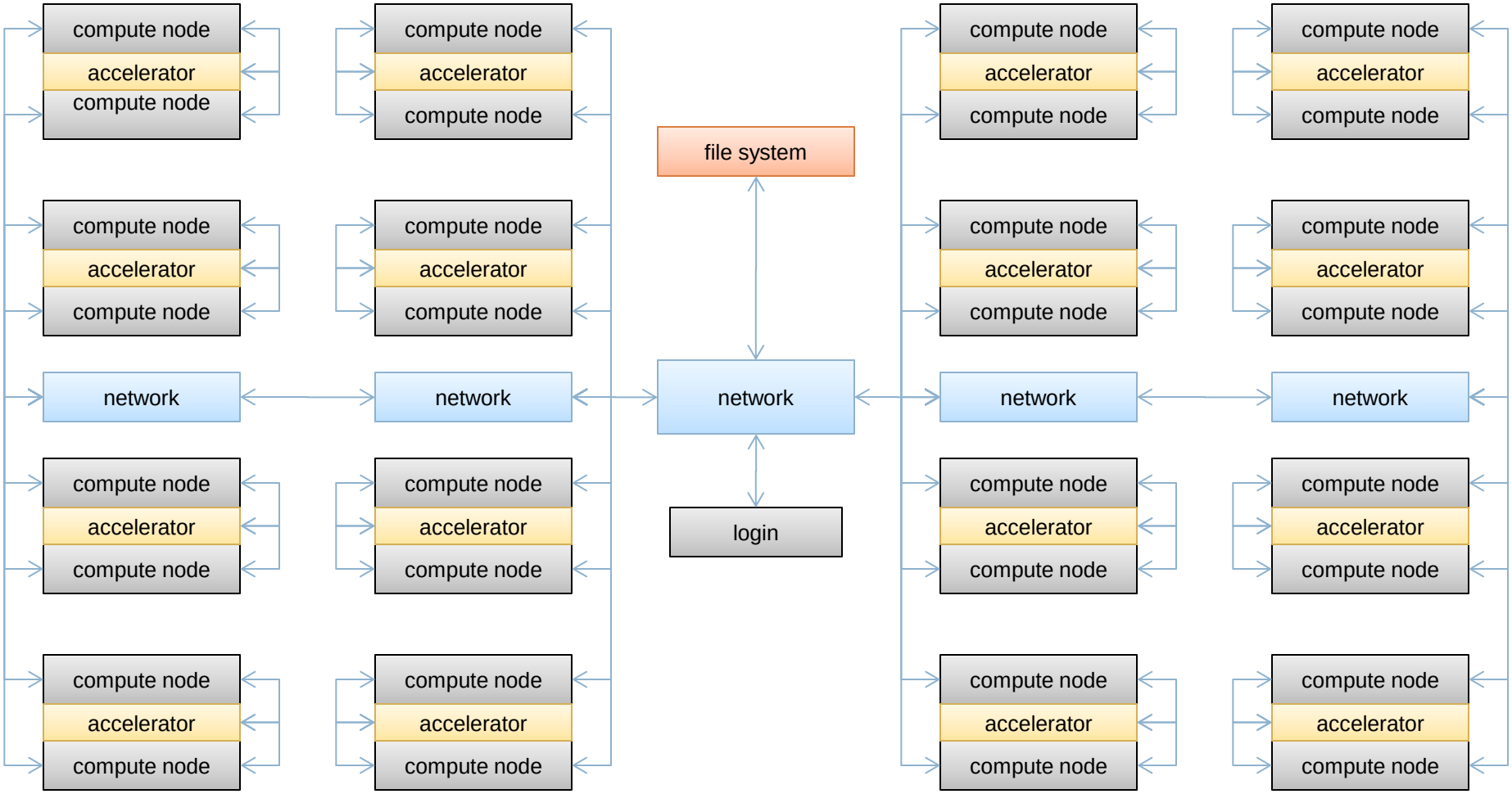

Haswell architecture with 12 cores per CPU

Source: pics.computerbase.de

# AMD Magny Cours



# Building a cluster

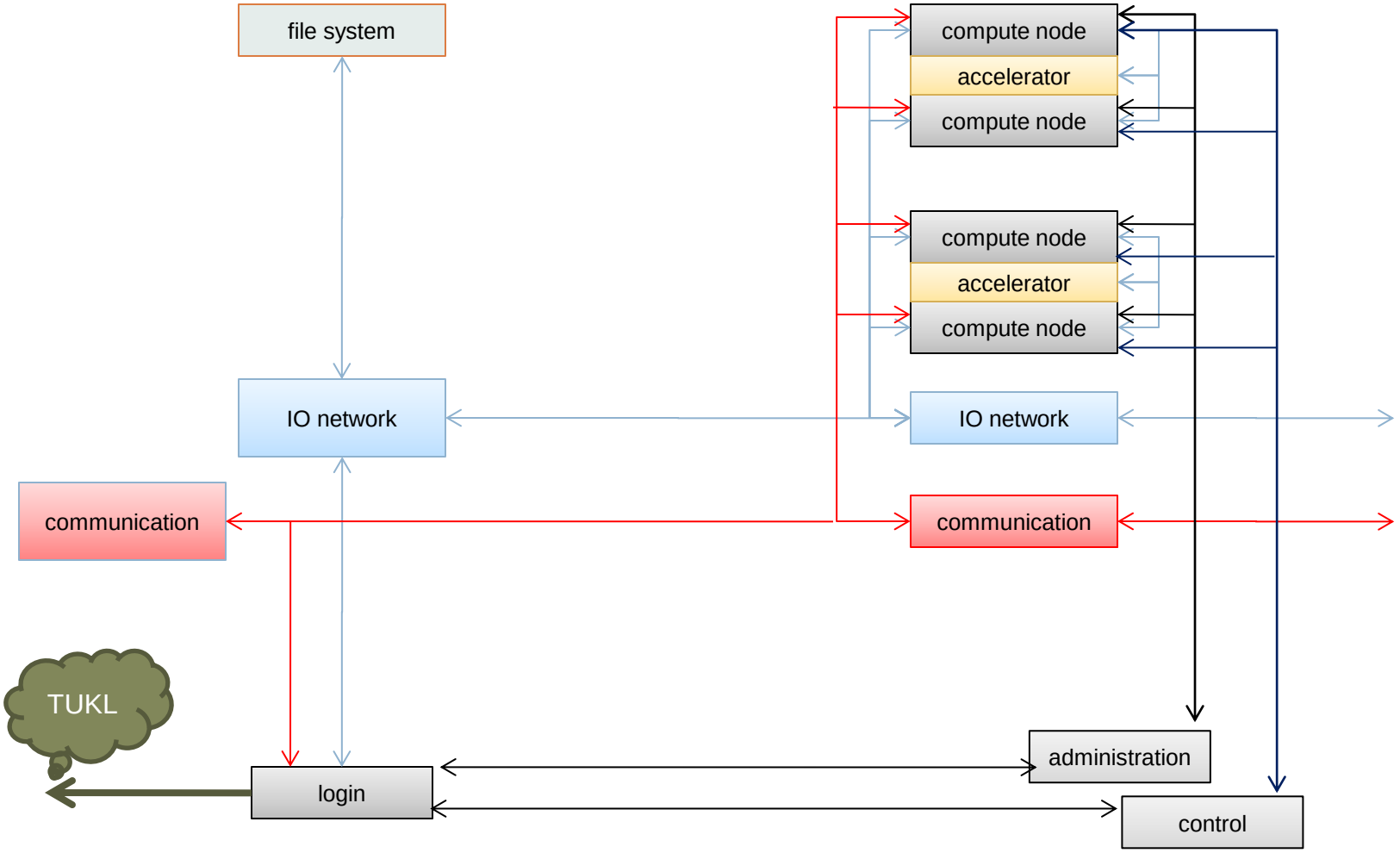


# Building a cluster

**Close look on cluster network**

| function | demands |
|---|---|
| computer control (power on/off, sensor data) | cheap, simple network (100 Mb) |
| computer administration (resource management, boot images) | cheap, medium quality (1GE) |
| File operations (IO) | high bandwidth |
| interprocess communication | low latency |

