UL HPC School 2016

Overview and Challenges of the UL HPC Facility at the Belval Horizon

Sebastien Varrette, PhD

Nov. 25th, 2016, MSA auditorium 3.330

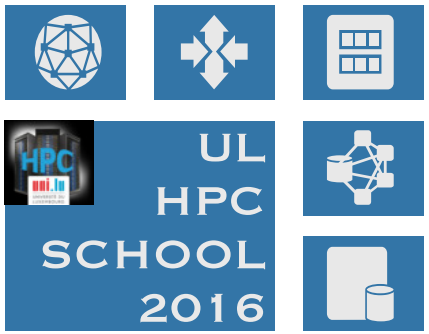
University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Summary

Welcome to the UL HPC School 2016

https://hpc.uni.lu/hpc-school/



- **4th edition** of this training session...
  - ↪ previous editions: May 2014, Mar/Jun 2015
  - ↪ This one mainly targeting newcomers
    - ✓ 1-day event, focusing on the basic tutorials

- Next edition planned for **March, 2017** in Belval
  - ↪ 2-days event,
  - ↪ 3 sessions in parallel
  - ↪ covering all usages, from basic to advanced

Agenda: Nov 25th, 2016

Time	Session
09:30 – 10:30	**Keynote**: Overview and Challenges of the UL HPC Facility at the Belval Horizon
10:30 – 10:45	**Coffee Break**
10:45 – 12:30	PS1A: **Getting Started on the UL HPC platform**
12:30 – 13:45	**Lunch** (and UL HPC Data center visit for those interested)
13:45 – 15:00	PS2A: **HPC workflow with sequential jobs**
15:00 – 16:00	PS3A: **Debugging, profiling and performance analysis**
16:00 – 16:15	**Coffee break**
16:15 – 18:00	PS4A: **HPC workflow with Parallel/Distributed jobs**
18:00 –	**Beers ;)**

PS = *Practical Session using your laptop*

- **Requirement**:
  - ↪ your favorite laptop with your favorite OS
    - ✓ Linux / Mac OS preferred, but Windows accepted
  - ↪ basic knowledge in Linux command line
  - ↪ ability to take notes (Markdown etc.)

Summary

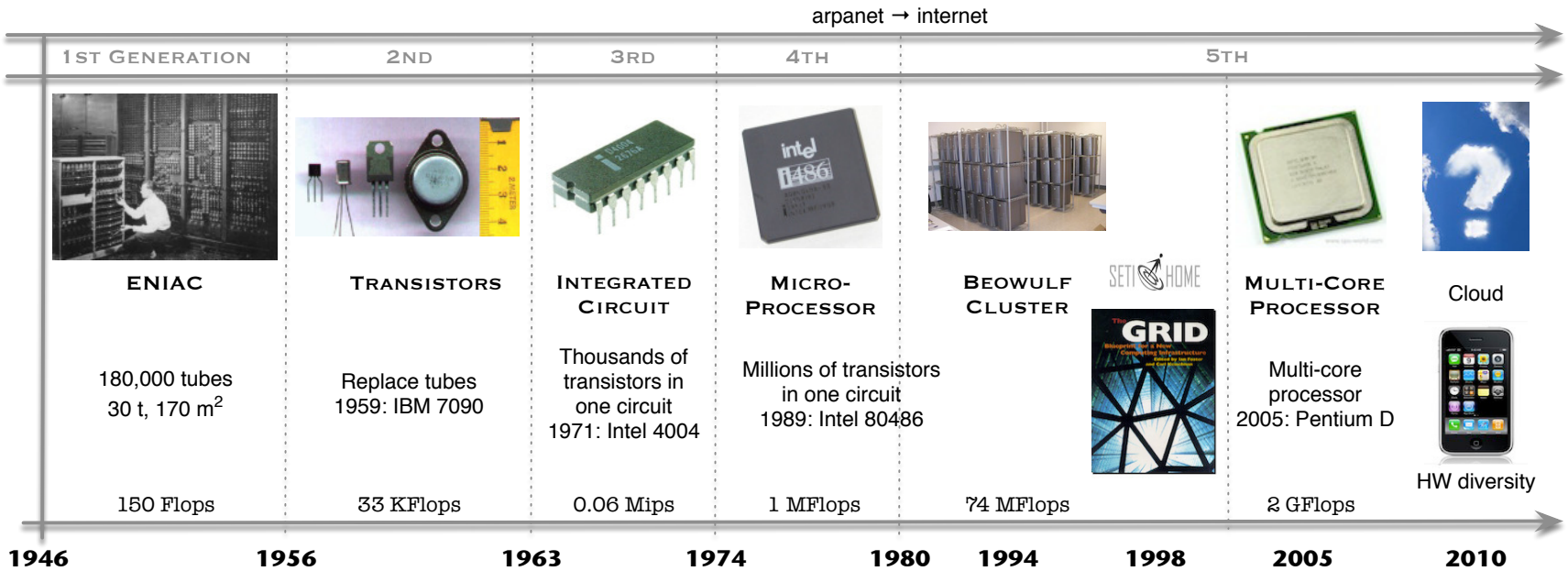
Prerequisites

- **HPC**: **H**igh **P**erformance **C**omputing

Main HPC Performance Metrics

- Computing Capacity/speed: often measured in **flops** (or **flop/s**)
  - ↪ **Fl**oating **p**oint **o**perations per **s**econds (often in DP)
  - ↪ **GFlops** $= 10^9$ Flops **TFlops** $= 10^{12}$ Flops **PFlops** $= 10^{15}$ Flops
- Storage Capacity: measured in multiples of **bytes** $= 8$ **bits**
  - ↪ **GB** $= 10^9$ bytes **TB** $= 10^{12}$ bytes **PB** $= 10^{15}$ bytes
  - ↪ **GiB** $= 1024^3$ bytes **TiB** $= 1024^4$ bytes **PiB** $= 1024^5$ bytes
- Transfert rate on a medium measured in **Mb/s** or **MB/s**
- Other metrics: Sequential vs Random **R/W speed**, **IOPS** ...

Evolution of Computing Systems



Why High Performance Computing ?

"The country that out-computes will be the one that out-competes". — *Council on Competitiveness*

- **Accelerates** research by accelerating **computations**

$\simeq$ 20 **G**Flops (Dual-core i5 1.6GHz)

85.543 **T**Flops (522 computing nodes,5420 cores)

- Increases **storage** capacity and velocity for Big Data processing

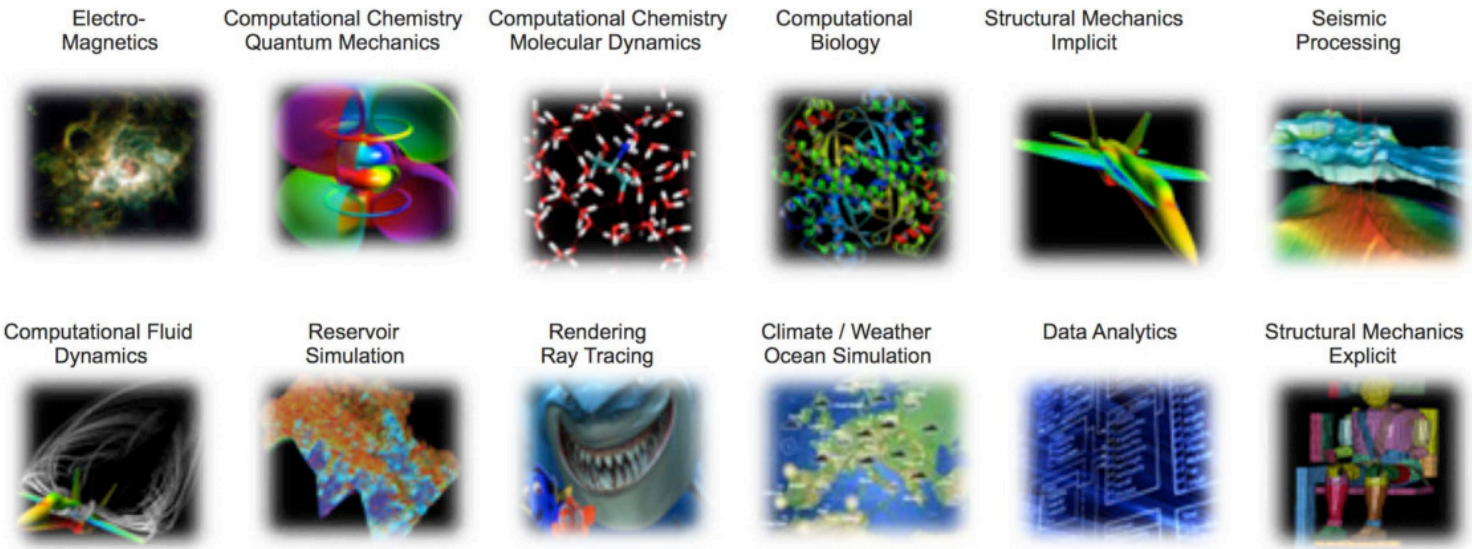
2TB (1 disk, 300 MB/s)

5354.4TB (1558 disks, 7 GB/s)

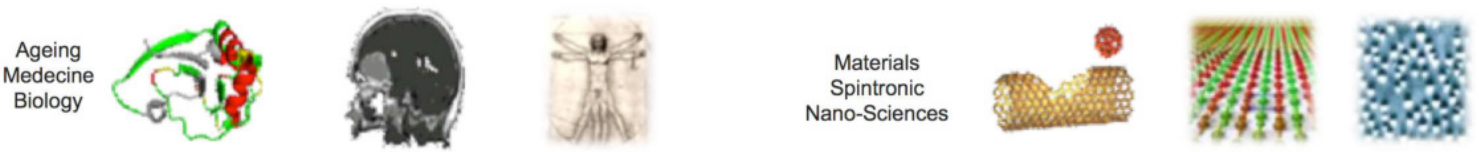
- **Communicates faster** 1 GbE (1 Gb/s) vs Infiniband QDR (40 Gb/

HPC at the Heart of our Daily Life

- **Today**: Research, Industry, Local Collectivities



- **. . . Tomorrow**: applied research, digital health, nano/bio techno

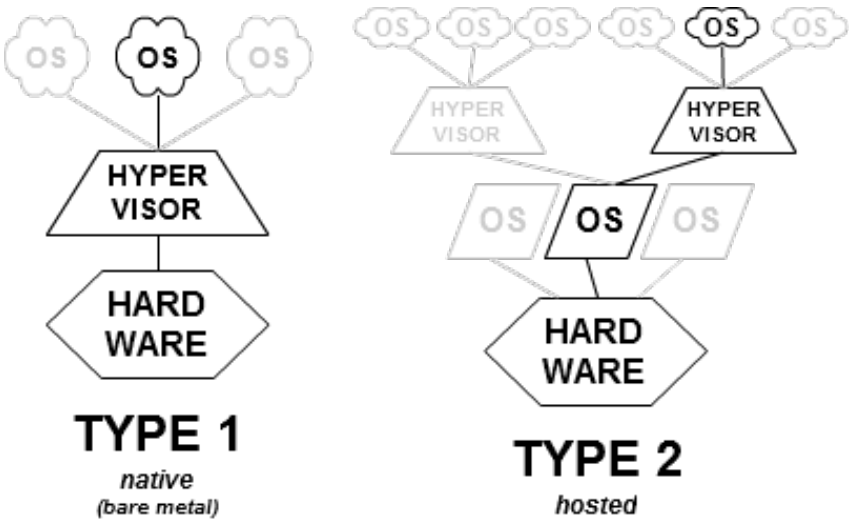


Computing for Researchers: Laptop

- Regular PC / Local Laptop / Workstation
  - ↪ Native OS (Windows, Linux, Mac etc.)

Computing for Researchers: Laptop

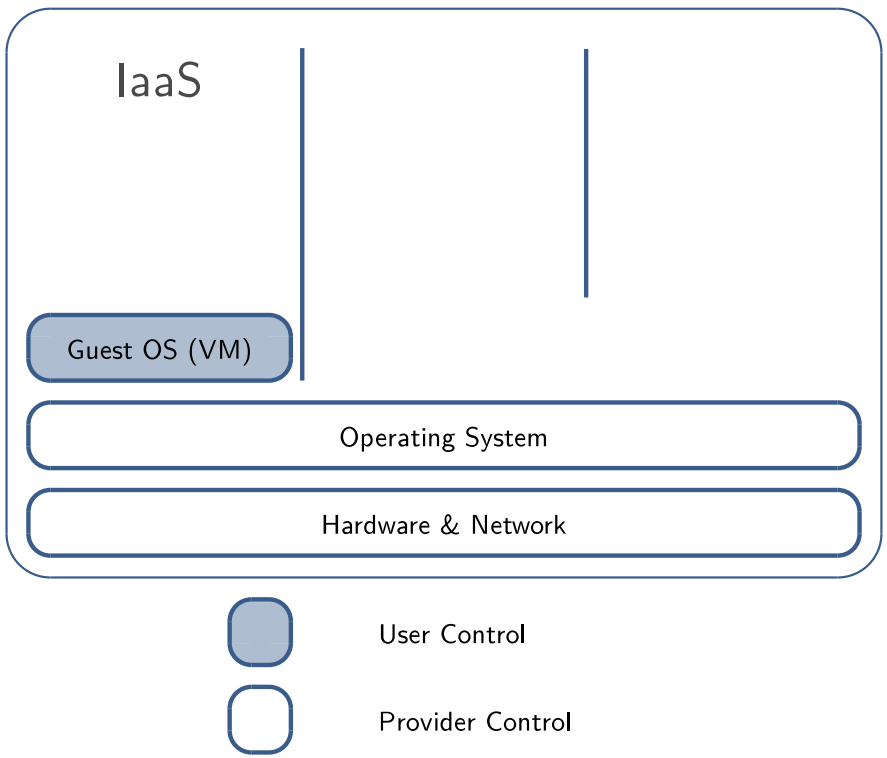
- Regular PC / Local Laptop / Workstation
  - ↪ Native OS (Windows, Linux, Mac etc.)
  - ↪ Virtualized OS through an **hypervisor**
    - ✓ Hypervisor: core virtualization engine / environment
    - ✓ **Performance loss:** $\geq 20\%$



Xen, VMWare ESXi, KVM VirtualBox

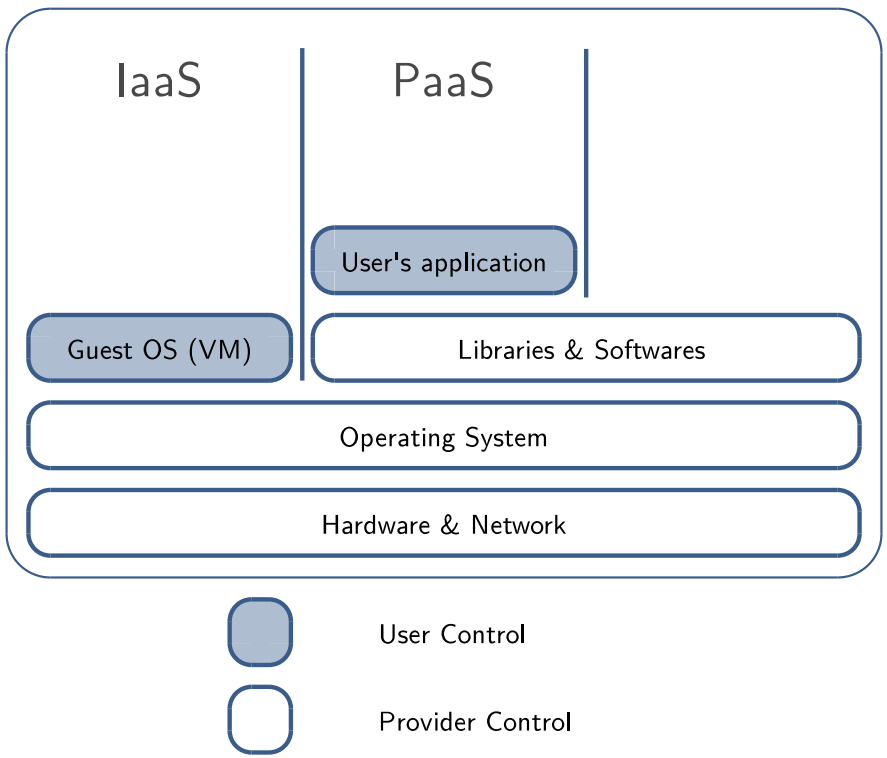
Computing for Researchers: Cloud

- Cloud Computing Platform
  - ↪ **Infrastructure** as a Service (**S**aaS)



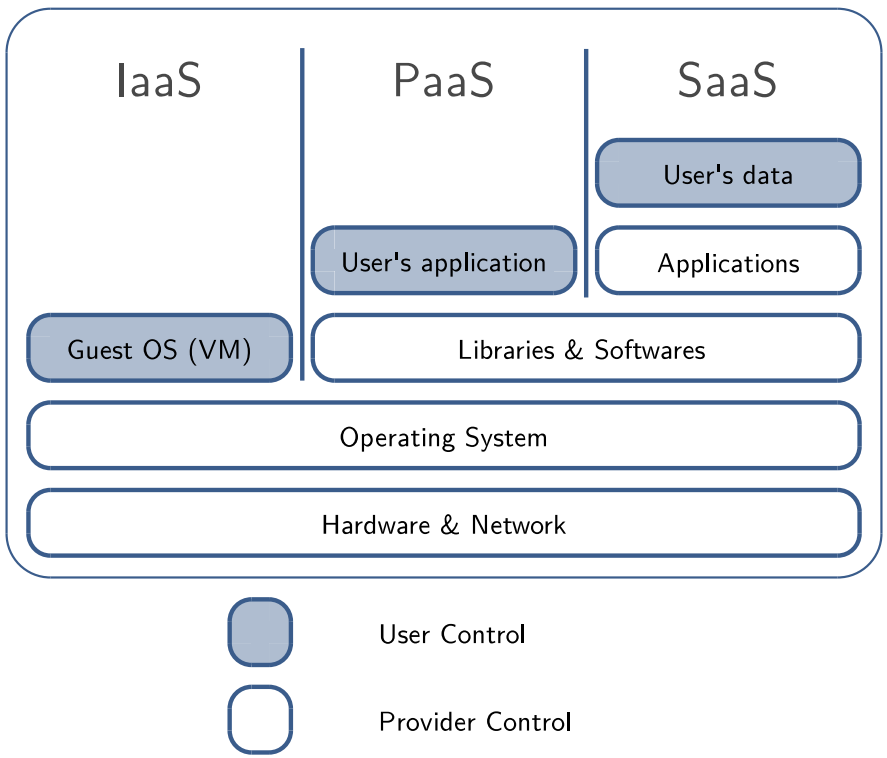
Computing for Researchers: Cloud

- Cloud Computing Platform
  - ↪ **Platform** as a Service (**P**aaS)



Computing for Researchers: Cloud

- Cloud Computing Platform
  - ↪ **Software** as a Service (**I**aaS)



Computing for Researchers: HPC

- High Performance Computing platforms
  - ↪ For **Speedup**, **Scalability** and **Faster Time to Solution**

Computing for Researchers: HPC

- High Performance Computing platforms
  - ↪ For **Speedup**, **Scalability** and **Faster Time to Solution**

YET...

$$\mathbf{PC} \neq \mathbf{HPC}$$

Computing for Researchers: HPC

- High Performance Computing platforms
  - ↪ For **Speedup**, **Scalability** and **Faster Time to Solution**

YET...

PC $\neq$ HPC

- HPC $\simeq$ Formula 1
  - ↪ can end badly, even after minor errors

Jobs, Tasks & Local Execution

```
$> ./myprog
```

Jobs, Tasks & Local Execution

```
$> ./myprog
```

Jobs, Tasks & Local Execution

```
$> ./myprog
$> ./myprog -n 10
```

Jobs, Tasks & Local Execution

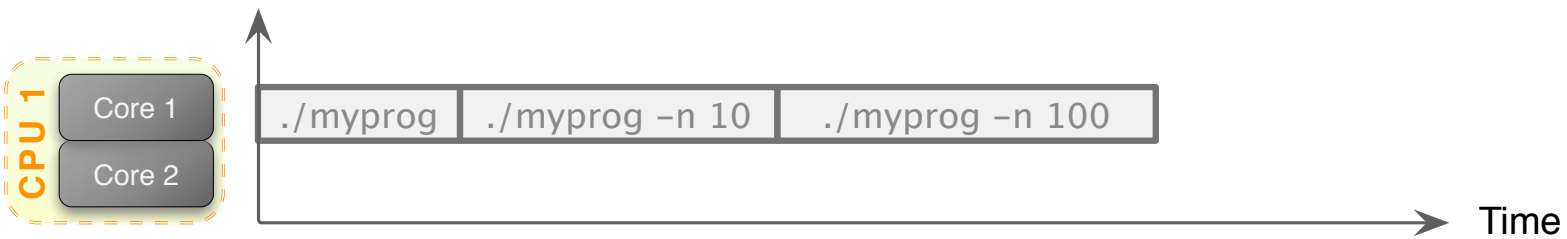
```
$> ./myprog
$> ./myprog -n 10
```

Jobs, Tasks & Local Execution

```
$> ./myprog
$> ./myprog -n 10
$> ./myprog -n 100
```

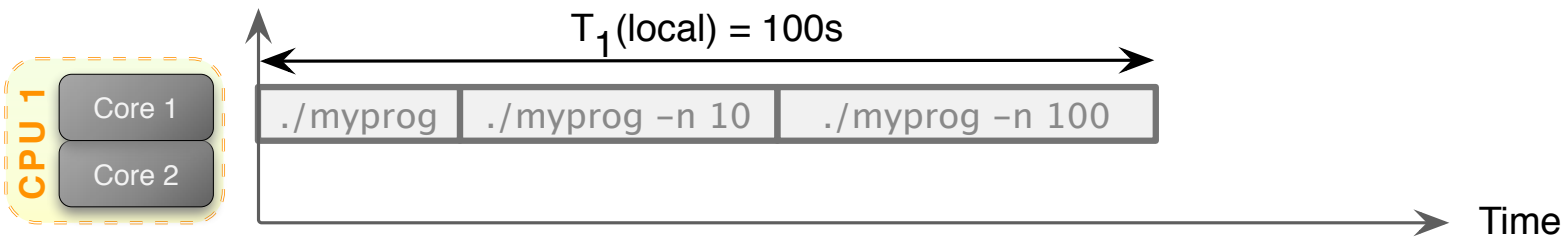
Jobs, Tasks & Local Execution

```
$> ./myprog
$> ./myprog -n 10
$> ./myprog -n 100
```



Jobs, Tasks & Local Execution

```
$> ./myprog
$> ./myprog -n 10
$> ./myprog -n 100
```

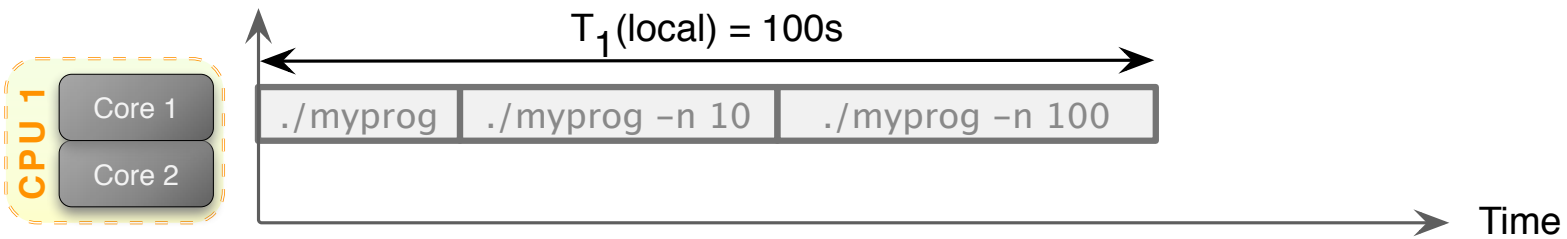


Jobs, Tasks & Local Execution

```
$> ./myprog
$> ./myprog -n 10
$> ./myprog -n 100
```

Job(s) 3

Task(s) 3



Jobs, Tasks & Local Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

Jobs, Tasks & Local Execution

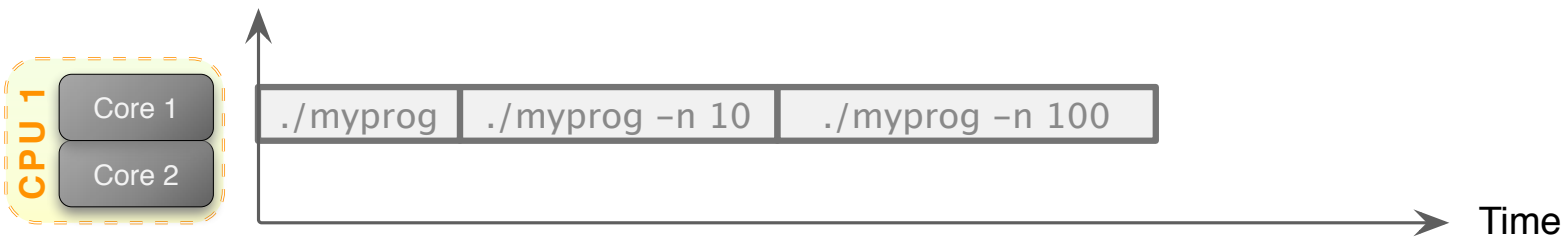
```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

Jobs, Tasks & Local Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

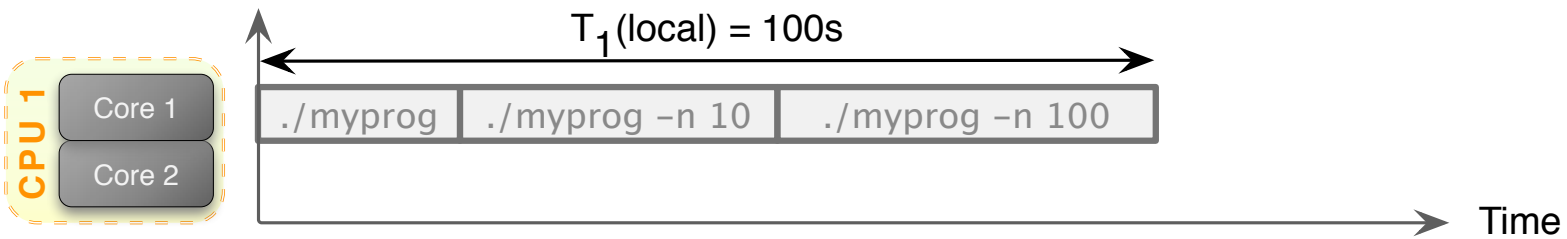
Jobs, Tasks & Local Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```



Jobs, Tasks & Local Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

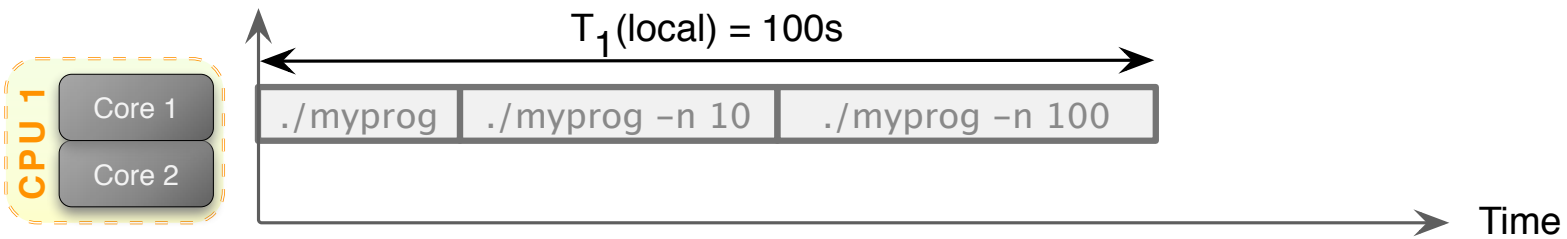


Jobs, Tasks & Local Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

job(s) 1

Task(s) 3



Jobs, Tasks & Local Execution

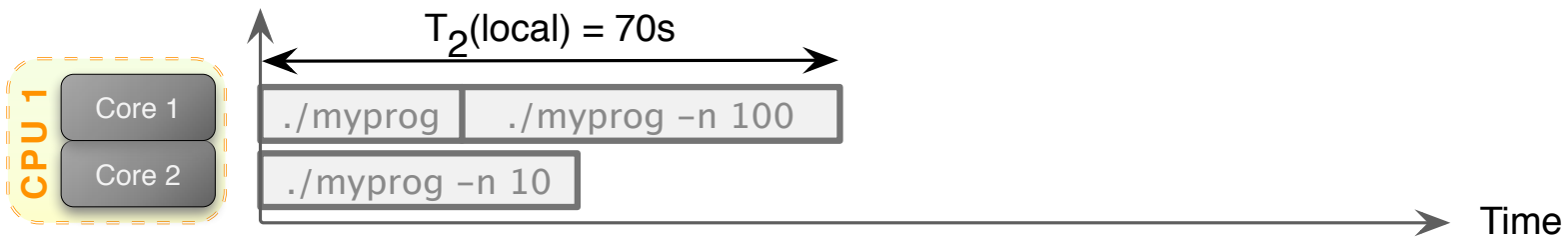
```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

