



# Introduction to HPC

## 2nd RSG Luxembourg National Congress 2017

---

UL High Performance Computing (HPC) Team

Sarah Peter

University of Luxembourg (UL), Luxembourg

<http://hpc.uni.lu>

---

# Summary

## 1 Introduction

## 2 High Performance Computing (HPC) @ UL

Overview

UL HPC Cluster Organisation

## 3 Hands-on



# Summary

## 1 Introduction

## 2 High Performance Computing (HPC) @ UL

Overview  
UL HPC Cluster Organisation

## 3 Hands-on

# Why High Performance Computing ?

*“The country that out-competes will be the one that out-competes”. Council on Competitiveness*

- Accelerates research by accelerating computation



$\approx 64$  GFlops

(Dual-core i5 2GHz)



198.172 TFlops

(594 computing nodes, 8228 cores)

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



4 TB

(1 disk, 250 MB/s)



6856.4TB

(1558 disks, 10 GB/s)

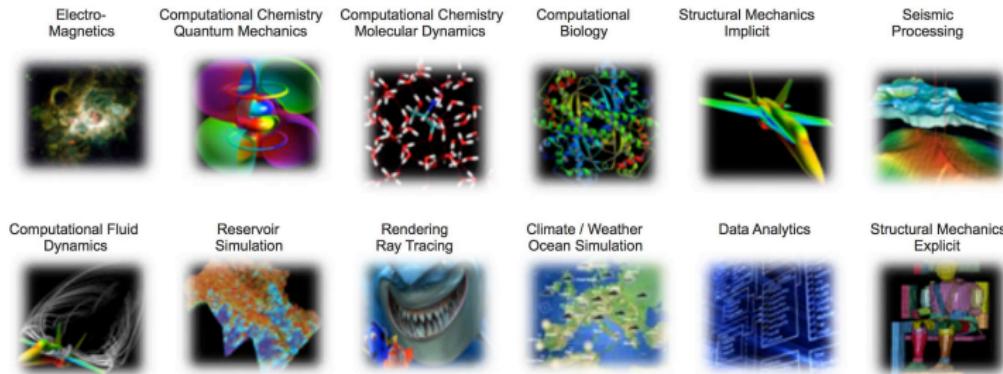
- Communicates faster

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



# HPC at the Heart of our Daily Life

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



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



# Computing for Researchers: Laptop

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



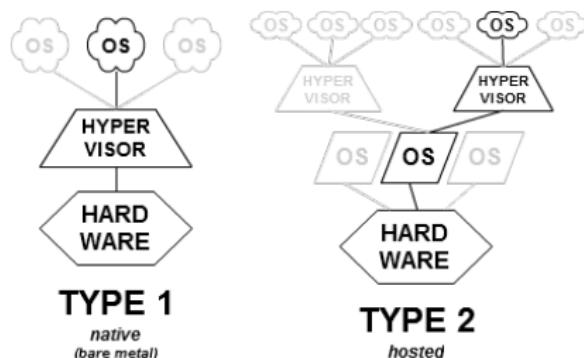
# Computing for Researchers: Laptop

- 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

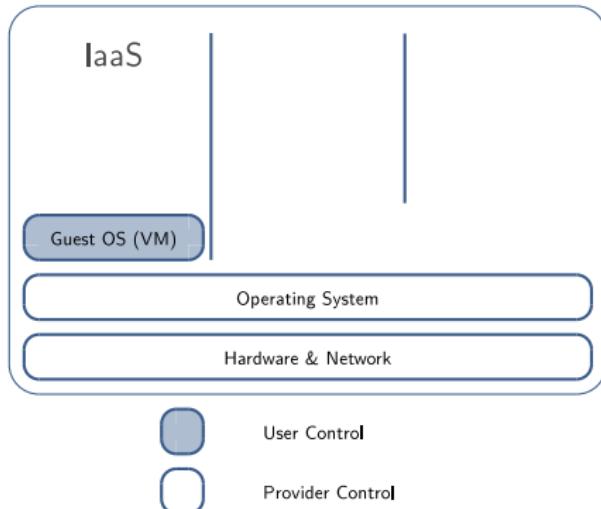
VirtualBox

# Computing for Researchers: Cloud



- Cloud Computing

- access to shared (*generally virtualized*) resources in a pay-per-use manner
- **Infrastructure as a Service (SaaS)**

