



Clusters & Queues

Isabel Campos Plasencia



Grids & e-Science Course at IFCA, from 19th to 23rd of June 2006

Outline

1. Overview about cluster computing
2. Designing a cluster
 1. Processor architecture considerations
 2. Network technologies
 3. Storage
3. Managing the workload: Batch systems

Cluster Definition

Cluster: / klʌstə(r) /n **1** number of things of the same kind growing closely together ◦ a cluster of berries, flowers, curls; **2** number of people, animals or things grouped closely together ◦ a cluster of houses, spectators, bees, islands, diamonds, stars ◦ a consonant cluster, eg *str* in strong.

Oxford Advanced Learner's Dictionary of Current English

our clusters are made of computers

Computer Cluster

- Beowulf Cluster
 - 1994 (16 PCs) NASA
- Distributed memory parallel computer
- Standard Hardware (off-the-shell)
 - Optimal ratio performance/price
- Open Source OS and development Software



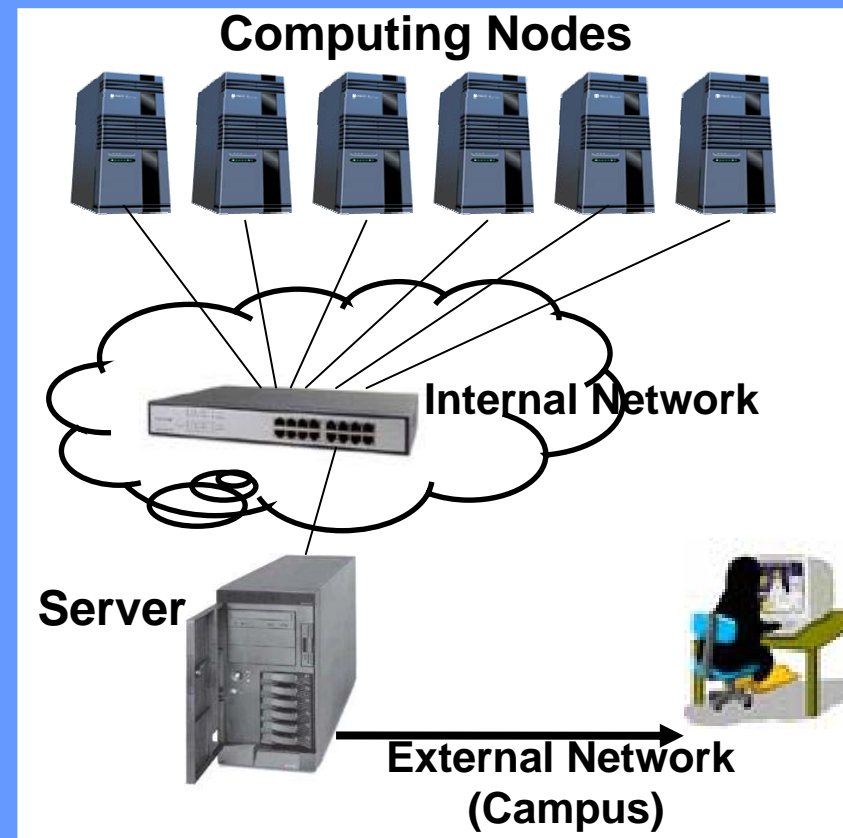
*If you were plowing a field, what would you rather use?
a strong ox, or 1024 chickens ?*
Seymour Cray (1925-1996)

- The fastest 10 computers in the world are computer clusters
 - www.top500.org
- Established companies produce clusters ready to use (IBM, HP, SG,..)
- Companies have grown and developed as Cluster specialist (Megaware)
- Triggered the development of more advanced/fast Network technologies
- We are not talking anymore about off-the-shell, but still pays off.



Designing Clusters

- Basic components
- Requirements of a High Performance Cluster
- Network technologies
- Data i/o and storage
- Maintenance and Physical conditions



Basic Components

- Set of standard PCs → **nodes**
- Connected by certain **Network** Technology
- Two types of Nodes
 - **Computing Nodes** used for the intensive calculations
 - **Server Nodes:**
 - export to the computer nodes a "*Networkfilesystem*"
 - *Application Software*
 - *Input/Output areas*
 - Interactive login
 - Automatic installation of the operating system on the nodes
 - **How many servers?**
 - How big is the cluster
 - How homogeneous

Designing Clusters:

What do you need the cluster for?

- CPU/Memory
intensive calculations
- Massive Input/Output
- Communication
Intensive applications
- Other special
applications

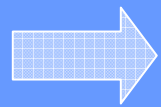
■ Important decisions to take

- How many nodes
- Single CPU or SMP nodes ?
- How about I/O
 - Diskless nodes?
- CPU Architecture
- Network Technology

Speed, quality, price. Pick any two.

James M. Wallace

CPU Architecture



Parameters

- ✓ CPU Frequency (~ 3 – 4 GHz)
- ✓ How many and how big Caches L1/L2/L3
- ✓ How much RAM (1-2GB)
- ✓ System Bus (ej. 400, 533, 800MHz)
- ✓ Motherboard. Chipset,...

Bandwidth

-Speed of data transmission
between RAM and CPU
- Bottleneck in HPC



CPU-RAM Bottleneck

```
REAL*8 (SIZE): A,B,C,D  
DO ITER=1,NITER  
  DO i=1,N  
    A(i) = B(i) + C(i) * D(i)  
  ENDDO  
ENDDO
```

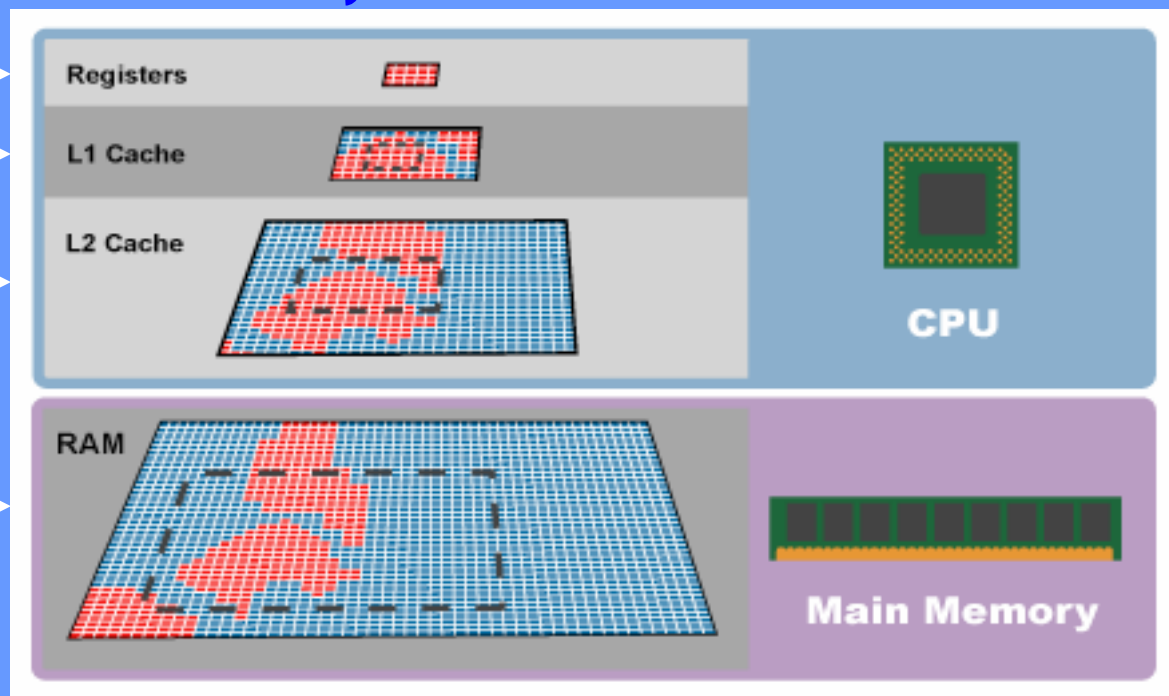
RINF Benchmark

1-3 ns (1K)