Jobs, Tasks & Local Execution

```
# launcher2
"Run in //:"
 ./myprog
 ./myprog -n 10
 ./myprog -n 100
```

Jobs, Tasks & Local Execution

```
# launcher2
"Run in //:"
 ./myprog
 ./myprog -n 10
 ./myprog -n 100
```

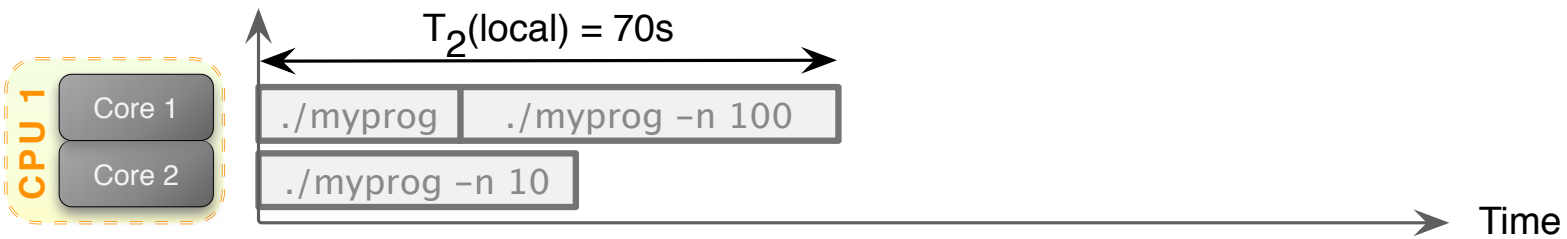


Jobs, Tasks & Local Execution

```
# launcher2
"Run in //:"
 ./myprog
 ./myprog -n 10
 ./myprog -n 100
```

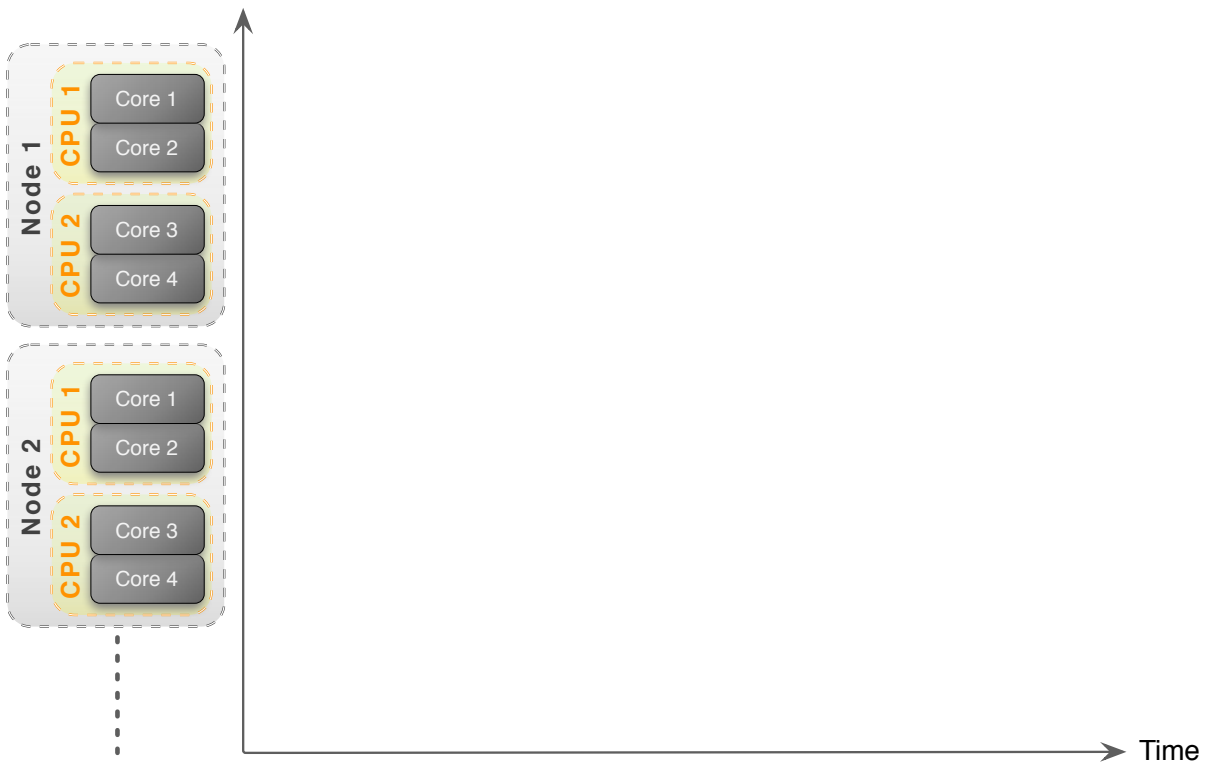
job(s) 1

Task(s) 3



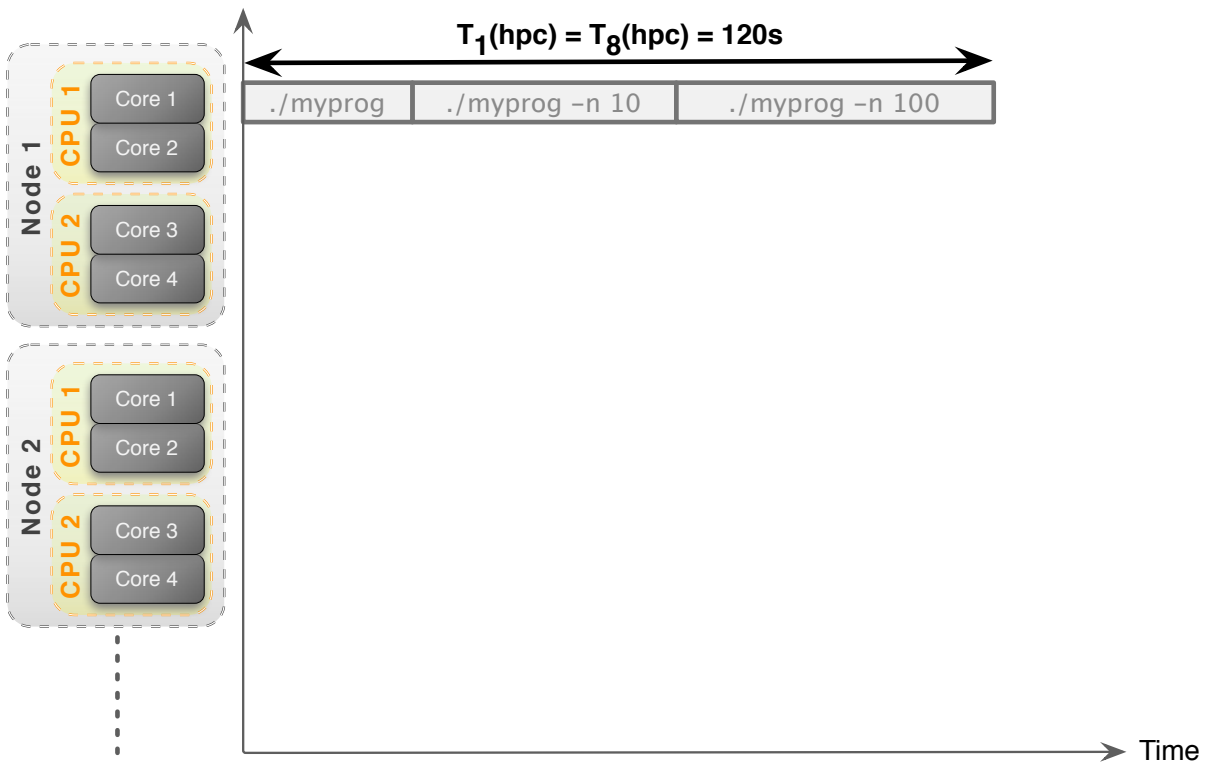
Jobs, Tasks & HPC Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

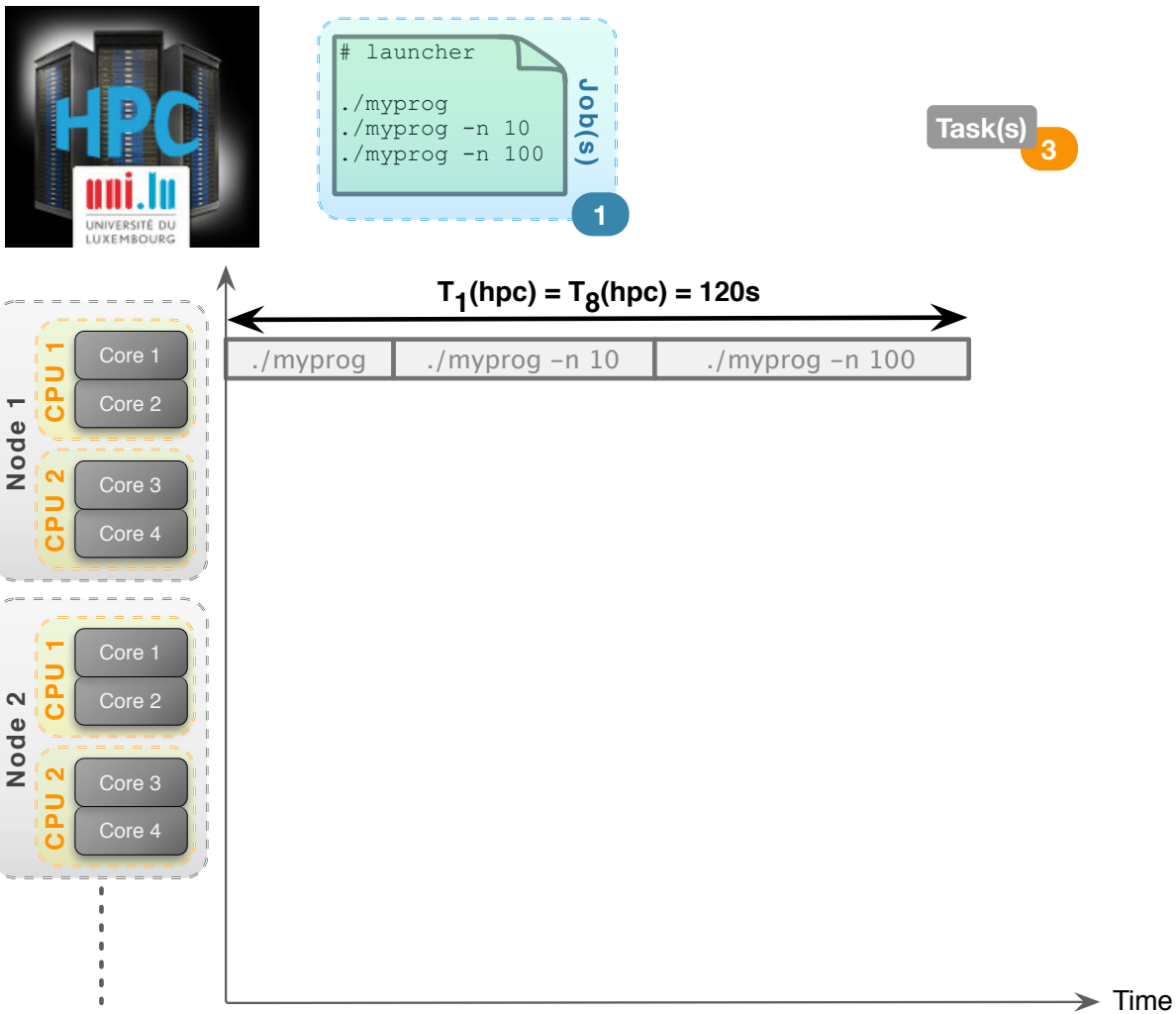


Jobs, Tasks & HPC Execution

```
# launcher

./myprog
./myprog -n 10
./myprog -n 100
```

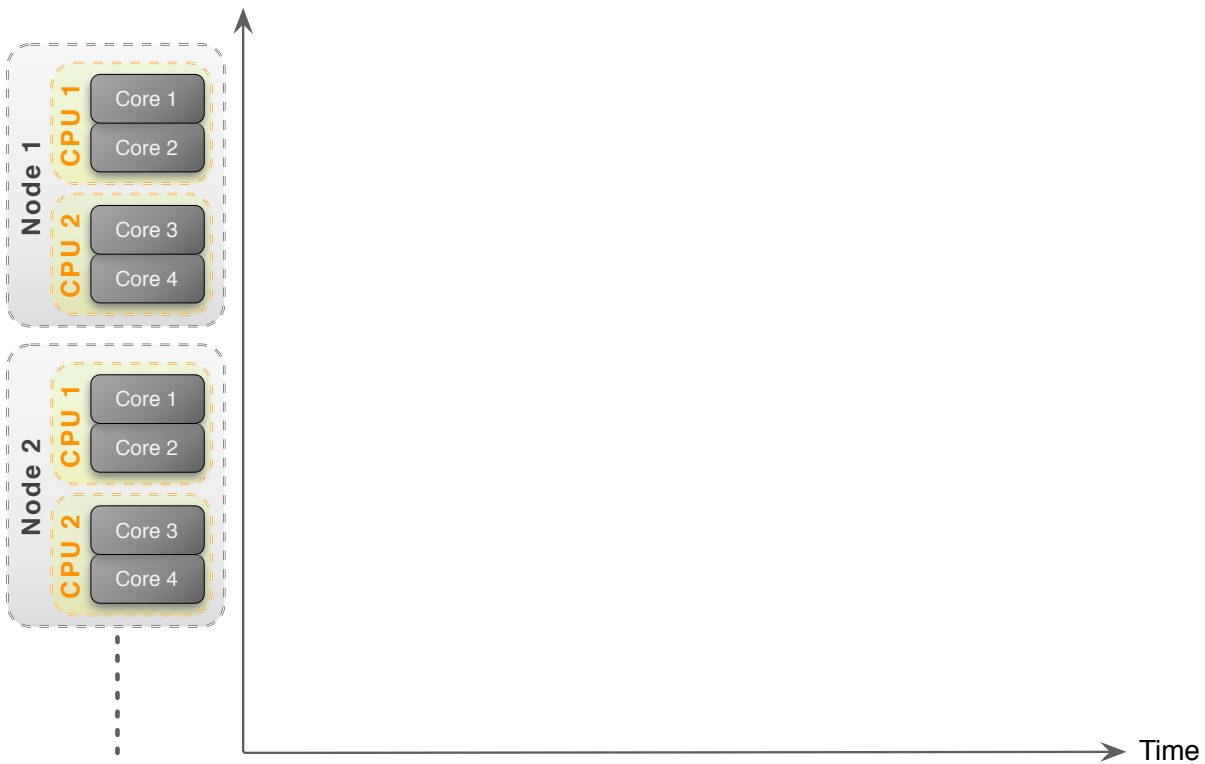


Jobs, Tasks & HPC Execution



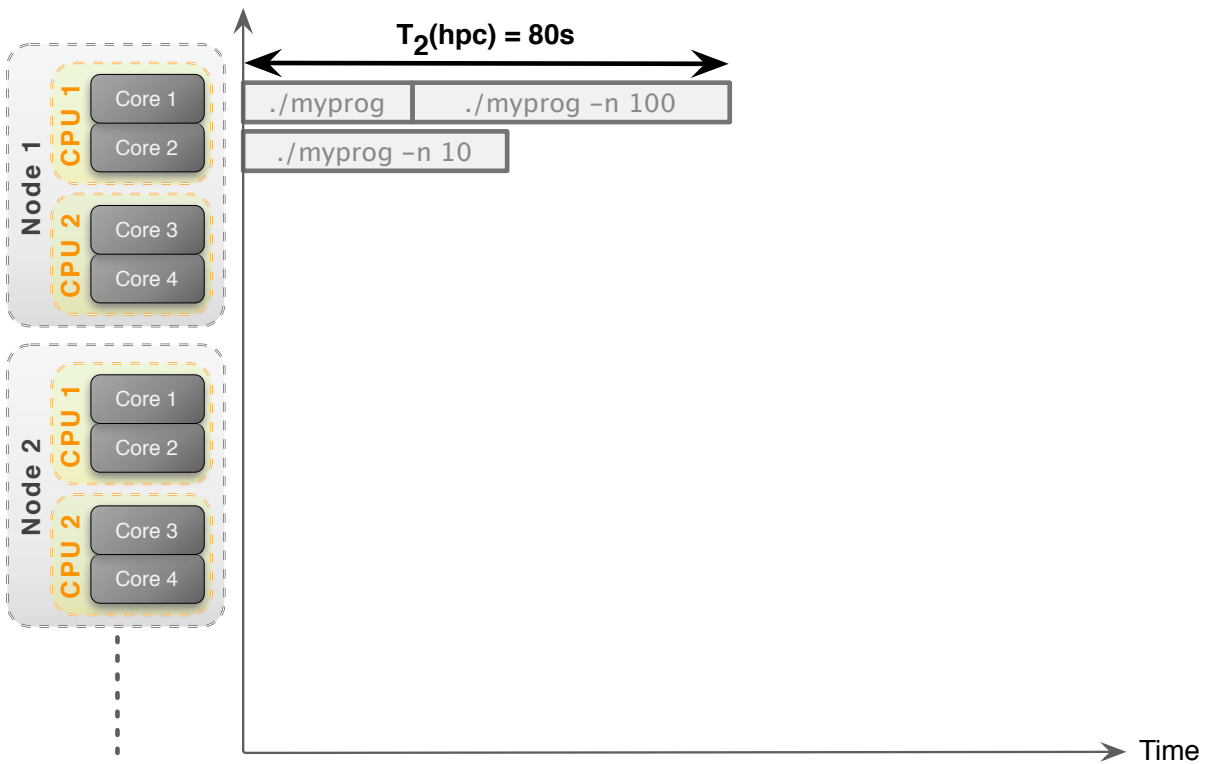
Jobs, Tasks & HPC Execution

```
# launcher2
"Run in //:"
 ./myprog
 ./myprog -n 10
 ./myprog -n 100
```

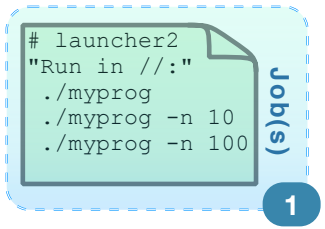


Jobs, Tasks & HPC Execution

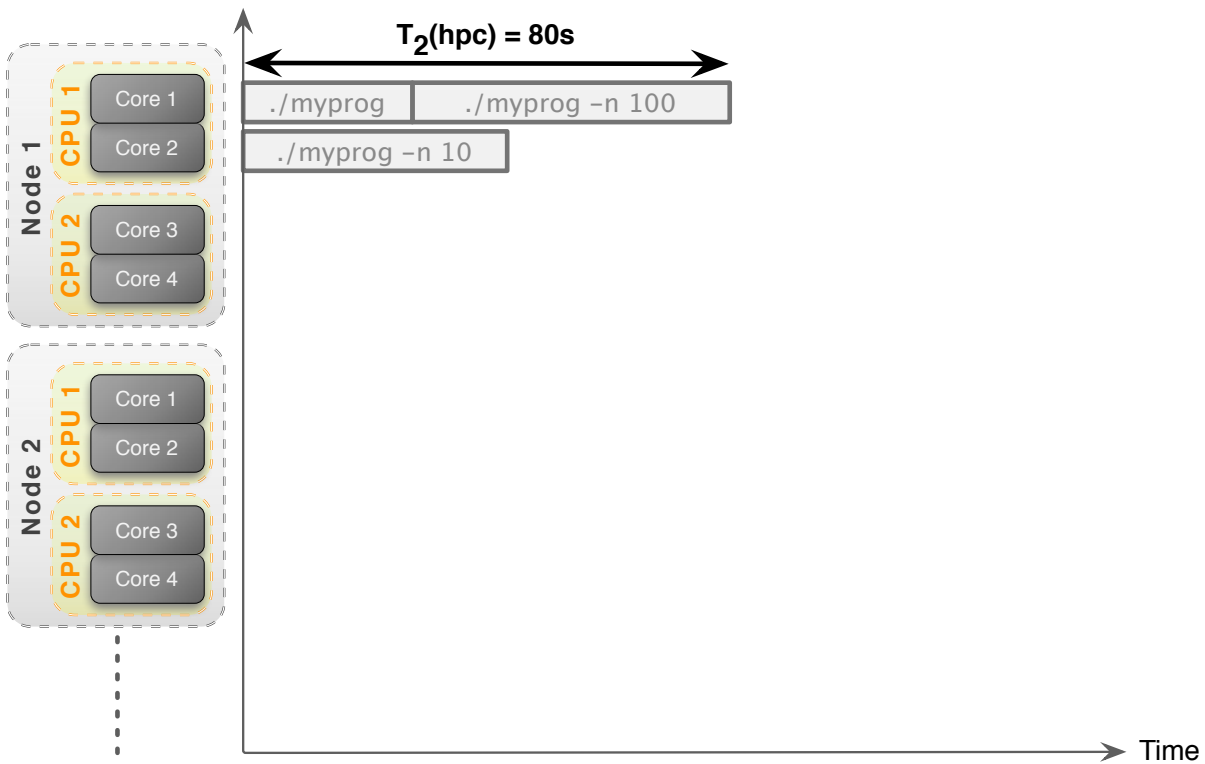
```
# launcher2
"Run in //:"
 ./myprog
 ./myprog -n 10
 ./myprog -n 100
```



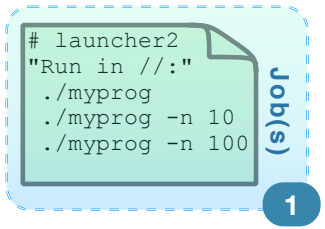
Jobs, Tasks & HPC Execution



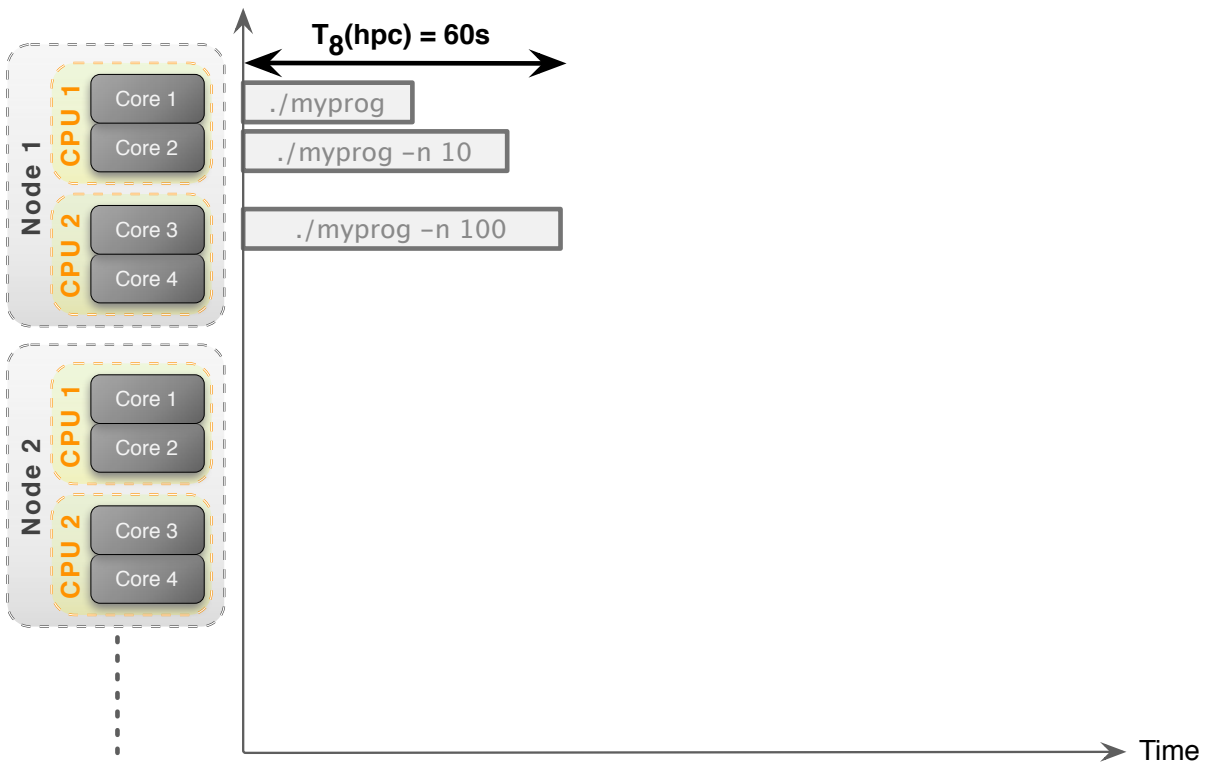
Task(s) 3



Jobs, Tasks & HPC Execution



Task(s) 3



Local vs. HPC Executions

Context	Local PC	HPC
Sequential	$T_1(\text{local}) = 100$	$T_1(\text{hpc}) = \mathbf{120s}$
Parallel/Distributed	$T_2(\text{local}) = 70\text{s}$	$T_2(\text{hpc}) = 80\text{s}$
		$T_8(\text{hpc}) = \mathbf{60s}$

Local vs. HPC Executions

Context	Local PC	HPC
Sequential	$T_1(\text{local}) = 100$	$T_1(\text{hpc}) = \mathbf{120s}$
Parallel/Distributed	$T_2(\text{local}) = 70\text{s}$	$T_2(\text{hpc}) = 80\text{s}$
		$T_8(\text{hpc}) = \mathbf{60s}$

- **Sequential** runs **WON'T BE FASTER** on HPC
  - ↪ Reason: Processor Frequency (typically 3GHz vs 2.26GHz)

Local vs. HPC Executions

Context	Local PC	HPC
Sequential	$T_1(\text{local}) = 100$	$T_1(\text{hpc}) = \mathbf{120s}$
Parallel/Distributed	$T_2(\text{local}) = 70\text{s}$	$T_2(\text{hpc}) = 80\text{s}$
		$T_8(\text{hpc}) = \mathbf{60s}$

- **Sequential** runs **WON'T BE FASTER** on HPC
  - ↪ Reason: Processor Frequency (typically 3GHz vs 2.26GHz)

- **Parallel/Distributed** runs **DO NOT COME FOR FREE**
  - ↪ runs **will be sequential** even if you reserve ≥ 2 cores/nodes
  - ↪ you have to **explicitly** adapt your jobs to benefit from the multi-cores/nodes

Identifying Potential Parallelism

In your workflow

```
$> ./my_sequential_prog -n 1
$> ./my_sequential_prog -n 2
$> ./my_sequential_prog -n 3
$> ./my_sequential_prog -n 4
$> ./my_sequential_prog -n 5
$> ./my_sequential_prog -n 6
$> ./my_sequential_prog -n 7
$> ...
```

Identifying Potential Parallelism

```
x = initX(A, B);
y = initY(A, B);
z = initZ(A, B);

for(i = 0; i < N_ENTRIES; i++)
        x[i] = compX(y[i], z[i]);

for(i = 1; i < N_ENTRIES; i++)
        x[i] = solveX(x[i-1]);

finalize1 (&x, &y, &z);
finalize2 (&x, &y, &z);
finalize3 (&x, &y, &z);
```

Identifying Potential Parallelism

```
x = initX(A, B);
y = initY(A, B);
z = initZ(A, B);
```

Functional Parallelism

Identifying Potential Parallelism

```
x = initX(A, B);
y = initY(A, B);
z = initZ(A, B);
```

Functional Parallelism

```
for(i = 0; i < N_ENTRIES; i++)
        x[i] = compX(y[i], z[i]);
```

Data Parallelism

Identifying Potential Parallelism

```
x = initX(A, B);
y = initY(A, B);
z = initZ(A, B);
```

Functional Parallelism

```
for(i = 0; i < N_ENTRIES; i++)
        x[i] = compX(y[i], z[i]);
```

Data Parallelism

```
for(i = 1; i < N_ENTRIES; i++)
        x[i] = solveX(x[i-1]);
```

Pipelining

Identifying Potential Parallelism

```
x = initX(A, B);
y = initY(A, B);
z = initZ(A, B);
```

Functional Parallelism

```
for(i = 0; i < N_ENTRIES; i++)
        x[i] = compX(y[i], z[i]);
```

Data Parallelism

```
for(i = 1; i < N_ENTRIES; i++)
        x[i] = solveX(x[i-1]);
```

Pipelining

```
finalize1 (&x, &y, &z);
finalize2 (&x, &y, &z);
finalize3 (&x, &y, &z);
```

No good?