# Building a cluster



# Building a cluster

Networks for I/O and communication are sometimes

- combined to a single network
  - Quality of service
- used as failover/double bandwidth (dual rail)

# network technologies

simplest strategy: daisy chain


Source: Ebay Kleinanzeigen

Source: Reichelt, Online Shop

36 Port Ethernet Switch:

IN: 34 computers
IN/OUT: 2 ports to connect 2 neighboring switches.

# network technologies

What characterizes a high performance network?

- Which bandwidth is available for a point-to-point connection in a daisy chain?
- Which bandwidth is available for all pairwise p2p connections?
- Which latency is observed in a daisy chain (each network component adds a bit)?
- Which differencies in the latencies occur?

# network technologies

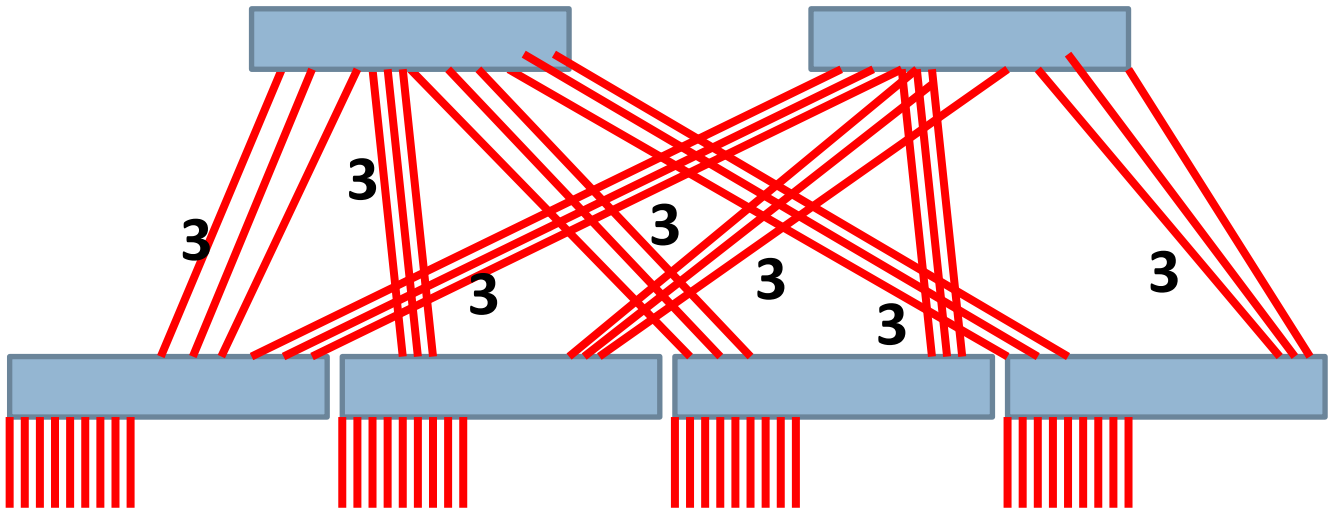
Bisectional bandwidth of a network:
The network is segmented in 2 parts – which aggregate bandwidth is possible between these parts?

A daisy chain's bisectional bandwidth is only twice as large as the bandwidth of each connection.

**Fully non blocking network**:
The bisectional bandwidth is the bandwidth of each connection multiplied with the number of connections.
All pairwise disjunct connections may at any time receive the full bandwidth of a single connection.

