MHPC
Master in High Performance Computing

Lecture 6
HPC platforms and associated tools

Stefano Cozzini

CNR/IOM and eXact-lab srl

SISSA
Scuola Internazionale Superiore di Studi Avanzati

ICTP The Abdus Salam International Centre for Theoretical Physics

MHPC
Master in High Performance Computing

Agenda for today

- Remind on computational infrastructure/ecosytem
- HPC cluster components
    - PCI -buses
    - Networks
    - storage
- Software tools and how to use them at best
    - LRMS
    - MPI libraries
- EXERCISE:
    - Start evaluating the overall performance of ore than one node

SISSA ICTP

MHPC
Master in High Performance Computing

What is a computational infrastructure?

SISSA

ICTP

MHPC
Master in High Performance Computing

Infrastructures for computational science

- powerful and modern clusters of multicores.
  - hardware
  - software

High Performance Computing

- pooling of resources geographically distributed
- distribute collaborations

GRID COMPUTING

- Infrastructure as a Service

CLOUD COMPUTING

SISSA ICTP

MHPC
Master in High Performance Computing

From infrastructure to ecosystem

Goal:

provide a computational ecosystem to satisfy all the different requirements posed by users

Which kind of requirements ?

All you need to to solve their computational problems !

SISSA ICTP

MHPC
Master in High Performance Computing

Elements of the computational environment

- powerful and modern parallel hardware (HPC)
- pooling of resources geographically distributed (GRIDs)
- Infrastructure as as service (CLOUD)

HARDWARE

- Middleware for HPC/
- Scientific software: Models &analysis
- Software for distribute collaboration and data sharing

SOFTWARE

- IT- skilled computational scientists
- Partnership between IT people and scientists

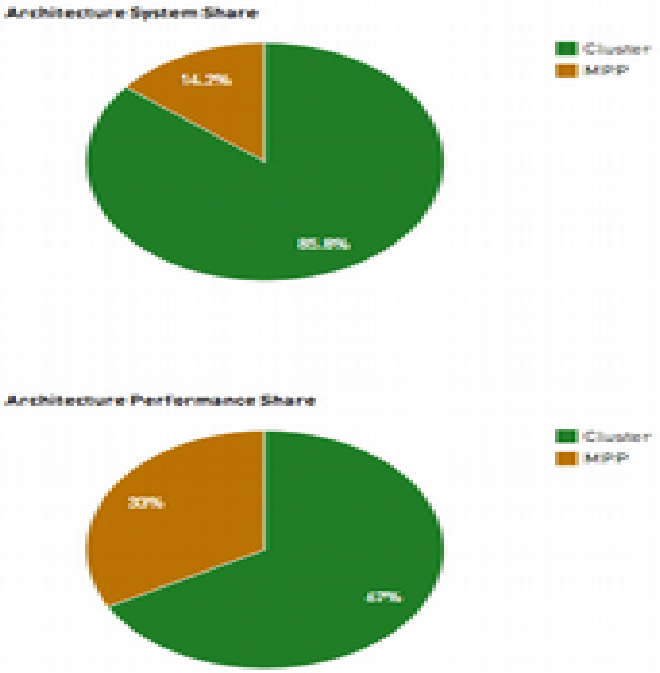
BRAINWARE

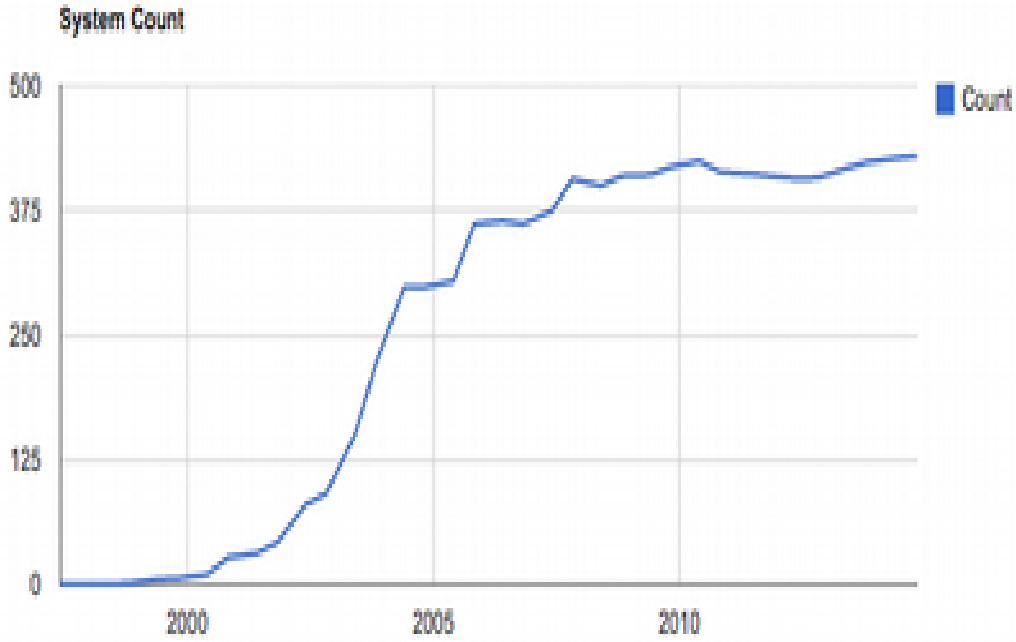
SISSA ICTP

MHPC
Master in High Performance Computing

HPC infrastructures..

.. mean
CLUSTERS !!





MHPC
Master in High Performance Computing

What's a cluster?

- A cluster **needs**:
  - Several computers, nodes, often in special cases for easy mounting in a rack
  - One or more networks (interconnects) to hook the nodes together
  - Software that allows the nodes to communicate with each other (e.g. MPI)
  - Software that reserves resources to individual users
- A cluster **is**: all of those components working together to form one big computer

SISSA ICTP

MHPC
Master in High Performance Computing

What is a Beowulf Clusters ?

- Subject: Re: [Beowulf] about concept of beowulf clusters

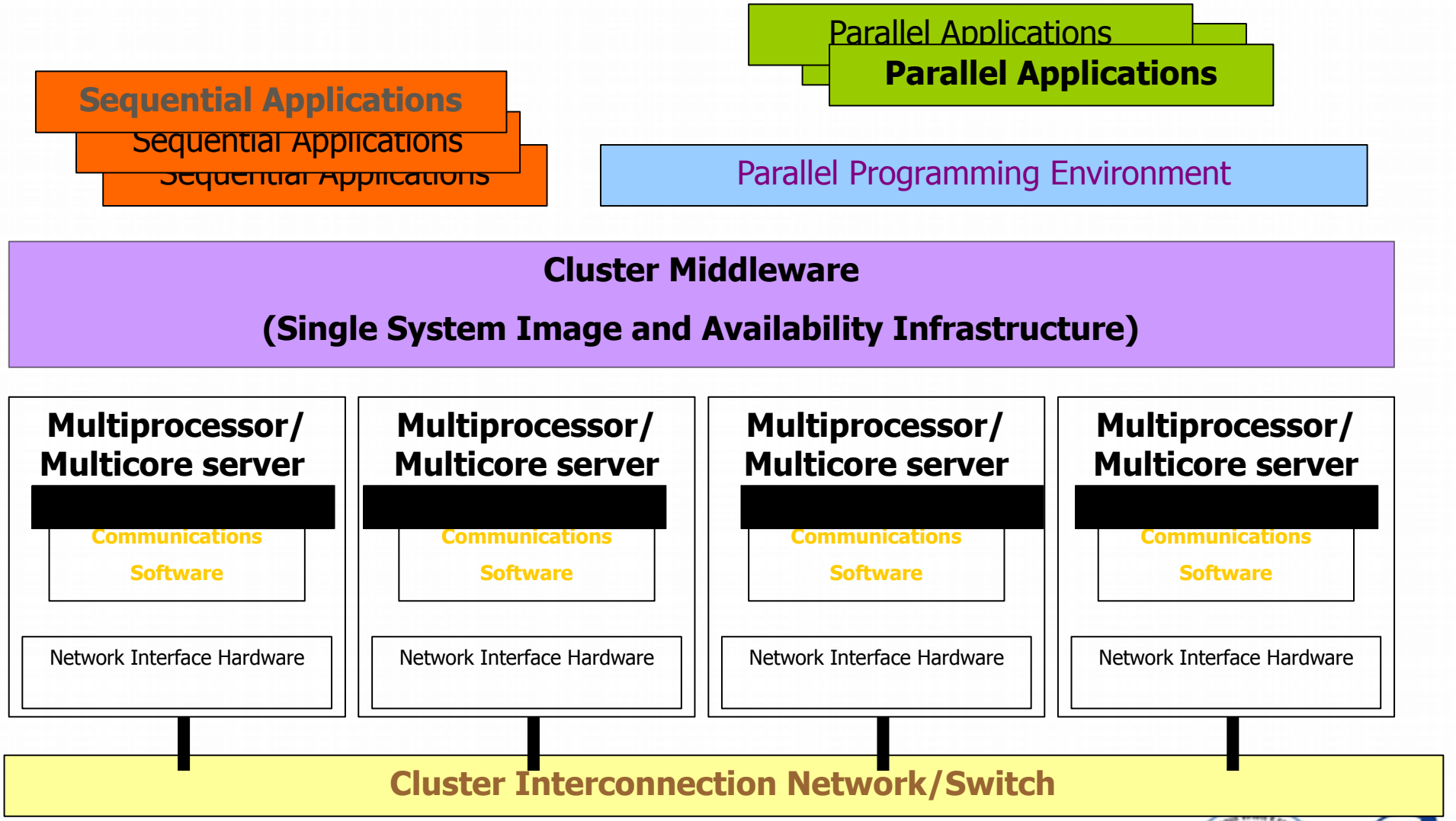
  Date: Thu, 24 Feb 2005 19:41:22 -0500 (EST)

  From: Donald Becker <becker@scyld.com>
- The Beowulf definition is commodity machines connected by a private cluster network running an open source software infrastructure for scalable performance computing
- this means:
  - commodity machines: exclude custom built hardware e.g. an IBM BlueGene is not a Beowulf cluster
  - connected by a cluster network: These machines are dedicated to being a cluster, at least temporarily. This excludes Network of workstations and others
  - running an open source infrastructure The core elements of the system are open source and verifiable
  - for scalable performance computing The goal is to scale up performance over many dimension. Ideally a cluster incrementally scales both up and down, rather than being a fixed size.

SISSA ICTP

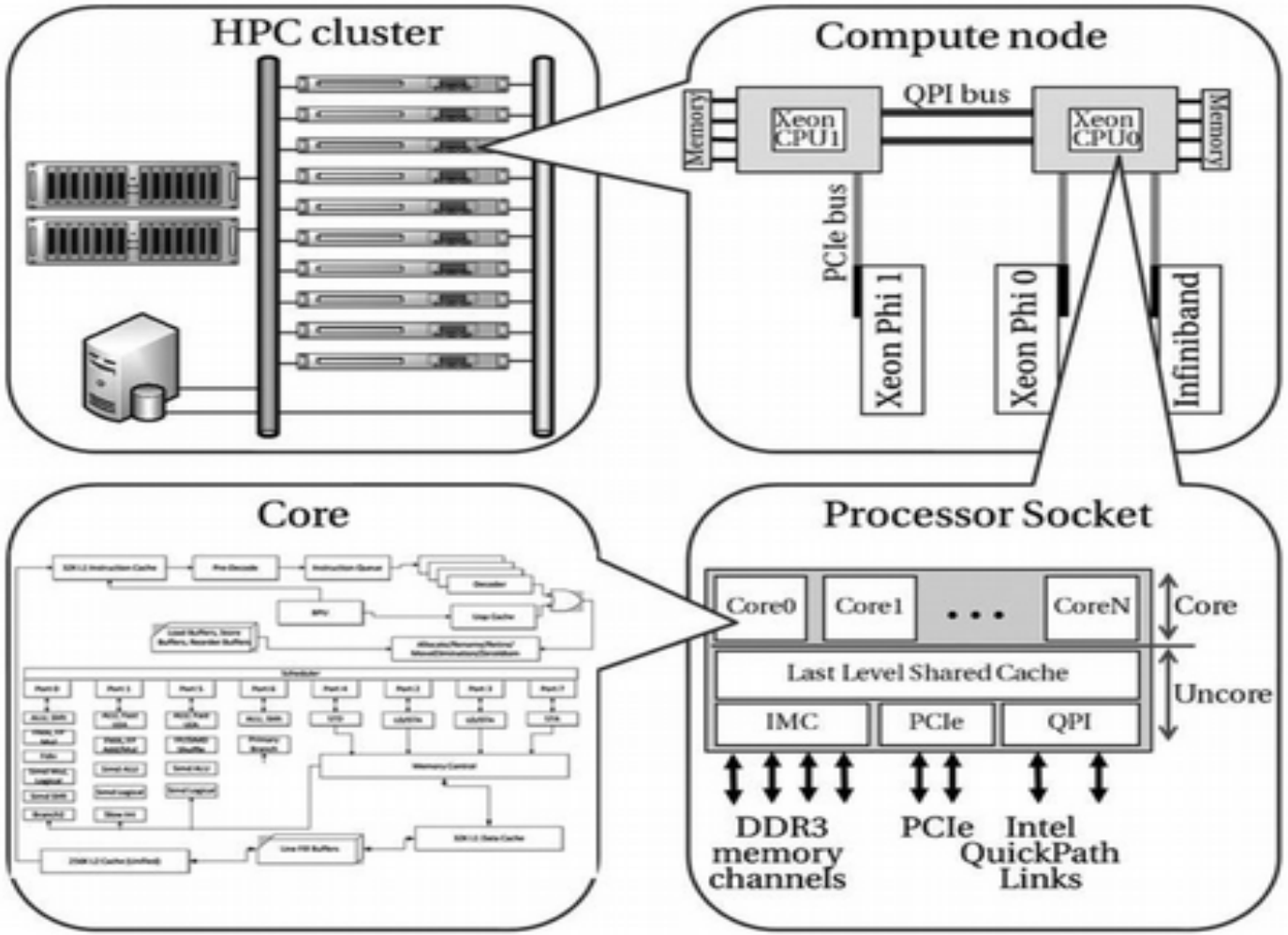
MHPC
Master in High Performance Computing

Cluster Computer Architecture



MHPC
Master in High Performance Computing

Building blocks of a cluster: nodes !



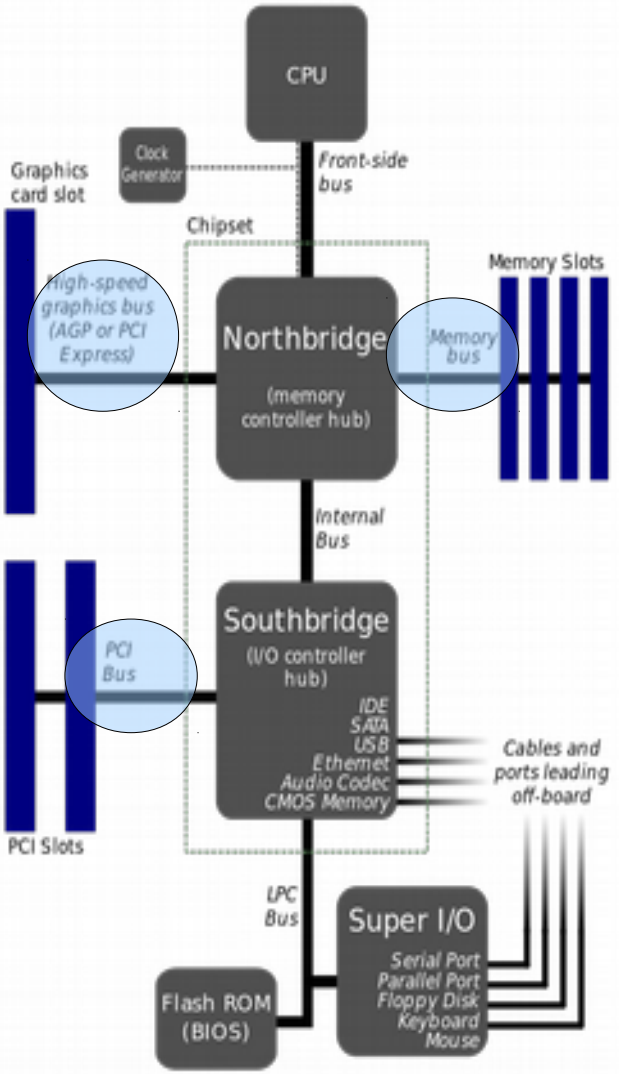
SISSA

ICTP

MHPC
Master in High Performance Computing

standard modern architecture

- All data communication from one CPU to another must travel over the same bus used to communicate with the Northbridge.
- All communication with RAM must pass through the Northbridge.
- Communication between a CPU and a device attached to the Southbridge is routed through the Northbridge.



SISSA ICTP

MHPC
Master in High Performance Computing

How fast are memories ?

