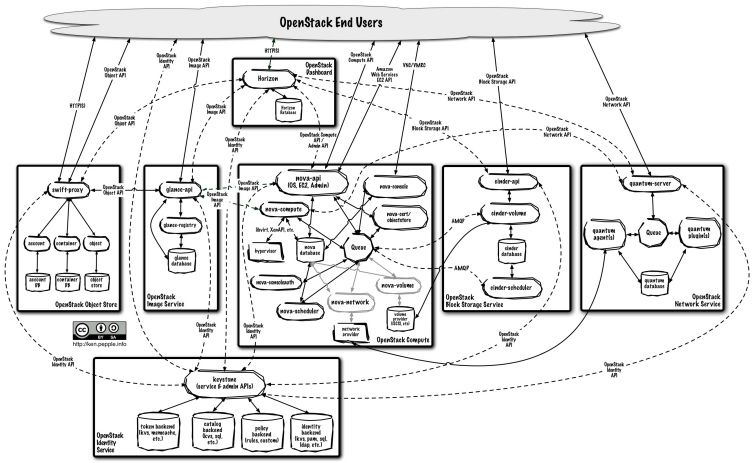
# Enabling Large-Scale Testing of IaaS Cloud Platforms on the Grid'5000 Testbed

Sébastien Badia, Alexandra Carpen-Amarie,
Adrien Lèbre, Lucas Nussbaum




# Testing IaaS clouds stacks




- IaaS Cloud stacks: complex software
- Needs to be tested in realistic setups
- But testing often limited to:
  - Single-machine installations
  - Static deployments

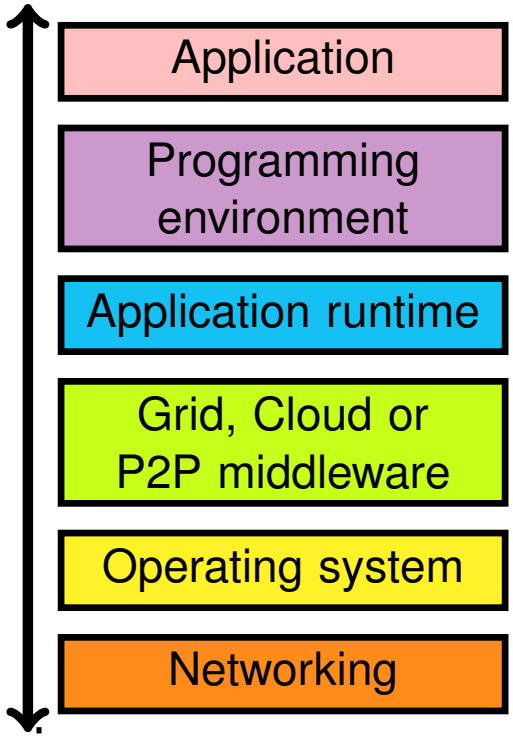
**This talk:**
**enabling large-scale testing of IaaS Cloud stacks**
**on a shared, reconfigurable testbed**

# Outline

1. Quick overview of the Grid'5000 testbed
2. Support for Virtualization and Cloud on Grid'5000
3. Deploying IaaS Clouds on Grid'5000

# Grid'5000

- **Testbed for research on distributed systems**
  - High Performance Computing
  - Grids
  - Peer-to-peer systems
  - Cloud computing
- History:
  - 2003: Project started (ACI GRID)
  - 2005: Opened to users

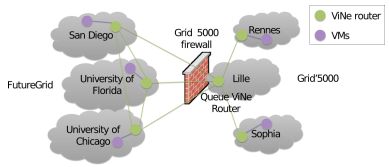



- Funding: Inria, CNRS and many local entities (regions, universities)
- Only for research on distributed systems → no production usage
  Litmus test: *are you interested in the result of the computation?*
  - Free nodes during daytime to prepare experiments
  - Large-scale experiments during nights and week-ends
- Also a scientific object: how does one design such a testbed?

# Leading to results in several fields

## Cloud: Sky computing on FutureGrid and Grid'5000

- Nimbus cloud deployed on 450+ nodes
- Grid'5000 and FutureGrid connected using ViNe




## HPC: factorization of RSA-768

- Feasibility study: prove that it can be done
- Different hardware $\rightsquigarrow$ understand the performance characteristics of the algorithms