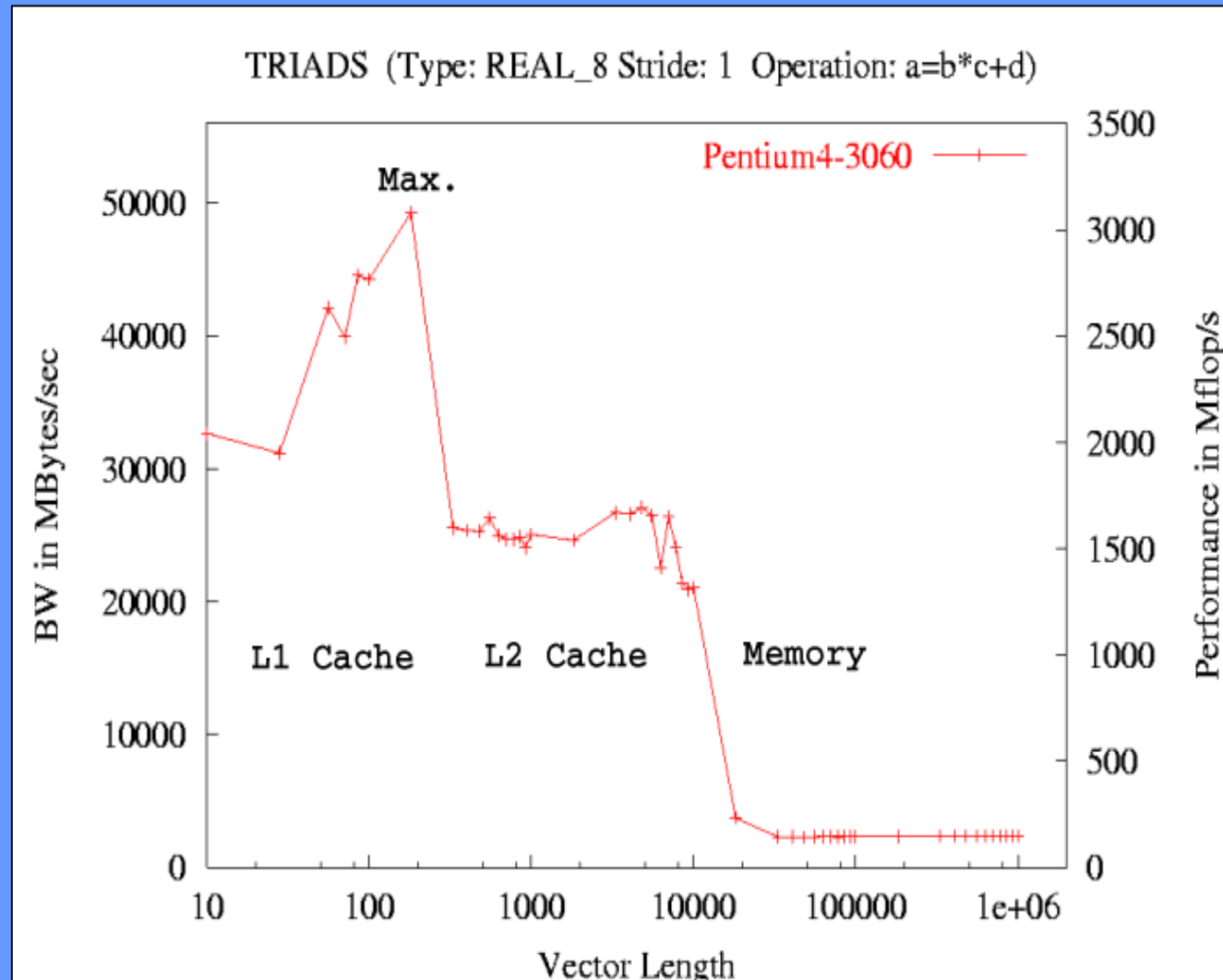
2-8 ns (8-128K)

5-12 ns (0.5-8M)

10-60ns (64M-2GB)



Benchmark RINF for the P4-3,06 GHz



6 Measurements

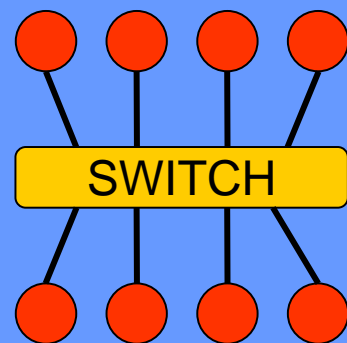
- Peak: **3066 MFlop/s**
- Memory: **189MFlop/s**

I am not sure how I will program a Petaflop machine, but I am sure that I will need MPI somewhere.

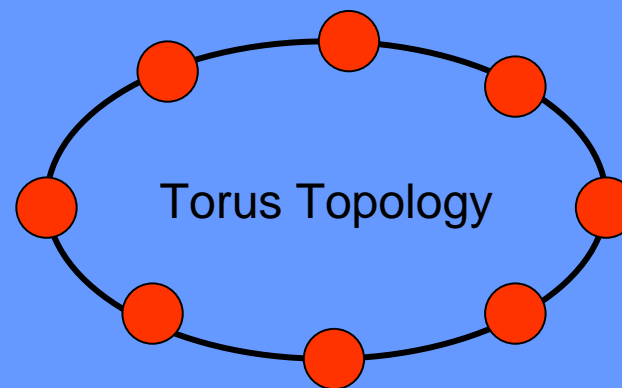
Horst D. Simon

Network Technology

- The nodes of the cluster can work together to solve a complex problem provided there is a Network connecting them and coordinating its work



Star Topology

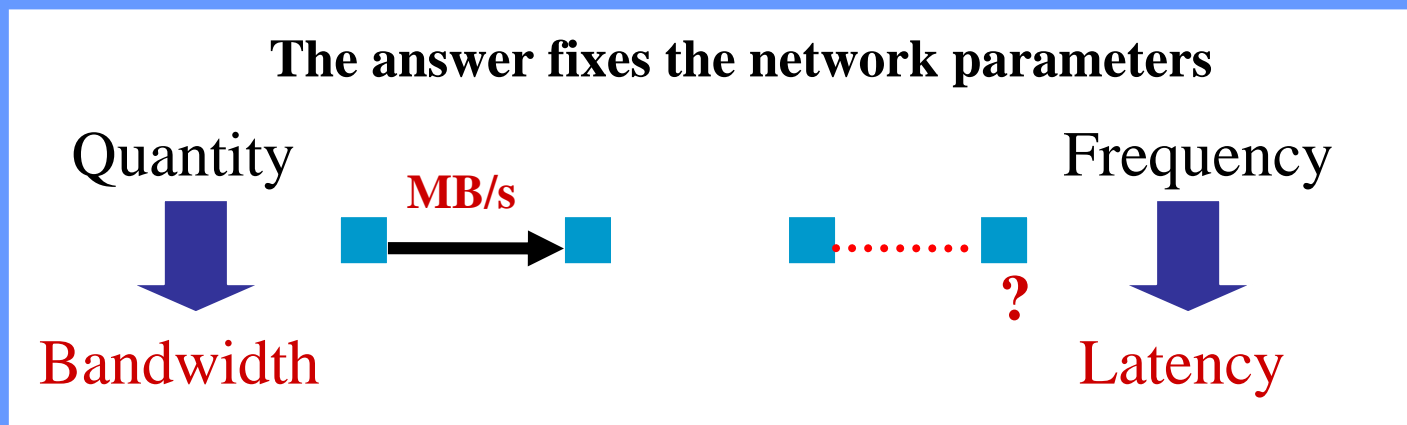


Network Technology, which one?

