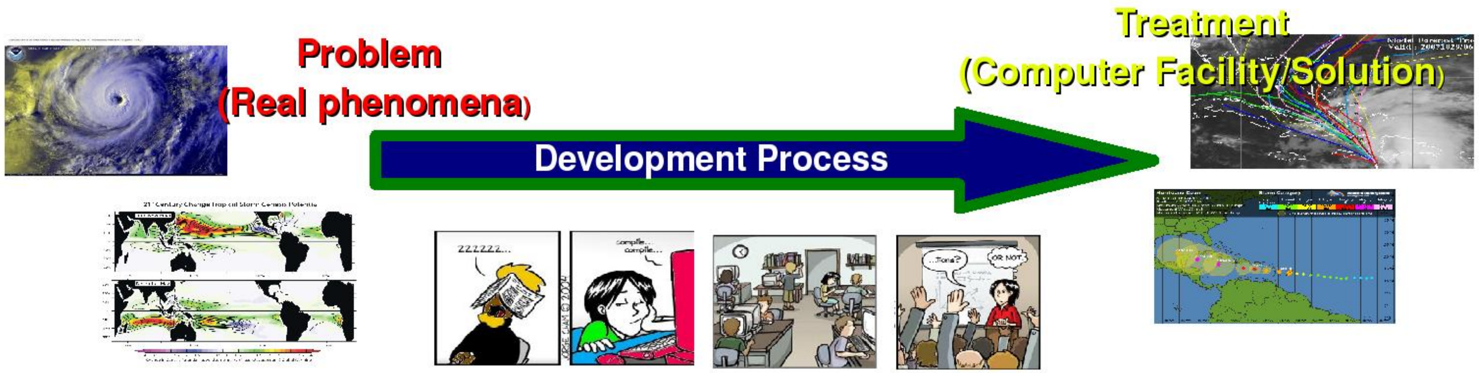
Performance Evaluation

« Performance Evaluation on Scalable Architectures, a first Approach »

Carlos Jaime BARRIOS-HERNÁNDEZ

Conception, Development and use of Computer Solutions



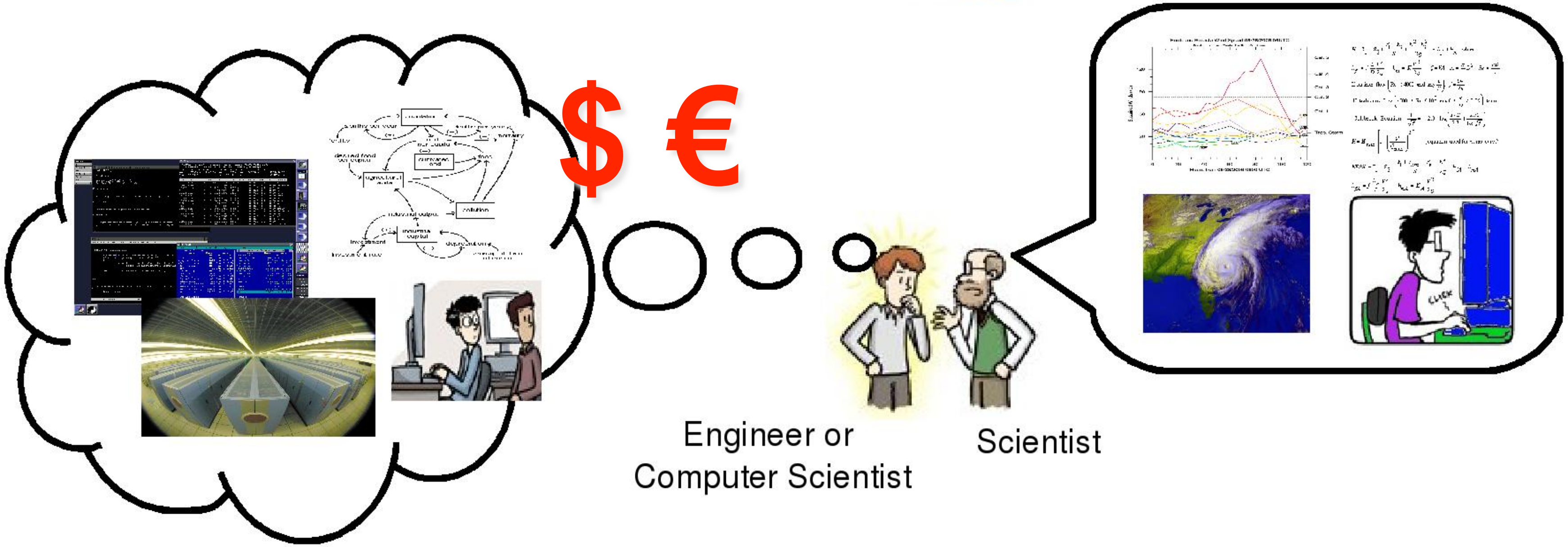
Problem

(Real phenomena)

Development Process

Treatment

(Computer Facility/Solution)



$ €

Engineer or
Computer Scientist

Scientist

So?

- In all steps of a development process, performance evaluation is mandatory.
  - Monitoring
  - Benchmarking
  - Tests (Non-intrusive / Intrusive)
- Performance Evaluation in any Enginnering Systems project allows to take factible and good decisions

The Computer Science Problem

Is Science!

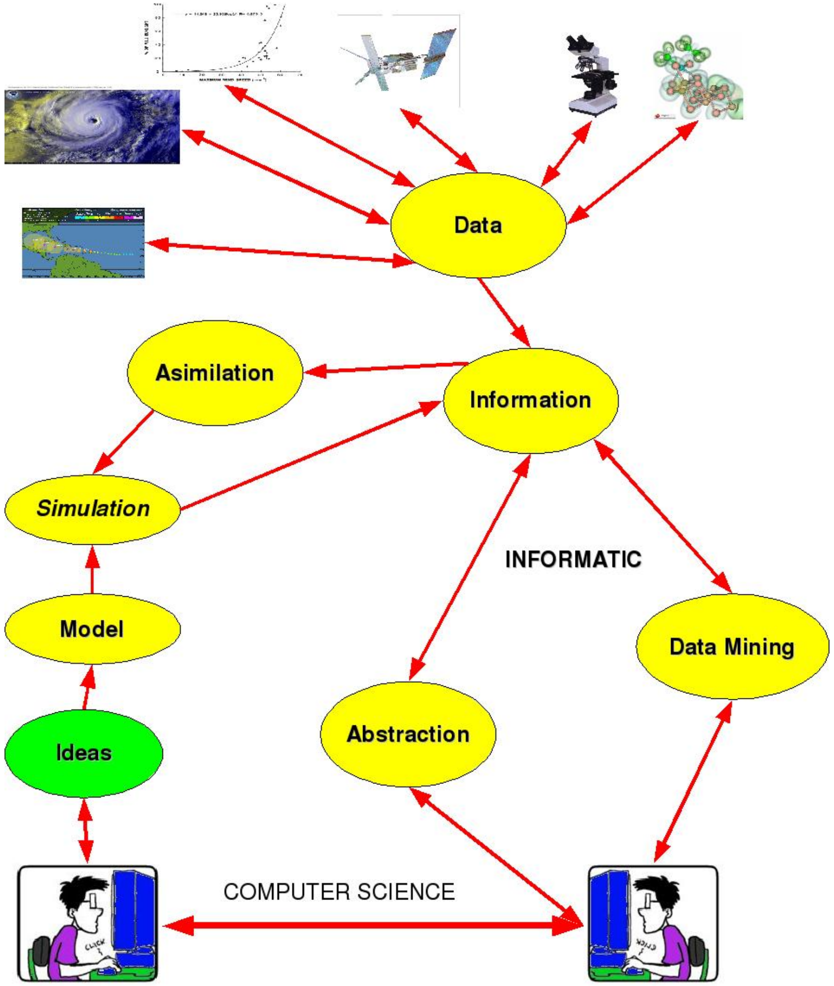
Is Engineering!