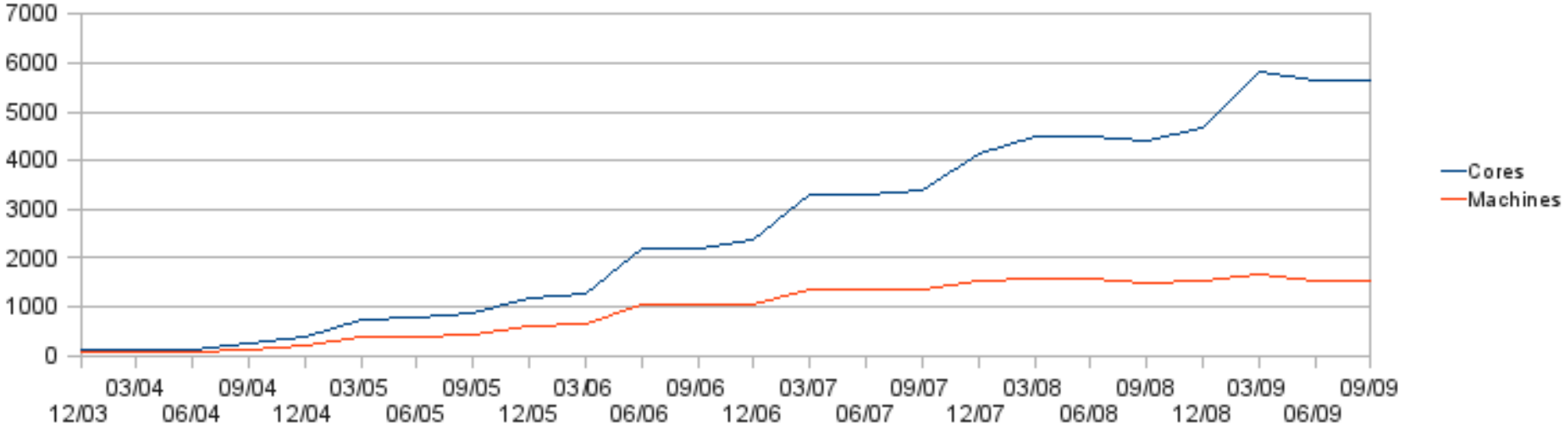

## Grid: evaluation of the gLite grid middleware

- Fully automated deployment and configuration on 1000 nodes (9 sites, 17 clusters)

