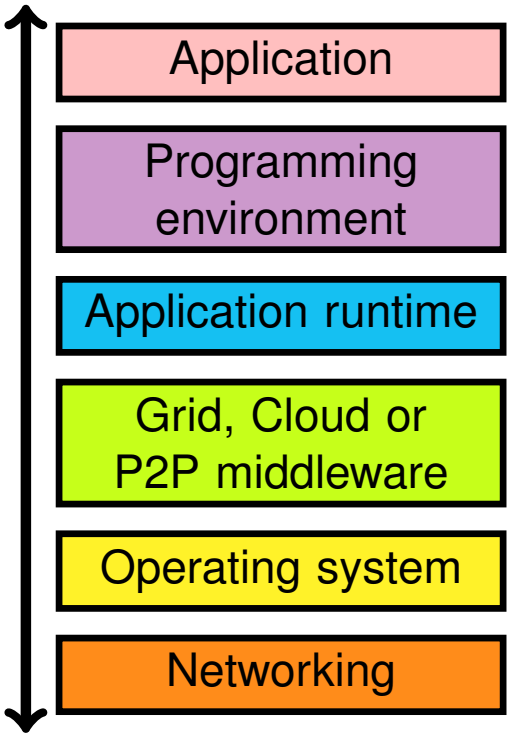
Resources Description, Selection, Reservation and Verification on a Large-scale Testbed

David Margery, Emile Morel, Lucas Nussbaum,
Olivier Richard Cyril Rohr

Grid'5000

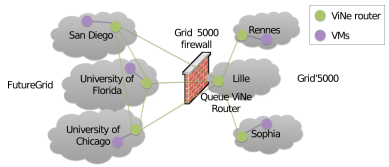
- **Testbed for research on distributed systems**
  - High Performance Computing
  - Grids
  - Peer-to-peer systems
  - Cloud computing
- History:
  - 2003: Project started (ACI GRID)
  - 2005: Opened to users
- Funding: Inria, CNRS and many local entities
- Only for research on distributed systems → no production usage
  Litmus test: *are you interested in the result of the computation?*
- 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)

