CLOSER 2012

Porto, Portugal

18 - 21 April, 2012

2nd International Conference on Cloud Computing and Services Science

Experimental Computer Science

Approaches and instruments

F. Desprez

INRIA Grenoble Rhône-Alpes, LIP ENS Lyon, Team Avalon

Joint work with E. Jeannot, A. Lèbre, D. Margery, L. Nussbaum, C. Perez, O. Richard

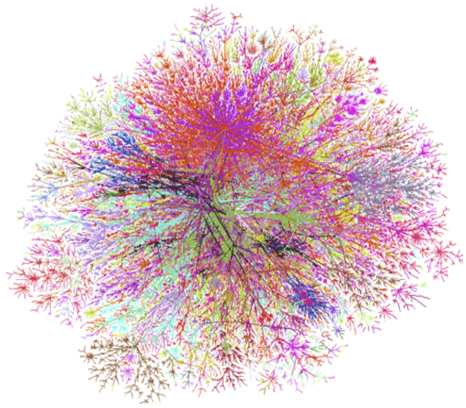
"*One could determine the different ages of a science by the technic of its measurement instruments*"

Gaston Bachelard

The Formation of the scientific mind

Agenda

- Experimental computer Science
- Overview of GRID’5000
- GRID’5000 Experiments
- Related Platforms



EXPERIMENTATION FOR DISTRIBUTED SYSTEMS

The discipline of computing: an experimental science

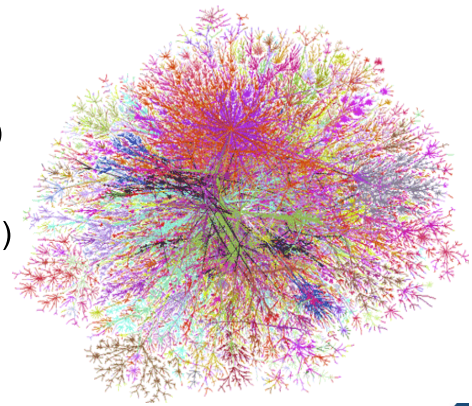
The reality of computer science:

-information
-computers, network, algorithms, programs, etc.

Studied objects (hardware, programs, data, protocols, algorithms, network): more and more complex.

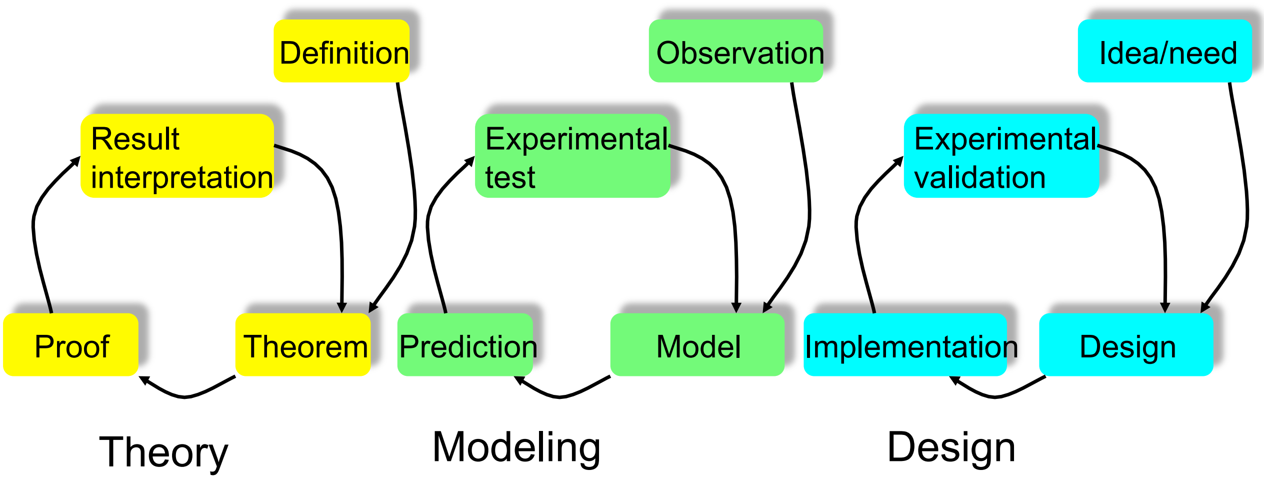
Modern infrastructures:

- Processors have very nice features
  - Cache
  - Hyperthreading
  - Multi-core
- Operating system impacts the performance (process scheduling, socket implementation, etc.)
- The runtime environment plays a role (MPICH≠OPENMPI)
- Middleware have an impact (Globus≠GridSolve)
- Various parallel architectures that can be:
  - Heterogeneous
  - Hierarchical
  - Distributed
  - Dynamic



Three paradigms of computer science

Three feedback loops of the three paradigm of CS [Denning 89], [Feitelson 07]



Experimental culture: great successes

Experimental computer science at its best [Denning1980]:

- Queue models (Jackson, Gordon, Newel, '50s and 60's). Stochastic models validated experimentally
- Paging algorithms (Belady, end of the 60's). Experiments to show that LRU is better than FIFO

ACM President's Letter

Performance Analysis:
Experimental Computer Science at Its Best

Peter J. Denning

What is experimental computer science?[1] This question has been widely discussed ever since the Feldman Report was published in 1979 [13]. Surprisingly, computer scientists disagree on the answer. Some believe that it is large system development projects, i.e., computer and software engineering. Some believe that it is all the nontheoretical activities of computer science, especially those conferring "hands-on" experience. Some believe that computer science is still too young and no operational definition is yet available.

I disagree. There are well-established standards for experimental science. The field of performance analysis meets these standards and provides the best examples of experimental computer science.

Hypotheses, Apparatus, and Tests

Science classifies knowledge. Experimental science classifies knowledge derived from observations. The experimenters set up an apparatus, use it to collect data about a phenomenon, and then analyze the data to sustain or refute hypotheses about that phenomenon. The result of one line of investigation may be a model that becomes the apparatus for a later line of investigation.

The experimental apparatus may be a real system, a subsystem, or a model. The hypothesis may concern a law of nature, characteristics of people, design principles of computers, or the quality of models.

The key affects of experimental science are an apparatus for collecting data, a hypothesis, and systematic analysis to see whether the data supports the hypothesis. There is considerable latitude in the types of hypotheses and apparatuses that may be used.

Two Examples

It is no accident that the best examples of experimental computer science can be found in the field called performance analysis. The primary aim of this field is the construction, validation, and evaluation of computer-system models that are robust enough to be used for prediction.

I will cite two examples of experimental science in this field. The first is the M44/44X project at the IBM T.J. Watson Research Lab in the middle 1960s; this project evaluated concepts of time sharing, especially memory policies and program behavior, by implementing and measuring them. This example illustrates experimental work in computer systems architecture. The second example is the study of queueing network models since 1971; this line of investigation illustrates how strong interaction between theory and experiment can lead to a conceptually simple model that may serve as the starting point for future lines of investigation. This example illustrates how yesterday's theorems can become tomorrow's definitions.

The M44/44X Project

The M44/44X project was conducted at the IBM Research Center in Yorktown Heights, N.Y., in the middle 1960s. Its purpose was to evaluate the emerging concepts of time-sharing systems by reducing them to practice and measuring them. The central principle of its architecture was a set of virtual machines, one for each user. The main machine was an IBM 7044 (M44 for short) and each virtual machine was an experimental image of the 7044 (44X for short). Virtual memory and multiprogramming were used to implement the address spaces of the 44Xs in the memory hierarchy of the M44. This machine served as the

[1] Part of this essay is based on my editorial, "What is Experimental Computer Science?" in *Communications of the ACM*, October 1980, pp. 543–544. The rest is based on my speech at the ACM SIGMETRICS Symposium on Performance Modeling, September 15, 1981.

725 Communications of the ACM November 1981 Volume 24 Number 11

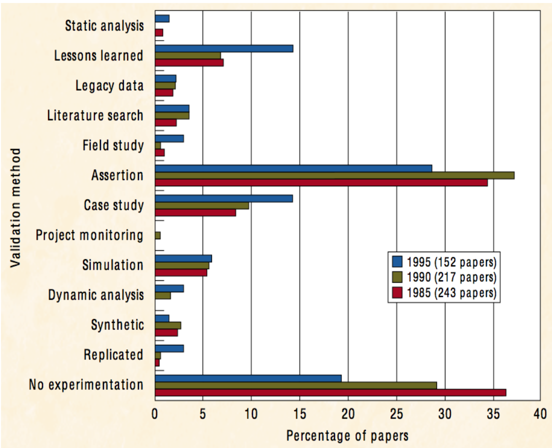
Experimental culture not comparable with other science

Different studies:

- In the 90's: between 40% and 50% of CS ACM papers requiring experimental validation had none (15% in optical engineering) [Lukovicz et al.]
- "*Too many articles have no experimental validation*" [Zelkowitz and Wallace 98]:

612 articles published by IEEE.

- Quantitatively more experiments with times



Computer science not at the same level than some other sciences:

- Nobody redo experiments (no funding)
- Lack of tool and methodologies

M.V. Zelkowitz and D.R. Wallace. Experimental models for validating technology. Computer, 31(5):23-31, May 1998.

Computer Science Experiments

Many domains:

- Complex system modeling and algorithm design (clouds, parallel machines, modern processors, network)
- Bio-informatics and others sciences (geology, atmosphere, etc.)
- Computer-System Security (virus)
- Human–computer Interaction (HCI)
- Computational linguistic
- Etc.

"Good experiments"

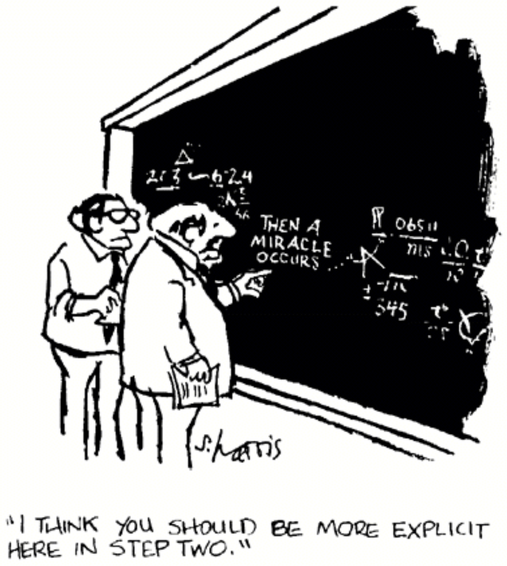
A **good experiment** should fulfill the following properties

- **Reproducibility**: *must* give the same result with the same input
- **Extensibility:** *must* target possible comparisons with other works and extensions (more/other processors, larger data sets, different architectures)
- **Applicability:** *must* define realistic parameters and *must* allow for an easy calibration
- **"*Revisability*":** when an implementation does not perform as expected, *must* help to identify the reasons

Analytic modeling

Purely analytical (mathematical) models

- Demonstration of properties (theorem)
- Models need to be tractable: over-simplification?
- Good to understand the basic of the problem
- Most of the time ones still perform a experiments (at least for comparison)



For a practical impact (especially in distributed computing): analytic study not always possible or not sufficient

Experimental Validation

A good alternative to analytical validation

- Provides a comparison between algorithms and programs
- Provides a validation of the model or helps to define the validity domain of the model

Several methodologies

- **Simulation** (SimGrid, NS, …)
- **Emulation** (MicroGrid, Wrekavoc, …)
- **Benchmarking** (NAS, SPEC, Linpack, ….)
- **Real-scale** (Grid'5000, FutureGrid, OpenCirrus, PlanetLab, …)

Properties of methodologies

Enabling good experiments:

Control:

- essential to know which part of the model or the implementation are evaluated
- allows testing and evaluating each part independently

Reproducibility:

- base of the experimental protocol
- Ensured experimental environment

Realism:

- Experimental condition: always (somehow) synthetic conditions
- Level of abstraction depends on the chosen environment
- Three levels of realism:
  1. **Qualitative:** experiment says $A_1 \geq A_2$ then in reality $A_1 \geq A_2$
  2. **Quantitative:** experiment says $A_1 = k * A_2$ then in reality $A_1 = k * A_2$
  3. **Predictive.**
- Problem of validation

Simulation

Simulation: predict parts of the behavior of a system using an approximate model

- Model = Collection of attributes + set of rules governing how elements interact
- Simulator: computing the interactions according to the rules

Models wanted features

- Accuracy/realism: correspondence between simulation and real-world
- Scalability: actually usable by computers (fast enough)
- Tractability: actually usable by human beings (understandable)
- *"Instanciability"*: can actually describe real settings (no magic parameters)

$\Rightarrow$ Scientific challenges

H. Casanova, A. Legrand and M. Quinson. **SimGrid: a Generic Framework for Large-Scale Distributed Experiments**. 10th IEEE International Conference on Computer Modeling and Simulation, 2008.

