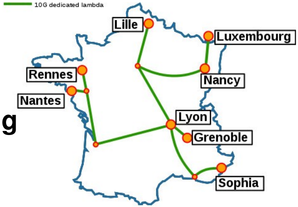
Grid'5000

Lucas Nussbaum

Grid'5000 Technical Director

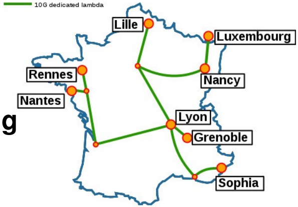
2018-04-03

The Grid'5000 testbed



- **A large-scale testbed for distributed computing**
  - 8 sites, 30 clusters, 840 nodes, 8490 cores
  - Dedicated 10-Gbps backbone network
  - 600 users and 100 publications per year

The Grid’5000 testbed



- **A large-scale testbed for distributed computing**
  - 8 sites, 30 clusters, 840 nodes, 8490 cores
  - Dedicated 10-Gbps backbone network
  - 600 users and 100 publications per year
- A meta-grid, meta-cloud, meta-cluster, meta-data-center:
  - Used by CS researchers in HPC / Clouds / Big Data / Networking
  - To experiment in a fully controllable and observable environment
  - Similar problem space as Chameleon and Cloudlab (US)
  - Design goals:
    - Support high-quality, reproducible experiments
    - On a large-scale, shared infrastructure

Organization and governance

- **Director** – Frédéric Desprez
- ***Bureau*** (6 members: FD, LN, Christian Perez, Adrien Lebre, Laurent Lefevre, David Margery)
- ***Comité des responsables de sites***
- **Technical Director** – Lucas Nussbaum
  - Technical team: 9 full-time engineers
- **Architects committee** (6 membres)
- ***Conseil de groupement***
  - Inria, CNRS, RENATER, CEA, CPU, CDEFI, Mines-Telecom
- ***Conseil scientifique***
  - 10 members

institutional and scientific steering

technical steering

advisory and evaluation bodies

Landscape – cloud & experimentation[1]

- Public cloud infrastructures (AWS, Azure, Google Cloud Platform, etc.)
  - ☹ No information/guarantees on placement, multi-tenancy, real performance
- Private clouds: Shared observable infrastructures
  - ☺ Monitoring & measurement
  - ☹ No control over infrastructure settings
  - ⇝ Ability to understand experiment results
- Bare-metal as a service, fully reconfigurable infrastructure (Grid'5000)
  - ☺ Control/alter all layers (virtualization technology, OS, networking)
  - ⇝ *In vitro* Cloud

[1] Inspired from a slide by Kate Keahey

Landscape – cloud & experimentation[1]

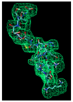
- Public cloud infrastructures (AWS, Azure, Google Cloud Platform, etc.)
  - ☹ No information/guarantees on placement, multi-tenancy, real performance
- Private clouds: Shared observable infrastructures
  - ☺ Monitoring & measurement
  - ☹ No control over infrastructure settings
  - ⇝ Ability to understand experiment results
- Bare-metal as a service, fully reconfigurable infrastructure (Grid'5000)
  - ☺ Control/alter all layers (virtualization technology, OS, networking)
  - ⇝ *In vitro* Cloud

And the same applies to all other environments (e.g. HPC)

[1] Inspired from a slide by Kate Keahey

Some results from Grid'5000 users

HPC: In Situ Analytics[2]

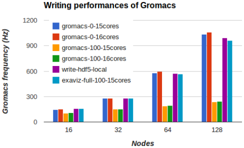


Goal: improve organization of simulation and data analysis phases

- Simulate on a cluster; move data; post-mortem analysis
  - Unsuitable for Exascale (data volume, time)
- Solution: analyze on nodes, during simulation
  - Between or during simulation phases? dedicated core? node?

Grid'5000 used for development and test, because control:

- Of the software environment (MPI stacks)
- Of CPU performance settings (Hyperthreading)
- Of networking settings (Infiniband QoS)



Then evaluation at a larger scale on the Froggy supercomputer (CIMENT center, Grenoble)

[2] Matthieu Dreher and Bruno Raffin. "A Flexible Framework for Asynchronous in Situ and in Transit Analytics for Scientific Simulations". In: *CCGrid*. 2014.