gLite
Lightweight Middleware for Grid Computing

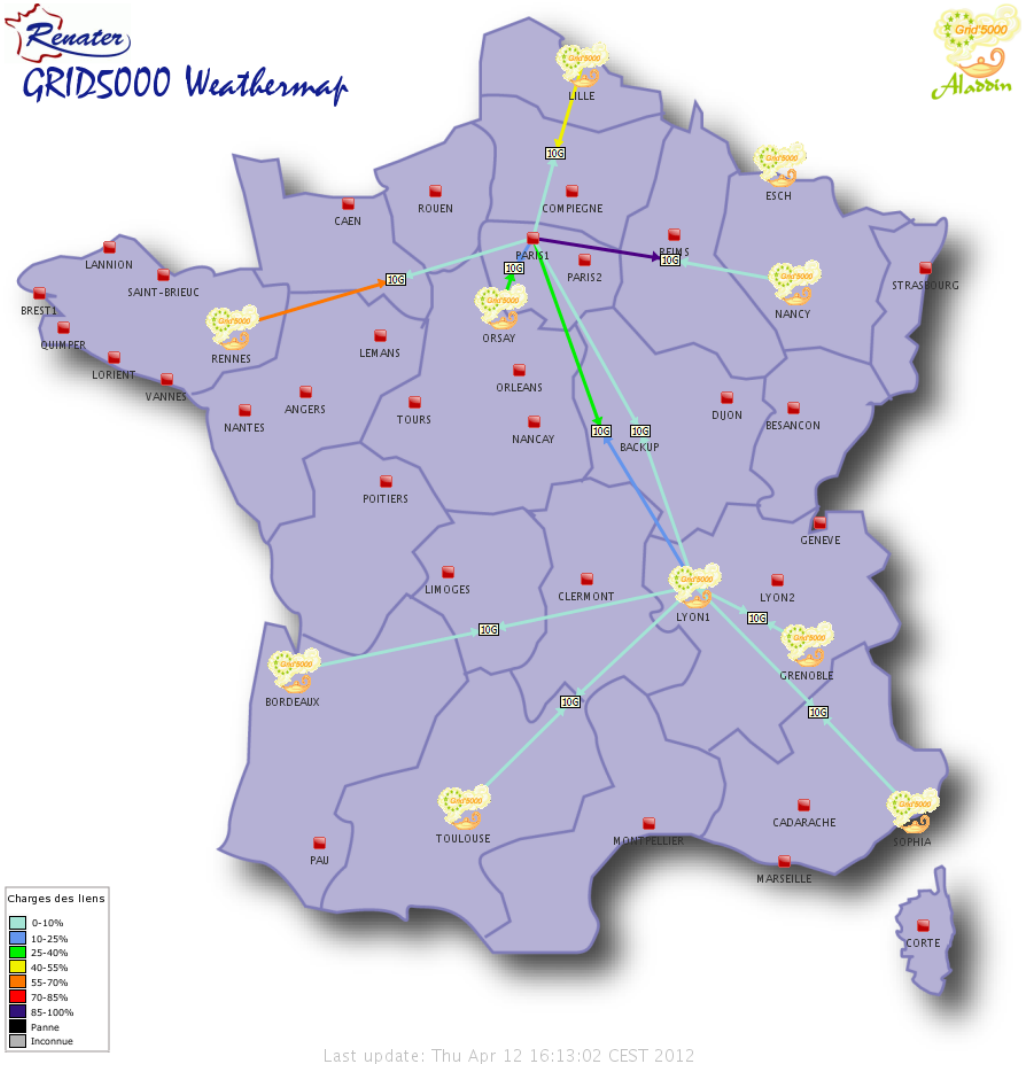
# Current status

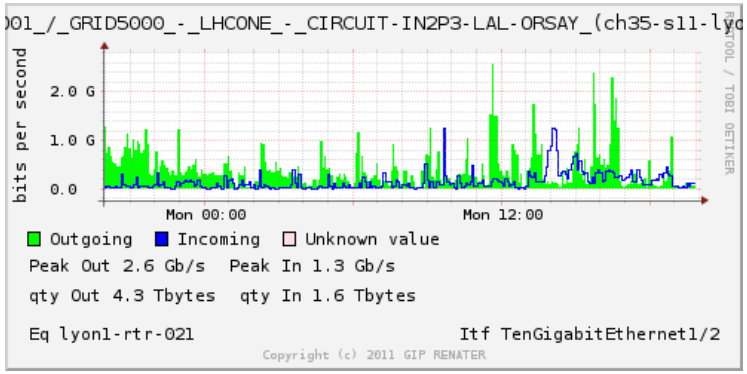
- 11 sites (1 outside France)
- 26 clusters
- 1700 nodes
- 7400 cores
- Diverse technologies:
  - Intel (60%), AMD (40%)
  - CPUs from one to 12 cores
  - Myrinet, Infiniband {S,D,Q}DR
  - Two GPU clusters
- **500+ users per year**

# Backbone network

Dedicated 10 Gbps backbone provided by RENATER (french NREN)