Emulation

Emulation: executing a real application on a model of the environment

Two approaches

- Sandbox/virtual machine: confined execution on (a) real machine(s). syscall catch. Ex: MicroGrid
- Degradation of the environment (to make it heterogeneous): direct execution. Ex: Wrekavoc/distem

Benchmark

Synthetic application

- Test workload
- Model of a real application workload
- Shared by other scientists
- Do not care for the output (e.g. random matrix multiplication).

Classical benchmark

- NAS parallel benchmarks (diff. kernels, size and class).
- Linpack (Top 500)
- SPEC
- Montage workflow
- Archive
  - Grid Workload archive (GWA)
  - Failure trace archive (FTA)

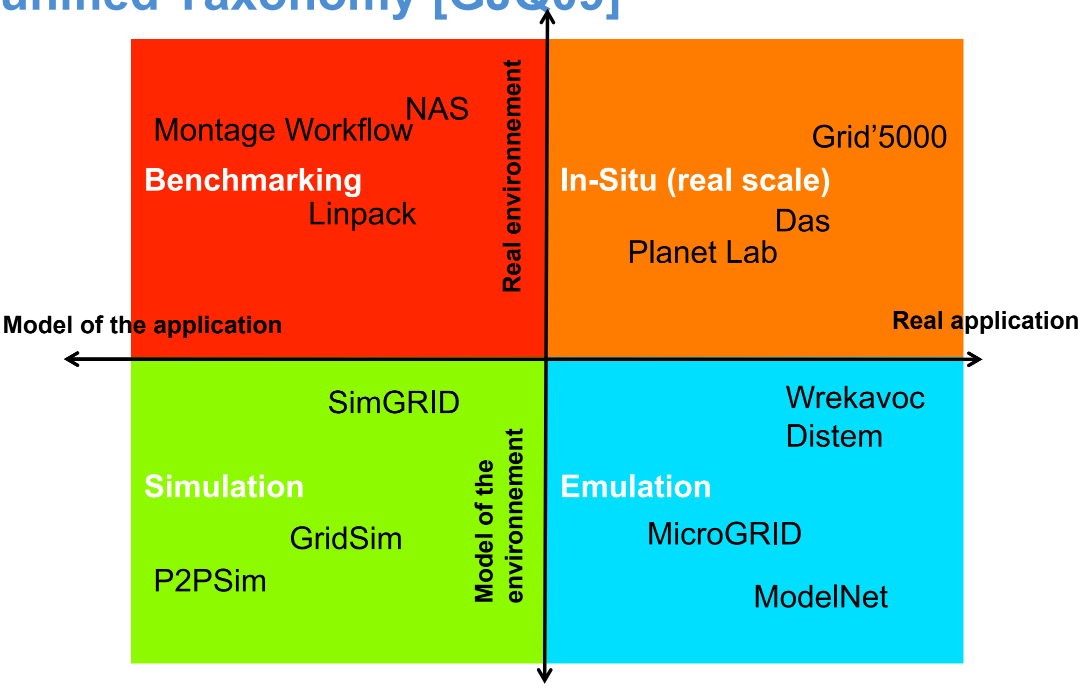
In-situ/Real scale

Real application executed on real (dedicated) hardware/environment

Challenges

- Configuration
- *"Genericity"*
- Experiment cycle time
- Ease of use
- Cost, availability

A unified Taxonomy [GJQ09]

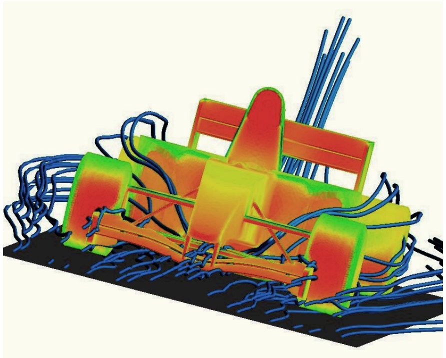


Warning: running a benchmark on an emulator is different than doing a simulation

J. Gustedt, E. Jeannot and M. Quinson **Experimental Methodologies for Large-Scale Systems: a Survey.** PPL, 19(3):399–418, September 2009

Experimentation for distributed systems

Simulation



1. Model application
2. Model environment
3. Compute interactions

Real-scale experiments

Execute the real application on real machines

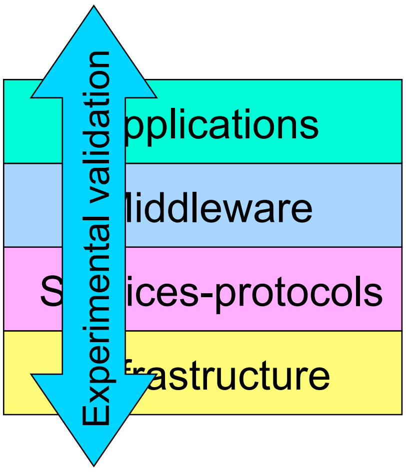
Complementary solutions

Work on algorithms
Scalable, more user friendly

Work on applications
Closer to production use

F. Desprez - Closer 2012 13/04/12 - 19

Environment Stack



Research issues at each layer of the stack

- algorithms
- software
- data
- models
- …

Problem of experiments

- Testing and validating solutions and models as a scientific problematic
- Questions:
  - what is a good experiment ?
  - which methodologies and tools to perform experiments?
  - advantages and drawbacks of these methodologies/tools?

Shared/Common Testbeds (i.e. prod. Grids)

Not designed for long term exclusive access for a project

- Difficult to use as a always on demonstrator of your work
- But if the testbed is not well established, difficult to use to prove your point

Not tailored to specific needs

- Always a setup cost, and an adaptation cost as the facility evolves
- A compromise must be found to ensure setup cost stays small in respect to usage time

Are themselves subject to research

- The gap between an abstract description of the testbed needed by a particular project and a concrete implementation on one testbed has not been bridged yet

Experiment-driven research has a lot of benefits, but also a cost for the researcher

- Experiments have to be planned and well thought out

GRID'5000

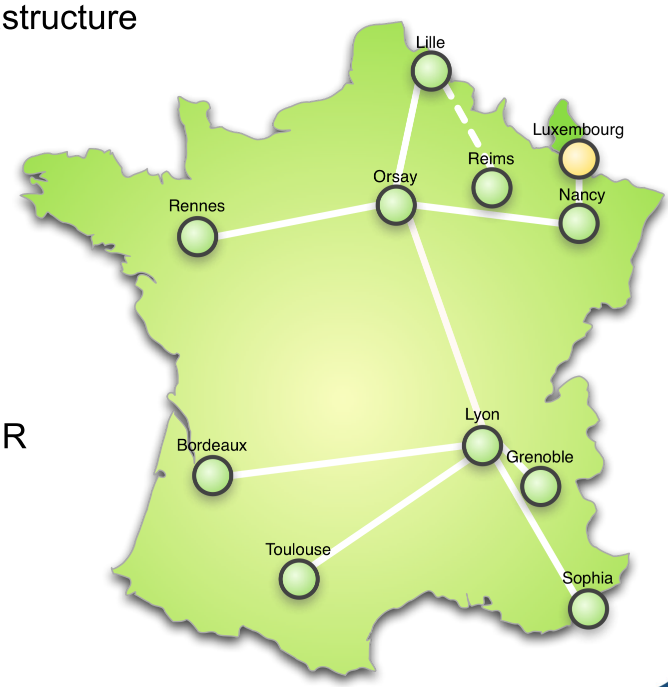
www.grid5000.fr/

GRID'5000

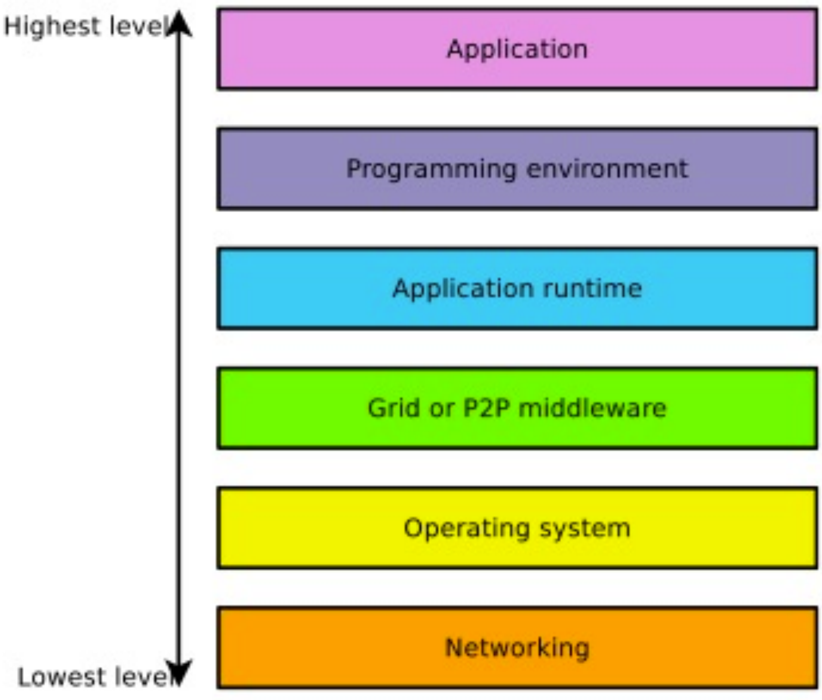
- **Testbed for research on distributed systems**
  - Born from the observation that me need a better and larger testbed
  - High Performance Computing, Grids, Peer-to-peer systems, Cloud computing
  - A complete access to the nodes' hardware in an exclusive mode (from one node to the whole infrastructure)
  - RIaaS : Real Infrastructure as a Service ! ?
- **History, a community effort**
  - 2003: Project started (ACI GRID)
  - 2005: Opened to users
- **Funding**
  - Inria, CNRS,and many local entities (regions, universities)
- **One rule:** only for research on distributed systems
  - → no production usage
  - Free nodes during daytime to prepare experiments
  - Large-scale experiments during nights and week-ends

Current Status

- 11 sites (1 outside France)
  - New sites are joining the infrastructure (Nantes, Porto-Allegre)
- 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
- More than **500 users** per year

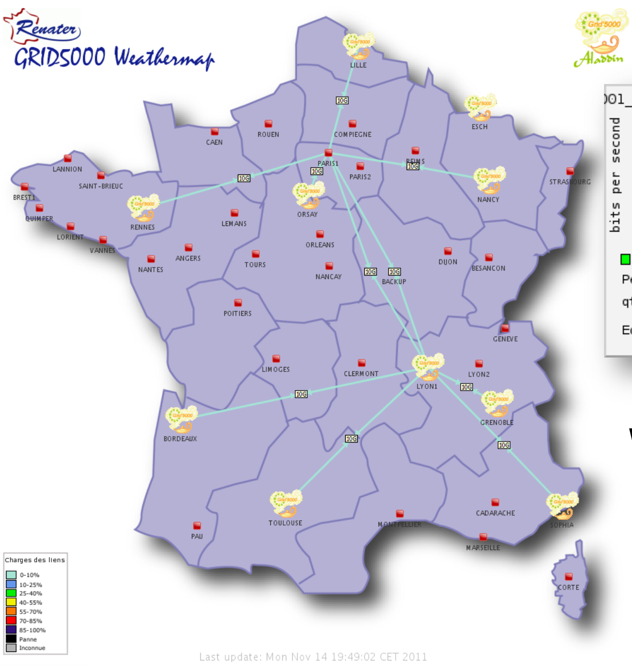


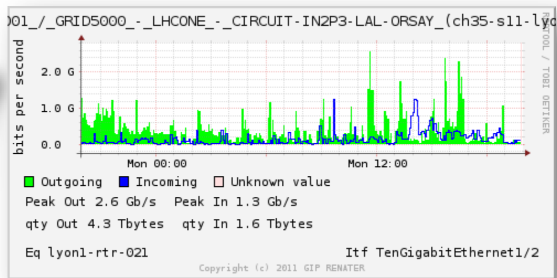
A Large Research Applicability



Backbone Network

Dedicated 10 Gbps backbone provided by Renater (french NREN)



