UL HPC School 2017 [bis]
Overview & Challenges of the UL HPC facility at the EuroHPC Horizon

Prof. Pascal Bouvry, Dr. Sebastien Varrette and the UL HPC Team
(V. Plugaru, S. Peter, H. Cartiaux & C. Parisot)

Nov. 9th, 2017, MSA 3.070
University of Luxembourg (UL), Luxembourg
Welcome to the UL HPC School 2017

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

- **6th edition** of this training...
  - started in 2014
  - This one is the **short** version
    - 1-day event
    - Parallel sessions, feat. basic tutorials

- **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.)
Welcome to the UL HPC School 2017

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

- **6th edition** of this training...
  - started in 2014
  - This one is the **short** version
    - 1-day event
    - Parallel sessions, feat. basic tutorials

- **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.)

- **Next edition** planned for **June., 2018** in Belval
  - Full 2-days event, addressing advanced tutorials
## Agenda Nov 9th, 2017

### Main Track (MSA 3.070)

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>9h00-10h00</td>
<td><strong>PS1a</strong>: Getting Started on the UL HPC platform</td>
</tr>
<tr>
<td>10h00-10h30</td>
<td><strong>Coffee break</strong></td>
</tr>
<tr>
<td>10h30-11h45</td>
<td><strong>Keynote</strong>: Overview and Challenges of the UL HPC Facility at the Belval and EuroHPC Horizon</td>
</tr>
<tr>
<td>11h45-12h30</td>
<td><strong>PS1b</strong>: Getting Started on the UL HPC platform - continued</td>
</tr>
<tr>
<td>12h30-13h30</td>
<td><strong>LUNCH</strong></td>
</tr>
<tr>
<td>13h30-15h30</td>
<td><strong>PS2</strong>: HPC workflow with sequential jobs (test cases on GROMACS, Java and Python)</td>
</tr>
<tr>
<td>15h30-16h00</td>
<td><strong>Coffee break</strong></td>
</tr>
<tr>
<td>16h00-17h00</td>
<td><strong>PS3</strong>: UL HPC Monitoring in practice: why, what, how, where to look</td>
</tr>
<tr>
<td>17h00-18h00</td>
<td><strong>PS4</strong>: HPC workflow with Parallel/Distributed MPI jobs</td>
</tr>
</tbody>
</table>

### Advanced Parallel Track (MSA 3.100)

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>13h30-14h30</td>
<td><strong>PS5</strong>: Advanced Scheduling (Slurm, OAR)</td>
</tr>
<tr>
<td>14h30-15h30</td>
<td><strong>PS6</strong>: Debugging, profiling and performance analysis</td>
</tr>
<tr>
<td>15h30-16h00</td>
<td><strong>Coffee break</strong></td>
</tr>
<tr>
<td>16h00-17h00</td>
<td><strong>PS7</strong>: [Advanced] Prototyping with Python</td>
</tr>
<tr>
<td>17h00-18h00</td>
<td><strong>PS8</strong>: MATLAB (interactive, passive and sequential jobs)</td>
</tr>
</tbody>
</table>

**PS** = Practical Session using your laptop
Summary

1 Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
1 Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

HPC Strategy in Europe & Abroad

Conclusion & Perspectives
**Introduction**

**Prerequisites**

- **HPC**: High Performance Computing

---

**Main HPC Performance Metrics**

- **Computing Capacity/speed**: often measured in **flops** (or **flop/s**)  
  - Floating point operations per seconds (often in DP)  
  - $\text{GFlops} = 10^9$ Flops $\text{TFlops} = 10^{12}$ Flops $\text{PFlops} = 10^{15}$ Flops

- **Storage Capacity**: measured in multiples of **bytes** = 8 **bits**  
  - $\text{GB} = 10^9$ bytes $\text{TB} = 10^{12}$ bytes $\text{PB} = 10^{15}$ bytes  
  - $\text{GiB} = 1024^3$ bytes $\text{TiB} = 1024^4$ bytes $\text{PiB} = 1024^5$ bytes

- **Transfer rate** on a medium measured in **Mb/s** or **MB/s**

- **Other metrics**: Sequential vs Random **R/W speed**, **IOPS**...
Introduction

Why High Performance Computing?

The country that out-computes will be the one that out-competes  

Council on Competitiveness

- **Accelerates** research by accelerating **computation**

  \[ \sim 64 \text{ GFlops} \quad 206.772 \text{ TFlops} \]
  
  (Dual-core i5 2GHz) (602 computing nodes, 8452 cores)

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

  4 TB \quad 7952.4TB
  
  (1 disk, 250 MB/s) (2015 disks, 10 GB/s)

- **Communicates faster**

  1 GbE (1 Gb/s) vs Infiniband EDR (100 Gb/s)
Introduction

HPC at the Heart of our Daily Life

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

- Electro-Magnetics
- Computational Chemistry Quantum Mechanics
- Computational Chemistry Molecular Dynamics
- Computational Biology
- Structural Mechanics Implicit
- Seismic Processing
- Computational Fluid Dynamics
- Reservoir Simulation
- Rendering Ray Tracing
- Climate / Weather Ocean Simulation
- Data Analytics
- Structural Mechanics Explicit

• ... **Tomorrow:** applied research, digital health, nano/bio tech.

Ageing Medicine Biology

Materials Spintronic Nano-Sciences
Different HPC Needs per Domains

Material Science & Engineering

- #Cores
- Network Bandwidth
- Flops/Core
- Network Latency
- Storage Capacity
- I/O Performance
Introduction

Different HPC Needs per Domains

Biomedical Industry / Life Sciences

- #Cores
- Network Bandwidth
- Flops/Core
- Network Latency
- Storage Capacity
- I/O Performance

S. Varrette & al. (HPC © University of Luxembourg)
Introduction

Different HPC Needs per Domains

Deep Learning / Cognitive Computing

- #Cores
- Network Bandwidth
- Flops/Core
- Network Latency
- Storage Capacity
- I/O Performance

S. Varrette & al. (HPC @ University of Luxembourg)

UL HPC School 2017 [bis]
Different HPC Needs per Domains

IoT, FinTech

- #Cores
- Network Bandwidth
- Flops/Core
- Network Latency
- Storage Capacity
- I/O Performance
Introduction

Different HPC Needs per Domains

Material Science & Engineering
Biomedical Industry / Life Sciences
IoT, FinTech
Deep Learning / Cognitive Computing

#Cores
Network Bandwidth
I/O Performance
Flops/Core
Storage Capacity

ALL Research Computing Domains
Summary

1 Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
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 (VM) through an hypervisor**
  - Hypervisor: core virtualization engine / environment
    - Ex: Xen, VMWare ESXi, KVM, VirtualBox
    - Non-negligible Performance loss: ≥ 20%
Introduction

Computing for Researchers: Laptop

- **Regular PC / Local Laptop / Workstation**
  - Native OS (Windows, Linux, Mac etc.)
- **Virtualized OS (VM) through an hypervisor**
  - Hypervisor: core virtualization engine / environment
    - Ex: Xen, VMWare ESXi, KVM, VirtualBox
    - Non-negligible Performance loss: ≥ 20%
- **Container-based Virtualization**
  - similar to VMs . . .
    - yet containers **share** the system kernel of the host with others
    - Ex: Docker, Singularity, Shifter
Cloud Computing

- access to shared (generally virtualized) resources
- pay-per-use approach
- Infrastructure as a Service (IaaS)
Introduction

Computing for Researchers: Cloud

- **Cloud Computing**
  - access to shared *generally virtualized* resources
  - pay-per-use approach
  - **Platform** as a Service (PaaS)

![Diagram of IaaS and PaaS layers]

- **IaaS**
  - Guest OS (VM)
  - Operating System
  - Hardware & Network

- **PaaS**
  - User's application
  - Libraries & Softwares

- **User Control**
- **Provider Control**
Introduction

Computing for Researchers: Cloud

**Cloud Computing**

- access to shared (generally virtualized) resources
- pay-per-use approach
- **Software as a Service (SaaS)**

![Cloud Computing Diagram]

- **IaaS**
  - Guest OS (VM)
- **PaaS**
  - User's application
  - Libraries & Softwares
  - Operating System
- **SaaS**
  - User's data
  - Applications
  - Hardware & Network

User Control

Provider Control
Cloud Computing

- access to shared (generally virtualized) resources
- pay-per-use approach
- XXX as a Service ($<X>aaS$)
High Performance Computing (HPC) platforms

For **Speedup**, **Scalability** and **Faster Time to Solution**
Introduction

Computing for Researchers: HPC

- High Performance Computing (HPC) platforms
  - For Speedup, Scalability and Faster Time to Solution

YET...

