Fault tolerance in Grid and Grid'5000

IFIP WG 10.4 on dependable Computing and Fault Tolerance

Franck Cappello
INRIA
Director of Grid'5000
fci@lri.fr

- Fault tolerance in Grid
- Grid'5000

Applications requiring Fault tolerance in Grid

Domains (grid applications connecting databases, supercomputers, instruments, visualization tools):

- Finance,
- Health care,
- eScience, Cyber Infrastructure (EGEE, Virtual observatory, TeraGrid, etc.)
- Nature and industrial disasters prevention and management
- etc.

Key technology:

- Web Services (with some extensions: WSRF)

The EGEE project (Enabling Grid for E-SciencE)

- Building and Maintaining a large scale computing infrastructure
- Provide support for Scientists using it.

Size:

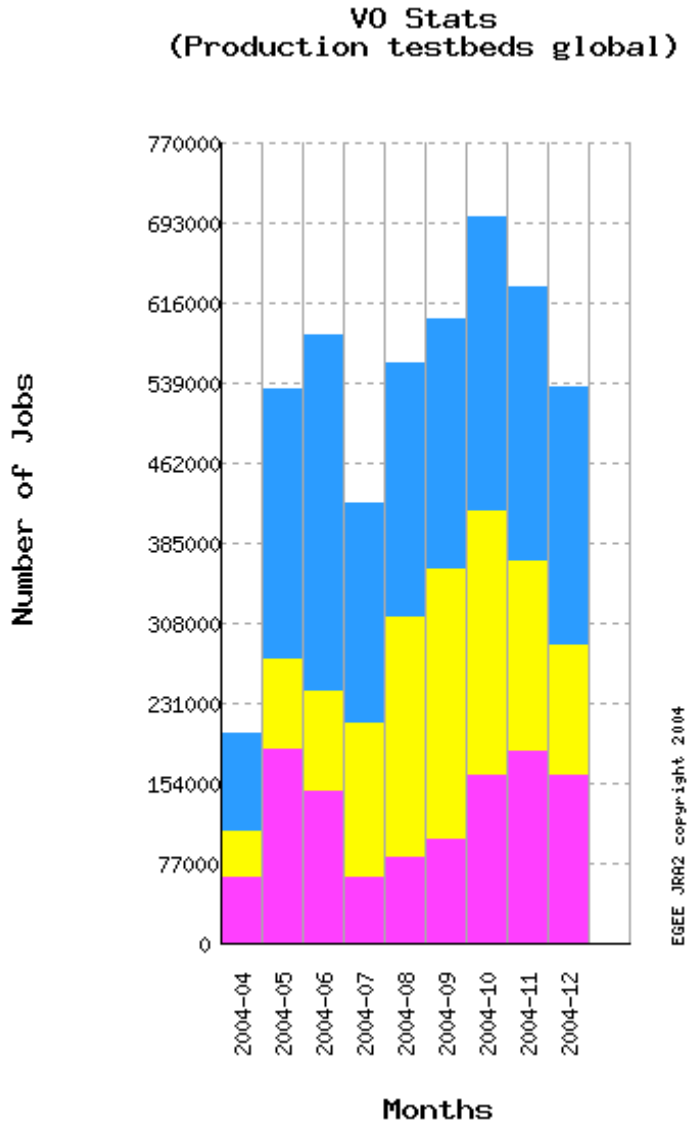
Users: 3000
Institutes: 70
Countries: 27
Sites: 148
CPU: > 13000
Disk > 98 PB

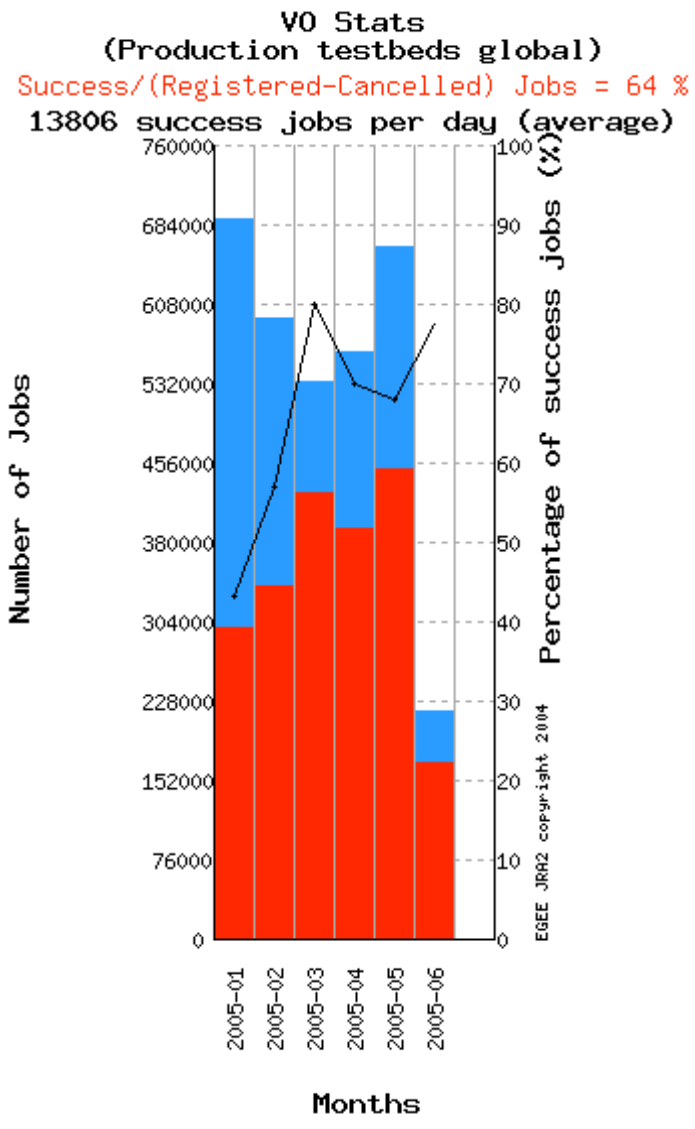
Duration: 2 years
Cost: 32M€
Next: EGEE2

Pilot applications:
LHC experiment (Alice, Atlas, CMS
→ Scale, high bandwidth data tra

Biomedical experiments:
→ Security, Ease of use, distributed data base

Job Statistics in EGEE (Enabling Grid for E-SciencE)





- Successful jobs are jobs whose execution is terminated (done jobs) AND whose done status is OK.
- Registered jobs are registered by the User Interface.
- Cancelled jobs are jobs indicated as cancelled by user.

EGEE issues and problems

- Hardware / Software issues
  - Heterogeneous hardware, software, OS are a BIG problems !
  - Example: User Interface
  - Example: floating point accuracy
  - Example: dynamic libraries
  - Example: distributed application across diffferent platforms
  - Revival of the interpreter, JIT ?
  - Security and accounting – IntraGrid vs. InterGrid
  - Submission times ???
- Political Issues
  - Different communities – different agendas / hidden agendas
  - coordination between partners
  - typical problems of large, heterogeneous organisations
  - small and dynamic vs. large and powerful organisations

Job Efficiency in EGEE

Execution time : ET = D3-D2 , Waiting Time :WT = D2-D1
Grid Efficiency : GE = ET/(ET+WT)

Overall

Month	Short jobs	Medium jobs	Long jobs	Infinite jobs
2005-01	EG= 0.62 % WT=54.05 min ET=0.34 min	EG= 30.06 % WT= 54.71 min ET= 23.52 min	EG= 54.88 % WT= 54.77 min ET= 66.61 min	EG= 78.81 % WT= 312.42 min ET= 1162.22 min
2005-02	EG= 0.69 % WT=65.71 min ET=0.45 min	EG= 5.43 % WT= 364.81 min ET= 20.96 min	EG= 38.96 % WT= 115.38 min ET= 73.63 min	EG= 60.25 % WT= 682.46 min ET= 1034.21 min
2005-03	EG= 3.89 % WT=18.72 min ET=0.76 min	EG= 19.47 % WT= 85.03 min ET= 20.56 min	EG= 41.14 % WT= 109.18 min ET= 76.30 min	EG= 77.38 % WT= 212.17 min ET= 725.83 min
2005-04	EG= 3.23 % WT=21.28 min ET=0.71 min	EG= 16.14 % WT= 111.94 min ET= 21.55 min	EG= 32.79 % WT= 154.33 min ET= 75.28 min	EG= 73.22 % WT= 263.64 min ET= 720.90 min
2005-05	EG= 0.72 % WT=62.89 min ET=0.46 min	EG= 7.17 % WT= 251.74 min ET= 19.44 min	EG= 22.64 % WT= 326.08 min ET= 95.45 min	EG= 75.79 % WT= 336.64 min ET= 1053.97 min
Average Results	EG= 1.39 % WT=41.46 min ET=0.58 min	EG= 10.85 % WT= 170.72 min ET= 20.78 min	EG= 28.24 % WT= 211.58 min ET= 83.28 min	EG= 71.56 % WT= 379.74 min ET= 955.28 min

Software Status in TERA GRID 1/2

TeraGrid:
-integrated, persistent computational resource.
-Deployment completed in September 2004,
-40 teraflops of computing power
-nearly 2 petabytes of storage,
-interconnections at 10-30 gigabits/sec. (via a dedicated national network.)