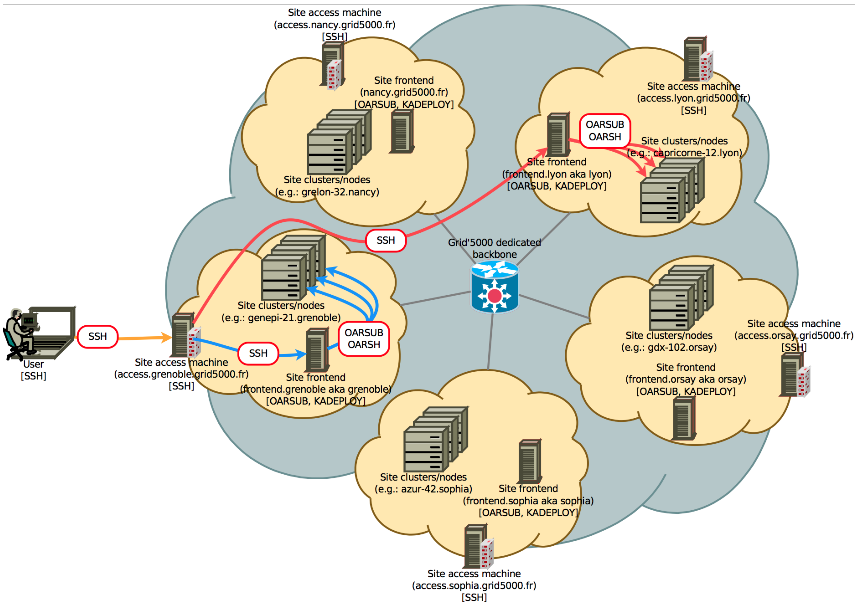


Work in progress

- Packet-level and flow level monitoring
- Bandwith reservation and limitation

Using GRID'5000: User's Point of View

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



GRID'5000 Software Stack

- Resource management: **OAR**
- System reconfiguration: **Kadeploy**
- Network isolation: **KaVLAN**
- Monitoring: **Ganglia, Kaspied, Energy**
- Putting all together **GRID'5000 API**

Resource Management: OAR

Batch scheduler with specific features

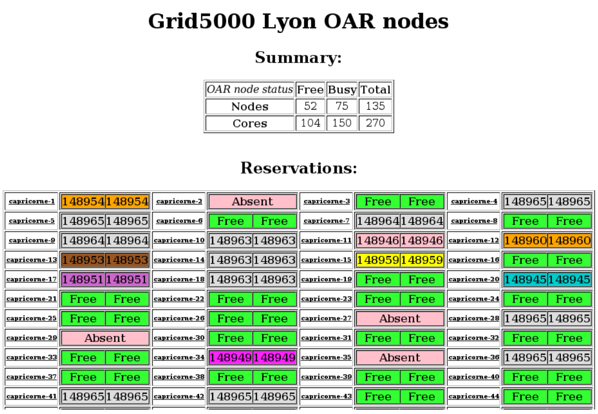
- interactive jobs
- advance reservations
- powerful resource matching

- **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: OAR, Visualization



Grid5000 Lyon OAR nodes

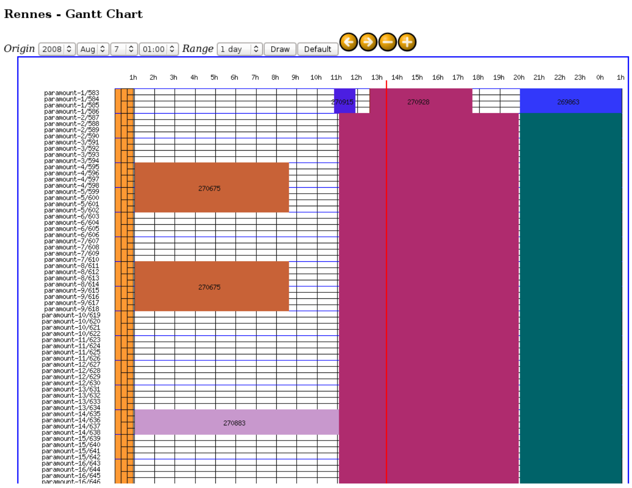
Summary:

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

Resource status



Gantt chart

Kadeploy – Scalable Cluster Deployment Tool

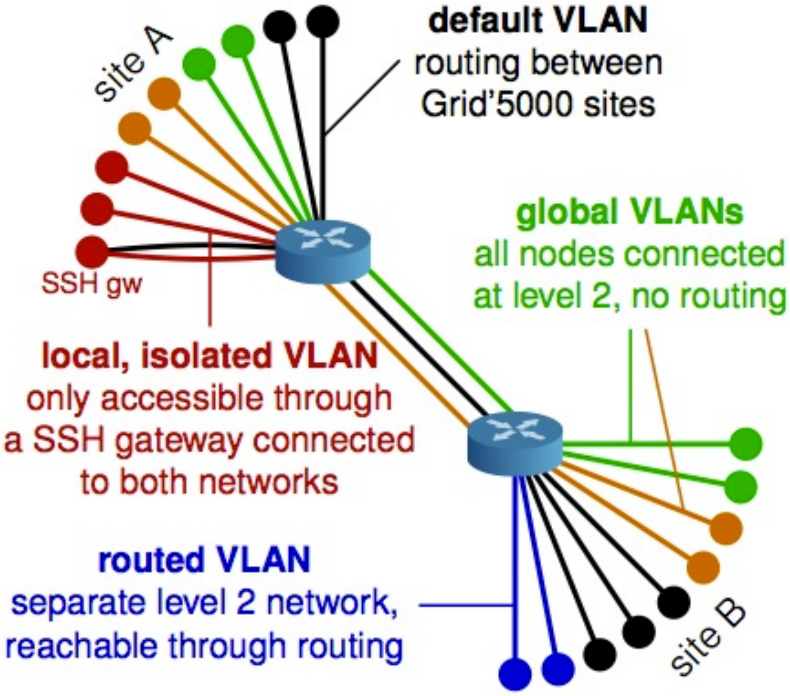
- Provides a *Hardware-as-a-Service* Cloud infrastructure
- Built on top of PXE, DHCP, TFTP
- **Scalable, efficient, reliable and flexible**
  - Chain-based and BitTorrent environment broadcast
- **255 nodes deployed in 7 minutes** (latest scalability test 4000 nodes)
- Support of a **broad range of systems** (Linux, Xen, *BSD, etc.)
- Command-line interface & asynchronous interface (REST API)
- Similar to a cloud/virtualization provisionning tool (but on real machines)
- Choose a system stack and deploy it over GRID'5000 !

kadeploy3.gforge.inria.fr

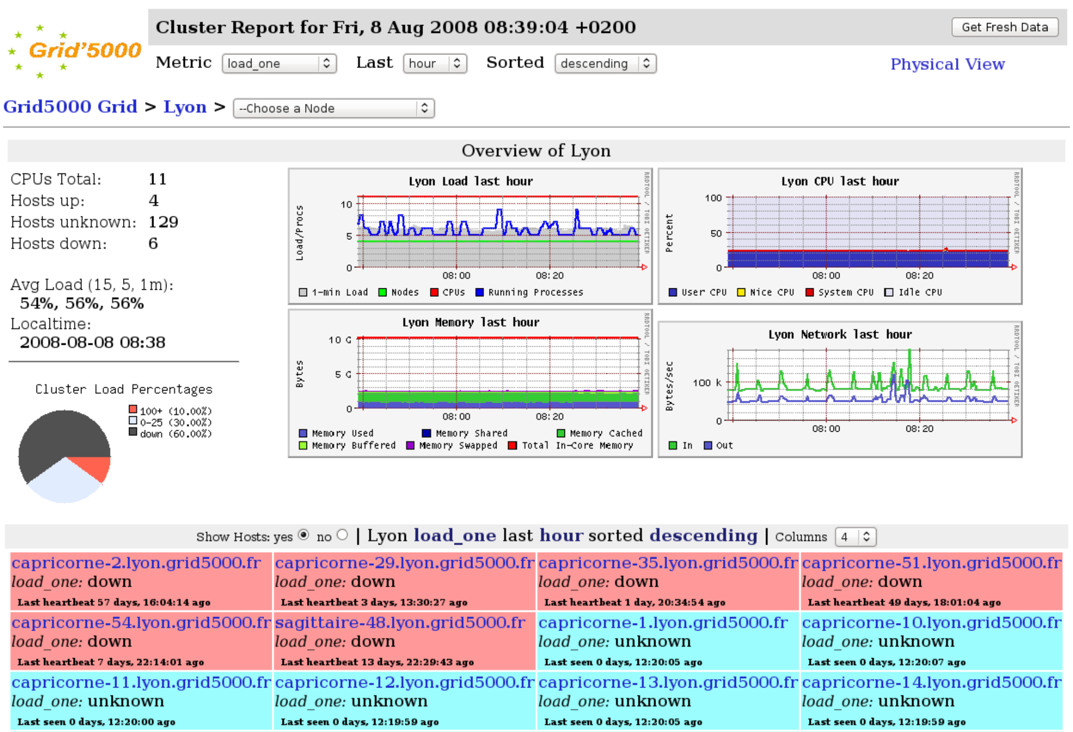
Network Isolation: KaVLAN

- Reconfigures switches for the duration of a user experiment to **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

Network Isolation: KaVLAN, cont

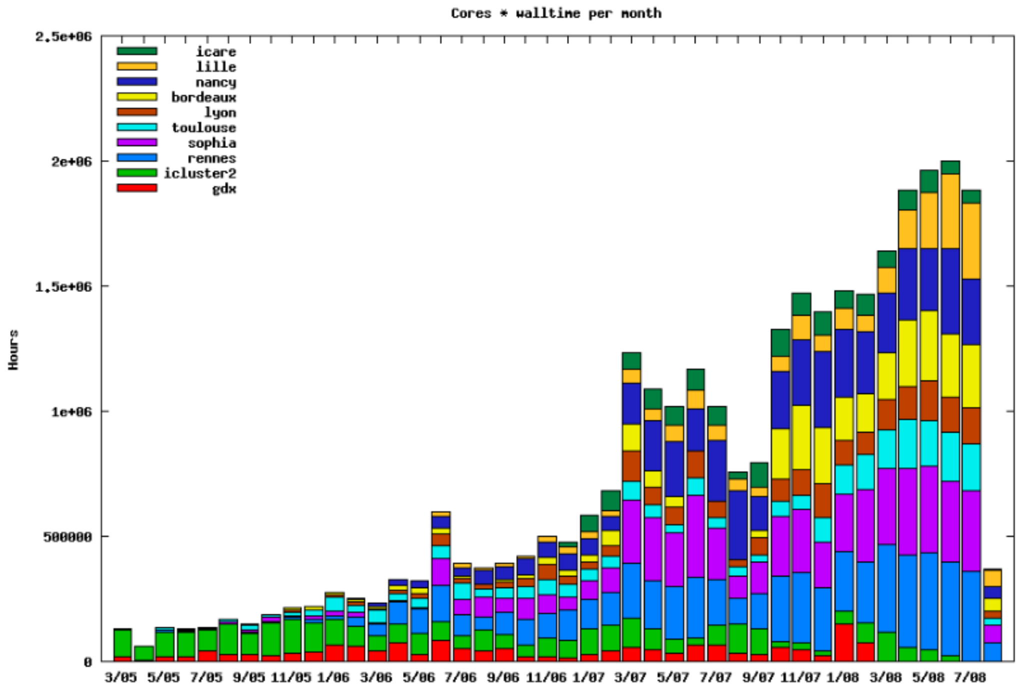


Monitoring, Ganglia



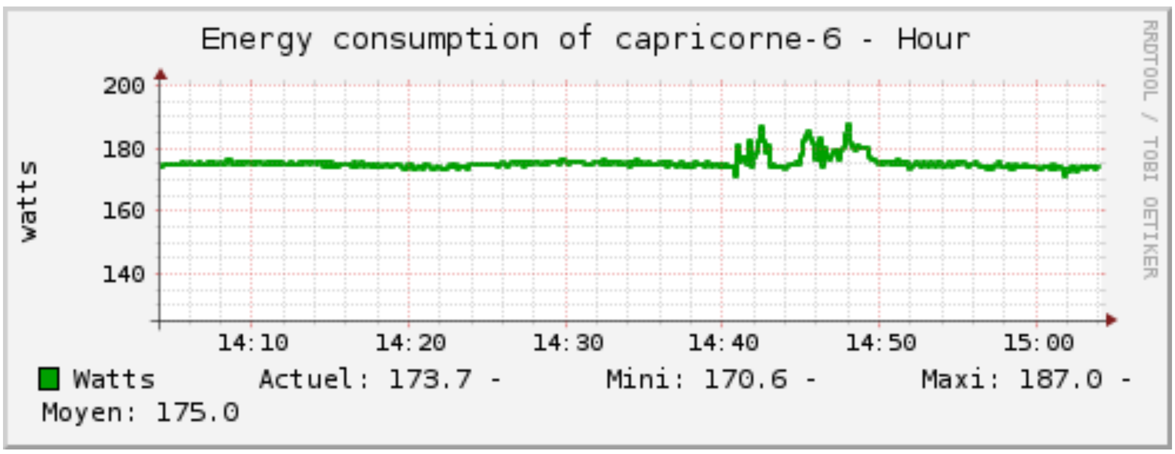
Monitoring, Kaspied

Usage per month per cluster (GRID'5000 usage over time)