Is Art

So.. In performance evaluation context?

- Science is formal and based in scientific method (Mathematical description of phenomena), then computer science/informatics is science.
  - Observations and experimentations in controlled or non-controlled environments
- Engineering is based on science (Applied Science, Mathematic tools to implement models), then systems engineering is an applied science.
  - Tests and benchkmarking
- Informatics requires creativity, passion, inspiration and intuition…
  - Contemplation, esthetical views?

The paradigm…



- What process are part of a scientific approach?
- What process are a technological or engineering approach?
- What process are made with intuition, experience or inspiration?

The Focus Problem

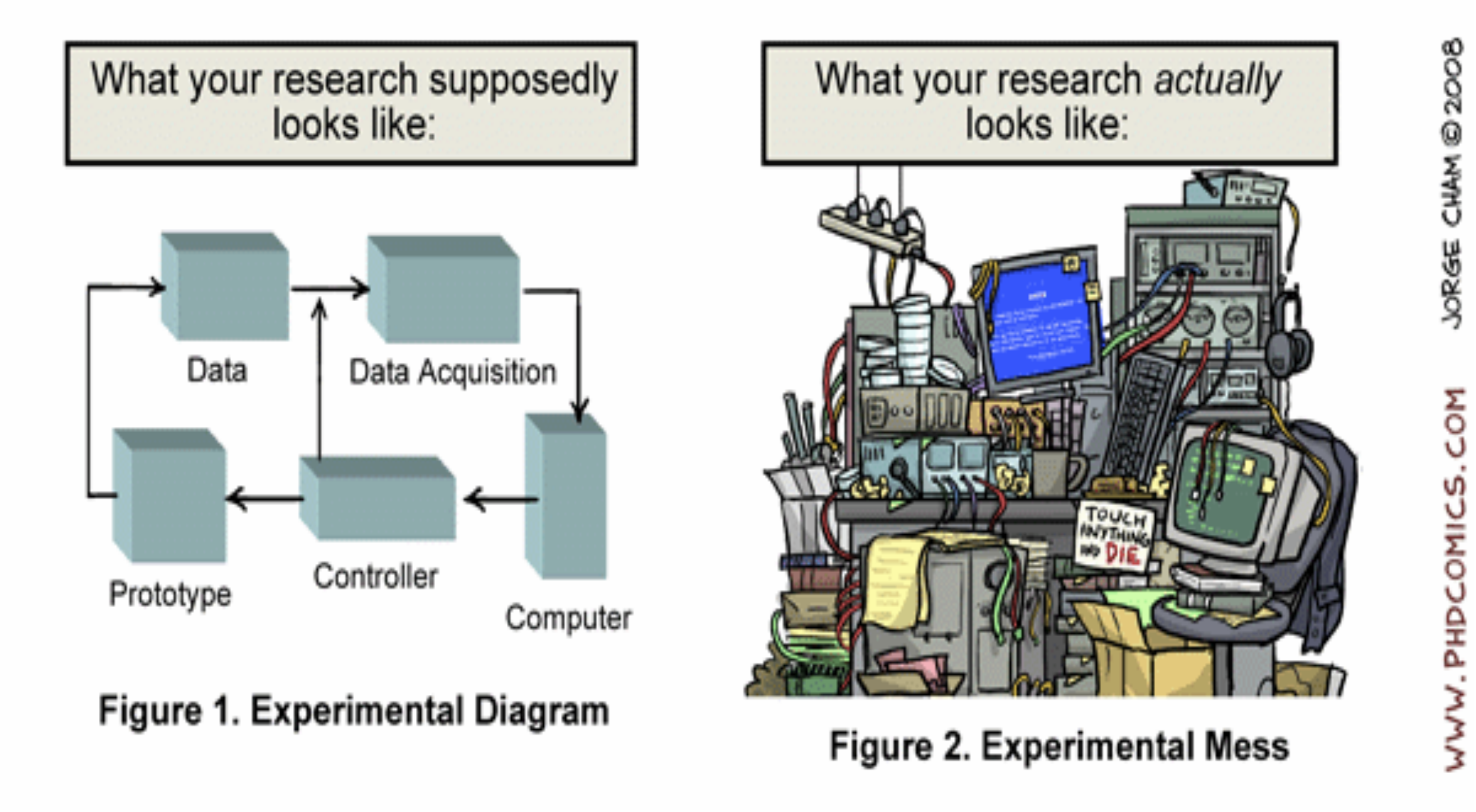
- Theoretical:
  - I know everything... but nothing works.
- Practical:
  - Everything works... but I don't know why.
- Theoretical-Practical (Hybrid):
  - Nothing works... and nobody knows why.

In any case, we need predict the behavior of our system

Outline

- Introduction
  - Experimentation in Computer Science
- Performance Evaluation
  - Techniques
  - Metrics
- The Grid Computing Case (and Scalable Architectures)
  - Grid'5000 Case
- Open Questions