Cloud: DISCOVERY project

Goal: design a distributed IaaS cloud, based on OpenStack

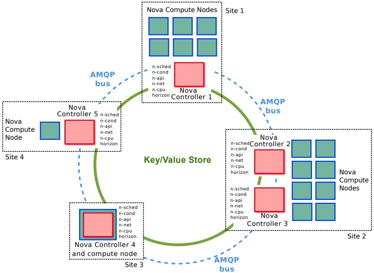
- Move services as close as possible to users
  - Legal reasons, network latency
- Leverage regional data centers
- Increase resillience (no SPOF)
- P2P and self-* approaches

Grid'5000 as a testbed already provides:

- Start and control your own OpenStack
- Possibly modified
- Running at large scale

Collaborations:

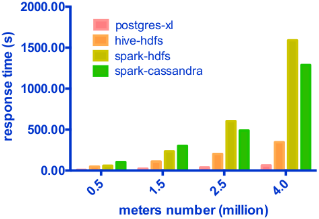
- Inria, RENATER, Orange, Mines Nantes



`http://beyondtheclouds.github.io/`

Big Data: smart power meters[3]

- Goal: which big data solution for Linky smart meters data?
- Collaboration with ERDF
- 4 Big Data solutions installed and compared on Grid'5000
  - Postgres-XL, Hadoop, Spark, Cassandra
  - Up to 140 nodes
  - 1.7 TB of data (≈ 5 million meters, 1 mes/h, 1 year)



[3] Houssem Chihoub and Christine Collet. "A scalability comparison study of smart meter data management approaches". In: *Grid'5000 Winter School*. 2016.

Other usage areas

- Deep learning
- OS research
- Networking
- Reproducible research

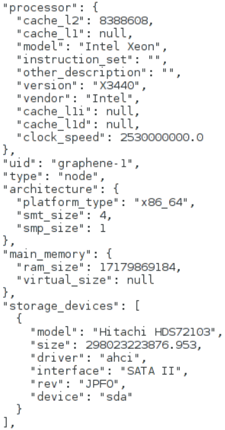
An experiment’s outline

1. Discovering resources from their description
2. Reconfiguring the testbed to meet experimental needs
3. Monitoring experiments, extracting and analyzing data
4. Controlling experiments: API

Discovering resources from their description

- Describing resources ⇝ understand results
  - Covering nodes, network equipment, topology
  - Machine-parsable format (JSON) ⇝ scripts
  - Archived (*State of testbed 6 months ago?*)
  - Soon: better hardware description on the wiki
- Verifying the description
  - Avoid inaccuracies/errors ⇝ wrong results
  - Could happen frequently: maintenance, broken hardware (e.g. RAM)
  - Our solution: g5k-checks
    - Runs at node boot (or manually by users)
    - Acquires info using OHAI, ethtool, etc.
    - Compares with Reference API
- Selecting resources
  - OAR database filled from Reference API

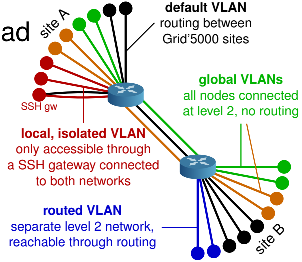
```
oarsub -p "wattmeter='YES' and gpu='YES'"
oarsub -l "cluster='a'/nodes=1+cluster='b' and
      eth10g='Y'/nodes=2,walltime=2"
```



```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "platform_type": "x86_64",
  "smt_size": 4,
  "smp_size": 1
},
"main_memory": {
  "ram_size": 17179869184,
  "virtual_size": null
},
"storage_devices": [
  {
    "model": "Hitachi HDS72103",
    "size": 298023223876.953,
    "driver": "ahci",
    "interface": "SATA II",
    "rev": "JPFO",
    "device": "sda"
  }
],
```