Summary

HPC Components: [GP]CPU

CPU

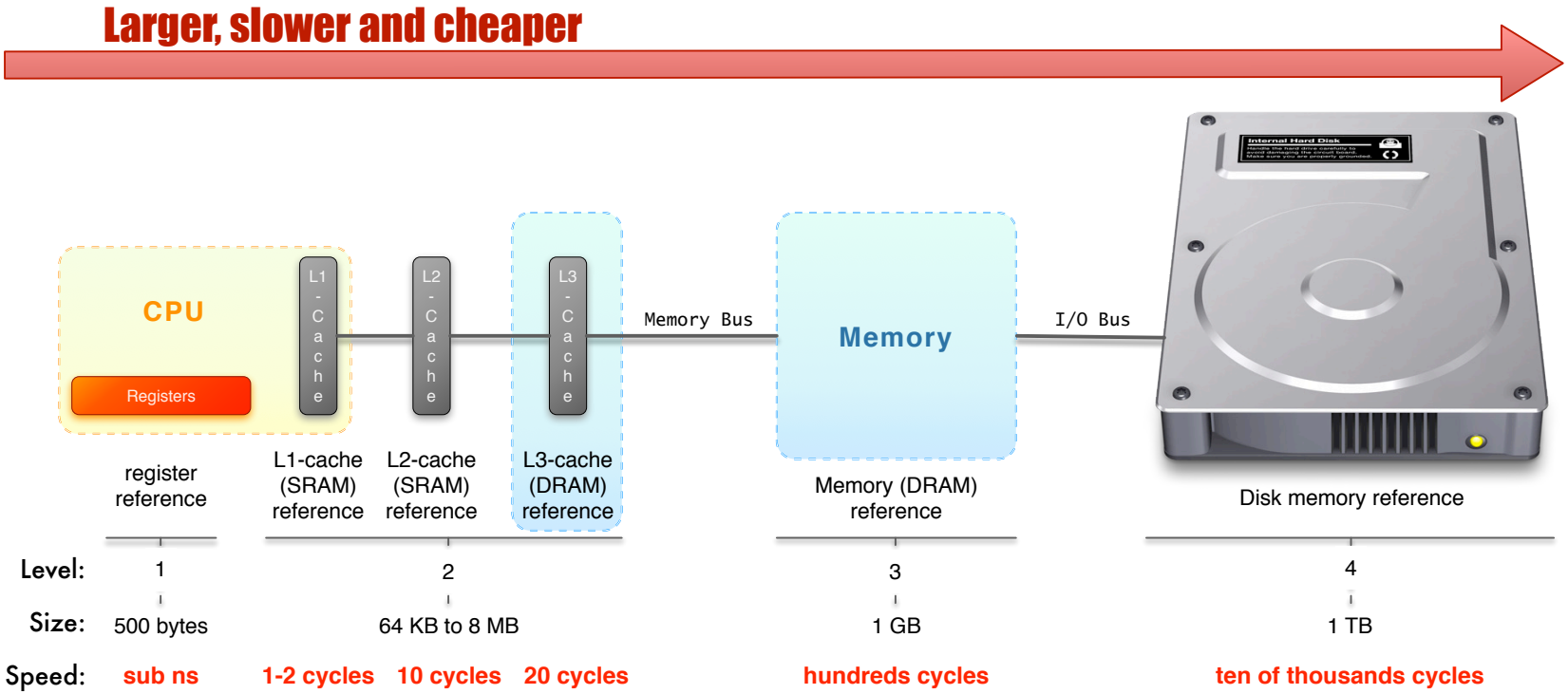
- Always multi-core
- Ex: Intel Core i7-970 (July 2010) $R_{peak} \simeq 100$ GFlops (DP)
  - ↪ 6 cores @ 3.2GHz (32nm, 130W, 1170 millions transistors)

GPU / GPGPU

- Always multi-core, optimized for vector processing
- Ex: Nvidia Tesla C2050 (July 2010) $R_{peak} \simeq 515$ GFlops (DP)
  - ↪ 448 cores @ 1.15GHz

$\simeq$ **10 Gflops** for 50 €

HPC Components: Local Memory

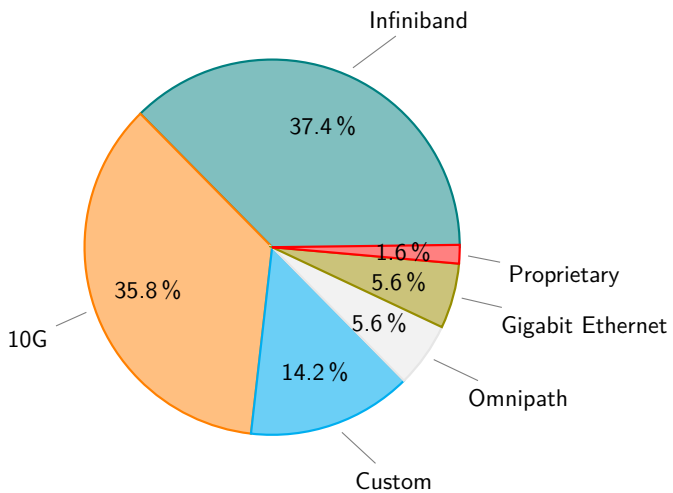


- SSD R/W: 560 MB/s; 85000 IOps **1000 €/TB**
- HDD (SATA @ 7,2 krpm) R/W: 100 MB/s; 190 IOps **100 €/TB**

HPC Components: Interconnect

- **latency**: time to send a minimal (0 byte) message from A to B
- **bandwidth**: max amount of data communicated per unit of time

Technology	Effective Bandwidth		Latency
Gigabit Ethernet	1 Gb/s	125 MB/s	40μs to 300μs
10 Gigabit Ethernet	10 Gb/s	1.25 GB/s	4μs to 5μs
Infiniband QDR	40 Gb/s	5 GB/s	1.29μs to 2.6μs
Infiniband EDR	100 Gb/s	12.5 GB/s	0.61μs to 1.3μs
100 Gigabit Ethernet	100 Gb/s	1.25 GB/s	30μs
Intel Omnipath	100 Gb/s	12.5 GB/s	0.9μs

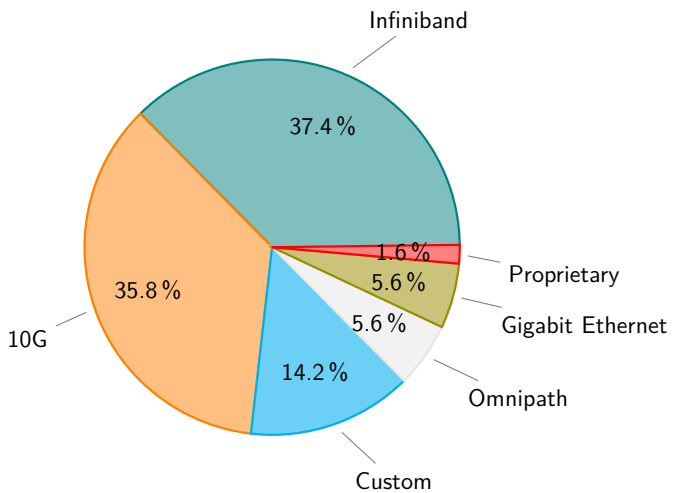


[Source : www.top500.org, Nov. 2016]

HPC Components: Interconnect

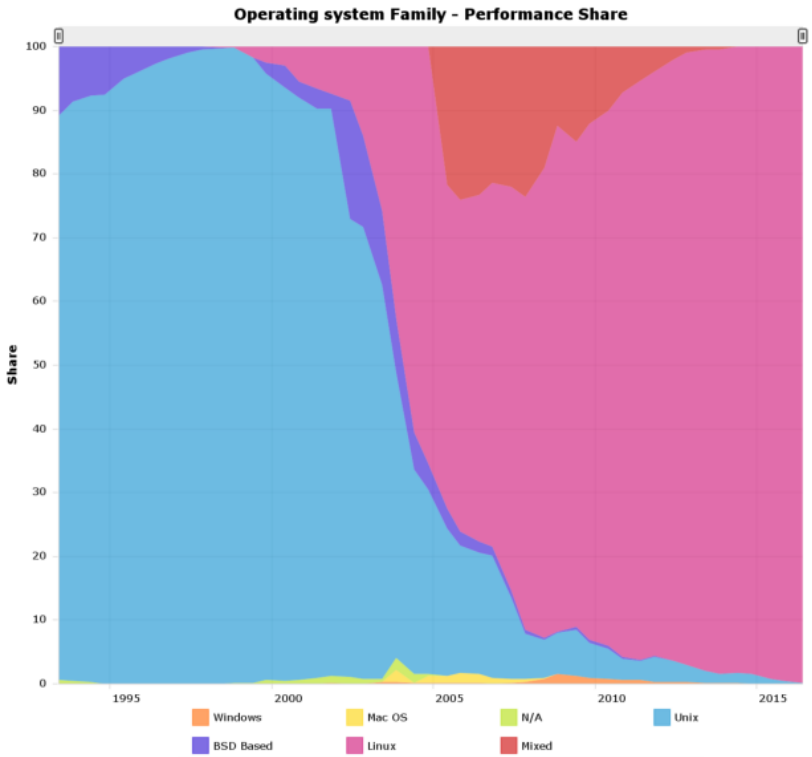
- **latency**: time to send a minimal (0 byte) message from A to B
- **bandwidth**: max amount of data communicated per unit of time

Technology	Effective Bandwidth		Latency
Gigabit Ethernet	1 Gb/s	125 MB/s	40μs to 300μs
10 Gigabit Ethernet	10 Gb/s	1.25 GB/s	4μs to 5μs
Infiniband QDR	40 Gb/s	5 GB/s	1.29μs to 2.6μs
Infiniband EDR	100 Gb/s	12.5 GB/s	0.61μs to 1.3μs
100 Gigabit Ethernet	100 Gb/s	1.25 GB/s	30μs
Intel Omnipath	100 Gb/s	12.5 GB/s	0.9μs



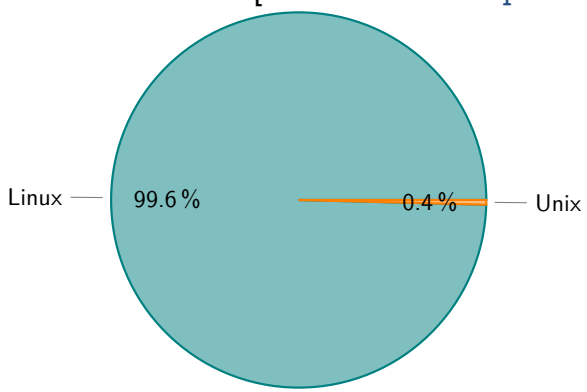
[Source : www.top500.org, Nov. 2016]

HPC Components: Operating System

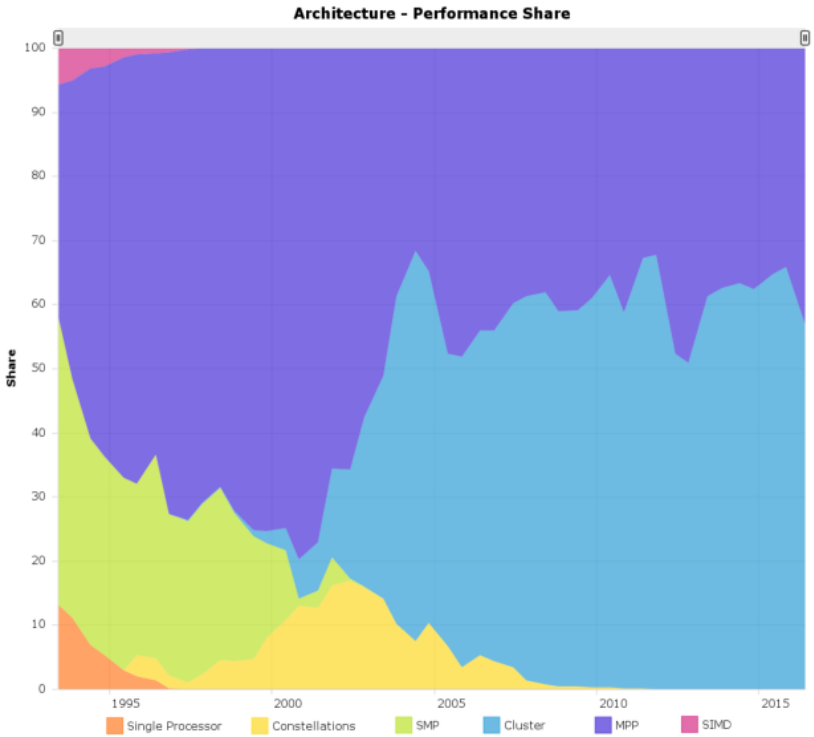


- Exclusively Linux-based (99.6%)
  - ↪ . . . or Unix (0.4%)
- Reasons:
  - ↪ stability
  - ↪ prone to devels

[Source : www.top500.org, Nov 2016]

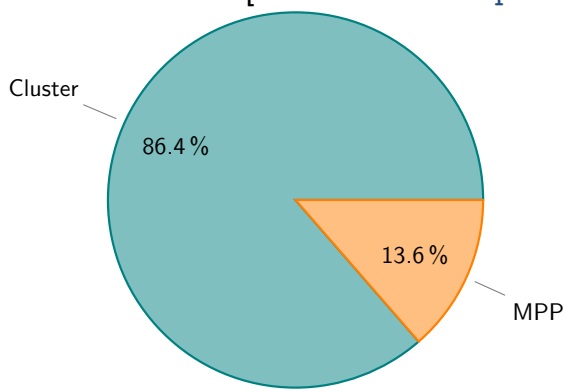


HPC Components: Architecture



- Mainly Cluster-based (86.4%)
  - ↪ . . . or MPP (13.6%)
- Reasons:
  - ↪ scalable
  - ↪ cost-effective

[Source : www.top500.org, Nov 2016]

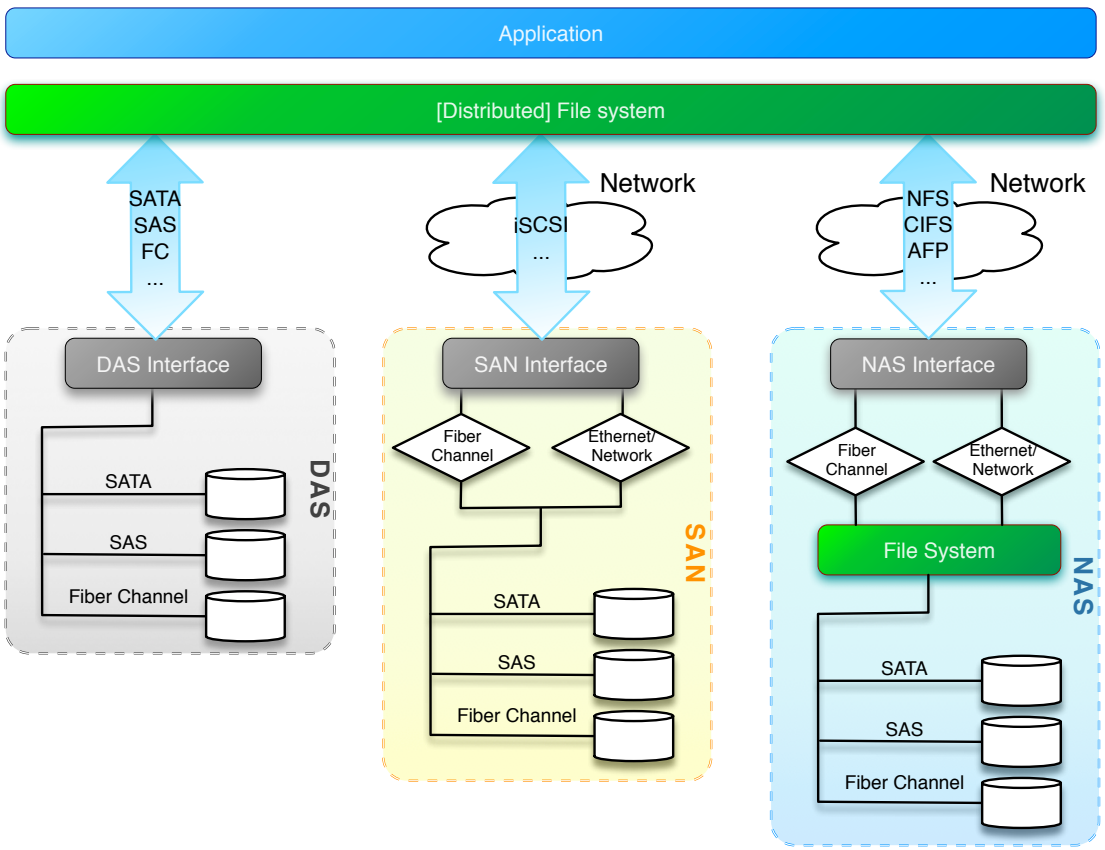


HPC Components: Software Stack

- **Remote connection to the platform** SSH
- **Identity Management / SSO**: LDAP, Kerberos, IPA...
- **Resource management**: job/batch scheduler
  - ↪ SLURM, OAR, PBS, MOAB/Torque...
- (Automatic) **Node Deployment**:
  - ↪ FAI, Kickstart, Puppet, Chef, Ansible, Kadeploy...
- (Automatic) **User Software Management**:
  - ↪ Easybuild, Environment Modules, LMod
- **Platform Monitoring**:
  - ↪ Nagios, Icinga, Ganglia, Foreman, Cacti, Alerta...

[Big]Data Management

Storage architectural classes & I/O layers



[Big]Data Management: Disk Encl.

- $\simeq$ **120 K€** / enclosure – 48-60 disks (4U)
  - ↪ incl. redundant (**i.e.** 2) RAID controllers (master/slave)

[Big]Data Management: File Systems

File System (FS)

- Logical manner to **store**, **organize**, **manipulate** & **access** data

[Big]Data Management: File Systems

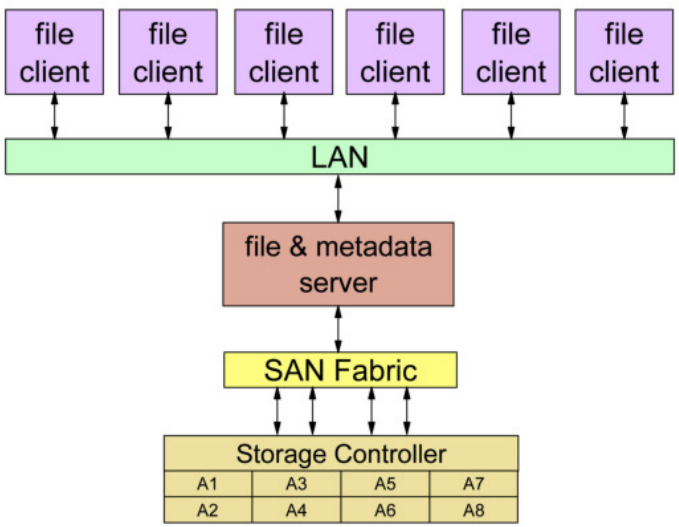
File System (FS)

- Logical manner to **store**, **organize**, **manipulate** & **access** data

- (local) **Disk FS** : `FAT32`, `NTFS`, `HFS+`, `ext{3,4}`, `{x,z,btr}fs`...
  - ↪ manage data on permanent storage devices
  - ↪ 'poor' perf. **read**: 100 → 400 MB/s | **write**: 10 → 200 MB/s

[Big]Data Management: File Systems

- **Networked FS**: `NFS, CIFS/SMB, AFP`
  - ↪ disk access from remote nodes via network access
  - ↪ poorer performance for HPC jobs especially parallel I/O
    - ✓ **read**: only 381 MB/s on a system capable of 740MB/s (16 tasks)
    - ✓ **write**: only 90MB/s on system capable of 400MB/s (4 tasks)

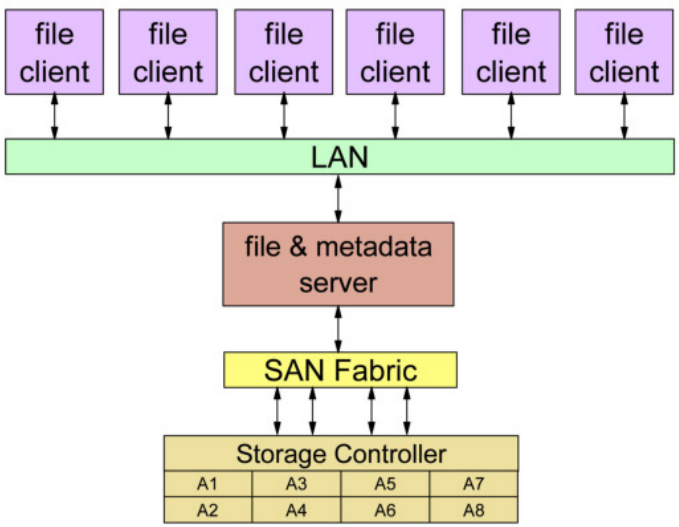


[Source : LISA'09] Ray Paden: *How to Build a Petabyte Sized Storage System*

COMMENT:
Traditionally, a single NFS/CIFS file server manages both user data and metadata operations which "gates" performance/scaling and presents a single point of failure risk. Products (e.g., CNFS) are available that provide multiple server designs to avoid this issue.

[Big]Data Management: File Systems

- **Networked FS**: `NFS, CIFS/SMB, AFP`
  - ↪ disk access from remote nodes via network access
  - ↪ poorer performance for HPC jobs especially parallel I/O
    - ✓ **read**: only 381 MB/s on a system capable of 740MB/s (16 tasks)
    - ✓ **write**: only 90MB/s on system capable of 400MB/s (4 tasks)



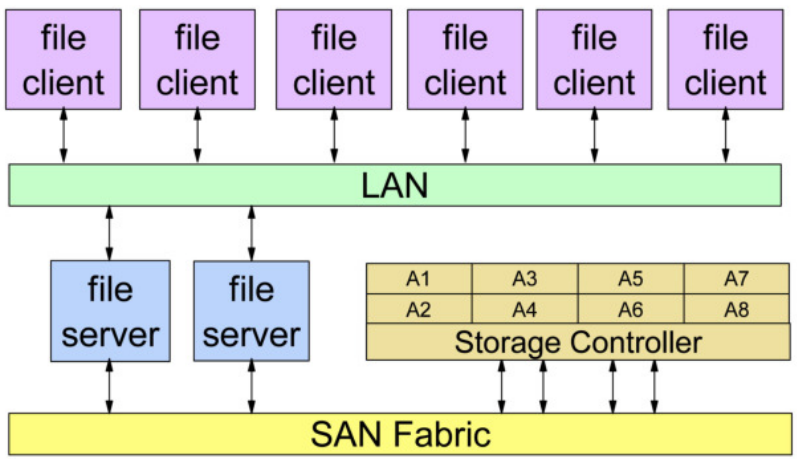
COMMENT:
Traditionally, a single NFS/CIFS file server manages both user data and metadata operations which "gates" performance/scaling and presents a single point of failure risk. Products (e.g., CNFS) are available that provide multiple server designs to avoid this issue.

[Source : LISA'09] Ray Paden: *How to Build a Petabyte Sized Storage System*

- [scale-out] **NAS**
  - ↪ aka Appliances `OneFS...`
  - ↪ Focus on CIFS, NFS
  - ↪ Integrated HW/SW
  - ↪ **Ex**: **EMC (Isilon)**, IBM (SONAS), **DDN**...

[Big]Data Management: File Systems

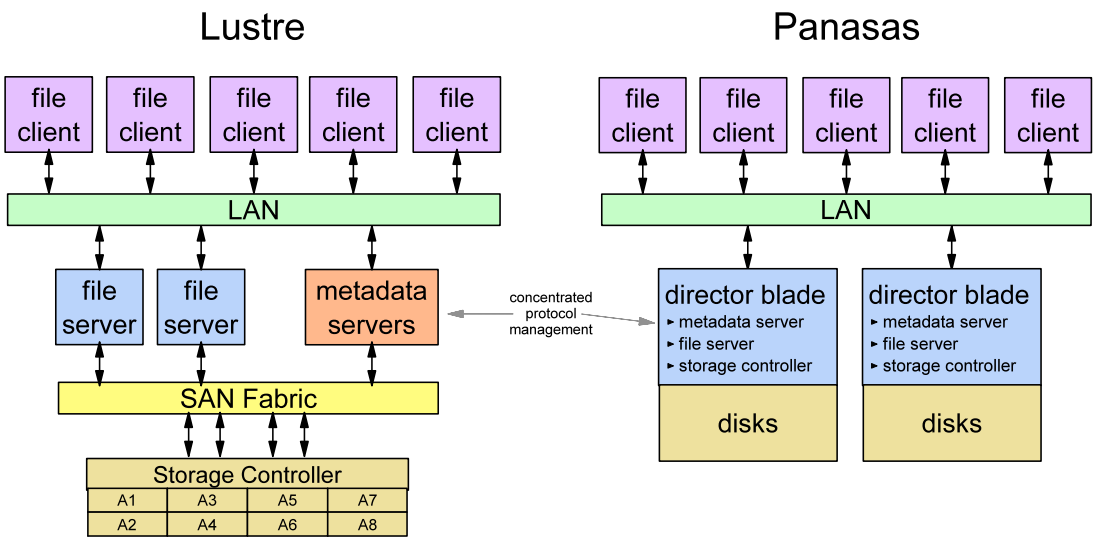
- **Basic Clustered FS** GPFS
  - ↪ File access is parallel
  - ↪ File System overhead operations is distributed and done in parallel
    - ✓ **no** metadata servers
  - ↪ File clients access file data through file servers via the LAN



File system overhead operations are *distributed* across the entire cluster and is done in parallel; it is not concentrated in any given place. There is no single server bottleneck. User data and metadata flows betweem all nodes and all disks via the file servers.

[Big]Data Management: File Systems

- **Multi-Component Clustered FS** Lustre, Panasas
  - ↪ File access is parallel
  - ↪ File System overhead operations on dedicated components
    - ✓ metadata server (Lustre) or director blades (Panasas)
  - ↪ Multi-component architecture
  - ↪ File clients access file data through file servers via the LAN



[Big]Data Management: FS Summary

- **{ Basic | Multi-Component } Clustered FS**
  ≃ Parallel/Distributed FS: `GPFS, Lustre`
  - ↪ for Input/Output (I/O)-intensive HPC systems
  - ↪ data are stripped over multiple servers for high performance
  - ↪ generally add robust failover and recovery mechanisms

Main Characteristic of Parallel/Distributed File Systems

capacity and performance increase with #servers

Name	Type	Read* [GB/s]	Write* [GB/s]
`ext4`	Disk FS	0.426	0.212
`nfs`	Networked FS	0.381	0.090
`gpfs`	Parallel/Distributed FS	7.74	6.524
`lustre`	Parallel/Distributed FS	4.5	2.956

* maximum **random** read/write, per IOZone or IOR measures, using 15 concurrent nodes for networked FS. Measured performed on the UL HPC facility in Jan. 2015

HPC Components: Data Center

Definition (Data Center)

Facility to house computer systems and associated components

↪ Basic storage component: **rack** (height: 42 RU)

HPC Components: Data Center

Definition (Data Center)

Facility to house computer systems and associated components

↪ Basic storage component: **rack** (height: 42 RU)

Challenges: Power (UPS, battery), Cooling, Fire protection, Security

- Power/Heat dissipation per rack:
  - ↪ 'HPC' (computing) racks: 30 to 90 kW
  - ↪ 'Storage' racks: 15 kW
  - ↪ 'Interconnect' racks: 5 kW

Power Usage Effectiveness

$$PUE = \frac{Total\ facility\ power}{IT\ equipment\ power}$$

HPC Components: Data Center

HPC Components: Summary

Running an HPC Facility involves...

- A **data center** / server room carefully designed
- Many **computing** elements: CPU, GPGPU, Accelerators
- **Fast interconnect** elements
  - ↪ high *bandwidth* and low *latency*
- [Big]-Data **storage** elements: HDD/SDD, disk enclosure,
  - ↪ disks are virtually aggregated by RAID/LUNs/FS
  - ↪ parallel and distributed FS
- A flexible software stack
- Automated management everywhere

Above all: **expert** system administrators !

Summary

SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase

SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase

- SSH is used as a secure backbone channel for **many** tools
  - ↪ Remote shell **i.e** remote command line
  - ↪ File transfer: `rsync`, `scp`, `sftp`
  - ↪ versionning synchronization (`svn`, `git`), github, gitlab etc.