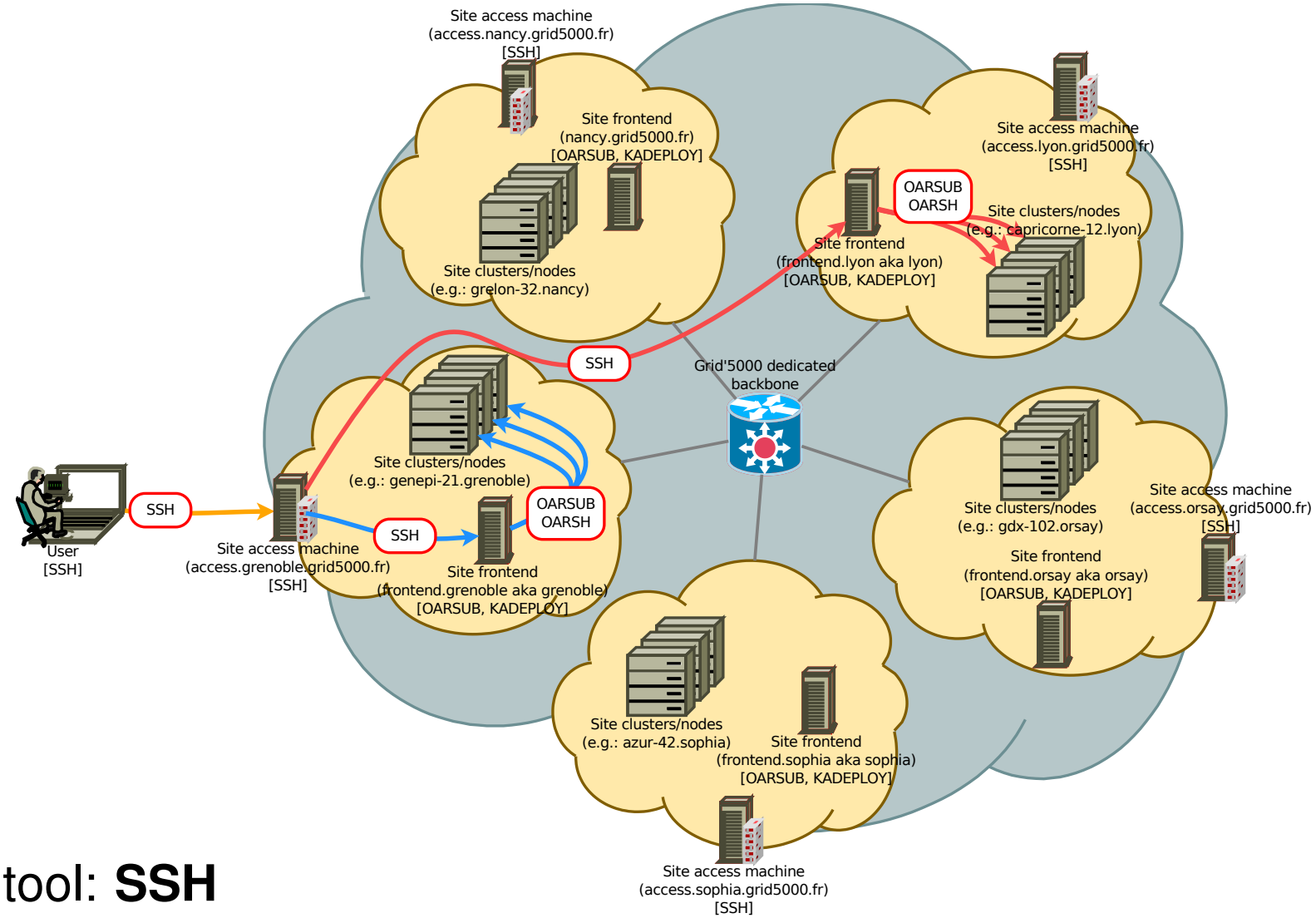



Work in progress:

- packet-level and flow-level monitoring
- bandwidth reservation and limitation

# Using Grid'5000: the user's point of view




- Key tool: **SSH**
- Private network: connect through access machines
- Data storage: **NFS** (one server per Grid'5000 site)

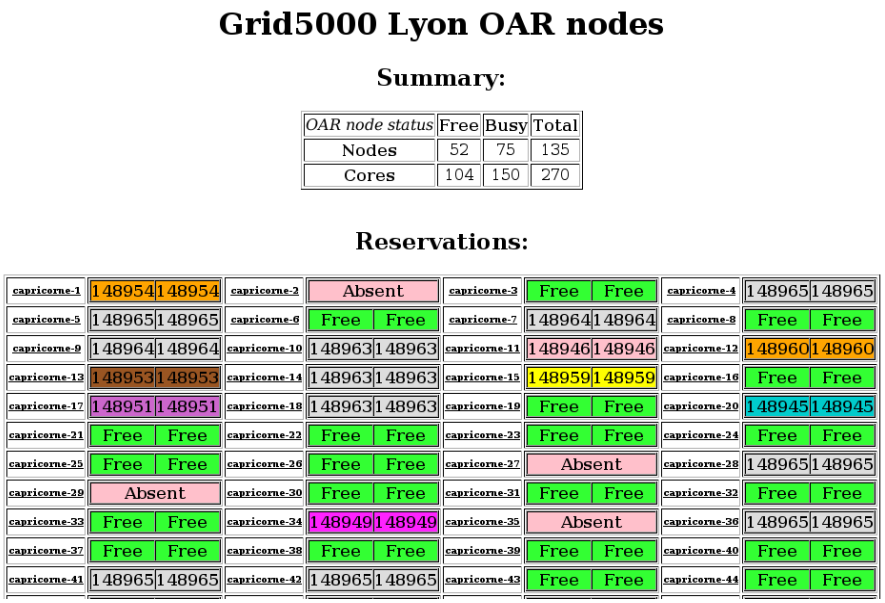
# Resource management with OAR

- Batch scheduler with specific features
  - interactive jobs
  - advance reservations
  - powerful resource matching
- Resources hierarchy: cluster / switch / node / cpu / core
- Properties: memory size, disk type & size, hardware capabilities, network interfaces, . . .
- Other kind of resources: VLANs, IP ranges for virtualization

*I want 1 core on 2 nodes of the same cluster with 4096 GB of memory and Infiniband 10G + 1 cpu on 2 nodes of the same switch with dualcore processors for a walltime of 4 hours. . .*

```
oarsub -I -l "{memnode=4096 and
ib10g='YES'}/cluster=1/nodes=2/core=1
+{cpucore=2}/switch=1/nodes=2/cpu=1,walltime=4:0:0"
```