During the calculations the nodes need to exchange information among them and/or with the servers

1. *Input/output traffic*
 - Good connectivity to the server
2. *Parallel Computation*
 - The different nodes need to work collaboratively

How much and how frequently does this communication takes place ?



(How are data transmitted and exchanged)

❑ What is a BUS?

- ❑ Set of cables through which all internal components of the PC are connected to the CPU and the Main Memory (RAM)

❑ Parameters defining a Bus

- ❑ Width: 32-bit, o 64-bit
- ❑ Clock Frequency in MHz: How many times per second is the bus able to send those 32 or 64-bits

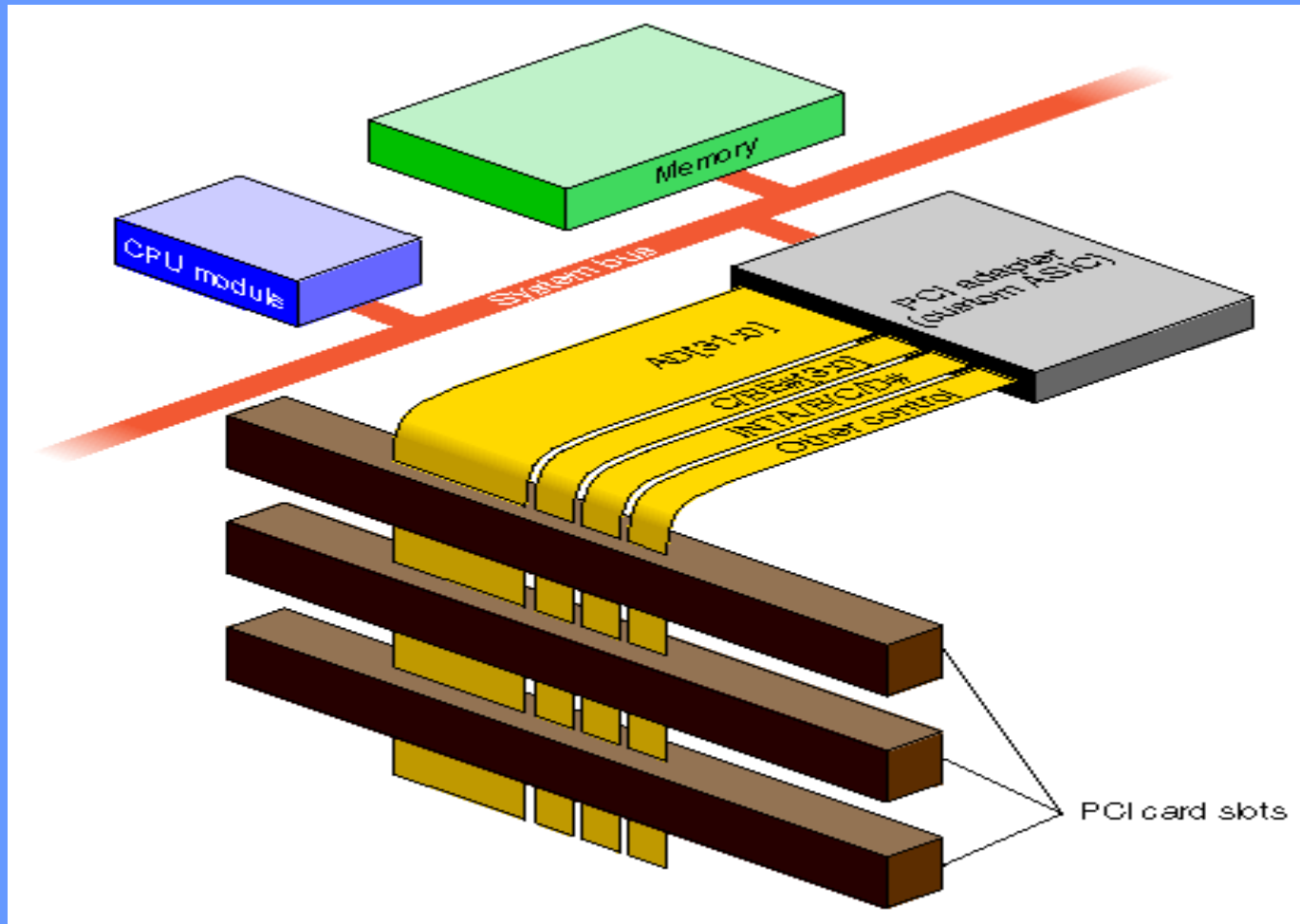
❑ A Bus consists of two parts

- ❑ Data Bus: data transmission itself
- ❑ Address Bus: transmission of the address where those data are send.

❑ What is the local BUS?

Special BUS used for transmission of those data requiring a specially fast transmiión especialmente. The local bus is directly connected to the CPU

Standar of local Bus: PCI



The status of data transmission

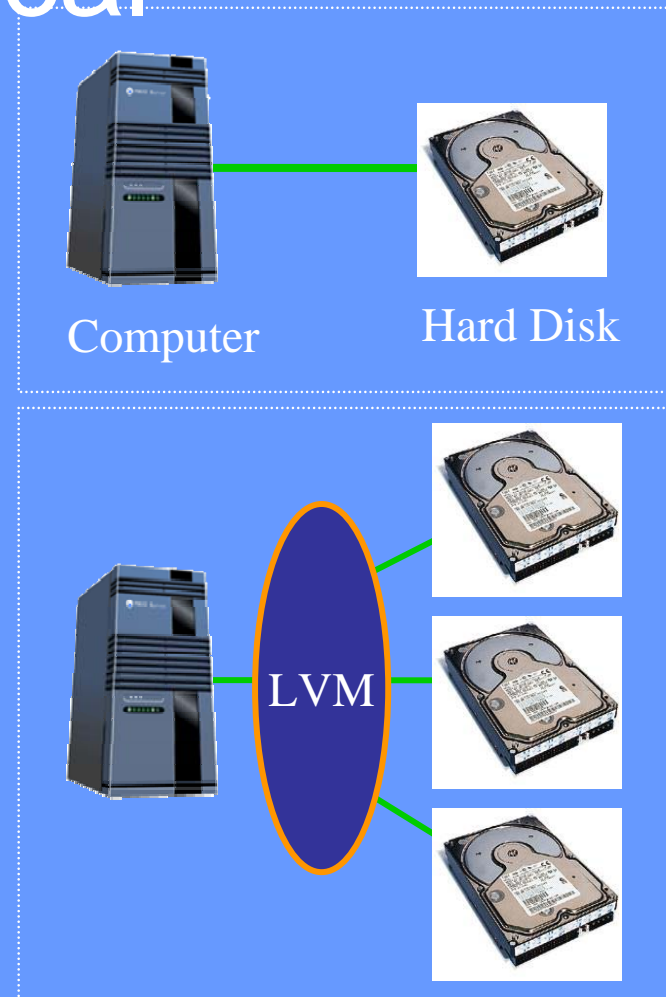
Technology	Type	BW (MB/s)	Latency (microsec)	Cost per node (€)	Make
Main Memory	BUS	~ 1000	< 0.01	-	-
Fast Ethernet	switched	11	70	~ 25	several
Gigabit Ethernet	switched	110	30	~100	several
Myrinet 2000	switched	~250	6	~ 1000	Myricom
Infiniband	switched	~1000	4	~ 1500	several
SCI	point-to- point	~330	2	1100-1600	Dolphin Intercon.