PC ≠ Cloud ≠ HPC
Computing for Researchers: HPC

- High Performance Computing (HPC) platforms
  - For **Speedup**, **Scalability** and **Faster Time to Solution**

**YET...**

**PC ≠ Cloud ≠ HPC**

- HPC ≃ Formula 1
  - relies on ultra efficient hardware / interconnect (IB EDR...)  
  - ... when Cloud has to stay standard ([10] GbE etc...)

- **Does not mean the 3 approaches cannot work together**
Jobs, Tasks & Local Execution

$> ./myprog

Time

CPU 1
Core 1
Core 2
Introduction

Jobs, Tasks & Local Execution

```bash
$> ./myprog
```

CPU 1
Core 1
Core 2

Time
Introduction

Jobs, Tasks & Local Execution

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

CPU 1
Core 2

Time

S. Varrette & al. (HPC @ University of Luxembourg)
**Introduction**

**Jobs, Tasks & Local Execution**

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

![Diagram showing the execution of `myprog` with options]

Time
Jobs, Tasks & Local Execution

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

CPU 1
Core 2

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

Time
Jobs, Tasks & Local Execution

S. Varrette & al. (HPC © University of Luxembourg)
Introduction

Jobs, Tasks & Local Execution

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

CPU 1
Core 1
Core 2

T₁(local) = 100s

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

Time

S. Varrette & al. (HPC © University of Luxembourg)

UL HPC School 2017 [bis]
Introduction

Jobs, Tasks & Local Execution

Job(s)

Task(s)

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

CPU 1
Core 2
Core 1

Time

T₁ (local) = 100s

./myprog ./myprog -n 10 ./myprog -n 100
Introduction

Jobs, Tasks & Local Execution

# launcher
./myprog
./myprog -n 10
./myprog -n 100
# Jobs, Tasks & Local Execution

- # launcher
- `.myprog`
- `.myprog -n 10`
- `.myprog -n 100`

CPU 1
Core 1
Core 2

Time
Introduction

Jobs, Tasks & Local Execution

# launcher
.
./myprog
./myprog -n 10
./myprog -n 100

CPU 1
Core 1
Core 2

./myprog ./myprog -n 10

Time
# Jobs, Tasks & Local Execution

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

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

CPU 1

Core 1

Core 2

Time
Introduction

Jobs, Tasks & Local Execution

# launcher
./myprog
./myprog -n 10
./myprog -n 100

T₁(local) = 100s

./myprog  ./myprog -n 10  ./myprog -n 100
Introduction

Jobs, Tasks & Local Execution

# launcher
./myprog
./myprog -n 10
./myprog -n 100

T_1 (local) = 100s

CPU 1
Core 1
Core 2

Time
Introduction

Jobs, Tasks & Local Execution

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

CPU 1

Core 1

Core 2

Time
Introduction

Jobs, Tasks & Local Execution

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

CPU 1

Core 1

Core 2

Time

S. Varrette & al. (HPC © University of Luxembourg)
Introduction

Jobs, Tasks & Local Execution

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

CPU 1

- Core 1
- Core 2

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

\[ T_2(\text{local}) = 70s \]
Jobs, Tasks & Local Execution

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

CPU 1

Time

```
Core 1
```

```
Core 2
```

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

\[ T_2(\text{local}) = 70s \]
# Jobs, Tasks & HPC Execution

```bash
# launcher
./myprog
./myprog -n 10
./myprog -n 100
```
Introduction

Jobs, Tasks & HPC Execution

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

```
T_1(hpc) = T_8(hpc) = 120s
```

```
./myprog . ./myprog -n 10 . ./myprog -n 100
```
Jobs, Tasks & HPC Execution

Introduction

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

```
1
```

```
Task(s)
```

```
T_1(hpc) = T_8(hpc) = 120s
```

Node 1
- CPU 1
  - Core 1
  - Core 2
- CPU 2
  - Core 3
  - Core 4

Node 2
- CPU 1
  - Core 1
  - Core 2
- CPU 2
  - Core 3
  - Core 4

Time

S. Varrette & al. (HPC © University of Luxembourg)
Introduction

Jobs, Tasks & HPC Execution

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

Node 1
CPU 1
Core 1
Core 2
CPU 2
Core 3
Core 4

Node 2
CPU 2
Core 1
Core 2
CPU 1
Core 3
Core 4

Time
Introduction

Jobs, Tasks & HPC Execution

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

```
Node 1
CPU 1
Core 1
Core 2

CPU 2
Core 3
Core 4

Node 2
CPU 1
Core 1
Core 2

CPU 2
Core 3
Core 4
```

\[ T_2(hpc) = 80s \]
Introduction

Jobs, Tasks & HPC Execution

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

```
Node 1
CPU 1
Core 1
Core 2
CPU 2
Core 3
Core 4

Node 2
CPU 2
Core 1
Core 2
CPU 1
Core 3
Core 4
```

\[ T_2(hpc) = 80s \]
Introduction

Jobs, Tasks & HPC Execution

```
T_8(hpc) = 60s
```

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

```
Task(s)

1

2

3
```

```
Job(s)

1

2

3
```

```
Node 1
CPU 1
Core 1
Core 2
Core 3
Core 4

Node 2
CPU 2
Core 1
Core 2
Core 3
Core 4
```

```
Time

16 / 84
```
## Local vs. HPC Executions

<table>
<thead>
<tr>
<th>Context</th>
<th>Local PC</th>
<th>HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>( T_1(\text{local}) = 100 )</td>
<td>( T_1(\text{hpc}) = 120 \text{s} )</td>
</tr>
<tr>
<td>Parallel/Distributed</td>
<td>( T_2(\text{local}) = 70 \text{s} )</td>
<td>( T_2(\text{hpc}) = 80 \text{s} )</td>
</tr>
<tr>
<td></td>
<td>( T_8(\text{hpc}) = 60 \text{s} )</td>
<td></td>
</tr>
</tbody>
</table>

Sequential runs WON'T BE FASTER on HPC → Reason: Processor Frequency (typically \( \geq 3 \text{GHz} \) vs \( \geq 2 \text{GHz} \))

Parallel/Distributed runs DO NOT COME FOR FREE → runs will be sequential even if you reserve \( \geq 2 \) cores/nodes → you have to explicitly adapt your jobs to benefit from the multi-cores/nodes.
## Local vs. HPC Executions

<table>
<thead>
<tr>
<th>Context</th>
<th>Local PC</th>
<th>HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>$T_1(\text{local}) = 100$</td>
<td>$T_1(\text{hpc}) = 120s$</td>
</tr>
<tr>
<td>Parallel/Distributed</td>
<td>$T_2(\text{local}) = 70s$</td>
<td>$T_2(\text{hpc}) = 80s$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_8(\text{hpc}) = 60s$</td>
</tr>
</tbody>
</table>

- **Sequential** runs **WON’T BE FASTER** on HPC
  - **Reason:** Processor Frequency (typically $\geq 3\text{GHz}$ vs $\geq 2\text{GHz}$)
## Introduction

### Local vs. HPC Executions

<table>
<thead>
<tr>
<th>Context</th>
<th>Local PC</th>
<th>HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>$T_1(\text{local}) = 100$</td>
<td>$T_1(\text{hpc}) = 120s$</td>
</tr>
<tr>
<td></td>
<td>$T_2(\text{local}) = 70s$</td>
<td>$T_2(\text{hpc}) = 80s$</td>
</tr>
<tr>
<td></td>
<td>$T_8(\text{hpc}) = 60s$</td>
<td></td>
</tr>
</tbody>
</table>

- **Sequential** runs **WON’T BE FASTER** on HPC
  - Reason: Processor Frequency (typically $\geq 3\text{GHz}$ vs $\geq 2\text{GHz}$)

- **Parallel/Distributed** runs **DO NOT COME FOR FREE**
  - Runs will be sequential even if you reserve $\geq 2\text{ cores/nodes}$
  - You have to **explicitly** adapt your jobs to benefit from the multi-cores/nodes
Introduction

Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

HPC Strategy in Europe & Abroad

Conclusion & Perspectives
**HPC Components: [GP]CPU**

### CPU
- Always multi-core
- **Ex:** Intel Core i7-7700K (Jan 2017) \( R_{peak} \simeq 268.8 \text{ GFlops (DP)} \)
  - 4 cores @ 4.2GHz (14nm, 91W, 1.75 billion transistors)
  - + integrated graphics (24 EUs) \( R_{peak} \simeq +441.6 \text{ GFlops} \)

### GPU / GPGPU
- Always multi-core, optimized for vector processing
- **Ex:** Nvidia Tesla V100 (Jun 2017) \( R_{peak} \simeq 7 \text{ TFlops (DP)} \)
  - 5120 cores @ 1.3GHz (12nm, 250W, 21 billion transistors)
  - focus on Deep Learning workloads \( R_{peak} \simeq 112 \text{ TFLOPS (HP)} \)

\(~ 100 \text{ Gflops} \text{ for 130$ (CPU), 214$? (GPU)}\)
Introduction

HPC Components: Local Memory

Larger, slower and cheaper

<table>
<thead>
<tr>
<th>Level</th>
<th>Size</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500 bytes</td>
<td>sub ns</td>
</tr>
<tr>
<td>2</td>
<td>64 KB to 8 MB</td>
<td>1-2 cycles</td>
</tr>
<tr>
<td>3</td>
<td>1 GB</td>
<td>10 cycles</td>
</tr>
<tr>
<td>4</td>
<td>1 TB</td>
<td>20 cycles</td>
</tr>
</tbody>
</table>

- **SSD (SATA3)** R/W: 550 MB/s; 100000 IOPS
- **HDD (SATA3 @ 7,2 krpm)** R/W: 227 MB/s; 85 IOPS

450 €/TB

54 €/TB

S. Varrette & al. (HPC © University of Luxembourg)
**Introduction**

**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

<table>
<thead>
<tr>
<th>Technology</th>
<th>Effective Bandwidth</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigabit Ethernet</td>
<td>1 Gb/s</td>
<td>125 MB/s</td>
</tr>
<tr>
<td>10 Gigabit Ethernet</td>
<td>10 Gb/s</td>
<td>1.25 GB/s</td>
</tr>
<tr>
<td>Infiniband QDR</td>
<td>40 Gb/s</td>
<td>5 GB/s</td>
</tr>
<tr>
<td>Infiniband EDR</td>
<td>100 Gb/s</td>
<td>12.5 GB/s</td>
</tr>
<tr>
<td>100 Gigabit Ethernet</td>
<td>100 Gb/s</td>
<td>1.25 GB/s</td>
</tr>
<tr>
<td>Intel Omnipath</td>
<td>100 Gb/s</td>
<td>12.5 GB/s</td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th>Technology</th>
<th>Effective Bandwidth</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigabit Ethernet</td>
<td>1 Gb/s</td>
<td>125 MB/s</td>
</tr>
<tr>
<td>10 Gigabit Ethernet</td>
<td>10 Gb/s</td>
<td>1.25 GB/s</td>
</tr>
<tr>
<td>Infiniband QDR</td>
<td>40 Gb/s</td>
<td>5 GB/s</td>
</tr>
<tr>
<td>Infiniband EDR</td>
<td>100 Gb/s</td>
<td>12.5 GB/s</td>
</tr>
<tr>
<td>100 Gigabit Ethernet</td>
<td>100 Gb/s</td>
<td>1.25 GB/s</td>
</tr>
<tr>
<td>Intel Omnipath</td>
<td>100 Gb/s</td>
<td>12.5 GB/s</td>
</tr>
</tbody>
</table>

(Note: Specific values and units are approximate and subject to manufacturing tolerances.)

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

Network Topologies

- **Direct vs. Indirect** interconnect
  - *direct*: each network node attaches to at least one compute node
  - *indirect*: compute nodes attached at the edge of the network only
    - ✓ many routers only connect to other routers.

- **Main HPC Topologies**
  - **CLOS Network Fat-Trees** (Indirect)
    - can be fully non-blocking (1:1) or blocking (x:1)
    - typically enables best performance
    - ✓ Non blocking bandwidth, lowest network latency
  - **Mesh 3D-torus** (Direct)
    - Blocking network, cost-effective for systems at scale
    - ✓ Great performance solutions for applications with locality
    - ✓ Simple expansion for future growth
Introduction

Network Topologies

- **Direct** vs. **Indirect** interconnect
  - direct: each network node attaches to at least one compute node
  - indirect: compute nodes attached at the edge of the network only
    - many routers only connect to other routers.

Main HPC Topologies

- **CLOS Network / Fat-Trees** [Indirect]
  - can be fully non-blocking (1:1) or blocking (x:1)
  - typically enables **best performance**
    - Non blocking bandwidth, lowest network latency
Introduction

Network Topologies

- **Direct vs. Indirect** interconnect
  - *direct*: each network node attaches to at least one compute node
  - *indirect*: compute nodes attached at the edge of the network only
    - ✓ many routers only connect to other routers.

Main HPC Topologies

- **CLOS Network / Fat-Trees** [Indirect]
  - can be fully non-blocking (1:1) or blocking (x:1)
  - typically enables **best performance**
    - ✓ Non blocking bandwidth, lowest network latency

- **Mesh or 3D-torus** [Direct]
  - Blocking network, cost-effective for systems at scale
  - Great performance solutions for applications with locality
  - Simple expansion for future growth
Introduction

HPC Components: Operating System

- Exclusively Linux-based (99.6%)
- ... or Unix (0.4%)

Reasons:
- stability
- prone to devals

[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...
Introduction

[Big]Data Management: Disk Encl.

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

[Big]Data Management: FS Summary

- **File System (FS)**: Logical manner to store, organize & access data
  - (local) **Disk FS**: FAT32, NTFS, HFS+, ext4, \{x,z,btr\}fs...
  - **Networked FS**: NFS, CIFS/SMB, AFP
  - **Parallel/Distributed FS**: SpectrumScale/GPFS, Lustre
    - typical FS for HPC / HTC (High Throughput Computing)
[Big]Data Management: FS Summary

- **File System (FS):** Logical manner to store, organize & access data
  - (local) **Disk FS:** FAT32, NTFS, HFS+, ext4, {x,z,btr}fs...
  - **Networked FS:** NFS, CIFS/SMB, AFP
  - **Parallel/Distributed FS:** SpectrumScale/GPFS, Lustre
    - typical FS for HPC / HTC (High Throughput Computing)

**Main Characteristic of Parallel/Distributed File Systems**

*Capacity and Performance* increase with *#servers*
**File System (FS):** Logical manner to store, organize & access data

- **(local) Disk FS:** FAT32, NTFS, HFS+, ext4, \{x,z,btr\}fs...
- **Networked FS:** NFS, CIFS/SMB, AFP
- **Parallel/Distributed FS:** SpectrumScale/GPFS, Lustre
  ✓ typical FS for HPC / HTC (High Throughput Computing)

---

**Main Characteristic of Parallel/Distributed File Systems**

**Capacity and Performance** increase with \#servers

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Read* [GB/s]</th>
<th>Write* [GB/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext4</td>
<td>Disk FS</td>
<td>0.426</td>
<td>0.212</td>
</tr>
<tr>
<td>nfs</td>
<td>Networked FS</td>
<td>0.381</td>
<td>0.090</td>
</tr>
<tr>
<td>gpfs (iris)</td>
<td>Parallel/Distributed FS</td>
<td>10.14</td>
<td>8.41</td>
</tr>
<tr>
<td>gpfs (gaia)</td>
<td>Parallel/Distributed FS</td>
<td>7.74</td>
<td>6.524</td>
</tr>
<tr>
<td>lustre</td>
<td>Parallel/Distributed FS</td>
<td>4.5</td>
<td>2.956</td>
</tr>
</tbody>
</table>

* maximum random read/write, per IOZone or IOR measures, using 15 concurrent nodes for networked FS.
Introduction

HPC Components: Data Center

**Definition (Data Center)**

- Facility to house computer systems and associated components
  - Basic storage component: *rack* (height: 42 RU)
Introduction

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: 30-120 kW
  - Storage racks: 15 kW
  - Interconnect racks: 5 kW

Various Cooling Technology

- Airflow
- Direct-Liquid Cooling, Immersion...