# Resource management with OAR - visualization



**Grid5000 Lyon OAR nodes**

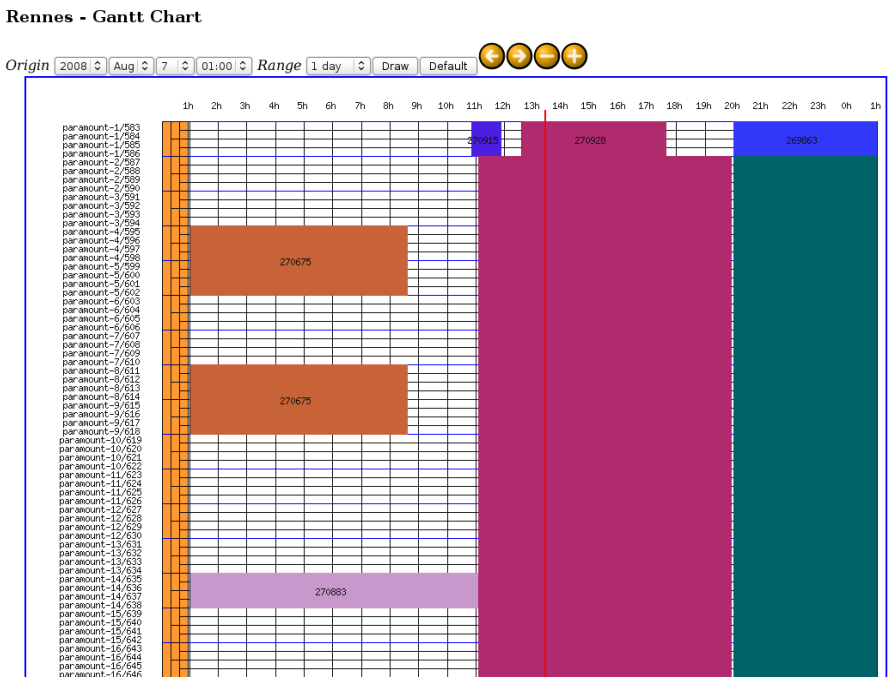
**Summary:**

| *OAR node status* | Free | Busy | Total |
|---|---|---|---|
| Nodes | 52 | 75 | 135 |
| Cores | 104 | 150 | 270 |

**Reservations:**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| capricorne-1 | 148954 | 148954 | capricorne-2 | Absent | | capricorne-3 | Free | Free | capricorne-4 | 148965 | 148965 |
| capricorne-5 | 148965 | 148965 | capricorne-6 | Free | Free | capricorne-7 | 148964 | 148964 | capricorne-8 | Free | Free |
| capricorne-9 | 148964 | 148964 | capricorne-10 | 148963 | 148963 | capricorne-11 | 148946 | 148946 | capricorne-12 | 148960 | 148960 |
| capricorne-13 | 148953 | 148953 | capricorne-14 | 148963 | 148963 | capricorne-15 | 148959 | 148959 | capricorne-16 | Free | Free |
| capricorne-17 | 148951 | 148951 | capricorne-18 | 148963 | 148963 | capricorne-19 | Free | Free | capricorne-20 | 148945 | 148945 |
| capricorne-21 | Free | Free | capricorne-22 | Free | Free | capricorne-23 | Free | Free | capricorne-24 | Free | Free |
| capricorne-25 | Free | Free | capricorne-26 | Free | Free | capricorne-27 | Absent | | capricorne-28 | 148965 | 148965 |
| capricorne-29 | Absent | | capricorne-30 | Free | Free | capricorne-31 | Free | Free | capricorne-32 | Free | Free |
| capricorne-33 | Free | Free | capricorne-34 | 148949 | 148949 | capricorne-35 | Absent | | capricorne-36 | 148965 | 148965 |
| capricorne-37 | Free | Free | capricorne-38 | Free | Free | capricorne-39 | Free | Free | capricorne-40 | Free | Free |
| capricorne-41 | 148965 | 148965 | capricorne-42 | 148965 | 148965 | capricorne-43 | Free | Free | capricorne-44 | Free | Free |

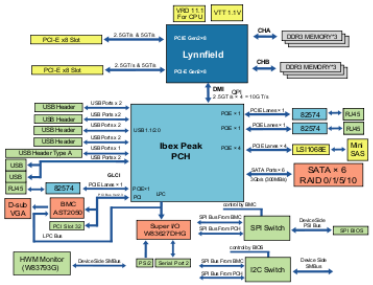
Resources status

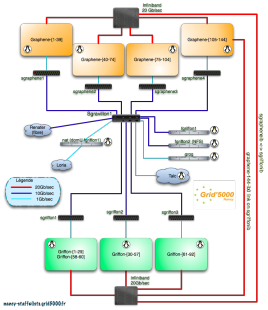



Gantt chart

# Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)

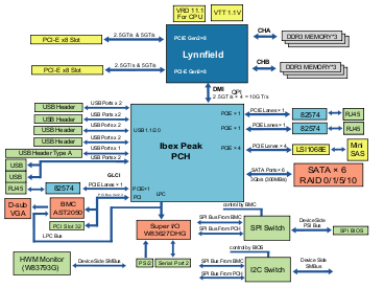


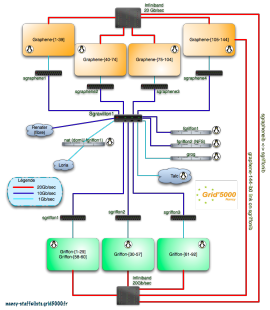




```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "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"
  }
],
```

# Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)



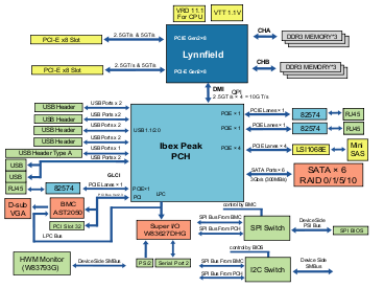

```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "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"
  }
],
```