Cluster Storage: local

- Local Filesystems: ext2, XFS, NTFS, JFS, ...
- „Harddisk“ as data storage
- **Logical Volume Manager (LVM)**

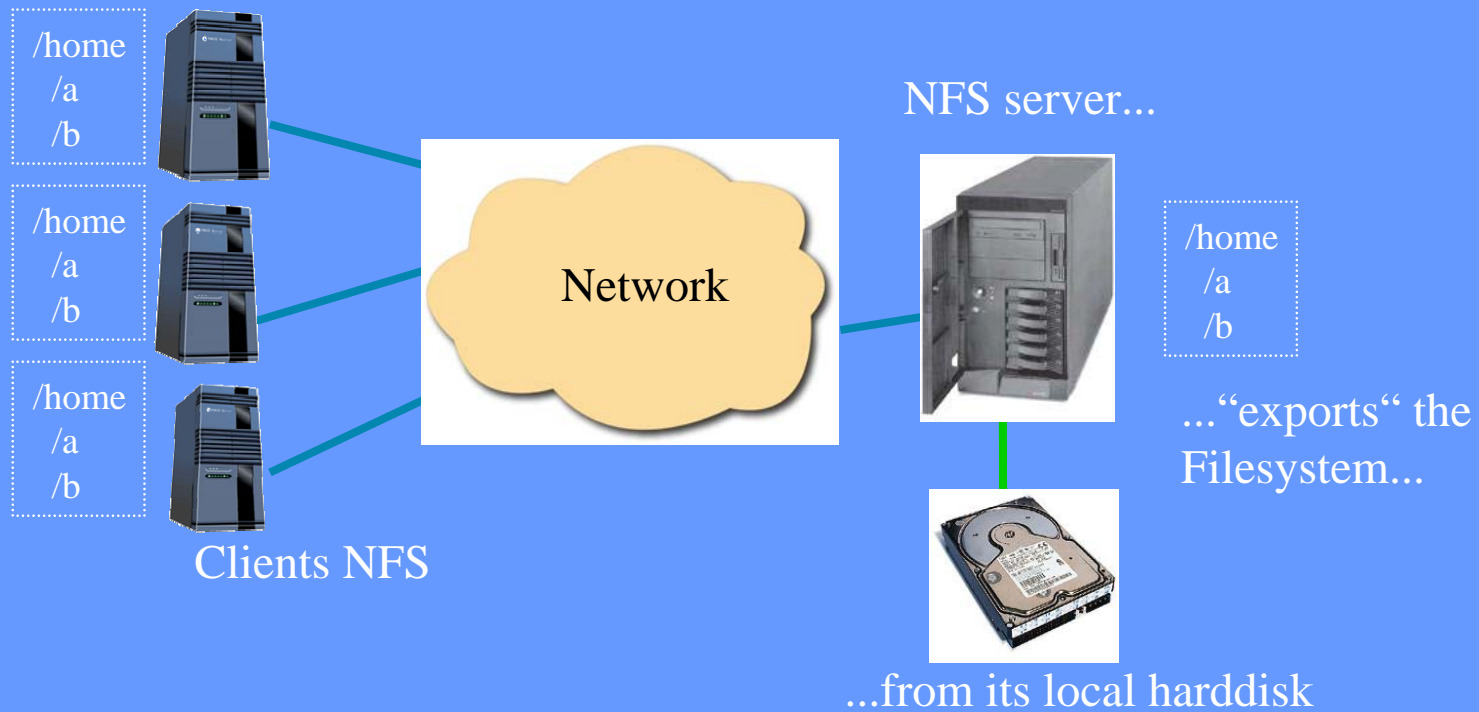
Makes maintenance easier

- Add or remove disks
- Striping between several disk



Storage: Network filesystem

- NFS – Network File System
Client-Server



- More advanced, using a similar concept is the AFS

Storage: scalability problem

- The NFS servers are overloaded with work when the cluster is "big"

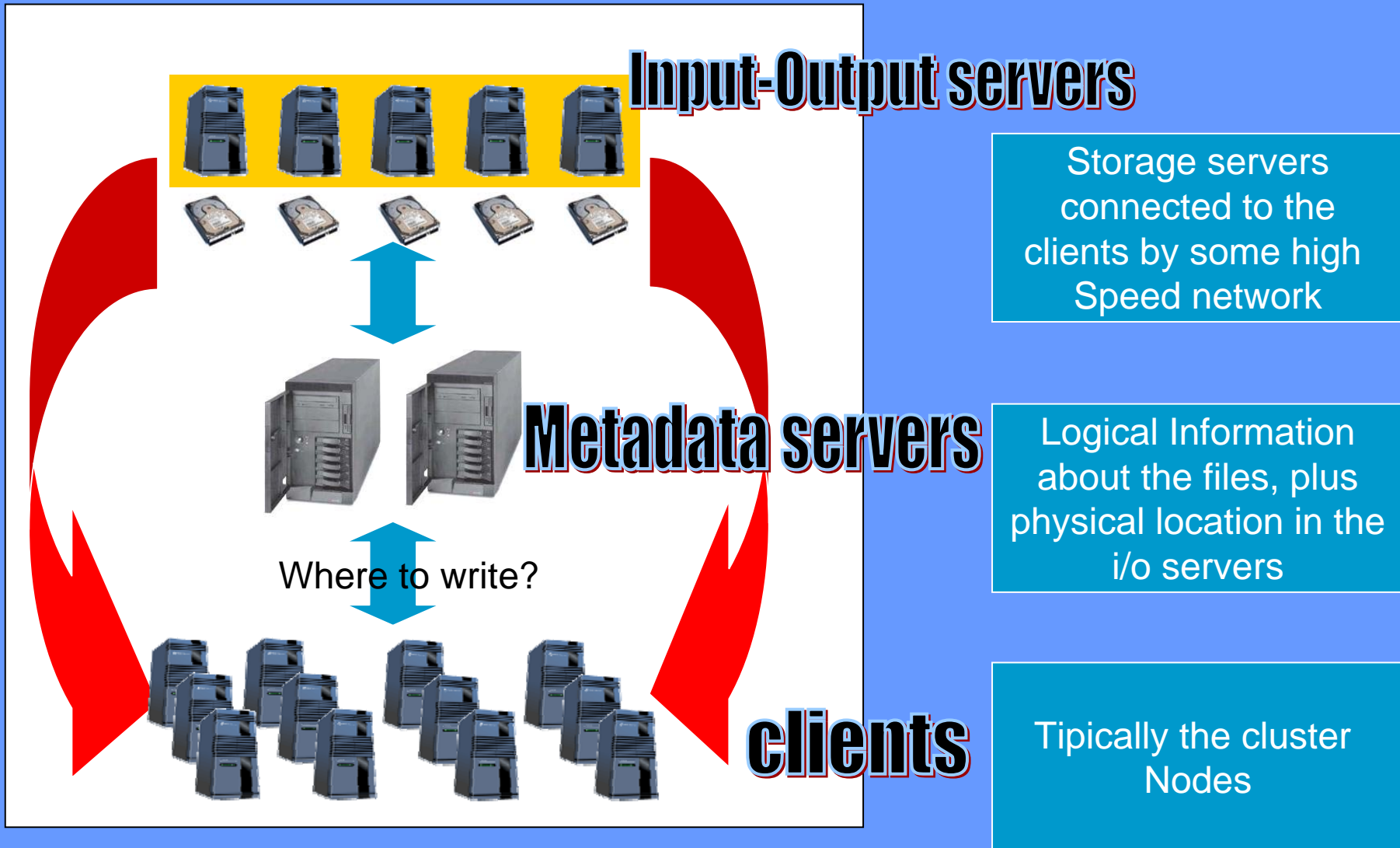
NFS does not scale with the cluster size
(is a resource sharing solution)

- Just adding more NFS servers

Besides the obvious administrative problem
data are not equally distributed between the servers
nodo1,..nodo20 → iosever1 ,.... nodo80,..nodo100 → iosever5

- ➔ Data must be distributed homogeneously among all the input/output servers **Parallel Filesystem** without human intervention...