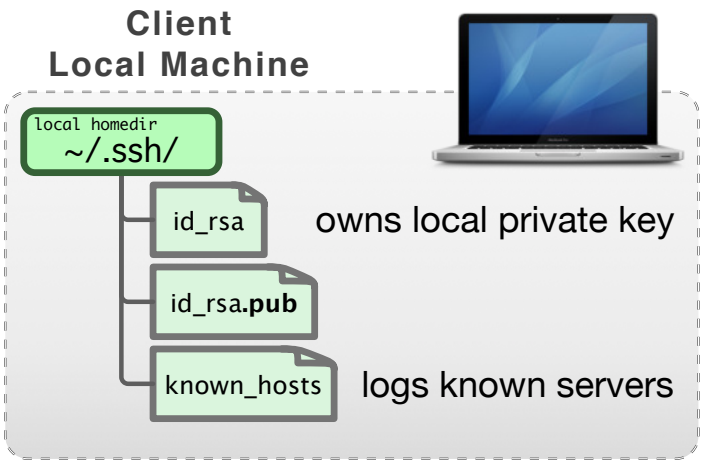
SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase

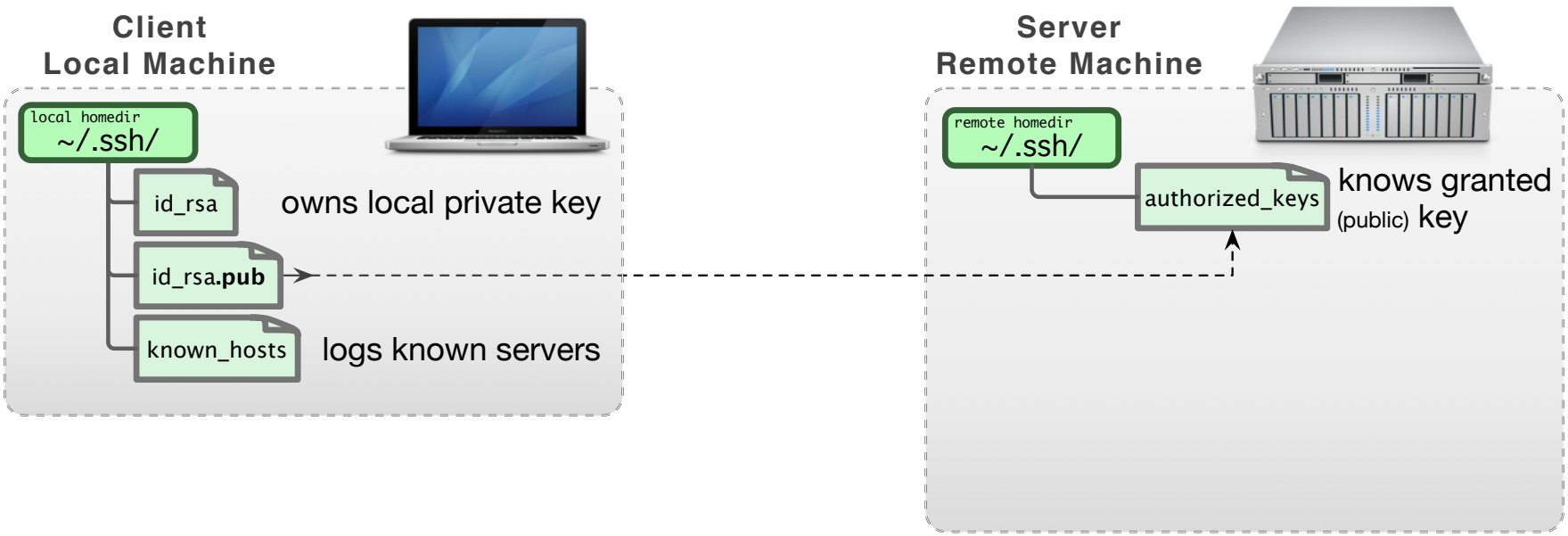
- SSH is used as a secure backbone channel for **many** tools
  - ↪ Remote shell **i.e** remote command line
  - ↪ File transfer: `rsync`, `scp`, `sftp`
  - ↪ versionning synchronization (`svn`, `git`), github, gitlab etc.

- Authentication:
  - ↪ ~~password~~ (disable if possible)
  - ↪ (*better*) **public key authentication**

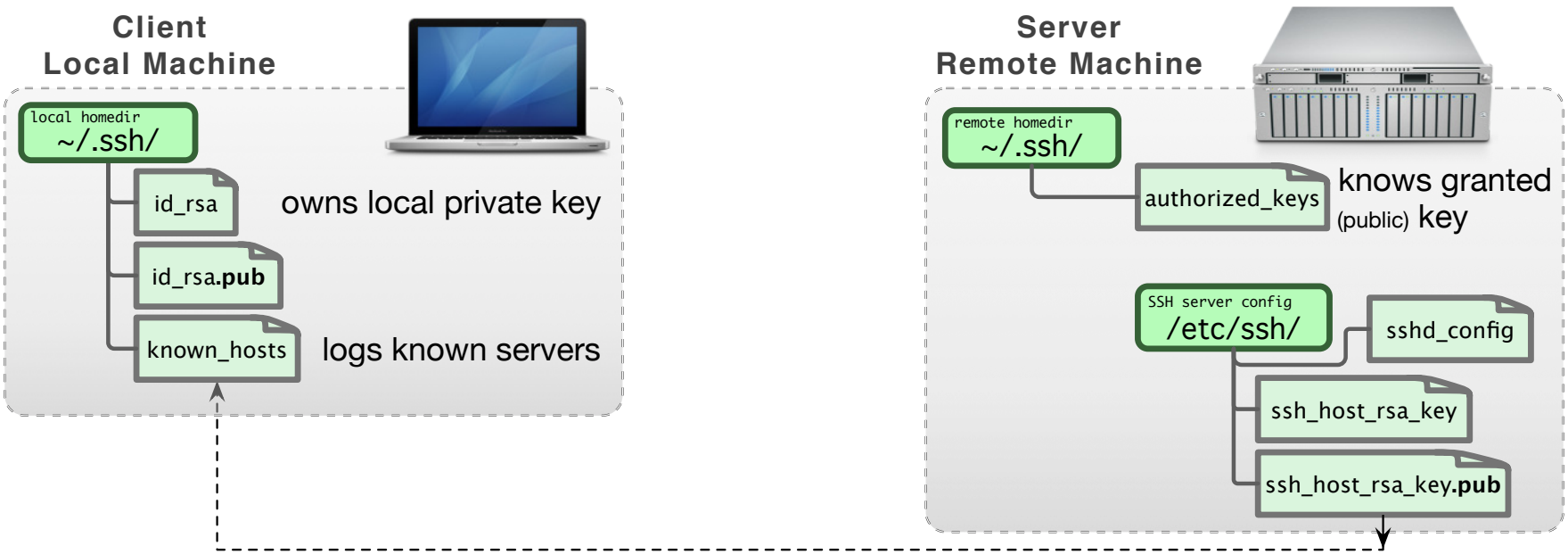
SSH: Public Key Authentication



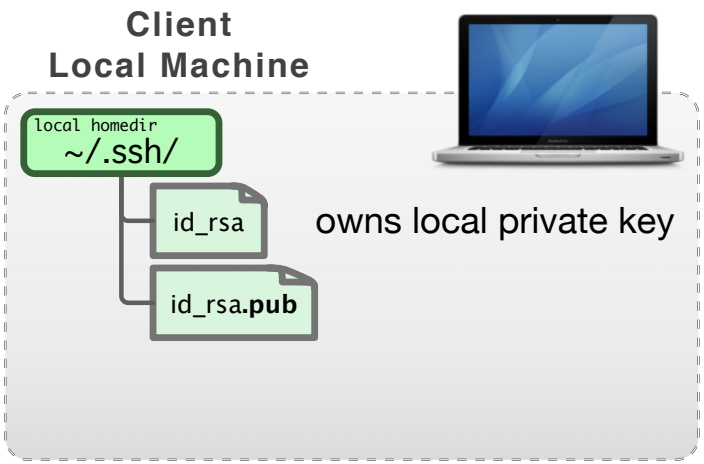
SSH: Public Key Authentication

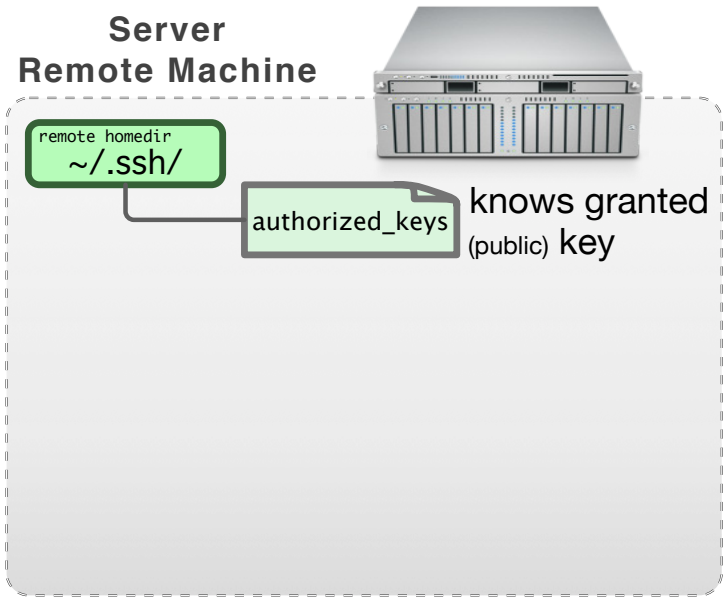


SSH: Public Key Authentication

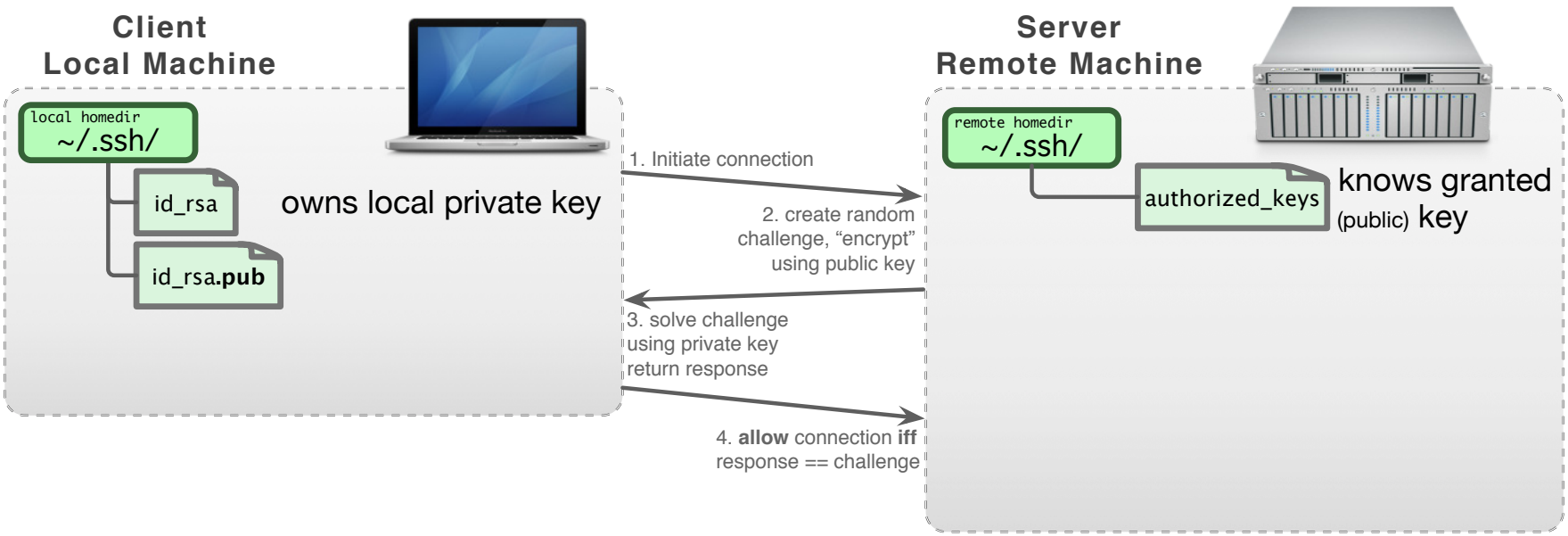


SSH: Public Key Authentication





SSH: Public Key Authentication



- **Restrict** to public key authentication: `/etc/ssh/sshd_config`:

```
PermitRootLogin no
# Disable Passwords
PasswordAuthentication no
ChallengeResponseAuthentication no
```

```
# Enable Public key auth.
RSAAuthentication yes
PubkeyAuthentication yes
```

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ **drawback:** passphrase must be typed to use your key

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

DSA and RSA 1024 bit are deprecated now!

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

DSA and RSA 1024 bit are deprecated now!

```
$> ssh-keygen -t rsa -b 4096 -o -a 100          # 4096 bits RSA
(better)  $> ssh-keygen -t ed25519 -o -a 100    # new sexy Ed25519
```

Private (identity) key

`~/.ssh/id_{rsa,ed25519}`

Public Key

`~/.ssh/id_{rsa,ed25519}.pub`

SSH Setup on Windows

- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI
  - ↪ PuTTYgen, an RSA and DSA key generation utility

SSH Setup on Windows

- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI
  - ↪ PuTTYgen, an RSA and DSA key generation utility

PuTTY ≠ OpenSSH

SSH Setup on Windows

- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI
  - ↪ PuTTYgen, an RSA and DSA key generation utility

PuTTY ≠ OpenSSH

- Putty keys are **NOT** supported by OpenSSH (yet can be exported)
- Binding Pageant with OpenSSH agent is **NOT** natively supported
  - ↪ Third-party tools like ssh-pageant are made for that
  - ↪ Combine nicely with Git bash https://git-for-windows.github.io/
- with PLink, hostnames eventually refer to **PuTTY Sessions**
  - ↪ **NEVER** to SSH entries in `~/.ssh/config`
  - ↪ This usage might be hidden... Ex: `$GIT_SSH` etc.

SSH in Practice ~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-chaos.uni.lu
```

```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion

    $> ssh cha<TAB>

SSH in Practice ~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-chaos.uni.lu
```

```
Host work
    User          localuser
    Hostname      myworkstation.uni.lux
Host *.ext_ul
    ProxyCommand ssh -q chaos-cluster \
                 "nc -q 0 %h %p"
# UL HPC Platform -- http://hpc.uni.lu
Host chaos-cluster
    Hostname      access-chaos.uni.lu
Host gaia-cluster
    Hostname      access-gaia.uni.lu
Host *-cluster
    User          login #ADAPT accordingly
    Port          8022
    ForwardAgent no
```

```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion

    $> ssh cha<TAB>

SSH in Practice ~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-chaos.uni.lu
```

```
Host work
    User          localuser
    Hostname      myworkstation.uni.lux
Host *.ext_ul
    ProxyCommand ssh -q chaos-cluster \
                 "nc -q 0 %h %p"
# UL HPC Platform -- http://hpc.uni.lu
Host chaos-cluster
    Hostname      access-chaos.uni.lu
Host gaia-cluster
    Hostname      access-gaia.uni.lu
Host *-cluster
    User          login #ADAPT accordingly
    Port          8022
    ForwardAgent no
```

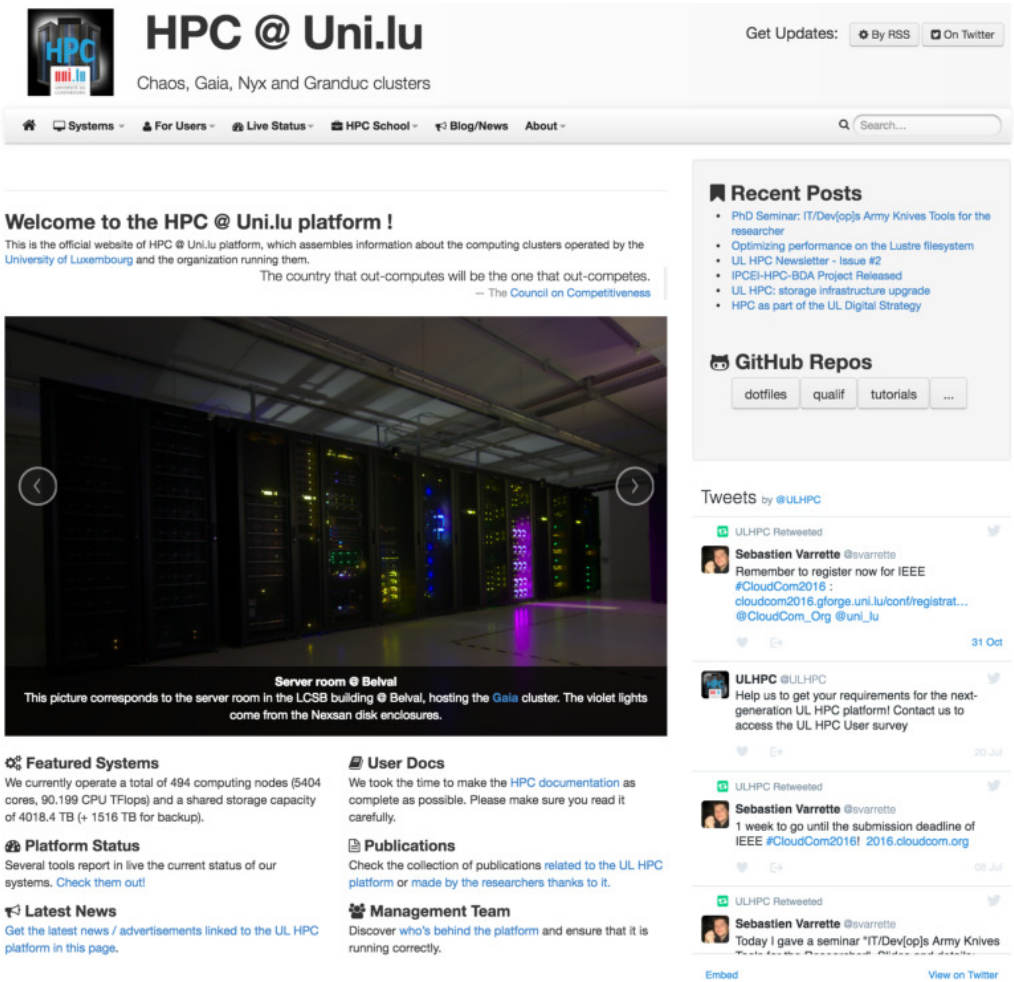
```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion
    $> ssh cha<TAB>

```
$> ssh chaos-cluster
$> ssh work
$> ssh work.ext_ul
```

Summary

High Performance Computing @ UL

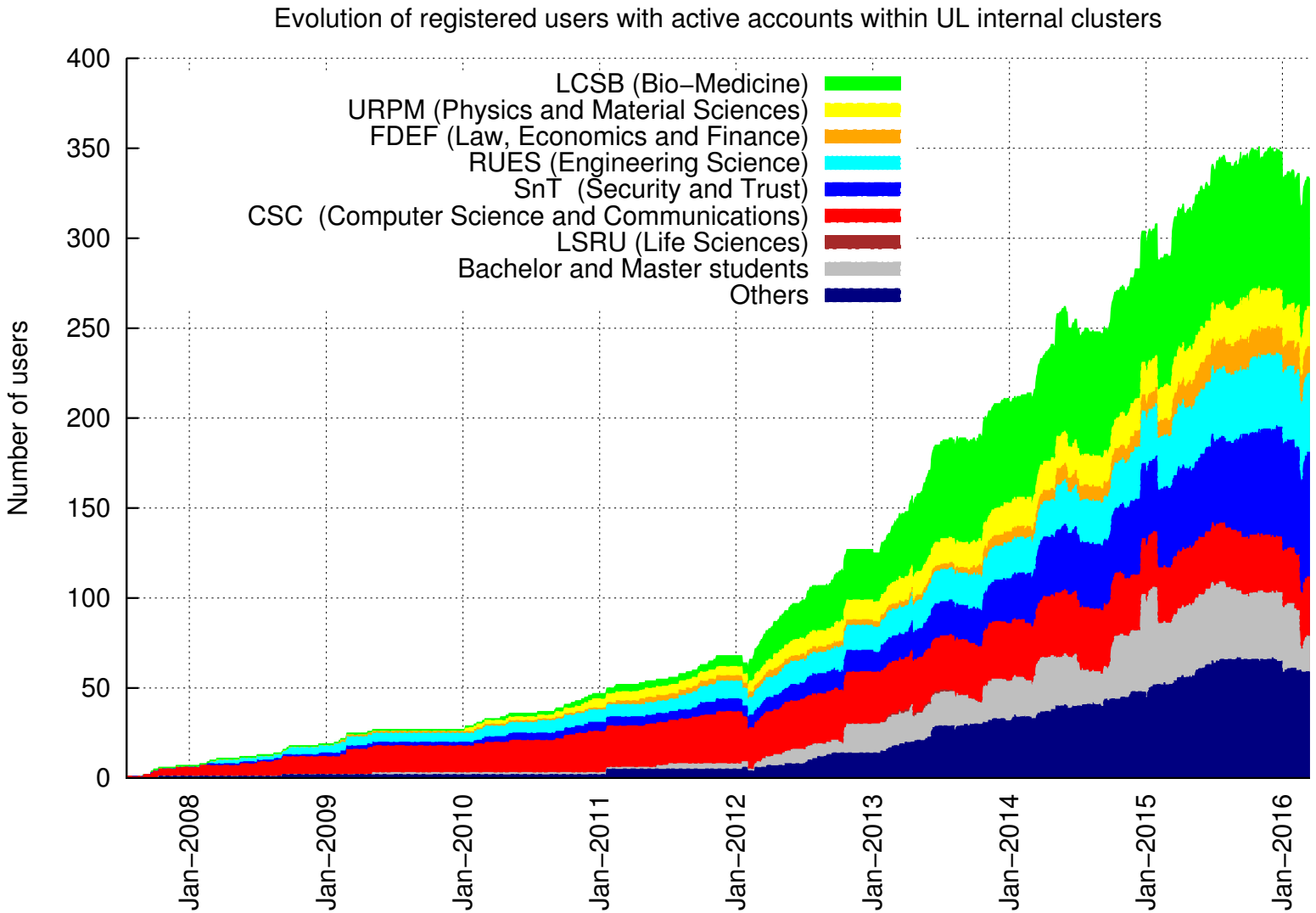


Key numbers

- 344 users
- 98 servers
- 522 nodes
  - ↪ 5420 cores
  - ↪ **85.543 TFlops**
- **5354.4 TB**
- 4 sysadmins
- 2 sites
  - ↪ Kirchberg
  - ↪ Belval

http://hpc.uni.lu

High Performance Computing @ UL



High Performance Computing @ UL

- **Enables & accelerates** scientific discovery and innovation
- **Largest facility** in Luxembourg (after GoodYear R&D Center)

			(CPU)	TFlops	TB (Shared)
Country	**Institute**	**#Nodes**	**#Cores**	**R_{peak}**	**Storage**
Luxembourg	**UL HPC** (Uni.lu)	**522**	**5420**	**85.543**	**5354.4**
	LIST	58	800	6.21	144
France	LORIA (G5K), Nancy	320	2520	26.98	82
	ROMEO, Reims	174	3136	49.26	245
Belgium	NIC4, University of Liège	128	2048	32.00	20
	Université Catholique de Louvain	112	1344	13.28	120
	UGent / VSC, Gent	440	8768	275.30	1122
Germany	bwGrid, Heidelberg	140	1120	12.38	32
	bwForCluster, Ulm	444	7104	266.40	400
	bwHPC MLS&WISO, Mannheim	604	9728	371.60	420

UL HPC Team

Prof. Pascal Bouvry
Head of ILIAS Laboratory, Director of DS-CSCE, Leader of PCO Group
Senior advisor for the president as regards the HPC strategy

Sébastien Varrette, PhD
CDI, Research Scientist (CSC, FSTC)

Valentin Plugaru, MSc.
CDI, Research Collaborator (CSC, FSTC)

Hyacinthe Cartiaux
CDI, Support (SIU)

Sarah Diehl, MSc.
CDD, Research Associate (LCSB)

UL HPC Services

Horizontal HPC & storage services

- for the three UL Faculties and their Research Units
- for the two UL Inter-disciplinary Centres
  - ↪ LCSB, SnT
- . . . and their external partners
- on UL **strategic research priorities**
  - ↪ computational sciences
  - ↪ systems biomedicine
  - ↪ security, reliability and trust
  - ↪ finance

UL HPC Services

Horizontal HPC & storage services

- for the three UL Faculties and their Research Units
- for the two UL Inter-disciplinary Centres
  - ↪ LCSB, SnT
- . . . and their external partners
- on UL **strategic research priorities**
  - ↪ computational sciences
  - ↪ systems biomedicine
  - ↪ security, reliability and trust
  - ↪ finance

- **Complementary research related services** **Total**: 80 servers
  - ↪ On demand VM hosting for development, frontends, etc.
  - ↪ Project management & collaboration (GForge, GitLab. . .)
  - ↪ Cloud storage (OwnCloud) . . . and many others!