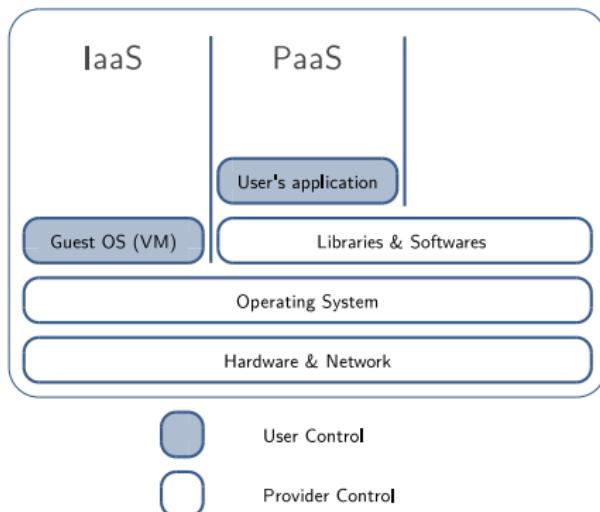


# Computing for Researchers: Cloud



- Cloud Computing

- access to shared (*generally virtualized*) resources in a pay-per-use manner
- **Platform as a Service (PaaS)**

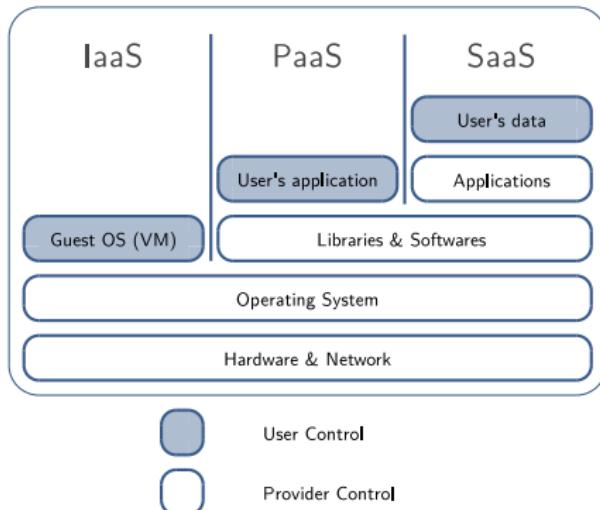


# Computing for Researchers: Cloud



- Cloud Computing

- access to shared (*generally virtualized*) resources in a pay-per-use manner
- **Software as a Service (IaaS)**



# Computing for Researchers: HPC

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



# 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**

- $\text{HPC} \simeq \text{Formula 1}$ 
  - ↪ 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
```



# Jobs, Tasks & Local Execution