Storage: Parallel Filesystems



Summarizing

- ✓ Set of PC's interconnected with current network technologies:
 - GigabitEth, Myrinet, Infiniband, ...**
- ✓ Linux as Operating System
- ✓ From the user/administration point of view a cluster is a "single entity" consisting in a number of computing nodes and
 - **One (or more) server nodes**
 - **One (or more) global filesystems**
 - **Batch System to manage the job's dispatching**

Batch Systems: Some definitions

- **BATCH**: a group of similar items produced or gathered together as a single item
- **QUEUE**: A queue is a buffer where "events" wait to be processed
 - Events → for us , this is the program we want to execute
 - An event of this type is called a JOB
 - JOBS are send to the queue using a particular syntax: Batch Scripts
 - Submitting a JOB means "sending" it to the queue
 - `qsub batch_script.sh`
- A **BATCH QUEUE** is a system software processing user request for job execution in an ordered way.
- Users submit to the queue their jobs for batch processing

Batch Systems: why are they needed?

- More effective use of the computers (ej. Uniform Load)
- Resource availability 24/7
- Resource allocation according to defined rules (who gets how much CPU and when)
- Fastest job execution (the systems finds out about the less busy nodes to allocate new tasks)

Our Objective:

The user tells the system the name of a file containing the description of the job (batch script)

- Executable and paths
- Serial or parallel
- Memory/CPU requirements

The Batch System should guarantee the fastest turn around

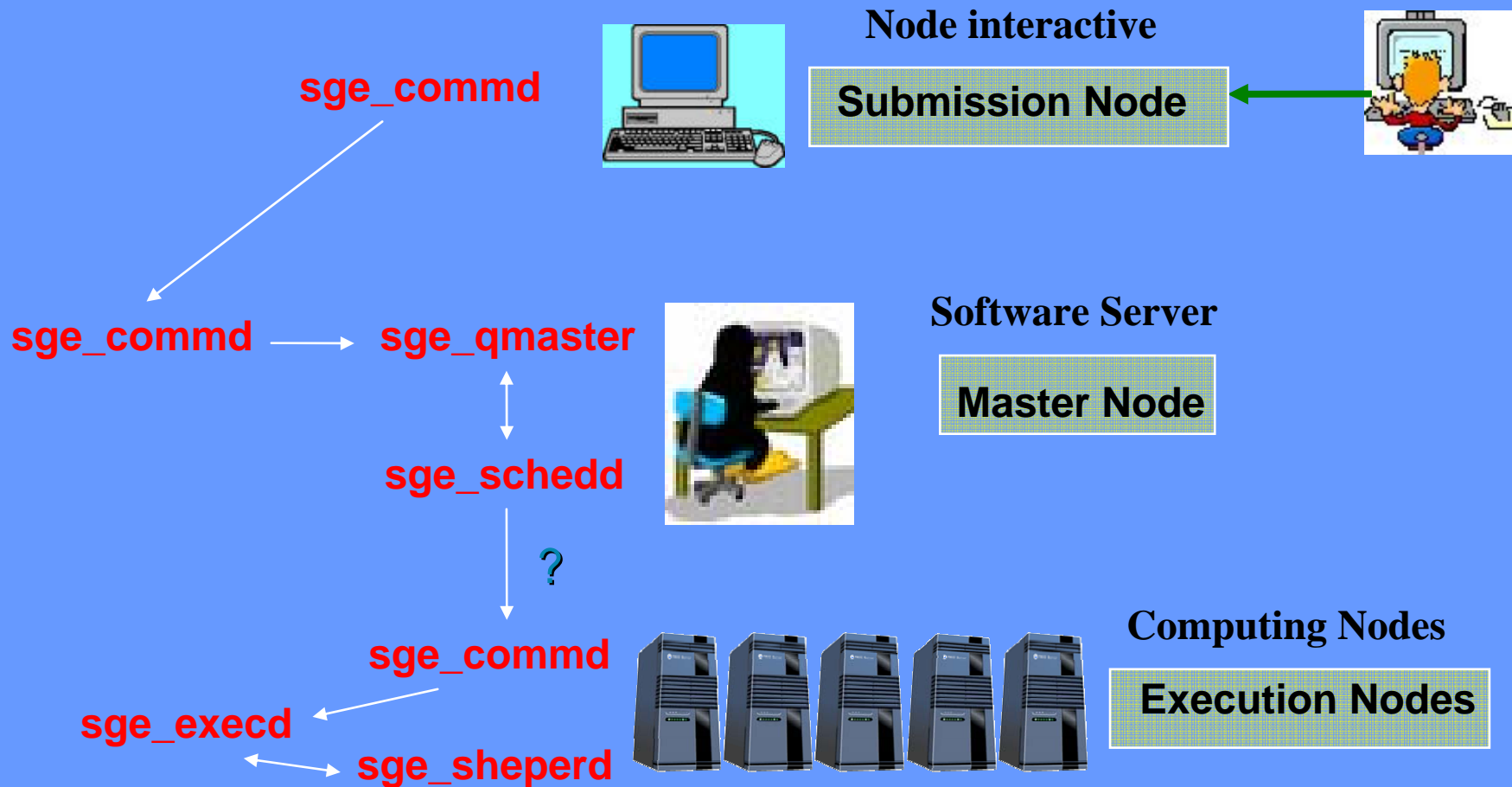
Some popular batch systems

- OpenPBS
- TORQUE
- Sun Grid Engine
- Condor
- LSF

We will explain how a batch queue works using SGE as an example

<http://gridengine.sunsource.net>

SGE cluster point of view



How does it work?

The user "submits" a job to a batch queue, and then

- **sgc_qmaster:** controls and monitors the activity
- **sgc_commd:** runs on every node (listening over a TCP/IP port)
- **sgc_sched:** if there are free processors, it allocates the job to one of them, if not, the job is queued and scheduled for later according to certain rules. The scheduler decides which job from the queue is next
 - Different policies: first in-firs out (FIFO), fairshare, etc...
- **sgc_shepherd:** takes care of starting-completion mechanism of jobs
 - Stops execution when limits are reached...
- **sgc_execd**
 - Executes the job

Queue configuration

Queue Configuration - Modify

Queue: lxbifi01

Hostname: lxbifi01.bifi.unizar.es

Buttons: Clone, Reset, Refresh, Ok, Cancel, Help

Complexes | Subordinates | User Access | Owners

General Configuration | Execution Method | Checkpointing | Load/Suspend Thresholds | Limits

Sequence Nr: 0 | Calendar: [] | Type: Batch, Interactive, Checkpointing, Parallel

Processors: UNDEFINED | Notify Time: 00:00:60

tmp Directory: /tmp | Job's Nice: 0

Shell: /bin/csh | Slots: 1

Shell Start Mode: NONE | Initial State: default | Rerun Jobs:

- Type: batch, interactive parallel
- Access rights: users handling
- Limits: CPU, RAM, etc...
- Execution Methods: how to handle jobs at starting time