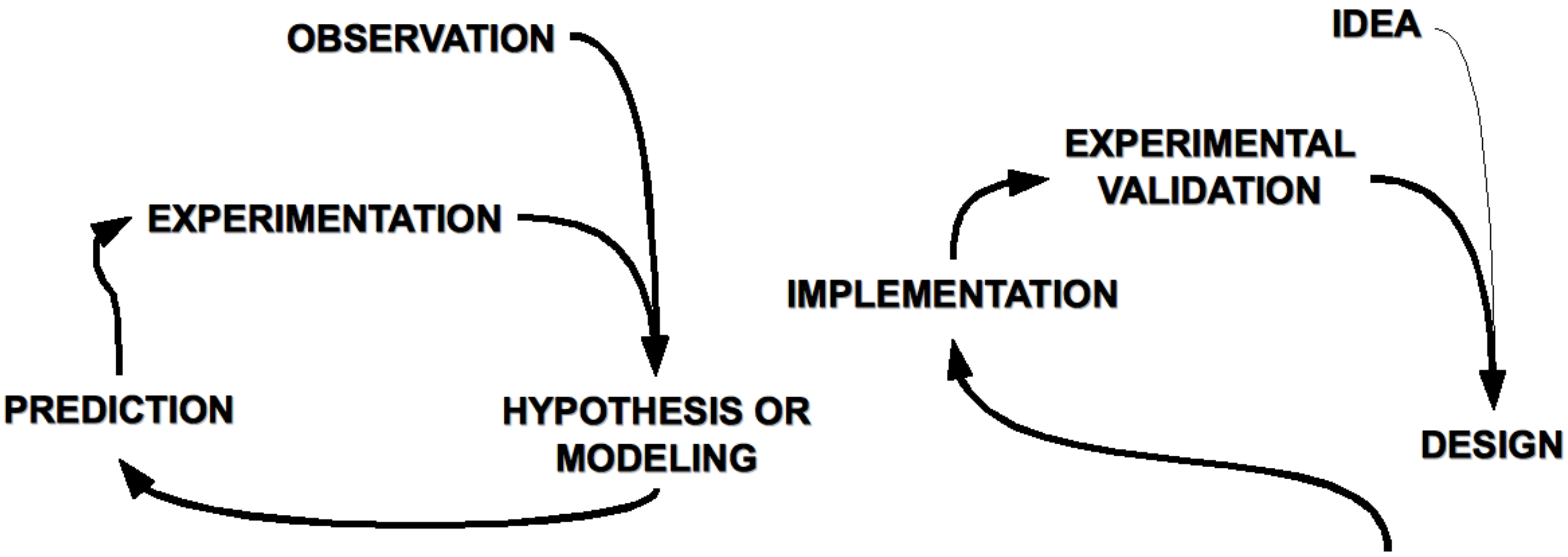
Experimentation Mess


What your research supposedly looks like:
Data
Data Acquisition
Prototype
Controller
Computer
Figure 1. Experimental Diagram
What your research *actually* looks like:
TOUCH ANYTHING AND DIE
Figure 2. Experimental Mess
JORGE CHAM © 2008
WWW.PHDCOMICS.COM

Bad Habits

- No emphasis on design
- Performance evaluation is relegated
  - Absecense of Test Plans
  - Incorrect Metrics to observate
  - Form and Esthetic over Funcional feautures
- No documentation in different levels
  - Developper
  - User
  - Administrator
- Systemic Think forgetfulness

Conception of Systems



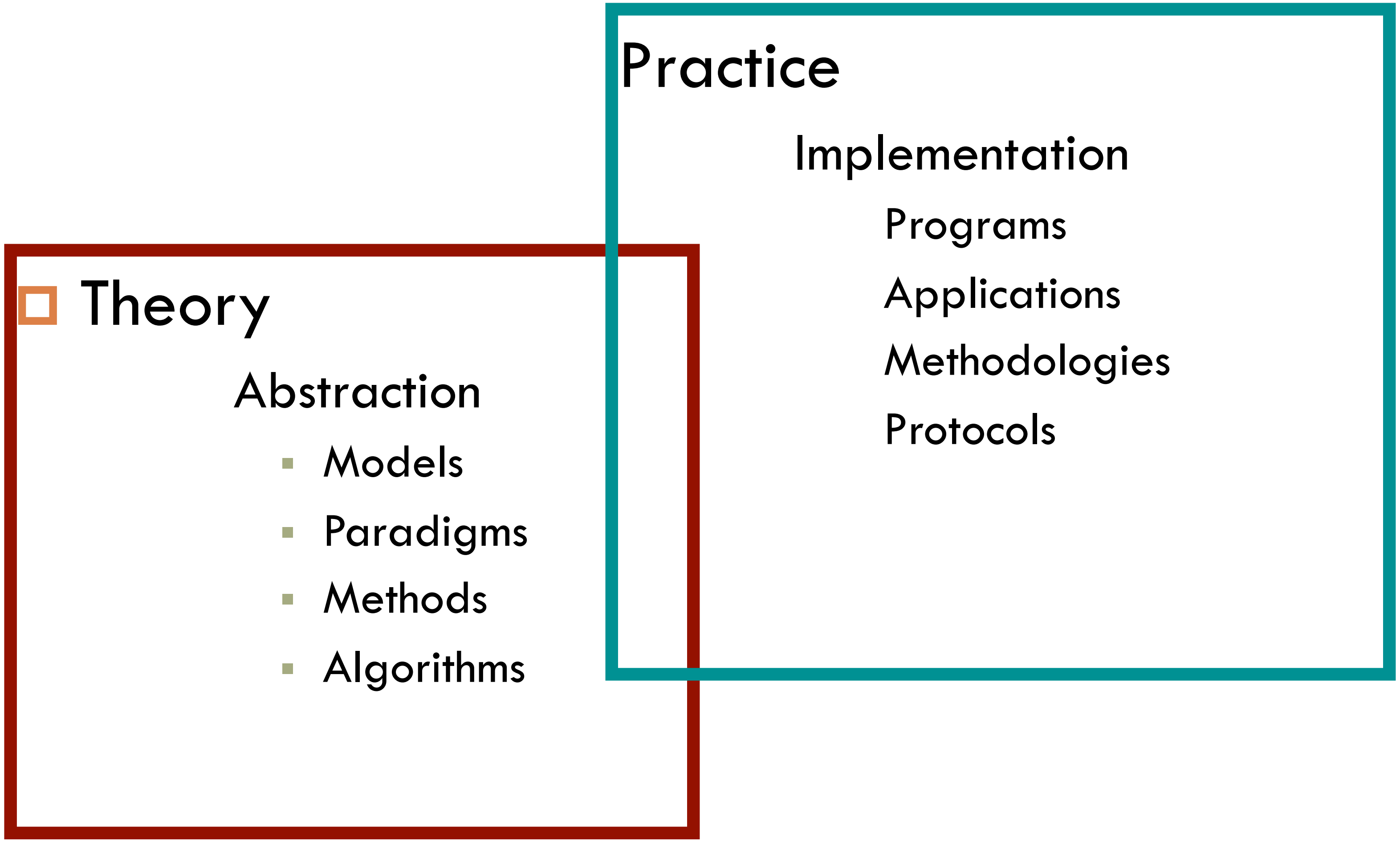
Scientific Method

Conception and Design of Systems

- From Special section in Communications of the ACM 50(11), Nov 2007

Theory and Practice

(Point of view of Systems Conception)



