

Grid'5000: Running a Large Instrument for Parallel and Distributed Computing Experiments

F. Desprez

INRIA Grenoble Rhône-Alpes, Corse Team

Joint work with G. Antoniu, Y. Georgiou, D. Glesser, A. Lebre, L. Lefèvre, M. Liroz, D. Margery, L. Nussbaum, C. Perez, L. Pouilloux



F. Desprez - INRIA/EPFL workshop 2015

9/01/15 - 1

Agenda



- Experimental Computer Science
- Overview of GRID'5000
- GRID'5000 Experiments
- Related Platforms
- Conclusions and Open Challenges



F. Desprez - INRIA/EPFL workshop 2015

9/01/15 - 2

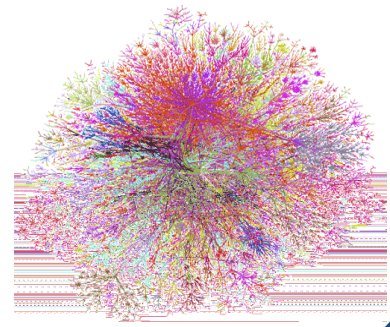
The reality of computer science

- Information
- Computers, networks, algorithms, programs, etc.

Studied objects (hardware, programs, data, protocols, algorithms, networks) are more and more complex

Modern infrastructures

- Processors have very nice features: caches, hyperthreading, multi-core
- Operating system impacts the performance (process scheduling, socket implementation, etc.)
- The runtime environment plays a role (MPICH \neq OPENMPI)
- Middleware have an impact
- Various parallel architectures that can be heterogeneous, hierarchical, distributed, dynamic



“Good experiments”

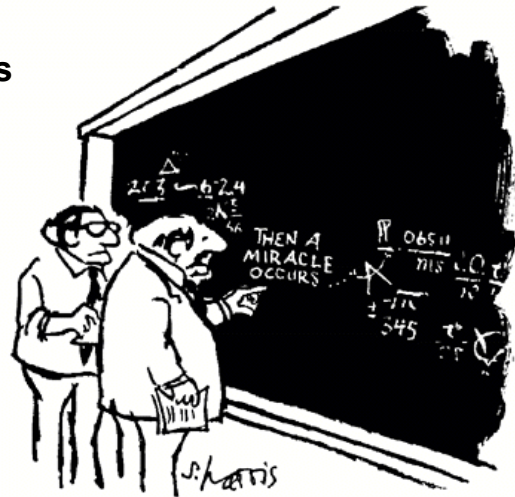


A **good experiment** should fulfill the following properties

- **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

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)



"I THINK YOU SHOULD BE MORE EXPLICIT
HERE IN STEP TWO."

**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

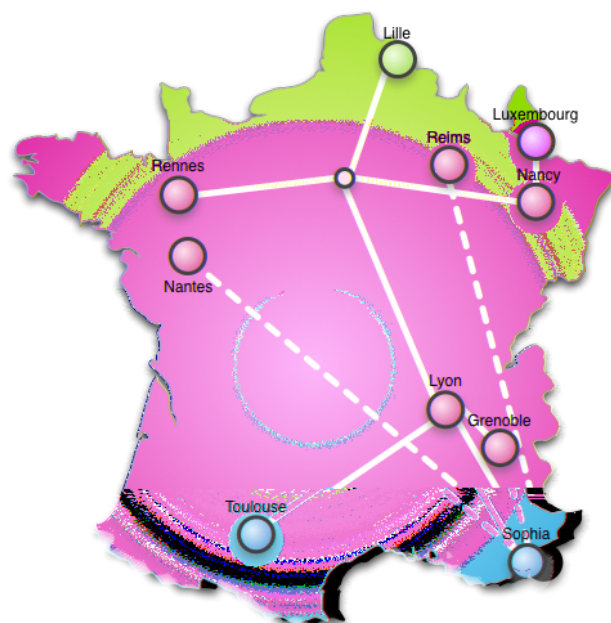
Several methodologies

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

- **Testbed for research on distributed systems**
 - Born from the observation that we 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): Hardware as a service
 - RlaaS : 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 (Sept. 2014 data)

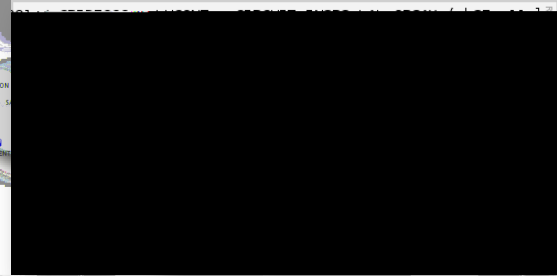
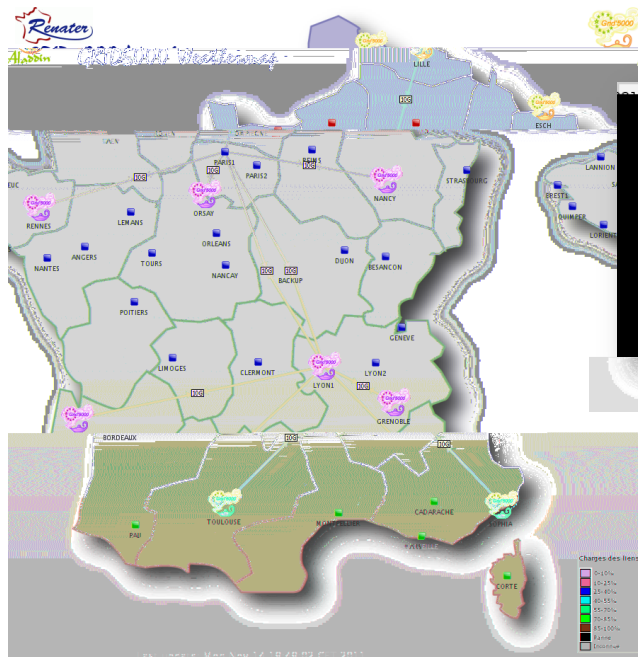
- 10 sites (1 outside France)
- 24 clusters
- 1006 nodes
- 8014 cores
- **Diverse technologies**
 - Intel (65%), AMD (35%)
 - CPUs from one to 12 cores
 - Ethernet 1G, 10G,
 - Infiniband {S, D, Q}DR
 - Two GPU clusters
 - 2 Xeon Phi
 - 2 data clusters (3-5 disks/node)
- More than **500 users** per year
- Hardware renewed regularly