Summary of Common TeraGrid Software and Services 2.0

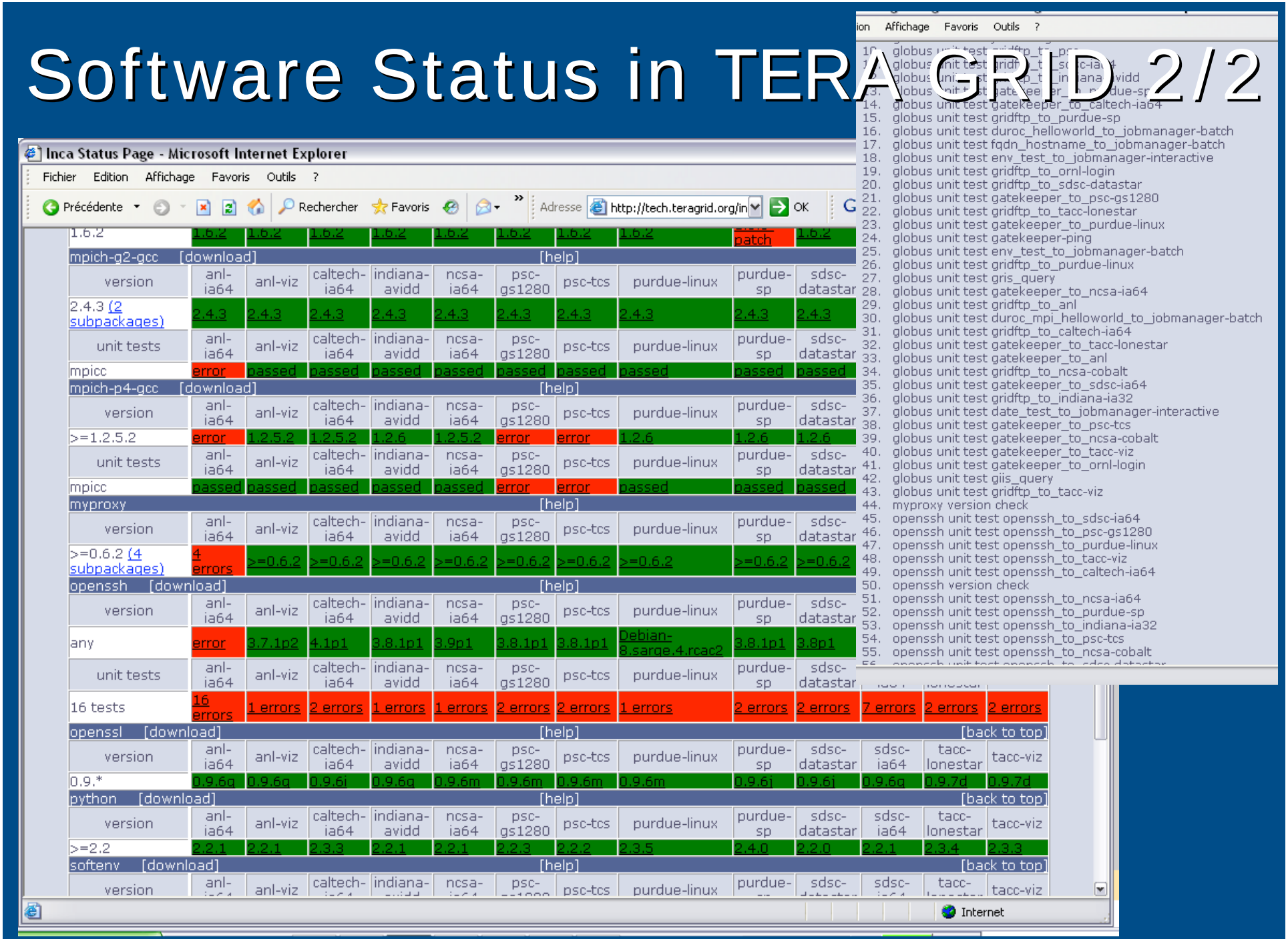
Page generated by Inca: 06/27/05 10:24 CDT

This page offers a summary of results for critical grid, development, and cluster tes
test results are available by clicking on the resource name in the "Site-Resource" c

Site-Resource	Grid	Development	Compute	Total Pass
anl-ia64	Pass: 7 Fail: 12 36% passed	Pass: 4 Fail: 5 44% passed	Pass: 2 Fail: 1 66% passed	Pass: 13 Fail: 18 41% passed
anl-viz	Pass: 14 Fail: 5 73% passed	Pass: 9 Fail: 0 100% passed	Pass: 3 Fail: 0 100% passed	Pass: 26 Fail: 5 83% passed
caltech-ia64	Pass: 13 Fail: 6 68% passed	Pass: 9 Fail: 0 100% passed	Pass: 3 Fail: 0 100% passed	Pass: 25 Fail: 6 80% passed
indiana-avidd	Pass: 18 Fail: 1 94% passed	Pass: 9 Fail: 0 100% passed	Pass: 3 Fail: 0 100% passed	Pass: 30 Fail: 1 96% passed
ncsa-ia64	Pass: 19 Fail: 0 100% passed	Pass: 9 Fail: 0 100% passed	Pass: 3 Fail: 0 100% passed	Pass: 31 Fail: 0 100% passed
psc-qs1280	Pass: 8 Fail: 11 42% passed	Pass: 7 Fail: 2 77% passed	n/a	Pass: 15 Fail: 13 53% passed
psc-tcs	Pass: 12 Fail: 7 63% passed	Pass: 8 Fail: 1 88% passed	n/a	Pass: 20 Fail: 8 71% passed
purdue-linux	Pass: 17 Fail: 2	Pass: 9 Fail: 0	Pass: 3 Fail: 0	Pass: 29 Fail: 2

http://tech.teragrid.org/inca-prod/cgi-bin//primaryhtmlmap.cgi?mapfile=/var/www/tech.teragrid.org/inca/TG/html/preload.state&topkey=exec

Software Status in TERAGRID 2/2



http://tech.teragrid.org/inca-prod/cgi-bin//primaryhtmlmap.cgi?mapfile=/var/www/tech.teragrid.org/inca/TG/html/preload.state&topkey=stack_compute

Why FT in Grid is difficult (1/2)

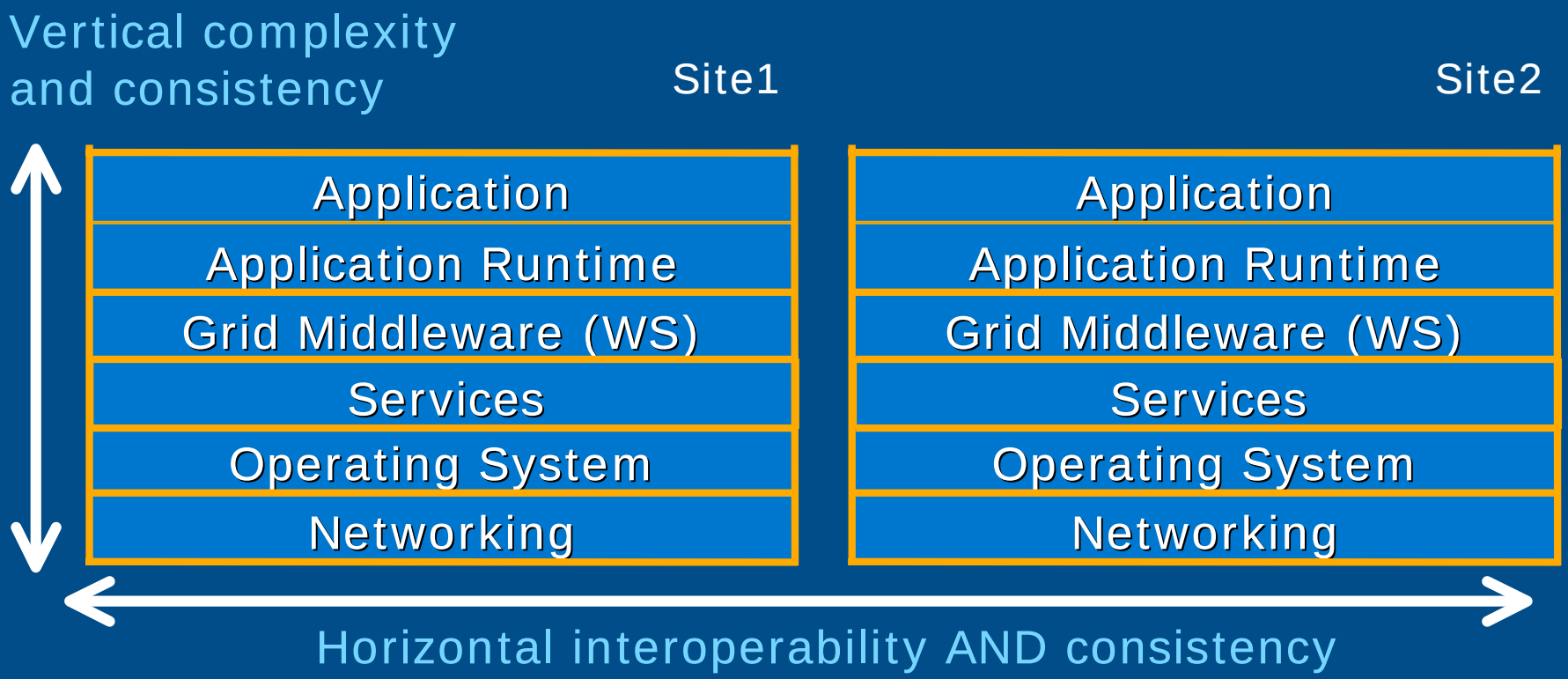
- Grids are installed, administered and controlled by humans
  -local priority may lead to stop or freeze jobs
  -modifications and updates take times and introduce configuration inconsistencies
  -upgrades and modifications may introduce errors