- Synchronous dynamic random-access memory (SDRAM)
- Double Data Rate (DDR) with ECC
- DDR ->DDR2->DDR3-->DDR4

DDR SDRAM Standard	Internal rate (MHz)	Bus clock (MHz)	Prefetch	Data rate (MT/s)	Transfer rate (GB/s)	Voltage (V)
SDRAM	100-166	100-166	1n	100-166	0.8-1.3	3.3
DDR	133-200	133-200	2n	266-400	2.1-3.2	2.5/2.6
DDR2	133-200	266-400	4n	533-800	4.2-6.4	1.8
DDR3	133-200	533-800	8n	1066-1600	8.5-14.9	1.35/1.5
DDR4	133-200	1066-1600	8n	2133-3200	17-21.3	1.2

SISSA ICTP

MHPC
Master in High Performance Computing

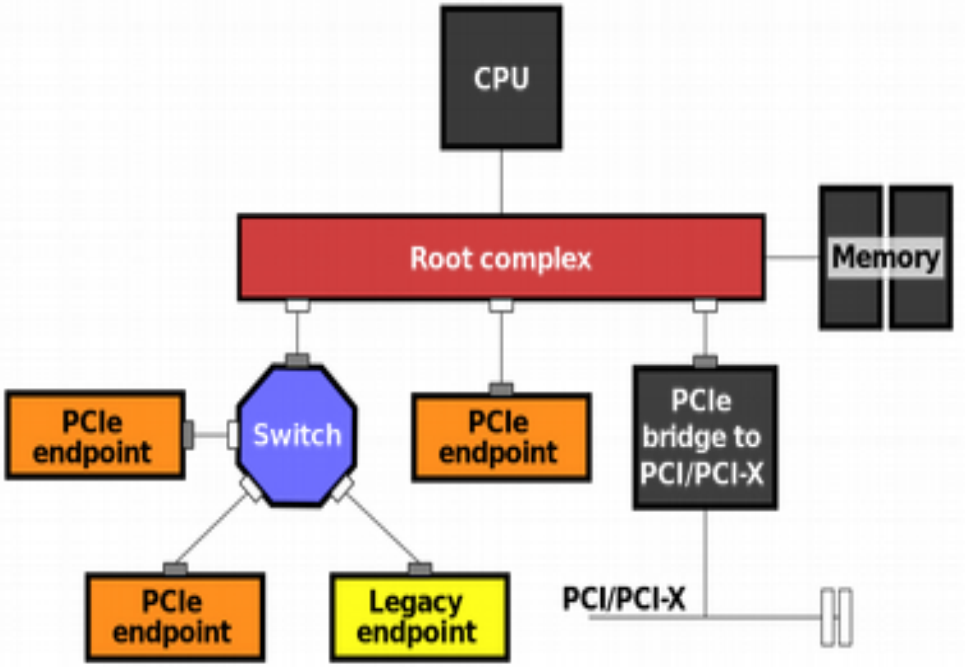
Buses on modern HPC nodes

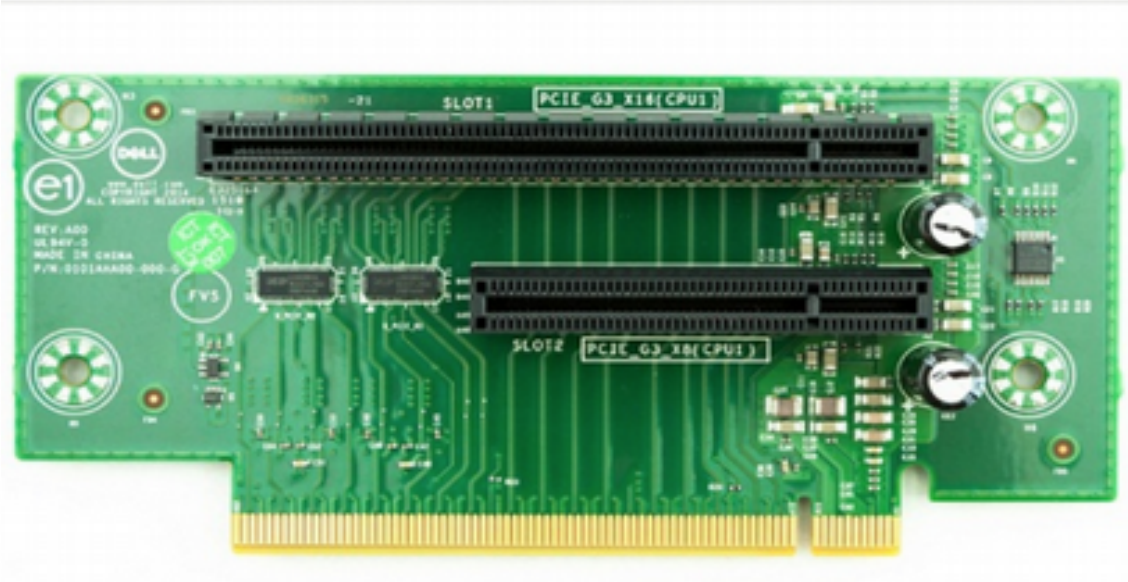
- Peripheral Component Interconnect (PCI) buses:
  - PCI: Developed by Intel in 1992
    - several version : v3.0 last one in 2004
  - PCI-X: designed in 1999
    - 66 MHz (can be found on older servers)
    - 133 MHz (most common on modern servers)
  - PCIe: designed adopted in 2004
    - version v4.0 recently released
    - Version 2.0/version 3.0 adopted on modern HPC nodes
- Several of them on one node with different characteristics

SISSA ICTP

MHPC
Master in High Performance Computing

PCI-express layout:





SISSA
ICTP

MHPC
Master in High Performance Computing

PCI-express speed (from wikipedia)

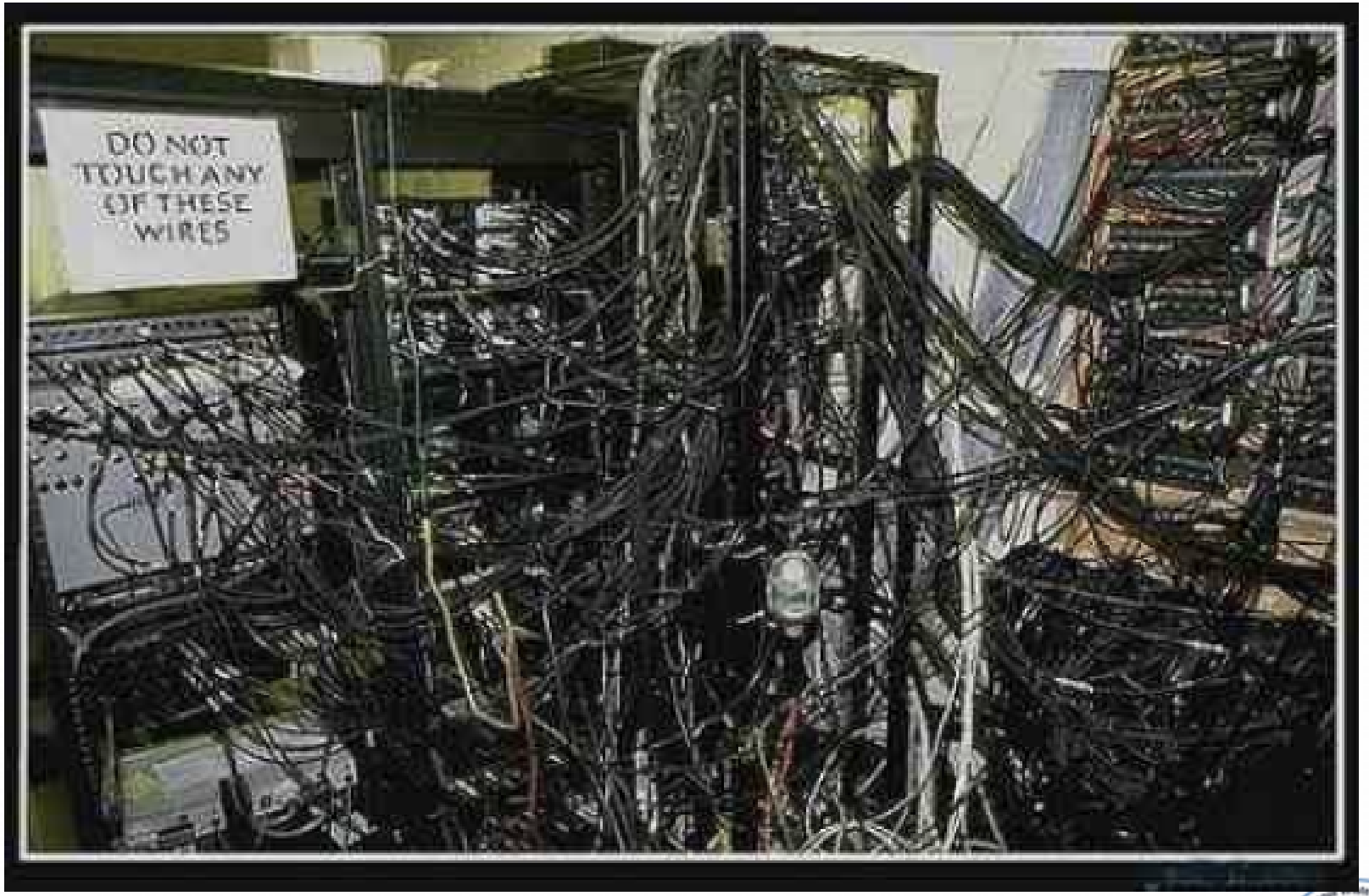
PCI Express link performance[30][31]

PCI Express version	Introduced	Line code	Transfer rate[i]	Throughput[i]				
				x1	x2	x4	x8	x16
1.0	2003	8b/10b	2.5 GT/s	250 MB/s	0.50 GB/s	1.0 GB/s	2.0 GB/s	4.0 GB/s
2.0	2007	8b/10b	5.0 GT/s	500 MB/s	1.0 GB/s	2.0 GB/s	4.0 GB/s	8.0 GB/s
3.0	2010	128b/130b	8.0 GT/s	984.6 MB/s	1.97 GB/s	3.94 GB/s	7.88 GB/s	15.8 GB/s
4.0	2017	128b/130b	16.0 GT/s	1969 MB/s	3.94 GB/s	7.88 GB/s	15.75 GB/s	31.5 GB/s
5.0[32][33]	*expected in Q2 2019*[34]	128b/130b	32.0 GT/s[ii]	3938 MB/s	7.88 GB/s	15.75 GB/s	31.51 GB/s	63.0 GB/s

SISSA
ICTP

MHPC
Master in High Performance Computing

network for clusters..



SISSA ICTP

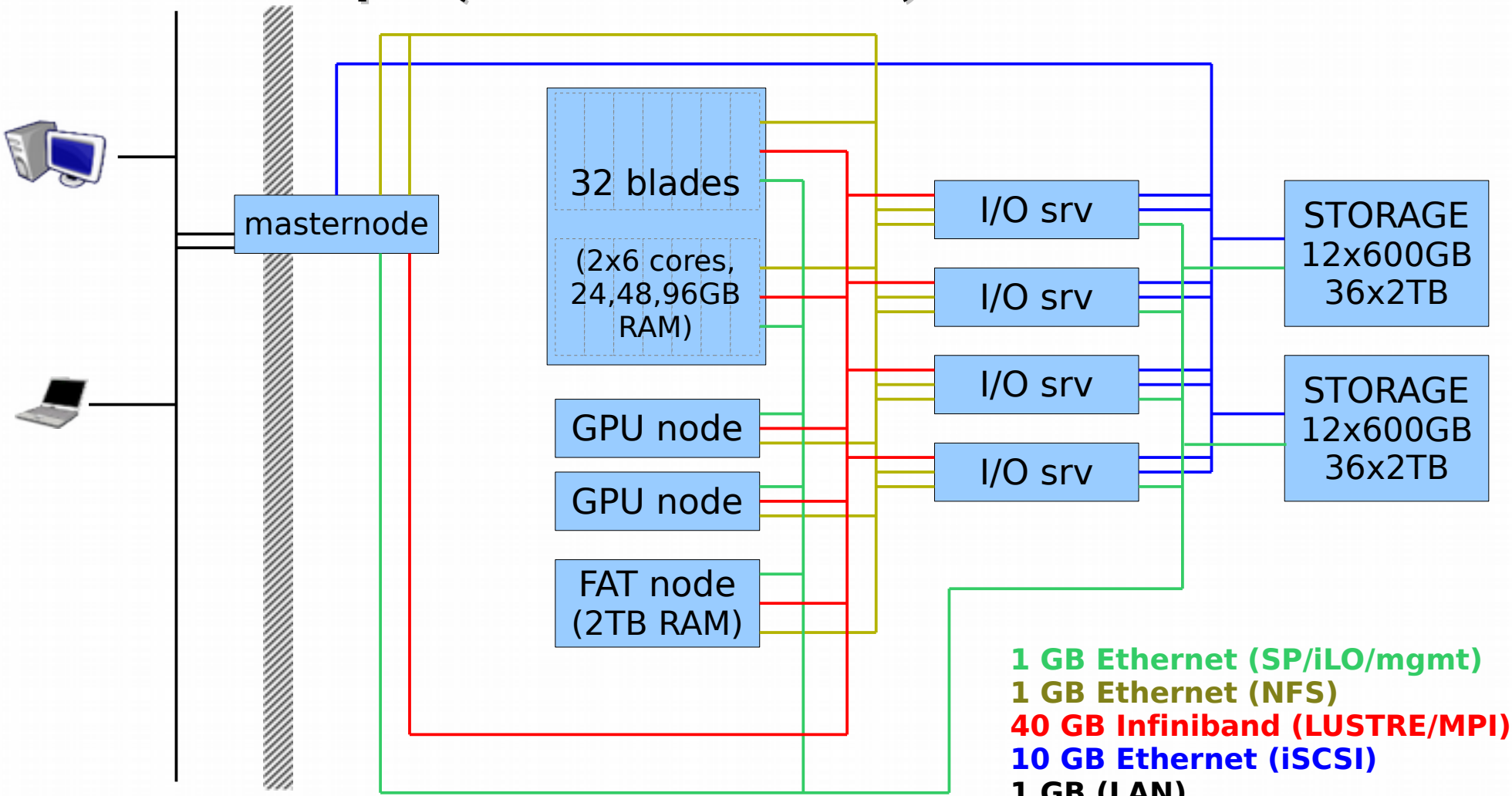
MHPC
Master in High Performance Computing

Network Clusters classification

- HIGH SPEED NETWORK
  - parallel computation
    - low latency /high bandwidth
    - Usual choices: Infiniband...
- I/O NETWORK
  - I/O requests (NFS and/or parallel FS)
    - latency not fundamental/ good bandwidth
    - GIGABIT could be ok /10Gb and/or Infiniband better
- Management network
  - management traffic
    - any standard network

SISSA ICTP

Cluster example (internal network)



MHPC
Master in High Performance Computing

Network speed acceleration in the last 15 years

Ethernet (1979 -)	10 Mbit/sec
Fast Ethernet (1993 -)	100 Mbit/sec
Gigabit Ethernet (1995 -)	1000 Mbit /sec
ATM (1995 -)	155/622/1024 Mbit/sec
Myrinet (1993 -)	1 Gbit/sec
Fibre Channel (1994 -)	1 Gbit/sec
InfiniBand (2001 -)	2 Gbit/sec (1X SDR)
10-Gigabit Ethernet (2001 -)	10 Gbit/sec
InfiniBand (2003 -)	8 Gbit/sec (4X SDR)
InfiniBand (2005 -)	16 Gbit/sec (4X DDR)
	24 Gbit/sec (12X SDR)
InfiniBand (2007 -)	32 Gbit/sec (4X QDR)
40-Gigabit Ethernet (2010 -)	40 Gbit/sec
InfiniBand (2011 -)	54.6 Gbit/sec (4X FDR)
InfiniBand (2012 -)	2 x 54.6 Gbit/sec (4X Dual-FDR)
25-/50-Gigabit Ethernet (2014 -)	25/50 Gbit/sec
100-Gigabit Ethernet (2015 -)	100 Gbit/sec
Omni-Path (2015 -)	100 Gbit/sec
InfiniBand (2015 -)	100 Gbit/sec (4X EDR)
InfiniBand (2016 -)	200 Gbit/sec (4X HDR)

SISSA ICTP

MHPC
Master in High Performance Computing

Latency&bandwidth

NETWORK	Latency	Bandwidth (GB/sec)
Gigabit	70-40	~ 0.125
10G	<5	~1.250
Infiniband 4DDR	~1.5/1.9	~ 3.2
Infiniband FDR	<1.0	~ 5

What is the UNIT OF MEASURE OF LATENCY ?