Monitoring, Energy

Power consumption



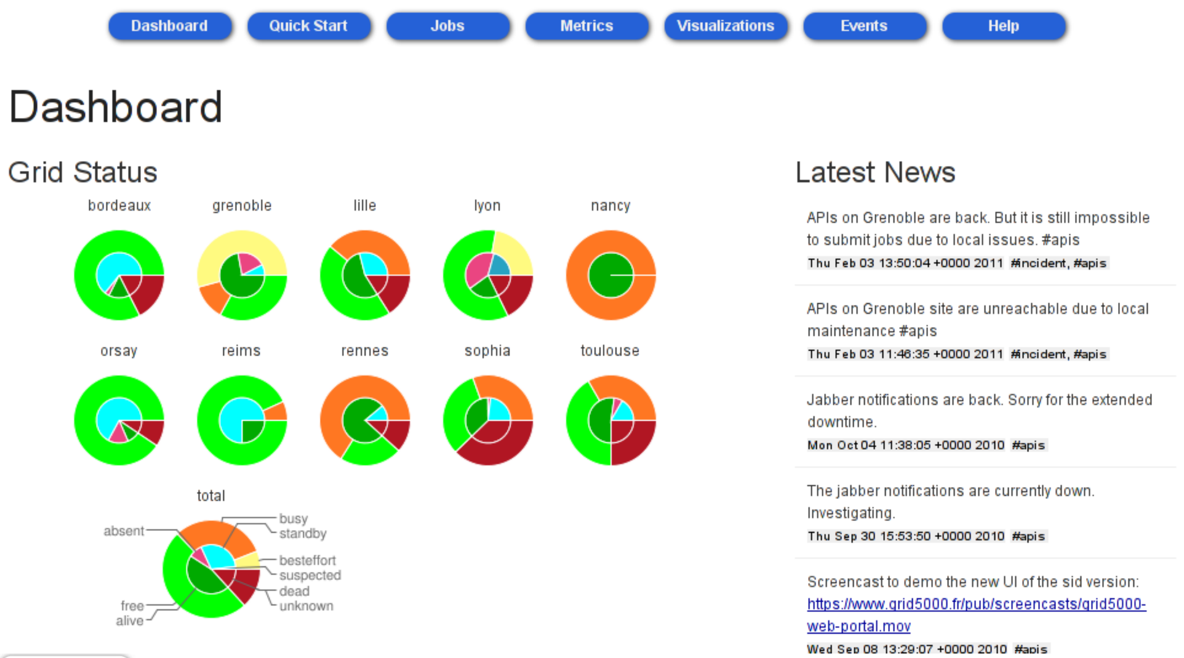
Putting it all together: GRID’5000 API

- Individual services & command-line interfaces are painful

- **REST API for each Grid'5000 service**
  - **Reference API**: versioned description of Grid'5000 resources
  - **Monitoring API**: state of Grid'5000 resources
  - **Metrology API**: Ganglia data
  - **Jobs API**: OAR interface
  - **Deployments API**: Kadeploy interface
  - …

Putting it all together: GRID’5000 API, cont

Also some nice Web interfaces on `https://api.grid5000.fr/`



Grid'5000 Cheat Sheet

Text between double brackets are wiki pages.
See https://www.grid5000.fr/

For events and maintenance on platform
See https://www.grid5000.fr/status

Ssh [[External_Access]]

```
# ~/.ssh/config
Host g5k
  Hostname access.lille.grid5000.fr
  User g5k_login
  IdentityFile ~/.ssh/id_dsa

Host *.g5k
  User g5k_login
  ProxyCommand ssh g5k "nc -q 0 `basename %h .g5k` %p"
```

Use it

```
ssh nancy.g5k
ssh edel-1.grenoble.g5k
scp ~/foo rennes.g5k:bar/
```

Text in color MUST to be substitued by appropriate values

Oar Cluster [[Cluster_experiment]]

Jobs states

```
oarstat
oarstat -f -j JOB_ID
oarstat -u G5K_LOGIN
```

Nodes states

```
oarnodes
oarnodes --sql "cpucore='4'"
```

Submission : Interactive

```
oarsub -I
  env | grep OAR
  cat $OAR_NODE_FILE
```

Reserve IPs

```
oarsub -I -l slash_22=1
g5k-subnets
```

20 nodes on griffon during 2h with 20G ib cards

```
oarsub -I -l nodes=20,walltime=2 \
  -p "cluster='griffon'" -p "ib20G='YES'"
```

Submission : Passive

```
oarsub ~/my-script
```

5 nodes during 2h with 10G ib cards

```
oarsub -l nodes=5,walltime=2 -p "ib10G='YES'"~/prog
  --> cat OAR.OAR_JOB_ID.std{err,out}
```

Connection to a running job

```
oarsub -C OAR_JOB_ID
```

on a node in your reservation

```
oarsh node.fqdn
```

Submission : Reservation (passive mode)

```
oarsub -r '2011-05-16 14:20:00' \
  -l nodes=10,walltime=0:10:00 ~/my-script
```

Reservation with deploy type (interactive mode)

```
oarsub -t deploy -r '2011-05-16 14:30:00' \
  -l nodes=5,walltime=2 -p "ib10G='YES'" -n "Prog42"
```

Delete a reservation

```
oardel OAR_JOB_ID
```

API [[API_Main_Pratical]] [[API]]

API Sid
- https://api.grid5000.fr/sid/ui/index.html

Grid'5000 Nodes API
- https://api.grid5000.fr/2.0/ui/nodes.html

Sync data [[Syncing_data]]

```
rsync --dry-run --delete -avz ~/synced site.grid5000.fr

for site in bordeaux lyon toulouse; do
  rsync --delete -avz ~/synced ${site}.grid5000.fr:~;
done
```

Open for comments :: support-staff@lists.grid5000.fr

Oar Grid [[Grid_experiment]]

Discovering resources

```
disco cluster_name
disco site1 site2
```

Jobs Grid stats

```
oargridstat
oargridstat GRID_JOB_ID
```

Submission : Interactive

```
oargridsub -t allow_classic_ssh \
  -w '0:20:00'CLUSTER1:rdef="/nodes=2",CLUSTER2:rdef="/nodes=3"
```

Create a node file

```
oargridstat -w -l GRID_JOB_ID | sed '/^$/d' > ~/nodes
```

Distribute node file

```
OAR_JOB_ID=CLUSTER_JOB_ID oarcp -i\
  /tmp/oargrid/oargrid_ssh_key_LOGIN_GRID_JOB_ID~/machines \
  `head -n 1 machines`:
```

Connect on first node

```
OAR_JOB_ID=CLUSTER_JOB_ID oarsh -i \
  /tmp/oargrid/oargrid_ssh_key_LOGIN_GRID_JOB_ID ` head -n 1 machines`
```

Ending

```
oargriddel  GRID_JOB_ID
```

Submission : Reservation (passive mode)

```
oargridsub -t allow_classic_ssh CLUSTER1:rdef="/nodes=1",\
  CLUSTER2:rdef="/nodes=4" -s '2011-05-16 14:20:00'\
  -w '0:10:00' -p /prog42/helloworld
```

View results

```
tail -f OAR.CLUSTER_JOB_ID.std{err,out}
```

Hardware Overview [[Special:G5KHardware]]

	#nodes	cpu (Intel | Amd)	memory	disk	GPU	network
Bordeaux						
Bordeplage	51	2x1cores @3.0Ghz	2GB	61GB	-	ib10g ddr
Bordereau	93	2x2cores @2.6Ghz	4GB	69GB	-	-
Borderline	10	4x2cores @2.6Ghz	32GB	520GB	-	{mx,ib}10g
Grenoble						
Adonis	12	2x4cores @2.26Ghz	24GB	217GB	C1070	ib40g qdr
Edel	72	2x4cores @2.27Ghz	24GB	52GB	-	ib40g qdr
Genepi*	34	2x4cores @2.5Ghz	8GB	139GB	-	ib20g ddr
Lille						
Chicon	26	2x2cores @2.6Ghz	4GB	69GB	-	mx10g
Chimint	20	2x4cores @2.4Ghz	16GB	260GB	-	-
Chinqchint	46	2x4cores @2.83Ghz	8GB	217GB	-	mx10g
Chirloutte	8	2x4cores @2.4Ghz	8GB	260GB	M2050	-
Lyon						
Capricorne*	56	2x1core @2.0Ghz	2GB	69GB	-	mx10g
Sagittaire*	79	2x1core @2.4Ghz	2GB	63GB	-	-
Nancy						
Griffon	92	2x4cores @2.5Ghz	16GB	278GB	-	ib20g ddr
Graphene	144	1x4cores @2.6Ghz	16GB	278GB	-	ib20g ddr
Orsay						
Gdx	310	2x1core @2.0, 2.4Ghz	2GB	69GB	-	mx10g
Netgdx	30	2x1core @2.0Ghz	2GB	69GB	-	-
Reims						
Stremi	44	2x12cores @1.7Ghz	48GB	232GB	-	-
Rennes						
Paradent	64	2x4cores @2.5Ghz	32GB	139GB	-	-
Paramount	33	2x2cores @2.33Ghz	8GB	520GB	-	mx10g
Parapide	25	2x4cores @2.93Ghz	24GB	434GB	-	ib20g ddr
Parapluie	40	2x12cores @1.7Ghz	48GB	232GB	-	ib20g ddr
Sophia						
Helios	56	2x2cores @2.2Ghz	4GB	63GB	-	mx10g
Sol	50	2x2cores @2.6Ghz	4GB	217GB	-	-
Suno	45	2x4cores @2.26Ghz	32GB	519GB	-	-
Toulouse						
Pastel	80	2x2cores @2.61Ghz	8GB	217GB	-	-
Violette*	52	2x1core @2.19Ghz	2GB	63GB	-	-

Deploy [[Deploy_environment-OAR2]]

Locate a suitable image

```
kaenv3 -l
kaenv3 -l -u LOGIN
kaenv3 -p squeeze-x64-base -u deploy
```

Use deploy type for your job

```
oarsub -I -t deploy -l nodes=2
  cat $OAR_NODE_FILE
```

Deploy an environment

```
kadeploy3 -e squeeze-x64-base -m node.site.grid5000.fr
kadeploy3 -e squeeze-x64-base -f $OAR_NODE_FILE
```

with your ssh key *(for a connection without password)*

```
kadeploy3 -e lenny-x64-base -f $OAR_NODE_FILE -k ~/.ssh/key.pub
kadeploy3 -e squeeze-x64-min -f $OAR_NODE_FILE -k
```

Save your deployed environment with tgz-g5k
(available on gforge, or installed on environments)
`tgz-g5k login@frontend:image.tgz` (from node)
`ssh root@node tgz-g5k > image.tgz` (from frontend)

Connection to the deployed environment

```
ssh root@node.site.grid5000.fr (password "grid5000")
```

with console *(useful if network doesn't work)*

```
kaconsole -m node.site.grid5000.fr
```

Deploy and save your environment

Generate a desciption file

```
kaenv3 -p squeeze-x64-base -u deploy > image.env
```

(edit file image.env to update with your values)

Deploy

```
kadeploy3 -f $OAR_NODE_FILE -a image.env
```

Save your image

```
kaenv3 -a image.env
```

Multi-sites deployment

```
kadeploy3 -e squeeze-x64-base -f ~/grid_nodes\
  --multi-server -k
```

Easy use with public share

```
kadeploy3 -f $OAR_NODE_FILE\
  -f http://public.nancy.grid5000.fr/~login/image.env -k
```