Practice

Implementation

Programs

Applications

Methodologies

Protocols

- Theory

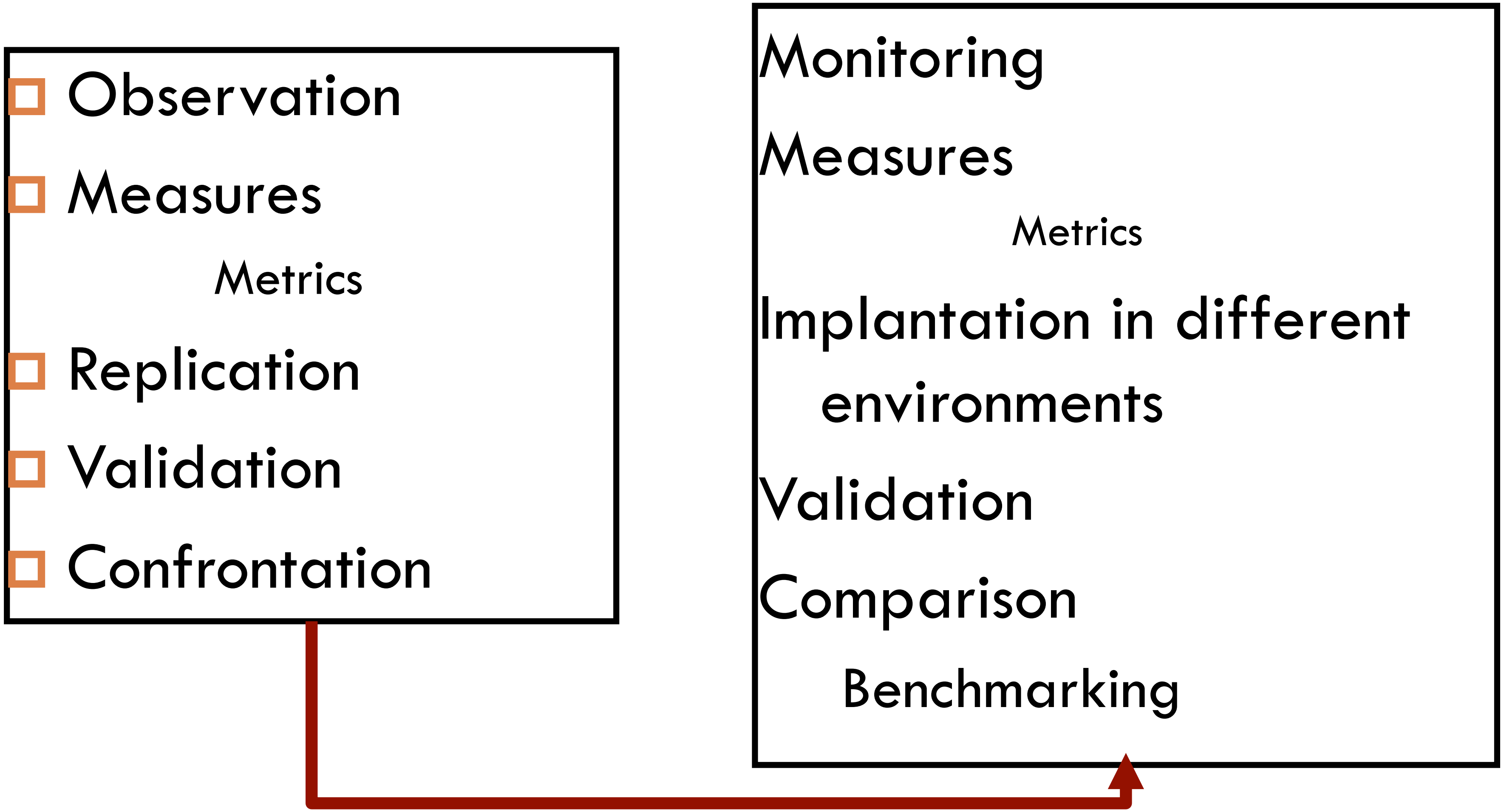
  Abstraction

  - Models
  - Paradigms
  - Methods
  - Algorithms

Science of the Computer Science

- Experimentation (tests) could be confirm or refute the accuracy (efficiency) of a software (system) design.
- Questions and theoric motivations with experiences (tests) produce « good » algorithms and programs.
- Development Cycle of software (systems) include: modeling (design), experimentation (tests – performance evaluation), build (programming)... (It's not a linear cycle).

Observing the Behavior of a System



- Observation
- Measures

  Metrics
- Replication
- Validation
- Confrontation

Monitoring

Measures

Metrics

Implantation in different environments

Validation

Comparison

Benchmarking

Experimental Computer Science

- Experimental Computer Science includes:
  - Observation
  - Confrontation of hypothesis
  - Reproduction of tests

Performance Evaluation

Performance Evaluation

- Application goal is to run with the maximum performance at least cost.
  - Thus, It's necessary Performance Evaluation
- Performance Evaluation is constant in all life cycle of the Application (or system)
  - Design
  - Building
  - Implementation
  - Implantation
  - Use
  - Actualization

Performance Evaluation

(From the Computer Science Problem Heritage)

- Performance Evaluation is a technique:
  - Processes
  - Methodology
  - Tools
- Performance Evaluation is a science:
  - Theoric Basis
  - Experimentation
  - Replication and Validation
- Performance Evaluation is Art:
  - Intuition (Deep Knowledge)
  - Abstraction Capacity
  - Creativity
  - Activity non repetitive
  - Tools

Performance Evaluation allows to know the capacities and limitations of a system.

Modeling, Measuring and Simulating

- Modeling: It allows to build formal abstractions
  - Mathematical Models
  - Analytical Models
  - Causal Models
- Measurement: It allows to characterize
  - Tests, Experiences in environments controlled known.
- Simulating: It allows to observe defined scenarios
  - In according with the modeling.

Techniques

- MODELING (Analytical Model)
- SIMULATION
- EXPERIMENTATION (TESTS – MEASUREMENT)
  - Tests in controlled systems
  - Tests « On Live » (also controlled)

Benchmarking

Tracing and Profiling

ANYONE COULD BE VALIDATE FOR ALMOST ANOTHER ONE!!!

Solution Techniques

	Technique		
Characteristic	**Analytical**	**Simulation**	**Measurement**
Flexibility	High♪	High♪	Low♪
Cost	Low♪	Medium♪	High♪
Believability	Low♪	Medium♪	High♪
Accuracy	Low♪	Medium♪	High♪

From Measuring Computer Performance: A Practitioner's Guide, David J. Lilja 2004

Performance Evaluation Steps

1. Stablish the goals of the study and define the system boundaries.
2. List system services and possible outcomes
3. Select performance metrics
4. List system and workload parameters.
5. Select factors and their values.
6. Select evaluation techniques.
7. Select the workload.
8. Design the experiments.
9. Analyze and interpret the data.
10. Present the results. Start over, if necessary.

- **From The Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling de Raj Jain .Wiley Computer Publishing, John Wiley & Sons, Inc.**

About the Metrics

- Performance metrics are
  - Count

    Of how many times an event occurs
  - Duration

    Of a time interval
  - Size

    Of some parameter
  - Derivated values from these measurements

Time-normalized metrics

- « Rate » metrics
  - Normalize metric to common time basis
    - Transactions per second
    - Bytes per second
  - (Number of events) ÷ (time interval over which events occurred)
- « Throughput »
  - Average rate of successful message delivery over a communication channel
- Useful for comparing measurements over different time intervals

Good Metrics Characteristics

- Allows accurate and detailed comparisons
- Leads to correct conclusions
- Is well understood by everyone
- Has a quantitative basis
- A good metric helps avoid erroneous conclusions

Good metrics is

Linear

If metric increases 2x, performance should increase 2x

Reliable

If metric A > metric B

Then, Perf. A > Perf.B

Repeatable

Easy to use

Consistent

Units and definition are constant across systems

Independent

Indepentent to pressure on manufacturers to *optimize* for a particular metric

Performance Metrics Summary

	Clock	MIPS	MFLOPS	SPEC	QUIPS	TIME
Linear					≈ ☺	☺
Reliable						≈ ☺
Repeatable	☺	☺	☺	☺	☺	☺
Easy to measure	☺	☺	☺	½ ☺	☺	☺
Consistent	☺			☺	☺	☺
Independent	☺	☺			☺	☺

From Measuring Computer Performance: A Practitioner's Guide, David J. Lilja 2004

Other metrics

- Response time
  - Elapsed time from request to response
- Throughput
  - Jobs, operations completed per unit time

    E.g. video frames per second
- Bandwidth
  - Bits per second
- *Ad hoc* metrics
  - Defined for a specific need

    Requests per transaction

About the means…

- Performance in systems is multidimensional
  - CPU time
  - I/O time
  - Network time
  - Read/Write speedup
  - Disk Access
  - Storage Capacity
  - Interactions of various components
  - …

About measurement tools and methodologies...