Microseconds: 3 order of magnitude larger than unit of measure of FP operations

SISSA ICTP

MHPC
Master in High Performance Computing

Communication interfaces within server

- Recent trends in I/O interfaces show that they are nearly matching state of the art network speeds

AMD HyperTransport (HT)	2001 (v1.0), 2004 (v2.0) 2006 (v3.0), 2008 (v3.1)	102.4Gbps (v1.0), 179.2Gbps (v2.0) 332.8Gbps (v3.0), 409.6Gbps (v3.1) (32 lanes)
PCI-Express (PCIe) by Intel	2003 (Gen1), 2007 (Gen2), 2009 (Gen3 standard), 2017 (Gen4 standard)	Gen1: 4X (8Gbps), 8X (16Gbps), 16X (32Gbps) Gen2: 4X (16Gbps), 8X (32Gbps), 16X (64Gbps) Gen3: 4X (~32Gbps), 8X (~64Gbps), 16X (~128Gbps) Gen4: 4X (~64Gbps), 8X (~128Gbps), 16X (~256Gbps)
Intel QuickPath Interconnect (QPI)	2009	153.6-204.8Gbps (20 lanes)

SISSA ICTP

MHPC
Master in High Performance Computing

Network Topology

- How the components are connected.
- Important properties
  - Diameter: maximum distance between any two nodes in the network (hop count, or # of links).
  - Nodal degree: how many links connect to each node.
  - Bisection bandwidth: The smallest bandwidth between half of the nodes to another half of the nodes.
- A good topology: small diameter, small nodal degree, large bisection bandwidth

SISSA ICTP

MHPC
Master in High Performance Computing

Topologies

Common network topologies

- Fat tree
- Mesh
- 3D torus
- CBB (Constant Bi-sectional Bandwidth)
  - type of Fat tree can be oversubscribed 2:1 to 8:1
  - oversubscription can reduce bandwidth but most applications do not fully utilize it anyway

SISSA ICTP

MHPC
Master in High Performance Computing

Bisection bandwidth

- Split N nodes into two groups of N/2 nodes such that the bandwidth between these two groups is minimum: that is the bisection bandwidth

SISSA ICTP

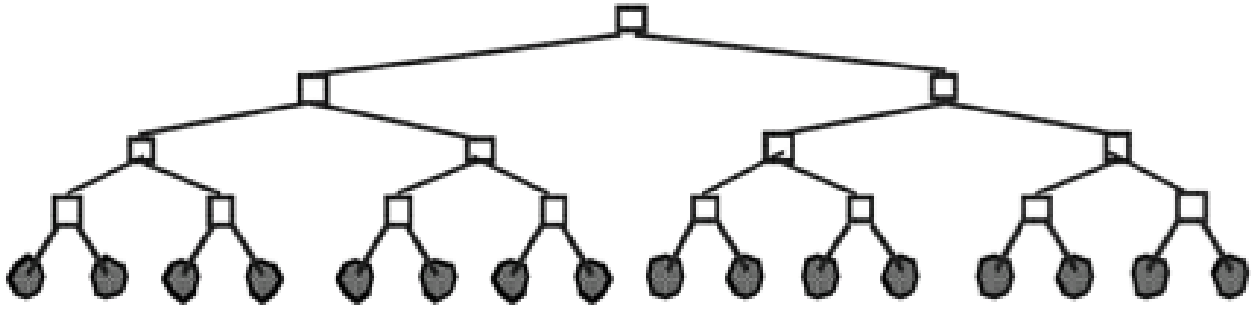
MHPC
Master in High Performance Computing

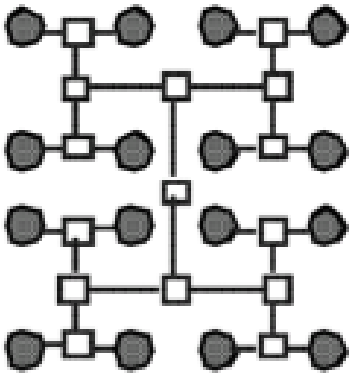
Why is Bisection Bandwidth relevant ?

- if traffic is completely random, the probability of a message going across the two halves is ½
- if all nodes send a message, the bisection bandwidth will have to be N/2
- The concept of bisection bandwidth confirms that some network topology network is not suited for random traffic patterns
- your worst case scenario of HPC workload is to have random traffic patterns..

SISSA ICTP

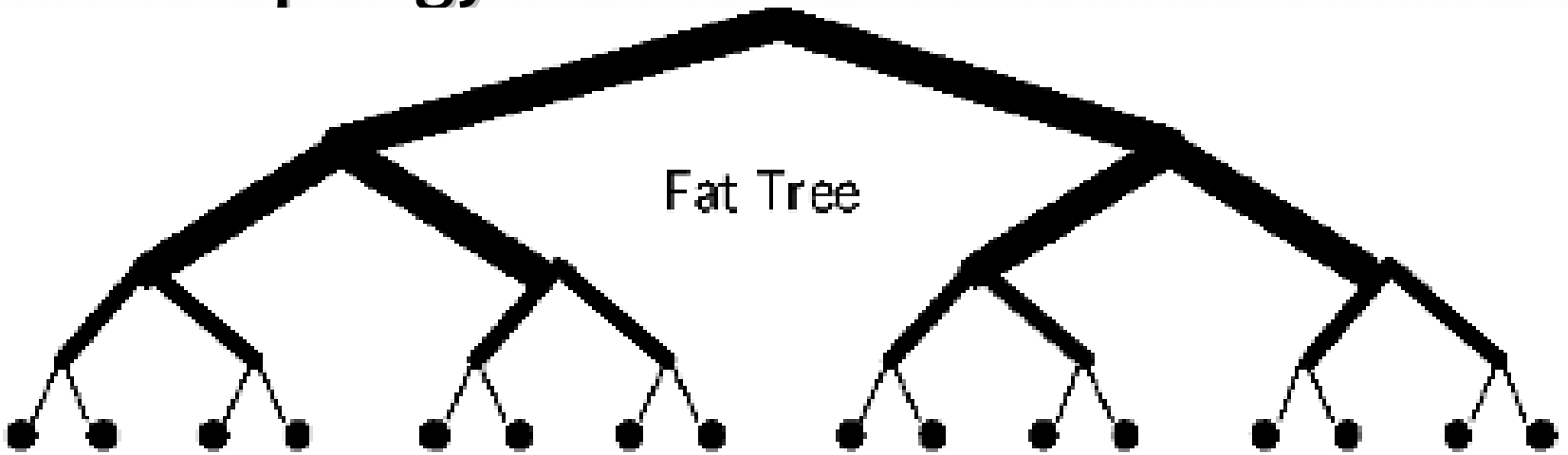
Tree Topology





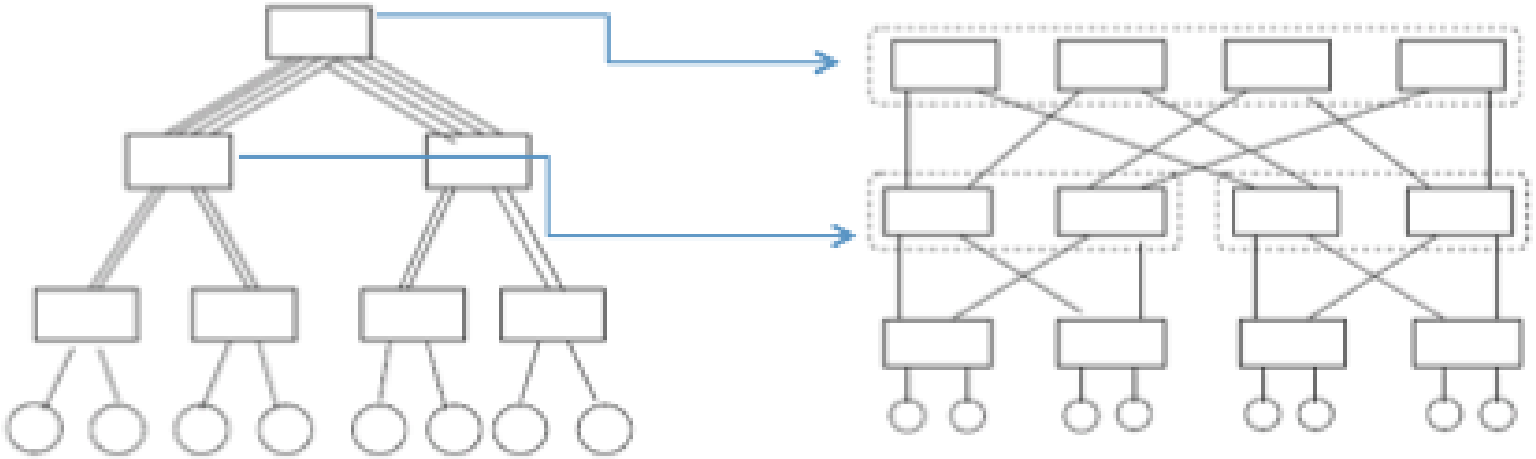
- Fixed degree, log(N) diameter, O(1) bisection bandwidth.
- Routing: up to the common ancestor than go down.

Fat tree topology



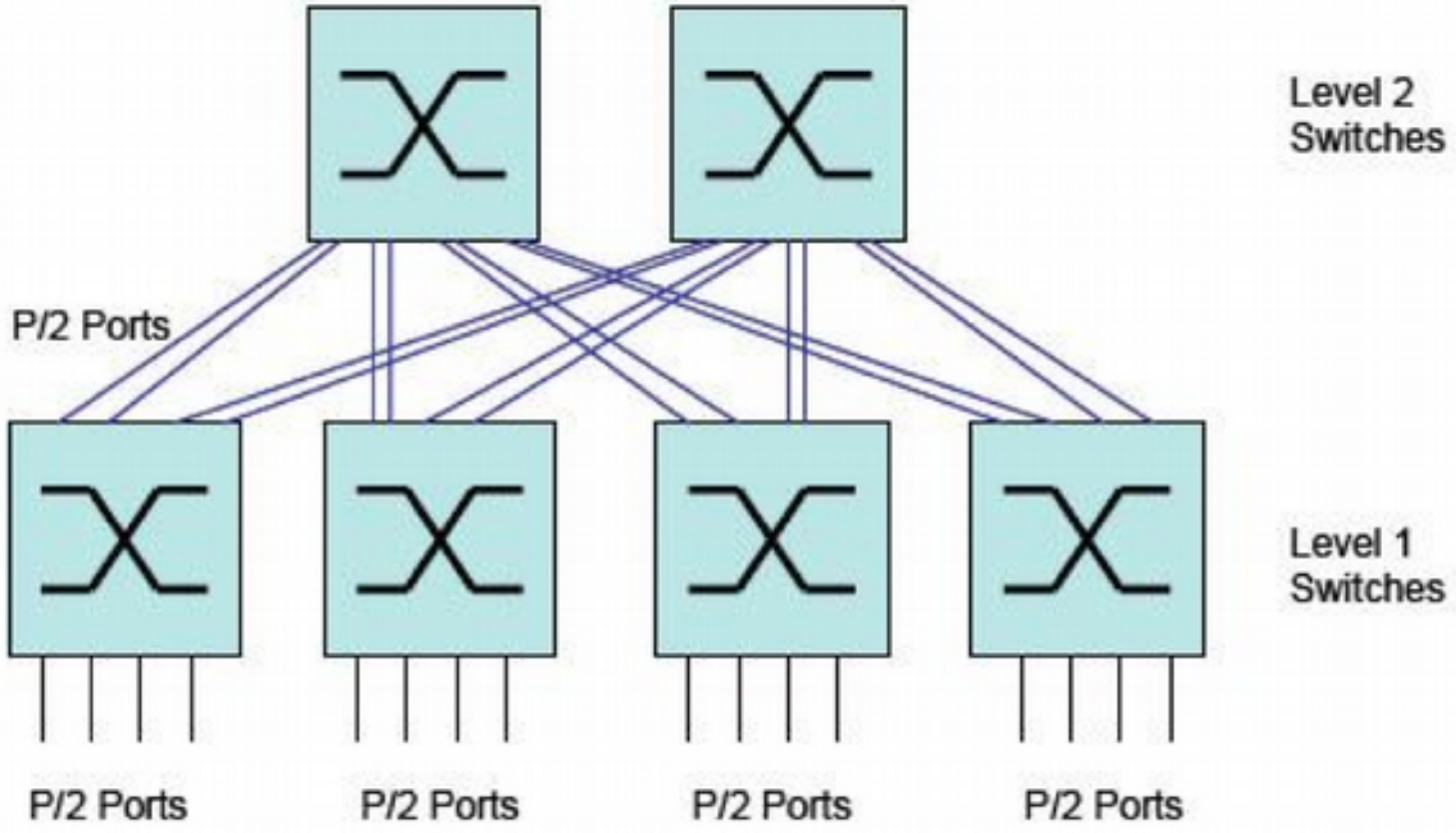
- Fatter links (really more of them) as you go up so bisection BW scales with N
- Not practical. Root is NXN switch

Practical fat tree topology



- Use smaller switches to approximate large switches.
- Most commodity large clusters use this topology.
- Also call constant bisection bandwidth network (CBB)

Two level CBB



MHPC
Master in High Performance Computing

Capabilities of high speed networks

- Intelligent Network Interface Cards
- Support entire protocol processing completely in hardware (hardware protocol offload engines)
- Provide a rich communication interface to applications
  - User-level communication capability
  - Gets rid of intermediate data buffering requirements
- No software signaling between communication layer
  - All layers are implemented on a dedicated hardware unit, and not on a shared host CPU

SISSA ICTP

MHPC
Master in High Performance Computing

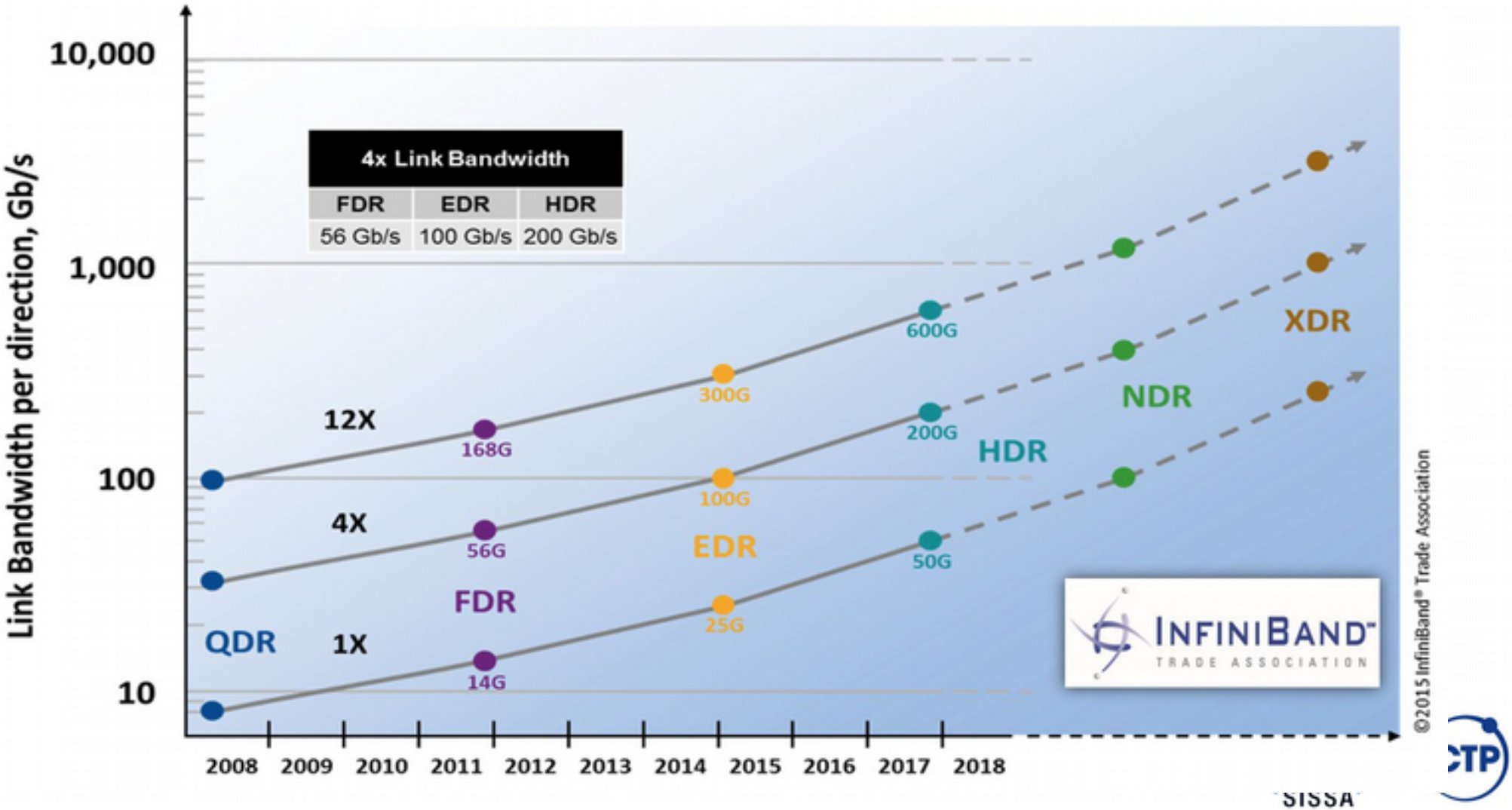
What is InfiniBand?