Links https://www.grid5000.fr/

DrawGantt (Nodes states in a temporal diagram)
https://helpdesk.grid5000.fr/oar/Site/drawgantt.cgi
Monika (Nodes states with properties)
https://helpdesk.grid5000.fr/oar/Site/monika.cgi
Ganglia (Nodes metrics)
https://helpdesk.grid5000.fr/ganglia/
Grid'5000 API
https://api.grid5000.fr/
UMS (Account, quotas extensions)
https://api.grid5000.fr/ui/account
Grid'5000 Software
[Grid5000:Software] on wiki.
DrawGanttGlobal
https://www.grid5000.fr/gridstatus/oargridgantt.cgi
MonikaGlobal
https://www.grid5000.fr/gridstatus/oargridmonika.cgi
Public share access from outside g5k (with http auth)
https://api.grid5000.fr/sid/grid5000/sites/site/public/login/
Public share access from inside g5k
https://public.site.grid5000.fr/~login/
Public share (populate your own public share)
drop files in your ~/public/ folder (see README in there)

* With electrical consumption.
See https://helpdesk.grid5000.fr/supervision/lyon/wattmetre/

#3679 -- version 0.8

GRID'5000 and Virtualization

Supporting virtualization experiments

- System images
  - Pre-built images maintained by the technical staff
  - Xen 3.x, KVM
- Network
  - Need reservation scheme for both IP and MAC addresses
  - Mac addresses are now randomly assigned
  - Sub-net range ccan be booked for Ips (/18, /19, …)

  - AAAAAAA FINIR

Industrial Relations

Alcatel-Lucent Bell Labs

- Traffic aware routers

Orange Labs

- Data placement algorithms on P2P architectures

Microsoft Resarch-INRIA

- Microsoft Azure: A-Brain (AzureBrain), « cloud » testbed for experimenting storage technologies (Kerdata)

EDF R&D (Myriads, GRAAL)

BULL (GRAAL, Runtime)

- Application mapping

IBM

- BlueWaters, Clouds

Startup companies

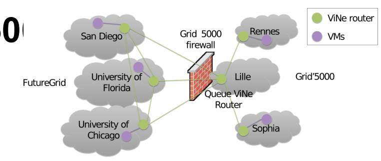
Three startups companies started by Grid'5000 researchers

- **LYaTiss** (LIP, ENS Lyon) around virtualization et network QoS
- **SysFera** (LIP, ENS Lyon) around large scale computing over Grids and Clouds
- **Activeon** (INRIA Sophia) around distributed computing

GRID'5000 EXPERIMENTS

Recent results in several fields

- **Cloud: Sky computing on FutureGrid and Grid'50**
  - 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 ➔ 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)

List of Open Challenges

Network

- Traffic Awareness

System

- Energy Profiling of Large Scale Applications
- Robustness of Large Systems in Presence of High Churn
- Orchestrating Experiments on the gLite Production Grid Middleware

Programming Paradigm

- Large Scale Computing for Combinatorial Optimization Problems
- Scalable Distributed Processing Using the MapReduce Paradigm

Domain Specific

- Multi-parametric Intensive Stochastic Simulations for Hydrogeology
- Thinking GRID for Electromagnetic Simulation of Oversized Structures

Traffic Awareness

Context

- Common Labs INRIA & Alcatel Bell Labs
- Design of traffic aware routers for high-speed networks

Objective

- Identify application classes from the behavioral (semantic) analysis of corresponding traffic
  - How does traffic behavior relate to flows semantic?
  - Which traffic characteristics are capturable on high speed networks?
  - Which constraints to get meaningful characteristics on-line?

Difficulties / Pitfalls

- Initial program hampered by
  - Difficulty to obtain (download or simulate) traffic traces characteristic of different applications
  - Semi-supervised learning (as primarily thought) does not seem to over-perform traditional decision tree algorithms

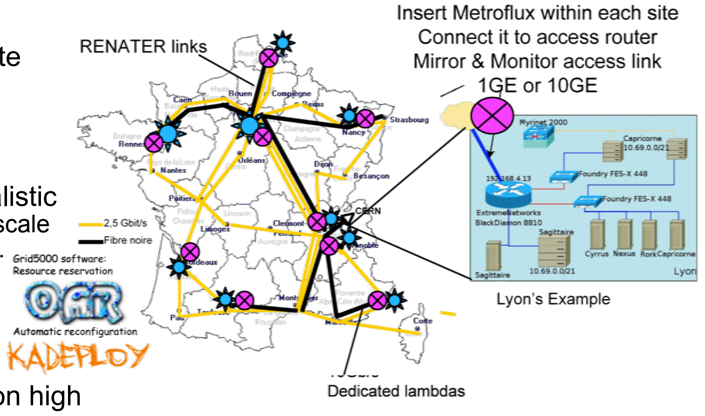
Traffic Awareness & Grid5000

How do we use Grid'5000?

- As a controllable testbed to emulate large-scale, high speed networks

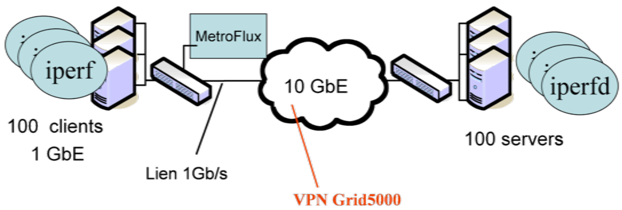
Why do we use Grid'5000?

- To reproduce the conditions of realistic environments …: Congestion, multi-scale aggregations, large size, heterogeneity.
- that can alter the flows' semantic



Technological advances

- MetroFlux: Packet capture probe on high speed links and under controlled situations
- Virtualization: Deployment of a physical infrastructure (open flow routers, switches) to emulate a virtual sub-network
- Trans-national link: Construction, through Grid'5000, of a 1Gbps dedicated link between France and Japan



Energy Profiling of Large Scale Applications (Energy)

Issues

- Reduce energy consumption of large-scale infrastructure
- Management of physical resources & virtualized resources

Objective

- Handle energy efficiency aspects of large scale applications deployed on multiple sites

Roadmap

- Model (complex) energy consumptions of systems and applications
  - Need to profile applications
- Develop software to log, store and expose energy usage
  - Make use of the G5K energy sensing infrastructure
- Experiments on large scale and heterogeneous infrastructure

How to Decrease Energy Consumption without Impacting Performance?

How to monitor and to analyze the usage and energy consumption of large scale platforms?

How to apply energy leverages (large scale coordinated shutdown/ slowdown)?

How to design energy aware software frameworks?

How to help users to express theirs Green concerns and to express tradeoffs between performance and energy efficiency?

Energy: Challenges

Exploring energy aspects at large scale

Two focus

- Applications deployed on real physical resources
- Applications and services deployed on virtualized resources

Providing feedback on large scale applications
Extending the Green Grid5000 infrastructure
Analyzing energy usage of large scale applications per components
Designing energy proportional frameworks (computing, memory or network usage)

Robustness of Large Systems in Presence of High Churn (P2P-Ch)

Issues

- Large scale distributed, heterogeneous platforms
  - 10K-100K nodes
- Frequency of connections/disconnections (churn)

Objective

- Maintain the platform connectivity in presence of high churn

Roadmap

- Develop a formal model to characterize the dynamics
  - Failure Trace Archive – http://fta.inria.fr
- Design algorithms for basic blocks of distributed systems on a churn-resilient overlay
- Experiments these algorithms on G5K

Robustness of Large Systems in Presence of High Churn (P2P-Ch)

Distributed algorithms for dynamic systems

- Variable number of peer, dynamic topology, mobility

Two approaches

- Determinist
  - Consensus, mutual exclusion (1 internship Regal)
- Probabilistic
  - High volatility, partitioning management

Integrate models / traces in fault injection tools

- FCI-FAIL – (Orsay)

Large scale experiments on Grid'5000

Orchestrating Experiments on the gLite Production Grid Middleware (Orchestration)

Issues

- Production Grid Middleware

Objective

- Explore the use of the Grid'5000 testbed as a test environment for production grid software such as gLite and other related services

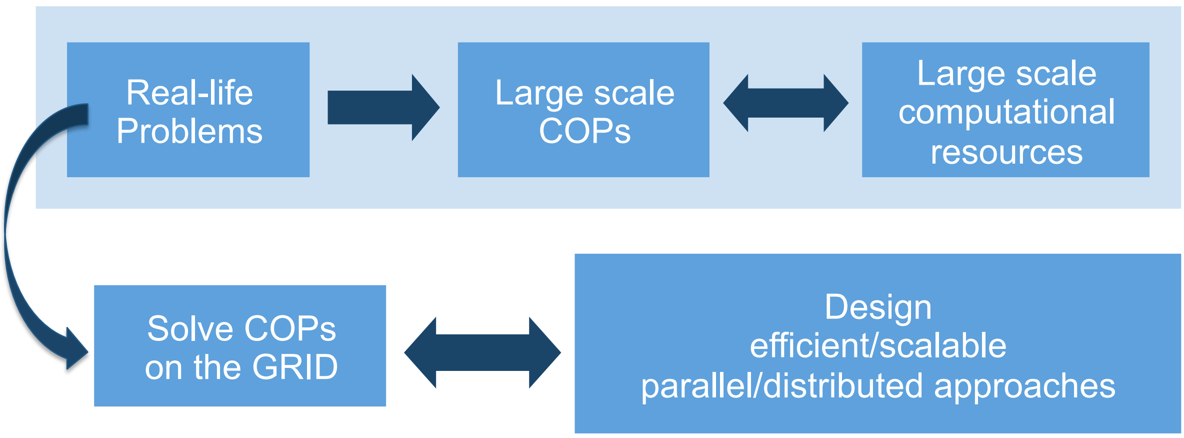
Roadmap

- Define a detailed procedure to deploy the gLite middleware on Grid'5000
- Define reusable services: Control of a large number of nodes, data management, experimental condition emulations, load and fault injection, instrumentation and monitoring, etc.
- Develop experiment orchestration middleware
- Perform large-scale experiments involving the gLite middleware and applications from production grids

Large Scale Computing for Combinatorial Optimization Problems (COPs)

Objectives

- Solve optimally large scale Combinatorial Optimization Problems (COPs) using huge amount of computational resources



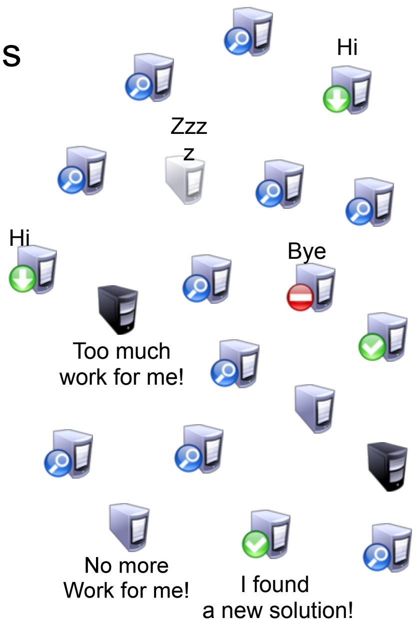
Large Scale Computing for Combinatorial Optimization Problems (COPs)

Goals at the application level

- Solve optimally previously unsolved COPs
- New specific COPs approaches

Goals at the algorithmic level

- How to gain in scalability?
  - Pure peer-to-peer approaches
  - Fully distributed algorithms
- How to address latencies/resources volatility?
  - Fault-tolerant/dynamic algorithms
  - Redundancy vs efficiency



Large Scale Computing for Combinatorial Optimization Problems (COPs)

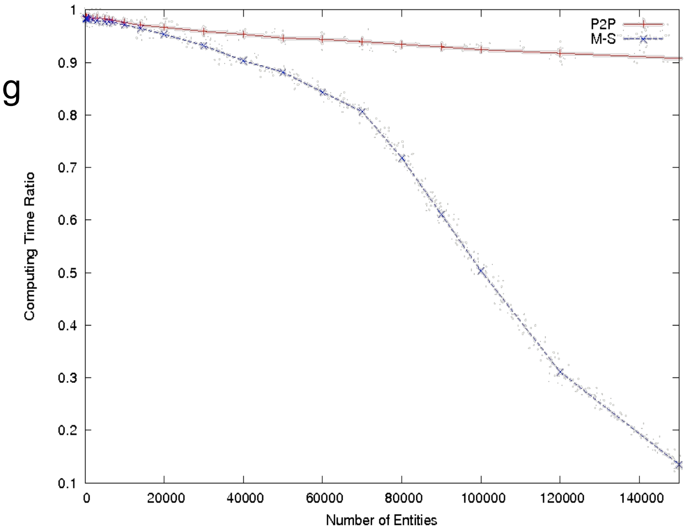
How GRID5000 can help?