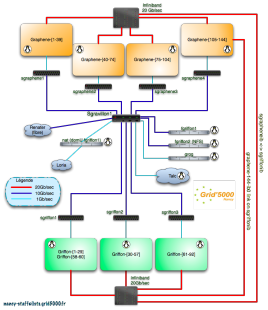

- Selecting resources
  - OAR database filled from JSON

```
oarsub -p "wattmeter='YES' and gpu='YES'"
```

# Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)





```
"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"
  }
],
```


- Selecting resources
  - OAR database filled from JSON

```
oarsub -p "wattmeter='YES' and gpu='YES'"
```

- Verifying resources
  - *G5K-checks*: validates resources against their description (detect hardware failures and misconfigurations at each boot)

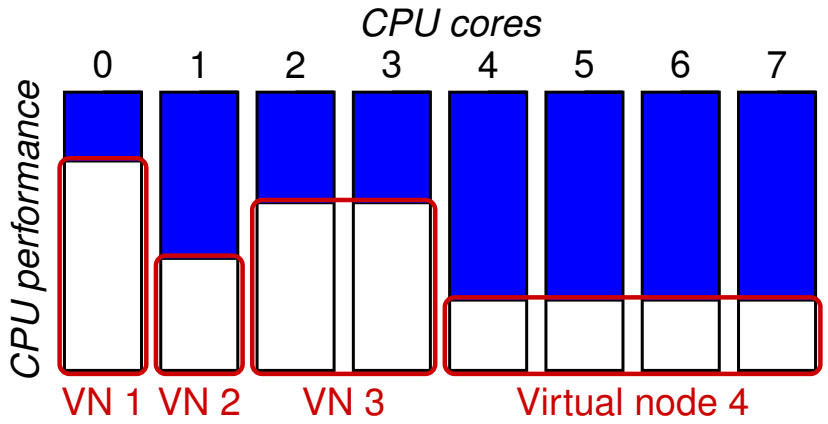
# Reconfiguring the testbed with Kadeploy

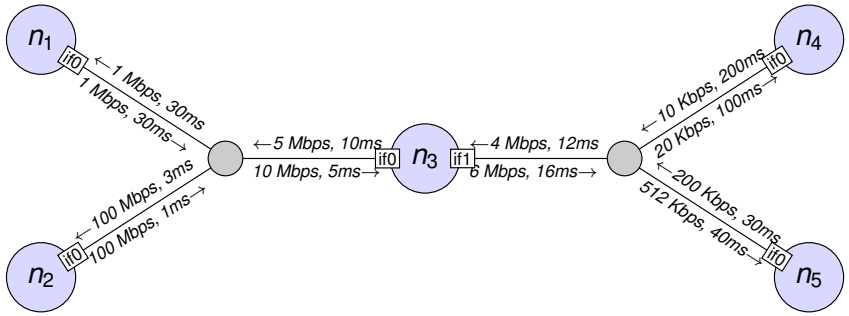
- Provides a *Hardware-as-a-Service* Cloud infrastructure
- Enable users to deploy their own software stack & get *root* access
- Standard environments provided to users
  - Customizations automated using Kameleon
- **Scalable, efficient, reliable and flexible**:
  - Chain-based and BitTorrent environment broadcast
  - **255 nodes deployed in 3 minutes**
- Command-line interface & REST API for scripting

**http://kadeploy3.gforge.inria.fr/**

# Customizing the experimental environment

- Reconfigure experimental conditions with Distem
  - Introduce heterogeneity in an homogeneous cluster
  - Emulate complex network topologies




**http://distem.gforge.inria.fr/**

# Virtualisation & Cloud XP requirements

- Efficient provisionning of machines $\rightsquigarrow$ Kadeploy
- IP addresses for Virtual Machines
- Two different solutions on Grid'5000:
  - G5K-Subnets
  - KaVLAN

# Network reservation with G5K-subnets

- Grid'5000 enable different users to run experiments concurrently
  - Need to mechanism to provide IP ranges for virtual machines
- G5K-subnets adds IP ranges reservation to OAR