# network technologies

Given Switches with 36 Ports – How to set up a nonblocking network for 72 compute nodes?




Known as fat tree topology

# network technologies

nonblocking:
• 18 Ports to nodes,
• 18 ports to switch in next layer
preserving full bandwidth for all pairs

CLOS2 network
full fat tree

- Number of hops 3.
- Maximum of 2 hops difference in pairwise connections.

more information: Clos network

# Building a cluster

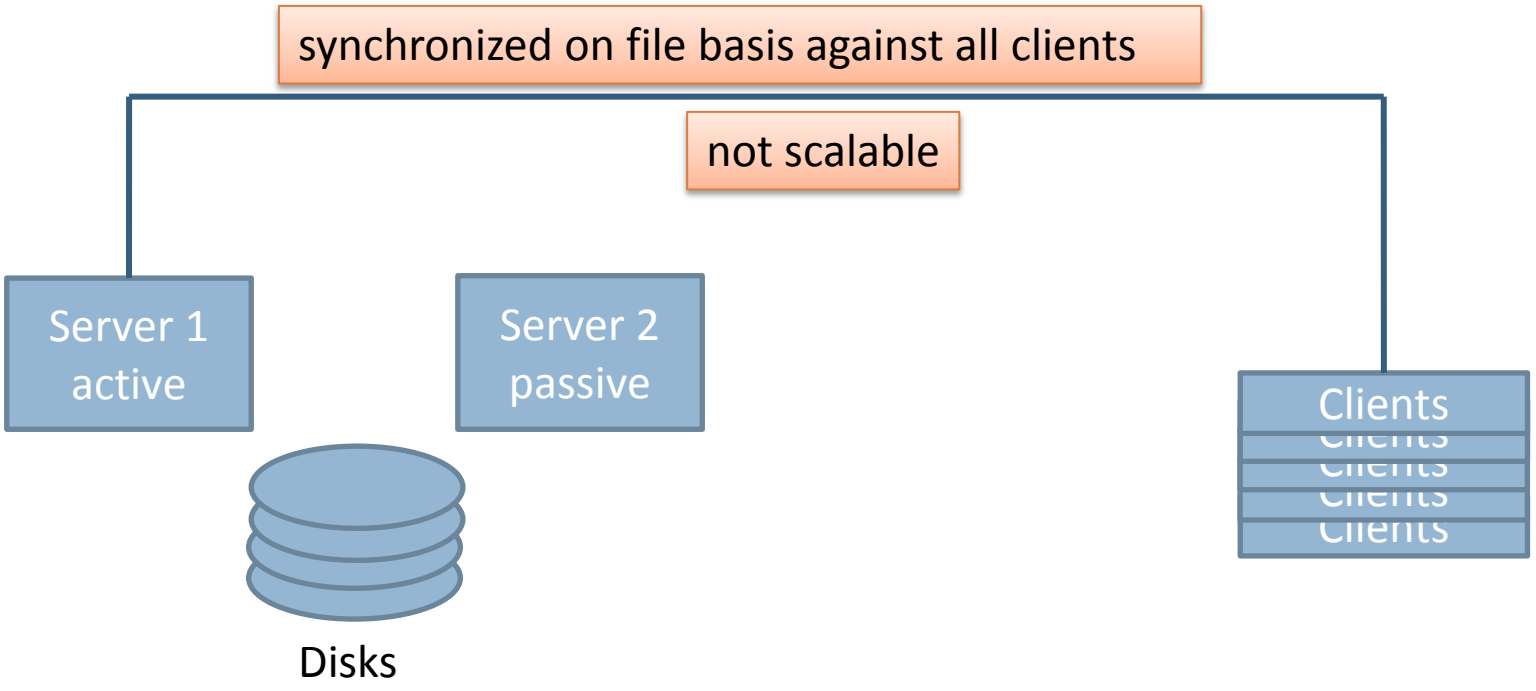
Filesystem requests

- Network boot or local boot (local disk on each node)
- small HOME available on all nodes (backed up)
  - read/write mainly from login nodes
  - read mainly from computing nodes
  - important task: compilation (many small files)
- large SCRATCH available on all nodes (short lifetime files)
  - read/write from computing nodes
  - important task: store large temporary files
  - write a file in parallel from several nodes