- At the application level (make it a success story)
  - Effectively find unknown and optimal COPs solutions
- At the algorithmic level (make it smart)
  - Experiments/simulations are mandatory to validate our algorithms
  - Measure the scalability / efficiency / congestion / fault-tolerance robustness of our approach

COPS: First Results

P2P Branch&Bound

- Fully distributed
  - Work sharing / Load balancing
  - Termination detection
  - Network congestion (messages)
- Topology independent
- Validated using a Pastry-like overlay and up to 150,000 processes



4th SCALE Challenge Finalist (CCGRID 2011)

COPs: Next Challenging Issues

Extensions to a dynamic, volatile and fully distributed environment

- Maintain overlay connectivity distributely
- Efficient fault-tolerant distributed algorithms

Study the impact of network heterogeneity

Study the proposed distributed protocol under some formal model capturing the dynamicity of the network

- Related to high churn challenge

Study the scalability of the proposed dynamic approach

- Large scale experimentations, simulations, emulation

Scalable Distributed Processing Using the MapReduce Paradigm

Issues

- Distributed data-intensive applications (Peta-bytes)
- Data storage layer
  - Efficient, fine-grain, high throughput accesses to huge files
  - Heavy concurrent access to the same file (R/W)
  - Data location awareness
  - Volatility

Objective

- Ultra-scalable MapReduce-based data processing on various physical platform (clouds, grids & desktop computing)

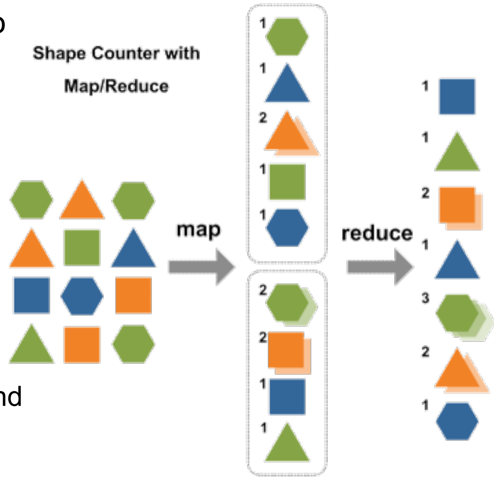
Roadmap

- Advanced data & meta-data management techniques
- MapReduce on desktop grid platforms
- Scheduling issues
  - Data & computation, heterogeneity, replication, etc.

Scalable Distributed Processing Using the MapReduce Paradigm

ANR Project Map-Reduce (2010-2014) associated to the MapReduce HEMERA Challenge

- Partners
  - INRIA (KerData, GRAAL), Argonne National Lab, UIUC, JLPC, IBM, IBCP, MEDIT
- Goal
  - High-performance map-reduce processing through concurrency-optimized data processing
- *http://mapreduce.inria.fr*
- An objective of the project
  - Use BlobSeer as back-end storage for VM images and cloud application data



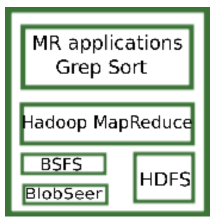
Experiments done on Grid'5000

- Up to 300 nodes
- Plans: joint deployment G5K+FutureGrid (USA)

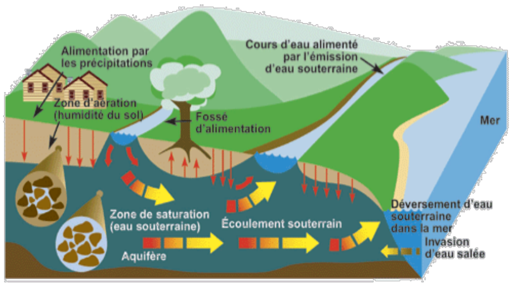
Results to be transferred on real clouds

- Nimbus (ANL): ANR MapReduce project
- Microsoft Azure: A-Brain project (MSR-INRIA)

First results : HPDC 2011



Multi-Parametric Intensive Stochastic Simulations for Hydrogeology (Hydro)



Issues

- Groundwater resource management & remediation
- Limited knowledge
  - Highly heterogeneous and fractured geological formations
- Numerical models
  - Probabilistic data + uncertainty quantification methods
    - Stochastic framework (multiple simulations)
    - Various physical parameters
- Large size geological domain to discretize

Objective

- Efficient execution of multi-parametric heavy computation simulations

Roadmap

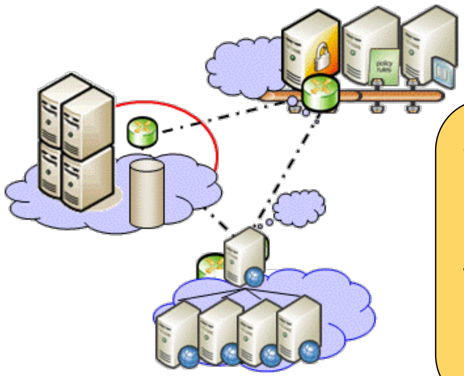
- Study how to program, deploy & schedule the application
- Validate the approach for increasing level of parallelism for 2D problems then 3D problems

BonFIRE

BONFIRE

BonFIRE data sheet

Type of project: Integrated Project
Project coordinator: ATOS
Project start date: 1st June 2010
Duration: 42 months
EC contribution: **7.2**M€ (orig 6.7 M€)
(1.34 M€ for 2 open calls)



The **BonFIRE** (*Building service testbeds for Future Internet Research and Experimentation*) project is designing, building and operating a multi-site cloud facility to support research across applications, services and systems targeting services research community on Future Internet.

Facility for services experimentation

6 sites
- 4 sites running a customized OpenNebula stack
- 1 site running a customized Emulab instance (Virtual Wall, IBBT)
- 1 site running HP Cells

Real and emulated networks
- Emulab-based Virtual Wall
- Controlled networks on the way (GEANT AutoBAHN and FEDERICA)

Experiment Descriptors
- Portal – use point and click to run an experiment
- “Restfully” – describe the experiment programmatically
- JSON DSL (OVF on the way) – describe the experiment statically

Advanced monitoring
- Zabbix on all VMs
- Infrastructure monitoring (understand what is happening on the machines hosting your VMs)

Experiment at scale using on-request resources

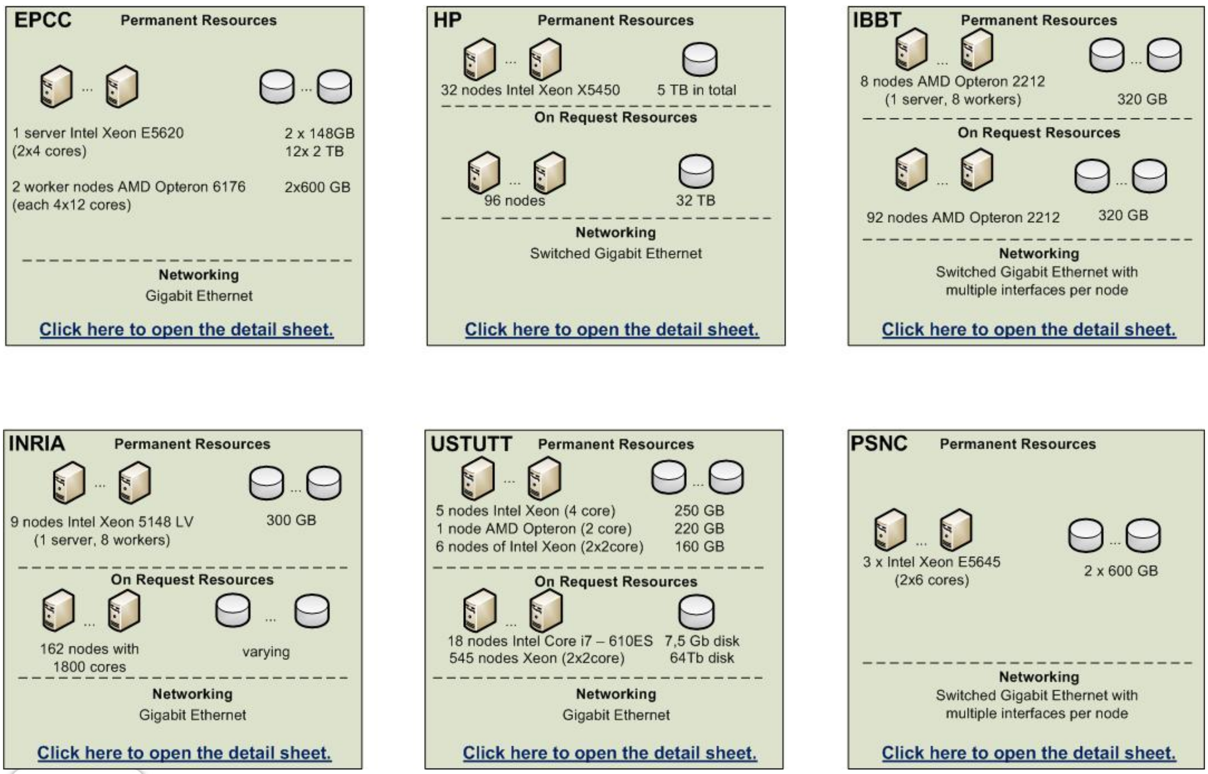
Sites operate a permanent testbed

The fr-inria site can be extended on request over the Grid'5000 resources located in Rennes

- BonFIRE user reserves the resources (and gets exclusive access to the hardware)
  - Just another user for the Grid'5000 stack
- At the start of the reservation, Grid'5000 machines
  - get deployed as OpenNebula worker nodes
  - Get moved to the BonFIRE Vlan
  - Get added as a new cluster to the running OpenNebula frontend

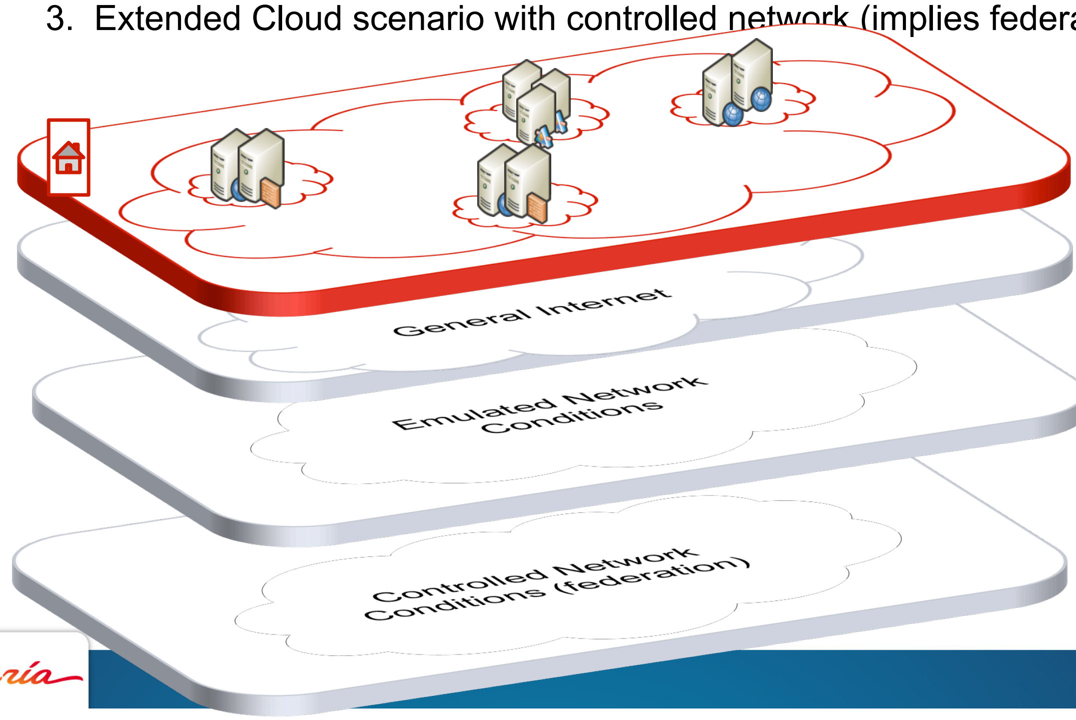
BonFIRE users get exclusive access to a 162 nodes/ 1800 core OpenNebula infrastructure (screencast at http://vimeo.com/39257324)