```
$> ./myprog
```



# Jobs, Tasks & Local Execution



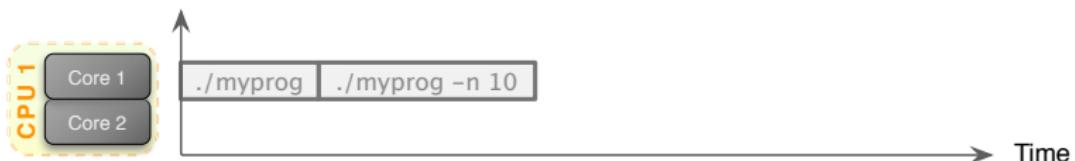
```
$> ./myprog  
$> ./myprog -n 10
```



# Jobs, Tasks & Local Execution



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



# Jobs, Tasks & Local Execution



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



# Jobs, Tasks & Local Execution



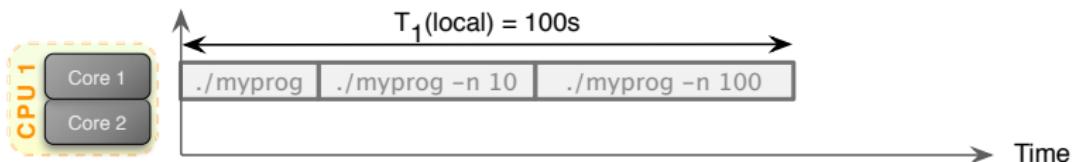
```
$> ./myprog  
$> ./myprog -n 10  
$> ./myprog -n 100
```



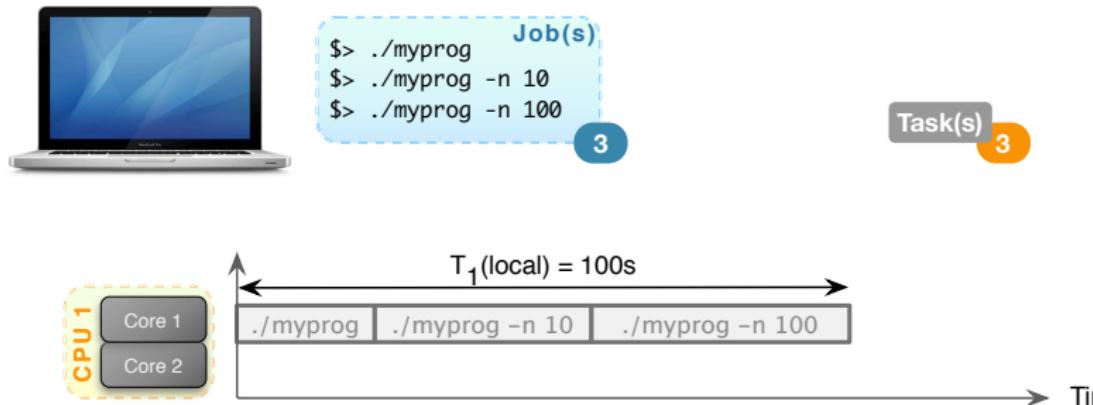
# Jobs, Tasks & Local Execution



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



# Jobs, Tasks & Local Execution



# 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



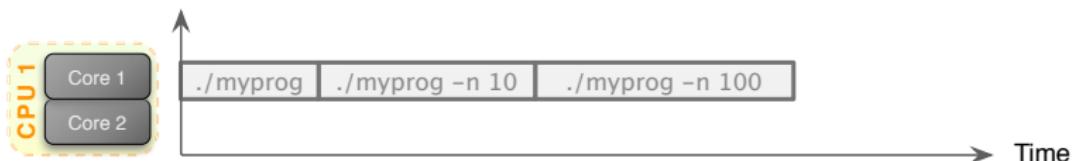
```
# launcher
./myprog
./myprog -n 10
./myprog -n 100
```



# Jobs, Tasks & Local Execution



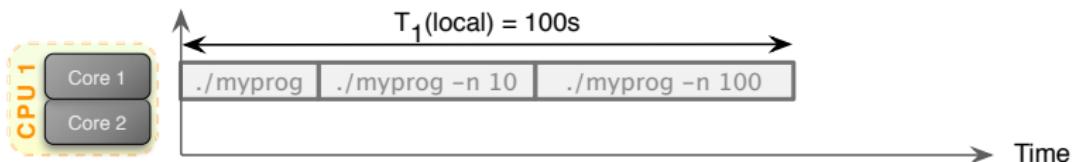
```
# launcher  
./myprog  
./myprog -n 10  
./myprog -n 100
```



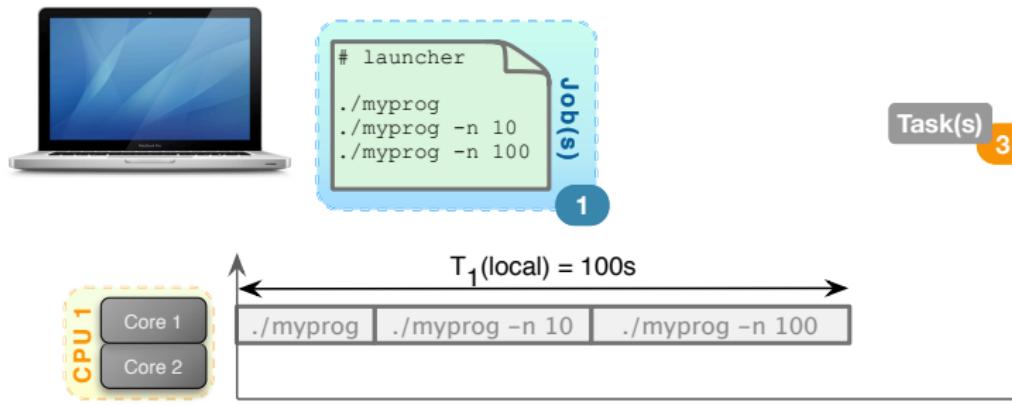
# Jobs, Tasks & Local Execution



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