QMON +++ Queue Control

SGE **Queue Control**

 Ixsrvt143 Ixsrvt143 Slots: 2 (2)	 Ixsrvt144 Ixsrvt144 Slots: 1 (2)	 Ixsrvt015 Ixsrvt15 Slots: 1 (1)	 Ixsrvt016 Ixsrvt16 Slots: 1 (1)	 Ixsrvt017 Ixsrvt17 Slots: 1 (1)	 Ixsrvt018 Ixsrvt18 Slots: 1 (1)	 Ixsrvt019 Ixsrvt19 Slots: 2 (2)
 Ixsrvt002 Ixsrvt2 Slots: 0 (2)	 Ixsrvt020 Ixsrvt20 Slots: 2 (2)	 Ixsrvt021 Ixsrvt21 Slots: 2 (2)	 Ixsrvt022 Ixsrvt22 Slots: 2 (2)	 Ixsrvt023 Ixsrvt23 Slots: 2 (2)	 Ixsrvt024 Ixsrvt24 Slots: 0 (2)	 Ixsrvt025 Ixsrvt25 Slots: 2 (2)
 Ixsrvt026 Ixsrvt26 Slots: 2 (2)	 Ixsrvt027 Ixsrvt27 Slots: 2 (2)	 Ixsrvt028 Ixsrvt28 Slots: 4 (4)	 Ixsrvt029 Ixsrvt29 Slots: 4 (4)	 Ixsrvt003 Ixsrvt3 Slots: 0 (2)	 Ixsrvt030 Ixsrvt30 Slots: 4 (4)	 Ixsrvt031 Ixsrvt31 Slots: 2 (4)
 Ixsrvt032 Ixsrvt32 Slots: 2 (2)	 Ixsrvt033 Ixsrvt33 Slots: 1 (1)	 Ixsrvt034 Ixsrvt34 Slots: 2 (2)	 Ixsrvt035 Ixsrvt35 Slots: 2 (2)	 Ixsrvt036 Ixsrvt36 Slots: 2 (2)	 Ixsrvt037 Ixsrvt37 Slots: 2 (2)	 Ixsrvt038 Ixsrvt38 Slots: 2 (2)
 Ixsrvt039 Ixsrvt39 Slots: 2 (2)	 Ixsrvt004 Ixsrvt4 Slots: 0 (2)	 Ixsrvt040 Ixsrvt40 Slots: 0 (2)	 Ixsrvt041 Ixsrvt41 Slots: 2 (2)	 Ixsrvt042 Ixsrvt42 Slots: 0 (2)	 Ixsrvt043 Ixsrvt43 Slots: 1 (2)	 Ixsrvt044 Ixsrvt44 Slots: 1 (2)
 Ixsrvt045 Ixsrvt45 Slots: 0 (2)	 Ixsrvt046 Ixsrvt46 Slots: 0 (2)	 Ixsrvt047 Ixsrvt47 Slots: 0 (2)	 Ixsrvt048 Ixsrvt48 Slots: 1 (2)	 Ixsrvt049 Ixsrvt49 Slots: 0 (2)	 Ixsrvt005 Ixsrvt5 Slots: 0 (2)	 Ixsrvt050 Ixsrvt50 Slots: 2 (2)
 Ixsrvt051 Ixsrvt51 Slots: 1 (1)	 Ixsrvt052 Ixsrvt52 Slots: 1 (1)	 Ixsrvt053 Ixsrvt53 Slots: 1 (1)	 Ixsrvt054 Ixsrvt54 Slots: 2 (2)	 Ixsrvt055 Ixsrvt55 Slots: 1 (1)	 Ixsrvt056 Ixsrvt56 Slots: 1 (1)	 Ixsrvt057 Ixsrvt57 Slots: 1 (1)

Key

- Running
- Suspended
- Disabled
- Alarm
- Error
- Calendar Suspend
- Calendar Disable

Refresh

Add

Modify

Force

Suspend

Resume

Disable

Enable

Reschedule

Clear Error

Delete

Customize

Done

Help

Submitting serial Jobs to SGE

```
#!/bin/sh
#$ -o $HOME/mydir/myjob.out -j y
#$ -N myjob
#$ -M abcd@mydomain
#$ -l 32bit=yes
. /etc/profile.local
cd mydir
./myprog
```

} Batch_serial.sh

```
qsub batch_script.sh
```

Submitting Parallel Jobs in SGE

```
#!/bin/sh
#$ -o $HOME/mydir/myjob.out -j y
#$ -N myjob
#$ -M abcd@mydomain
#$ -l 32bit=yes
#$ -pe infiniband 4-8
```

← **How many processes ??**

```
. /etc/profile.local
. mpi.setup -e infiniband
cd mydir
```

← **Setting up \$ENV**

```
mpirun -np $NSLOTS ./myprog
```

**Variable \$NSLOTS contains the number of
← allocated processes**

Multiple Parallel environments

QMON +++ Parallel Environment Configuration <@lxsrv1>

GRID ENGINE Parallel Environment Configuration

PE List	Configuration
amber64	PE Name infiniband
eirene	Slots 14
infiniband	Queues node100 node101 node102 node103 node104 node105 node106
lammpi	Users LatticeQCD
make	Xusers NONE
mpi	Start Proc Args /usr/local/sys/sge/mpi/startinfiniband.sh \$pe_hostfile
openmpi	Stop Proc Args /usr/local/sys/sge/stopinfiniband.sh
	Allocation Rule \$round_robin
	Control Slaves false
	Job is first task false

Buttons: Add, Modify, Delete, Done, Help

Infiniband Parallel Queues setting

Name: infiniband Slots: 14

Queue List: node100, node101, node102, node103, node104, node105, node106

User Lists: LatticeQCD

Xuser Lists: (empty)

Start Proc Args: /usr/local/sys/sge/mpi/startinfiniband.sh \$pe_hostfile

Stop Proc Args: /usr/local/sys/sge/stopinfiniband.sh

Allocation Rule: \$round_robin

Control Slaves Job is first task

Buttons: Ok, Cancel

Parallel Environments in SGE

■ MPICH, OpenMPI and MVAPICH

- Start procedure
 - SGE allocates hosts
 - PEHostfile
 - The PEHostfile is translated to the language of the MPI
\$TMPDIR/machines
- Stop procedure
 - Deletes
\$TMPDIR/machines

■ LAMMPI

- Start procedure
 - SGE allocates hosts
 - PEHostfile
 - The PEHostfile is translated to the language of the \$machines
 - Starts lamboot everywhere
\$LAMMPIR_H/bin/lamboot \$machines
- Stop procedure
 - kills lamboot daemon
 - Deletes /tmp/machines

Limitations of the SGE scheduler



Starvation in the queue of jobs requesting large number of resources
(Memory, PE, Disk,...)