Sites / Data centers

Kirchberg

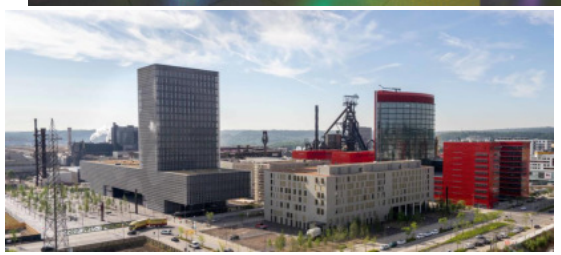
CS.43, AS. 28

Belval

Biotech I, CDC/MSA

2 sites, $\geq$ 4 server rooms

Sites / Data centers



Kirchberg

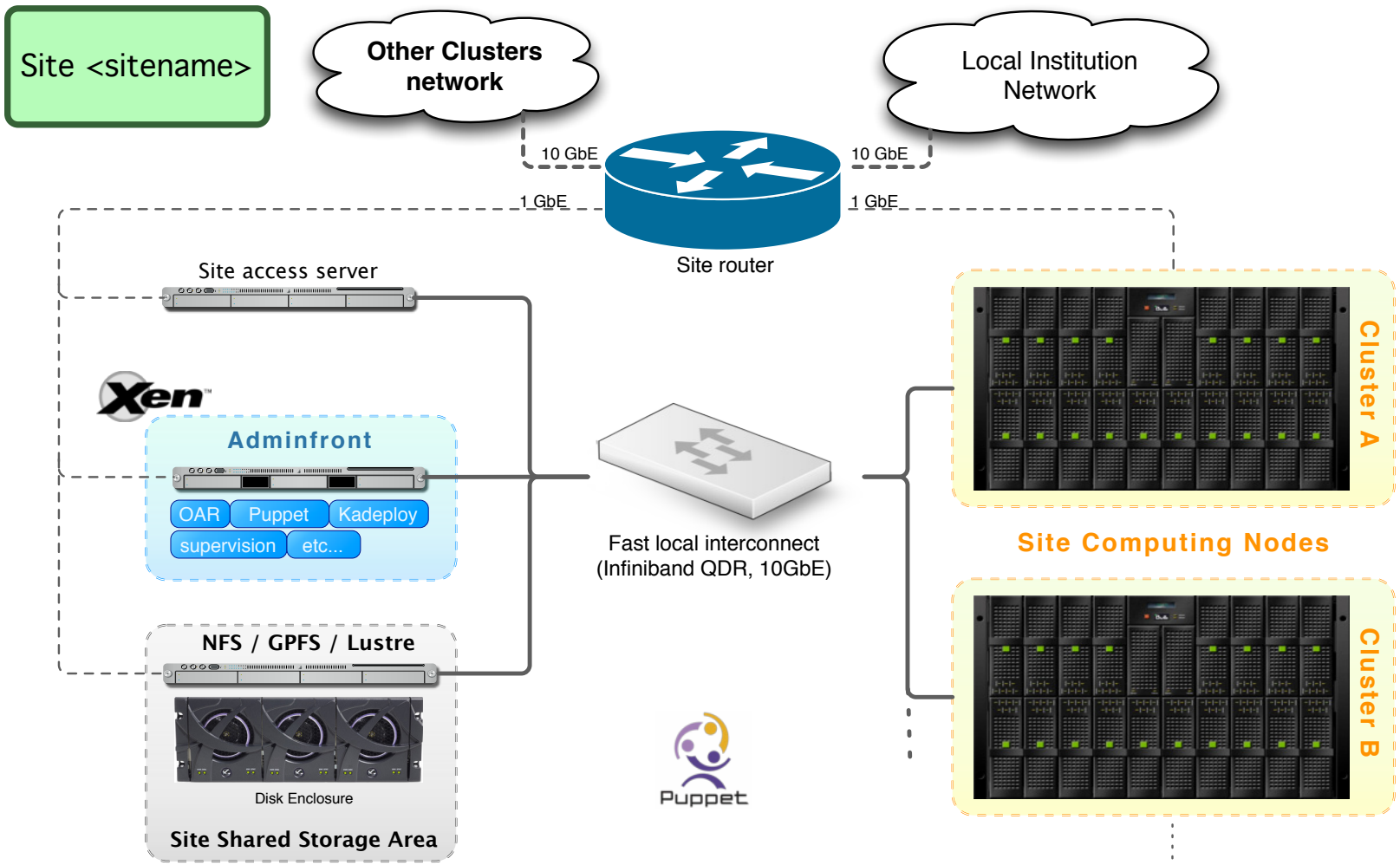
CS.43, AS. 28

Belval

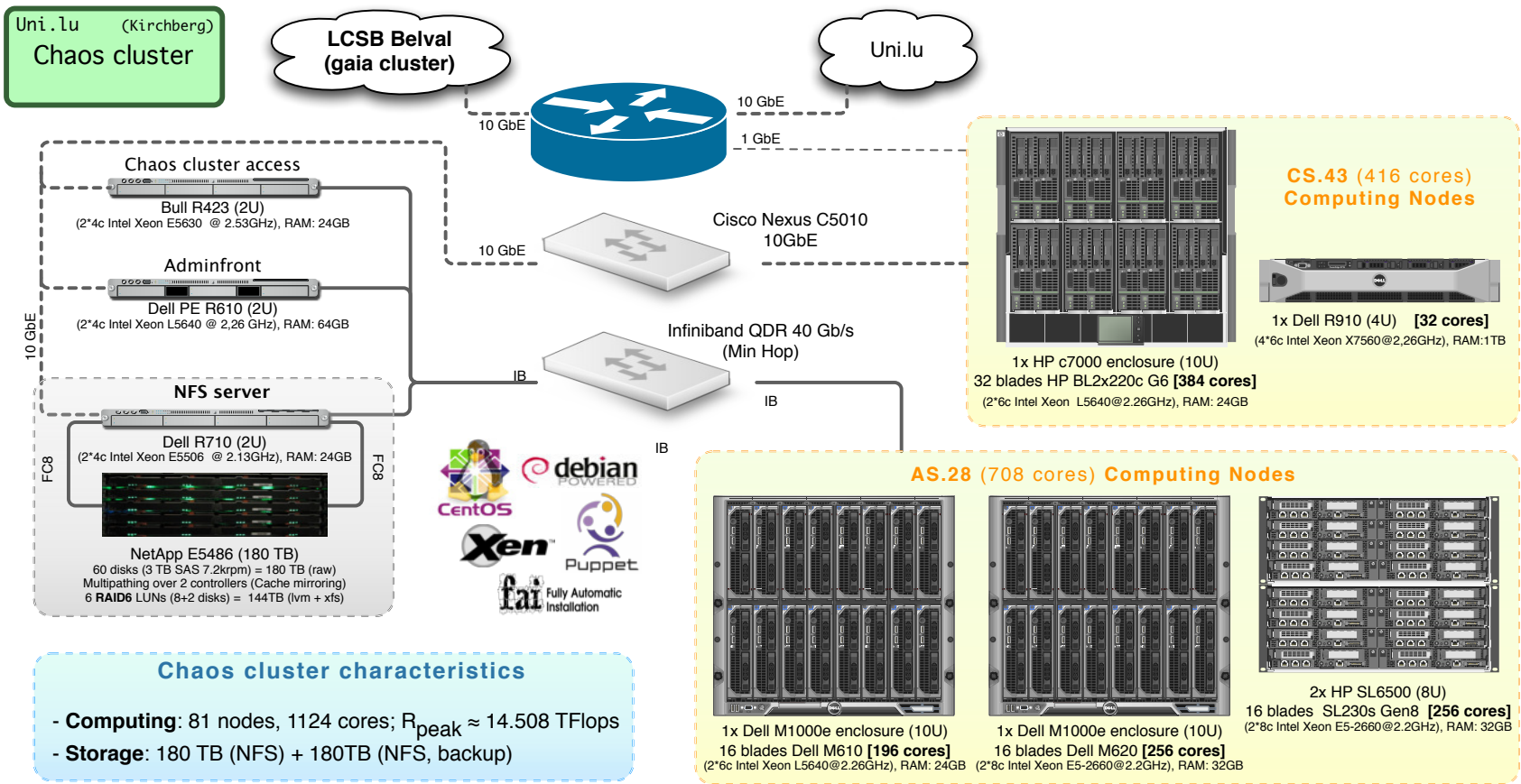
Biotech I, CDC/MSA

2 sites, $\geq$ 4 server rooms

UL HPC: General cluster organization



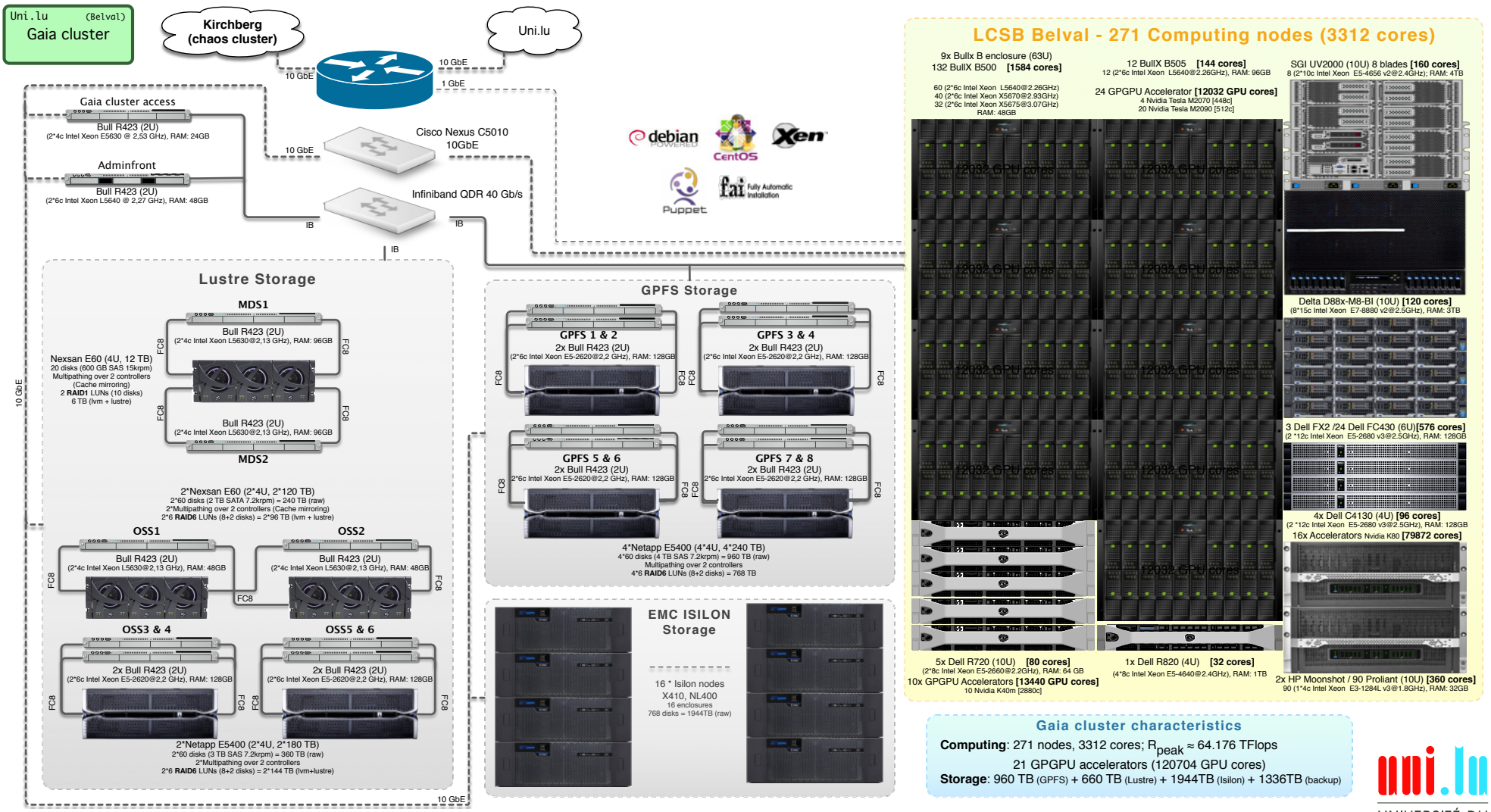
Ex: The chaos cluster



Chaos cluster characteristics

- **Computing**: 81 nodes, 1124 cores; $R_{peak} \approx 14.508$ TFlops
- **Storage**: 180 TB (NFS) + 180TB (NFS, backup)

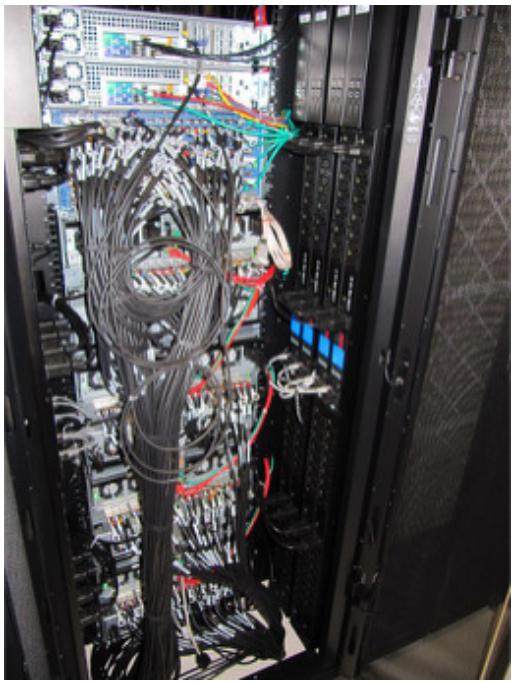
Ex: The gaia cluster



Gaia cluster characteristics

Computing: 271 nodes, 3312 cores; $R_{peak} \approx 64.176$ TFlops
21 GPGPU accelerators (120704 GPU cores)
Storage: 960 TB (GPFS) + 660 TB (Lustre) + 1944TB (Isilon) + 1336TB (backup)

UL HPC Computing capacity



4 clusters

85.543 TFlops

522 nodes

5420 CPU cores

34512 GPU cores

UL HPC – Detailed Computing Nodes

	Date	Vendor	Proc. Description		#N	#C	R_{peak}
chaos	2010	HP	Intel Xeon L5640@2.26GHz	2 × 6C,24GB	32	384	3.472 TFlops
	2011	Dell	Intel Xeon L5640@2.26GHz	2 × 6C,24GB	16	192	1.736 TFlops
	2012	Dell	Intel Xeon X7560@2,26GHz	4 × 6C, 1TB	1	32	0.289 TFlops
	2012	Dell	Intel Xeon E5-2660@2.2GHz	2 × 8C,32GB	16	256	4.506 TFlops
	2012	HP	Intel Xeon E5-2660@2.2GHz	2 × 8C,32GB	16	256	4.506 TFlops
				chaos TOTAL:	**81**	**1120**	**14.495 TFlops**
gaia	2011	Bull	Intel Xeon L5640@2.26GHz	2 × 6C,48GB	72	864	7.811 TFlops
	2012	Dell	Intel Xeon E5-4640@2.4GHz	4 × 8C, 1TB	1	32	0.614 TFlops
	2012	Bull	Intel Xeon E7-4850@2GHz	16 × 10C,1TB	1	160	1.280 TFLops
	2013	Dell	Intel Xeon E5-2660@2.2GHz	2 × 8C,64GB	5	80	1.408 TFlops
	2013	Bull	Intel Xeon X5670@2.93GHz	2 × 6C,48GB	40	480	5.626 TFlops
	2013	Bull	Intel Xeon X5675@3.07GHz	2 × 6C,48GB	32	384	4.746 TFlops
	2014	Delta	Intel Xeon E78880@2.5 GHz	8 × 15C,1TB	1	120	2.4 TFlops
	2014	SGi	Intel Xeon E54650@2.4 GHz	16 × 10C,4TB	1	160	3.072 TFlops
	2015	Dell	Intel Xeon E5-2680@2.5 GHz	2 × 12C,128GB	28	672	26.88 TFlops
	2015	HP	Intel E3-1284Lv3, 1.8GHz	1 × 4C,32GB	90	360	10.368 TFlops
				gaia TOTAL:	**271**	**3312**	**64.18 TFlops**
g5k	2008	Dell	Intel Xeon L5335@2GHz	2 × 4C,16GB	22	176	1.408 TFlops
	2012	Dell	Intel Xeon E5-2630L@2GHz	2 × 6C,24GB	16	192	3.072 TFlops
				granduc/petitprince TOTAL:	**38**	**368**	**4.48 TFlops**

Testing cluster:

nyx	2012	Dell	Intel Xeon E5-2420@1.9GHz	1 × 6C,32GB	2	12	0.091 TFlops
	2013	Viridis	ARM A9 Cortex@1.1GHz	1 × 4C,4GB	96	384	0.422 TFlops
				nyx/viridis TOTAL:	**128**	**516**	**0.52 TFlops**

OpenStack IaaS cluster:

pyro	2015	Dell	Intel Xeon E5-2630Lv2@2.4GHz	2 × 6C,32GB	2	24	0.460 TFlops
	2015	Dell	Intel Xeon E5-2660v2@2.2GHz	2 × 10C,32GB	4	80	1.408 TFlops
				pyro TOTAL:	**4**	**104**	**1.868 TFlops**

UL HPC Storage capacity

4 distributed/parallel FS

1558 disks

5354.4 TB

(incl. 1.516 PB for Backup)

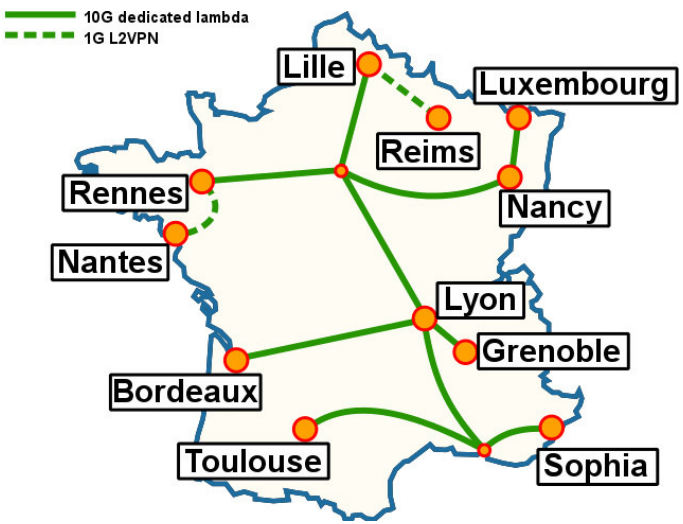
UL HPC Software Stack

- **Operating System**: Linux Debian (CentOS on storage servers)
- **Remote connection to the platform**: SSH
- **User SSO**: OpenLDAP-based
- **Resource management**: job/batch scheduler: OAR
- **(Automatic) Computing Node Deployment**:
  - ↪ FAI (Fully Automatic Installation)
  - ↪ Puppet
  - ↪ Kadeploy
- **Platform Monitoring**: OAR Monika, OAR Drawgantt, Ganglia, Nagios, Puppet Dashboard etc.
- **Commercial Softwares**:
  - ↪ Intel Cluster Studio XE, TotalView, Allinea DDT, Stata etc.

The case of Grid'5000

http://www.grid5000.fr

- Large scale nation wide infrastructure
  - ↪ for large scale parallel and distributed computing research.

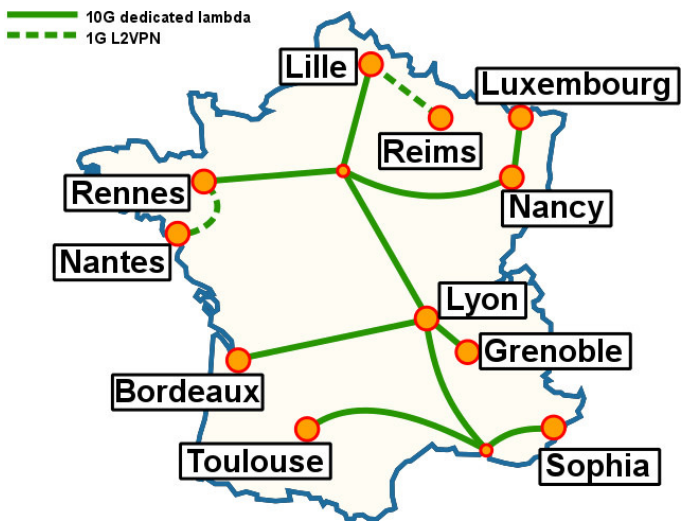


- 10 sites in France
  - ↪ **Abroad**: Luxembourg, Porto Allegre
  - ↪ Total: **7782** cores over **26** clusters
- 1-10GbE / Myrinet / Infiniband
  - ↪ **10Gb/s dedicated** between all sites
- Unique software stack
  - ↪ **kadeploy, kavlan, storage5k**

The case of Grid'5000

http://www.grid5000.fr

- Large scale nation wide infrastructure
  - ↪ for large scale parallel and distributed computing research.



- 10 sites in France
  - ↪ **Abroad**: Luxembourg, Porto Allegre
  - ↪ Total: **7782** cores over **26** clusters
- 1-10GbE / Myrinet / Infiniband
  - ↪ **10Gb/s dedicated** between all sites
- Unique software stack
  - ↪ **kadeploy, kavlan, storage5k**

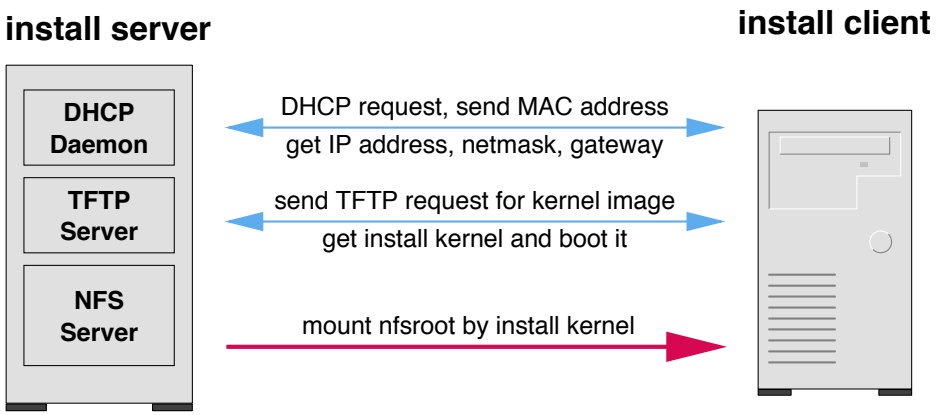
- **Out of scope for this talk**
  - ↪ General information: https://hpc.uni.lu/g5k
  - ↪ Grid'5000 website and documentation: https://www.grid5000.fr

Computing nodes Management

Node deployment by FAI

http://fai-project.org/

- Boot via network card (PXE)
  - ↪ ensure a running diskless Linux OS



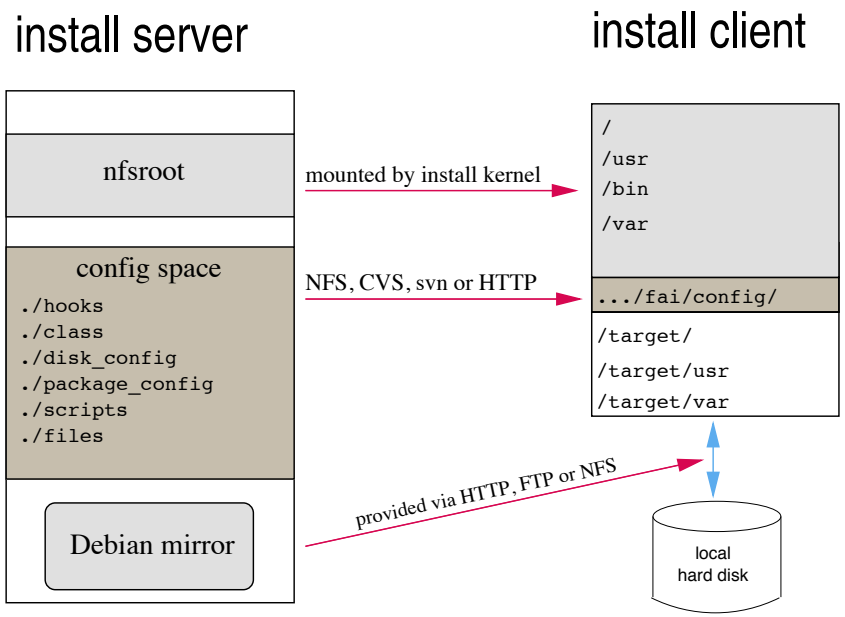
Computing nodes Management

Node deployment by FAI

http://fai-project.org/

fai Fully Automatic Installation

- Boot via network card (PXE)
  - ↪ ensure a running diskless Linux OS
- Get configuration data (NFS)



Computing nodes Management

Node deployment by FAI

http://fai-project.org/

- Boot via network card (PXE)
  - ↪ ensure a running diskless Linux OS
- Get configuration data (NFS)
- Run the installation
  - ↪ partition local hard disks and create filesystems
  - ↪ install software using apt-get command
  - ↪ configure OS and additional software
  - ↪ save log files to install server, then reboot new system

Computing nodes Management

Node deployment by FAI

http://fai-project.org/

- Boot via network card (PXE)
  - ↪ ensure a running diskless Linux OS
- Get configuration data (NFS)
- Run the installation
  - ↪ partition local hard disks and create filesystems
  - ↪ install software using apt-get command
  - ↪ configure OS and additional software
  - ↪ save log files to install server, then reboot new system

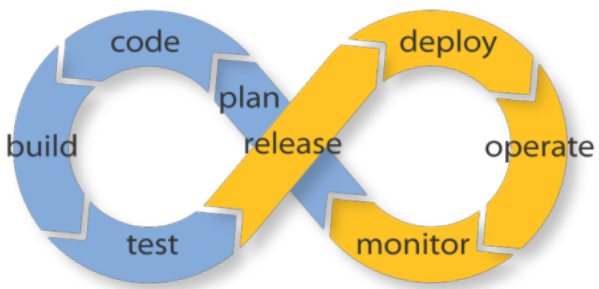
Average reinstallation time: $\simeq$ 500s

IT Serv[er|ice] Management: Puppet

Server/Service configuration by Puppet

http://puppetlabs.com

- **IT Automation** for configuration management
  - ↪ idempotent
  - ↪ agent/master OR stand-alone architecture
  - ↪ cross-platform through Puppet's Resource Abstraction Layer (RAL)
  - ↪ Git-based workflow
  - ↪ PKI-based security (X.509)
- DevOps tool of choice for configuration management
  - ↪ Declarative Domain Specific Language (DSL)



Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

IT Serv[er|ice] Management: Puppet

Server/Service configuration by Puppet

http://puppetlabs.com

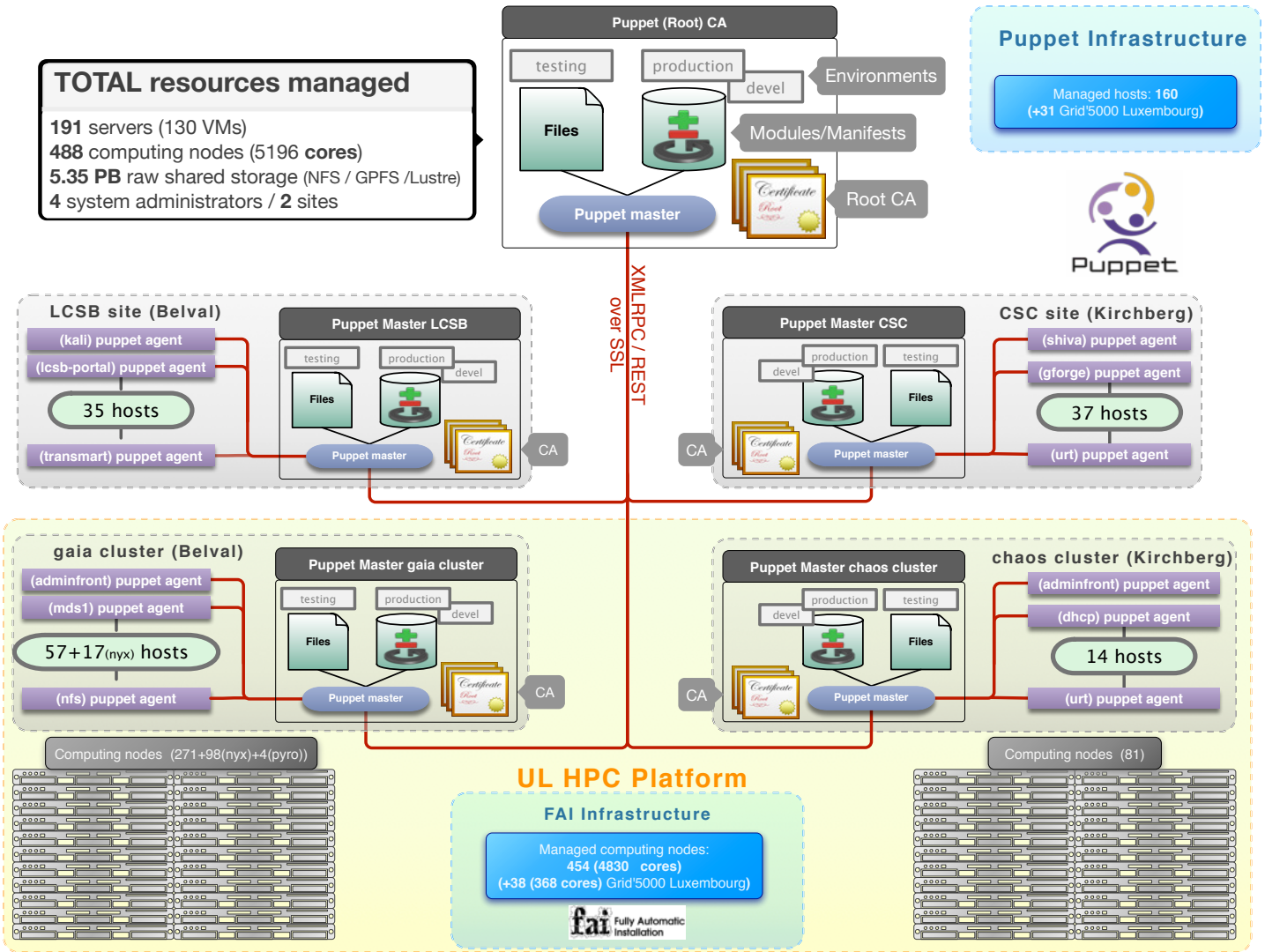
- **IT Automation** for configuration management
  - ↪ idempotent
  - ↪ agent/master OR stand-alone architecture
  - ↪ cross-platform through Puppet's Resource Abstraction Layer (RAL)
  - ↪ Git-based workflow
  - ↪ PKI-based security (X.509)
- DevOps tool of choice for configuration management
  - ↪ Declarative Domain Specific Language (DSL)

Average server installation/configuration time: $\simeq$ 3-6 min

Components of a Puppet architecture

- **Tasks to be deal with**:
  - ↪ definition of the classes to be included in each node
  - ↪ definition of the parameters to use for each node
  - ↪ definition of the configuration files provided to the nodes
- **Components**
  - ↪ Master, CA, and agents
  - ↪ (optional) ENC - External Node Classifier
  - ↪ (optional) ldap/IPA backend
  - ↪ Hiera - Data key-value backend
  - ↪ Public modules - Public shared modules — Puppet Forge
  - ↪ Site modules - Local custom modules

ULHPC Puppet Infrastructure



Software/Modules Management

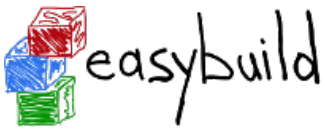
https://hpc.uni.lu/users/software/

- Based on Environment Modules / LMod
  - ↪ convenient way to dynamically change the users' environment `$PATH`
  - ↪ permits to easily load software through `module` command
- Currently on UL HPC:
  - ↪ **133 software packages**, in *multiple* versions, within **18 categories**
  - ↪ hierarchical organization **Ex**: `toolchain/ictce`

```
$> module avail                                    # List evailable modules
```

```
$> module load <category>/<software>[/<version>]
```