- **Industry standard** defined by the InfiniBand Trade Association – Originated in 1999
- **InfiniBand specification** defines an input/output **architecture** used to interconnect servers, communications infrastructure equipment, storage and embedded systems
- InfiniBand is a pervasive, **low-latency, high-bandwidth interconnect** which requires low processing overhead and is ideal to carry multiple traffic types (clustering, communications, storage, management) over a single connection.
- InfiniBand is now used in thousands of high-performance compute clusters and beyond that scale from small scale to large scale: **de-facto standard**

SISSA ICTP

MHPC
Master in High Performance Computing

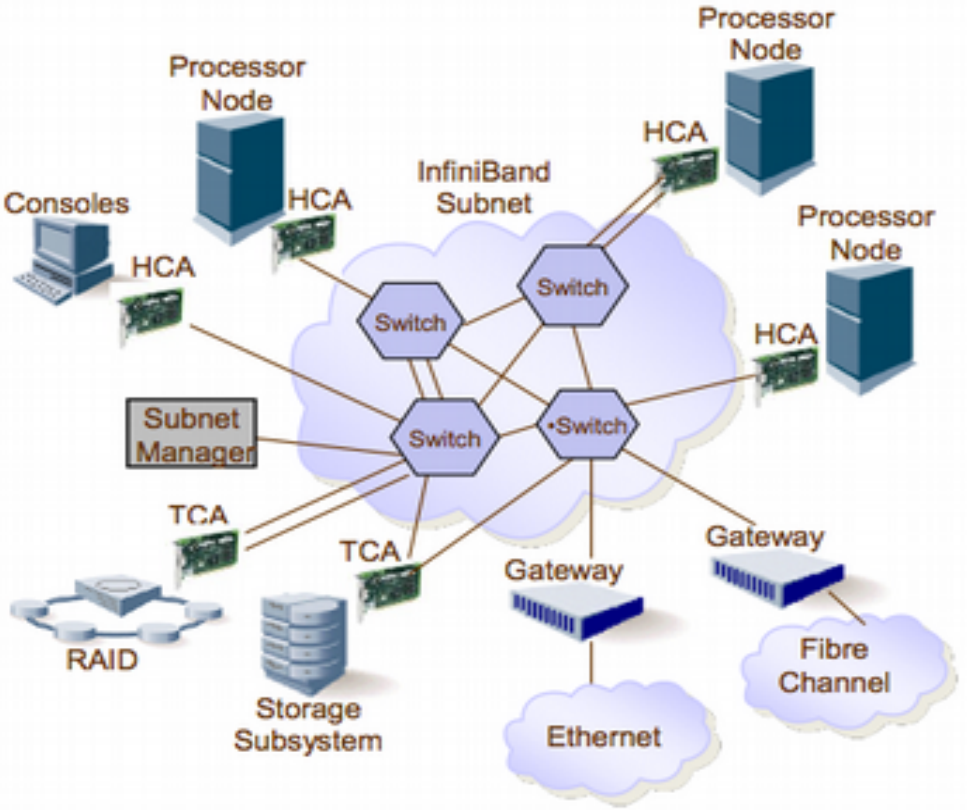
Infiniband roadmap



MHPC
Master in High Performance Computing

InfiniBand Architecture

- Defines System Area Network architecture
  - Comprehensive specification: from physical to applications Processor
- Architecture supports
  - Host Channel Adapters (HCA)
  - Target Channel Adapters (TCA)
  - Switches
  - Routers
- Facilitated HW design for
  - Low latency / high bandwidth
  - Transport offload



InfiniBand speed (physical layer)

- InfiniBand uses serial stream of bits for data transfer
- Linkwidth
  - 1x – One differential pair per Tx/Rx
  - 4x – Four differential pairs per Tx/Rx
  - 12x - Twelve differential pairs per Tx and per Rx
- LinkSpeed
  - Single Data Rate (SDR) - 2.5Gb/s per lane (10Gb/s for 4x)
  - Double Data Rate (DDR) - 5Gb/s per lane (20Gb/s for 4x)
  - Quad Data Rate (QDR) - 10Gb/s per lane (40Gb/s for 4x)
  - Fourteen Data Rate (FDR) - 14Gb/s per lane (56Gb/s for 4x)
  - Enhanced Data rate (EDR) - 25Gb/s per lane (100Gb/s for 4x)
- • Linkrate
  - Multiplication of the link width and link speed
  - Most common shipping today is 4x ports QDR

MHPC
Master in High Performance Computing

Infiniband speed for data transfer..

- For SDR, DDR and QDR, links use 8b/10b encoding:
  - every 10 bits sent carry 8bits of data
- Thus single, double, and quad data rates carry 2, 4, or 8 Gbit/s useful data, respectively.
- For FDR and EDR, links use 64b/66b encoding
  - every 66 bits sent carry 64 bits of data.

SISSA ICTP

MHPC
Master in High Performance Computing

Infiniband performance on local clusters

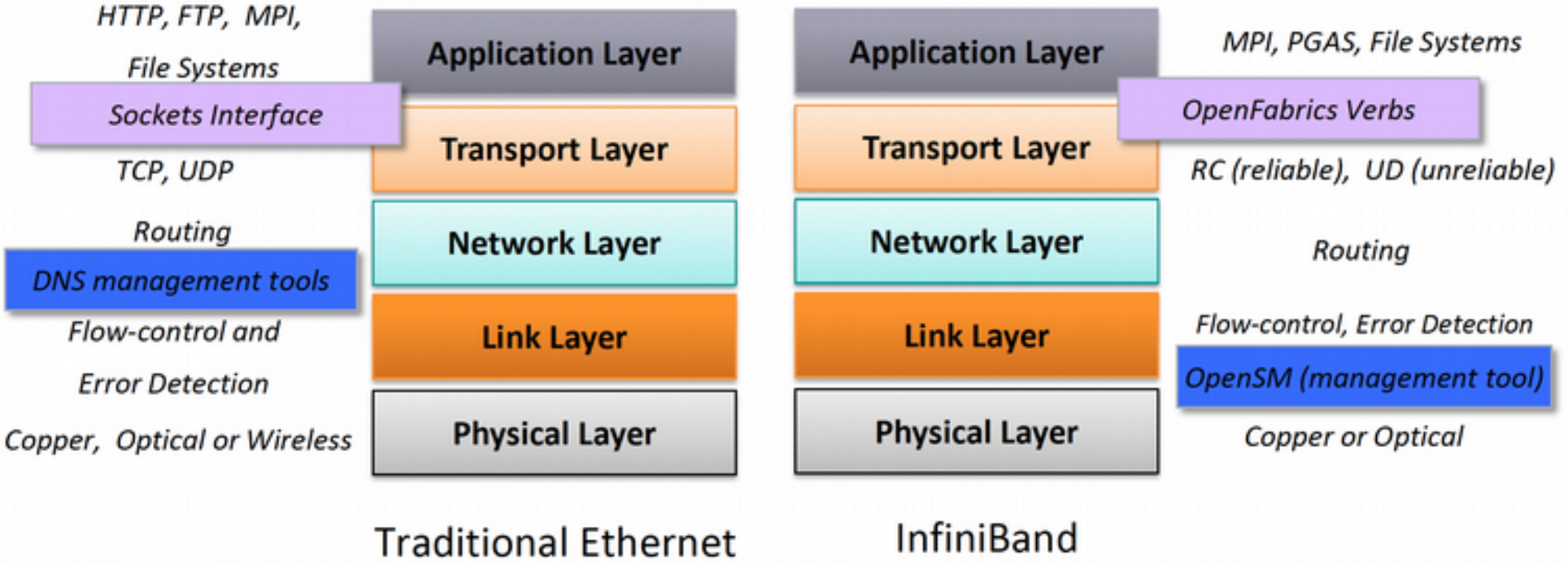
Infiniband type	LInk Speed	Data Speed	Max Bandwidth (at application level)
SDR 4x	10 Gigabit	8 Gigabit	1 GB/sec
DDR 4X	20 Gigabit	16 Gigabit	2GB/sec
QDR 4X	40 Gigabit	32 Gigabit	4GB/sec

We do not take into account the additional physical layer overhead requirements for common characters or software protocol requirements..

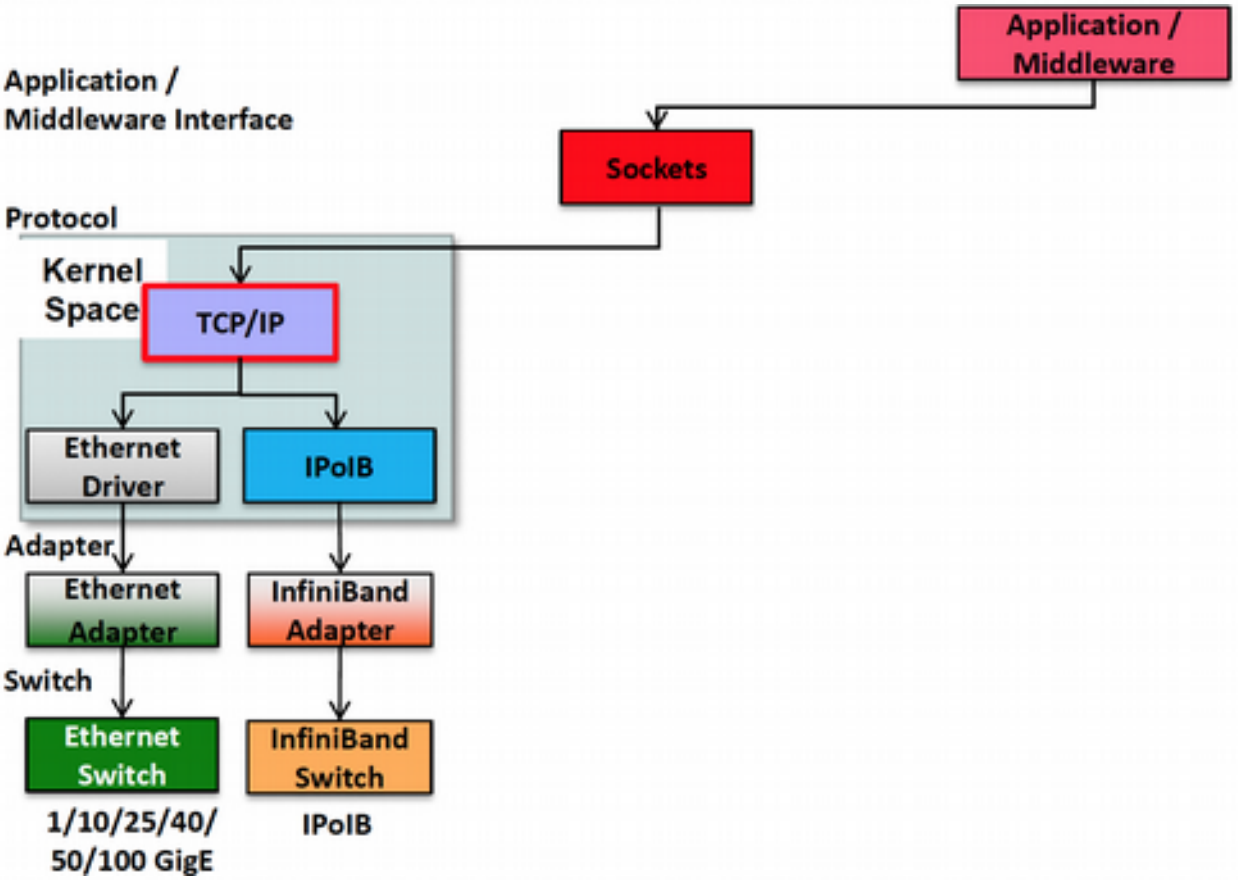
SISSA ICTP

MHPC
Master in High Performance Computing

Infiniband vs Ethernet..

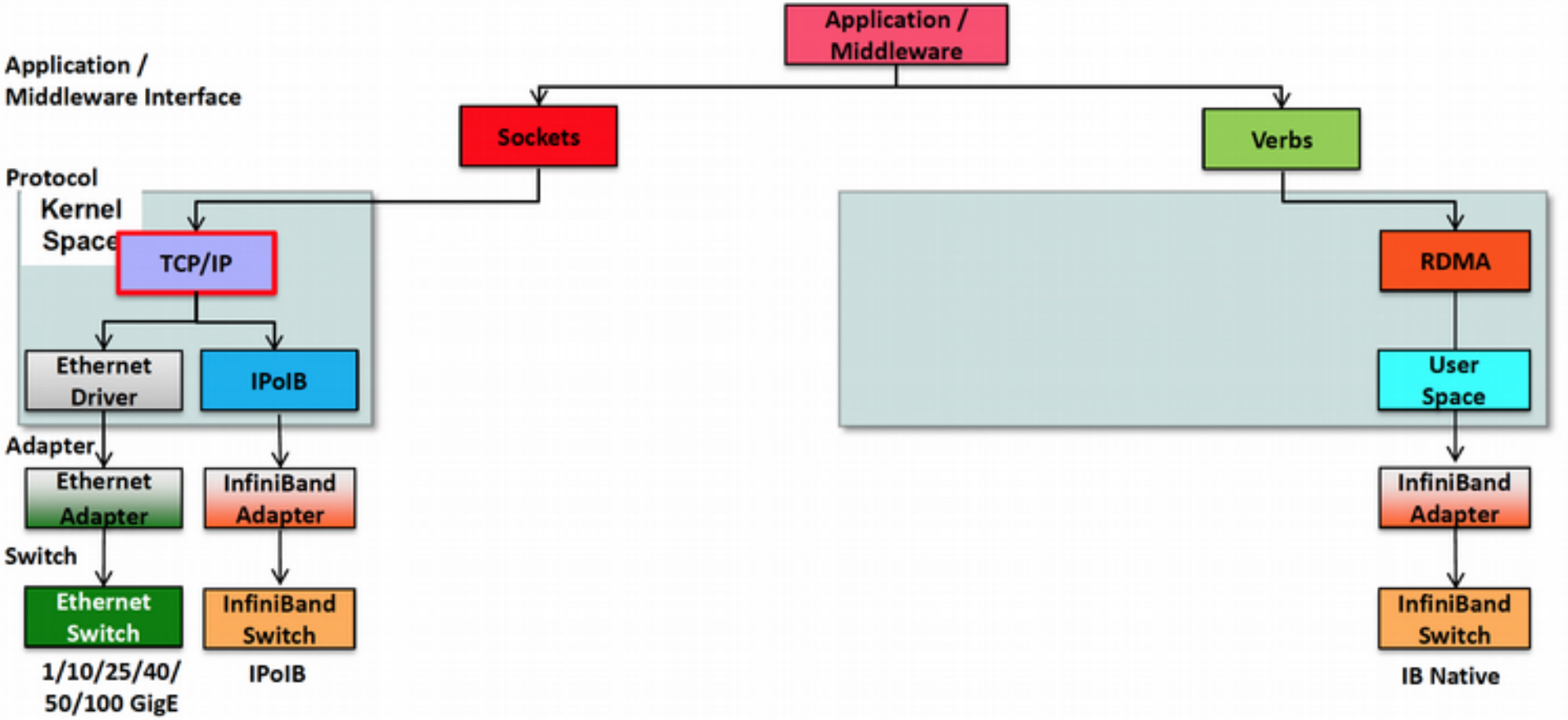


TCP/IP and IPoIB protocol



MHPC
Master in High Performance Computing

TCP/IP and IPoIB protocol vs native infiniband ones



SISSA ICTP

MHPC
Master in High Performance Computing

Vendors

- Two IB vendors: Mellanox and OmniPath
  - Aligned with many server vendors: IBM, HP, Dell
  - And many integrators: Appro, Advanced Clustering, Microway
- Broadly two kinds of adapters
  - Offloading (Mellanox) and Onloading (OmniPath)
- Adapters with different interfaces:
  - Dual port 4X with PCI-X (64 bit/133 MHz),
  - PCIe x8,
  - PCIe 2.0 and HT
- MemFree Adapter
  - No memory on HCA
  - Uses System memory (through PCIe)

SISSA ICTP

MHPC
Master in High Performance Computing

IB software

- Provided by OpenFabric (www.openfabrics.org)
- Open source organization (formerly OpenIB)
- Support for Linux and Windows Design of complete stack with `best of breed' components
- Linux Distribution is now including it (check out carefully which version)
- Users can download the entire stack and run
  - Latest release is OFED 3.18
- OmniPath and Mellanox could have special add-ons

SISSA ICTP

MHPC
Master in High Performance Computing

OFED...

OFED stack includes:

- device drivers
- performance utilities
- diagnostic utilities
- protocols (IPoIB, SDP, SRP,...)
- MPI implementations (OpenMPI, MVAPICH)
- libraries
- subnet manager

SISSA ICTP

MHPC
Master in High Performance Computing

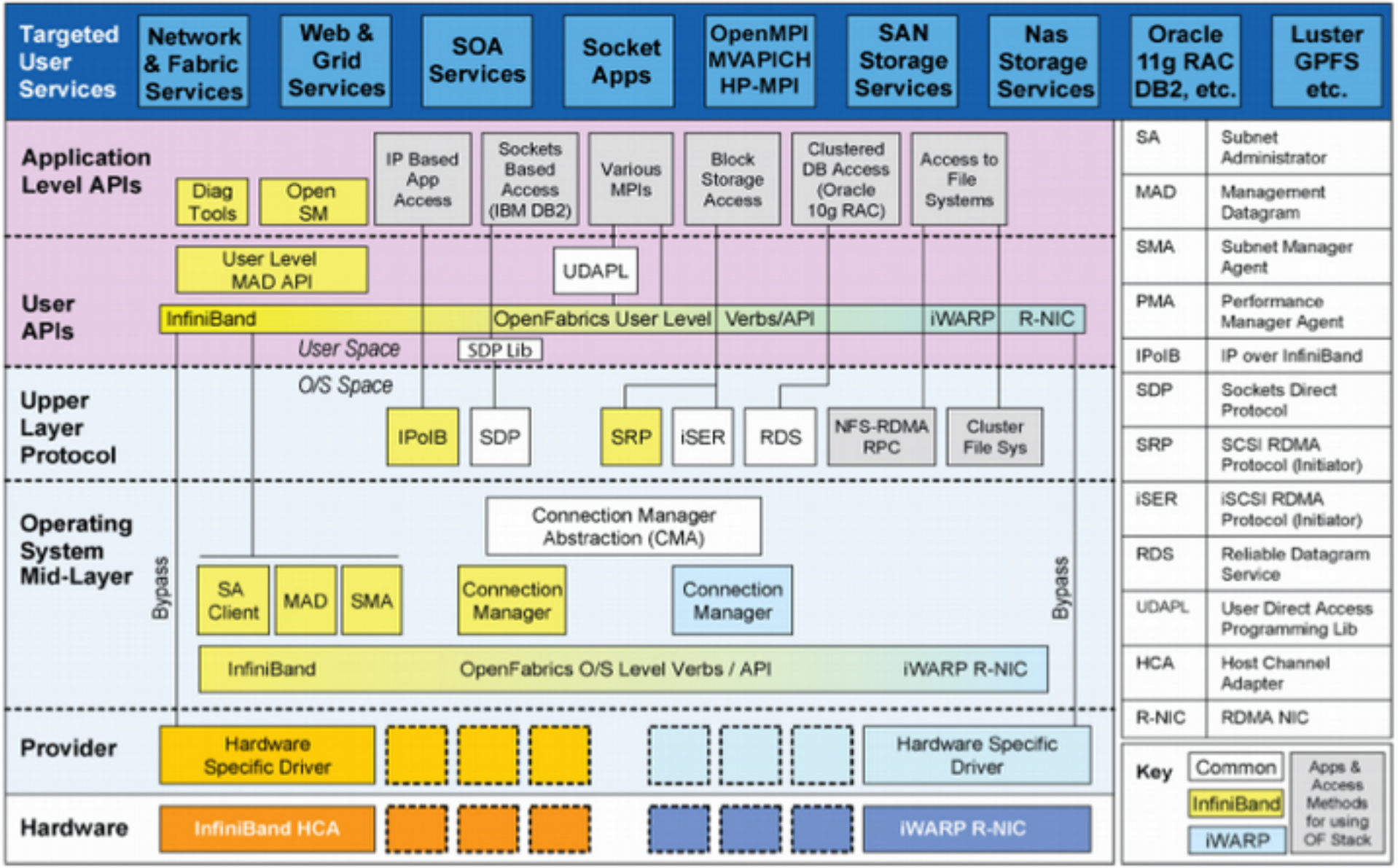
Subnet Manager

- The Subnet Manager (SM) is mandatory for setting up port ID, links and routes
- OpenSM is an Infiniband compliant subnet manger included with OFED
- Ability to run several instance of osm on the cluster in a Master/Slave(s)configuration for redundancy.
- Routing is typically static:The subnet manager tries to balance the routes on the switches.
  - A sweep is done every 10 seconds to look for new ports or ports that are no longer present.
  - Established routes will typically remain in effect if possible.
  - Enhanced routing algorithms:
    - Min-hop, up-down, fat-tree, LASH, DOR, Torus2QOS

SISSA ICTP

MHPC
Master in High Performance Computing

IB software stack..