# Jobs, Tasks & Local Execution



# Jobs, Tasks & Local Execution



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



# Jobs, Tasks & Local Execution



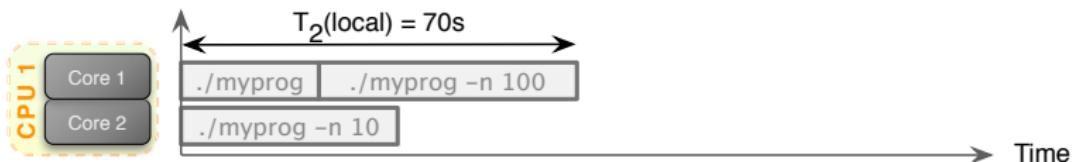
```
# launcher2
"Run in //:""
./myprog
./myprog -n 10
./myprog -n 100
```



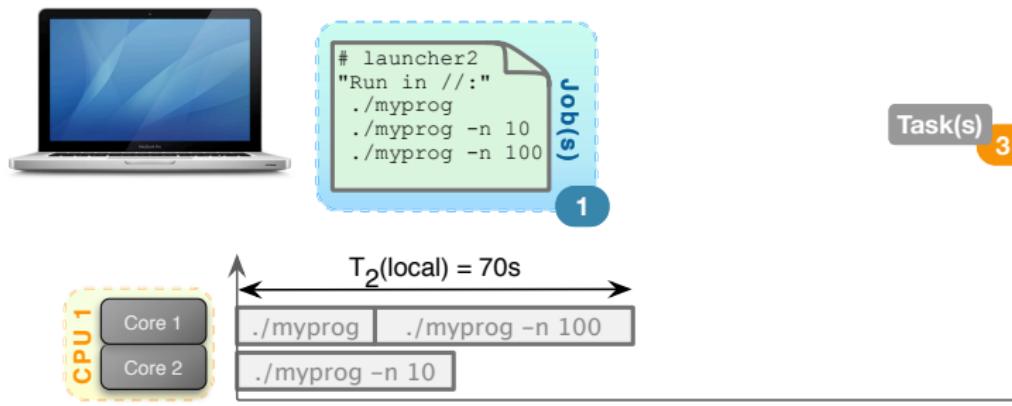
# Jobs, Tasks & Local Execution



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



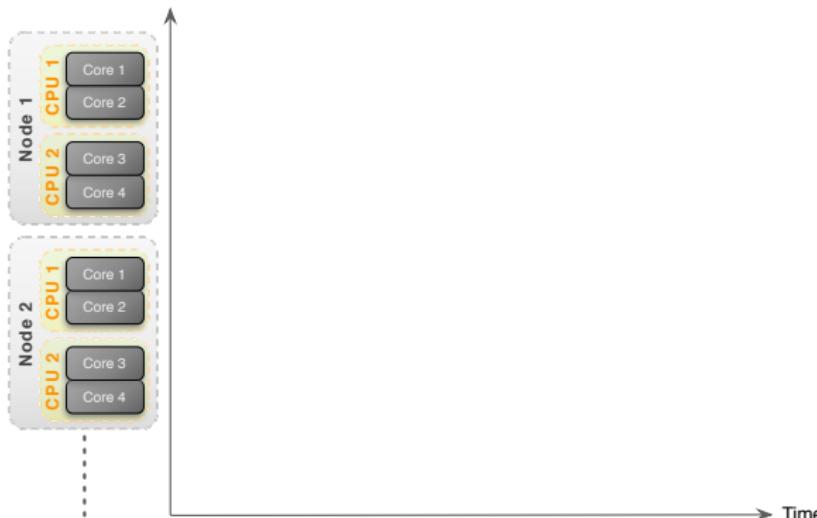
# Jobs, Tasks & Local Execution



# Jobs, Tasks & HPC Execution



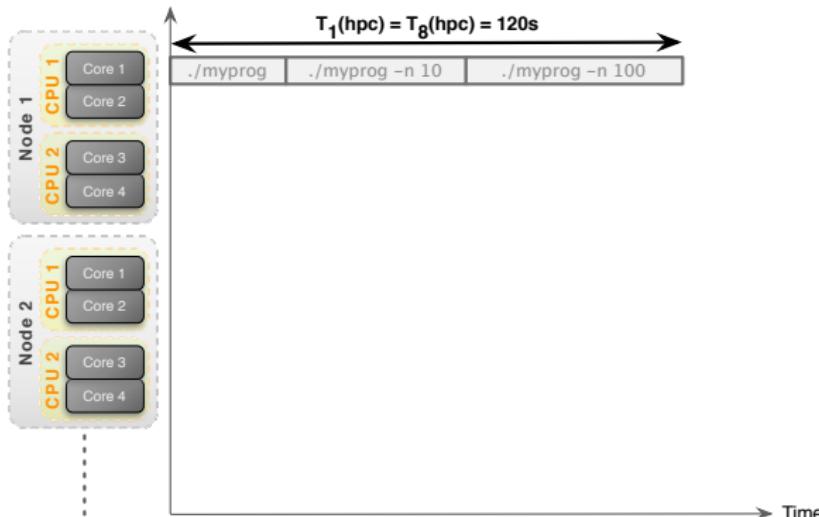
```
# launcher
./myprog
./myprog -n 10
./myprog -n 100
```



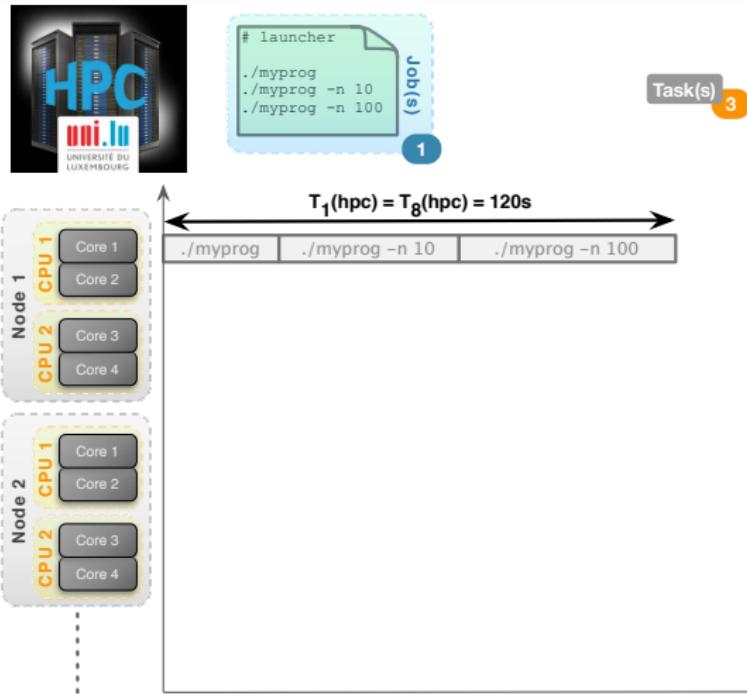
# Jobs, Tasks & HPC Execution



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



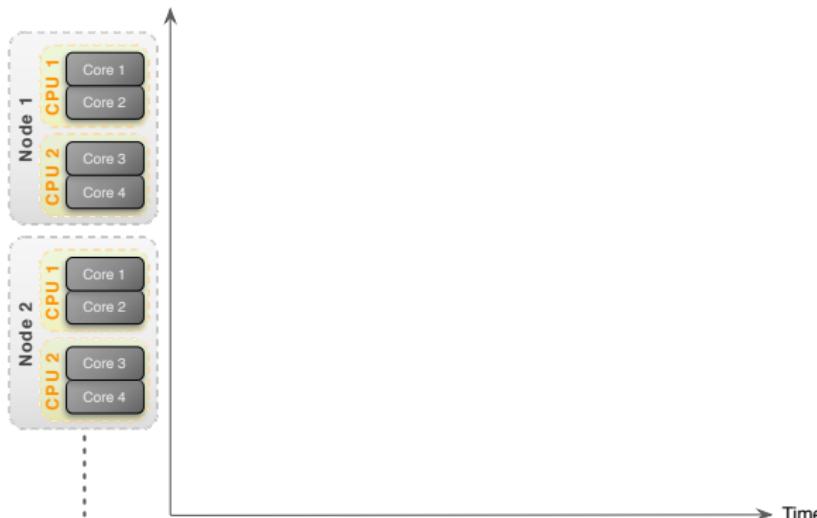
# Jobs, Tasks & HPC Execution



# Jobs, Tasks & HPC Execution



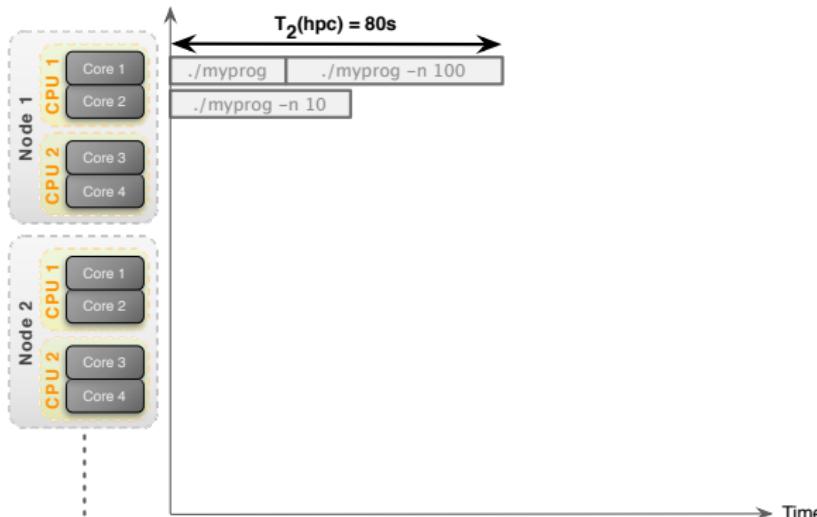
```
# launcher2
"Run in //:""
./myprog
./myprog -n 10
./myprog -n 100
```



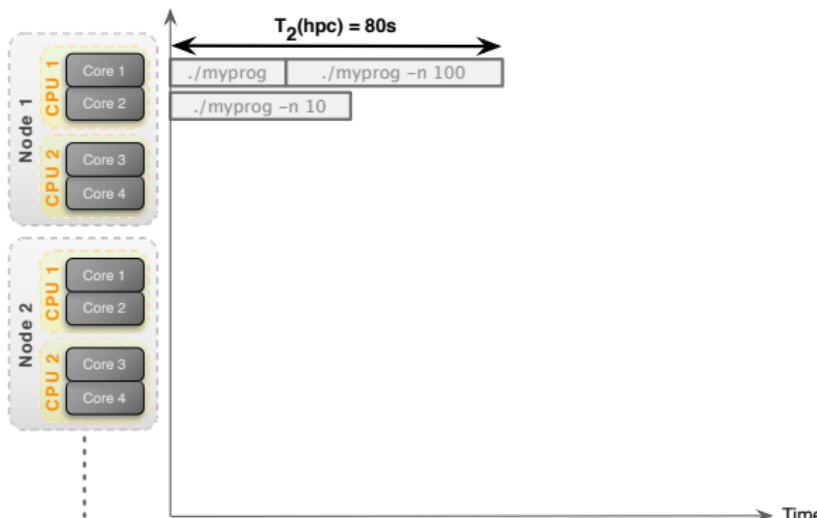
# Jobs, Tasks & HPC Execution



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