# Building a cluster

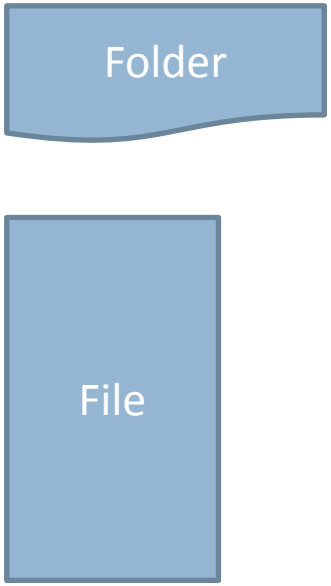
Filesystems in Kaiserslautern:

• NFS for HOME, realized with two fileservers (fault tolerant, active/passive coupling)

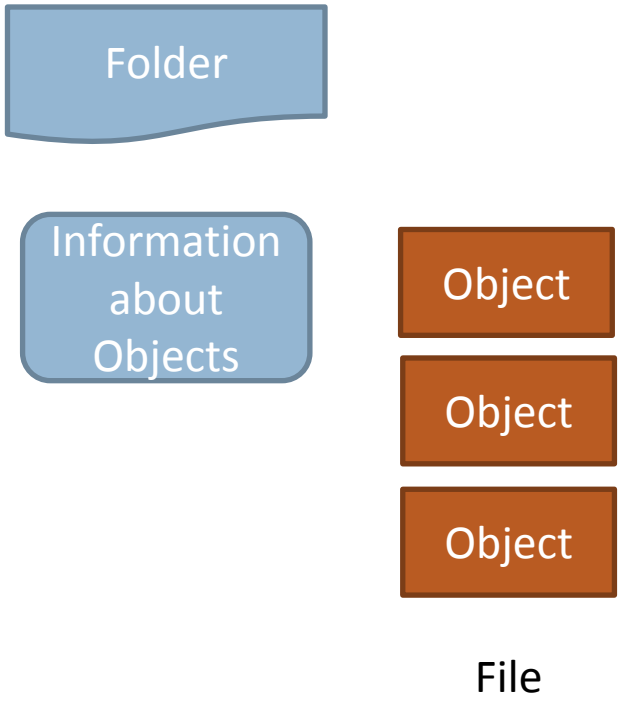


# Scalable File System

**normal view**
a file is stored and linked to in folder structure
Link and file build a unity




**object based storage**
file is divided into objects
objects may be handled independently
information on objects is linked in folder structure (**meta data**)



# Building a cluster

Filesystems in Kaiserslautern:
parallel FS (FhGFS) for SCRATCH, realized with 10 fileservers in 5 pairs.

- Objects are independent
  - scalable with number of fileservers (offers 10-times the bandwidth for the cluster than 1 fileserver)
  - No synchronization among clients when accessing objects

- decoupling of data and directory structure
  - extra servers for meta data
  - scalable with number of meta data servers

# Building a cluster

Resource management

- all users log in on login nodes (4)
- interactive nodes with high performing graphics cards (10)
- calculations on compute node
- How do we alloc requested compute resources fairly?
  - 2 nodes with 4 proc. each for user A
  - 1 node with 21 proc. for user B
  - 16 nodes with 8 proc. each for user C
- no knowledge on which of the nodes a job will start

Solution: batch scheduling software (load sharing facility (LSF) from IBM (previous Platform Computing).

Einführung in das Hochleistungsrechnen

Introduction to High Performance Computing

# VIELEN DANK
# THANK YOU