MHPC
Master in High Performance Computing

Why is (low) latency so important?

According to **Amdahl's law:**

- **a high-performance parallel system tends to be bottlenecked by its slowest sequential process**

- in all but the most embarrassingly parallel supercomputer workloads, **the slowest sequential process is often the latency of message transmission across the network**

SISSA ICTP

MHPC
Master in High Performance Computing

A few final considerations

- In general the compute/communication ratio in a parallel program remains fairly constant.
- So as the computational power increases the network speed must also be increased.
- As multi-core processors proliferate, it is increasingly common to have 8, 10 or even 16 MPI processes sharing the same network device.
- Contention for the interconnect device can have a significant impact on performance.

SISSA ICTP

MHPC
Master in High Performance Computing

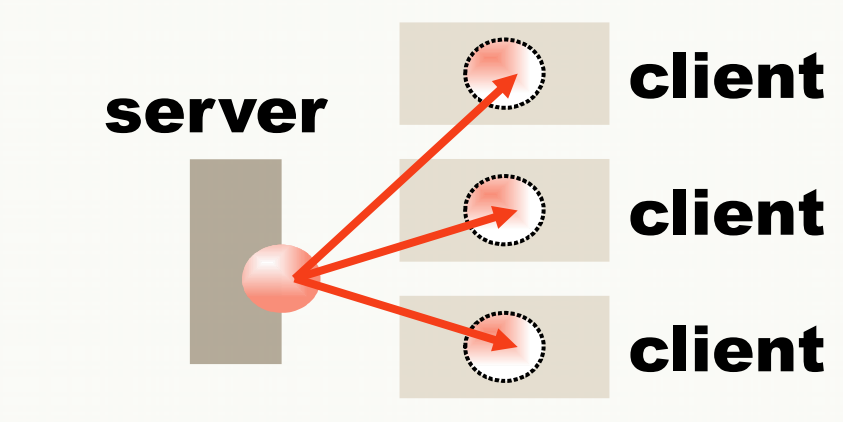
Cluster middleware

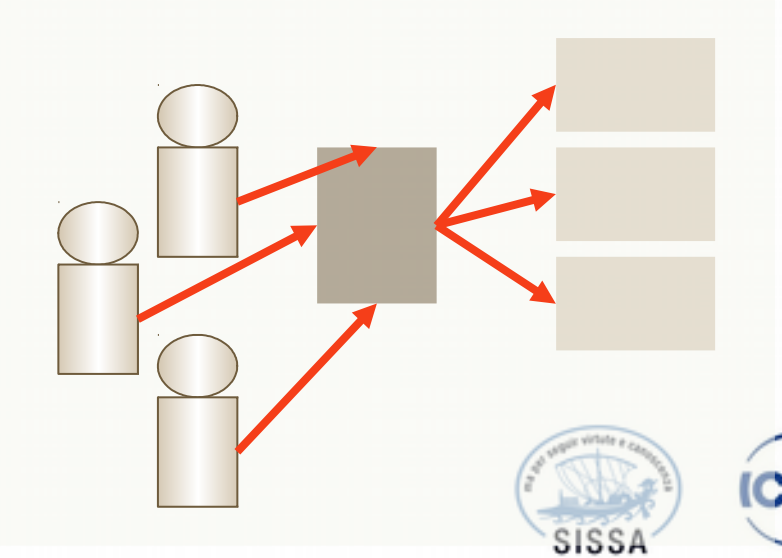
Administration software:

- user accounts
- NTP/NFS/ etc…

Resource management and scheduling software (LRMS)

- Process distribution
- Load balance
- Job scheduling of multiple tasks

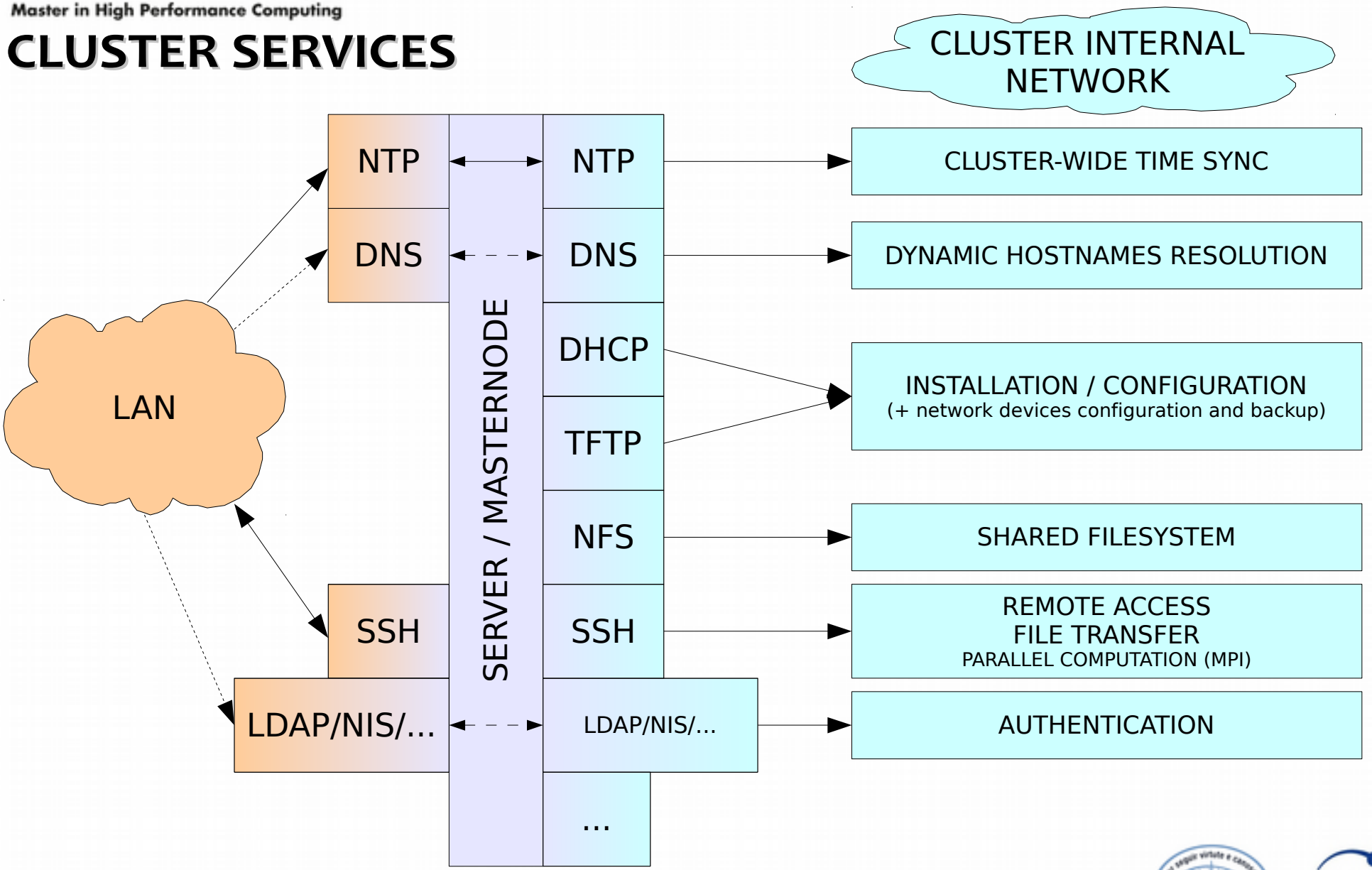




SISSA ICTP

MHPC
Master in High Performance Computing

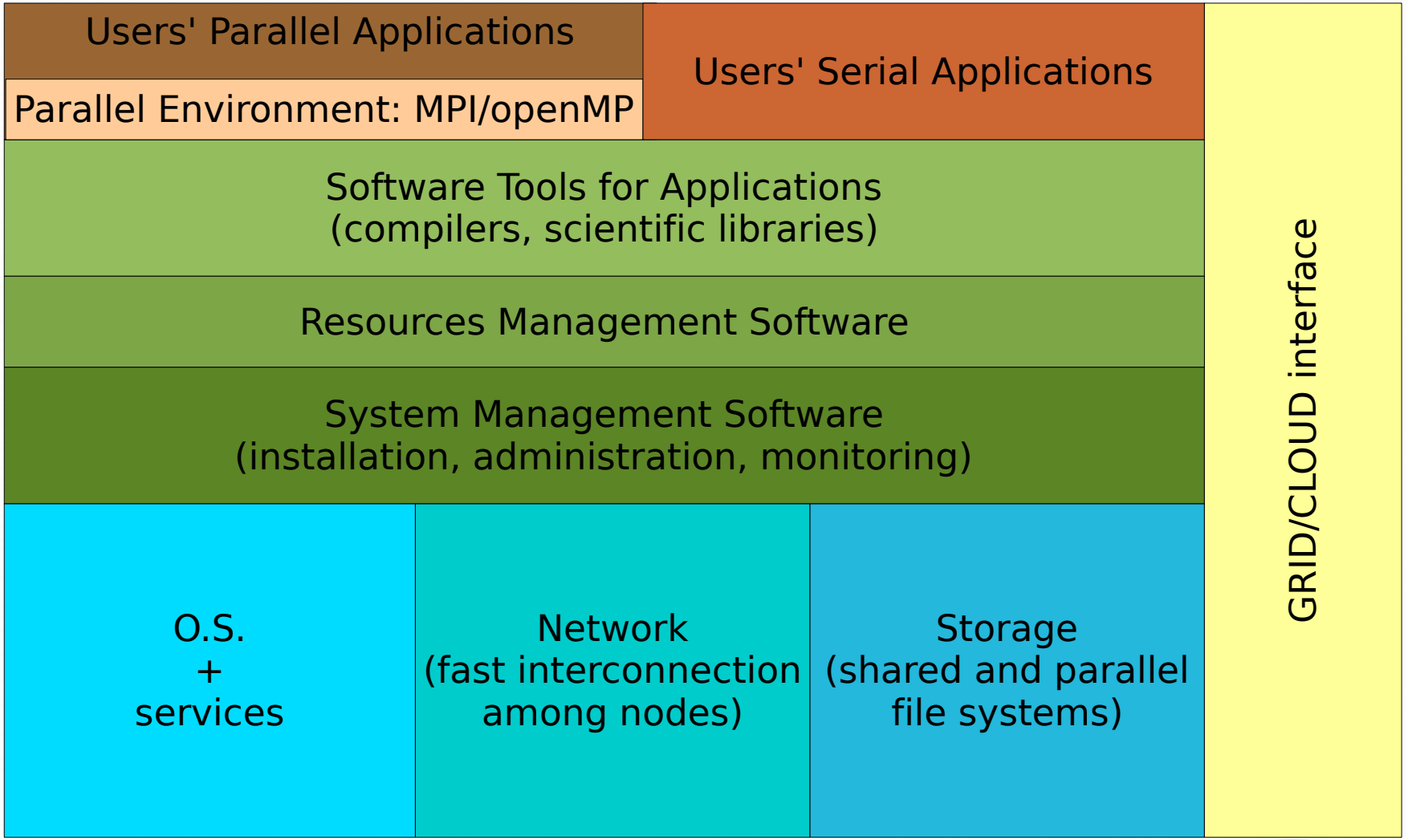
CLUSTER SERVICES



SISSA ICTP

MHPC
Master in High Performance Computing

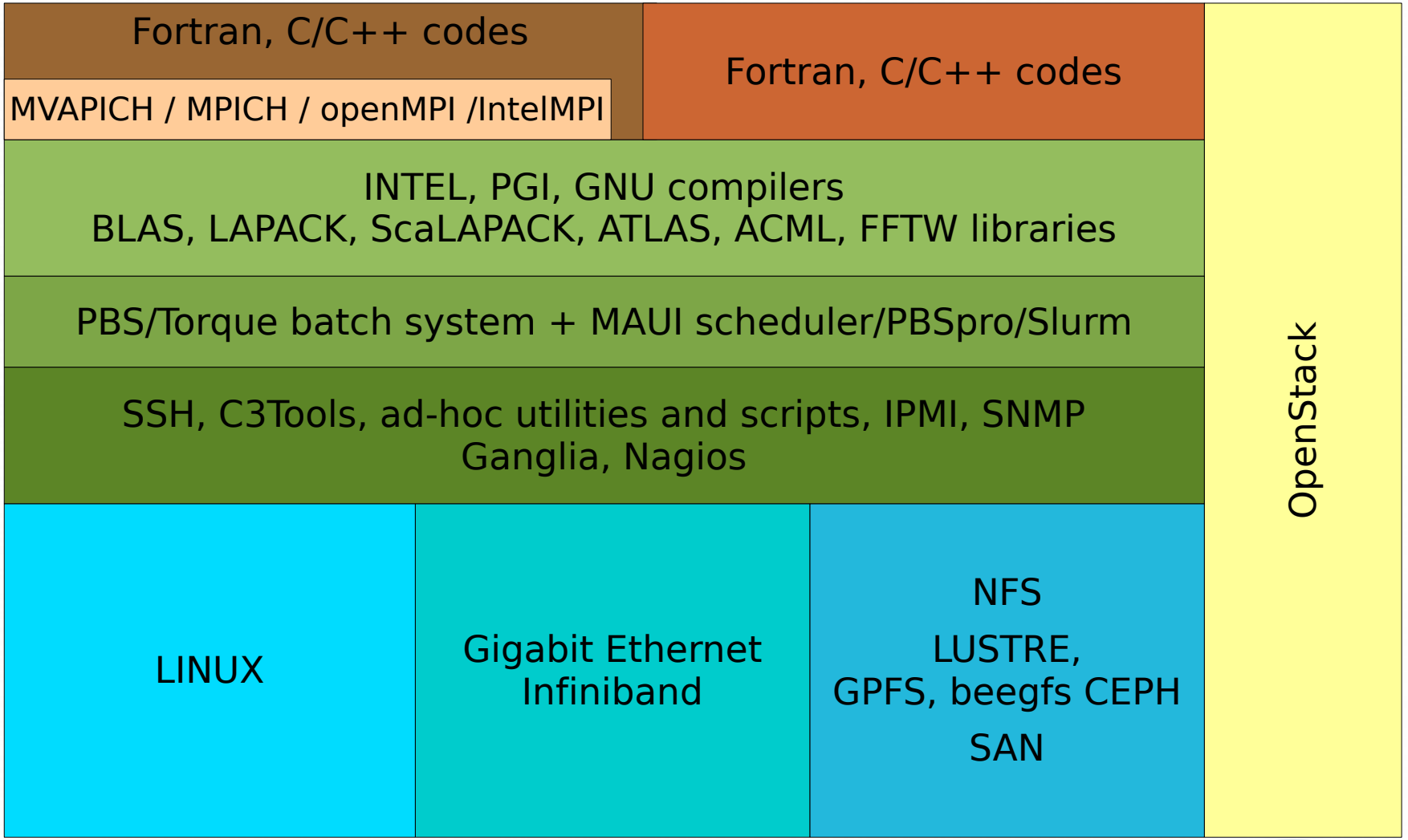
HPC SOFTWARE INFRASTRUCTURE: Overview



SISSA ICTP

MHPC
Master in High Performance Computing

HPC SOFTWARE: what you can find... (our experience)



SISSA ICTP

MHPC
Master in High Performance Computing

Resource Management Problem

We have a pool of users and a pool of resources, then what?

- some software that controls available resources
- some other software that decides which application to execute based on available resources
- some other software devoted to actually execute applications

SISSA ICTP

MHPC
Master in High Performance Computing

What are we speaking about ?

REPLACE THE CAKE WITH HPC RESOURCE

SISSA ICTP

MHPC
Master in High Performance Computing

Some definitions

Batch Scheduler
Software responsible for scheduling the users' jobs on the cluster.

Resources Manager
Software that enable the jobs to connect the nodes and run.

Node (aka Computing Node)
Computer used for its computational power.