Power Usage Effectiveness

\[ PUE = \frac{\text{Total facility power}}{\text{IT equipment power}} \]
Summary

1 Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
Summary

1 Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
High Performance Computing (HPC) @ UL

Started in 2007, under resp. of Prof P. Bouvry & Dr. S. Varrette

→ expert UL HPC team (S. Varrette, V. Plugaru, S. Peter, H. Cartiaux, C. Parisot)

→ 8,173,747€ cumulative investment in hardware

Key numbers

- 416 users
- 602 computing nodes
  - 8452 cores, 206.772 TFlops
  - 50 accelerators (+ 76.22 TFlops)
- 7952.4 TB storage
- 130 (+ 71) servers
- 5 sysadmins
- 2 sites: Kirchberg / Belval

http://hpc.uni.lu

S. Varrette & al. (HPC © University of Luxembourg)
**Enables & accelerates** scientific discovery and innovation

**Largest facility** in Luxembourg (after GoodYear R&D Center)

<table>
<thead>
<tr>
<th>Country</th>
<th>Institute</th>
<th>#Nodes</th>
<th>#Cores</th>
<th>$R_{peak}$</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luxembourg</td>
<td>UL HPC (Uni.lu) LIST</td>
<td>602</td>
<td>8452</td>
<td>206.772</td>
<td>7952.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58</td>
<td>800</td>
<td>6.21</td>
<td>144</td>
</tr>
<tr>
<td>France</td>
<td>LORIA (G5K), Nancy ROMEO, Reims</td>
<td>320</td>
<td>2520</td>
<td>26.98</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>174</td>
<td>3136</td>
<td>49.26</td>
<td>245</td>
</tr>
<tr>
<td>Belgium</td>
<td>NIC4, University of Liège, Université Catholique de Louvain UGent / VSC, Gent</td>
<td>128</td>
<td>2048</td>
<td>32.00</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>112</td>
<td>1344</td>
<td>13.28</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td></td>
<td>440</td>
<td>8768</td>
<td>275.30</td>
<td>1122</td>
</tr>
<tr>
<td>Germany</td>
<td>bwGrid, Heidelberg bwForCluster, Ulm</td>
<td>140</td>
<td>1120</td>
<td>12.38</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>444</td>
<td>7104</td>
<td>266.40</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>bwHPC MLS&amp;WISO, Mannheim</td>
<td>604</td>
<td>9728</td>
<td>371.60</td>
<td>420</td>
</tr>
</tbody>
</table>
High Performance Computing (HPC) @ UL

UL HPC User Base

416 Active HPC Users

Evolution of registered users within UL internal clusters

- LCSB (Bio-Medicine)
- URPM (Physics and Material Sciences)
- FDEF (Law, Economics and Finance)
- RUES (Engineering Science)
- SnT (Security and Trust)
- CSC (Computer Science and Communications)
- LSRU (Life Sciences)
- Bachelor and Master students
- Other UL users (small groups aggregated)
- External partners

Number of users

23 computational domains accelerated on UL HPC

- for the UL Faculties, Research Units and Interdisciplinary Centres
  - incl. LCSB, SnT... and now C2DH thematics
  - UL strategic research priorities
    - computational sciences, finance (fintech)
    - systems biomedicine, security, reliability and trust

- UL HPC feat. special systems targeting specific workloads:
  - Machine Learning & AI: GPU accelerators
    - 10 Tesla K40 + 16 Tesla K80 + 24 Tesla M20*: 76 GPU Tflops
  - BigData analytics & data driven science: large memory systems
    - Large SMP systems with 1, 2, 3 & 4 TB RAM
  - Scale-out workloads: energy efficient systems
    - 90 HP Moonshot servers + 96 viridis ARM-based systems
>140 software packages available for researchers

- General purpose, statistics, optimization:
  - Matlab, Mathematica, R, Stata, CPLEX, Gurobi Optimizer...

- Bioinformatics
  - BioPython, STAR, TopHat, Bowtie, mpiHMMER...

- Computer aided engineering:
  - ANSYS, ABAQUS, OpenFOAM...

- Molecular dynamics:
  - NAMD, ABINIT, Q.ESPRESSO, GROMACS...

- Visualisation: ParaView, VisIt, VMD, XCS portal
- Compilers, libraries, performance modeling tools
- [Parallel] debugging tools aiding development

https://hpc.uni.lu/users/software/
High Performance Computing (HPC) @ UL

UL HPC Team

Prof. Pascal Bouvry
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 Associate (CSC, FSTC)

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

Hyacinthe Cartiaux
CDI, Support (SIU)

Clément Parisot
CDI, Support (CSC, FSTC)
Sites / Data centers

Kirchberg
CS.43, AS. 28

Belval
Biotech I, CDC/MSA

2 sites, ≥ 4 server rooms
High Performance Computing (HPC) @ UL

Sites / Data centers

Kirchberg
CS.43, AS. 28

Belval
Biotech I, CDC/MSA

2 sites, ≥ 4 server rooms
High Performance Computing (HPC) @ UL

UL HPC: General cluster organization

- Site <sitename>
- Other Clusters network
- Local Institution Network
- Fast local interconnect (Infiniband EDR) 100 Gb/s
- [Redundant] Load balancer
- [Redundant] Site access server(s)
- [Redundant] Adminfront(s)
- GPFS / Lustre
- Disk Enclosures
- Site Shared Storage Area
- Slurm
- Site router
- 10/40 GbE QSFP+
- 10 GbE

S. Varrette & al. (HPC © University of Luxembourg)
UL HPC Computing capacity

5 clusters
206,772 TFlops
602 nodes
8452 cores
34512GPU cores
## UL HPC Computing Clusters

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Location</th>
<th>#N</th>
<th>#C</th>
<th>Rpeak</th>
<th>GPU Rpeak</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>CDC S-01</td>
<td>108</td>
<td>3024</td>
<td>116.12</td>
<td>0</td>
</tr>
<tr>
<td>gaia</td>
<td>BT1</td>
<td>273</td>
<td>3440</td>
<td>69.296</td>
<td>76</td>
</tr>
<tr>
<td>chaos</td>
<td>Kirchberg</td>
<td>81</td>
<td>1120</td>
<td>14.495</td>
<td>0</td>
</tr>
<tr>
<td>g5k</td>
<td>Kirchberg</td>
<td>38</td>
<td>368</td>
<td>4.48</td>
<td>0</td>
</tr>
<tr>
<td>nyx (experimental)</td>
<td>BT1</td>
<td>102</td>
<td>500</td>
<td>2.381</td>
<td>0</td>
</tr>
</tbody>
</table>

**TOTAL:** 602 8452 206.772 + 76 TFlops
### UL HPC - Detailed Computing Nodes

<table>
<thead>
<tr>
<th>Date</th>
<th>Vendor</th>
<th>Proc. Description</th>
<th>#N</th>
<th>#C</th>
<th>R_{peak}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017</td>
<td>Dell</td>
<td>Intel Xeon E5-2680 <a href="mailto:v4@2.4GHz">v4@2.4GHz</a></td>
<td>108</td>
<td>3024</td>
<td>116.12 TFlops</td>
</tr>
</tbody>
</table>