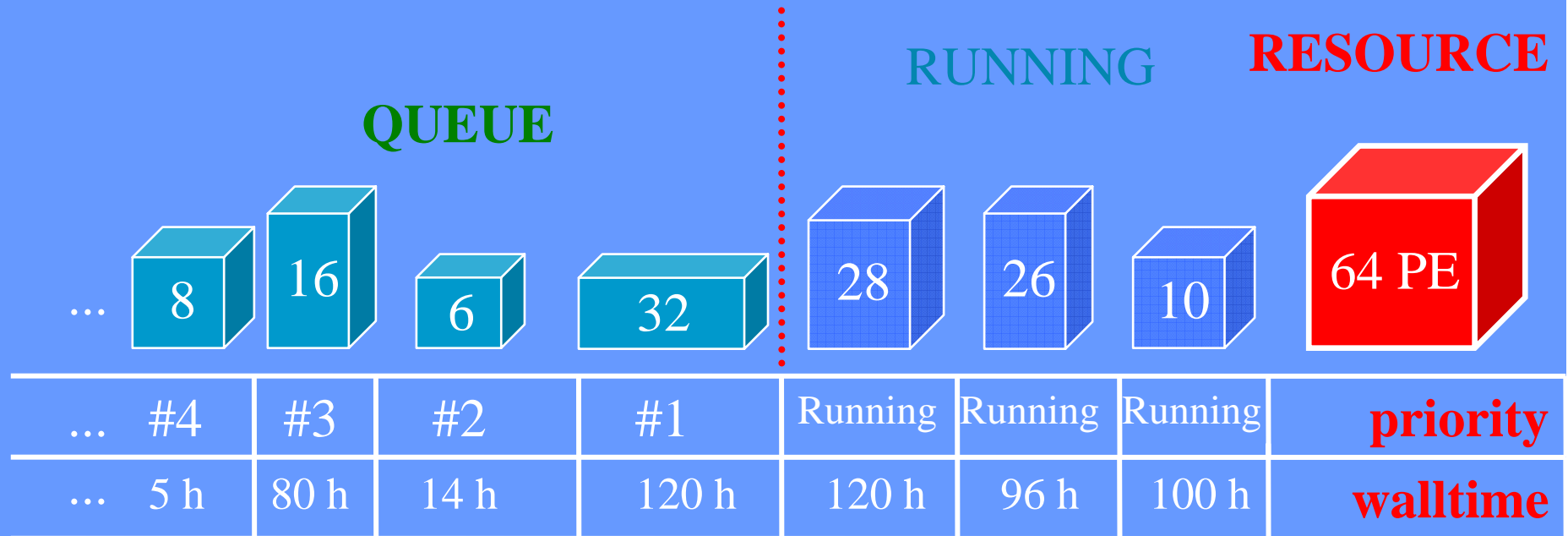
- Example: 64 Processors cluster

```
$ qstat -a
```

JOBID	USER	Proc. REQ	STATUS	submitted/started at
1222	t3828	6	Running	7/07/03 22:08:09
134	a2825	24	Queued	1/06/03 16:02:34
1224	a2802	2	Running	13/07/03 18:55:44

MAUI Scheduler provides mechanisms to avoid job starvation

SGE Scheduler: where is the problem?



The queued jobID #1 will have problems to start
the machine is not likely to have 32 processors free simultaneously

Maui Scheduler: Generalities

■ Availability

- Free download
<http://supercluster.org/maui>
- Operating Systems
Linux, AIX, Tru-64, Solaris,...
- Resource Managers supported
PBS, SGE, Loadleveler, LSF,...

■ Installation

- `gtar -xzvf maui-3.2.6.tgz`
- `cd maui-3.2.6`
- `./configure & make`

■ Configuration

- Parameters given in
`/etc/maui.cfg`

■ Start-up

- Remove initialization of batch system scheduler
- Restart batch system
- `/usr/local/bin/maui`

■ Testing environment

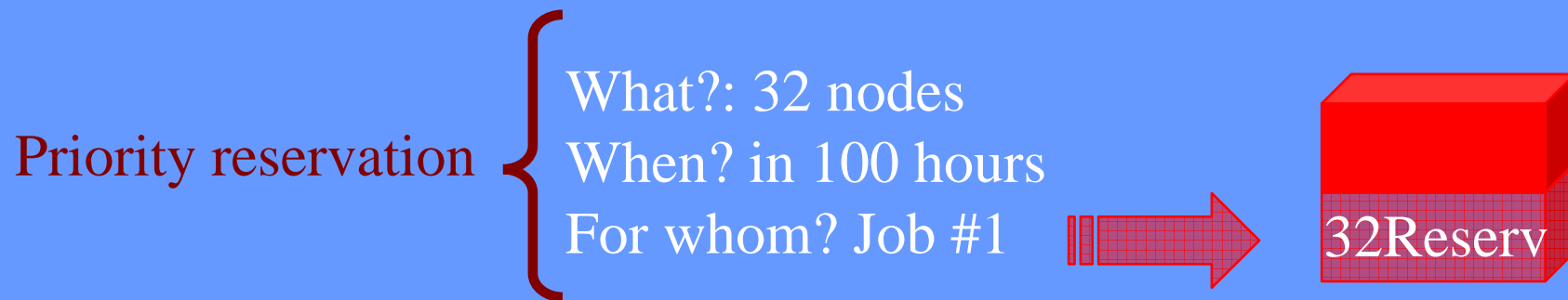
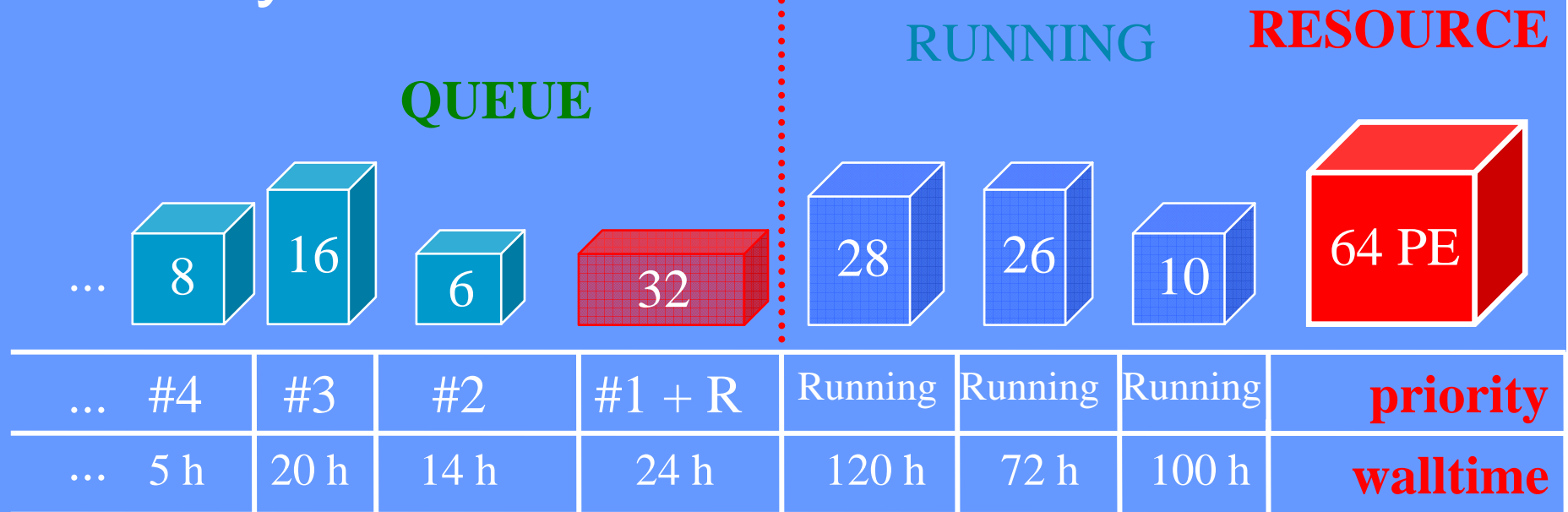
- Normal mode
- Test mode
- Simulation mode