- Actually, measurement tools are based in ***events***:
  - Some predefined change to system state
- Event definition depends on metric being measured
  - Memory reference
  - Disk access
  - Change in a register’s state
  - Network message
  - Processor interrupt

Some measurement techniques comparaison

	Event count	Tracing	Sampling
Resolution	Exact count	Detailed info	Statistical summary
Overhead	Low	High	Constant
Perturbation	~ #events	High	Fixed

From Measuring Computer Performance: A Practitioner's Guide, David J. Lilja 2004

Grid Computing Case

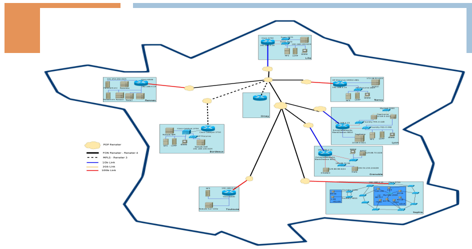
- Distributed environments are too complex to model.
  - Grid Computing is a distributed system
- Grid Computing is heterogeneous, dynamic, pervasive...
  - HPC Utilization (sometimes HTC use too)
  - Infrastructure Services
  - Virtual Communities
    - Different users
    - Different goals
  - Different architectures and dynamic behaviors

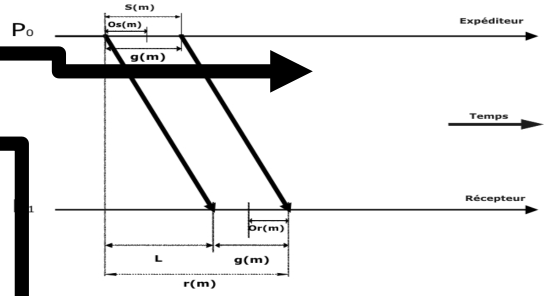
What Evaluate?

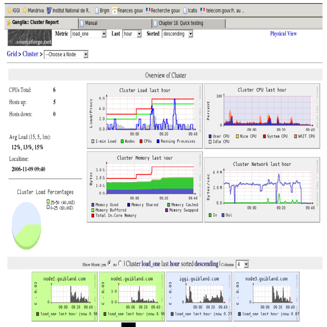
- Infrastructure
  - Monitoring
  - Benchmarking
  - Emulating
- Applications
  - Monitoring
  - Benchmarking
  - Tracing and Profiling
- Users
  - Monitoring
  - Organization Techniques

- Accuracy
  - In accord with your needs)
- Efficiency
  - In accord with the available resources
- Fault tolerance
- Security and Safety

A Case: A Grid'5000 Example







Metric load_one Last hour Sorted descending

Overview of Cluster

CPUs Total: 6

Hosts up: 5

Hosts down: 0

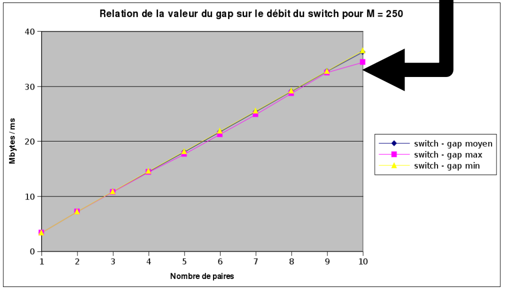
Avg Load (15, 5, 1m):
12%, 13%, 15%

Localtime:
2006-11-09 09:40

Show Hosts: yes ● no ○ | Cluster load_one last hour sorted descending | Columns 4

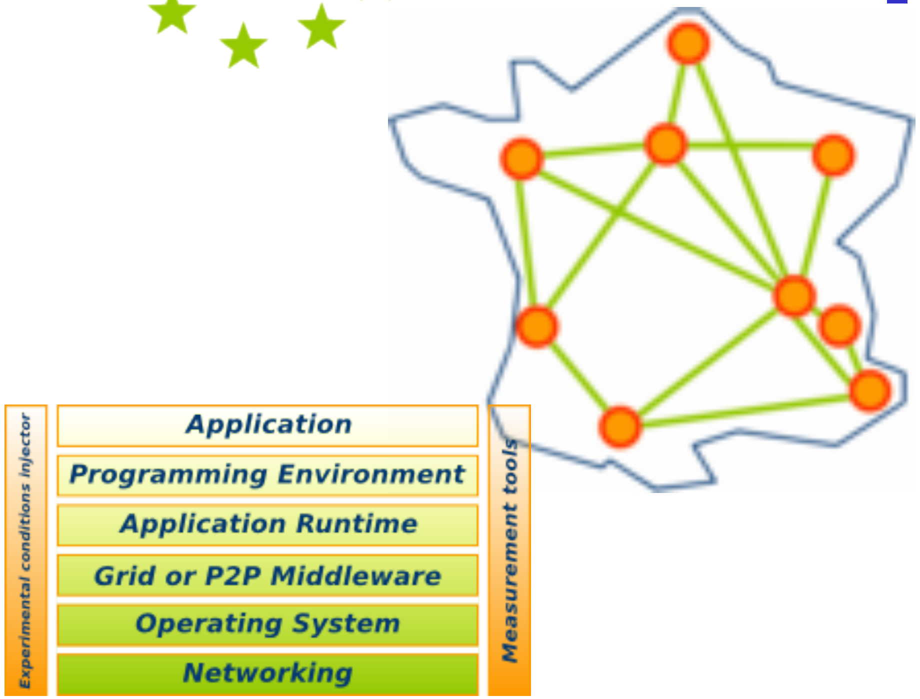
node2.guibland.com
node1.guibland.com
iggi.guibland.com
node3.guibland.com
load_one last hour (now 0.30
load_one last hour (now 0.30
load_one last hour (now 0.22
load_one last hour (now 0.07

$$T(m) = Os(m)+Or(m)+L(m)$$



Grid'5000
(Today Aladdin-Grid'5000)

Grid’5000



Experimental conditions injector

Application

Programming Environment

Application Runtime

Grid or P2P Middleware

Operating System

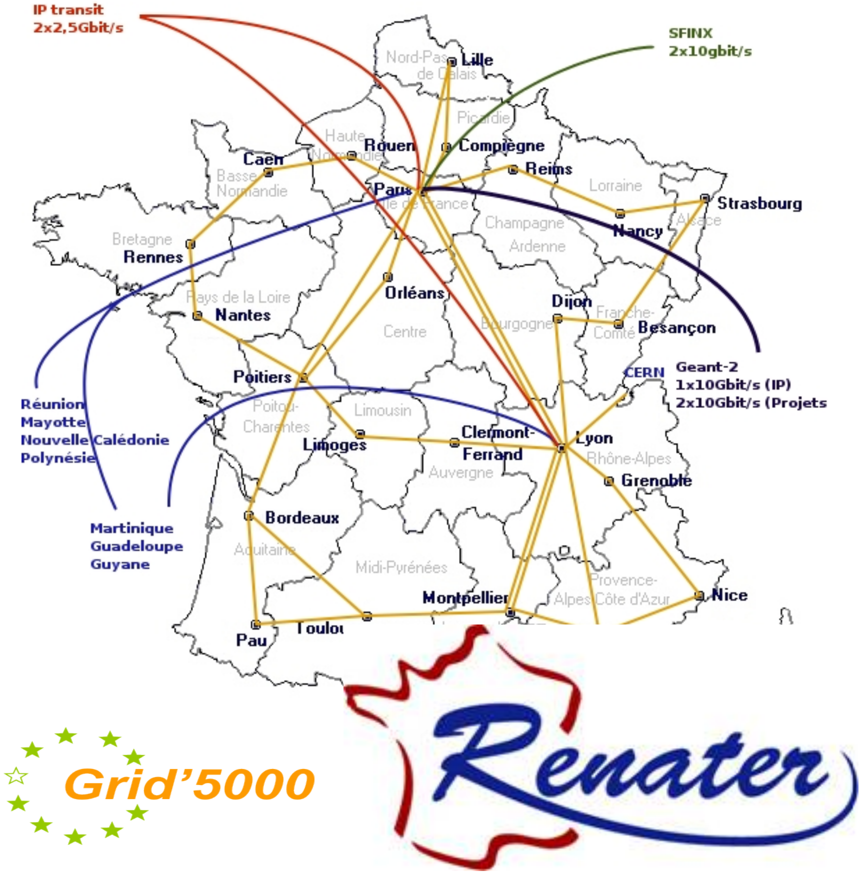
Networking

Measurement tools

- Grid'5000 is a research effort developping a large scale nation wide infrastructure for Grid research in France.
- Grid'5000 is highly reconfigurable, controlable and monitorable experimental Grid platform gathering 9 sites geographically distributed in France.