**iris TOTAL:** 108 3024 116.12 TFlops

<table>
<thead>
<tr>
<th>Date</th>
<th>Vendor</th>
<th>Proc. Description</th>
<th>#N</th>
<th>#C</th>
<th>R_{peak}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>Bull</td>
<td>Intel Xeon <a href="mailto:L5640@2.26GHz">L5640@2.26GHz</a></td>
<td>72</td>
<td>864</td>
<td>7.811 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-4640@2.4GHz">E5-4640@2.4GHz</a></td>
<td>1</td>
<td>32</td>
<td>0.614 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>Bull</td>
<td>Intel Xeon E7-4850@2GHz</td>
<td>1</td>
<td>160</td>
<td>1.280 TFlops</td>
</tr>
<tr>
<td>2013</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-2660@2.2GHz">E5-2660@2.2GHz</a></td>
<td>5</td>
<td>80</td>
<td>1.408 TFlops</td>
</tr>
<tr>
<td>2013</td>
<td>Bull</td>
<td>Intel Xeon <a href="mailto:X5670@2.93GHz">X5670@2.93GHz</a></td>
<td>40</td>
<td>480</td>
<td>5.626 TFlops</td>
</tr>
<tr>
<td>2013</td>
<td>Bull</td>
<td>Intel Xeon <a href="mailto:X5675@3.07GHz">X5675@3.07GHz</a></td>
<td>32</td>
<td>384</td>
<td>4.746 TFlops</td>
</tr>
<tr>
<td>2014</td>
<td>Delta</td>
<td>Intel Xeon E7-8880@2.5 GHz</td>
<td>1</td>
<td>120</td>
<td>2.4 TFlops</td>
</tr>
<tr>
<td>2014</td>
<td>SGi</td>
<td>Intel Xeon E5-4650@2.4 GHz</td>
<td>1</td>
<td>160</td>
<td>3.072 TFlops</td>
</tr>
<tr>
<td>2015</td>
<td>Dell</td>
<td>Intel Xeon E5-2660@2.5 GHz</td>
<td>28</td>
<td>672</td>
<td>26.88 TFlops</td>
</tr>
<tr>
<td>2015</td>
<td>HP</td>
<td>Intel E3-1284Lv3, 1.8GHz</td>
<td>90</td>
<td>360</td>
<td>10.368 TFlops</td>
</tr>
<tr>
<td>2016</td>
<td>Dell</td>
<td>Intel Xeon E7-8867@2.5 GHz</td>
<td>4</td>
<td>128</td>
<td>5.12 TFlops</td>
</tr>
</tbody>
</table>

**gaia TOTAL:** 273 3440 69.296 TFlops

<table>
<thead>
<tr>
<th>Date</th>
<th>Vendor</th>
<th>Proc. Description</th>
<th>#N</th>
<th>#C</th>
<th>R_{peak}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>HP</td>
<td>Intel Xeon <a href="mailto:L5640@2.26GHz">L5640@2.26GHz</a></td>
<td>32</td>
<td>384</td>
<td>3.472 TFlops</td>
</tr>
<tr>
<td>2011</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:L5640@2.26GHz">L5640@2.26GHz</a></td>
<td>16</td>
<td>192</td>
<td>1.736 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:X7560@2.26GHz">X7560@2.26GHz</a></td>
<td>1</td>
<td>32</td>
<td>0.289 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-2660@2.2GHz">E5-2660@2.2GHz</a></td>
<td>16</td>
<td>256</td>
<td>4.506 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>HP</td>
<td>Intel Xeon <a href="mailto:E5-2660@2.2GHz">E5-2660@2.2GHz</a></td>
<td>16</td>
<td>256</td>
<td>4.506 TFlops</td>
</tr>
</tbody>
</table>

**chaos TOTAL:** 81 1120 14.495 TFlops

<table>
<thead>
<tr>
<th>Date</th>
<th>Vendor</th>
<th>Proc. Description</th>
<th>#N</th>
<th>#C</th>
<th>R_{peak}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Dell</td>
<td>Intel Xeon L5335@2GHz</td>
<td>22</td>
<td>176</td>
<td>1.408 TFlops</td>
</tr>
<tr>
<td>2012</td>
<td>Dell</td>
<td>Intel Xeon E5-2630L@2GHz</td>
<td>16</td>
<td>192</td>
<td>3.072 TFlops</td>
</tr>
</tbody>
</table>

**g5k**

**granduc/petitprince TOTAL:** 38 368 4.48 TFlops

<table>
<thead>
<tr>
<th>Date</th>
<th>Vendor</th>
<th>Proc. Description</th>
<th>#N</th>
<th>#C</th>
<th>R_{peak}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-2420@1.9GHz">E5-2420@1.9GHz</a></td>
<td>2</td>
<td>12</td>
<td>0.091 TFlops</td>
</tr>
<tr>
<td>2013</td>
<td>Viridis</td>
<td>ARM A9 <a href="mailto:Cortex@1.1GHz">Cortex@1.1GHz</a></td>
<td>96</td>
<td>384</td>
<td>0.422 TFlops</td>
</tr>
<tr>
<td>2015</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-2630Lv2@2.4GHz">E5-2630Lv2@2.4GHz</a></td>
<td>2</td>
<td>24</td>
<td>0.460 TFlops</td>
</tr>
<tr>
<td>2015</td>
<td>Dell</td>
<td>Intel Xeon <a href="mailto:E5-2660v2@2.2GHz">E5-2660v2@2.2GHz</a></td>
<td>4</td>
<td>80</td>
<td>1.408 TFlops</td>
</tr>
</tbody>
</table>

**nyx/viridis TOTAL:** 102 500 2.381 TFlops
High Performance Computing (HPC) @ UL

UL HPC Storage capacity

4 distributed/parallel FS
2015 disks
7952.4 TB
(incl. 2116TB for Backup)
### UL HPC Shared Storage Capacities

<table>
<thead>
<tr>
<th>Cluster</th>
<th>GPFS</th>
<th>Lustre</th>
<th>Other (NFS...)</th>
<th>Backup</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>1440</td>
<td>0</td>
<td>6</td>
<td>600</td>
<td>2046 TB</td>
</tr>
<tr>
<td>gaia</td>
<td>960</td>
<td>480</td>
<td>0</td>
<td>1336</td>
<td>2776 TB</td>
</tr>
<tr>
<td>chaos</td>
<td>0</td>
<td>0</td>
<td>180</td>
<td>180</td>
<td>360 TB</td>
</tr>
<tr>
<td>g5k</td>
<td>0</td>
<td>0</td>
<td>32.4</td>
<td>0</td>
<td>32.4 TB</td>
</tr>
<tr>
<td>nyx (experimental)</td>
<td>0</td>
<td>0</td>
<td>242</td>
<td>0</td>
<td>242 TB</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td><strong>2400</strong></td>
<td><strong>480</strong></td>
<td><strong>2956.4</strong></td>
<td><strong>2116</strong></td>
<td><strong>7952.4 TB</strong></td>
</tr>
</tbody>
</table>
UL HPC Software Stack

- **Operating System:** Linux CentOS 7 (iris), Debian 8 (others)
- **Remote connection to the platform:** SSH
- **User SSO:** IPA, OpenLDAP
- **Resource management:** job/batch scheduler: Slurm (iris), OAR
- **(Automatic) Computing Node Deployment:**
  - FAI (Fully Automatic Installation) (gaia, chaos clusters)
  - Bright Cluster Manager (iris)
  - Puppet
  - Kadeploy
- **Platform Monitoring:**
  - OAR Monika/Drawgantt, Ganglia, Allinea Perf Report, SLURM
  - Icinga, NetXMS, PuppetBoard etc.
- **Commercial Softwares:**
  - ANSYS, ABAQUS, MATLAB, Intel Cluster Studio XE, Allinea DDT, Stata etc.
High Performance Computing (HPC) @ UL

The case of Grid’5000

- 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**
High Performance Computing (HPC) @ UL

The case of Grid’5000

• 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
CPU-year usage since 2010

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

---

Platform Yearly CPU Used

<table>
<thead>
<tr>
<th>Year</th>
<th>CPU Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>56</td>
</tr>
<tr>
<td>2011</td>
<td>378</td>
</tr>
<tr>
<td>2012</td>
<td>612</td>
</tr>
<tr>
<td>2013</td>
<td>1067</td>
</tr>
<tr>
<td>2014</td>
<td>1417</td>
</tr>
<tr>
<td>2015</td>
<td>2255</td>
</tr>
<tr>
<td>2016</td>
<td>2430</td>
</tr>
</tbody>
</table>
Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

HPC Strategy in Europe & Abroad

Conclusion & Perspectives
Computing nodes Management

Node deployment by FAI/Bright Manager

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

```
DHCP
Daemon

TFTP
Server

NFS
Server

DHCP request, send MAC address
get IP address, netmask, gateway

send TFTP request for kernel image
get install kernel and boot it

mount nfsroot by install kernel
```

Install server

Install client
Computing nodes Management

Node deployment by FAI/Bright Manager

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

- Get configuration data (NFS/other)

install server

- nfsroot
  - mounted by install kernel

- config space
  - ./hooks
  - ./class
  - ./disk_config
  - ./package_config
  - ./scripts
  - ./files

install client

- /target/
  - /usr
  - /bin
  - /var

Debian mirror

mounted by install kernel

NFS, CVS, svn or HTTP

provided via HTTP, FTP or NFS

local hard disk
Node deployment by FAI/Bright Manager

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

- Get configuration data (NFS/other)

- 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/Bright Manager

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

- Get configuration data (NFS/other)

- 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: \( \approx 500 \text{s} \)
**Server/Service configuration by Puppet**