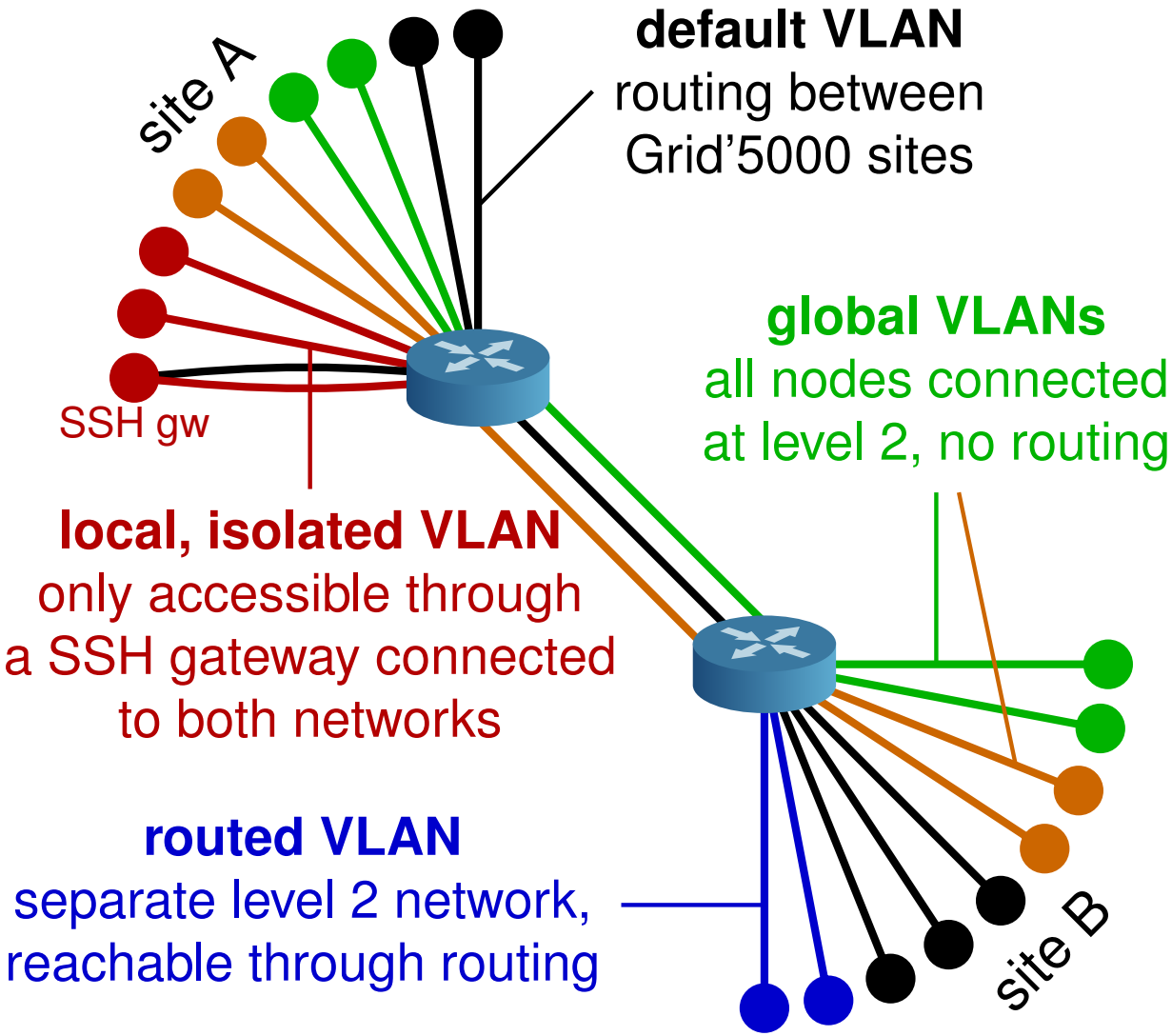
```
oarsub -l slash_22=2+nodes=8 -I
```

- IP ranges are routable inside Grid'5000
- But no isolation: one can *steal* IP addresses

# Network isolation with KaVLAN

- Reconfigures switches for the duration of a user experiment to achieve **complete level 2 isolation**:
  - Avoid network pollution (broadcast, unsolicited connections)
  - Enable users to start their own DHCP servers
  - Experiment on ethernet-based protocols
  - Interconnect nodes with another testbed without compromising the security of Grid'5000
- Relies on **802.1q (VLANs)**
- Compatible with many network equipments
  - Can use SNMP, SSH or telnet to connect to switches
  - Supports Cisco, HP, 3Com, Extreme Networks and Brocade
- Controlled with a command-line client or a REST API

