# Présentation des nouvelles machines Gemini

Simon Delamare

LIP / CNRS

14 novembre 2019

# Outline

- Grid'5000 summary
  - Platform goal & design
  - How to use it
- Presentation of the new Gemini cluster
  - HW spec
  - Integration in Grid'5000 and specific features

# Grid'5000

- A large scale testbed for distributed computing
  - HPC, Clouds, Big Data, Networking, AI
  - To experiment in a fully controllable and observable environment
  - More than 500 active users per year, 100-150 scientific publications
- How ?
  - bare-metal node reservation w/ reconfiguration capabilities
  - tools for experimenting: monitoring, scripting, etc.

$\Longrightarrow$ *Not a Cloud*: full control on all layers

$\Longrightarrow$ *Not a Computing Center*: $\approx$ same HW, but different capabilities and usage

$\Longrightarrow$ *Not a Grid...*

# How does it work ? (1)

- Infrastructure:
  - 8 sites, 35 clusters, 800 nodes, 15k cores
  - various HW available: Many different CPUs, network, GPU, disks...

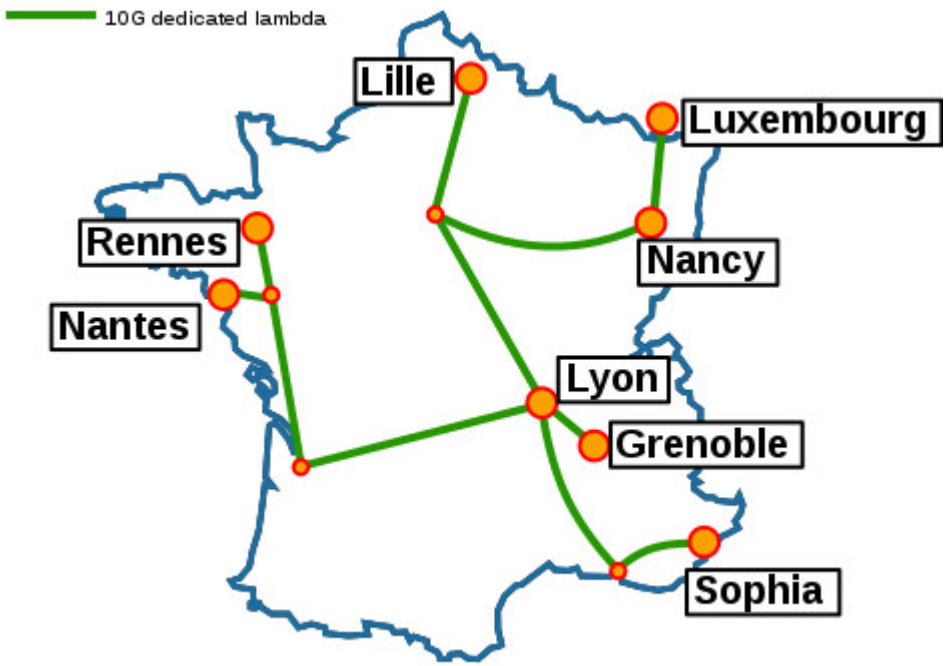


Figure 1: Grid'5000 Infrastructure

# How does it work ? (2)

Main steps for an experiment:

1. Select some nodes available in Grid'5000 according the spec. you need
2. Reserve nodes with *oar*
3. Optionally deploy the OS of your choice using *kadeploy*
4. Connect to nodes using SSH (even as *root*)
5. Do your experiment
6. Monitor performance or energy consumption

*All these steps can be automated using Grid'5000 API and high level programming language*

# How does it work ? (3)

Usage Policy for reservation:

- During day time: 2 hours.cluster allowed
- No limitation during night and WE

$\Longrightarrow$ Design experiments during daytime, schedule them to run during the night

*Also available: "best-effort" (preemptible) mode*

# Gemini hardware spec.

Gemini "cluster": gemini-1 & gemini-2, each node:

- *Model*: Nvidia DGX-1
- *CPU*: Intel Xeon E5-2698 v4 (Broadwell, 2.20GHz, 2 CPUs/node, 20 cores/CPU)
- *Memory*: 512 GiB
- *Storage*:
  - 480 GB SSD (used for system, remaining space available in /tmp)
  - 4 x 1.92 TB SSD (reservable)
- *Network*:
  - Ethernet 10Gbps
  - 4 x InfiniBand EDR 100Gbps interfaces (Mellanox ConnectX-5).
    Fully connected to a EDR IB switch, IPoIB configured on 1 interface
- *GPU*: 8 x Nvidia Tesla V100, w/ NVLink bus

# System consideration

- Default Grid'5000 OS: Debian 10 "buster" (*debian10-x64-std*)
- Once reserved, user can become *root* on the node by using *sudo-g5k* command
- *nvidia-docker* can be installed with the *g5k-setup-nvidia-docker* script
  - For nvidia's container: https://www.nvidia.com/en-us/gpu-cloud/containers/
  - Pre-built and optimized HPC & IA software for Nvidia's hardware
- Other systems are available using kadeploy:
  - Other flavors of Debian
  - Centos 7 & 8
  - Ubuntu 18.04
  - and an "unofficial" DGX-OS image is available

```
kadeploy -a http://public.lyon.grid5000.fr/~sdelamare/dgxos-4.1.0.dsc ...
```

# Specific reservations (1)

Reserve a subset of GPUs:

- Advised when using the 8 GPUs are not strictly needed (development/debugging, etc.)
- You get one (or more) GPU, and a subset of associated CPU cores (5 cores/GPU on gemini)
- Caveat: no sudo-g5k, no nvidia-docker

```
oarsub -I -p "cluster='gemini'" -l "gpu=1"
```

More info: https://www.grid5000.fr/w/Accelerators_on_Grid5000#Reserving_GPU_units_on_nodes_with_many_GPUs

# Specific reservations (2)

Reserve additional disks:

- Reserve extra SSD disks available on gemini during for a long period
- To avoid data transfer at every job start/end

```
oarsub -r "2019-11-14 09:00:00" -t noop -l \
 {"type='disk' and host='gemini-1.lyon.grid5000.fr'"}/host=1/disk=1,walltime=168
```

Then reserve gemini-1, the disk will be available under /dev/sdX (you currently need to format and mount it yourself as root/sudo-g5k)

More info: https://www.grid5000.fr/w/Disk_reservation

# HPC & IA software

- Mostly available using the “module” command
  - latest versions of cuda, cudnn, gcc, intel compilers, llvm, openmpi, etc.

```
$ module av
autoconf/2.69_gcc-6.4.0     cuda/9.1.85_gcc-6.4.0     gcc/6.4.0_gcc-6.4.0
automake/1.16.1_gcc-6.4.0   cuda/9.2.88_gcc-6.4.0     gcc/6.5.0_gcc-6.4.0
boost/1.69.0_gcc-6.4.0      cuda/10.0.130_gcc-6.4.0   gcc/7.4.0_gcc-6.4.0
cmake/3.13.4_gcc-6.4.0      cuda/10.1.243_gcc-6.4.0   gcc/8.3.0_gcc-6.4.0
cuda/7.5.18_gcc-6.4.0       cudnn/5.1_gcc-6.4.0       gmp/6.1.2_gcc-6.4.0
cuda/8.0.61_gcc-6.4.0       cudnn/6.0_gcc-6.4.0       hwloc/1.11.11_gcc-6.4.0
cuda/9.0.176_gcc-6.4.0      cudnn/7.3_gcc-6.4.0       hwloc/2.0.2_gcc-6.4.0

$ module load cuda
$ module load gcc/6.4.0

$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
```

# Energy monitoring

Using dedicated “Wattmetre” devices:

- 50 measurements per sec., behind PSU
- Raw measurements available on http://wattmetre.lyon.grid5000.fr/data

  (script to fetch data available at: https://gitlab.inria.fr/delamare/wattmetre-read/blob/master/tools/getwatt.py)
- Also available from Kwapi (provides and UI and API)


More info: https://www.grid5000.fr/w/Energy_consumption_monitoring_tutorial

# Concluding remarks

Getting an access:

- Ask for an account on www.grid5000.fr
- Available to French academics (Open Access program otherwise)

- Feedback very welcomed!
  - Experiment/benchmark/studies performed on *gemini*
  - Missing features for a better experimentation environment?
  - Any needs for training related to *gemini*? (architecture, usage, . . . )