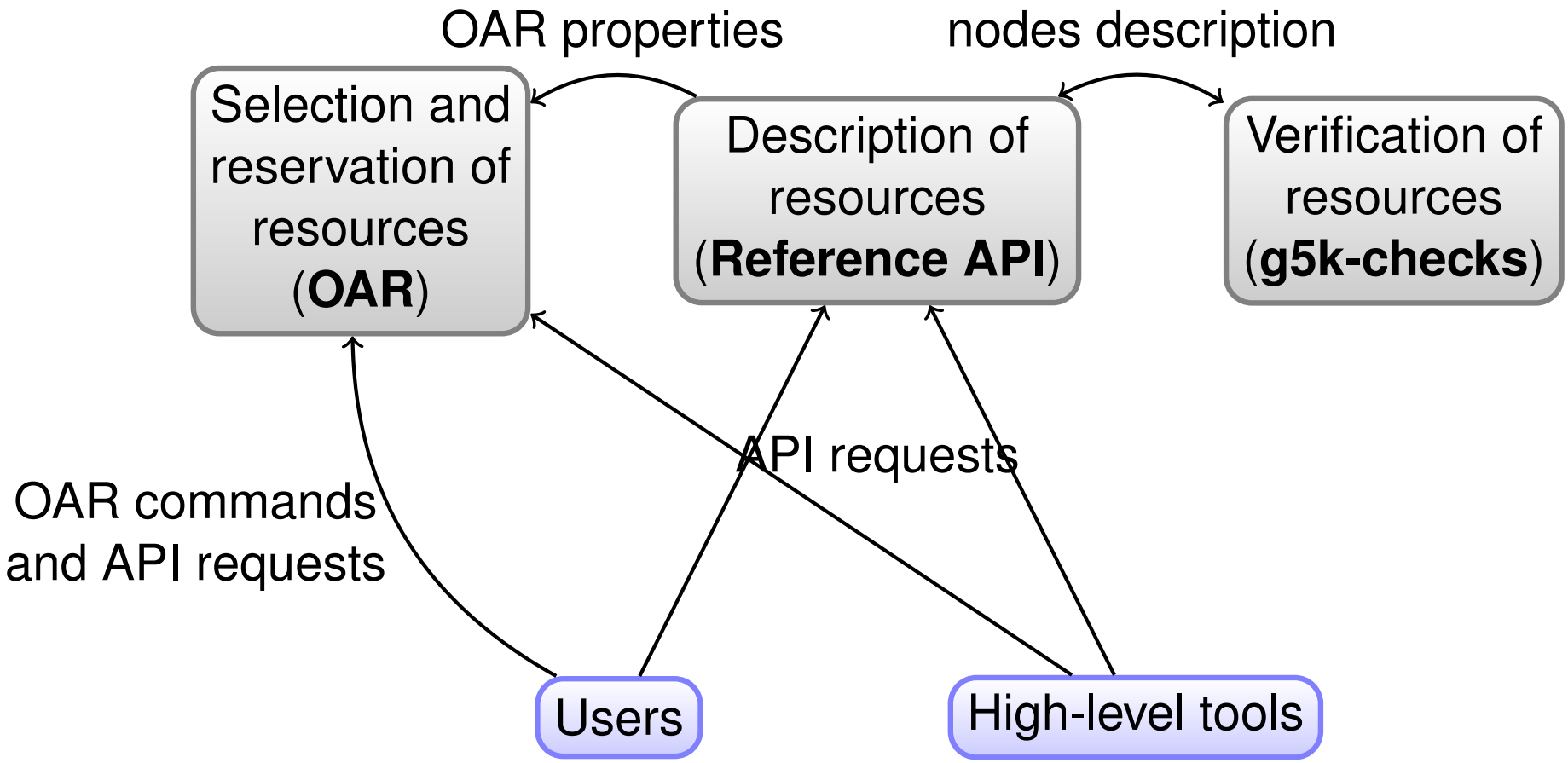
Current status

- 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**

This talk

- How we enable users to find suitable resources for experiments
- How we enable users to reserve those resources
- How we maintain an accurate description of resources

Overview of resources management



Resources description with the Reference API

- Centralized resources description:
  - As a set of JSON documents
  - Can be retrieved using a RESTful API
- Covering most of the testbed's resources:
  nodes, network equipment, power distribution units, etc.
- Detailed information: vendor/product/reference, connection, remote control and measurement access
- For users and for tools: build documentation and maps, high-level control tools
- Stored in a Git repository for archival
  State of the testbed 6 months ago?

One node in the Reference API


```
"supported_job_types" : {
   "deploy" : true,
   "besteffort" : true,
   "virtual" : "ivt"
},
"chassis" : {
   "serial" : "27Q7NZ1",
   "manufacturer" : "Dell Inc.",
   "name" : "PowerEdge R720"
},
"bios" : {
   "version" : 2,
   "release_date" : "08/29/2013",
   "vendor" : "Dell Inc."
},
"architecture" : {
   "platform_type" : "x86_64",
   "smp_size" : 2,
   "smt_size" : 16
},
"processor" : {
   "instruction_set" : "x86-64",
   "cache_l1i" : 32768,
   "version" : "E5-2650",
   "cache_l2" : 262144,
   "model" : "Intel Xeon",
   "cache_l1d" : 32768,
   "cache_l3" : 20971520,
   "vendor" : "Intel",
   "clock_speed" : 2000000000
},
```

```
"main_memory" : {
   "ram_size" : 270991937536,
},
"storage_devices" : [
   {
      "rev" : "DL10",
      "model" : "INTEL SSDSC2BB30",
      "interface" : "SATA II",
      "device" : "sda",
      "size" : 300069052416,
      "driver" : "megaraid_sas"
   },
   {
      "rev" : "DL10",
      "model" : "INTEL SSDSC2BB30",
      "interface" : "SATA II",
      "device" : "sdb",
      "size" : 300069052416,
      "driver" : "megaraid_sas"
   }
],
"mic" : {
   "mic_model" : "7120P",
   "mic" : true,
   "mic_count" : 1
},
"performance" : {
   "core_flops" : 13170000000,
   "node_flops" : 187900000000
},
```

```
"network_adapters" : [
   {
      "ip" : "172.16.68.1",
      "rate" : 10000000000,
      "mountable" : true,
      "interface" : "Ethernet",
      "mounted" : true,
      "mac" : "b8:ca:3a:69:12:68",
      "enabled" : true,
      "version" : "82599EB",
      "device" : "eth0",
      "switch_port" : "F1",
      "switch" : "gw-nancy",
      "management" : false,
      "driver" : "ixgbe",
      "vendor" : "intel"
   },
   {
      "version" : "IDRAC7",
      "ip" : "172.17.68.1",
      "device" : "bmc",
      "switch_port" : "1/0/41",
      "rate" : 100000000,
      "switch" : "sgraphene3-ipmi",
      "mountable" : false,
      "interface" : "Ethernet",
      "mounted" : false,
      "mac" : "f0:1f:af:e1:9a:0c",
      "management" : true,
      "vendor" : "DELL",
      "enabled" : true
   }
]
```

Resources selection and reservation with OAR

- Roots of Grid'5000 in the HPC community
  $\rightsquigarrow$ Natural idea to use a HPC Resource Manager
- Supports resources properties ($\approx$ tags)
  - Can be used to select resources (multi-criteria search)
  - Generated from Reference API
- Supports advance reservation of resources
  - In addition to typical HPC resource managers's *batch* mode
  - Request resources at a specific time
  - On Grid'5000: used for special policy:
    Large experiments during nights and week-ends
    Experiments preparation during day

Using properties to reserve specific resources

Reserving two nodes for two hours. Nodes must have a GPU and power monitoring:

```
oarsub -p "wattmeter='YES' and gpu='YES'" -l nodes=2,walltime=2 -I
```