# Jobs, Tasks & HPC Execution



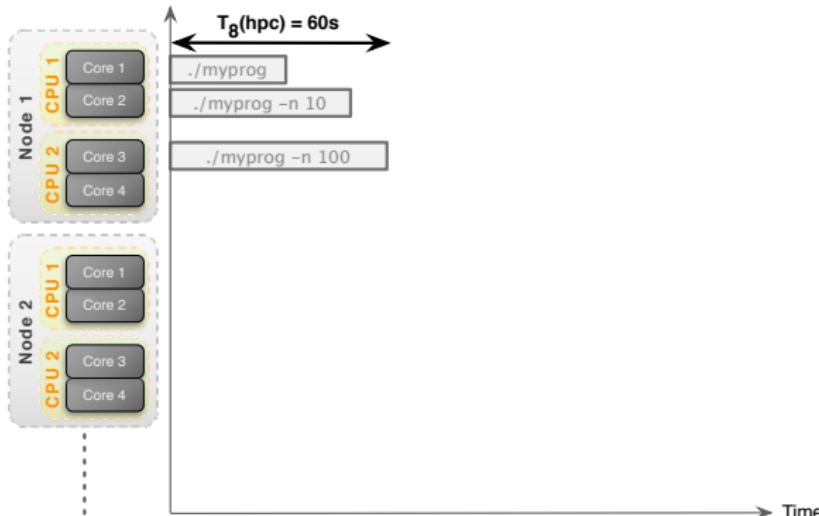
# Jobs, Tasks & HPC Execution



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

(s) 1

Task(s) 3



## Local vs. HPC Executions

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

## Local vs. HPC Executions

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

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

## Local vs. HPC Executions

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

- **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

## Summary

### 1 Introduction

### 2 High Performance Computing (HPC) @ UL

Overview

UL HPC Cluster Organisation

### 3 Hands-on



# High Performance Computing @ UL

**HPC @ Uni.lu**

Chaos, Gaia, Nyx and Granduc clusters

Get Updates:

Search...

Systems • For Users • Live Status • HPC School • Blog/News • About •

Welcome to the HPC @ Uni.lu platform !

This is the official website of HPC @ Uni.lu platform, which assemble information about the computing clusters operated by the University of Luxembourg and the organization running them.

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