- Heterogeneity (hardware and software, availability)

- Instability (hardware and software)

+ Resources belong to different administration domains!

Why FT in Grid is difficult (2/2)



→ When running applications on dynamic and heterogeneous Grid, we may experience many software failures

Research in Grid Fault Tolerance
(some aspects)

Computing models (application runtimes):
- Very few work (**RPC-V**, MPI: **MPICH-V, MPICH-GF**)

Infrastructure:
- Server fault tolerance (GridServices, Webservices, WSRF)
- Fault detectors (few results, Xavier'talk)
- High performance protocols (content distribution: BitTorrent)
- Resource discovery (DHT: Kadelmia)

FT techniques:
- Self stabilization (crash may append during stabilization)
- Consensus (impossibility result on asynchronous network)
- Majority voting (decisions may apply to a majority of nodes absent during the vote...)

Fault tolerance is one research topic of the CoreGrid NoE

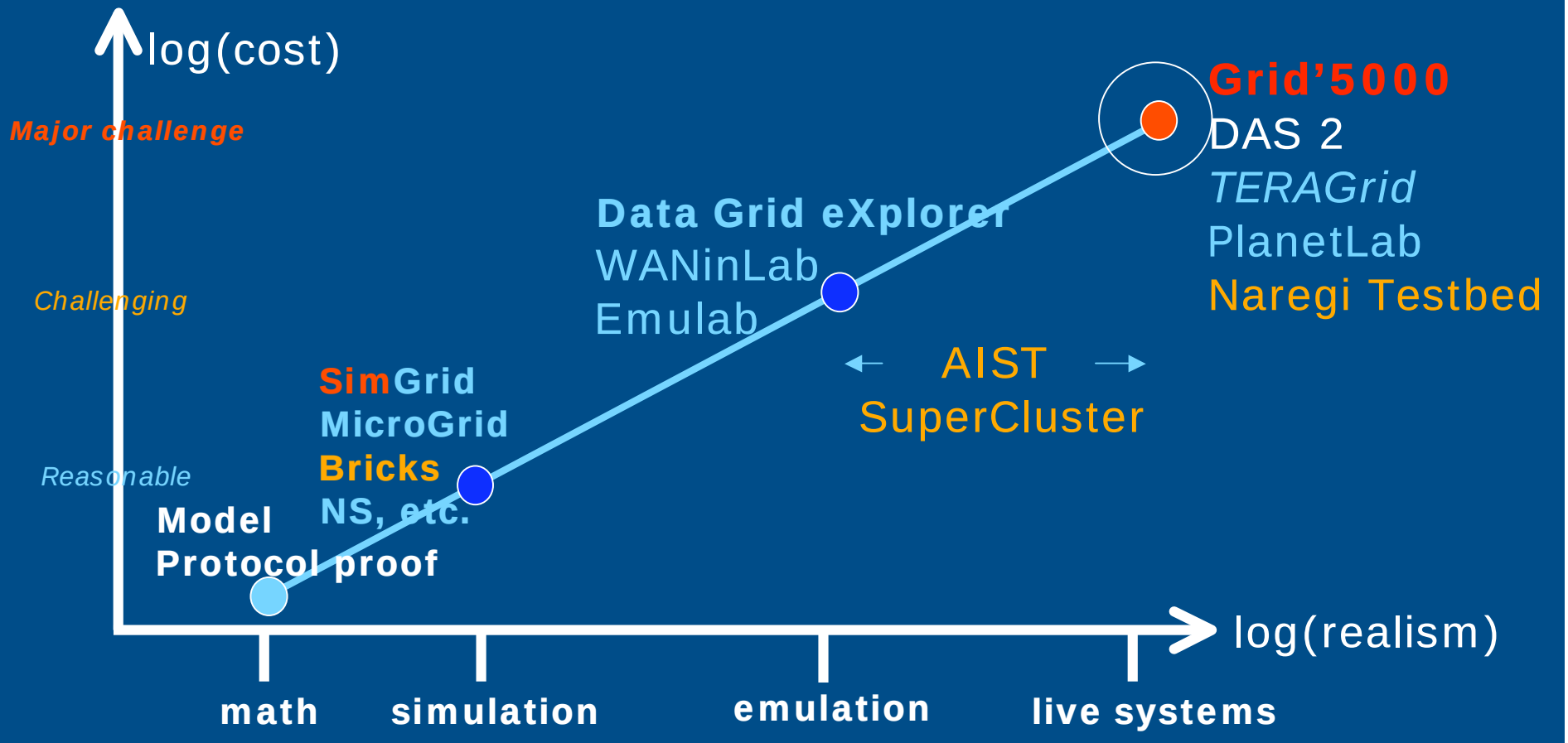
Grid still raises many issues
on fault tolerance,
BUT also on other topics:
performance, scalability, QoS,
resources usage, accounting,
security, etc.

No environment or tool
to test REAL Grid software
at large scale

We need Grid experimental tools

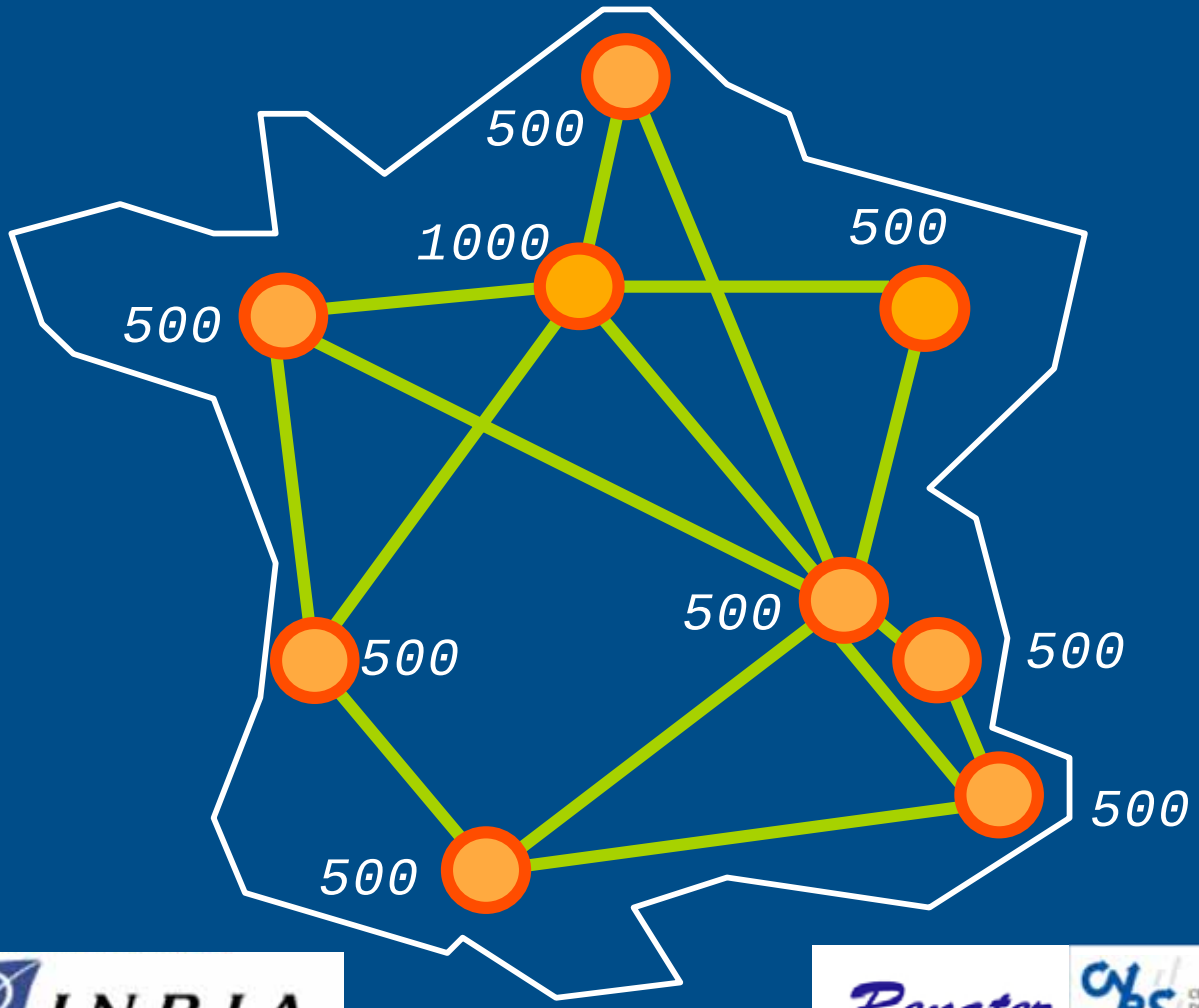
In the first ½ of 2003, the design and development of two Grid experimental platforms was decided:

→ Grid'5000 as a real life system



Grid'5000

The largest research Instrument to study Grid issues



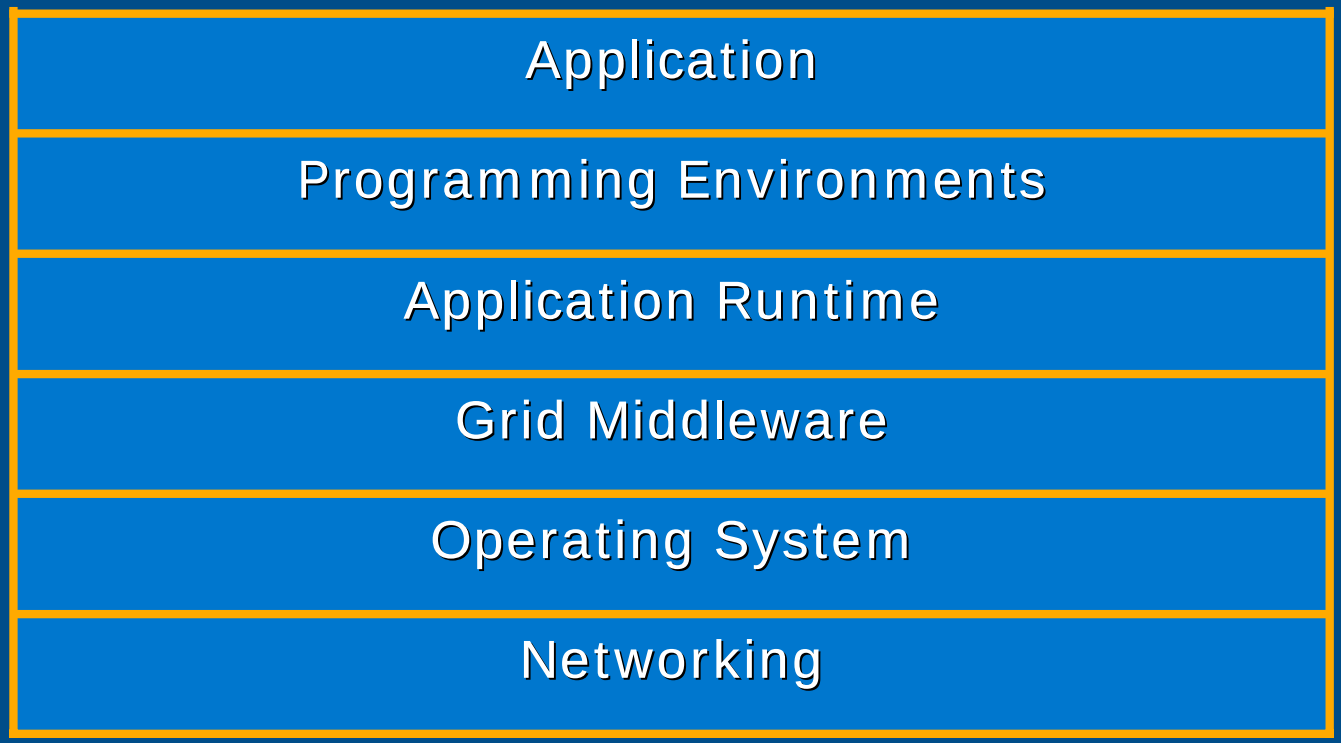
RENATER

Grid'5000 foundations: Collection of experiments to be done

- Networking
  - End host communication layer (interference with local communications)
  - High performance long distance protocols (improved TCP)
  - High Speed Network Emulation
- Middleware / OS
  - Scheduling / data distribution in Grid
  - Fault tolerance in Grid
  - Resource management
  - Grid SSI OS and Grid I/O
  - Desktop Grid/P2P systems
- Programming
  - Component programming for the Grid (Java, Corba)
  - GRID-RPC
  - GRID-MPI
  - Code Coupling
- Applications
  - Multi-parametric applications (Climate modeling/Functional Genomic)
  - Large scale experimentation of distributed applications (Electromagnetism, multi-material fluid mechanics, parallel optimization algorithms, CFD, astrophysics
  - Medical Images, Collaborating tools in virtual 3D environment

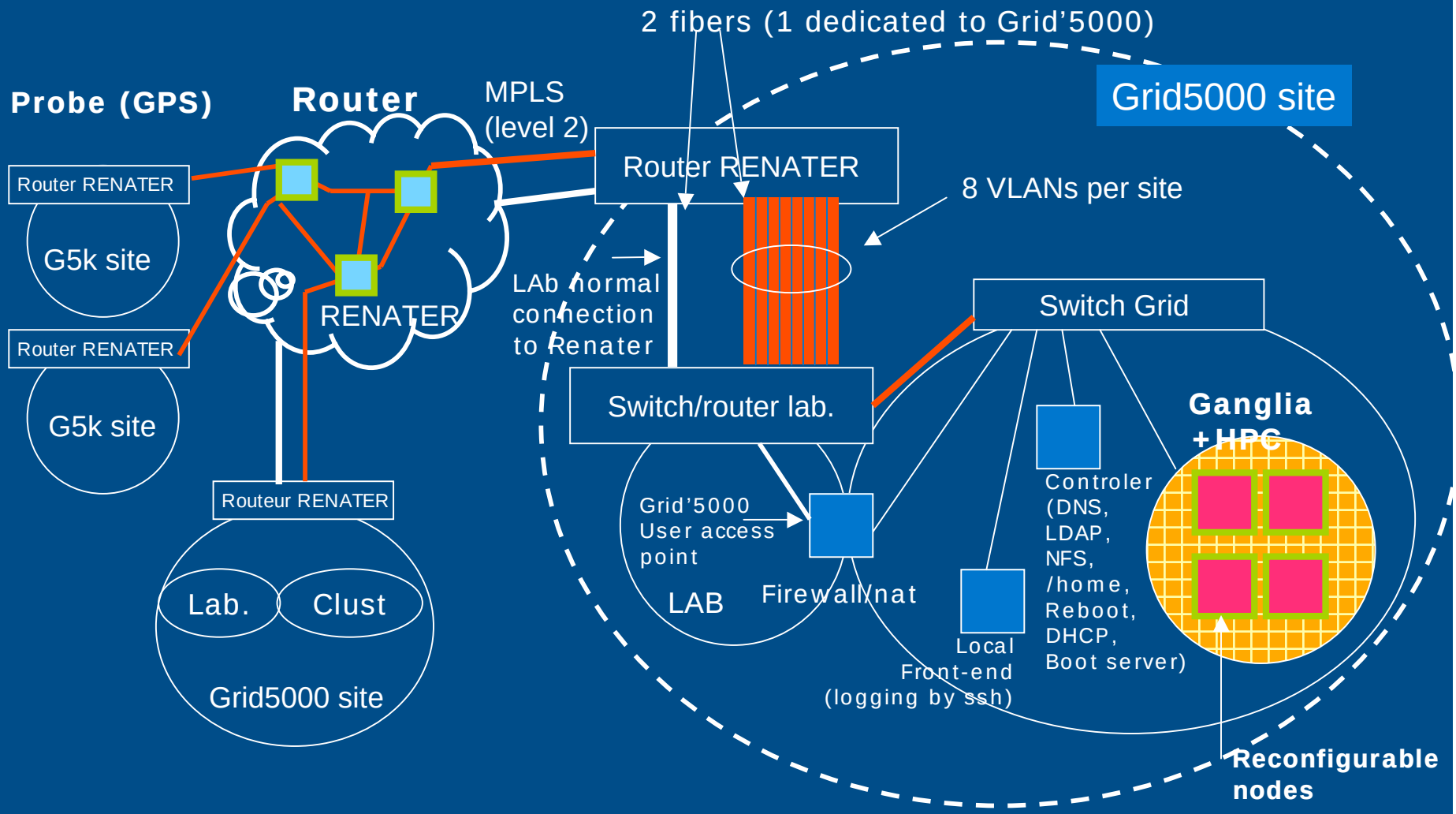
Grid'5000 goal:

Experimenting fault tolerance
and many other topics on
all layers of the Grid software stack

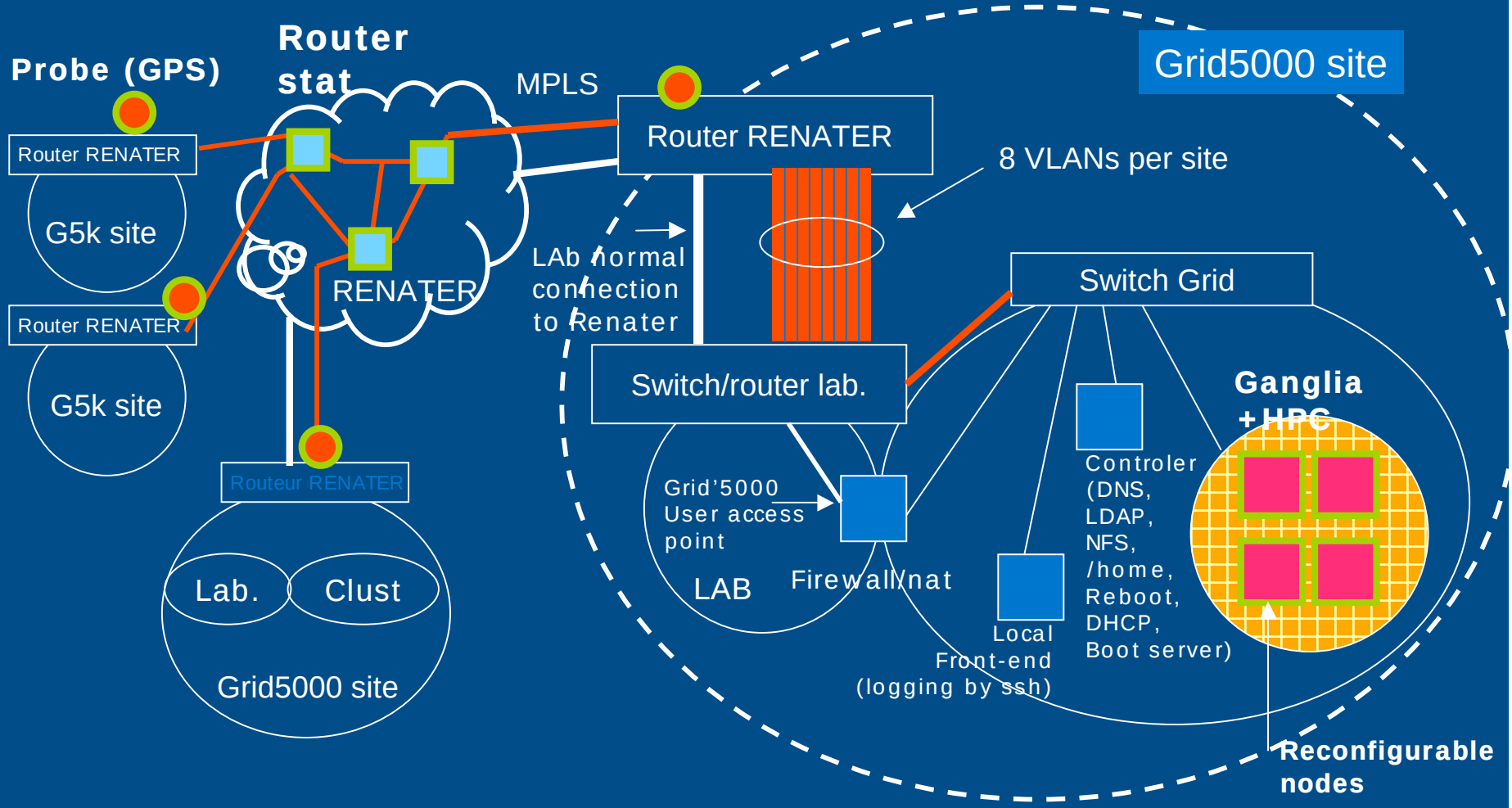


→ A highly reconfigurable, controllable and monitorable experimental platform

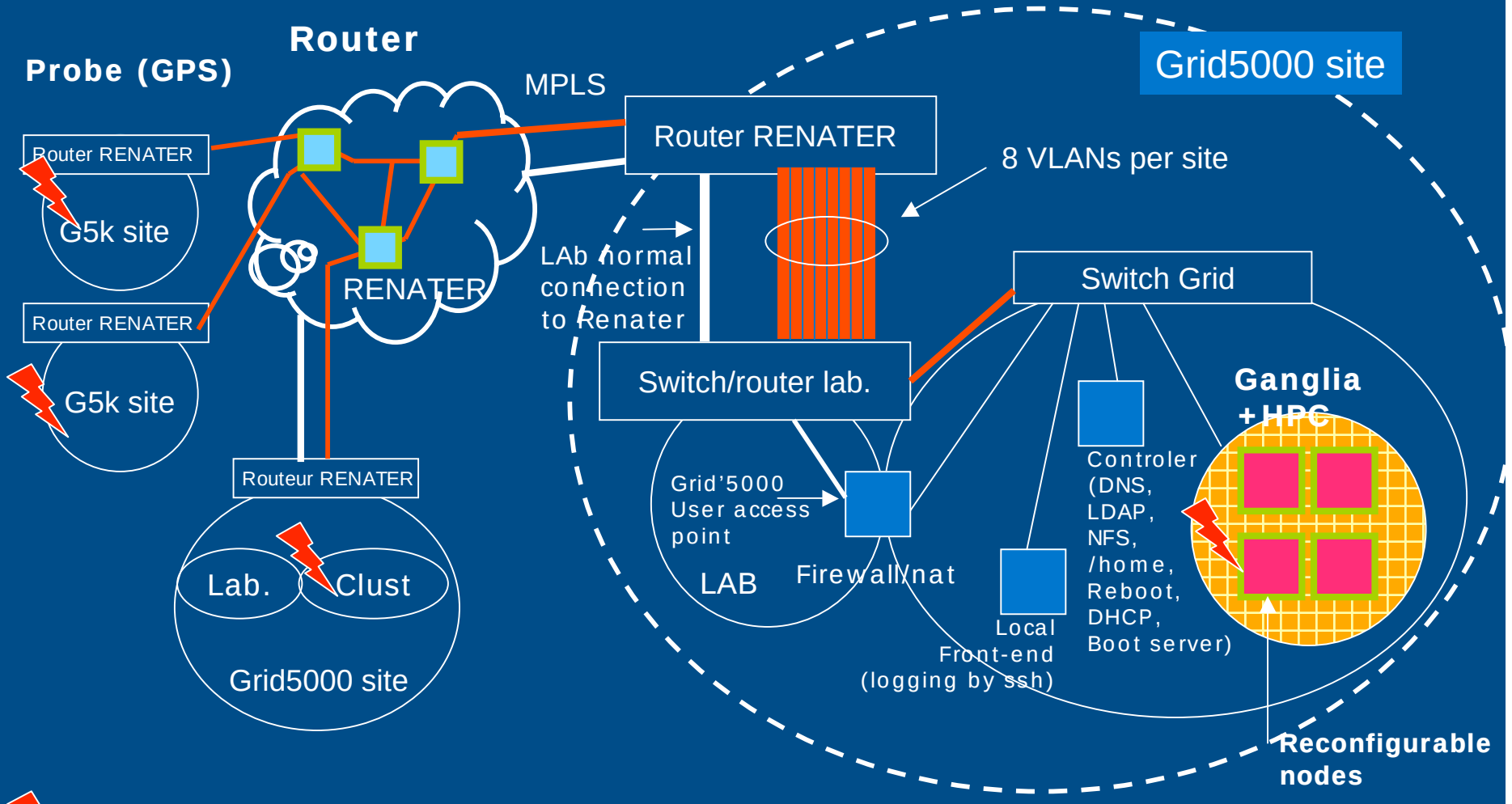
Confinement / isolation



Observation & Monitoring



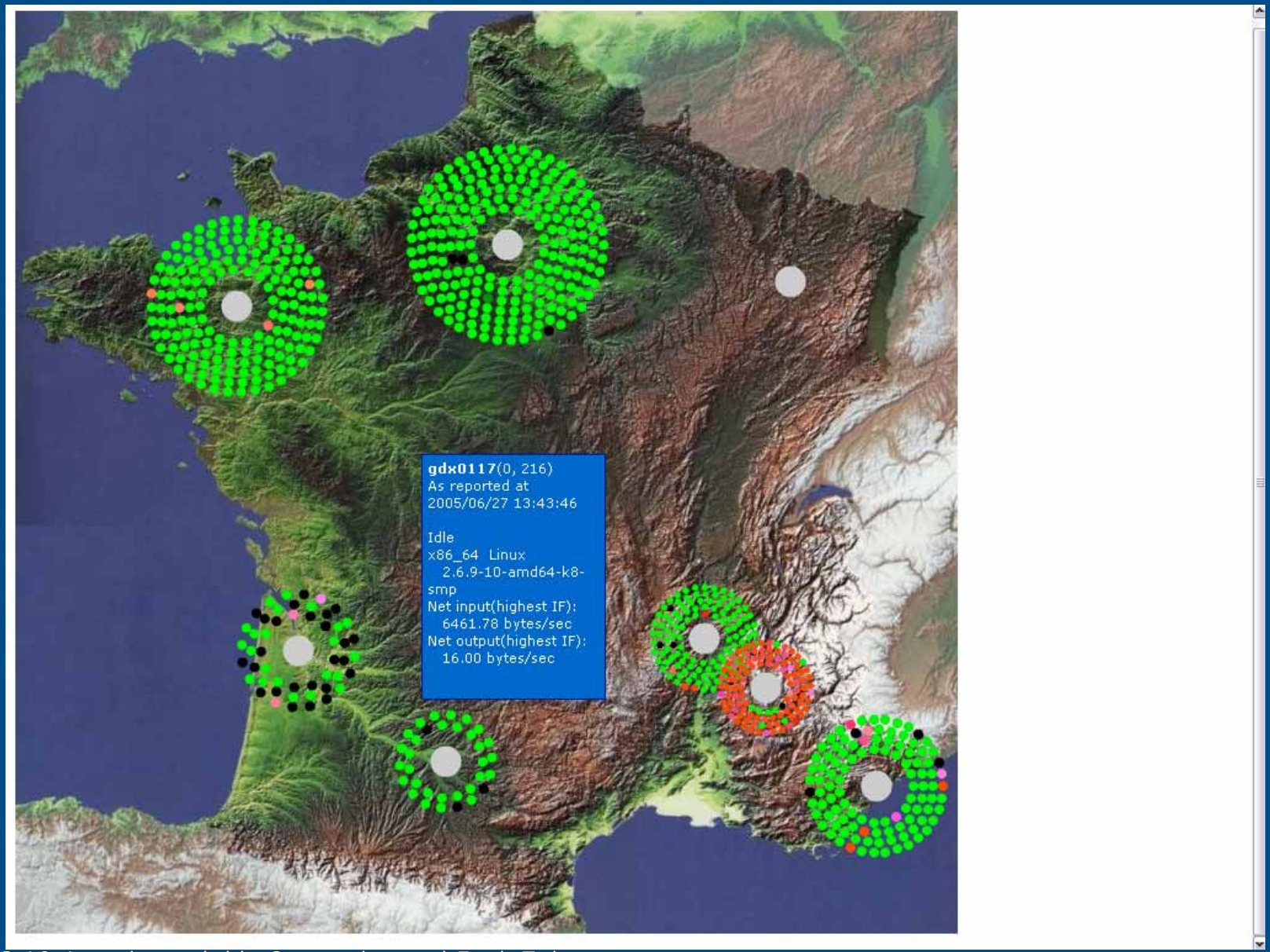
Workload/Traffic & Fault injection



Injectors (process, communication)