BonFIRE's Infrastructures and Resources

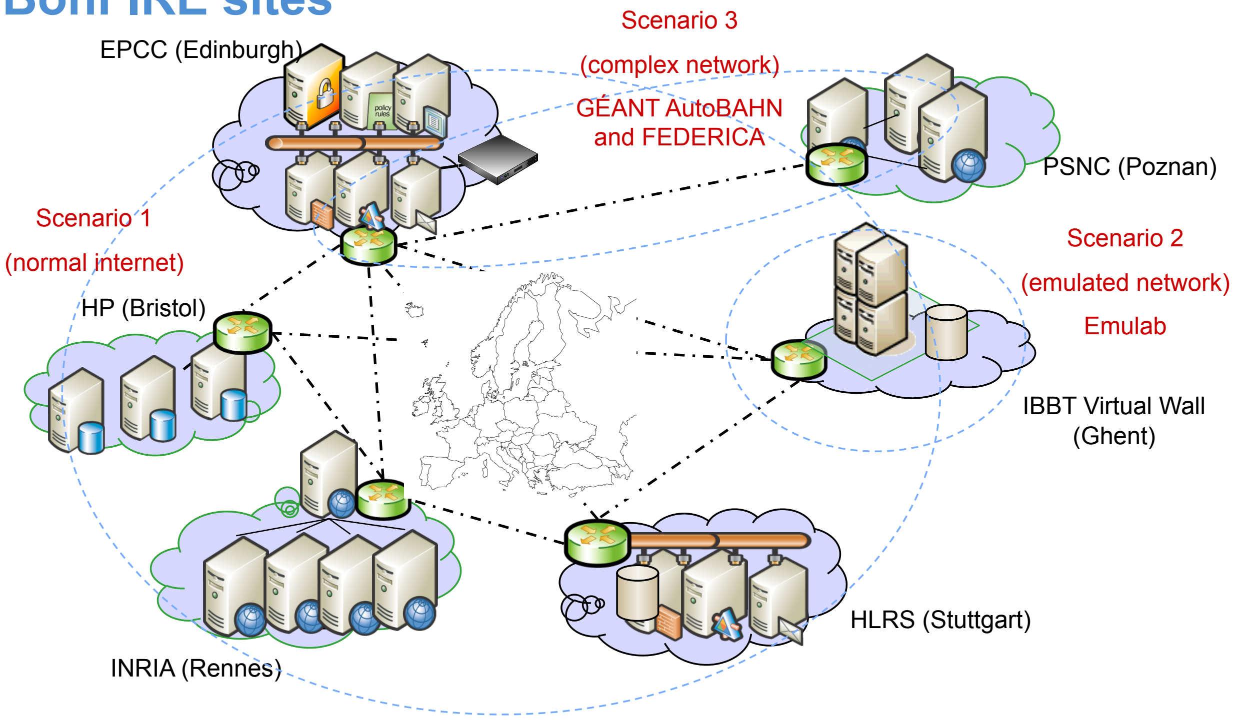


Three Scenarios – Service Experiments on top of three different Network Infrastructures

1. Extended multi-site clouds connected through standard internet
2. Cloud scenario with emulated network (IBBT's Virtual Wall based on Emulab)
3. Extended Cloud scenario with controlled network (implies federation)



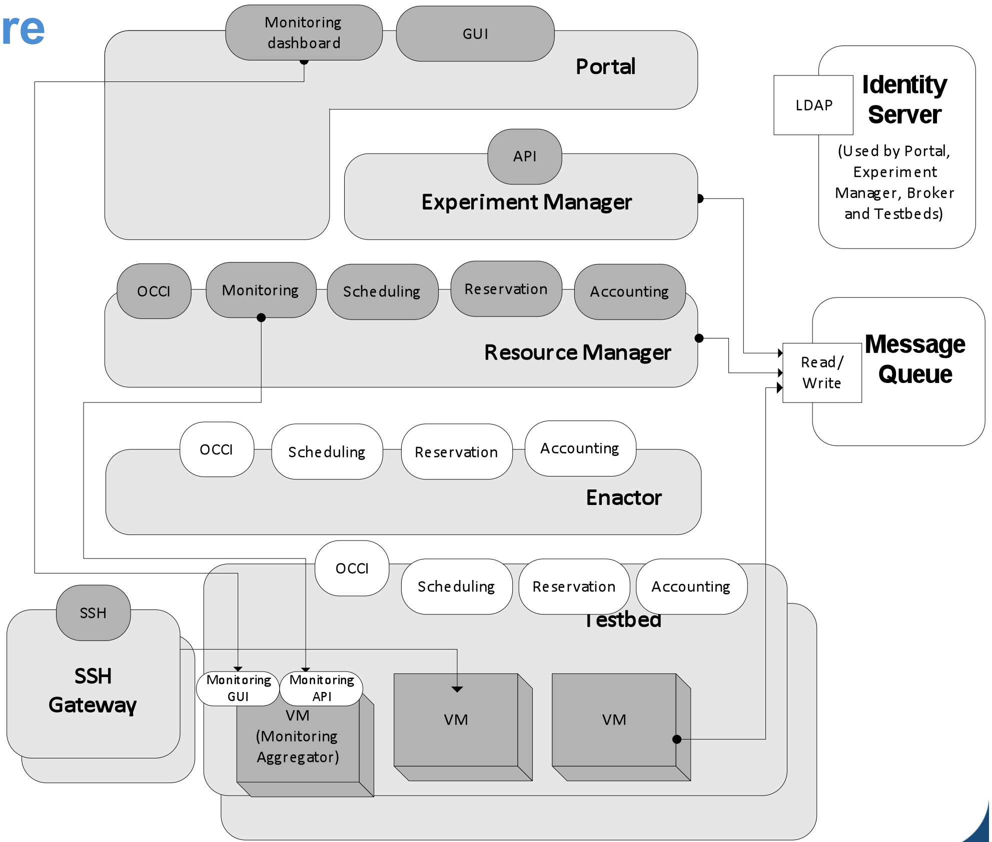
BonFIRE sites



Permanent (~350cores / 30TB) & On-Request (theoretically 3000+ cores) infrastructures

Note: network links indicative only

Architecture



BonFIRE Offering (1/2)

- Support experiments over multiple heterogeneous cloud testbeds using a single declarative experiment descriptor.
- Support geographically distributed experiments.
- Support experiment monitoring at both resource level (e.g. CPU usage, temperature, packet delay etc.) and application level.
- Support the deployment of different software stacks over a variety of differently configured resources (compute, storage, network etc.) in multiple heterogeneous cloud testbeds.
- Support elasticity within an experiment, i.e. dynamically create, update and destroy resources from a running node of the experiment, including. cross-testbed elasticity.

BonFIRE Offering (2/2)

- Support experiment management including experiment sharing, repeating and result collation and storage.
- Support the definition of an entire infrastructure in a single uniform experiment description.
- Study the possible federation of the BonFIRE testbeds with a variety of external cloud facilities, such as those provided by Federica or OpenCirrus.
- Support advanced network emulation via the Virtual Wall, including
  - Dynamic modifications of running experiments (at the moment the network topology and node images have to be fully configured at the start of the experiment.)
  - Additional generic network (e.g. overlay routing) and application layer functionality

RELATED PLATFORMS

Related Platforms

CONCLUSION

Conclusion and Open Challenges

- Computer-Science is also an experimental science
- There are different and complementary approaches for doing experiments in computer-science
- Computer-science is not at the same level than other sciences
- But, things are improving…

- GRiD'5000: a test-bed for experimentation on distributed systems with a unique combination of features
  - *Hardware-as-a-Service* cloud: redeployment of operating system on the bare hardware by users
  - Access to various technologies (CPUs, high performance networks, etc.)
  - Networking: dedicated backbone, monitoring, isolation
  - Programmable through an API

What Have We Learned?

Building such a platform was a real challenge !

- No on-the-shelf software available
- Need to have a team of highly motivated and highly trained engineers and researchers
- Strong help and deep understanding of involved institutions!

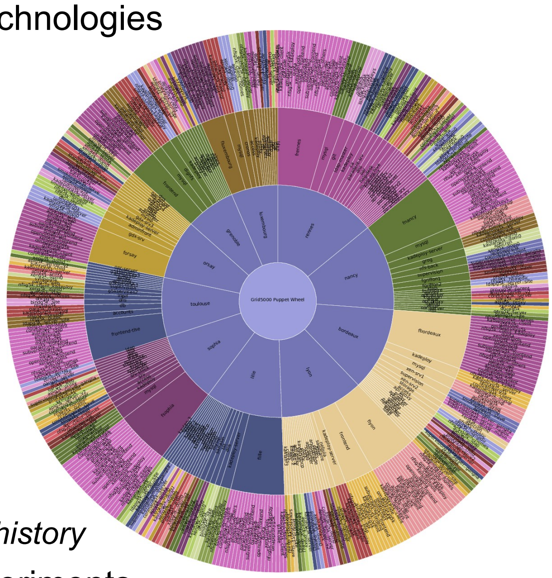
From our experience, experimental platforms should feature

- Experiment isolation
- Capability to reproduce experimental conditions
- Flexibility through high degree of reconfiguration
- The strong control of experiment preparation and running
- Precise measurement methodology

- Tools to help users prepare and run their experiments
- Deep on-line monitoring (essential to help observations understanding)
- Capability to inject real life (real time) experimental conditions
- (real Internet traffic)

Conclusion and Open Challenges, cont

- Testbeds optimize for experimental capabilities, not performance
- **Access** to the modern architectures / technologies
  - Not necessarily the fastest CPUs
  - But still expensive ➔ funding!
- Ability to **trust** results
  - Regular checks of testbed for bugs
- Ability to **understand** results
  - Documentation of the infrastructure
  - Instrumentation & monitoring tools
    - *network, energy consumption*
  - Evolution of the testbed
    - *maintenance logs, configuration history*
- Empower users to perform complex experiments
  - Facilitate access to advanced software tools



QUESTIONS ?

Special thanks to
E. Jeannot, A. Lèbre, D. Margery, L. Nussbaum, C. Perez, O. Richard

www.grid5000.fr

INVENTEURS DU MONDE NUMÉRIQUE

Frédéric DESPREZ
Frederic.Desprez@inria.fr

Software Validated on Grid'5000 (1/2)

- **CONFIIT,** Computation Over Network with Finite number of Independent and Irregular Tasks (Reims)
- **ParadisEO-G,** Parallel and Distributed Evolving Objects on top of Globus (Lille)
- **DeployWhere/FDF,** framework open source orienté composant pour le déploiement de logiciels distribués et hétérogènes (Lille)
- **Wrekavoc** (Nancy)
- **GridTPT**, plateforme de test distribuée pour prouveurs de formules (Nancy)
- **veriT**, solveur de formules SMT (Nancy)
- **GSOC**, Grid Security Operation Center (Besançon)
- **dPerf**, prédiction de performances des applications distribué en pair-à-pair (Besançon)
- **XtreemOS** (Rennes)
- **BlobSeer** (Rennes)
- Bibliothèque de mesures de la consommation électrique. Placement de tâches Energy-aware (Toulouse)

Software Validated on Grid'5000 (2/2)

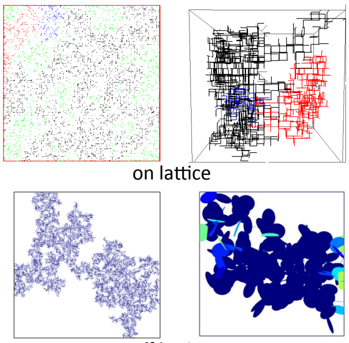
VMdeploy / Saline (Nantes)
KEntropy (Nantes)
Kargo (Nantes)
KaStore (Nantes)
kDFS (Nantes)
Metroflux (Lyon)
ANPI (Lyon)
OVNI5000 (Lyon)
SHOWATTS (Lyon)
MPI5000 (Lyon)
Green Grid5000 (Lyon)
ULCMi (Lyon)
HLCMi (Lyon)

DHICO (Lyon)
DIET (Lyon)
Grudu (Lyon)
P2P-MPI (Strasbourg)
MOTEUR workflow manager (Nice)

Influence of Fracture Network Complexity on Upscaling Hydrodynamic Laws

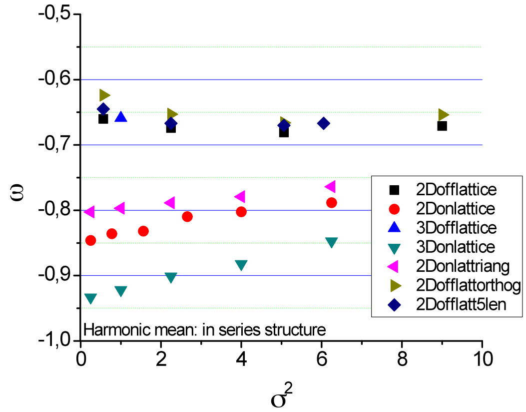
Objective: establish references results for more realistic fracture networks