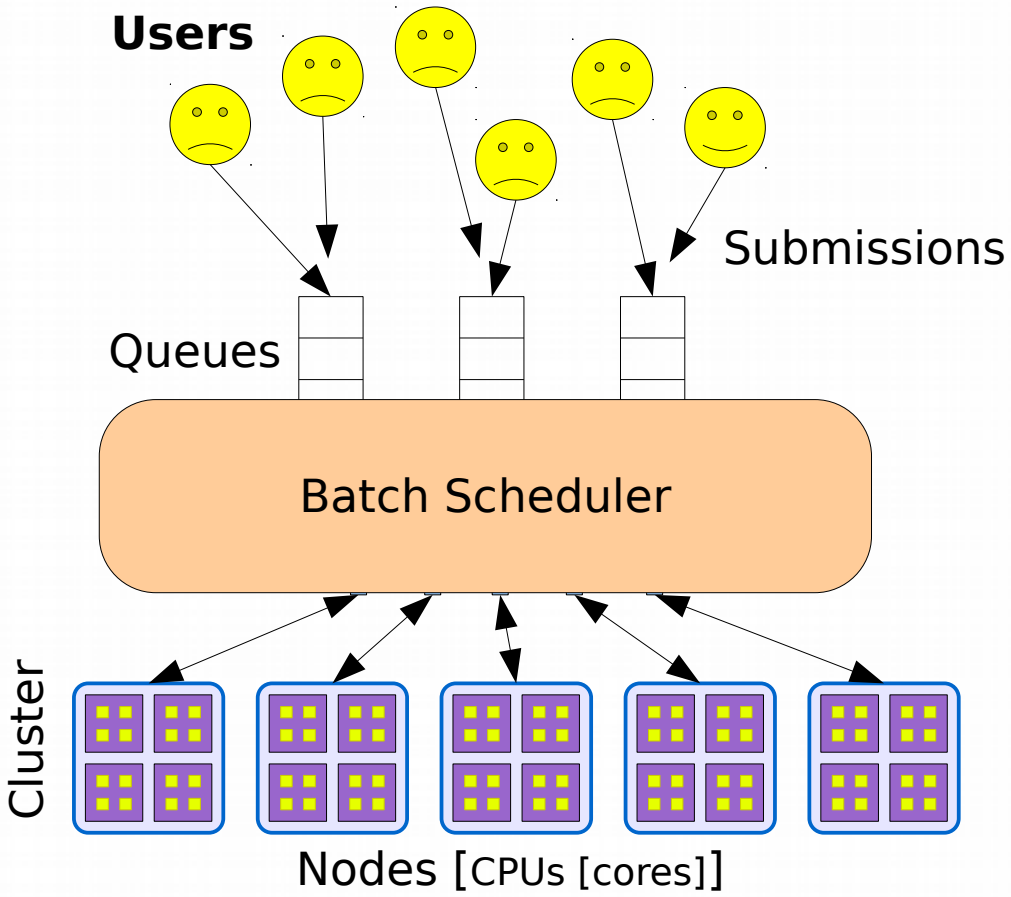
Frontend
It's through this node that the users will submit/launch/manage jobs.

Access Node
A cluster is usually isolated from outside for security purpose, this node is the access gateway.

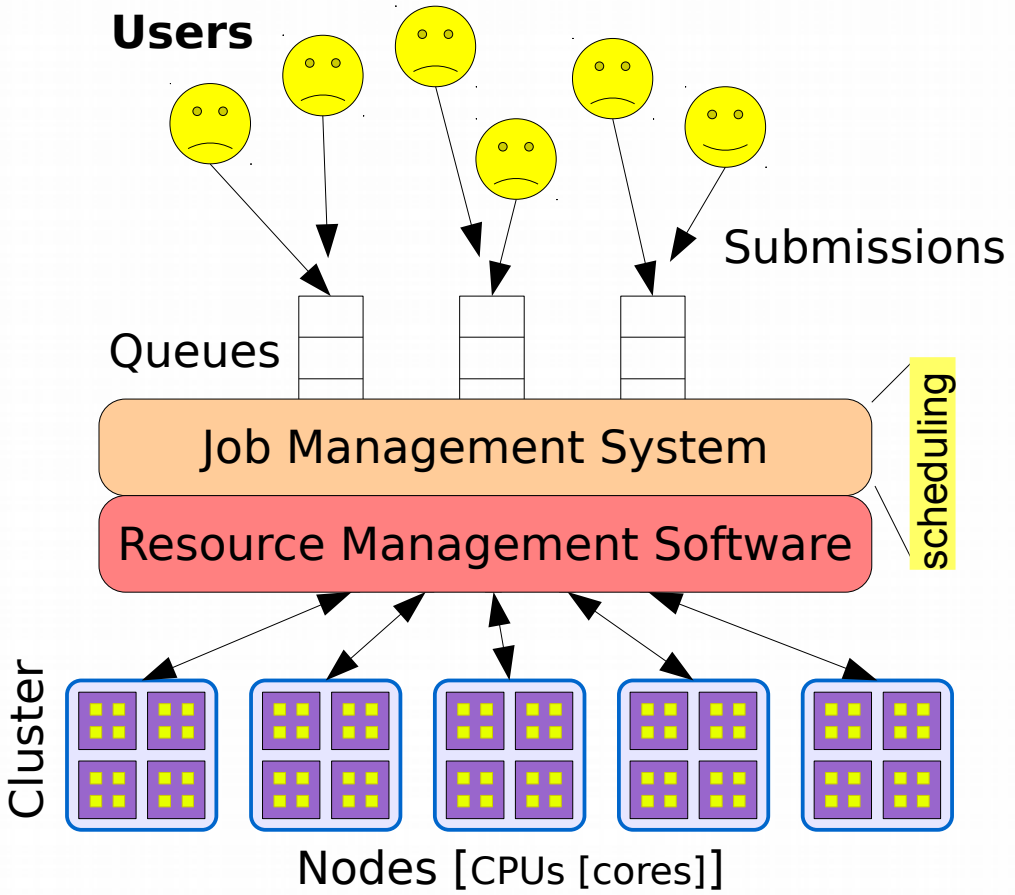
Master Node
Management server, that might as well act as frontend and access node.

SISSA ICTP

Batch Scheduler: a Global Picture 1

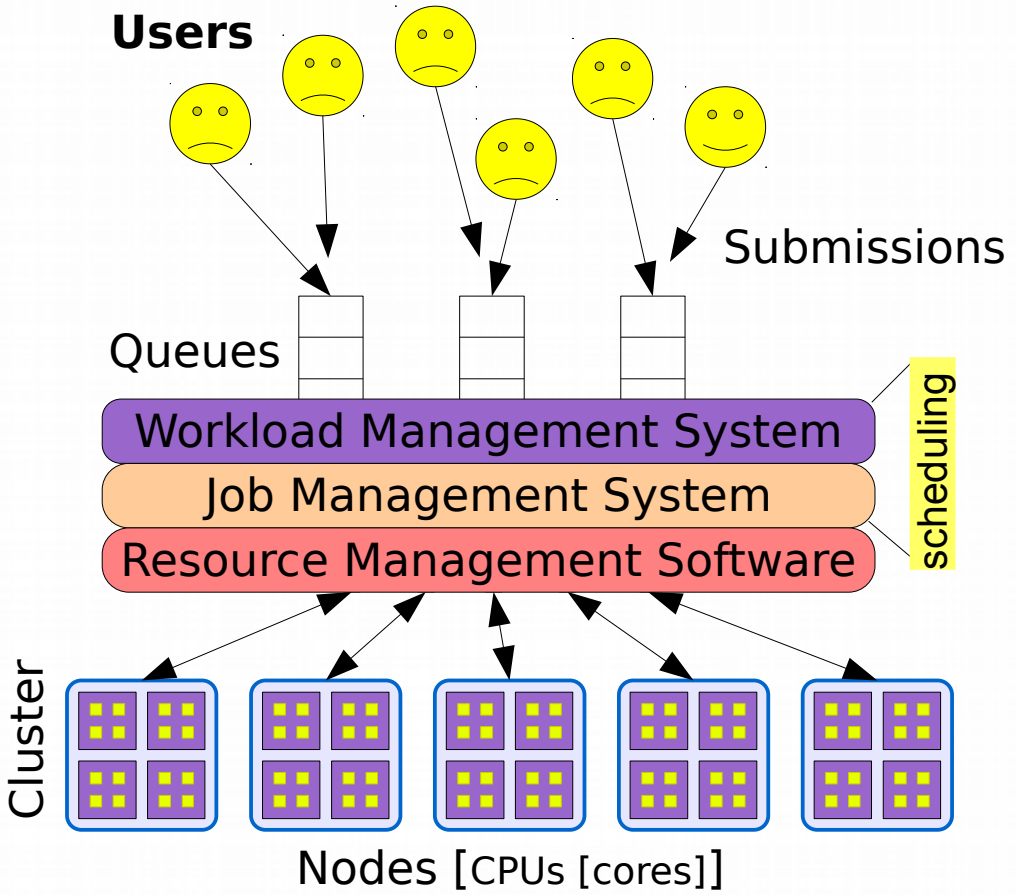


Batch Scheduler: a Global Picture 2



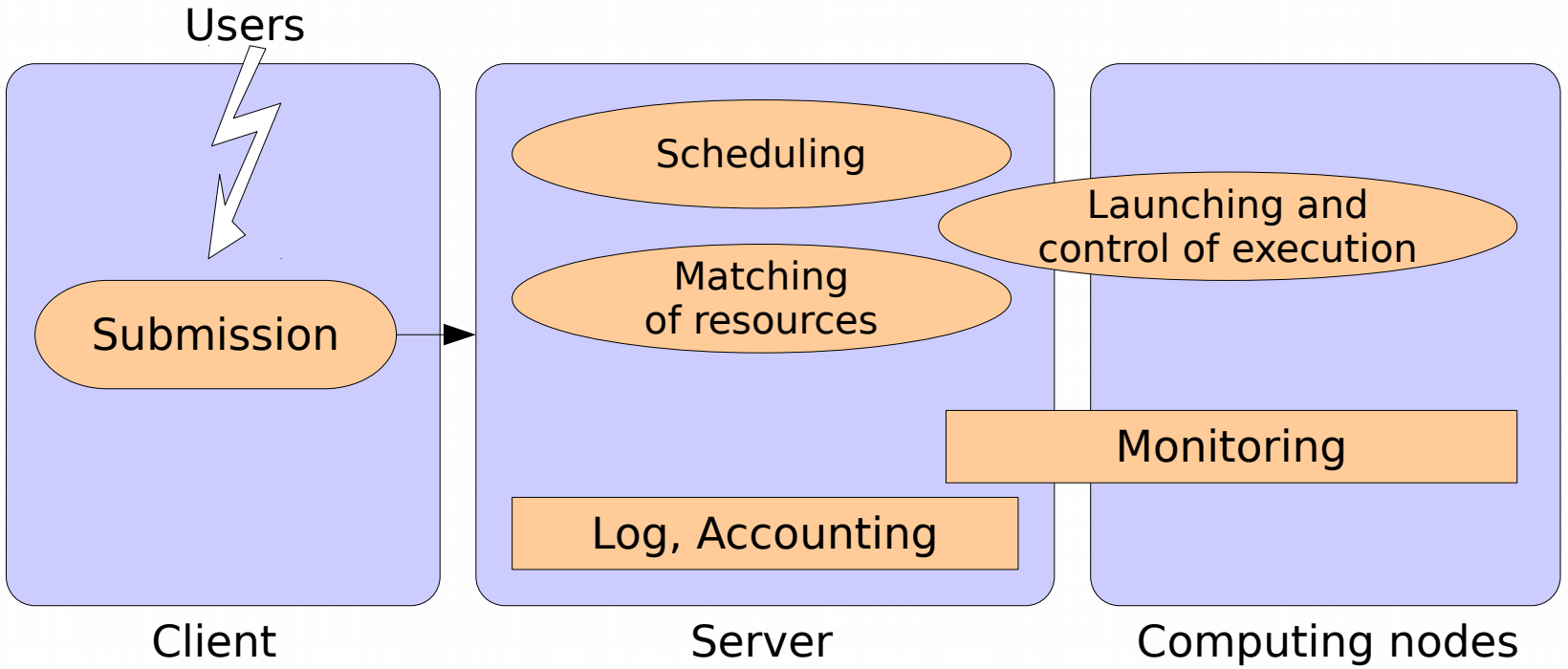
- Resource Management Layer: launching, cleaning, monitoring...
- Job Management Layer: batch/interactive job, Backfilling (EASY or Conservative) Scheduling, Suspend/Resume, Preemption, Dependencies, Resubmission, Advance Reservation...

Batch Scheduler: a Global Picture 3



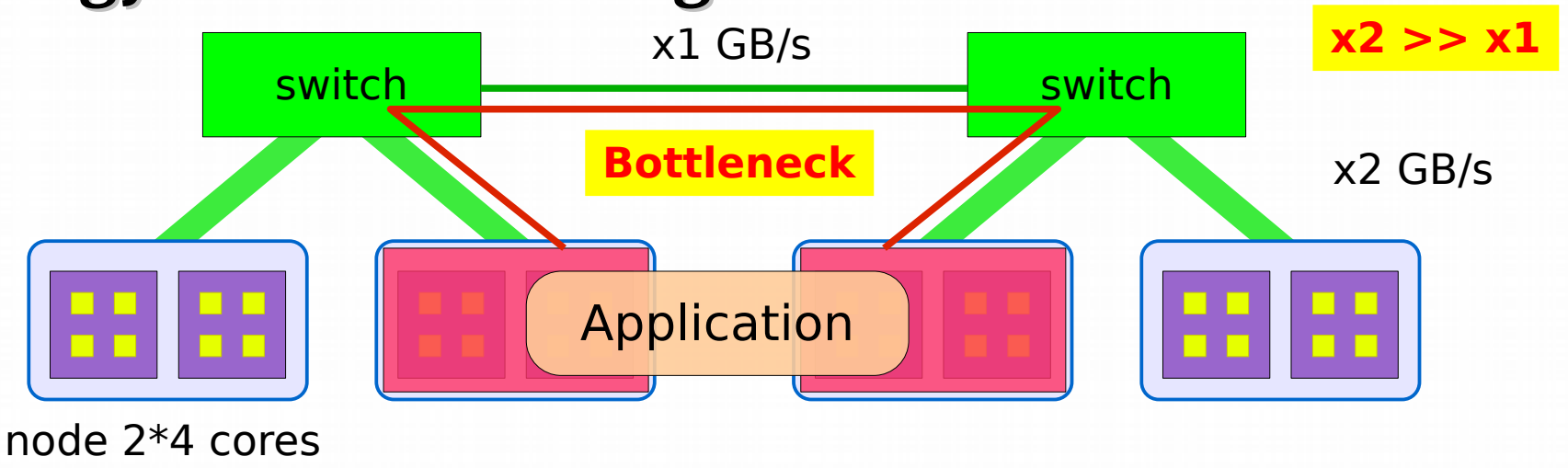
- Workload/Job Management: more complete job scheduling policies
- Fairsharing, Quality of Service (QoS), SLA (Service Level Agreement), Energy Saving, Time Varying Policies (day/night, week-end, holidays ...)
- Dedicated software: MAUI
- There is not true separation into some systems, for instance Slurm and PBSPRO

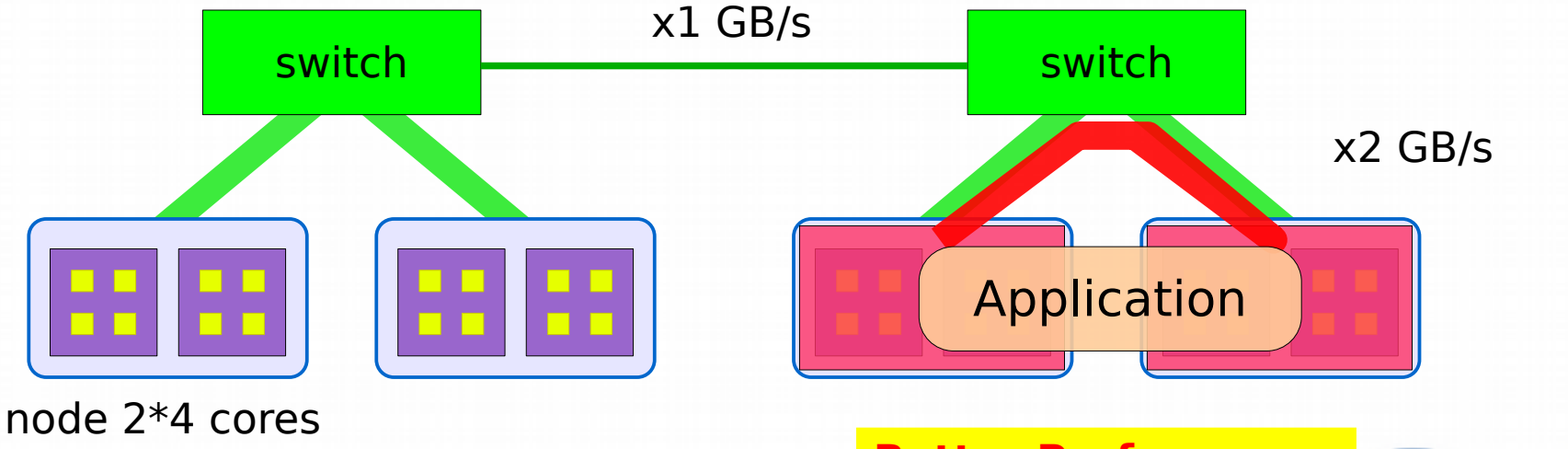
Architecture and main components



- Few components, but the number of jobs and resources states, plus the scheduling policies and a huge number of configurable parameters, lead to a great system complexity.
- It's not so easy to tune and to optimize a Batch Scheduler.

Topology-aware Scheduling





MHPC
Master in High Performance Computing

Some issues/challenges.