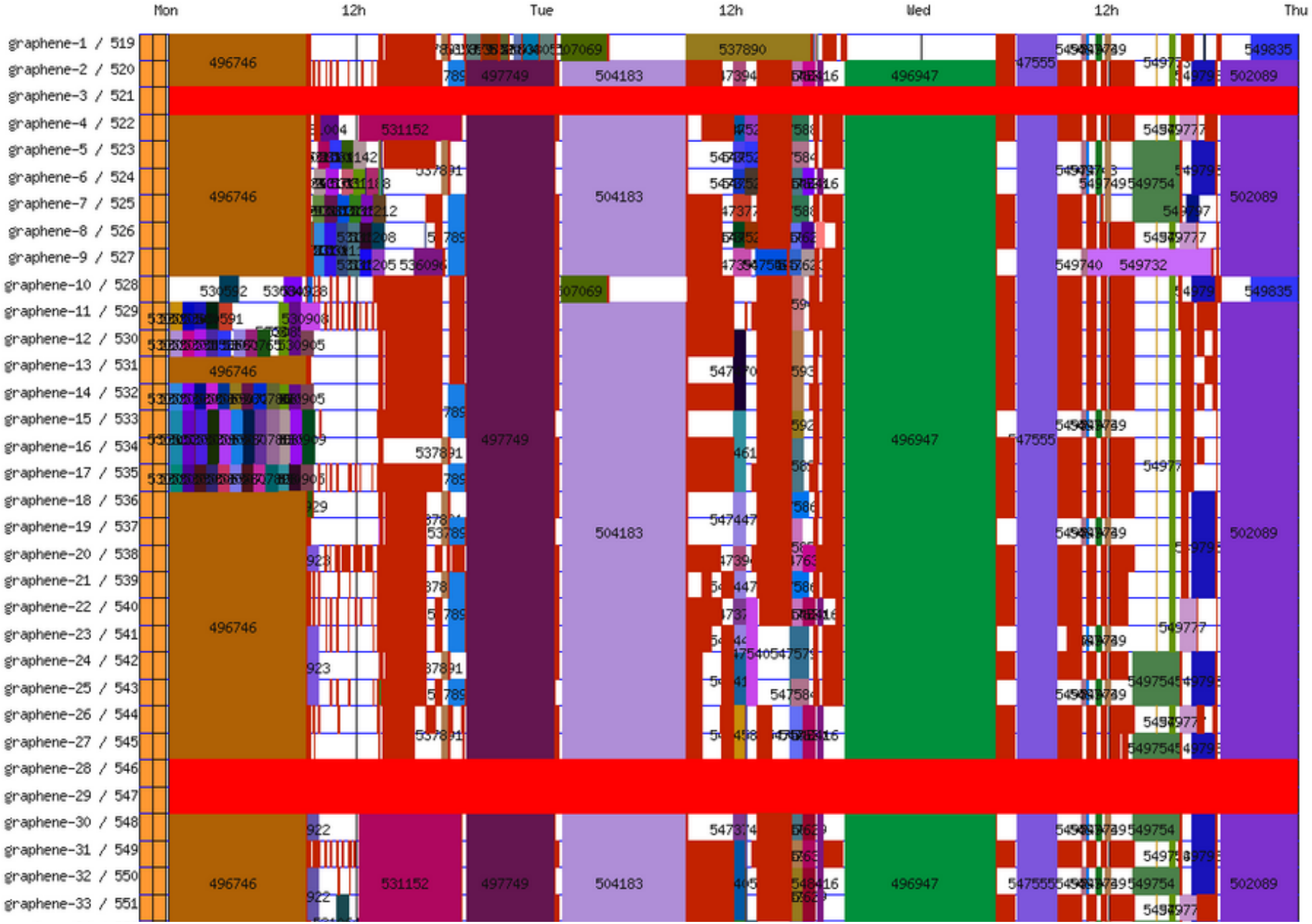
Reserving one node on cluster a, and two nodes with a 10 Gbps network adapter on cluster b:

```
oarsub -l "{cluster='a'}/nodes=1+{cluster='b' and eth10g='Y'}/nodes=2,walltime=2"
```

Advance reservation of 10 nodes on the same switch with support for Intel VT (virtualization):

```
oarsub -l "{virtual='ivt'}/switch=1/nodes=10,walltime=2" -r '2014-11-08 09:00:00'
```

Visualization of usage



Resources verification

- Inaccuracies in resources descriptions $\rightsquigarrow$ dramatic consequences:
  - Mislead researchers into making false assumptions
  - Generate wrong results $\rightsquigarrow$ retracted publications!
- Happen frequently: maintenance, broken hardware (e.g. RAM)
- Our solution: g5k-checks
  - Runs at node boot (can also be run manually)
  - Retrieves current description of node in Reference API
  - Acquire information on node using OHAI, ethtool, etc.
  - Compare with Reference API

Conclusions

- Integrated and functional solution for management of resources
  - Description
  - Selection and reservation
  - Verification
- Main area of future work: verification of resources
  - Check performance, not just description
    $\rightsquigarrow$ Discover more problems
  - Challenges: testing time, hardware wear out