Server room @ Belval  
This picture corresponds to the server room in the LCSS building @ Belval, hosting the **Gata** cluster. The violet lights come from the Nascent disk enclosures.

**Featured Systems**  
We currently operate a total of 494 computing nodes (540 cores, 90,199 CPU [T]cores) and a shared storage capacity of 4919.4 TB (+ 1916 TB for backup).

**Platform Status**  
Several tools report in live the current status of our systems. Check them out!

**Latest News**  
Get the latest news & advertisements linked to the UL HPC platform in this page.

**Recent Posts**

- HPC User Tools (DevOps) Army Kravas Tools for the researcher
- Optimizing performance on the Lustre Filesystem
- UL HPC Storage System
- IP2S-HPC R&D Project Released
- UL HPC storage infrastructure upgrade
- HPC as part of the UL Digital Strategy

**Github Repos**

doftiles qualif tutorials ...

Tweets by [@ulhpc](#)

**ulHPC** Retweeted [@Sebastien\\_Varette](#) (swarmrds)  
Remember to register now for IEEE #CloudCom2016! [https://cloudcom2016.cloudcom.org/CloudCom\\_2016/program/submitting\\_papers...@CloudCom\\_Org @hpc\\_lu](#) ... 31 Oct

**ulHPC** Retweeted [@Sebastien\\_Varette](#) (swarmrds)  
Help us to get your requirements for the next generation UL HPC platform! Contact us to access the UL HPC User Survey 31 Oct

**ulHPC** Retweeted [@Sebastien\\_Varette](#) (swarmrds)  
1 week to go until the submission deadline of IEEE #CloudCom2016! [2016.cloudcom.org](#) 31 Oct

**ulHPC** Retweeted [@Sebastien\\_Varette](#) (swarmrds)  
Today I gave a seminar IT/DevOps, Army Kravas 31 Oct

<http://hpc.uni.lu>

## Key numbers

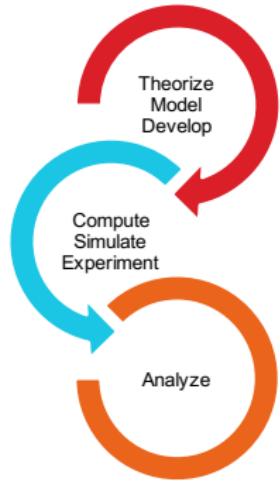
- 416 users
- 110 servers
- 594 nodes
  - 8228 cores
  - **198.172 TFlops**
  - 50 accelerators
  - (+ **76.22 TFlops**)
- **6856.4 TB**
- 5 sysadmins
- 2 sites
  - Kirchberg
  - Belval

## UL HPC Beneficiaries

### 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

# Accelerating UL Research



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

- >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

# 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)

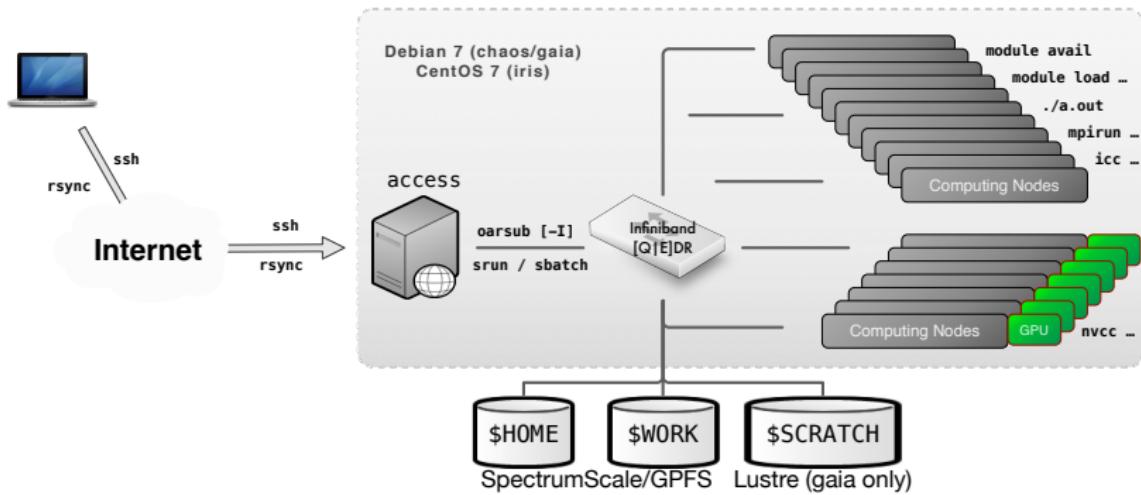
**Hyacinthe Cartiaux**

CDI, Support (SIU)

**Clément Parisot**

CDI, Support (FSTC)

# UL HPC: General cluster organization



# 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:
    - ↪ > **140 software packages**, in *multiple* versions, within **18 categ.**
    - ↪ reworked software set for iris cluster and soon deployed everywhere
      - ✓ RESIF v2.0, allowing [real] semantic versionning of released builds
    - ↪ hierarchical organization

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

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

# Summary

## 1 Introduction

## 2 High Performance Computing (HPC) @ UL

Overview

UL HPC Cluster Organisation

## 3 Hands-on



Hands-on

---

# Tutorial

<https://github.com/sarah-peter/hpc-tutorial>

# Questions?

<http://hpc.uni.lu>



## Sarah Peter & The UL HPC Team

University of Luxembourg, Belval Campus:  
Maison du Nombre, 4th floor  
2, avenue de l'Université  
L-4365 Esch-sur-Alzette  
mail: [hpc-sysadmins@uni.lu](mailto:hpc-sysadmins@uni.lu)

### 1 Introduction

### 2 High Performance Computing (HPC) @ UL

Overview  
UL HPC Cluster Organisation

### 3 Hands-on