- Topology constraint:
  - How can schedule correctly jobs taking into account network layout and node architecture?
- Energy Saving
  - How can I handle nodes not used (off/on/DVSF)?
  - Can I schedule/define policy based on energy cost ?
- Computing resources heterogeneity
  - How can I allocate GPU card /Xeon PHI processors and all the rest ?
- Overall productivity and profitability issues
  - How can keep my system 100% fully loaded and make users happy ?

SISSA ICTP

MHPC
Master in High Performance Computing

Scalability

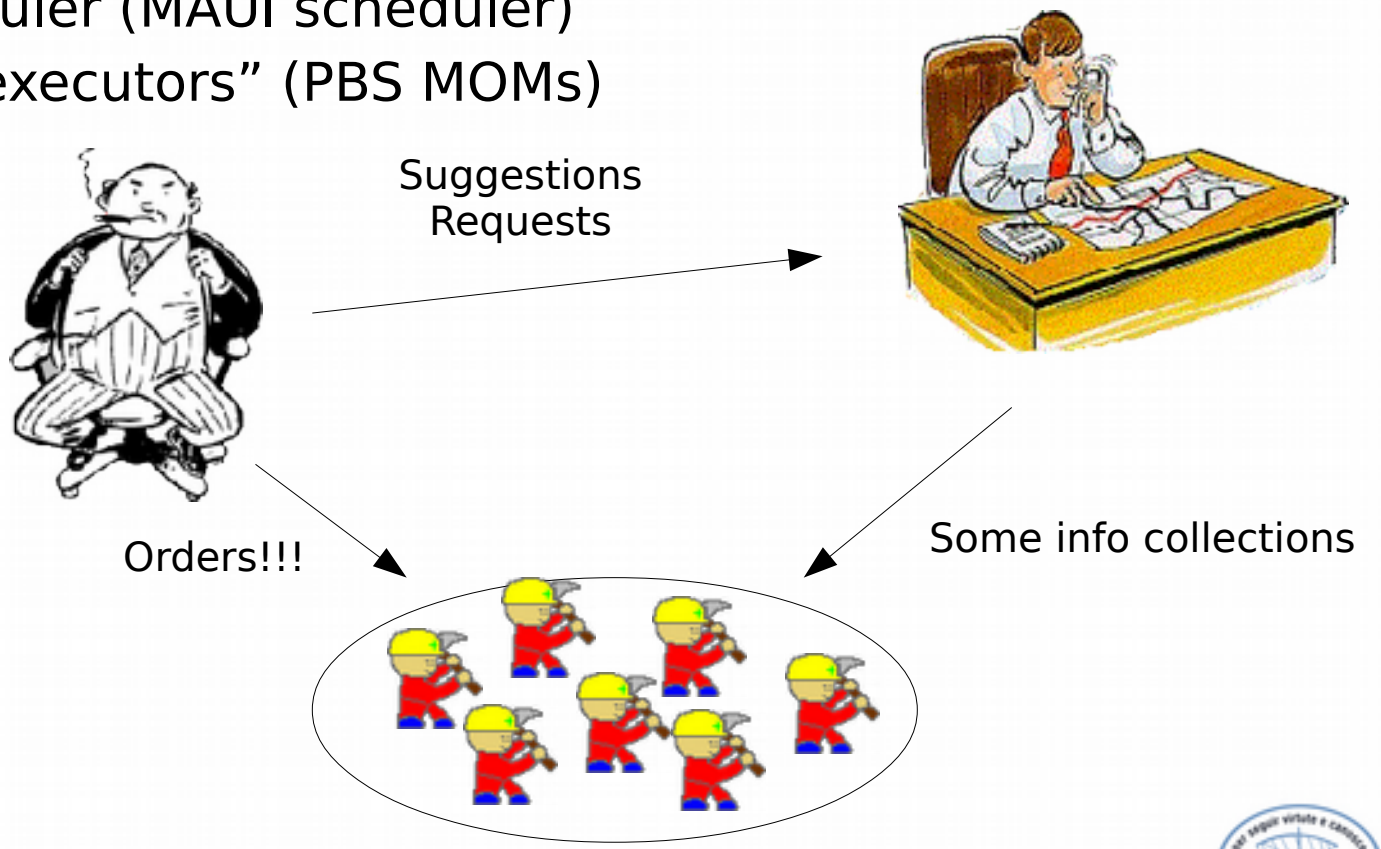
Which granularity for resource representation and manipulation

- core, thread (too fine)? (generally a flat data structure in batch scheduler)
- nodes (most used) (Slurm can manage upto 64K nodes, how many cores ?)
- add some policies for fine tuning (cpuset, cgroup, CPU affinity, Bulk I/O, (next steps bandwidth)...)
- partitions (set of nodes) (sometimes used in large cluster)

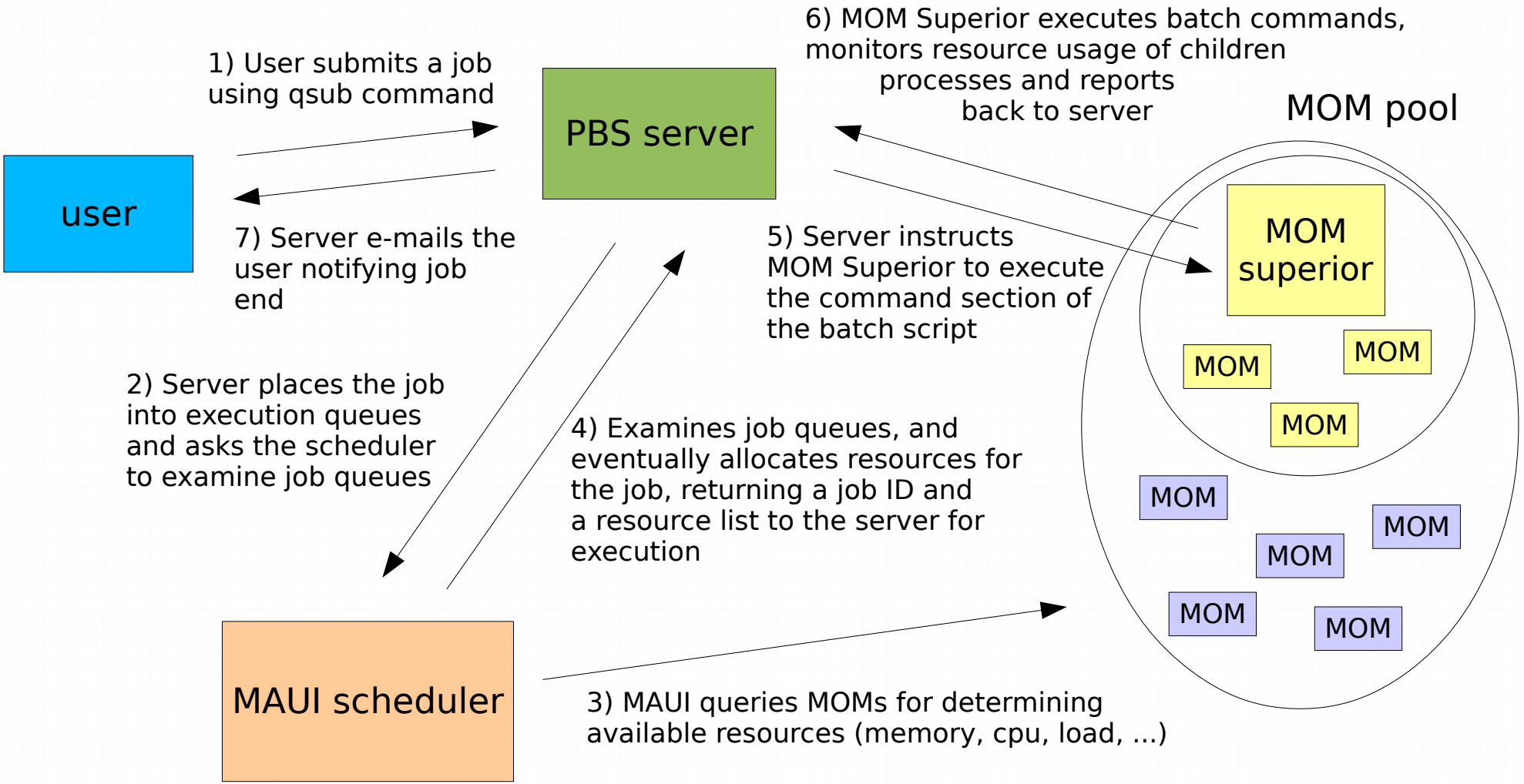
SISSA ICTP

The Queue System: PBS/TORQUE + MAUI

- General Components
  - A resource manager (PBS server)
  - A scheduler (MAUI scheduler)
  - Many “executors” (PBS MOMs)



A typical job session



MHPC
Master in High Performance Computing

Two Basic Scheduler features

The scheduler should have:

- Fair Share mechanism
- Backfill scheduling algorithm

SISSA ICTP

MHPC
Master in High Performance Computing

Backfill 1/2

Backfill is a scheduling optimization which allows a scheduler to make better use of available resources by running jobs out of order.

Consider this example with a 10 CPUs machine:

- Job1 (priority=20 walltime=10 nodes=6)
- Job2 (priority=50 walltime=30 nodes=4)
- Job3 (priority=40 walltime=20 nodes=4)
- Job4 (priority=10 walltime=10 nodes=1)

1) When Maui schedules, it prioritizes the jobs in the queue according to a number of factors and then re-orders the jobs into a 'highest priority first' sorted list.

Sorted list:

- Job2 (priority=50 walltime=30 nodes=4)
- Job3 (priority=40 walltime=20 nodes=4)
- Job1 (priority=20 walltime=10 nodes=6)
- Job4 (priority=10 walltime=10 nodes=1)

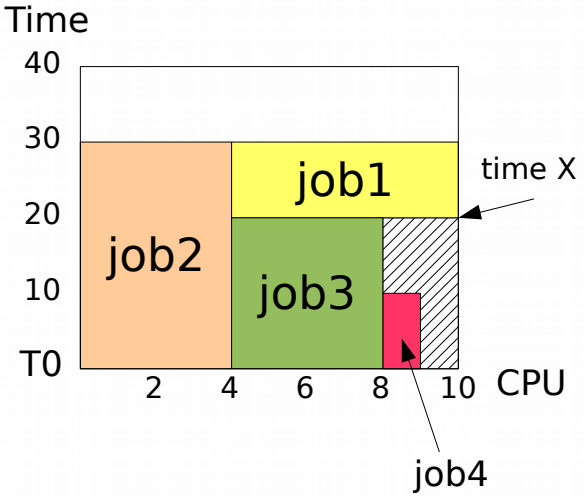
SISSA ICTP

Backfill 2/2

2) It starts the jobs one by one stepping through the priority list until it reaches a job which it cannot start.

3) All jobs and reservations have a start time and a walltime limit, so MAUI can determine:

- the completion time of all jobs in the queue
- the earliest the needed resources will become available for the highest priority job to start (time X)
- which jobs can be started without delaying this job (job4)



Job2 (priority=50 walltime=30 nodes=4)
Job3 (priority=40 walltime=20 nodes=4)
Job1 (priority=20 walltime=10 nodes=6)
Job4 (priority=10 walltime=10 nodes=1)

➔ Enabling backfill allows the scheduler to start other, lower-priority jobs so long as they do not delay the highest priority job, essentially filling in holes in node space.

➔ Backfill offers significant scheduler performance improvement:

- increases system utilization and turnaround time by an even greater amount in a typical large system
- backfill tends to favor smaller and shorter running jobs more than larger and longer running one

MHPC
Master in High Performance Computing

Recap on LRMS

- LRMS is a fundamental tool in the HPC management:
  - User: know it well and you will almost run !
  - Sys. Adm.: know it well and you will keep your system busy..
- Many possible choices
  - Concepts are similar /commands sometime also (to help survive: http://www.schedmd.com/slurmdocs/rosetta.pdf) .
  - Key point is THE scheduler
- Theoretically is almost all possible in resource scheduling with modern LRMS software to accommodate requests from users
- Practically is almost impossible satisfy all your users (and/or communities)

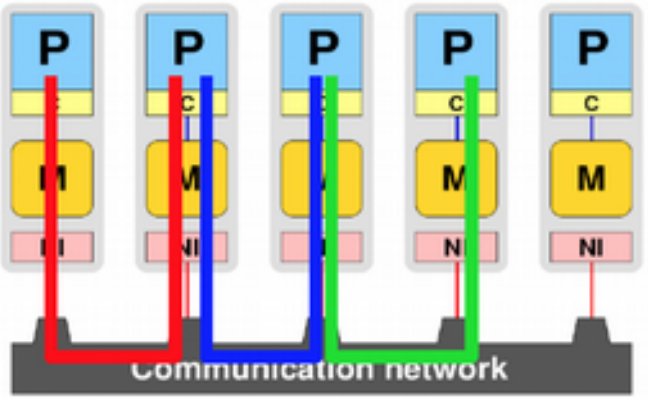
Resource sharing policies is not at all a technical problem !

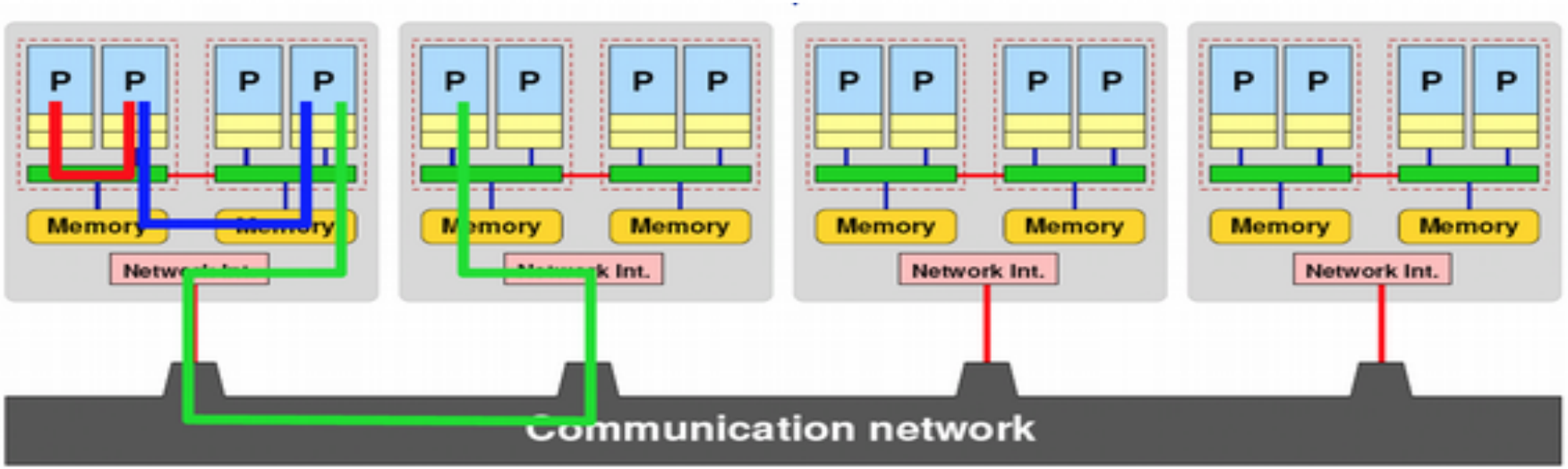
SISSA ICTP

MHPC
Master in High Performance Computing

Parallel programming model : MPI

- Machine structure is invisible to user
  - Very simple programming model
  - MPI "knows what to do"!?
- Performance issues
  - Intranode vs. internode MPI
  - Node/system topology





MHPC
Master in High Performance Computing

Parallel programming

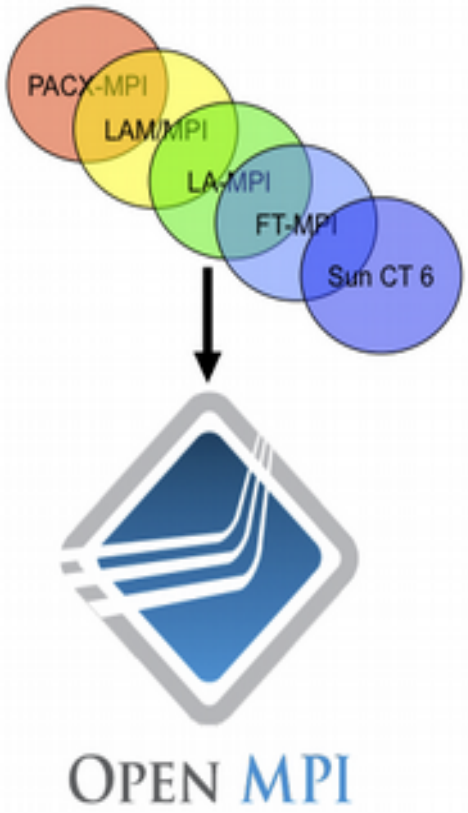
- MPI is a standard with many implementations
- You need a library to link to your MPI-enable parallel code
- Many implementation available:
  - OpenMPI
  - MVAPICH
  - IntelMPI
  - MPICH
  - Etc..