} **simultaneously**

Job Submission in SGE-Maui



How does Maui Scheduler help: Priority Reservations

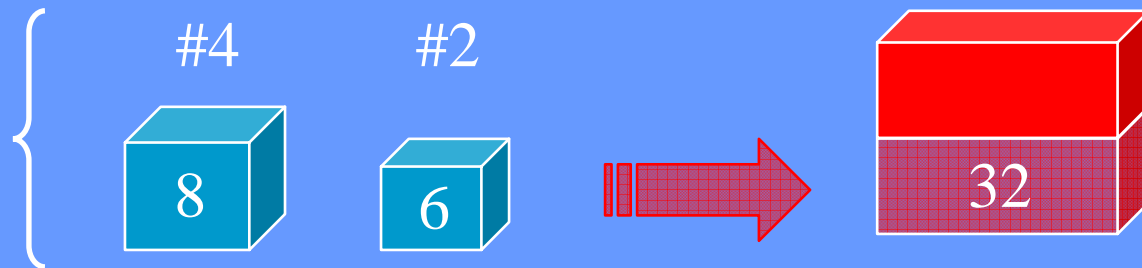


Further improvements

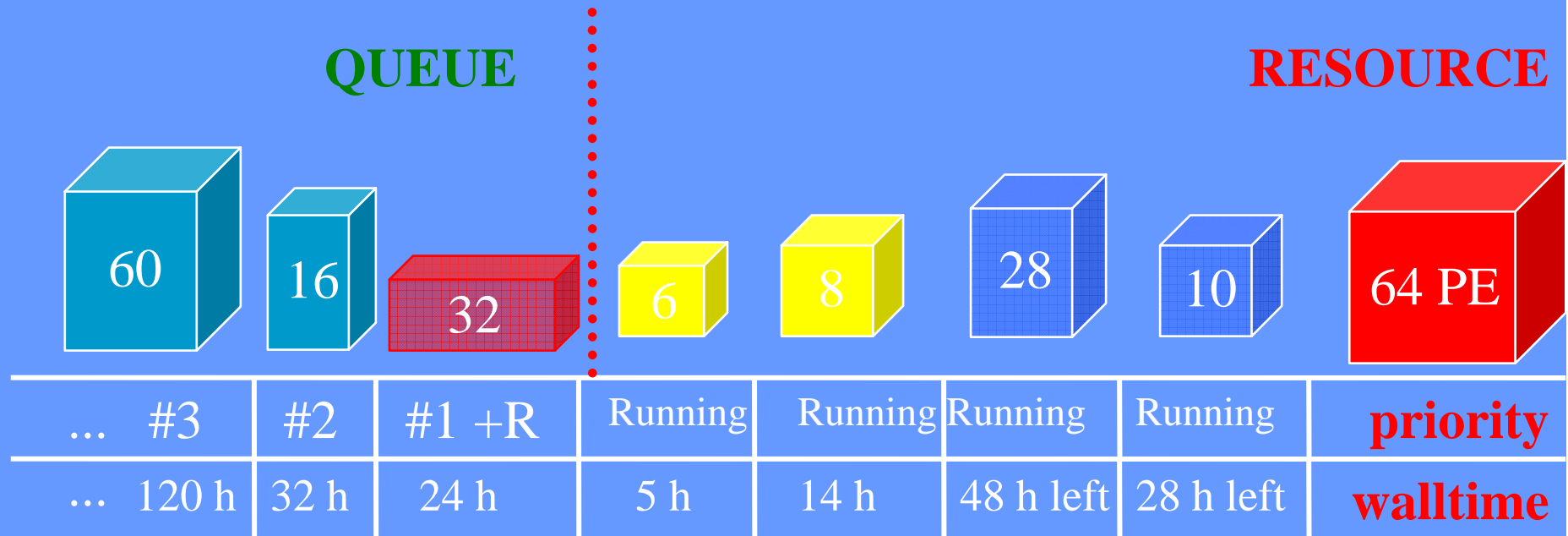
Backfill policy ON



Jobs #2 and #4 are **Backfilled** during Remaining 28 h



What happens 72 hours later ?



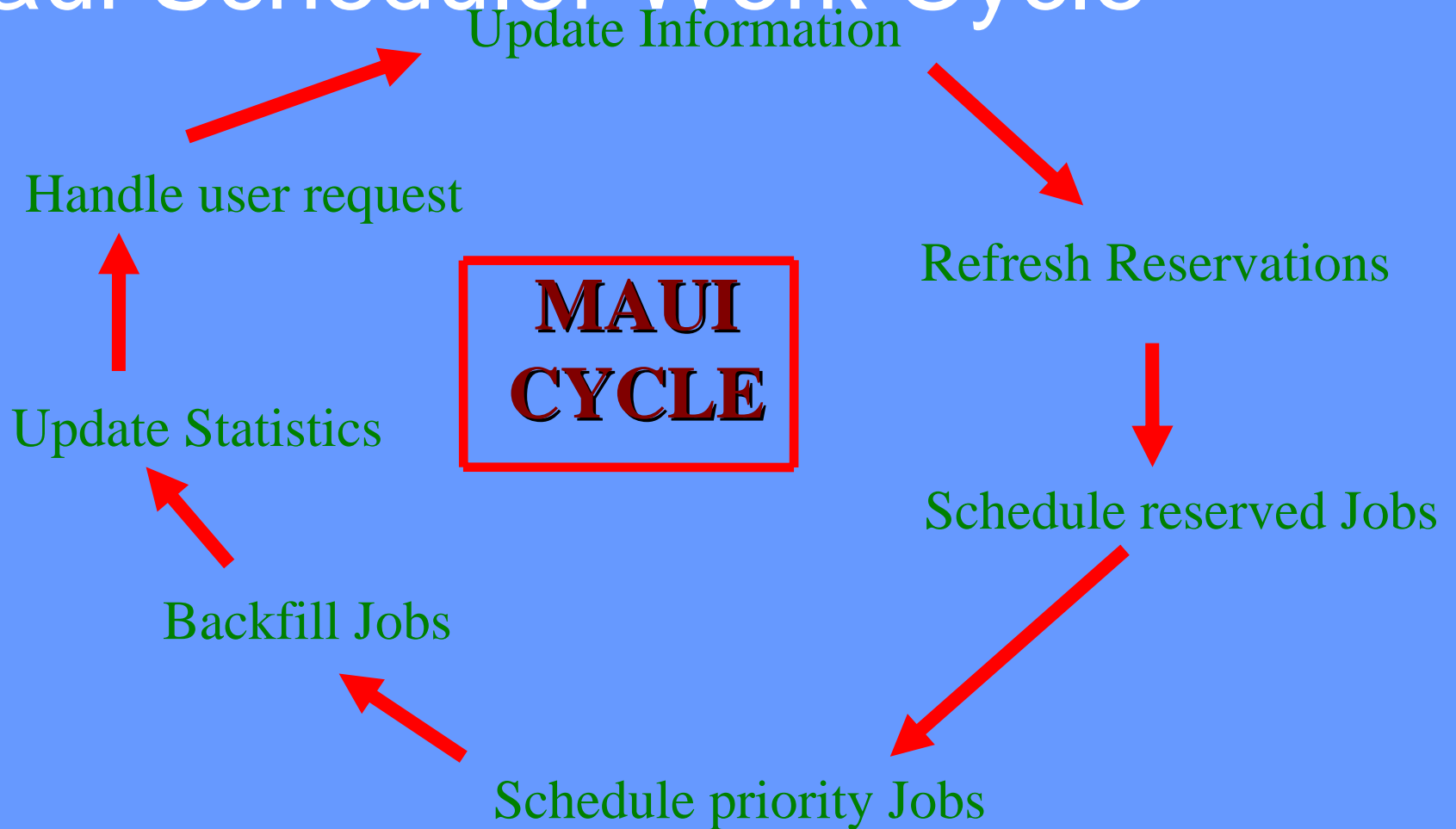
 Queued job

 Job running by SGE means

 PR job

 Backfilled job

Maui Scheduler Work Cycle



Maui Scheduler: Priority Reservation + Backfilling

- **BACKFILLING** out-of-order execution

Optimal resource usage



Precise knowledge
of walltime crucial
!!!!

- **PRIORITY RESERVATION** should be enabled to prevent job starvation

The background features a dark blue vertical bar on the left side. To its right, there is a grid of squares in various shades of blue and teal, arranged in a pattern that tapers to the right. The rest of the background is a solid light blue color.

Muchas gracias!