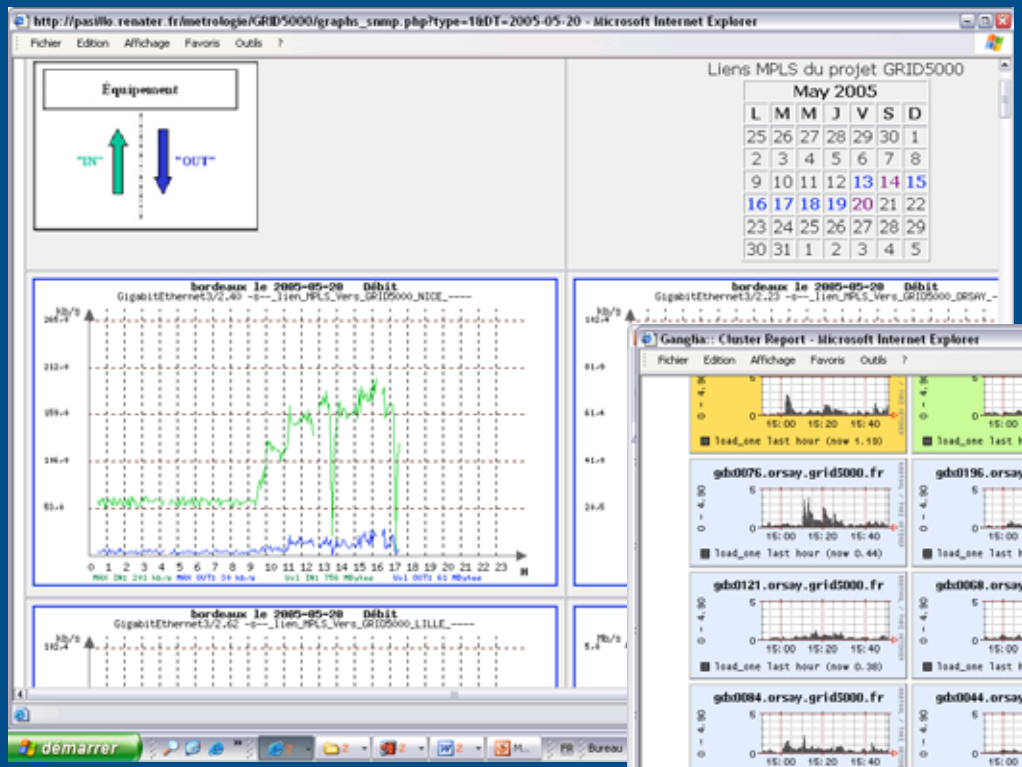

Rennes
Sophia
Lyon
Grenoble
Orsay
Bordeaux
Toulouse

Grid'5000 Global Observer

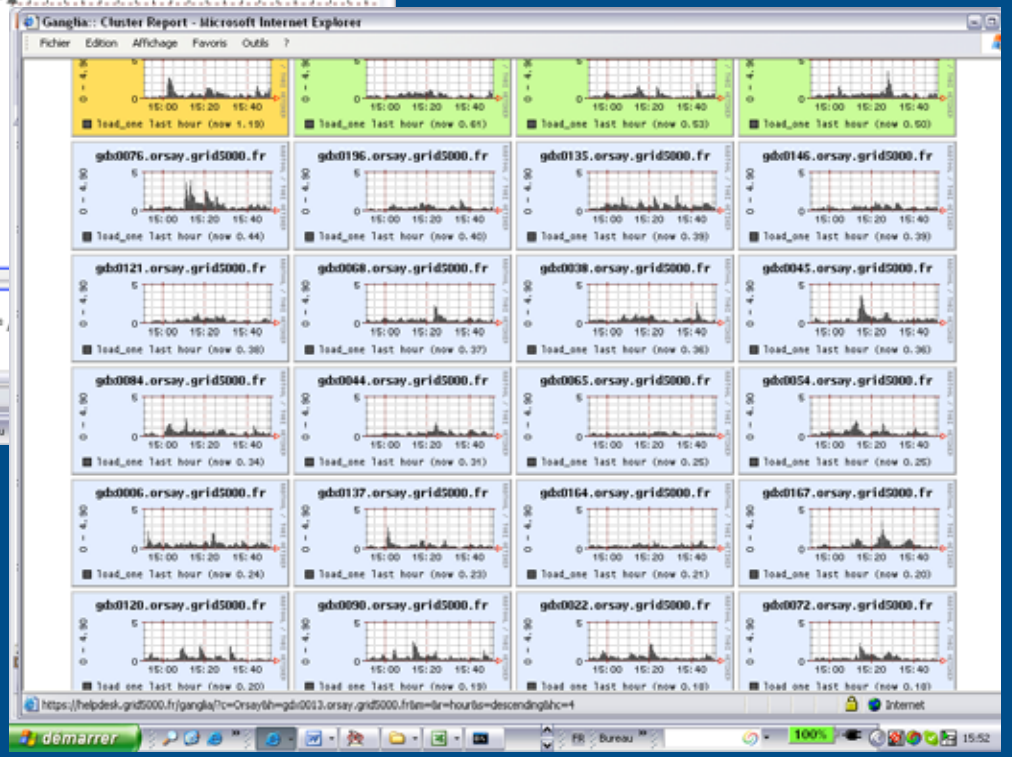


Grid'5000 Monitoring tools

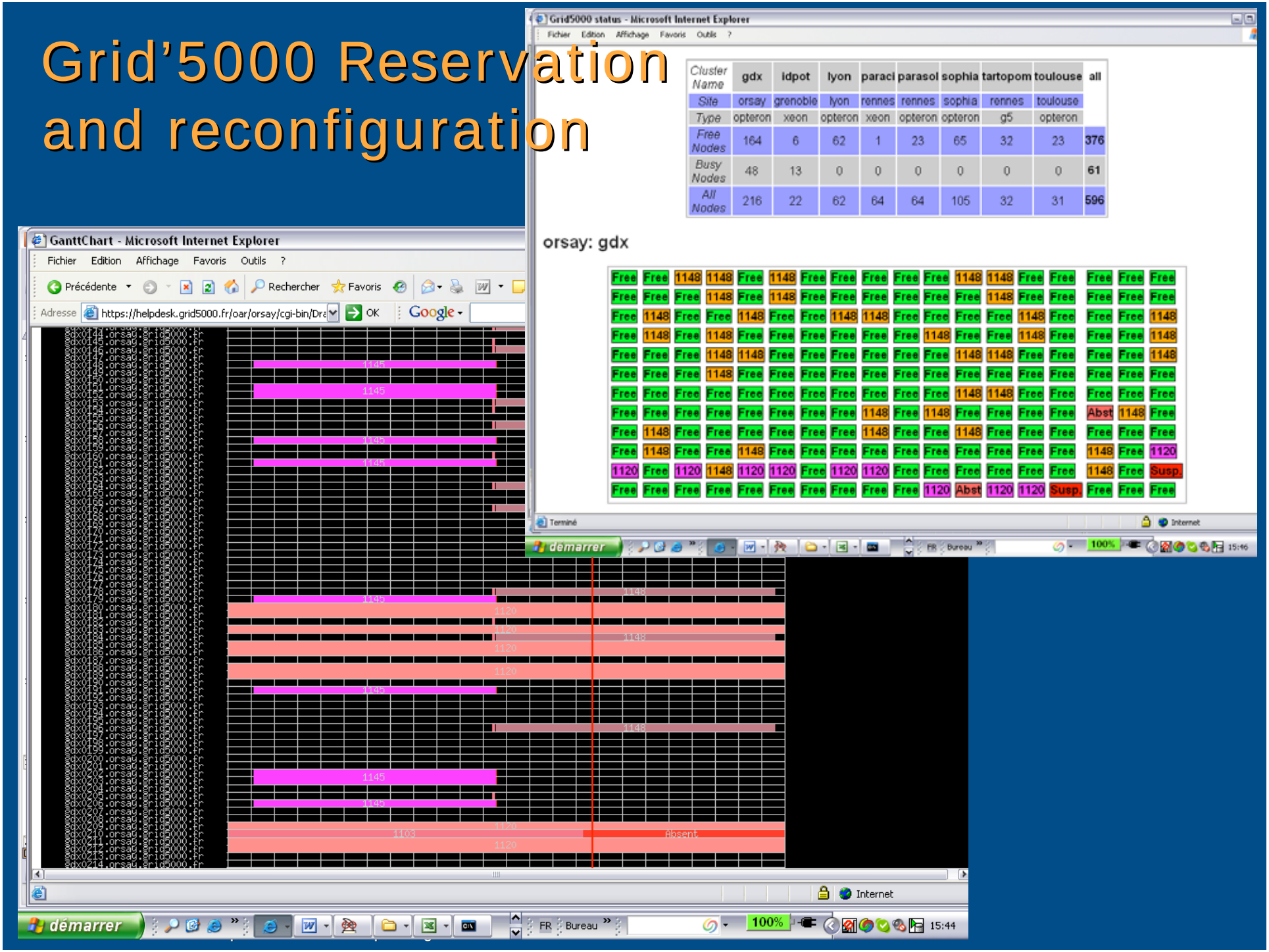
Network traffic



Ganglia

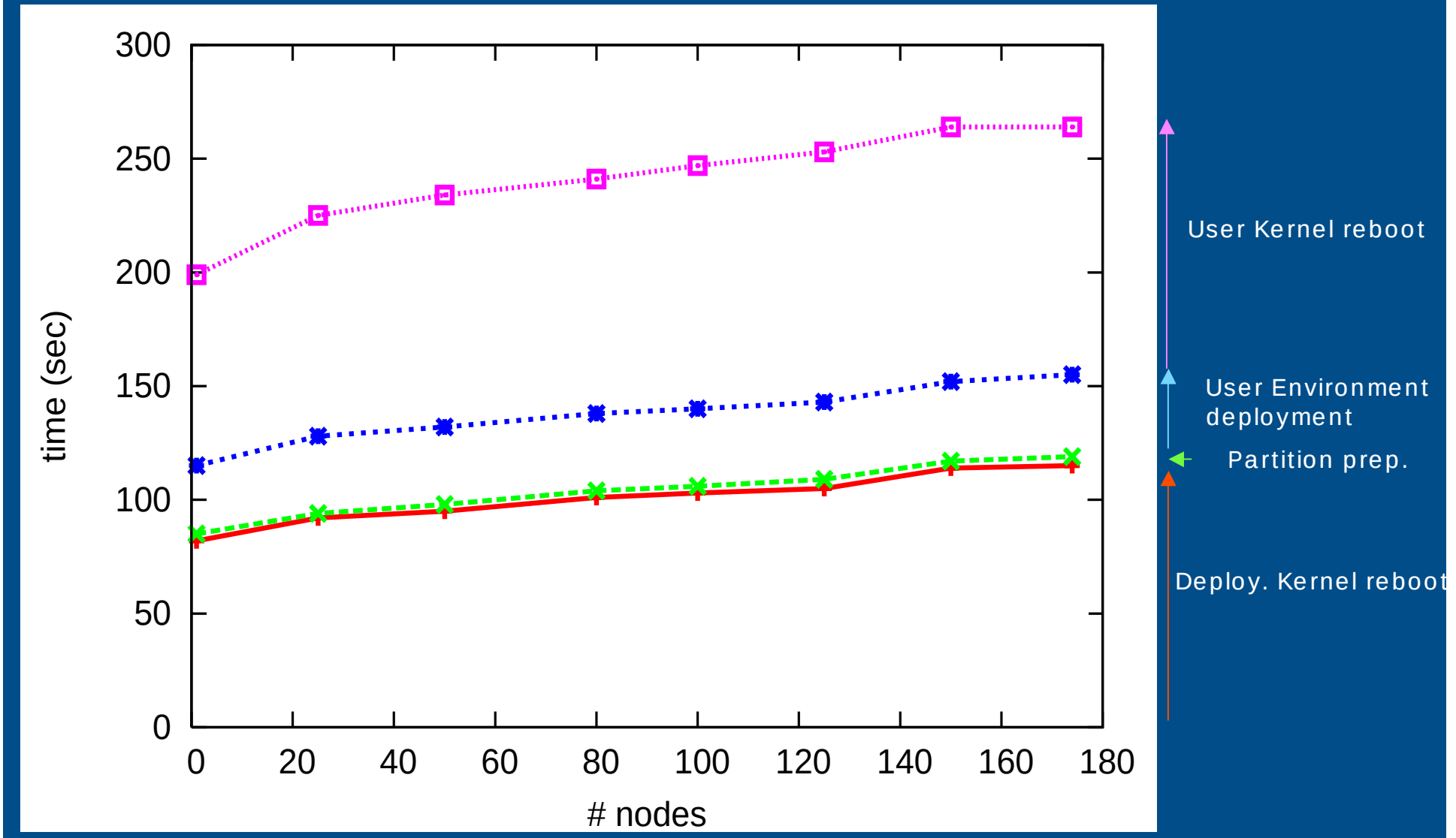


Grid'5000 Reservation and reconfiguration



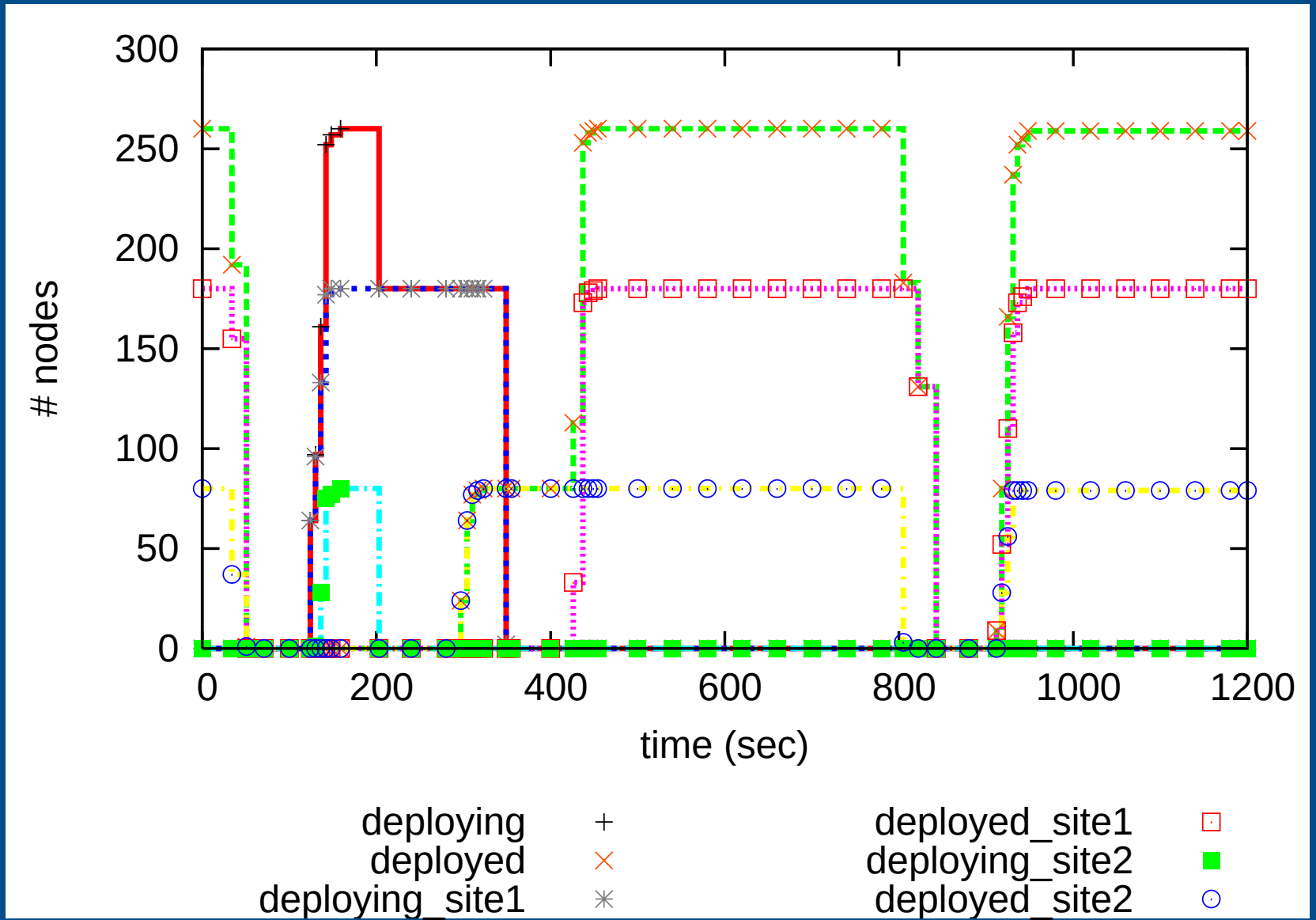
Grid'5000 Reconfiguration time

Time to reboot 1 cluster of Grid'5000 with Kadeploy



Grid'5000 Reconfiguration time