SISSA ICTP

MHPC
Master in High Performance Computing

openMPI

- Evolution of several prior MPI's
- Open source project and community
- Production quality
- Vendor-friendly
- Research- and academic-friendly
- MPI-2.1 compliant



https://www.open-mpi.org/

SISSA ICTP

MHPC
Master in High Performance Computing

OpeMPI and compilers

- OpenMPI works with several compiler suites
- Specify which compilers to use via configure when installing it:
- It is generally best to use single compiler suite
  - Use CC,CXX,FC,F77 to specify compilers
  - Can also specify CFLAGS,CXXFLAGS,FCFLAGS,F77FLAGS

SISSA ICTP

MHPC
Master in High Performance Computing

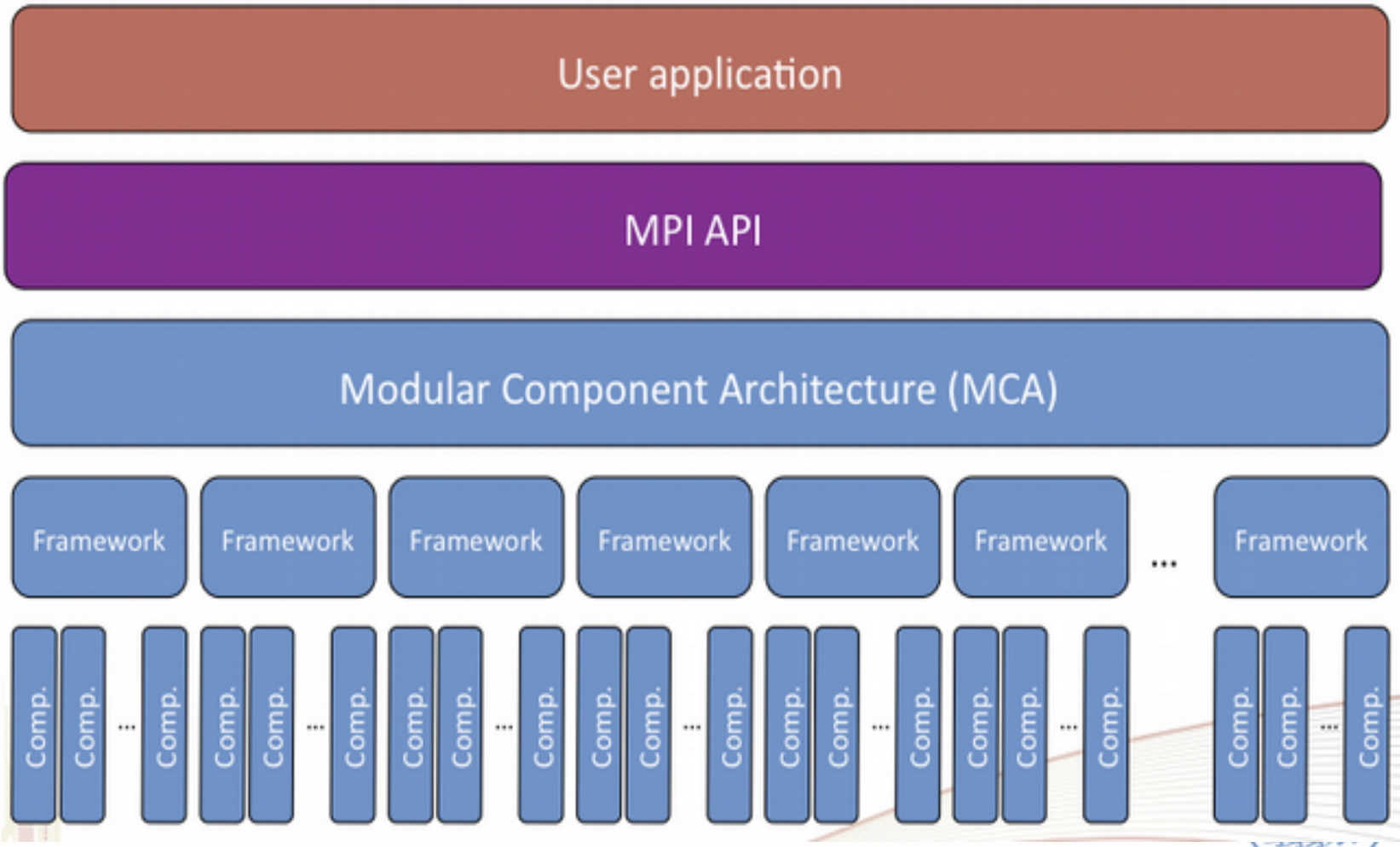
OpenMPI is Based on Plugins

- Lots and lots of plugin types
  - Back-end network
  - Resource manager support
  - Operating system support
- All can be loaded (or not) at runtime
  - Choice of network is a runtime decision
  - User applications no longer linked against network libraries (e.g.,libibverbs)

  Companion concept:run-time parameters

SISSA ICTP

Plugin high level view



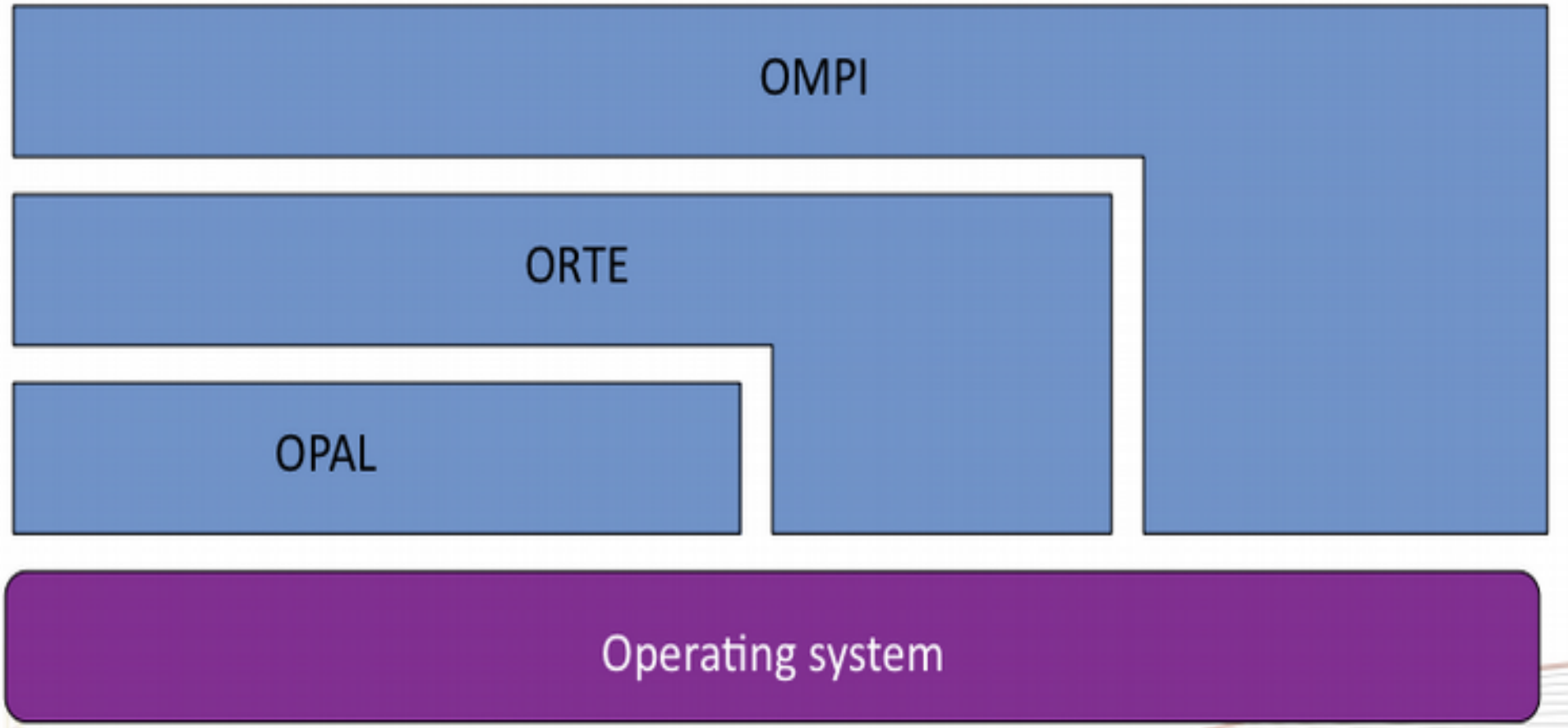
MHPC
Master in High Performance Computing

Three main code sections

- OpenMPI layer (OMPI)
  - Top-level MPI API and supporting logic
- OpenMPI Run-Time Environment(ORTE)
  - Interface to back-end run- time system
- Open Portability Access Layer(OPAL)
  - OS/utility code (lists,reference counting,etc.)
- Dependencies - not layers
  - OMPI➔ORTE➔OPAL

SISSA ICTP

Graphical view

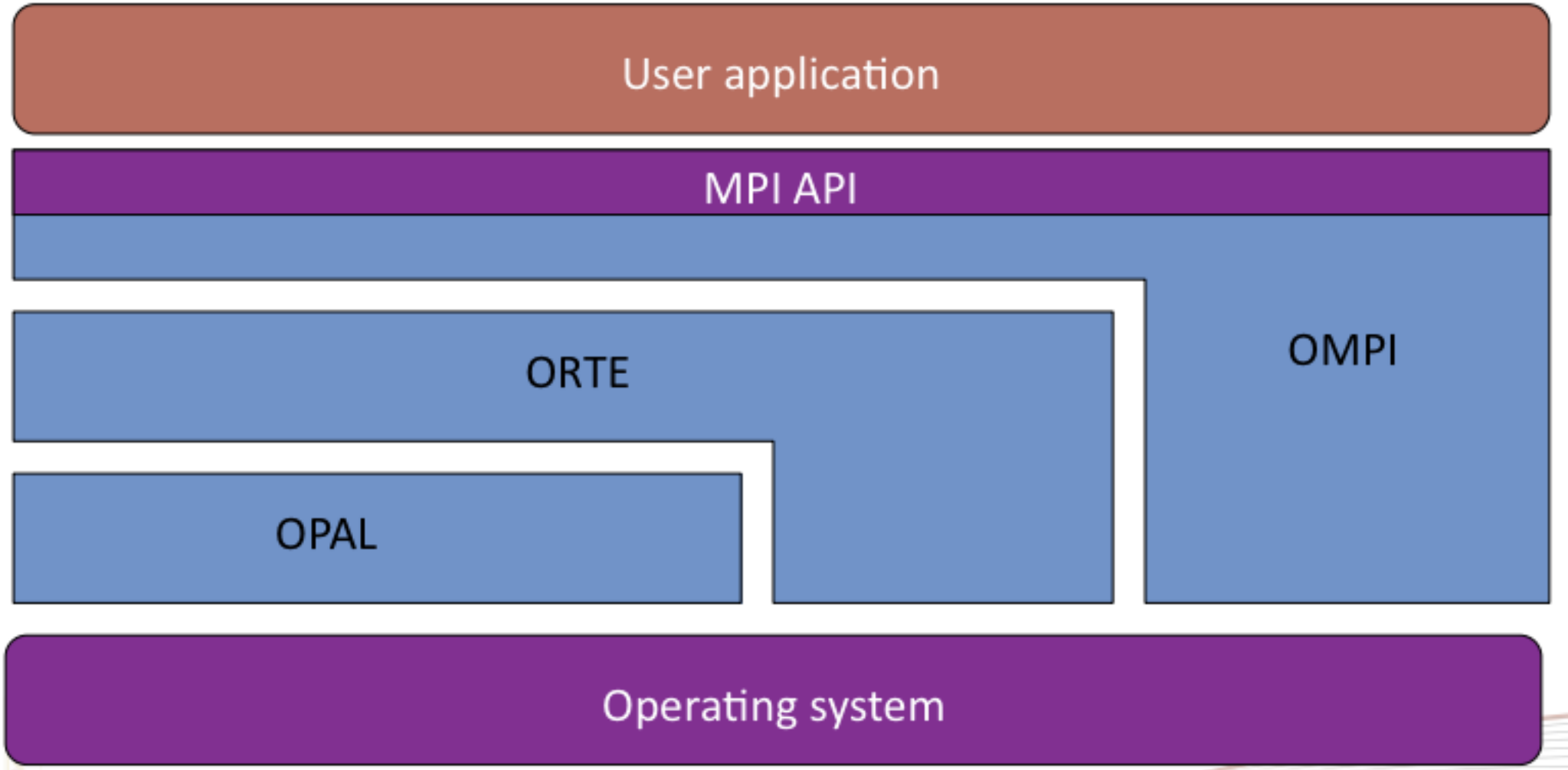


MHPC
Master in High Performance Computing

How to get performance

- "Good"performance defaults
  - No such thing as a"reasonable default" that will work everywhere
  - Balance between scalability and performance
  - Can't always have both!
- Hence:run-time tunable parameters
  - Allows per-system and per-application tuning
  - See examples later

SISSA ICTP


User application
MPI API
OMPI
ORTE
OPAL
Operating system

MHPC
Master in High Performance Computing

MCA parameters

- Run-time tunable values
  - Per layer
  - Per framework
  - Per component(“plugin”)
- Change behaviors of code at run-time
  - Does not require recompiling/relinking
- Simple example : Choose which network to (or not to) use for MPI communications

SISSA ICTP

MHPC
Master in High Performance Computing

Mpirun/mpiexec

- Mpirun and mpiexec
  - Completely identical (inOpenMPI)
- General form:
  - `mpirun -np X your_exe`
- If not using a scheduler,need a hostfile
  - `mpirun [-np X] --hostfile hostfile your_app`
- If using a scheduler,no need for hostfile or-np

SISSA ICTP

MHPC
Master in High Performance Computing

Mpirun useful options

- Assign only a certain number of MPI process on one node
  - `-npernode X`
- Indicates how many cores to bind per process
  - --cpus-per-proc <#perproc>
- Show how processes are bind to cores/sockets etc..
  - --report-bindings

SISSA ICTP

MHPC
Master in High Performance Computing

How many MCA ?

- Too many...
  - OpenMPI has:
    - ~30 frameworks
    - 100+components
- Each component has runtime tunableparameters
- To discover them:
  - `ompi_info --param <type> <plugin>`
  - Shows parameters and current values

SISSA ICTP

MHPC
Master in High Performance Computing

Example: specify BTL

- BTL:Byte Transfer Layer
  - Framework for MPI point-to-pointcommunications
  - Select which network to use for MPI communications

```
mpirun --mca btl tcp,self -np 4 my app
```

- Components
  - tcp:TCP sockets
  - self:Loopback (send-to-self)

SISSA ICTP

MHPC
Master in High Performance Computing

Example:specify openIB BTL

```
mpirun --mca btl openib,self -np 4 my
app
```

- Components
  - openib:OpenFabricsverbs(InfiniBand)
  - self:Loopback(send-to-self)

SISSA ICTP

MHPC
Master in High Performance Computing

What does this do ?

```
mpirun  -np 4 my app
```

- Use all available components
  - tcp,sm,openib,…
- TCP too?
  - Yes--and no
  - TCP will automatically disable itself in the presence of low latency components (e.g.,openib)

SISSA ICTP

MHPC
Master in High Performance Computing

What does this do ?

```
mpirun  -np 4 my app
```

- More specifically:
  - Open each BTL component
  - Query if it wants to be used
  - Keep all that say“yes”

    Rank by bandwidth and latency rank

you can check with `--verbose` option

SISSA ICTP

What does this do ?


```
mpirun  -np 4  --mca btl ^tcp my app
```

- Use all available components except tcp
- More specifically:
  - Open eachBTL component except tcp
  - Query if it wants to be used
  - Keep all that say“yes”

    Rank by bandwidth and latency rank

MHPC
Master in High Performance Computing

Building MPI Applications

- Use“wrapper”compilers
  - Adds inMP Icompiler/linker flags
  - Then invokes underlying compiler
  - Does not actually compile the program
- “--showme”option

```
[cozzini@grid2 ~]$ mpicc --showme
opencc -I/opt/openmpi/1.6.5/open64/4.5.2.1/include -pthread -L/opt/openmpi/1.6.5/ope
n64/4.5.2.1/lib -lmpi -ldl -lm -Wl,--export-dynamic -lrt -lnsl -lutil -lm -ldl
```

SISSA ICTP

MHPC
Master in High Performance Computing

Other languages

- mpiCC(only on case-sensitive filesystems)
  - a.k.a.mpic++
  - a.k.a.mpicxx
- mpif77
- mpif90
- All do similar functions

SISSA ICTP

MHPC
Master in High Performance Computing

MPI freely available benchmarks (2)

- IMB-4.0 (now IMB2017) (INTEL MPI benchmark)
  - MPI protocol ()
  - https://software.intel.com/en-us/articles/intel-mpi-benchmarks
- OSU benchmarks: http://mvapich.cse.ohio-state.edu/benchmarks/

SISSA ICTP

MHPC
Master in High Performance Computing

Suggested activities

- Play with Intel MPI benchmark
- Compile it using openMPI with different compiler
- Submit your job using two or more nodes
- Play with different BTL
- Report/understand difference

SISSA ICTP

MHPC
Master in High Performance Computing

Challenges, Recent Features and Trends

- Scalability (remains the number one issue)
- Topology constraint (hierarchy, NUMA, I/O Bandwidth)
- Energy Saving (node power on/off, DVFS, not so simple)
- Dynamic jobs, massive submission
- Infrastructure diversity (virtual compute node, multi-cluster, GPGPU...)
- Master the increase of (global) complexity
- How to track the global efficiency of the global computing infrastructure (and how to optimize it) ?

SISSA ICTP

MHPC
Master in High Performance Computing

HPL

- From http://icl.cs.utk.edu/hpl/index.html:

  The code solves a uniformly random system of linear equations and reports time and floating-point execution rate using a standard formula for operation count.

  - Number_of_floating_point_operations = $2/3n^3 + 2n^2$ (n=size of the system)

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR03R2L2       86000  1024     2     1             191.06              2.219e+03
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0043644 ...... PASSED
```

SISSA ICTP

MHPC
Master in High Performance Computing

HPL Concerns

- The gap between HPL predictions and real application performance will increase in the future.
- A computer system with the potential to run HPL at an Exaflop is a design that may be very unattractive for real applications.
- Future architectures targeted toward good HPL performance will not be a good match for most applications.
- This leads us to a think about a different metric

SISSA ICTP