# KaVLAN - different VLAN types

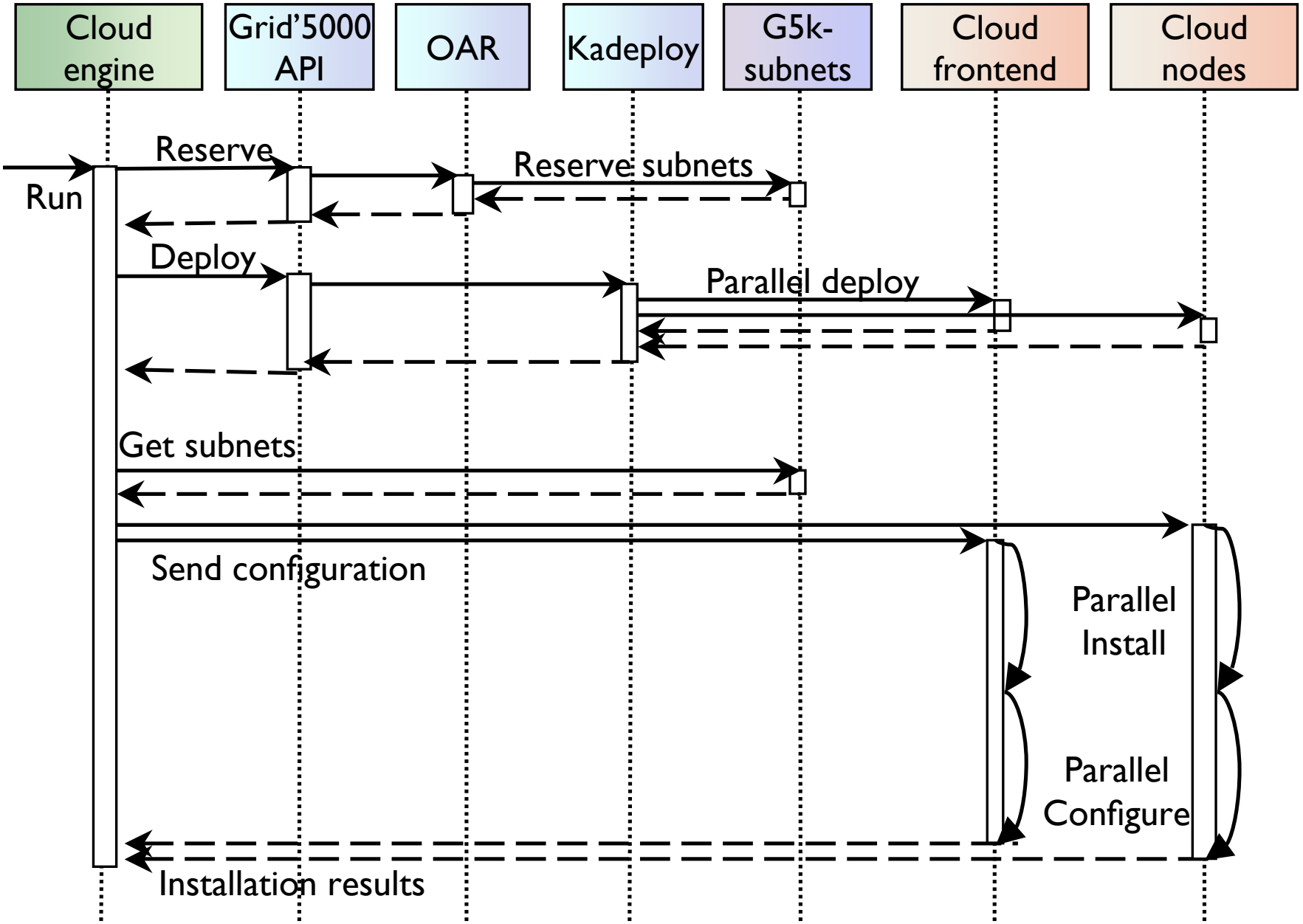


# Delivering IaaS clouds to users

- Kadeploy, G5K-subnets and KaVLAN are low-level mechanisms
- While it is possible to use them to deploy virtually any IaaS cloud stack, not everybody wants to do that
- Need for higher level tools that ease the deployment
- We will present two such tools

# Deploying IaaS Clouds with G5K-campaign

- G5K-campaign:
  - Framework for coordinating experiments
  - Relies on the Grid'5000 REST API
  - Extendable with engines
- Specific engines written for Clouds installation
  - Uses Chef cookbooks to describe the installation process
- Relies on G5K-subnets for IP ranges allocation


Cloud engine
Grid'5000 API
OAR
Kadeploy
G5k-subnets
Cloud frontend
Cloud nodes
Run
Reserve
Reserve subnets
Deploy
Parallel deploy
Get subnets
Send configuration
Parallel Install
Parallel Configure
Installation results

# Results

- Generic Cloud deployment engine supporting OpenNebula, CloudStack and Nimbus
- Can create a Cloud with hundreds of nodes
- Example deployment:
  - OpenNebula cloud
  - 80 nodes from 3 Grid'5000 sites
  - 350 virtual machines used to run Hadoop
  - less than 20 minutes to deploy
    - including 6 minutes for the initial Kadeploy run

# OpenStack on Grid'5000

- "default" mode: flatDHCP
  - OpenStack-provided DHCP server
  - cannot co-exist with the Grid'5000 DHCP server
  - Requires isolation $\rightsquigarrow$ KaVLAN
- Connection to the rest of Grid'5000 through KaVLAN gateways or dual-connected nodes
- Automated using Puppet recipes from PuppetLabs/StackForge
- Example deployment: 30 physical machines in 20 minutes
- Used as a staging area to port a bio-informatics workflow to AWS

# Future works

- Enlarge the scale of deployments
  - Requires improvements to orchestration of deployments
- Extend the testbed to support:
  - Network virtualization (OpenFlow)
  - Big Data experiments

# Conclusions

- Grid'5000: a versatile, reconfigurable testbed
  - Reconfigure the software stack using Kadeploy
  - Reserve IP ranges with G5K-subnets
  - Network isolation with KaVLAN
- Supports OpenNebula, CloudStack, Nimbus, OpenStack
- You can get an account. Mail me

`lucas.nussbaum@loria.fr`