Time to reboot 2 clusters (Paris + Nice) of Grid'5000 (Kadeploy)



Grid'5000 Fault Generator: Fail

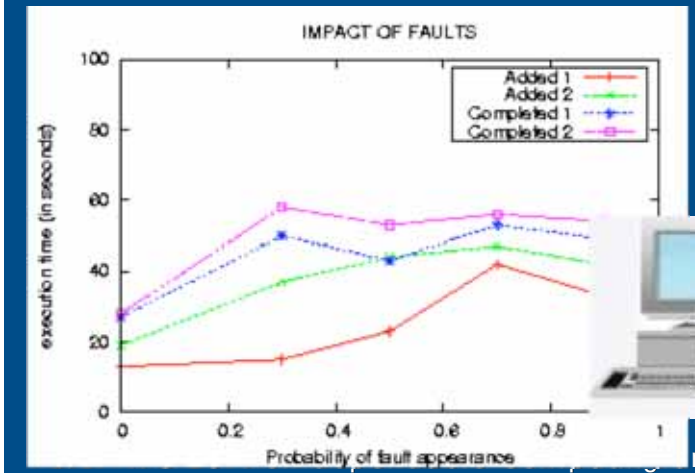
Objectives

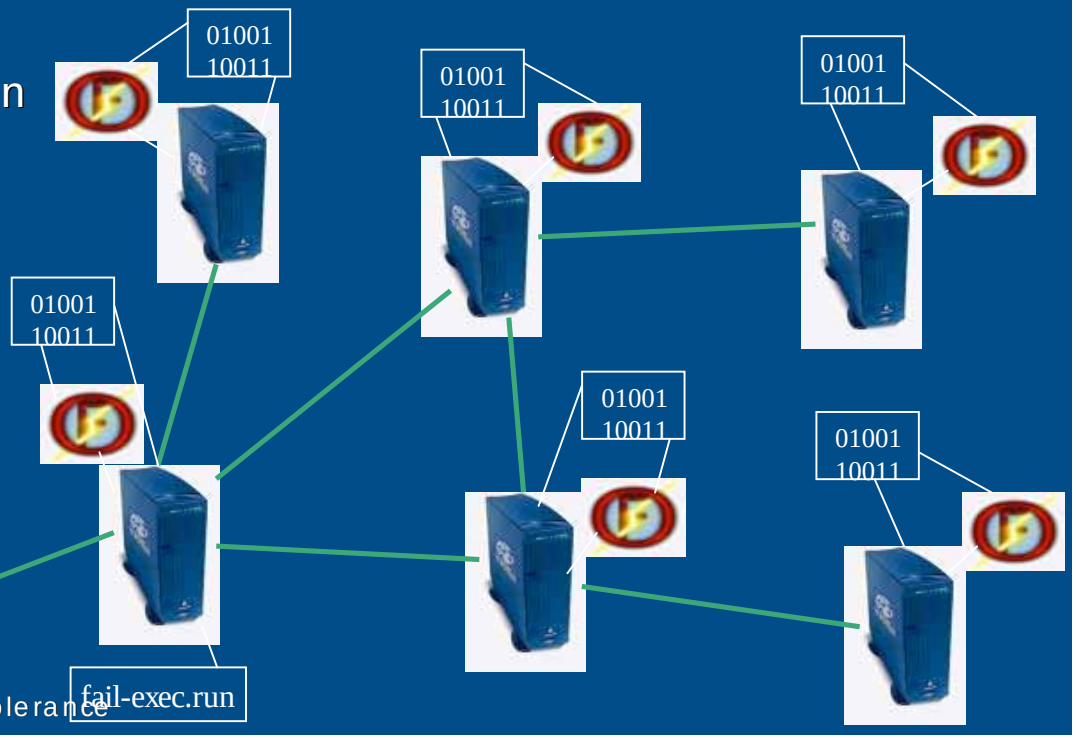
- Probabilistic and deterministic (reproducible) fault injection.
- Expressiveness of scenarios.
- No code modification.
- Scalable.

Concepts

- A dedicated language for fault scenario specification (FAIL: FAult Injection Language).
- Fine control of the code execution (through a debugger)

```
Daemon ADV2
  {
  time_g timer = 5;
  node 1 :
    always int rand = FAIL_RANDOM (1,10);
    timer && rand < 2 -> halt, goto 2;
  node 2 :
    always int rand = FAIL_RANDOM (1,10);
    timer && rand > 7 -> restart, goto 3;
  node 3 :
  }
```





Summary:

- Grid still raises many issues about fault tolerance
- Grid’5000 will offer a large scale infrastructure to study some of these issues (operational in September 2005)
- Grid’5000 will be opened to international collaborations