Grid'5000 interactuate with external sites: Netherlands (DAS-3), Japan (Naregi) and Brazil (UFRGS)

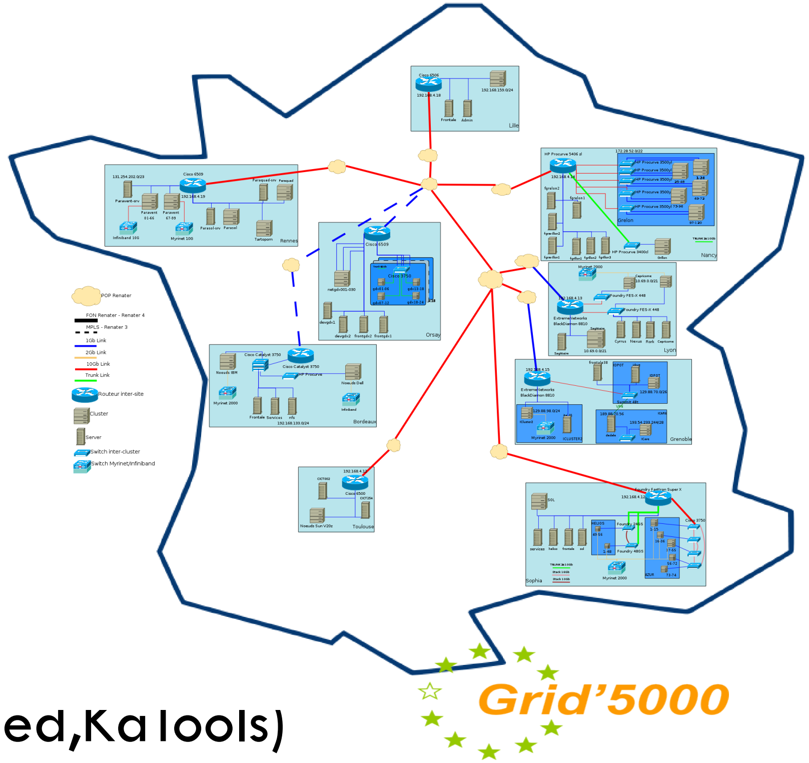
Grid'5000 Interconnexion



- 3 Levels:
  - IntraCluster
    - Myrinet
    - GigaEthernet / Infiniband
  - Grid
    - Giga Ethernet (Best Case 10GB/s, Worst Case: 1GB/s)
  - ExtraGrid
    - External links (~1GB/s)

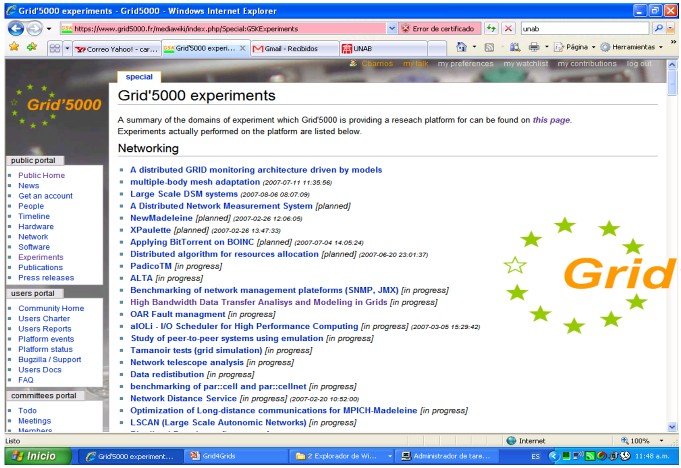
Heterogenity

- 9 National Sites:
  - Bordeaux, Grenoble, Lille, Lyon, Nancy, Orsay, Rennes, Sophia-Anipolis, Toulouse
  - 18 clusters
  - Processor Families (4792 Cores)
    - AMD Opteron (78%) (Now)
    - Intel Xeon EM64T (22%) (Now)
    - Intel Xeon IA32 (Past)
    - Intel Itanium -2 (Past)
    - IBM Power PC (Past)
  - Software Resources:
    - A General Scheduler (OAR)
    - A General Deployer (Kadeploy
    - Middlewares (Diet)
    - Monitoring tools (Nagios, Kaspied,KaTools)
    - Etc...



Users

- ~ 350 Experiments (Mainly in cyberstructures and e-Science)
- ~ 200 Users (Scientifics, enginners in different countries)
- Technical Committee (Engineers)
  - Developer/Support Team (Distributed in different sites ~30)
- Scientific Committee (~15)
  - Thierry Priol (INRIA – General Director)
  - Franck Cappello (INRIA – Scienfic Director)
  - David Margery (INRIA – Technical Director)



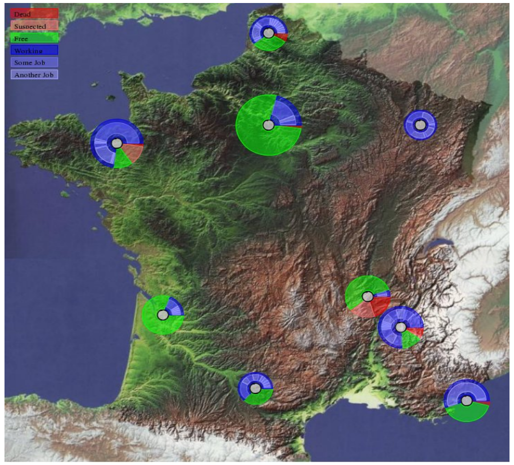
Grid'5000 experiments

A summary of the domains of experiment which Grid'5000 is providing a reseach platform for can be found on *this page*. Experiments actually performed on the platform are listed below.