- **IT Automation** for configuration management
  - idempotent
  - agent/master OR stand-alone architecture
  - cross-platform through Puppet Resource Abstraction Layer (RAL)
  - Git-based workflow
  - PKI-based security (X.509)

- **DevOps** tool of choice for configuration management
  - Declarative Domain Specific Language (DSL)
Server/Service configuration by Puppet

- **IT Automation** for configuration management
  - idempotent
  - agent/master OR stand-alone architecture
  - cross-platform through Puppet 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:** ≃ 3-6 min
General Puppet Infrastructure

Puppet Master
- Client descriptions
- testing
- production
dele
- Environments
- Modules/Manifests
- Certificate Authority

PuppetDB / dashboard

Files

MCollective / ActiveMQ or XMLRPC/REST over SSL

CentOS
Puppet agent
Puppet agent
Puppet agent

debian
redhat

Windows
Puppet agent
Puppet agent
Puppet agent

ubuntu

Client Site A
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:

- > 140 software packages, in multiple versions, within 18 categ.
- reworked software set for iris cluster and now deployed everywhere
  - RESIF v2.0, allowing [real] semantic versioning of released builds
- hierarchical organization
  - Ex: toolchain/{foss,intel}

```bash
$> module avail # List available modules
```

```bash
$> module load <category>/<software>[/<version>]
```
Easybuild: open-source framework to (automatically) build scientific software

Why?: "Could you please install this software on the cluster?"

→ Scientific software is often difficult to build
  ✓ non-standard build tools / incomplete build procedures
  ✓ hardcoded parameters and/or poor/outdated documentation

→ EasyBuild helps to facilitate this task
  ✓ consistent software build and installation framework
  ✓ includes testing step that helps validate builds
  ✓ automatically generates LMod modulefiles

$>\text{module use }/\text{path/to/easybuild}$
$>\text{module load tools/EasyBuild}$
$>\text{eb }-\text{S HPL} \quad \# \text{ Search for recipes for HPL software}$
$>\text{eb }\text{HPL-2.2-intel-2017a.eb} \quad \# \text{ Install HPL 2.2 w. Intel toolchain}$
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

Platform Monitoring

- **General Live Status**

  [http://hpc.uni.lu/status/overview.html/](http://hpc.uni.lu/status/overview.html/)
Platform Monitoring

Monika

http://hpc.uni.lu/{gaia,chaos,g5k}/monika
Platform Monitoring

- Drawgantt

http://hpc.uni.lu/{gaia,chaos,g5k}/drawgantt
Platform Monitoring

- **Ganglia**

  http://hpc.uni.lu/{gaia,chaos,g5k,iris}/ganglia
High Performance Computing (HPC) @ UL

Platform Monitoring

**SLURM-Web**

http://hpc.uni.lu/iris/slurm/
Platform Monitoring

- **CDash**

http://cdash.uni.lu/?
Platform Monitoring

- **Internal Monitoring**

Icinga / Puppet / NetXMS (networking)
High Performance Computing (HPC) @ UL

Platform Monitoring

- Internal Monitoring

[Disk] Enclosure status
Summary

1 Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
Installation of the new *iris* cluster

- **108 nodes**, 3024 cores, 116.12 TFlops
- **Dell C6320**, Intel Xeon E5-2680v4@2.4 GHz [2x14c], 128 GB RAM
- **SpectrumsScale GPFS**: 1440 TB raw
Beyond *iris* setup, we *introduced several new elements*

- Migration to **SLURM**
- Consolidation of the High Availability (HA) setup
- **Improved system automation** (Puppet 4.x / hiera)
Beyond iris setup, we introduced several new elements

- Migration to SLURM
- Consolidation of the High Availability (HA) setup
- Improved system automation (Puppet 4.x / hiera)

Continuous OS / software modules Upgrade

- Migration to Debian 8 on gaia and chaos
- RESIF v2, updated software sets
Beyond iris setup, we introduced several new elements
- Migration to SLURM
- Consolidation of the High Availability (HA) setup
- Improved system automation (Puppet 4.x / hiera)

Continuous OS / software modules Upgrade
- Migration to Debian 8 on gaia and chaos
- RESIF v2, updated software sets

Storage:
- Spectrumscale/GPFS Upgrades across all clusters (to 4.2.2.X)
- Lustre (2.7.x): saturated (> 95% storage capacity) thus unstable
  - update to new LTS version 2.10.X not possible without formatting

Drastic measures required for Lustre!!
- PENDING formatting and new quota policy
Back to 2017 Achievements

- **Feb.  2017 (HR):** Clement Parisot joined the UL HPC Team
- **June. 2017:** UL HPC School
Back to 2017 Achievements

- **Feb. 2017 (HR):** Clement Parisot joined the UL HPC Team
- **June. 2017:** UL HPC School
- **Mar-Sep 2017:** **UL crisis**
  - ... yet UL HPC Budget line maintained and part of 4Y plan
  - induced many delay in the delivery of CDC S-02
  - impact on UL credibility in general
High Performance Computing (HPC) @ UL

Back to 2017 Achievements

- **Feb. 2017 (HR)**: Clement Parisot joined the UL HPC Team
- **June. 2017**: UL HPC School
- **Mar-Sep 2017**: UL crisis
  - ... yet UL HPC Budget line maintained and part of 4Y plan
  - induced many delay in the delivery of CDC S-02
  - impact on UL credibility in general

- **Oct. 2017**: Luxembourg 25th country to join PRACE
  - Official representatives for Luxembourg from UL
    - Delegate: Prof. Pascal Bouvry
    - Advisor: Dr. Sebastien Varrette
High Performance Computing (HPC) @ UL

Back to 2017 Achievements

- **Feb. 2017 (HR):** Clement Parisot joined the UL HPC Team
- **June. 2017:** UL HPC School
- **Mar-Sep 2017: UL crisis**
  - yet UL HPC Budget line maintained and part of 4Y plan
  - induced many delay in the delivery of CDC S-02
  - impact on UL credibility in general

- **Oct. 2017:** Luxembourg 25th country to join PRACE
  - Official representatives for Luxembourg from UL
    - Delegate: Prof. Pascal Bouvry
    - Advisor: Dr. Sebastien Varrette

- **Still pending:**
  - official structure enabling **Research computing @ UL & abroad**
  - National commitment for having next-gen HPC in CDC S-02
Summary

1 Introduction
Preliminaries
[Parallel] Computing for Enhanced Research
Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
Overview
Platform Management
Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
General Considerations
Environment & Typical Workflow Overview
Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
Summary

1 Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives
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 too many simultaneous file transfers
    - regularly clean your directories from useless files
  - **Do not 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
  - **Acknowledgement**
    - Acknowledge / cite the UL HPC facility (using official banner)
    - Tag your publication upon registration on ORBiLu
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 too many simultaneous file transfers
    - regularly clean your directories from useless files
  - **Do not 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.
Summary

1. Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2. High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3. UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4. HPC Strategy in Europe & Abroad

5. Conclusion & Perspectives
OS: Debian 8 / CentOS 7

Storage usage: df-ulhpc

Env. modules: modules

→ Not available on frontends
→ *Only* on compute nodes

<table>
<thead>
<tr>
<th>Directory</th>
<th>Max size</th>
<th>Max #files</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME</td>
<td>500 GB</td>
<td>1.000.000</td>
<td>YES</td>
</tr>
<tr>
<td>$WORK (excl. iris)</td>
<td>3 TB</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>$SCRATCH</td>
<td>per request</td>
<td></td>
<td>NO</td>
</tr>
</tbody>
</table>
Resource and Job Management Systems

- **Resource and Job Management System (RJMS)**
  - *Glue* for a parallel computer to execute parallel jobs
  - **Goal**: satisfy users demands for computation
    - assign resources to user jobs with an efficient manner
Resource and Job Management Systems

- **Resource and Job Management System (RJMS)**
  - *Glue* for a parallel computer to execute parallel jobs
  - *Goal*: satisfy users demands for computation
    - assign resources to user jobs with an efficient manner

- **HPC Resources**:
  - Nodes (typically a unique IP address)
    - Sockets / Cores / Hyperthreads
    - Memory
    - Interconnect/switch resources
  - Generic resources (e.g. GPUs)
  - Licenses