STRUCTURES
Percolation fractals

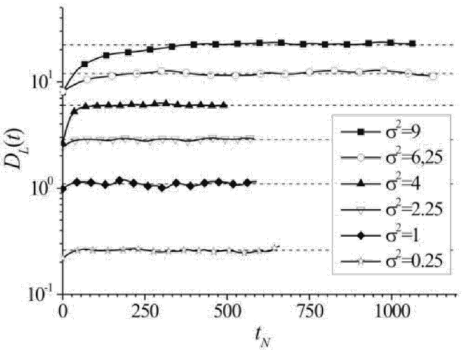


Local permeability distribution:
LN(μ,σ^2)

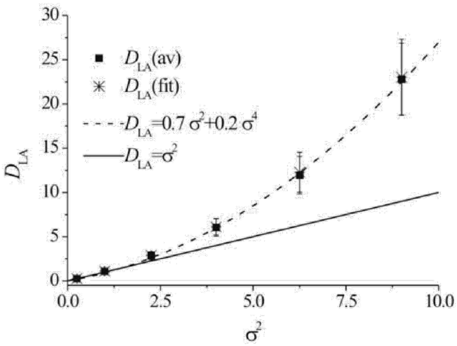
Global permeability
K(macro)=μ exp($\omega\sigma^2$/2)



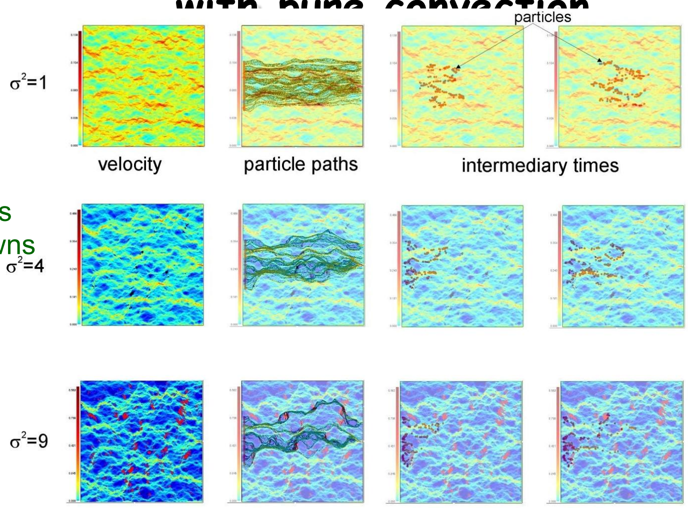
10^5 simulations of ~10^5-10^7 nodes
strong limitations for 3D off lattice networks



Each curve represents 100 simulations
on domains with 67.1 millions of unknowns

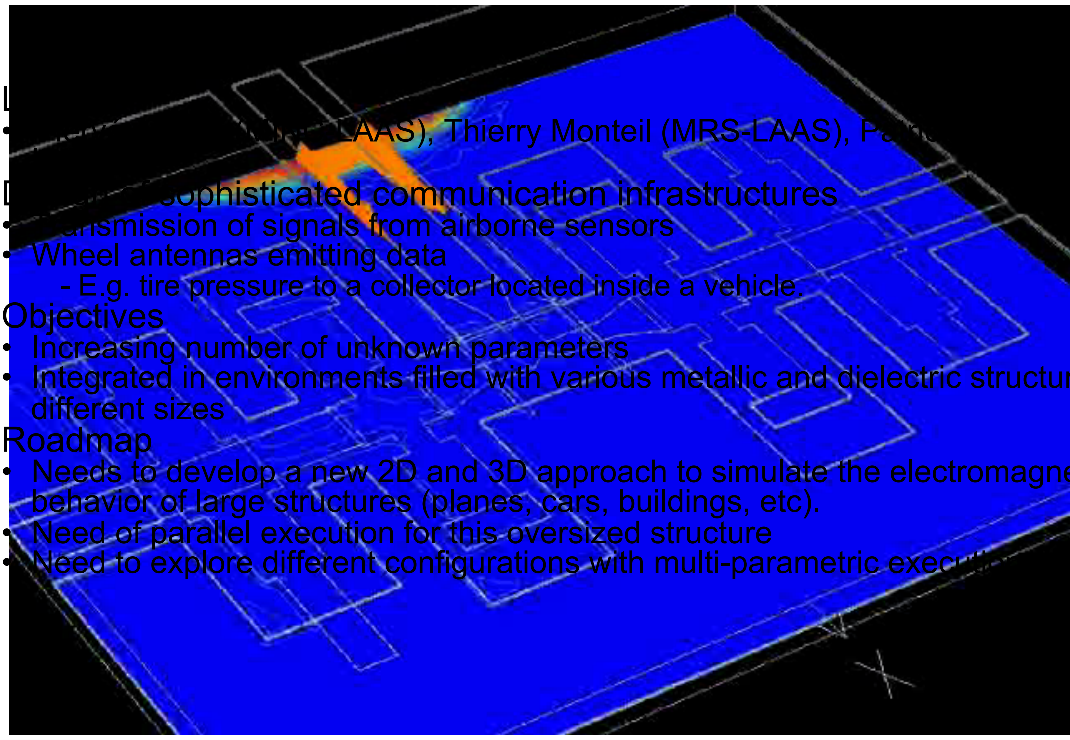


Longitudinal macro dispersion with pure convection



First numerical results obtained for such a large sigma!
From 2D to 3D

Thinking GRID for Electromagnetic Simulation of Oversized Structures (Electro)



L
• ...LAAS), Thierry Monteil (MRS-LAAS), P... ...STRE-

D... sophisticated communication infrastructures
• ...ansmission of signals from airborne sensors
• Wheel antennas emitting data
  - E.g. tire pressure to a collector located inside a vehicle.

Objectives
• Increasing number of unknown parameters
• Integrated in environments filled with various metallic and dielectric structures of different sizes

Roadmap
• Needs to develop a new 2D and 3D approach to simulate the electromagnetic behavior of large structures (planes, cars, buildings, etc).
• Need of parallel execution for this oversized structure
• Need to explore different configurations with multi-parametric executi...

Thinking GRID for Electromagnetic Simulation of Oversized Structures (Electro)

Utilization of multithreading and MPI over grid

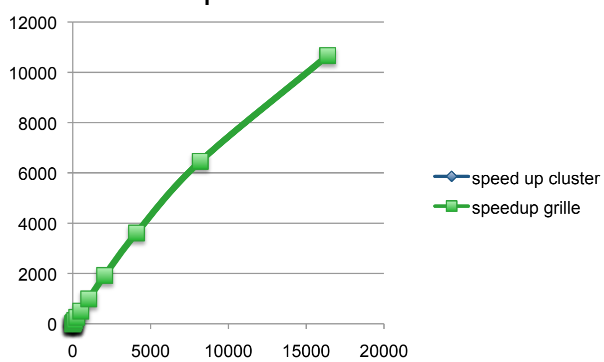
Collaboration between application, middleware and platform

Uses of autonomic policies:

- Breakdown or performance loss of a set of machines
- Automatic execution of new simulations in self adapting network set-ups
- Autonomic exploration of new solutions in multi-parametric mode

First theoretical estimation of speedup for oversized problem

- Challenge objectives
  - Autonomic deployment and reconfiguration on grid5000
  - Parallel algorithm for electromagnetic simulation
- Héméra objectives
  - Large scale experiment
  - Experiment and support for electromagnetic researchers



List of Working Groups

Transparent, Safe and Efficient Large Scale Computing
- Stéphane Genaud (ICPS), Fabrice Huet (OASIS)

Energy Efficient Large Scale Experimental Distributed Systems
- Laurent Lefèvre (RESO), Jean-Marc Menaud (ASCOLA)

Bring Grids Power to Internet-Users thanks to Virtualization Technologies
- Adrien Lèbre (ASCOLA), Yvon Jégou (MYRIADS)

Efficient exploitation of highly heterogeneous and hierarchical large-scale systems
- Olivier Beaumont (CEPAGE), Frédéric Vivien (GRAAL)

Efficient management of very large volumes of information for data-intensive applications
- Gabriel Antoniu (KERDATA), Jean-Marc Pierson (ASTRE)

Completing challenging experiments on Grid'5000
- Lucas Nussbaum (ALGORILLE), Olivier Richard (MESCAL)

Modeling Large Scale Systems and Validating their Simulators
- Martin Quinson (ALGORILLE), Arnaud Legrand (MESCAL)

Network metrology and traffic characterization

Transparent, Safe and Efficient Large Scale Computing

Leaders
- Stéphane Genaud (ICPS), Fabrice Huet (OASIS)

Scientific challenges
- Demonstrate which software architectural designs and programming models best match modern large-scale distributed systems

Grid'5000 allows to experimentally reproduce characteristics of such systems
- Network heterogeneity
  - High-latency WAN network links mixed with low-latency LAN
- Hierarchical architecture
- Virtualization of resources

Grid'5000 allows to test
- Programming Models
  - Combination of models ? New paradigms?
- Middleware
  - Which abstractions for runtime libraries or users?
- Complex Deployment

Energy Efficient Large Scale Experimental Distributed Systems

Leaders

- Laurent Lefèvre (RESO), Jean-Marc Menaud (ASCOLA)

Objective

- Energy aware software approaches able to reduce the energy consumption needed for high performance computing and networking operations in large scale distributed systems (datacenters, Grids and Clouds)

Working on three levels

- Hardware
- Infrastructure
- Application

Roadmap

- JTE «Aspects énergétiques du calcul» : 13/01/2011
  - Supported by Héméra
- JTE «Energie dans les centres de données» : Juin/2011
- SLA Energy / Cloud

Bring Grids Power to Internet-Users thanks to Virtualization Technologies

Leaders

- Adrien Lèbre (ASCOLA), Yvon Jégou (MYRIADS)

Context

- Job schedulers
- Exploit all VM capabilities

Objectives

- Cluster/Grid-Wide Context Switch
  - Manipulate vJobs (a job in VMs) instead of jobs
- From the Grid to the Desktop

Animation

- Wiki page (2009), mailing list, JTE, …

Efficient exploitation of highly heterogeneous and hierarchical large-scale systems

Leaders

- Olivier Beaumont (CEPAGE), Frédéric Vivien (GRAAL)

Potential research themes

- Mapping of data and computations
- (potentially with replication)
- Resource management
- Load-balancing
- Scheduling in probabilistic contexts
- (uncertainties, failures, etc.)
- Distributed scheduling
- Communication- and memory-aware scheduling
- Platform modeling (mainly, use of)

Efficient management of very large volumes of information for data-intensive applications

Leaders

- Gabriel Antoniu (KERDATA), Jean-Marc Pierson (ASTRE)

Objectives

- Explore research issues related to high-level services for information management
  - Search, mining, visualization, processing)
- For large volumes of distributed data
- Taking into account
  - Security, efficiency and heterogeneity
  - Applications requirements
  - Execution infrastructure (grids, clouds)

Issues

- Fault-tolerance, caching, transport, security (encryption, confidentiality), consistency, location transparency
- Interoperability among storage systems; Data indexing
- Data mining, data classification, data assimilation, knowledge extraction,

Completing Challenging Experiments on Grid'5000

Leaders

- Lucas Nussbaum (ALGORILLE), Olivier Richard (MESCAL)

Spin off the 'Orchestration' scientific challenge

Axis of work

- Methodology of the experimentation
  - Scenarios, experimental conditions, metrics, "cahier de laboratoire"
- Tools for the experimentation
  - Increasing the confidence in experimental results
    - DSL?

In conjunction with SimGrid

Modeling Large Scale Systems and Validating their Simulators

Leaders

- Martin Quinson (ALGORILLE), Arnaud Legrand (MESCAL)

Context

- Many studies rely on simulations
  - Easy to set up Reproducible Controlled Enable exploration
  - Fast Cheap Not disruptive
- Unfortunately models in most simulators are either simplistic, not assessed, or even plainly wrong.

Challenges

- Models need to be realistic, instantiable, and computationally tractable.

Outcome

- Better simulators with standard benchmark platforms
- Better understanding of resources, applications, and platform
- Interactions with other working groups regarding methodology (design of experiments, visualization, workload modeling, . . .)