Software/Modules Management



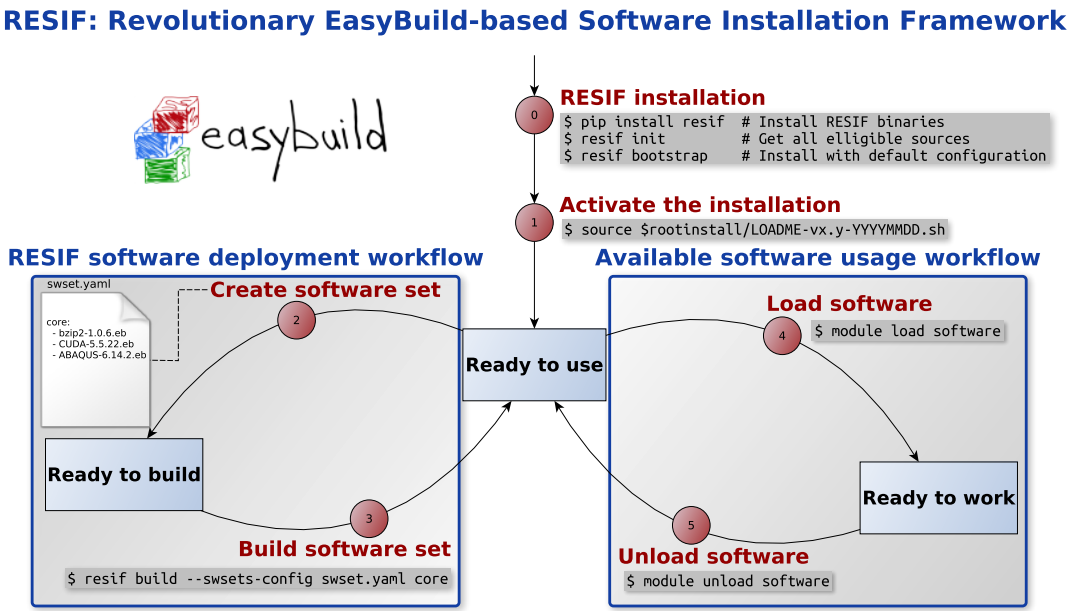
- Easybuild: open-source framework to (automatically) build scientific software
- **Why?**: *"Could you please install this software on the cluster?"*
  - ↪ Scientific software are often **painful** to build
    - ✓ non-standard build tools / incomplete build procedure
    - ✓ hardcoded parameters and/or poor/outdated documentation
  - ↪ EasyBuild helps to facilitate this task
    - ✓ consistent software build and installation framework
    - ✓ automatically generates LMod modulefiles
- cf Practical session / UL HPC Tutorial `http://hpcugent.github.io/easybuild/`

```
$> module avail EasyBuild
$> module load base/EasyBuild  toolchain/ictce/5.3.0
$> eb -S HPL     # Search for recipes for HPL software
$> eb HPL-2.0-ictce-5.3.0.eb # Install HPC 2.1 w. Intel toolchain
```

Software/Modules Management

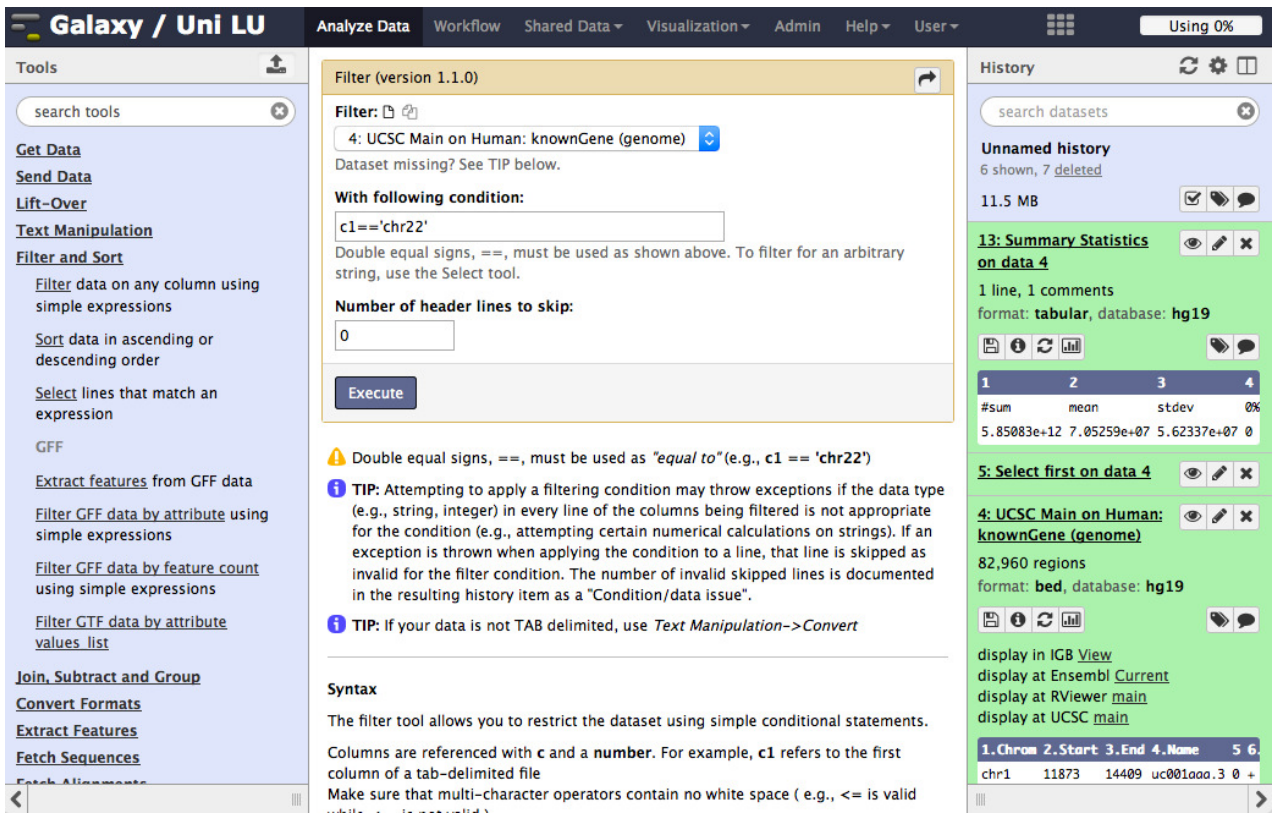
http://resif.readthedocs.io/en/latest/

- **RESIF**: Revolutionary EasyBuild-based Software Installation Framework
  - ↪ Automatic Management of **software sets**
  - ↪ Fully automates software builds and supports all available toolchains
  - ↪ Clean (hierarchical) modules layout to facilitate its usage
  - ↪ "Easy to use" yet **pending workflow rework**



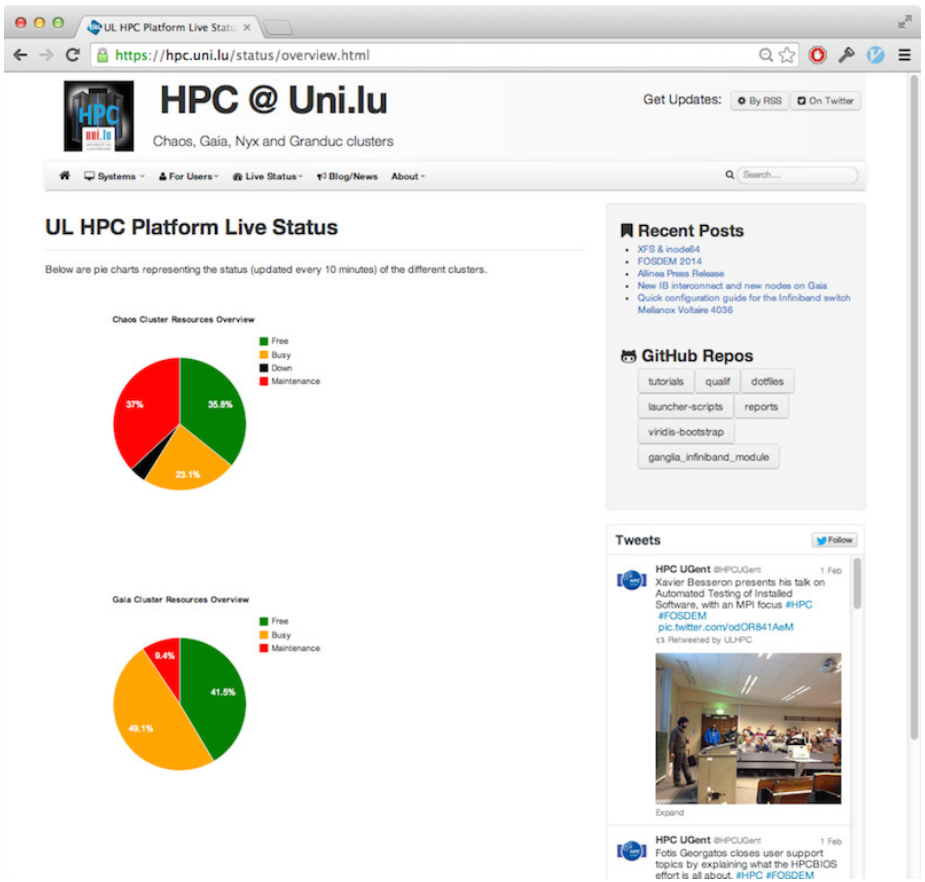
BIO Workflow Management

- Galaxy Portal `http://galaxy-server.uni.lu`
  - ↪ web-based platform for data intensive biomedical research



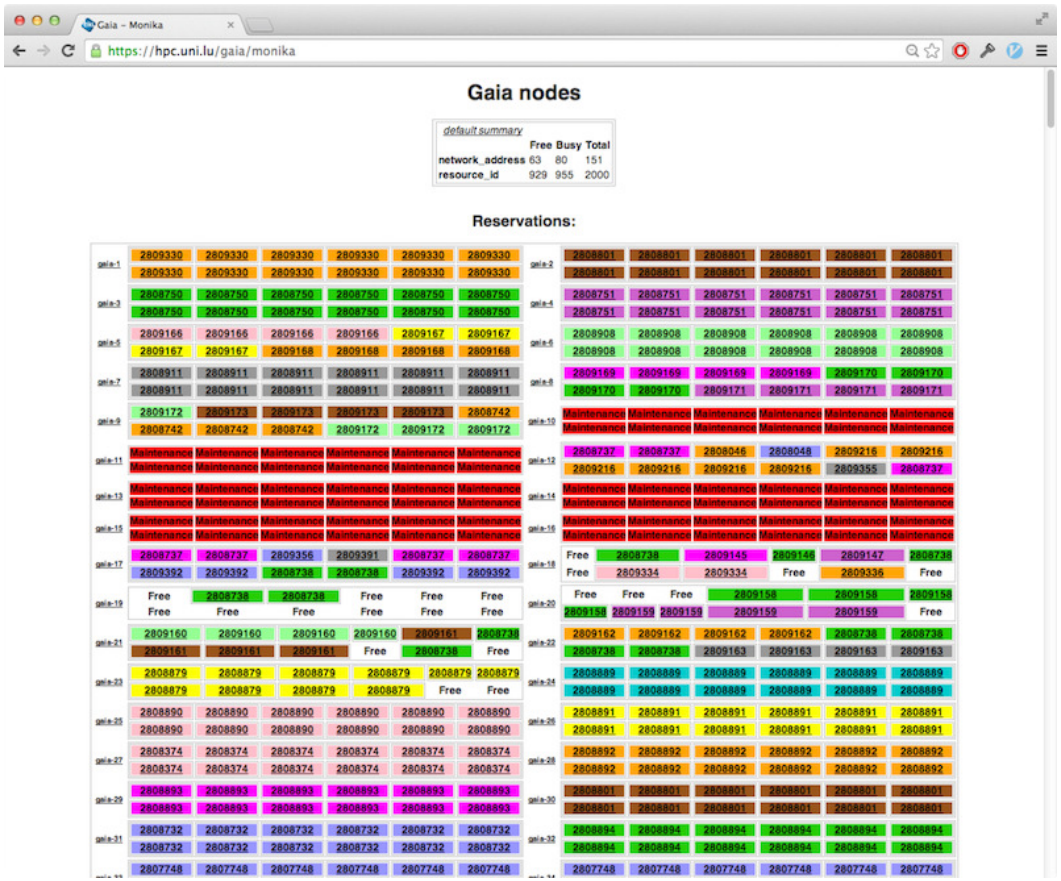
Platform Monitoring

- **General Live Status** `http://hpc.uni.lu/status/overview.html`



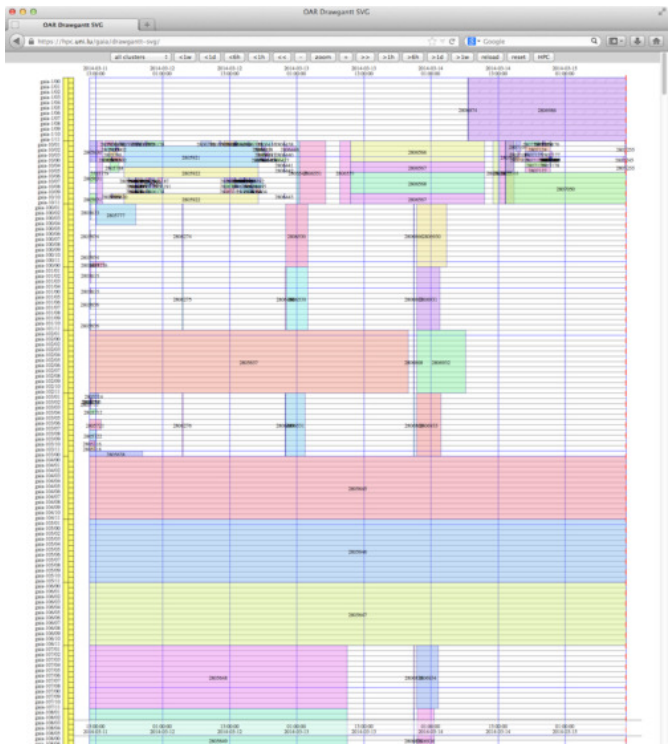
Platform Monitoring

- **Monika** `http://hpc.uni.lu/{chaos,gaia,g5k}/monika`



Platform Monitoring

- **Drawgantt** `http://hpc.uni.lu/{chaos,gaia,g5k}/drawgantt`

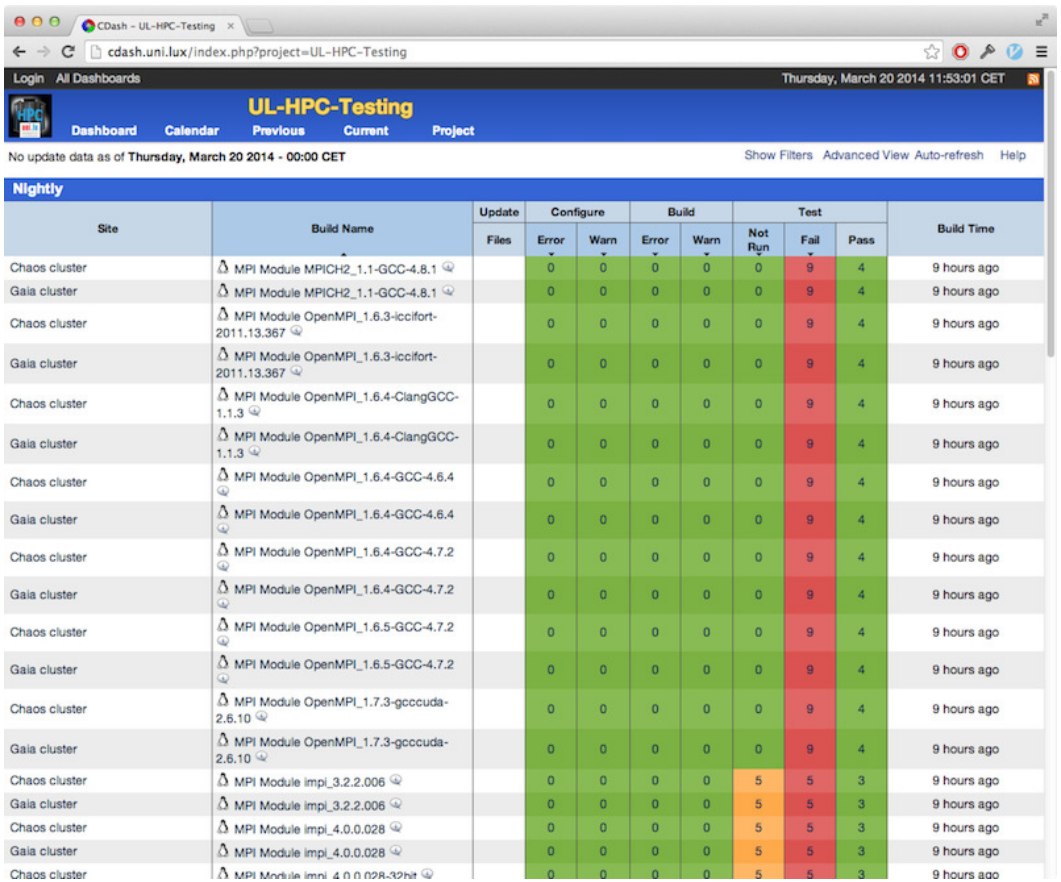


Platform Monitoring

- **Ganglia** `http://hpc.uni.lu/{chaos,gaia,g5k}/ganglia`

Platform Monitoring

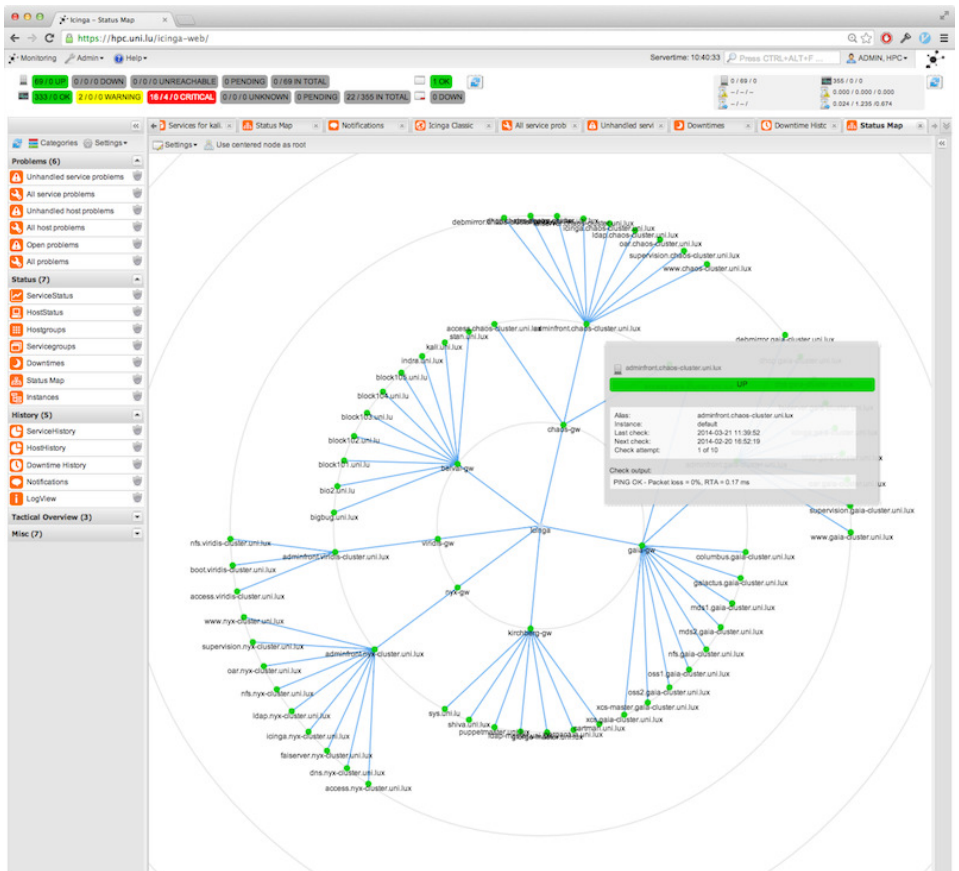
- **CDash** http://cdash.uni.lu/



Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
Chaos cluster	MPI Module MPICH2_1.1-GCC-4.8.1		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module MPICH2_1.1-GCC-4.8.1		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.6.3-iccifort-2011.13.367		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.6.3-iccifort-2011.13.367		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.6.4-ClangGCC-1.1.3		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.6.4-ClangGCC-1.1.3		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.6.4-GCC-4.6.4		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.6.4-GCC-4.6.4		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.6.4-GCC-4.7.2		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.6.4-GCC-4.7.2		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.6.5-GCC-4.7.2		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.6.5-GCC-4.7.2		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module OpenMPI_1.7.3-gcccuda-2.6.10		0	0	0	0	0	9	4	9 hours ago
Gaia cluster	MPI Module OpenMPI_1.7.3-gcccuda-2.6.10		0	0	0	0	0	9	4	9 hours ago
Chaos cluster	MPI Module impi_3.2.2.006		0	0	0	0	5	5	3	9 hours ago
Gaia cluster	MPI Module impi_3.2.2.006		0	0	0	0	5	5	3	9 hours ago
Chaos cluster	MPI Module impi_4.0.0.028		0	0	0	0	5	5	3	9 hours ago
Gaia cluster	MPI Module impi_4.0.0.028		0	0	0	0	5	5	3	9 hours ago
Chaos cluster	MPI Module impi_4.0.0.028-32bit		0	0	0	0	5	5	3	9 hours ago