- **Strategic Position**
  - Direct/constant knowledge of resources
  - Launch and otherwise manage jobs
Job Scheduling
Job Scheduling (backfilling)
Job Scheduling (suspension & requeue)
## List of the Main Job Schedulers

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Version*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLURM</td>
<td>SchedMD</td>
<td>17.02.9</td>
</tr>
<tr>
<td>LSF</td>
<td>IBM</td>
<td>10.1</td>
</tr>
<tr>
<td>OpenLava</td>
<td>LSF Fork</td>
<td>2.2</td>
</tr>
<tr>
<td>MOAB/Torque</td>
<td>Adaptative Computing</td>
<td>6.1</td>
</tr>
<tr>
<td>PBS</td>
<td>Altair</td>
<td>13.0</td>
</tr>
<tr>
<td>OAR (PBS Fork)</td>
<td>LIG</td>
<td>2.5.7</td>
</tr>
<tr>
<td>Oracle Grid Engine (formely SGE)</td>
<td>Oracle</td>
<td></td>
</tr>
</tbody>
</table>

* As of Nov. 2017
## List of the Main Job Schedulers

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Version*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLURM</td>
<td>SchedMD</td>
<td>17.02.9</td>
</tr>
<tr>
<td>LSF</td>
<td>IBM</td>
<td>10.1</td>
</tr>
<tr>
<td>OpenLava</td>
<td>LSF Fork</td>
<td>2.2</td>
</tr>
<tr>
<td>MOAB/Torque</td>
<td>Adaptative Computing</td>
<td>6.1</td>
</tr>
<tr>
<td>PBS</td>
<td>Altair</td>
<td>13.0</td>
</tr>
<tr>
<td><strong>OAR (PBS Fork)</strong></td>
<td><strong>LIG</strong></td>
<td><strong>2.5.7</strong></td>
</tr>
<tr>
<td>Oracle Grid Engine (formerly SGE)</td>
<td><strong>Oracle</strong></td>
<td></td>
</tr>
</tbody>
</table>

*: As of Nov. 2017
Typical Workflow on UL HPC resources

- Preliminary setup
  1. Connect to the frontend
  2. Synchronize your code
  3. Reserve a few interactive resources
     - or, (eventually) build your program
     - Test on small size problem
     - Prepare a launcher script

\[\begin{align*}
\text{ssh, screen} & \quad \text{scp/rsync/svn/git} \\
\text{oarsub -I [...]} & \quad \text{on iris: srun -p interactive [...]} \\
\text{gcc/icc/mpicc/nvcc..} & \quad \text{mpirun/srun/python/sh...} \\
& \quad \text{<launcher>.{sh|py}}
\end{align*}\]
Typical Workflow on UL HPC resources

### Preliminary setup
1. Connect to the frontend
   - `ssh`, `screen`
2. Synchronize you code
   - `scp/rsync/svn/git`
3. Reserve a few interactive resources
   - `oarsub -I [...]`
   - on iris: `srun -p interactive [...]`
   - ✓ (eventually) build your program
   - ✓ Test on small size problem
   - ✓ Prepare a launcher script

### Real Experiment
1. Reserve passive resources
   - `oarsub [...] <launcher>`
   - on iris: `sbatch -p {batch|long} [...] <launcher>`
2. Grab the results
   - `scp/rsync/svn/git ...`
Summary

1. Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2. High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3. UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4. HPC Strategy in Europe & Abroad

5. Conclusion & Perspectives
UL HPC in Practice: Toward an [Efficient] Win-Win Usage

Documentation

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

... aka the rtf,ine,m paradigm

Reference Documentation

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

http://hpc.uni.lu
Documentation

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

... aka the rtf_{ine}m 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
Reporting Problems

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> | ssh <node>}; htop on active jobs
   ✓ { oarsub -f -j <jobid> } post-mortem (check the events field)
     iris: scontrol show job <jobid> OR sacct --job <jobid> -l
   ✓ Ganglia on your node(s) https://hpc.uni.lu/status/ganglia.html

ONL Y NOW, consider the following depending on the severity:

→ Open an new issue on (preferred)
→ Mail (only now) us
→ Ask the help of other users

In all cases:
Carefully describe the problem and the context

Guidelines

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

**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> | ssh <node>}; htop **on active jobs**
   - ✓ { oarsub -f -j <jobid> } **post-mortem (check the events field)**
     - iris: scontrol show job <jobid> OR sacct --job <jobid> -l
   - ✓ 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](http://hpc-tracker.uni.lu) (preferred)
→ Mail (only now) us hpc-sysadmins@uni.lu hpc-users@uni.lu
→ **Ask the help of other users**
Reporting Problems

First checks

1. My issue is probably documented
2. An event is on-going
3. check the state of your nodes
   ✓ { oarsub -C <jobid> | ssh <node>}; htop on active jobs
   ✓ { oarsub -f -j <jobid> } post-mortem (check the events field)
   iris: scontrol show job <jobid> OR sacct --job <jobid> -l
   ✓ 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
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

1. Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2. High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3. UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4. HPC Strategy in Europe & Abroad

5. Conclusion & Perspectives
Global race toward Exascale Technology

IDC-Projected Exascale Investment Levels (In Addition to System Purchases)

<table>
<thead>
<tr>
<th>U.S.</th>
<th>EU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 to $2 billion a year in R&amp;D (including NRE)</td>
<td>About 5 billion euros in total</td>
</tr>
<tr>
<td>Investments by both governments &amp; vendors</td>
<td>Investments in multiple exascale and pre-exascale systems</td>
</tr>
<tr>
<td>Plans are to purchase multiple exascale systems</td>
<td>Investments mostly by country governments with a little from the EU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>China</th>
<th>Japan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over $1 billion a year in R&amp;D</td>
<td>Planned investment of just over $1 billion* (over 5 years) for both the R&amp;D and purchase of 1 exascale system</td>
</tr>
<tr>
<td>Investments by both governments &amp; vendors</td>
<td>To be followed by a number of smaller systems ~$100M to $150M each</td>
</tr>
<tr>
<td>Plans are to purchase multiple exascale systems each year</td>
<td>Creating a new processor and a new software environment</td>
</tr>
<tr>
<td>Already investing in 3 pre-exascale systems by 2017/18</td>
<td></td>
</tr>
</tbody>
</table>

*Note that this includes both the system and R&D

©Hyperion Research 2017
Global race toward Exascale Technology

IDC-Projected Exascale Dates and Suppliers

<table>
<thead>
<tr>
<th>Region</th>
<th>Sustained ES:</th>
<th>Peak ES:</th>
<th>Vendors:</th>
<th>Processors:</th>
<th>Initiatives:</th>
<th>Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S.</td>
<td>2023</td>
<td>2021</td>
<td>U.S.</td>
<td>U.S.</td>
<td>NSCI/ECP</td>
<td>$300-500M per system, plus heavy R&amp;D investments</td>
</tr>
<tr>
<td>EU</td>
<td>2023-24</td>
<td>2021</td>
<td>U.S., Europe</td>
<td>U.S., ARM</td>
<td>PRACE, ETP4HPC</td>
<td>$300-$350 per system, plus heavy R&amp;D investments</td>
</tr>
<tr>
<td>China</td>
<td>2023</td>
<td>2020</td>
<td>Chinese</td>
<td>Chinese (plus U.S.?)</td>
<td>13th 5-Year Plan</td>
<td>$350-500M per system, plus heavy R&amp;D</td>
</tr>
<tr>
<td>Japan</td>
<td>2023-24</td>
<td>Not planned</td>
<td>Japanese</td>
<td>Japanese</td>
<td></td>
<td>$600-850M, this includes both 1 system and the R&amp;D costs…will also do many smaller size systems</td>
</tr>
</tbody>
</table>
Exascale Feasibility

We Can Build an Exascale System Today?

Connect together 10 Sunway TaihuLight systems

Require 150 MW of power, programming for 100 M threads and $2.7B price tag.
European HPC strategy

- EU HPC strategy initiated in 2012
  - implementation within H2020 program

- Latest advances:
  - EU Member States sign EuroHPC (Mar. 2017)
    - common effort to create/grow European supercomputing ecosystem
    - Federation of national/regional HPC centers (see also PRACE2)

- EU Objective with EuroHPC:
  - 2-3 Pre-exascale systems by 2019,
  - exascale systems by 2021
HPC Strategy in Europe & Abroad

**European HPC strategy**

- **EU HPC strategy** initiated in 2012
  - implementation within H2020 program

- More recently:
  - IPCEI on HPC and Big Data (BD) Applications (Nov. 2015)
    - Luxembourg (leader), France, Italy & Spain
    - Testbed around Personalized Medicine, Smart Space, Industry 4.0, Smart Manufacturing, New Materials, FinTech, Smart City...

