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
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

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

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 $A_1 = k \times A_2$ then in reality $A_1 = k \times A_2$
  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

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

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

Environment Stack

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?

Research issues at each layer of the stack
• algorithms
• software
• data
• models
• …
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

- **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 -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

<table>
<thead>
<tr>
<th>Summary:</th>
<th>Free</th>
<th>Busy</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>57</td>
<td>18</td>
<td>133</td>
</tr>
<tr>
<td>Cluster</td>
<td>164</td>
<td>160</td>
<td>205</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reservations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 core</td>
</tr>
</tbody>
</table>

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!

![Diagram showing Kadeploy deployment process]

**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 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 can 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 Research-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'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 ➔ 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 overperform 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

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

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

4\textsuperscript{th} 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.

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
- 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
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

**EPCC**

- Permanent Resources
  
  1 server Intel Xeon E5620 (2x4 cores)
  2 worker nodes AMD Opteron 6176 (each 4x12 cores)

- Networking: Gigabit Ethernet

**HP**

- Permanent Resources
  
  32 nodes Intel Xeon X5450
  2TB in total

- On Request Resources
  
  36 nodes
  3TB

- Networking: Gigabit Ethernet

**IBBT**

- Permanent Resources
  
  8 nodes AMD Opteron 2212
  (1 server, 8 workers)

- On Request Resources
  
  92 nodes AMD Opteron 2212

- Networking: Gigabit Ethernet with multiple interfaces per node

**INRIA**

- Permanent Resources
  
  9 nodes Intel Xeon 5148 LV (1 server, 8 workers)

- On Request Resources
  
  162 nodes with 1800 cores

- Networking: Gigabit Ethernet

**USTUTT**

- Permanent Resources
  
  5 nodes Intel Xeon (4 core)
  250 GB

- On Request Resources
  
  16 nodes Intel Core i7 – 6100ES
  7.5 GB disk

- Networking: Gigabit Ethernet

**PSNC**

- Permanent Resources
  
  3 x Intel Xeon E5645 (2x6 cores)

- Networking: Gigabit Ethernet with multiple interfaces per node
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

- EPCC (Edinburgh)
- HP (Bristol)
- INRIA (Rennes)
- PSNC (Poznan)
- IBBT Virtual Wall (Ghent)
- HLRS (Stuttgart)

Scenario 1
(normal internet)

Scenario 2
(emulated network)

Scenario 3
(complex network)

Permanent (~350 cores / 30TB) & On-Request (theoretically 3000+ cores) infrastructures
Note: network links indicative only
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
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

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

First numerical results obtained for such a large sigma! From 2D to 3D
Thinking GRID for Electromagnetic Simulation of Oversized Structures (Electro)

Leaders
• Hervé Aubert (MINC-LAAS), Thierry Monteil (MRS-LAAS), Patricia Stolf (ASTRE-IRIT)

Design of sophisticated communication infrastructures
• Transmission 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 executions

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, data visualization; Metadatas management
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, . . . )