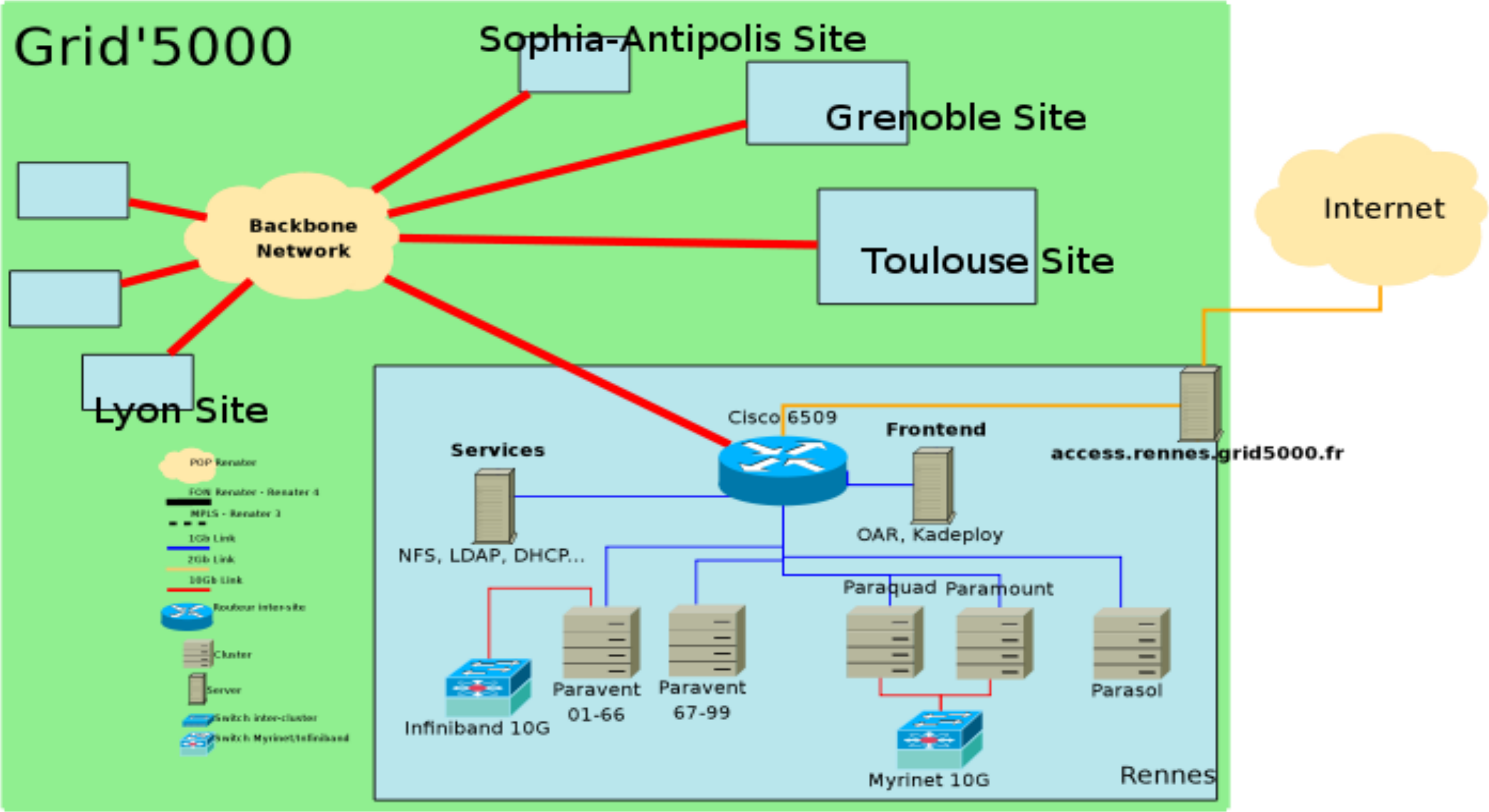
Networking

- **A distributed GRID monitoring architecture driven by models**
- **multiple-body mesh adaptation** *(2007-07-11 11:35:56)*
- **Large Scale DSM systems** *(2007-08-06 08:07:09)*
- **A Distributed Network Measurement System** *[planned]*
- **NewMadeleine** *[planned]* *(2007-02-26 12:06:05)*
- **XPaulette** *[planned]* *(2007-02-26 13:47:33)*
- **Applying BitTorrent on BOINC** *[planned]* *(2007-07-04 14:05:24)*
- **Distributed algorithm for resources allocation** *[planned]* *(2007-06-20 23:01:37)*
- **PadicoTM** *[in progress]*
- **ALTA** *[in progress]*
- **Benchmarking of network management plateforms (SNMP, JMX)** *[in progress]*
- **High Bandwidth Data Transfer Analisys and Modeling in Grids** *[in progress]*
- **OAR Fault managment** *[in progress]*
- **aIOLi - I/O Scheduler for High Performance Computing** *[in progress]* *(2007-03-05 15:29:42)*
- **Study of peer-to-peer systems using emulation** *[in progress]*
- **Tamanoir tests (grid simulation)** *[in progress]*
- **Network telescope analysis** *[in progress]*
- **Data redistibution** *[in progress]*
- **benchmarking of par::cell and par::cellnet** *[in progress]*
- **Network Distance Service** *[in progress]* *(2007-02-20 10:52:00)*
- **Optimization of Long-distance communications for MPICH-Madeleine** *[in progress]*
- **LSCAN (Large Scale Autonomic Networks)** *[in progress]*

Grid'5000



Topology Site



From Introduction to Grid'5000(c) 2008

Benchmark and Workload

- Benchmark:
  - Result of running a computer program, or a set of programs, in order to assess the relative performance of an object by running a number of standard tests and trials against it *(wikipedia)*
- Workload:
  - Quantified effort

    Adition Instructions

    Hybrid Instructions

    Syntetics Programs

    Kernels

    Benchmark Applications

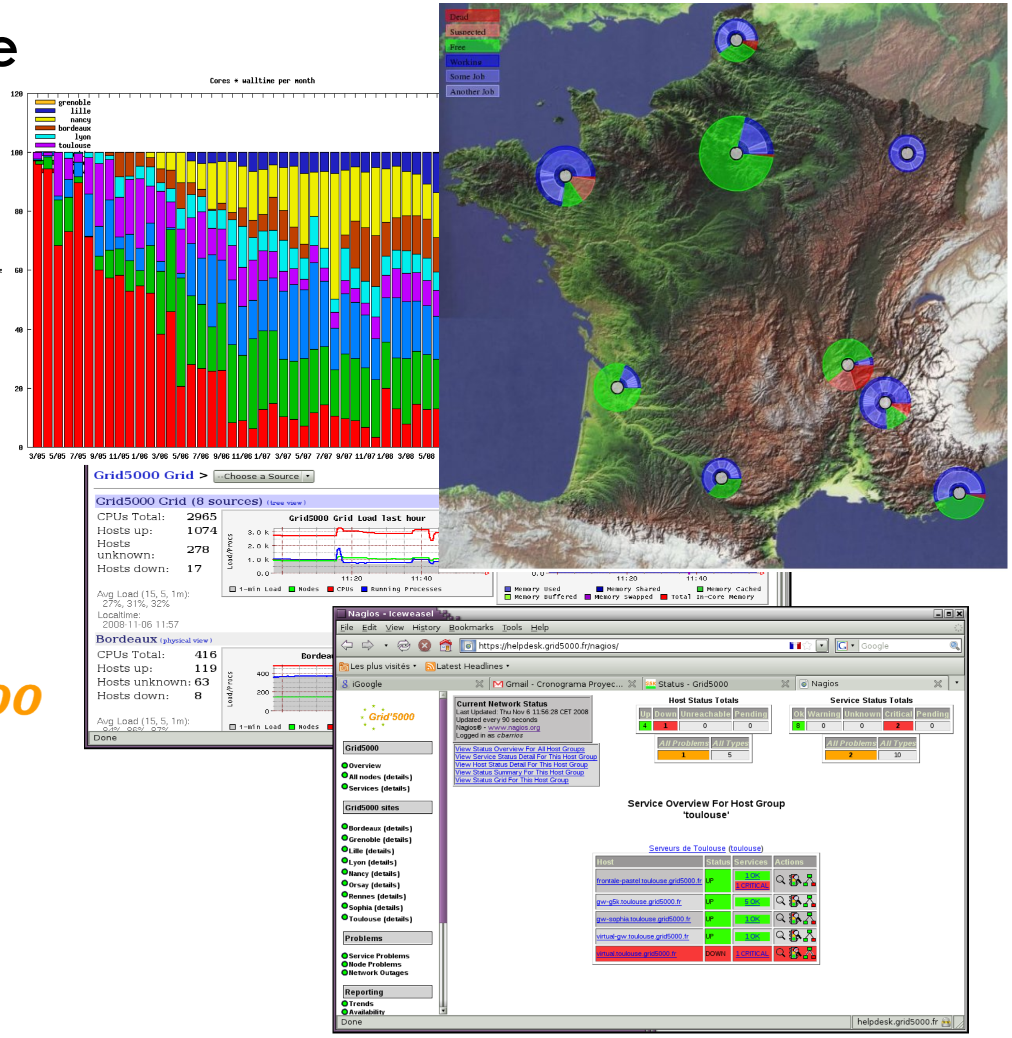
Examples of Benchmarks used in Grid'5000