- Latest advances:
  - EU Member States sign EuroHPC (Mar. 2017)
    - Common effort to create/grow European supercomputing ecosystem
    - Federation of national/regional HPC centers (see also PRACE2)

- EU Objective with EuroHPC:
  - 2-3 Pre-exascale systems by 2019, exascale systems by 2021
HPC Strategy in Europe & Abroad

European HPC strategy

EU HPC strategy initiated in 2012

\rightarrow implementation within H2020 program

More recently:

\rightarrow IPCEI on HPC and Big Data (BD) Applications

✓ Luxembourg (leader), France, Italy & Spain
✓ Testbed around Personalized Medicine, Smart Space, Industry 4.0, Smart Manufacturing, New Materials, FinTech, Smart City...

Latest advances:

\rightarrow EU Member States sign EuroHPC (Mar. 2017)
✓ common effort to create/grow European supercomputing ecosystem
✓ Federation of national/regional HPC centers (see also PRACE2)

\rightarrow EU Objective with EuroHPC:
✓ 2-3 Pre-exascale systems by 2019, 2 exascale systems by 2021
European Technology Platform (ETP) for HPC

- Industry-led forum feat. HPC stakeholders
- Providing EU framework to define HPC research priorities/actions
  - UL (P. Bouvry, S. Varrette, V. Plugaru) part of ETP4HPC (2016-)
  - See Strategic Research Agenda, 2017 European HPC Handbook...
EU HPC Strategy Implementation

- **European Technology Platform (ETP) for HPC**
  - Industry-led forum feat. HPC stakeholders
  - Providing EU framework to define HPC research priorities/actions
    - UL (P. Bouvry, S. Varrette, V. Plugaru) part of ETP4HPC (2016-)
    - See Strategic Research Agenda, 2017 European HPC Handbook...

- **EU COST Actions**
  - Ex: NESUS: Network for Sustainable Ultrascale Computing
EU HPC Strategy Implementation

- **European Technology Platform (ETP) for HPC**
  - Industry-led forum feat. HPC stakeholders
  - Providing EU framework to define HPC research priorities/actions
    - UL (P. Bouvry, S. Varrette, V. Plugaru) part of ETP4HPC (2016-)
    - See Strategic Research Agenda, 2017 European HPC Handbook...

- **EU COST Actions**
  - Ex: NESUS: Network for Sustainable Ultrascale Computing

- **PRACE** - Partnership for Advanced Computing in Europe
  - Non-profit association, 25 member countries
  - Providing access to Five EU Tier-0 compute & data resources
EU HPC Strategy Implementation

**European Technology Platform (ETP) for HPC**
- Industry-led forum feat. HPC stakeholders
- Providing EU framework to define HPC research priorities/actions
  - UL (P. Bouvry, S. Varrette, V. Plugaru) part of ETP4HPC (2016-)
  - See Strategic Research Agenda, 2017 European HPC Handbook...

**EU COST Actions**
- Ex: NESUS: Network for Sustainable Ultrascale Computing

**PRACE** - Partnership for Advanced Computing in Europe
- Non-profit association, 25 member countries
- Providing access to **Five EU Tier-0** compute & data resources

- Luxembourg officially entered **PRACE** on **Oct. 17th, 2017**
  - Official Delegate/Advisor (P. Bouvry/S. Varrette) from UL
HPC Strategy in Europe & Abroad

New Trends in HPC

- **Continued scaling** of scientific, industrial & financial applications
  - ... well beyond Exascale
- New trends changing the landscape for HPC
  - Emergence of **Big Data analytics**
  - Emergence of (Hyperscale) **Cloud Computing**
  - Data intensive Internet of Things (IoT) applications
  - Deep learning & cognitive computing paradigms

Special Study

Analysis of the Characteristics and Development Trends of the Next-Generation of Supercomputers in Foreign Countries

Earl C. Joseph, Ph.D. Robert Sorensen
Steve Conway Kevin Monroe

[Source: IDC RIKEN report, 2016]
Aiming at **scalable, flexible HPC infrastructures**

- Primary processing on CPUs and accelerators
  - ✔ HPC & Extreme Scale Booster modules
- Specialized modules for:
  - ✔ HTC & I/O intensive workloads; Data Analytics and AI

[Source: "Towards Modular Supercomputing: The DEEP and DEEP-ER projects", 2016]
Summary

1. Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2. High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3. UL HPC in Practice: Toward an [Efficient] Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4. HPC Strategy in Europe & Abroad

5. Conclusion & Perspectives
Conclusion & Perspectives

Conclusion

Luxembourg government priority on HPC

- sustained by University of Luxembourg HPC developments
  - started in 2007, under resp. of Prof P. Bouvry & Dr. S. Varrette
  - expert UL HPC team (S. Varrette, V. Plugaru, S. Peter, H. Cartiaux, C. Parisot)

- UL HPC (as of 2017): 206.772 TFlops, 7952.4TB (shared)

- consolidate and extend Europe efforts on HPC/Big Data
Conclusion

- Luxembourg government priority on HPC
  - sustained by University of Luxembourg HPC developments
    - started in 2007, under resp. of Prof P. Bouvry & Dr. S. Varrette
    - expert UL HPC team (S. Varrette, V. Plugaru, S. Peter, H. Cartiaux, C. Parisot)
  - UL HPC (as of 2017): **206.772 TFlops, 7952.4TB (shared)**
  - consolidate and extend Europe efforts on HPC/Big Data

---

Several On-going Strategic HPC efforts in Europe...

- ... in which **UL (HPC) is involved** ...
  - ETP4HPC, EU COST Action NESUS etc.
  - PRACE - Official representative for Luxembourg from UL
    - Delegate: Prof. Pascal Bouvry
    - Advisor: Dr. Sebastien Varrette
  - EuroHPC / IPCEI on HPC and Big Data (BD) Applications
Conclusion & Perspectives

Incoming Milestones

- MSA **CDC S-02** as the new UL HPC Data Center (DC)
  - \(\approx 1050\,\text{kW}\) per HPC room
    - Direct Liquid Cooling (DLC)
  - \(\approx 300\,\text{kW}\) per storage room
    - rooms 1, 2 & 5

[Diagram of CDC S-02]

Short term actions

- Official Research computing @ UL & abroad
  - cluster moving to CDC S-02-005
    - pending RFP 170035 (Lustre) + 60 nodes extension
- Luxembourg HPC-BD Competence Center (exp. July 2018)
- National commitment for having next-gen HPC in CDC S-02
- Entering PRACE2

S. Varrette & al. (HPC © University of Luxembourg)
Conclusion & Perspectives

Incoming Milestones

- MSA **CDC S-02** as the **new UL HPC Data Center (DC)**
  - $\approx 1050\text{kW}$ per **HPC** room
  - Direct Liquid Cooling (DLC)
  - $\approx 300\text{kW}$ per **storage** room
    - rooms 1, 2 & 5

**Short term actions**
- Official **Research computing @ UL & abroad** structure
- **iris** cluster moving to CDC S-02-005
  - ✓ pending RFP 170035 (Lustre) + 60 nodes extension
- **Luxembourg HPC-BD Competence Center** (exp. July 2018)
  - ✓ National commitment for having next-gen HPC in CDC S-02
- **Entering PRACE2**
Thank you for your attention...

Questions?

http://hpc.uni.lu

Prof. Pascal Bouvry
Dr. Sebastien Varrette & The UL HPC Team
(V. Plugaru, S. Peter, H. Cartiaux & C. Parisot)
University of Luxembourg, Belval Campus:
Maison du Nombre, 4th floor
2, avenue de l’Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu

1 Introduction
   Preliminaries
   [Parallel] Computing for Enhanced Research
   Overview of the Main HPC Components

2 High Performance Computing (HPC) @ UL
   Overview
   Platform Management
   Back to 2017 Achievements

3 UL HPC in Practice: Toward an [Efficient]
   Win-Win Usage
   General Considerations
   Environment & Typical Workflow Overview
   Documentation & Reporting (problems or results)

4 HPC Strategy in Europe & Abroad

5 Conclusion & Perspectives