Platform Monitoring

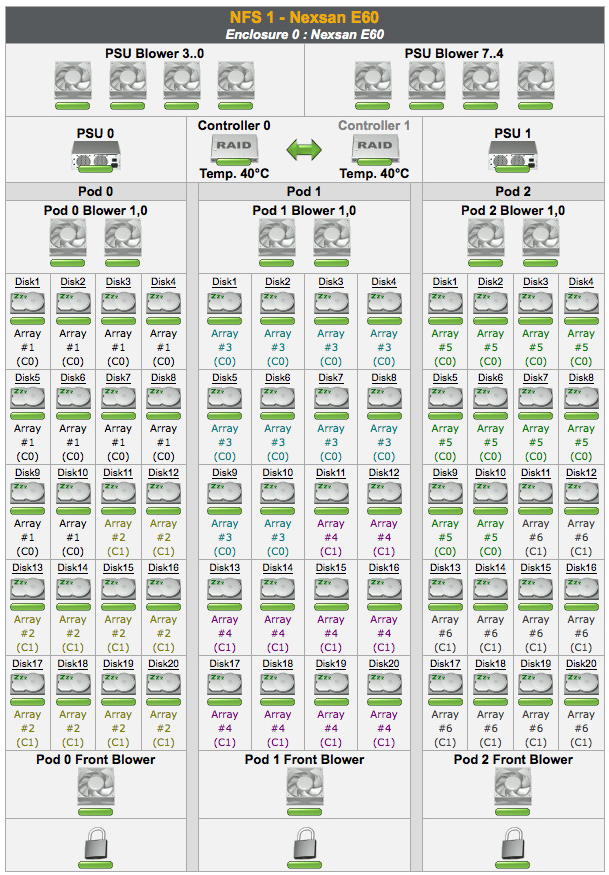
- **Internal Monitoring** `Icinga`



Platform Monitoring

Internal Monitoring

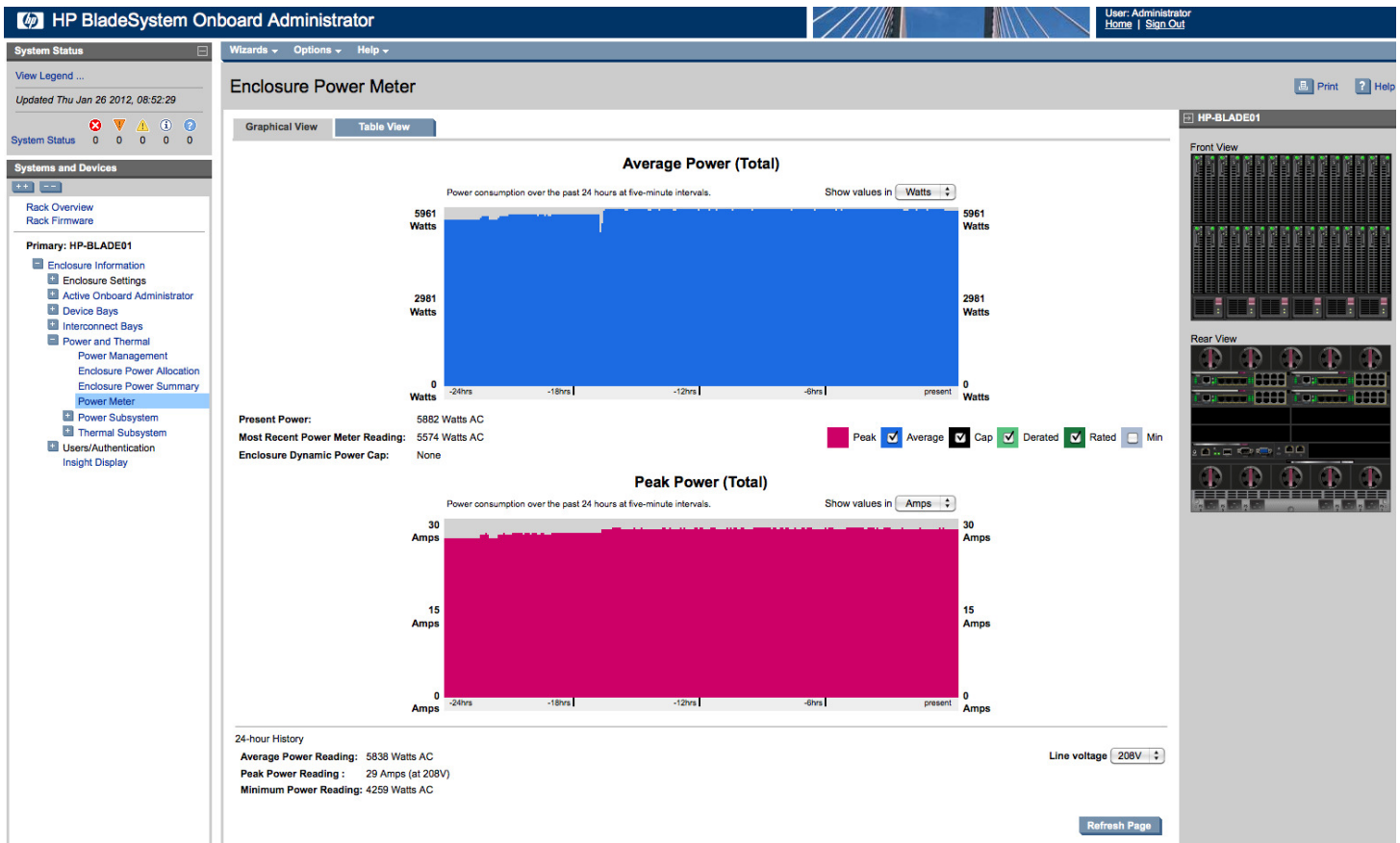
Disk Enclosure status



Platform Monitoring

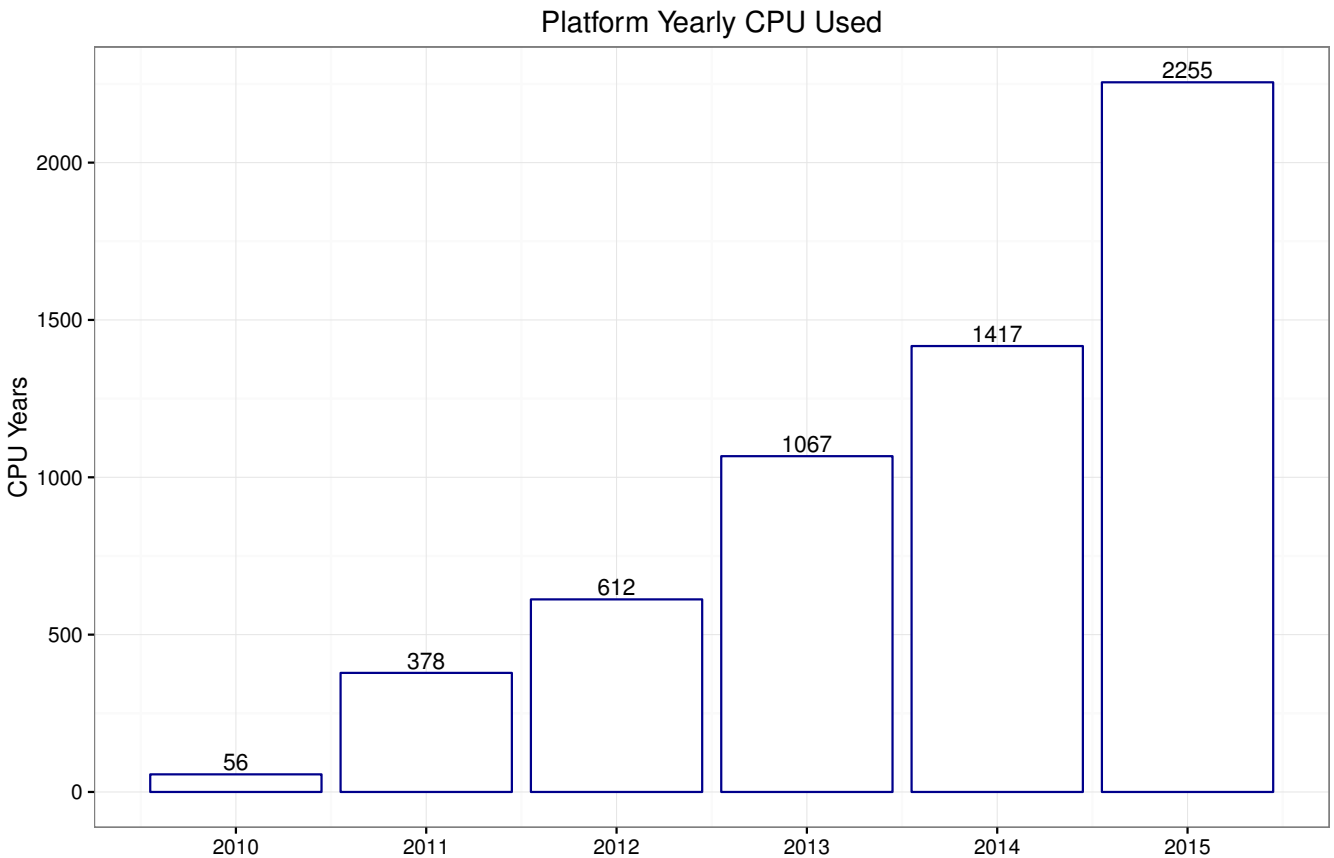
- **Internal Monitoring**

Enclosure status

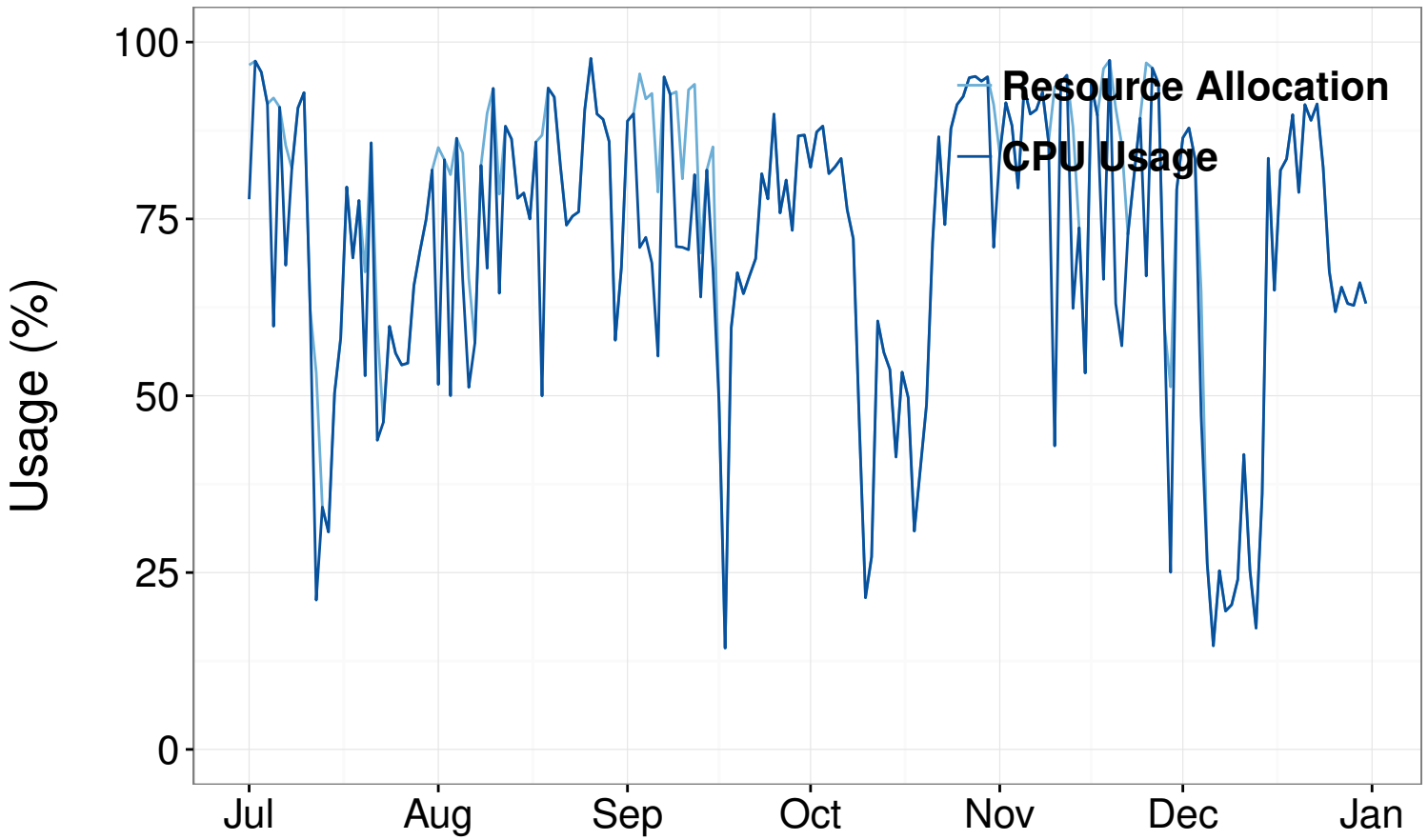


CPU-year usage since 2008

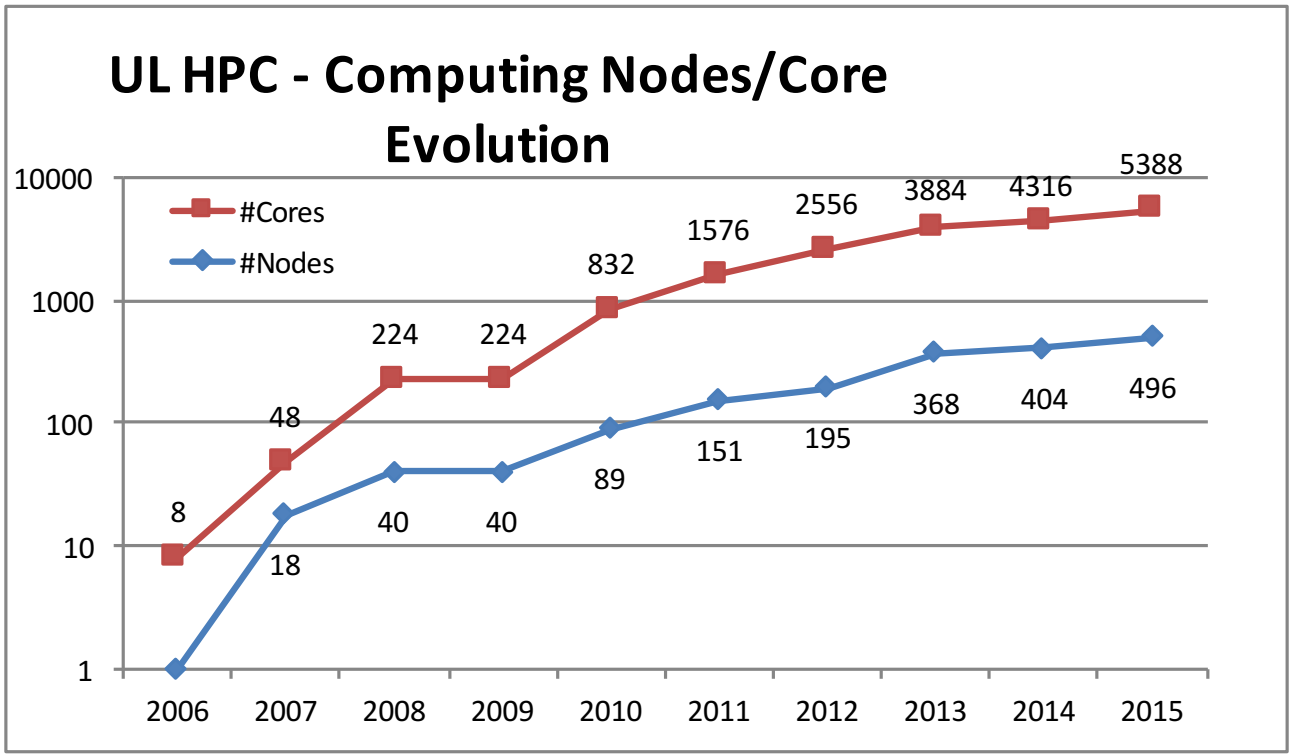
- **CPU-hour**: *work* done by a CPU in one hour of wall clock time



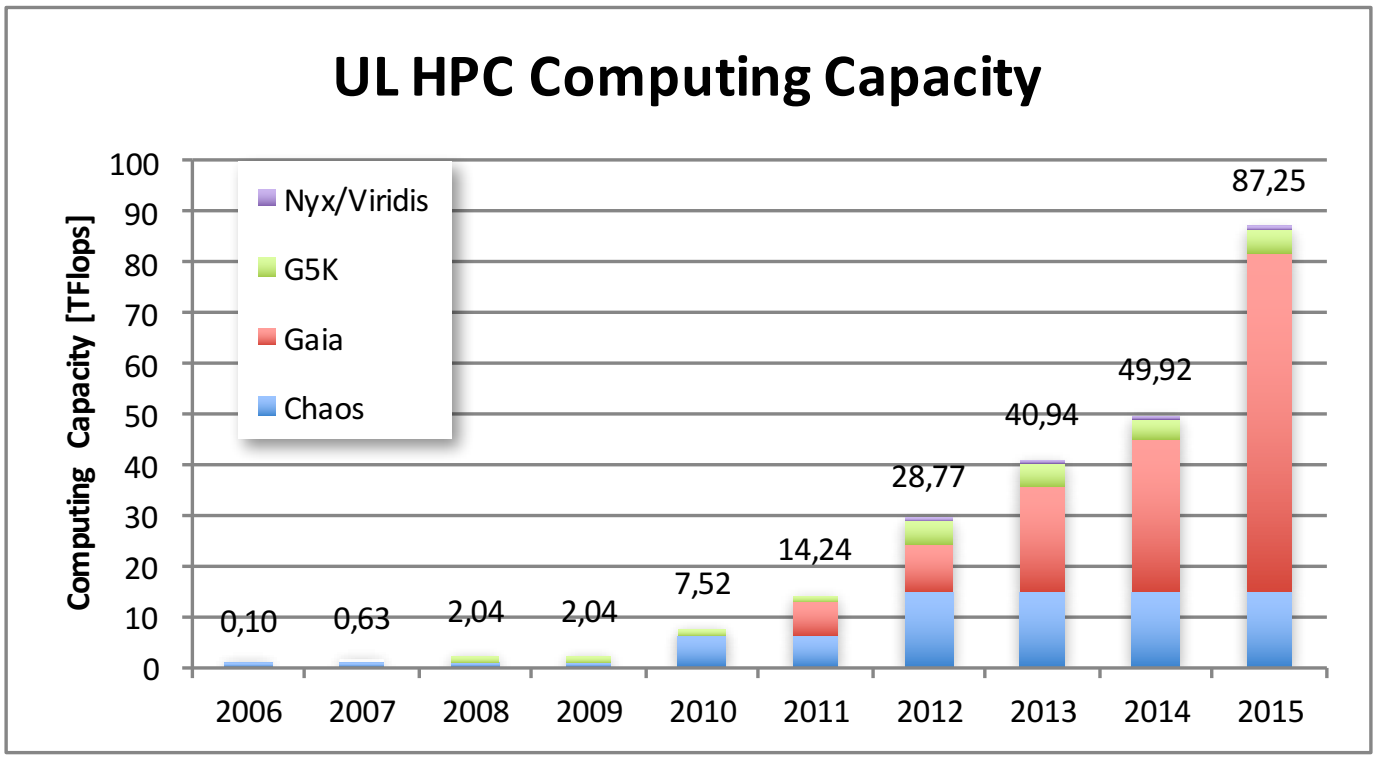
Load Evolution on Gaia...



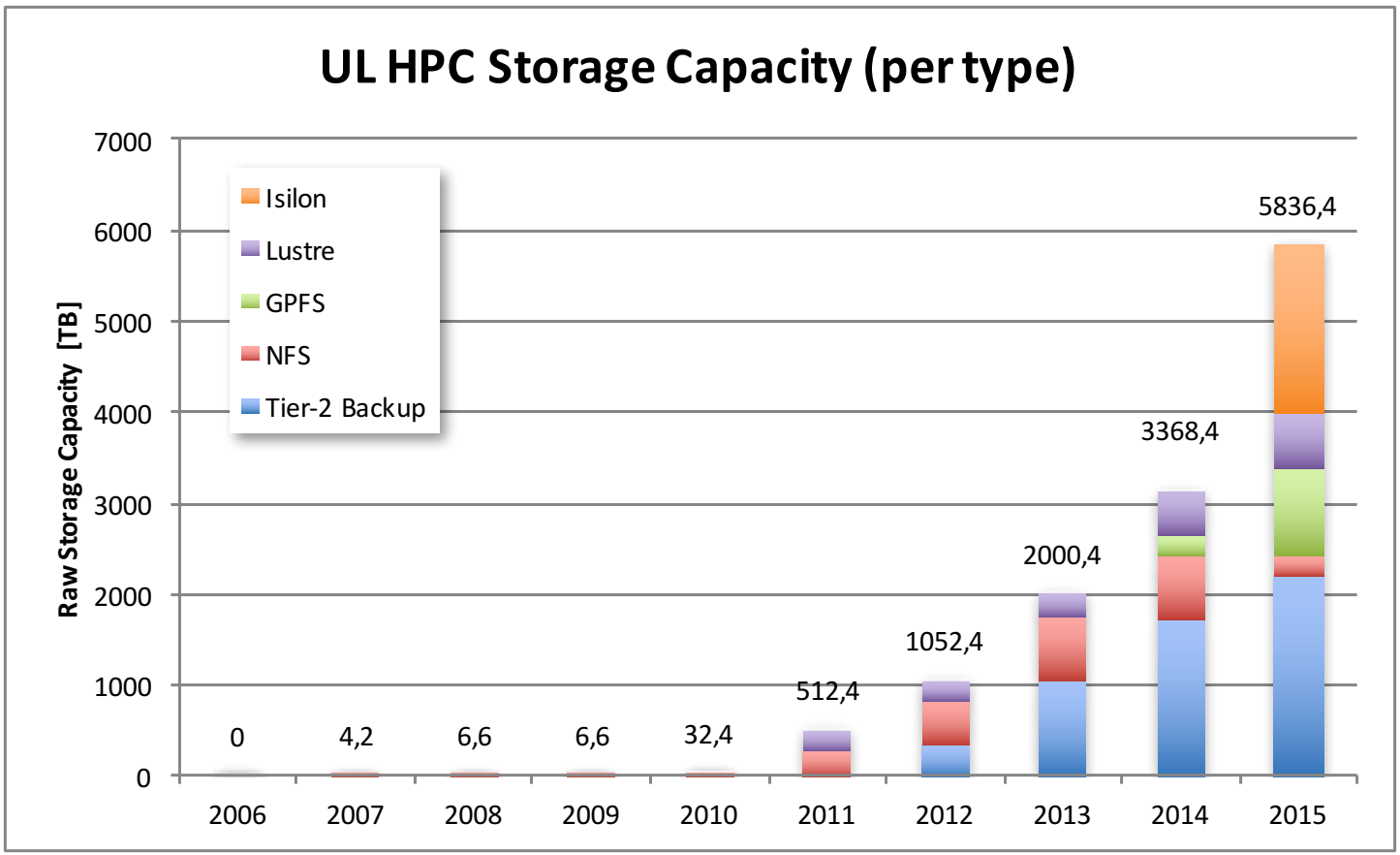
Computing Capacity Evolution



Computing Capacity Evolution



Storage Capacity Evolution



Summary

General Guidelines

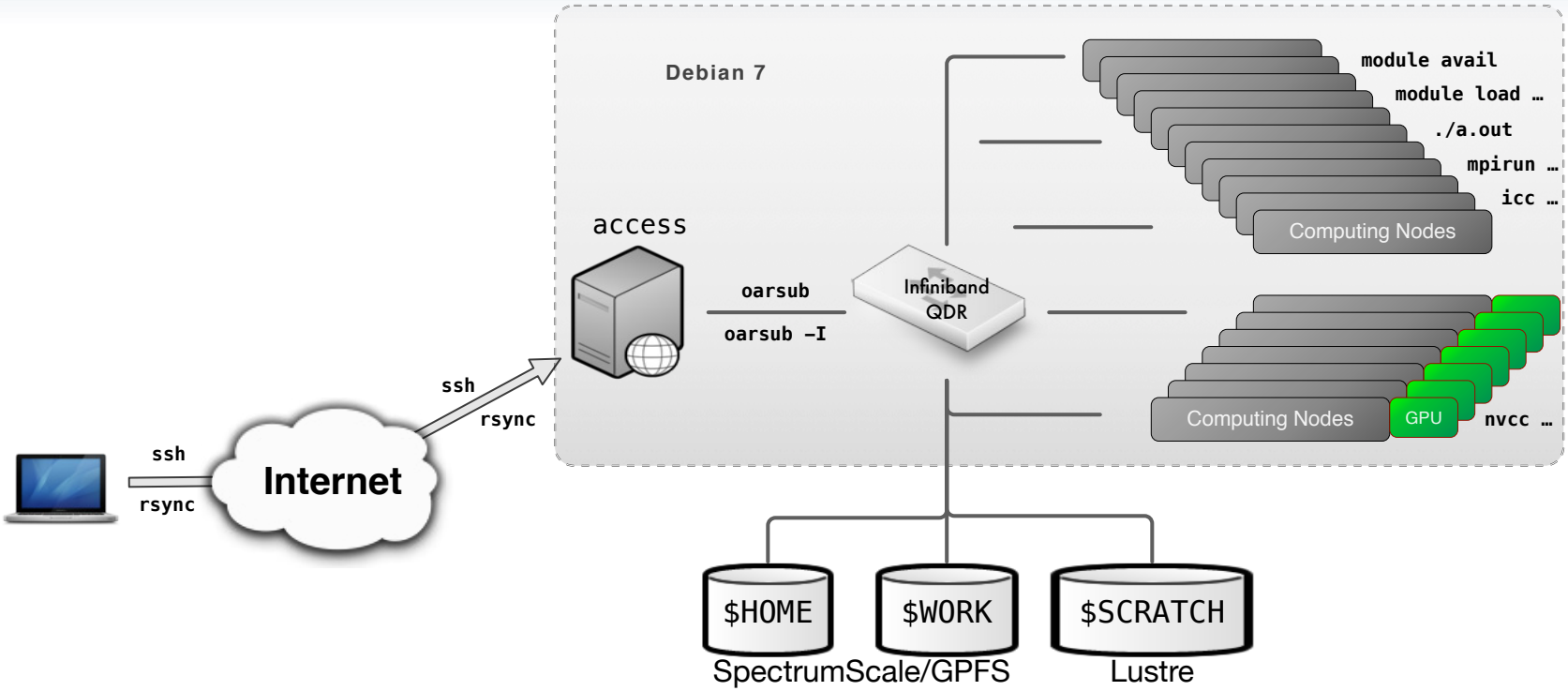
- The UL HPC is a ***shared*** resource
  - ↪ hundreds of users may be logged on at one time
  - ↪ hundreds of jobs may be running on all compute nodes,
- All users must practice ***good citizenship***
  - ↪ limit activities that may impact the system for other users.
  - ↪ **Do not abuse the shared filesystems**
    - ✓ Avoid running jobs in `$HOME`: prefer `$WORK` or `$SCRATCH`
    - ✓ Avoid too many simultaneous file transfers
    - ✓ regularly clean your directories from useless files
  - ↪ **Don't run programs on the login nodes**
  - ↪ Plan large scale experiments during night-time or week-ends
    - ✓ **no more than 120 cores** during working day and working hours

General Guidelines

- The UL HPC is a ***shared*** resource
  - ↪ hundreds of users may be logged on at one time
  - ↪ hundreds of jobs may be running on all compute nodes,
- All users must practice ***good citizenship***
  - ↪ limit activities that may impact the system for other users.
  - ↪ **Do not abuse the shared filesystems**
    - ✓ Avoid running jobs in `$HOME`: prefer `$WORK` or `$SCRATCH`
    - ✓ Avoid too many simultaneous file transfers
    - ✓ regularly clean your directories from useless files
  - ↪ **Don't run programs on the login nodes**
  - ↪ Plan large scale experiments during night-time or week-ends
    - ✓ **no more than 120 cores** during working day and working hours

- For **ALL** publications having results produced using the UL HPC
  - ↪ Acknowledge / cite the UL HPC facility (using official banner)
  - ↪ Tag your publication upon registration on ORBiLu.

Compute Nodes Environment

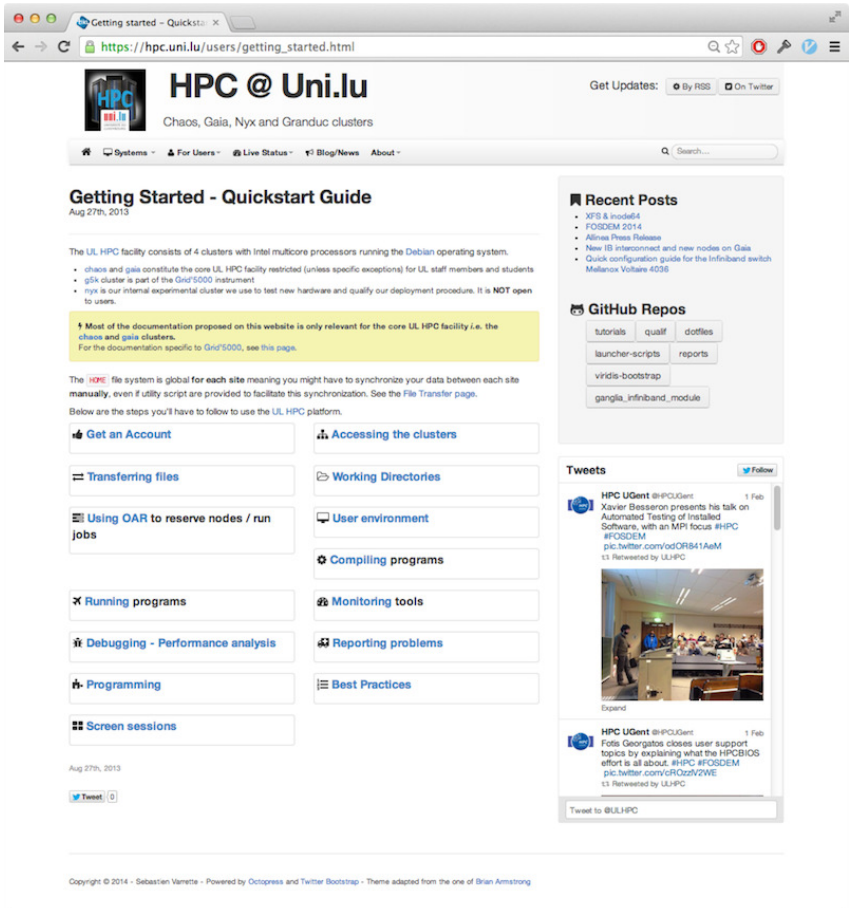


- *OS*: Debian 7 (Wheezy)
- **Storage usage**: `df-ulhpc`
- **Env. modules**: `modules`
  - ↪ **Not** available on frontends
  - ↪ ***Only*** on compute nodes

Directory	Max size	Max #files	Backup
`$HOME`	100 GB	1.000.000	YES
`$WORK`	3 TB		NO
`$SCRATCH`	10 TB		NO

Documentation

http://hpc.uni.lu/users/getting_started.html



http://hpc.uni.lu

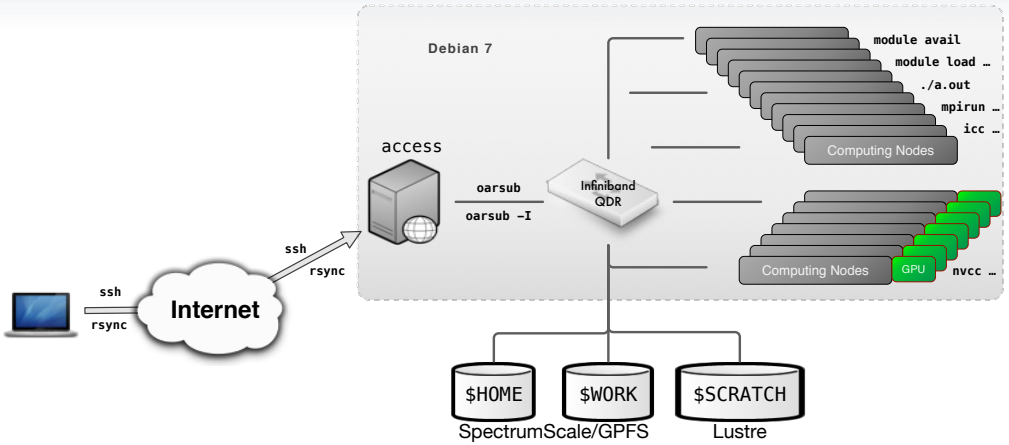
... aka the **rtfm** paradigm

Reference documentation

http://hpc.uni.lu/docs/

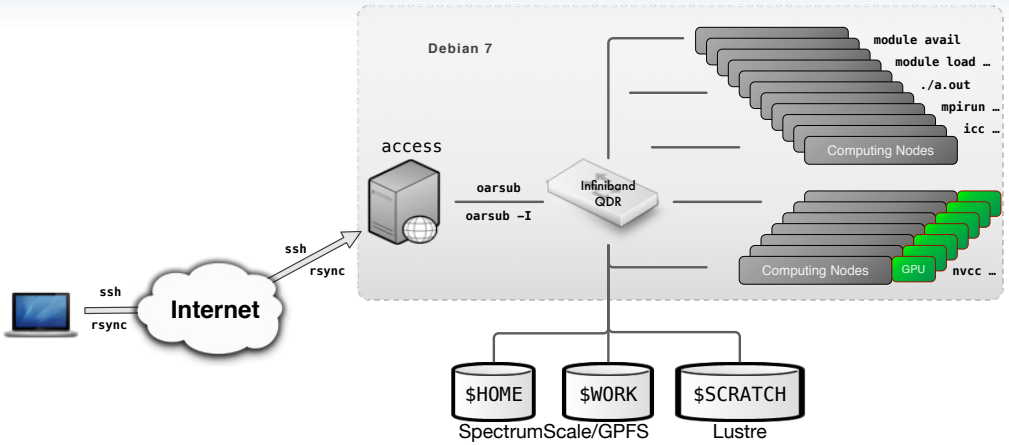
- Github Tutorials
  - ↪ http://ulhpc-tutorials.rtfd.io/
  - ↪ https://github.com/ULHPC/tutorials
- UL HPC Ticketing System
  - ↪ https://hpc-tracker.uni.lu/
- Ask other users hpc-users@uni.lu
  - ↪ ... or us hpc-sysadmins@uni.lu

Typical Workflow on UL HPC resources



- **Preliminary setup**
  1. Connect to the frontend `ssh`
  2. Synchronize you code `scp/rsync/svn/git`
  3. Reserve a few interactive resources `oarsub -I [...]`
     - ✓ *(eventually)* build your program `gcc/icc/mpicc/nvcc..`
     - ✓ Test your experiment on small size problem `mpirun/python/sh...`
     - ✓ Prepare a launcher script `<launcher>.{sh|py}`

Typical Workflow on UL HPC resources



- **Preliminary setup**
  1. Connect to the frontend `ssh`
  2. Synchronize you code `scp/rsync/svn/git`
  3. Reserve a few interactive resources `oarsub -I [...]`
     - ✓ *(eventually)* build your program `gcc/icc/mpicc/nvcc..`
     - ✓ Test your experiment on small size problem `mpirun/python/sh...`
     - ✓ Prepare a launcher script `<launcher>.{sh|py}`
- **Real Experiment**
  1. Connect to the frontend `ssh`
  2. Reserve passive resources `oarsub [...] <launcher>.{sh|py}`
  3. Grab the results `scp/rsync/svn/git`

UL HPC resource manager: OAR

The OAR Batch Scheduler

`http://oar.imag.fr`

- Versatile resource and task manager
  - ↪ schedule **jobs** for users on the cluster **resource**
  - ↪ OAR resource = a node or part of it (CPU/core)
  - ↪ OAR job = execution time (**walltime**) on a set of resources

UL HPC resource manager: OAR

The OAR Batch Scheduler

http://oar.imag.fr

- Versatile resource and task manager
  - ↪ schedule **jobs** for users on the cluster **resource**
  - ↪ OAR resource = a node or part of it (CPU/core)
  - ↪ OAR job = execution time (**walltime**) on a set of resources

OAR main features includes:

- **interactive vs. passive (aka. batch) jobs**
- **best effort jobs**: use more resource, accept their release any time
- **deploy jobs** (Grid5000 only): deploy a customized OS environment
  - ↪ ... and have full (root) access to the resources
- **powerful resource filtering/matching**

Main OAR commands

oarsub	submit/reserve a job	(by default: **1 core for 2 hours**)
oardel	delete a submitted job	
oarnodes	shows the resources states	
oarstat	shows information about running or planned jobs	

	Submission
interactive	`oarsub [options] -I`
passive	`oarsub [options] scriptName`

- Each created job receive an identifier `JobID`
  - ↪ Default passive job log files: `OAR.JobID.std{out,err}`
- You can make a reservation with `-r "YYYY-MM-DD HH:MM:SS"`

Main OAR commands

oarsub	submit/reserve a job	(by default: **1 core for 2 hours**)
oardel	delete a submitted job	
oarnodes	shows the resources states	
oarstat	shows information about running or planned jobs	