Reconfiguring the testbed

- Operating System reconfiguration with Kadeploy:
  - Provides a *Hardware-as-a-Service* cloud infrastructure
  - Enable users to deploy their own software stack & get *root* access
  - **Scalable, efficient, reliable and flexible**:
    200 nodes deployed in ~5 minutes
- Customize networking environment with KaVLAN
  - Protect the testbed from experiments (Grid/Cloud middlewares)
  - Avoid network pollution
  - Create custom topologies
  - By reconfiguring VLANS ⇝ almost no overhead



default VLAN
routing between
Grid'5000 sites

global VLANs
all nodes connected
at level 2, no routing

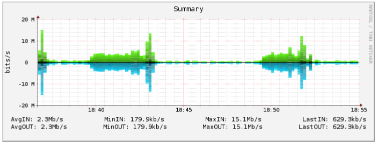
local, isolated VLAN
only accessible through
a SSH gateway connected
to both networks

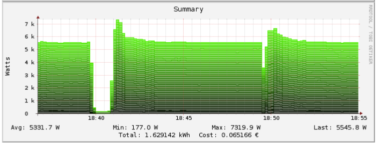
routed VLAN
separate level 2 network,
reachable through routing

Monitoring experiments

Goal: enable users to understand what happens during their experiment

- System-level probes (usage of CPU, memory, disk, with Ganglia)
- Infrastructure-level probes: Kwapi
  - Network, power consumption
  - Captured at high frequency (≈1 Hz)
  - Live visualization
  - REST API
  - Long-term storage
  - Should be fully operational again soon!





Controlling experiments: API

- Legacy way of performing experiments: shell commands
  - ☹ time-consuming
  - ☹ error-prone
  - ☹ details tend to be forgotten over time

- Promising solution: automation of experiments
  ⇝ Executable description of experiments

- Support from the testbed: Grid'5000 RESTful API
  (Resource selection, reservation, deployment, monitoring)

- Several higher-level tools to help automate experiments
  Execo (Python), Ruby-cute (Ruby)
  `https://www.grid5000.fr/w/Grid5000:Software`

Recent and little-known features

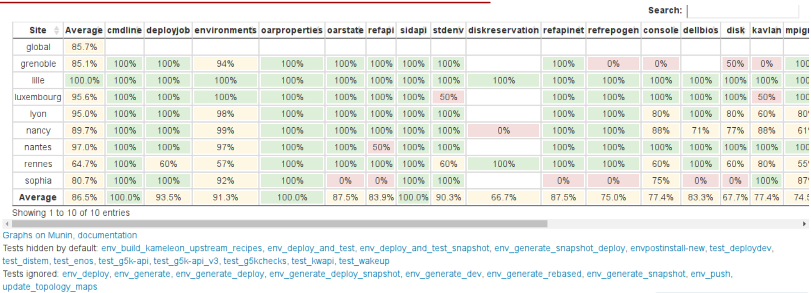
Data management portfolio

- Storage5k: reservation of storage space on an NFS server
- Managed Ceph clusters in Rennes and Nantes
- OSIRIM: large storage space made available in Toulouse
- Reservation of disks on nodes
  - To store large datasets between nodes reservations

Missing: long-term archival of experiment data

- Probably not a good idea to solve this on our own
- Feedback about CKAN-based servers (OpenAIRE / Zenodo), anyone?

Automated testing framework



Search:

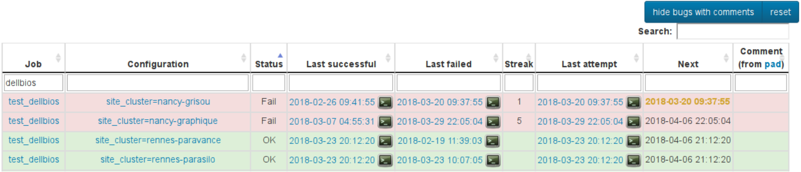
Site	Average	cmdline	deployjob	environments	oarproperties	oarstate	refapi	sidapi	stdenv	diskreservation	refapinet	refrepogen	console	dellbios	disk	kavlan	mpigi
global	85.7%																
grenoble	85.1%	100%	100%	94%	100%	100%	100%	100%	100%		100%	0%	0%		50%	0%	100
lille	100.0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100
luxembourg	95.6%	100%	100%	100%	100%	100%	100%	100%	50%		100%	100%	100%	100%	100%	50%	100
lyon	95.0%	100%	100%	98%	100%	100%	100%	100%	100%		100%	100%	80%	100%	80%	60%	80
nancy	89.7%	100%	100%	99%	100%	100%	100%	100%	100%	0%	100%	100%	88%	71%	77%	88%	61
nantes	97.0%	100%	100%	97%	100%	100%	50%	100%	100%		100%	100%	100%	100%	100%	100%	100
rennes	64.7%	100%	60%	57%	100%	100%	100%	100%	60%	100%	100%	100%	60%	100%	60%	80%	55
sophia	80.7%	100%	100%	92%	100%	0%	0%	100%	100%		0%	0%	75%	0%	0%	100%	87
Average	86.5%	100.0%	93.5%	91.3%	100.0%	87.5%	83.9%	100.0%	90.3%	66.7%	87.5%	75.0%	77.4%	83.3%	67.7%	77.4%	74.5

Showing 1 to 10 of 10 entries

Graphs on Munin, documentation

Tests hidden by default: env_build_kameleon_upstream_recipes, env_deploy_and_test, env_deploy_and_test_snapshot, env_generate_snapshot_deploy, envpostinstall-new, test_deploydev, test_distem, test_enos, test_g5k-api, test_g5k-api_v3, test_g5kchecks, test_kwapi, test_wakeup

Tests ignored: env_deploy, env_generate, env_generate_deploy, env_generate_deploy_snapshot, env_generate_dev, env_generate_rebased, env_generate_snapshot, env_push, update_topology_maps



Search:

Job	Configuration	Status	Last successful	Last failed	Streak	Last attempt	Next	Comment (from pad)
dellbios								
test_dellbios	site_cluster=nancy-grisou	Fail	2018-02-26 09:41:55	2018-03-20 09:37:55	1	2018-03-20 09:37:55	2018-03-20 09:37:55	
test_dellbios	site_cluster=nancy-graphique	Fail	2018-03-07 04:55:31	2018-03-29 22:05:04	5	2018-03-29 22:05:04	2018-04-06 22:05:04	
test_dellbios	site_cluster=rennes-paravance	OK	2018-03-23 20:12:20	2018-02-19 11:39:03		2018-03-23 20:12:20	2018-04-06 21:12:20	
test_dellbios	site_cluster=rennes-parasilo	OK	2018-03-23 20:12:20	2018-03-23 10:07:05		2018-03-23 20:12:20	2018-04-06 21:12:20	

- Detect regressions before experimenters
- 23 tests, 1055 configurations
- Still, does not cover everything ⇝ please report problems!

Other random stuff

- Additional switch on the *grisou* cluster
  - 48 nodes with 4x 10G interfaces and 1x 1G interface
  - 1G interfaces are connected to an ONIE-supported switch
- VPN to connect to Grid'5000
- `sudo-g5k`
- OAR job extensions (`oarwalltime` command)
- Persistent virtual machines
- News on the website (also: Twitter @grid5000, RSS)

What to expect by mid 2018 (hopefully)

- New clusters in:
  - Nantes: 48 nodes (CPER SEDUCE)
  - Grenoble: 72 nodes + 4 nodes (HPCDA project, HPC/BigData convergence with NVMe)
  - Nancy: 64 nodes (+ 48 older nodes) (*production* queue)
  - Lille: 8 nodes with P100 GPUs (CPER DATA)
- Kwapi fully operational again
- Full rework of hardware & network pages
- Wrap-up of the work around Debian 9 images (inc. move to predictable network interface names, and new postinstalls)
- Many other behind-the-scenes changes

Grid'5000-related tutorials during the school

- Tue 4:30pm, Wed 2pm – Getting Started with Grid'5000
- Wed 2pm, 4:30pm – Benchmarking OpenStack with EnOS
- Thu 2pm – Monitoring energy consumption in Grid'5000 experiments
- Thu 2pm – BigData Experiments with Grid5000
- Fri 9am – Distributed systems and networking emulation with Distem
- Fri 9am – Open session – any Grid'5000 tutorial

Come talk to the Grid'5000 team about features you need!