- Sieve
  - First N- prime numbers
  - Multiprocessor comparaison
- Debit-Credit Benchmark
  - Representaiton fo a bank network
  - Comparaison Standar to transaction process
- SPEC Benchmark Suite
  - Systems Performance Evaluation Cooperative (SPEC): 10 Benchmark Tools to evaluate scientific and enginnering applications.
- MagPIE Benchmark Tools
- NWS Benchkmark Tools
  - Communication Performance in Network Infrastructures

Grid'5000

Monitors in Grid'5000

- Visualization of the State of the System
  - Invasives (Add workload to the syste
  - Non-Invasives
- Ganglia
- MoniKa
- Nagios
- Kaspied
- Pajé (Allows Tracing and Profiling also)



Grid5000 Grid > --Choose a Source

Grid5000 Grid (8 sources) (tree view)

CPUs Total: 2965
Hosts up: 1074
Hosts unknown: 278
Hosts down: 17

Avg Load (15, 5, 1m):
27%, 31%, 32%
Localtime:
2008-11-06 11:57

Bordeaux (physical view)

CPUs Total: 416
Hosts up: 119
Hosts unknown: 63
Hosts down: 8

Avg Load (15, 5, 1m):

https://helpdesk.grid5000.fr/nagios/

Grid5000
- Overview
- All nodes (details)
- Services (details)

Grid5000 sites
- Bordeaux (details)
- Grenoble (details)
- Lille (details)
- Lyon (details)
- Nancy (details)
- Orsay (details)
- Rennes (details)
- Sophia (details)
- Toulouse (details)

Problems
- Service Problems
- Node Problems
- Network Outages

Reporting
- Trends
- Availability

Current Network Status
Last Updated: Thu Nov 6 11:56:28 CET 2008
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *cbarrios*

View Status Overview For All Host Groups
View Service Status Detail For This Host Group
View Host Status Detail For This Host Group
View Status Summary For This Host Group
View Status Grid For This Host Group

Host Status Totals

Up	Down	Unreachable	Pending
4	1	0	0

All Problems	All Types
1	5

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	0	2	0

All Problems	All Types
2	10

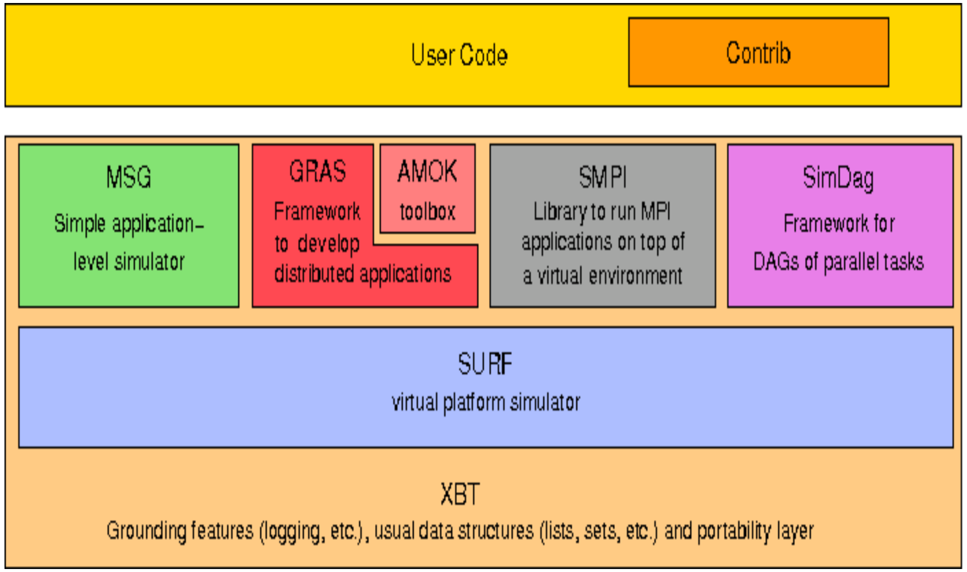
Service Overview For Host Group 'toulouse'

Serveurs de Toulouse (toulouse)

Host	Status	Services	Actions
frontale-pastel.toulouse.grid5000.fr	UP	1 OK 1 CRITICAL	
gw-g5k.toulouse.grid5000.fr	UP	5 OK	
gw-sophia.toulouse.grid5000.fr	UP	1 OK	
virtual-gw.toulouse.grid5000.fr	UP	1 OK	
virtual.toulouse.grid5000.fr	DOWN	1 CRITICAL	

helpdesk.grid5000.fr

Simulators: SIMGRID



User Code

Contrib

MSG
Simple application-level simulator

GRAS
Framework to develop distributed applications

AMOK
toolbox

SMPI
Library to run MPI applications on top of a virtual environment

SimDag
Framework for DAGs of parallel tasks

SURF
virtual platform simulator

XBT
Grounding features (logging, etc.), usual data structures (lists, sets, etc.) and portability layer

SiM GRID

Grid'5000

http://simgrid.gforge.inria.fr/

- SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments.
  - Network of Workstations
  - Grid Computing Platforms

The Critical Behaviors to Evaluate

- Data Transfer
  - High Bandwidth Data Transfer implies heterogenity, dynamicity, concurrence and so on.
- File System Sensibility
  - I/O Sensibility
- Adaptation and Effectiveness
- Scalability
- Fault Tolerance
- Security
- Energy Consumption
- … and the « Human invervention »
- Processing is critical but…

Open Questions

- Performance Evaluation of Systems is REALLY important... then,
  - How to increase the level of accuracy of performance models?

    Of course, it's necessary the definition of metrics and build tools.
  - How to implementate realistic models to performance evaluation?

    On live process (i.e. Production Grid Computing)
  - How to integrate the needs of scientifics and enginner/computer science scientist in performance evaluation?

    Typical example: Program adaptation from clusters to grids.

Recommended Lectures

- **The Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling de Raj Jain .Wiley Computer Publishing, John Wiley & Sons, Inc.**
- **Measuring Computer Performance: A Practitioner's Guide, David J. Lilja 2004**

"Houston, We've Had a Problem"

JAMES A. LOVELL

(NASA Apollo XIII Mission)