	Submission
interactive	`oarsub [options] -I`
passive	`oarsub [options] scriptName`

- Each created job receive an identifier `JobID`
  - ↪ Default passive job log files: `OAR.JobID.std{out,err}`
- You can make a reservation with `-r "YYYY-MM-DD HH:MM:SS"`

Direct access to nodes by `ssh` is forbidden: use `oarsh` instead

OAR job environment variables

Once a job is created, some environments variables are defined:

Variable	Description
`$OAR_NODEFILE`	Filename which lists all reserved nodes for this job
`$OAR_JOB_ID`	OAR job identifier
`$OAR_RESOURCE_PROPERTIES_FILE`	Filename which lists all resources and their properties
`$OAR_JOB_NAME`	Name of the job given by the "-n" option of oarsub
`$OAR_PROJECT_NAME`	Job project name

Useful for MPI jobs for instance:

```
$> mpirun -machinefile $OAR_NODEFILE /path/to/myprog
```

... Or to collect how many cores are reserved per node:

```
$> cat $OAR_NODEFILE | uniq -c
```

OAR job types

Job Type	Max Walltime (hour)	Max #active_jobs	Max #active_jobs_per_user
interactive	12:00:00	10000	5
default	120:00:00	30000	10
besteffort	9000:00:00	10000	1000

cf `/etc/oar/admission_rules/*.conf`

- **interactive**: useful to test / prepare an experiment
  - ↪ you get a shell on the first reserved resource
- **best-effort vs. default**: nearly unlimited constraints **YET**
  - ↪ a `besteffort` job can be killed as soon as a default job as no other place to go
  - ↪ enforce checkpointing (and/or idempotent) strategy

Characterizing OAR resources

Specifying wanted resources in a hierarchical manner

- Use the `-l` option of `oarsub`. Main constraints:

`enclosure=N`	number of enclosure
`nodes=N`	number of nodes
`core=N`	number of cores
`walltime=hh:mm:ss`	job's max duration

Specifying OAR resource properties

- Use the `-p` option of `oarsub`:

Syntax: `-p "property='value'"`

`gpu='{YES,NO}'`	has (or not) a GPU card
`host='fqdn'`	full hostname of the resource
`network_address='hostname'`	Short hostname of the resource
(Chaos only) `nodeclass='{k,b,h,d,r}'`	Class of node

OAR (interactive) job examples

- 2 cores on 3 nodes (same enclosure) for 3h15: Total: 6 cores

```
(frontend)$> oarsub -I -l /enclosure=1/nodes=3/core=2,walltime=3:15
```

OAR (interactive) job examples

- 2 cores on 3 nodes (same enclosure) for 3h15: Total: 6 cores

```
(frontend)$> oarsub -I -l /enclosure=1/nodes=3/core=2,walltime=3:15
```

- 4 cores on a GPU node for 8 hours Total: 4 cores

```
(frontend)$> oarsub -I -l /core=4,walltime=8 -p "gpu='YES'"
```

OAR (interactive) job examples

- 2 cores on 3 nodes (same enclosure) for 3h15: Total: 6 cores

```
(frontend)$> oarsub -I -l /enclosure=1/nodes=3/core=2,walltime=3:15
```

- 4 cores on a GPU node for 8 hours Total: 4 cores

```
(frontend)$> oarsub -I -l /core=4,walltime=8 -p "gpu='YES'"
```

- 2 nodes among the `h-cluster1-*` nodes (Chaos only) Total: 24 cores

```
(frontend)$> oarsub -I -l nodes=2 -p "nodeclass='h'"
```

OAR (interactive) job examples

- 2 cores on 3 nodes (same enclosure) for 3h15: Total: 6 cores

```
(frontend)$> oarsub -I -l /enclosure=1/nodes=3/core=2,walltime=3:15
```

- 4 cores on a GPU node for 8 hours Total: 4 cores

```
(frontend)$> oarsub -I -l /core=4,walltime=8 -p "gpu='YES'"
```

- 2 nodes among the `h-cluster1-*` nodes (Chaos only) Total: 24 cores

```
(frontend)$> oarsub -I -l nodes=2 -p "nodeclass='h'"
```

- 4 cores on 2 GPU nodes + 20 cores on other nodes Total: 28 cores

```
$> oarsub -I -l "{gpu='YES'}/nodes=2/core=4+{gpu='NO'}/core=20"
```

OAR (interactive) job examples

- 2 cores on 3 nodes (same enclosure) for 3h15: Total: 6 cores

```
(frontend)$> oarsub -I -l /enclosure=1/nodes=3/core=2,walltime=3:15
```

- 4 cores on a GPU node for 8 hours Total: 4 cores

```
(frontend)$> oarsub -I -l /core=4,walltime=8 -p "gpu='YES'"
```

- 2 nodes among the `h-cluster1-*` nodes (Chaos only) Total: 24 cores

```
(frontend)$> oarsub -I -l nodes=2 -p "nodeclass='h'"
```

- 4 cores on 2 GPU nodes + 20 cores on other nodes Total: 28 cores

```
$> oarsub -I -l "{gpu='YES'}/nodes=2/core=4+{gpu='NO'}/core=20"
```

- A full big SMP node Total: 160 cores on gaia-74

```
$> oarsub -t bigsmp -I l node=1
```

Some other useful features of OAR

Connect to a running job

```
(frontend)$> oarsub -C JobID
```

Cancel a job

```
(frontend)$> oardel JobID
```

Status of a jobs

```
(frontend)$> oarstat -state -j JobID
```

View the job

```
(frontend)$> oarstat
(frontend)$> oarstat -f -j JobID
```

Get info on the nodes

```
(frontend)$> oarnodes
(frontend)$> oarnodes -l
(frontend)$> oarnodes -s
```

Run a best-effort job

```
(frontend)$> oarsub -t besteffort ...
```

Designing efficient OAR job launchers

Resources/Example

https://github.com/ULHPC/launcher-scripts

- UL HPC grant access to **parallel computing** resources
  - ↪ ideally: OpenMP/MPI/CUDA/OpenCL jobs
  - ↪ if serial jobs/tasks: run them efficiently
- Avoid to submit purely serial jobs to the OAR queue a
  - ↪ waste the computational power (11 out of 12 cores on `gaia`).
  - ↪ use whole nodes by running at least 12 serial runs at once
- **Key**: understand difference between **Task** and **OAR job**

Designing efficient OAR job launchers

Resources/Example

https://github.com/ULHPC/launcher-scripts

- UL HPC grant access to **parallel computing** resources
  - ↪ ideally: OpenMP/MPI/CUDA/OpenCL jobs
  - ↪ if serial jobs/tasks: run them efficiently
- Avoid to submit purely serial jobs to the OAR queue a
  - ↪ waste the computational power (11 out of 12 cores on `gaia`).
  - ↪ use whole nodes by running at least 12 serial runs at once
- **Key**: understand difference between **Task** and **OAR job**

For more information...

- Incoming Practical Session **PS1**, **PS2**
  - ↪ HPC workflow with sequential jobs (C,python,java)

Reporting Problems

https://hpc.uni.lu/users/docs/report_pbs.html

- **First checks**
  1. My issue is probably documented — see User Doc
  2. An event is on-going — cf mail from `hpc-platform@uni.lu`
  3. check the state of your nodes
     - ✓ `oarsub -C <jobid>; htop` — *if reservation still active*
     - ✓ `oarsub -f -j <jobid>` — *post-mortem (check the `events` field)*
     - ✓ Ganglia on your node(s) — https://hpc.uni.lu/status/ganglia.html

Reporting Problems

https://hpc.uni.lu/users/docs/report_pbs.html

- **First checks**
  1. My issue is probably documented — see User Doc
  2. An event is on-going — cf mail from `hpc-platform@uni.lu`
  3. check the state of your nodes
     - ✓ `oarsub -C <jobid>; htop` — *if reservation still active*
     - ✓ `oarsub -f -j <jobid>` — *post-mortem (check the events field)*
     - ✓ Ganglia on your node(s) — https://hpc.uni.lu/status/ganglia.html

- **ONLY NOW**, consider the following depending on the severity:
  - ↪ Open an new issue on http://hpc-tracker.uni.lu (**preferred**)
  - ↪ Mail (only now) us — `hpc-sysadmins@uni.lu`
  - ↪ **Ask the help of other users** — `hpc-users@uni.lu`

Reporting Problems

https://hpc.uni.lu/users/docs/report_pbs.html

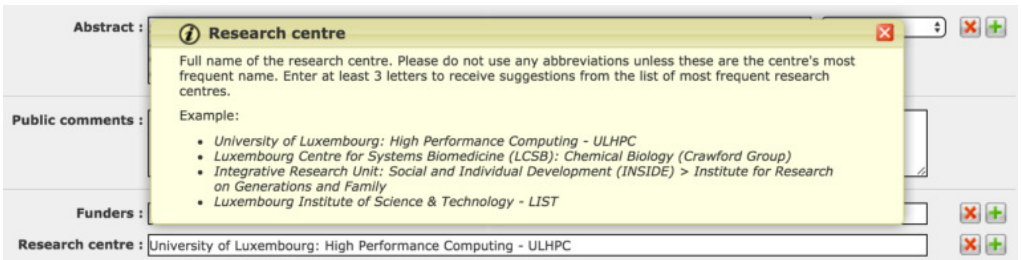
- **First checks**
  1. My issue is probably documented — see User Doc
  2. An event is on-going — cf mail from `hpc-platform@uni.lu`
  3. check the state of your nodes
     - ✓ `oarsub -C <jobid>; htop` — *if reservation still active*
     - ✓ `oarsub -f -j <jobid>` — *post-mortem (check the events field)*
     - ✓ Ganglia on your node(s) — https://hpc.uni.lu/status/ganglia.html

- **ONLY NOW**, consider the following depending on the severity:
  - ↪ Open an new issue on http://hpc-tracker.uni.lu (**preferred**)
  - ↪ Mail (only now) us — `hpc-sysadmins@uni.lu`
  - ↪ **Ask the help of other users** — `hpc-users@uni.lu`

- In all cases: **Carefully describe the problem and the context**
  - ↪ Guidelines

Reporting Obtained Results

- In your **scientific publications**: *as per Acceptable Use Policy (AUP)*
  - ↪ **acknowledge** your usage of the UL HPC platform
  - ↪ *(if possible)* **cite** the UL HPC paper `\cite{VBCG_HPCS14}`
- **More importantly**: add ULHPC Tag on your ORBi[lu] publication




```
@InProceedings{VBCG_HPCS14,
   author =       {S. Varrette and P. Bouvry and H. Cartiaux and F. Georgatos},
   title =        {Management of an Academic HPC Cluster: The UL Experience},
   booktitle =    {Proc. of the 2014 Intl. Conf. on High Performance Computing \& Simulation (HPCS 2014)},
   year =         {2014},
   pages =        {959--967},
   month =        {July},
   address =      {Bologna, Italy},
   publisher =    {IEEE},
}
```

Summary

Current Infrastructure Limitations

- **Emergency needs / UL Commitments** for 2016
  - ↪ Physics: new Prof. was promised 1500 cores
  - ↪ LCSB: + 1 PetaByte (Intl. NCER-PD & ELIXIR projects)
  - ↪ ... while increasing loads from finance (LSF), SnT and FSTC

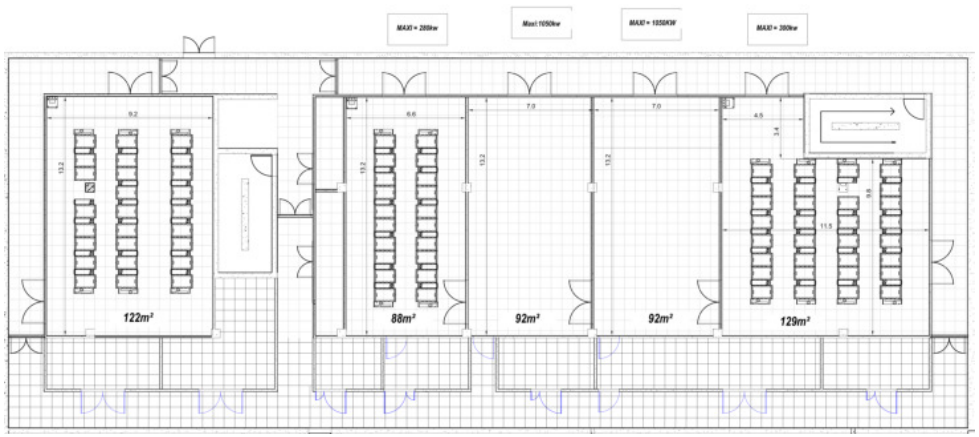
Kirchberg & Belval BT1 Data Centers saturated

- **Opportunities**:
  - ↪ External hosting considered yet too expensive and unadapted
  - ↪ **in-house MSA Centre de Calcul (CDC) S-01** already operational
    - ✓ 2 rooms currently used by Restena & SIU
    - ✓ **not** designed for HPC requirements...

Infrastructure Plans starting 2016

MSA CDC S-02 as the new UL HPC Data Center (DC)

- $\simeq$500m^2 for max. 5 server rooms sustaining HPC requirements
- DC preparation will result in **2 rooms** being ready early 2017
  - ↪ **RFP 1** (DC infrastructure): **Oct. 2016** (SIU)
  - ↪ **RFP 2 & 3** (HPC + storage equipment): **Sept. 2016** (HPC)
  - ↪ **RFP 4** (DLC HPC): **May. 2017** (HPC)

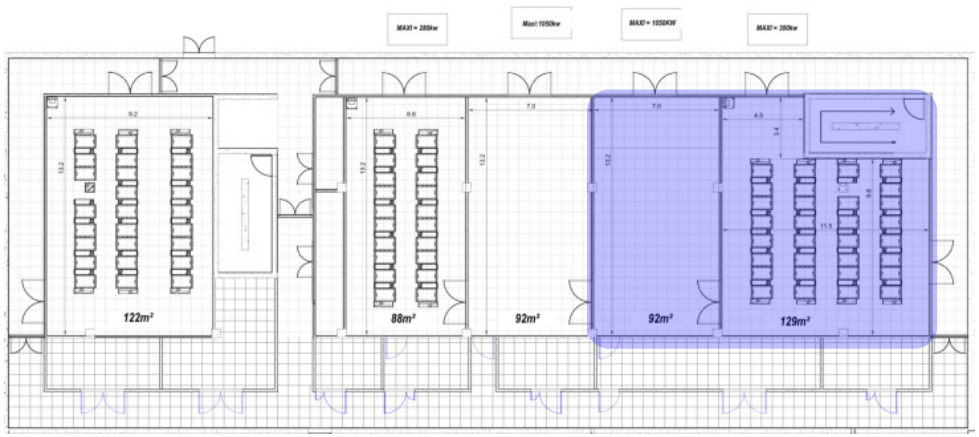


- $\simeq$ **1050kW** per **HPC** room
  - ↪ Direct Liquid Cooling (DLC)
- $\simeq$ **300kW** per **storage** room
  - ↪ rooms 1, 2 & 5

Infrastructure Plans starting 2016

MSA CDC S-02 as the new UL HPC Data Center (DC)

- $\simeq$500m^2 for max. 5 server rooms sustaining HPC requirements
- DC preparation will result in **2 rooms** being ready early 2017
  - ↪ **RFP 1** (DC infrastructure): **Oct. 2016** (SIU)
  - ↪ **RFP 2 & 3** (HPC + storage equipment): **Sept. 2016** (HPC)
  - ↪ **RFP 4** (DLC HPC): **May. 2017** (HPC)



- $\simeq$ **1050kW** per **HPC** room
  - ↪ Direct Liquid Cooling (DLC)
- $\simeq$ **300kW** per **storage** room
  - ↪ rooms 1, 2 & 5

2016 RFPs by UL HPC

- RFP 160019: **High Performance Storage** of capacity $>$ 1PB
  - ↪ 6 vendors solution received, 1 withdraw before the end
  - ↪ Selected solution: **1.44 PB (raw)**, $>$ 10Gb/s RW
    - ✓ `GPFS` over DDN enclosures

- RFP 160020: **HPC facility** of capacity $R_{max} > 80$ TFlops
  - ↪ 5 vendors solutions received
  - ↪ Selected solution: $\mathbf{R_{max} = 94.08}$ **TFlops** (not R_{peak})
  - ↪ 100 nodes, 2800 cores, and per node:
    - ✓ 2x Intel Xeon E5-2680v4 14C@2.2 GHz
    - ✓ 128GB RAM
    - ✓ Infiniband Mellanox EDR 100Gb/s

Disruptives Changes to come

- This new cluster (to be released in Feb. 2017) will serve as testbed for improved management workflow
- Brings many disruptive changes to the current setup!
  - ↪ new batch scheduler (SLURM)
  - ↪ containers support (Shifter), compliant with Docker
  - ↪ and so much more...

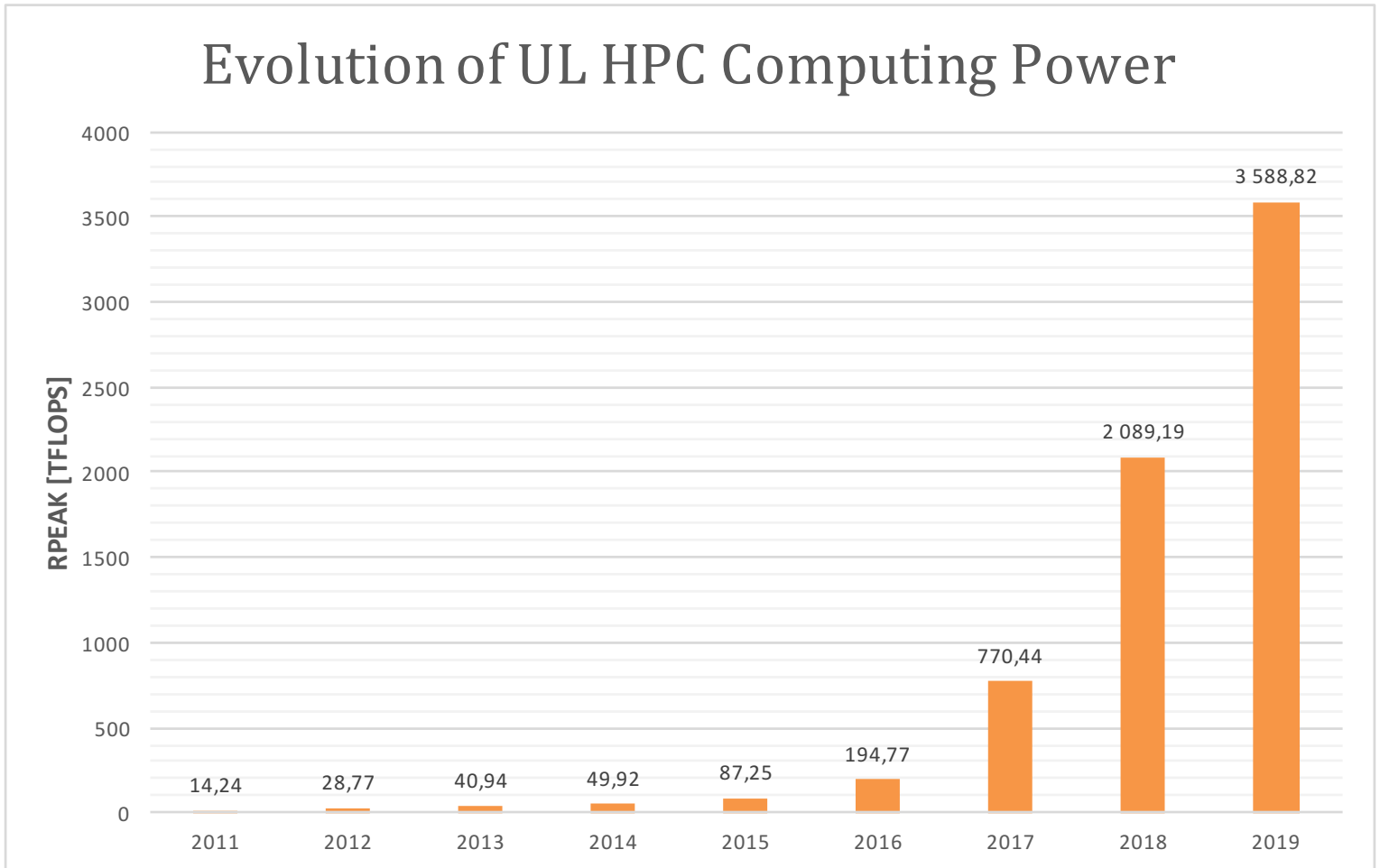
Changes come from UL HPC Survey...

- You can still fill it with your own needs:

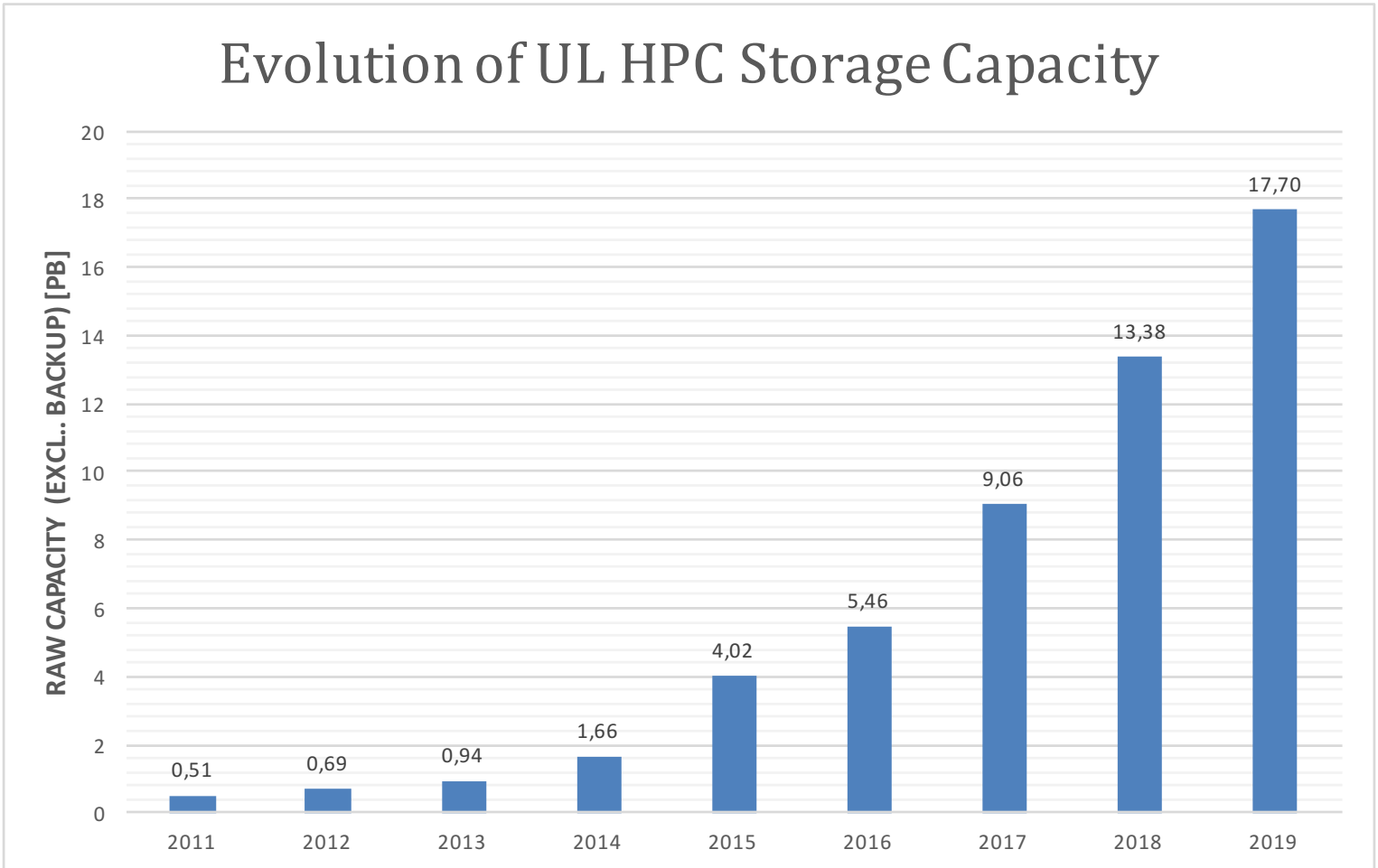
https://goo.gl/OflPyf

- 2017++ other changes: **New large scale HPC cluster**
  - ↪ DLC (Direct Liquid Cooling) based cooling, to enter Top 500

Planned Computing Evolution



Planned Storage Evolution



ETP4HPC

http://www.etp4hpc.eu

- **European Technology Platform (ETP) for HPC**
  - ↪ Industry-led forum founded by stakeholders of HPC technology
  - ↪ Providing the framework to define research priorities and actions
  - ↪ **Objective**: EU growth, competitiveness, sustainability by HPC
  - ↪ **Strategic Research Agenda**
    - ✓ **Creation** of new technologies within the entire HPC stack
    - ✓ **Improvement** of system characteristics (Extreme Scale Reqs.)
    - ✓ **New deployment** fields and **expansion** of HPC utilization

ETP4HPC

http://www.etp4hpc.eu

- **European Technology Platform (ETP) for HPC**
  - ↪ Industry-led forum founded by stakeholders of HPC technology
  - ↪ Providing the framework to define research priorities and actions
  - ↪ **Objective**: EU growth, competitiveness, sustainability by HPC
  - ↪ **Strategic Research Agenda**
    - ✓ **Creation** of new technologies within the entire HPC stack
    - ✓ **Improvement** of system characteristics (Extreme Scale Reqs.)
    - ✓ **New deployment** fields and **expansion** of HPC utilization

Since July 2016...

- **UL is an official member of ETP4HPC!**
  - ↪ participation of key UL HPC experts in various WG

EU HPC Initiatives In progress

PRACE

- Partnership for Advanced Computing in Europe
- Non-profit association with 25 member countries
- Providing access to EU Tier-0 compute & data resources
  - ↪ for large-scale scientific and engineering applications
  - ↪ **Objective**:
    - ✓ enable high impact scientific discovery and engineering R&D
    - ✓ enhance European competitiveness

EU HPC Initiatives In progress

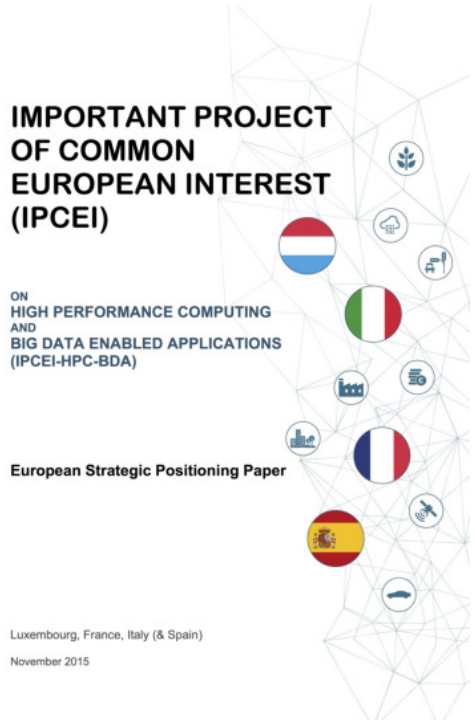
PRACE

- Partnership for Advanced Computing in Europe
- Non-profit association with 25 member countries
- Providing access to EU Tier-0 compute & data resources
  - ↪ for large-scale scientific and engineering applications
  - ↪ **Objective**:
    - ✓ enable high impact scientific discovery and engineering R&D
    - ✓ enhance European competitiveness

- UL to apply as official national representative for PRACE
  - ↪ nomination pending approval by ministry

EU HPC Initiatives In progress

- Important Project of Common European Interest
- IPCEI on *HPC and Big Data Application*
  - ↪ part of Junker plan
  - ↪ launched on Nov. 17th 2015 (at European Data Forum)
  - ↪ $\simeq$ 3 B€ investment
- Lead by Luxembourg through Ministry of Economy
  - ↪ Jean-Marie Spauss appointed as advisor to MECO
  - ↪ UL, LIST & Luxinnovation to support MECO



Final Remarks...

IEEE CloudCom 2016 - Dec 12th-15th, 2016

http://2016.cloudcom.org

- One of the top IEEE conference on Cloud Computing
  - world-class keynotes and technical papers
  - practical tutorials and business panels
- **Come and Register Now !**

Contacts:

Sebastien.Varrette@uni.lu
Pascal.Bouvry@uni.lu

FACULTY OF SCIENCES, TECHNOLOGY AND COMMUNICATION

1

Thank you for your attention...

Questions?

http://hpc.uni.lu

Sebastien Varrette
mail: sebastien.varrette@uni.lu
Office E-007
Campus Kirchberg
6, rue Coudenhove-Kalergi
L-1359 Luxembourg

UL HPC Management Team
mail: hpc-sysadmins@uni.lu