Backbone Network



Dedicated 10 Gbps backbone provided by Renater (french NREN)



Work in progress

- Packet-level and flow level monitoring



Grid'5000 Mission



Support high quality, reproducible experiments on a distributed system testbed

Two areas of work

- **Improve trustworthiness**
 - Testbed description
 - Experiment description
 - Control of experimental conditions
 - Automate experiments
 - Monitoring & measurement
- **Improve scope and scale**
 - Handle large number of nodes
 - Automate experiments
 - Handle failures
 - Monitoring and measurements

Both goals raise similar challenges



Description of an Experiment over Grid'5000

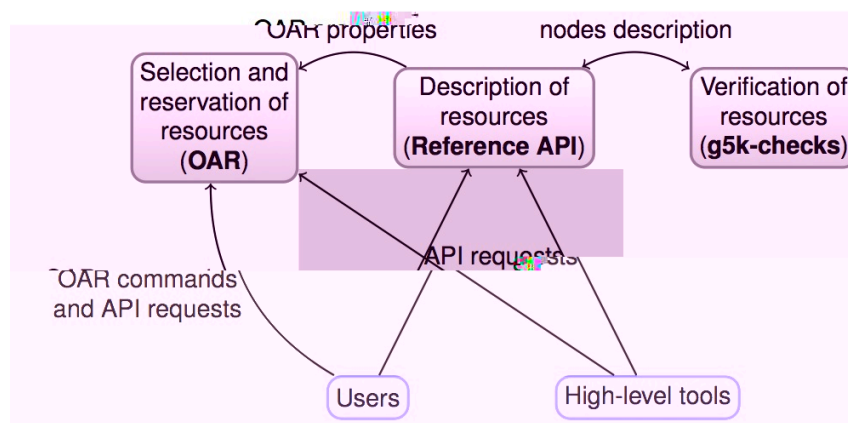


- Description and verification of the environment
- Reconfiguring the testbed to meet experimental needs
- Monitoring experiments, extracting and analyzing data
- Improving control and description of experiments

Description and verification of the environment



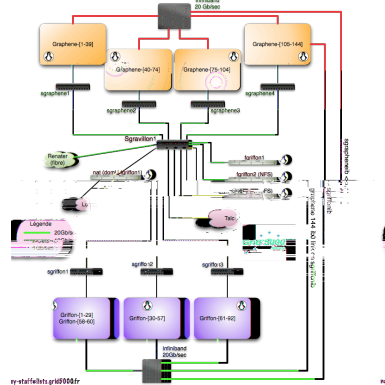
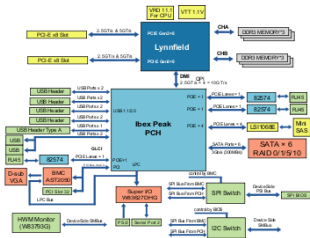
- **Typical needs**
 - How can I find suitable resources for my experiment?
 - How sure can I be that the actual resources will match their description?
 - What was the hard drive on the nodes I used six months ago?



Description and selection of resources

- **Describing resources understand results**

- Detailed description on the Grid'5000 wiki
- Machine-parsable format (JSON)
- Archived (State of testbed 6 months ago?)



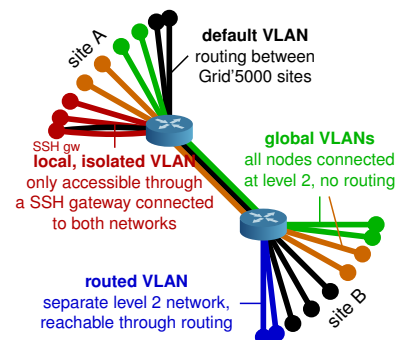
```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_li": null,
  "cache_lid": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "platform_type": "x86_64",
  "smt_size": 4,
  "smp_size": 1
},
"main_memory": {
  "ram_size": 17179869184,
  "virtual_size": null
},
"storage_devices": [
  {
    "model": "Hitachi HDS72103",
    "size": 29802322876.953,
    "driver": "ahci",
    "interface": "SATA II",
    "rev": "JPFO",
    "device": "sda"
  }
],
],
```

Reconfiguring the testbed



- **Typical needs**
 - How can I install \$SOFTWARE on my nodes?
 - How can I add \$PATCH to the kernel running on my nodes?
 - Can I run a custom MPI to test my fault tolerance work?
 - How can I experiment with that Cloud/Grid middleware?
- Likely answer on any production facility: you can't
 - Or: use virtual machines → experimental bias

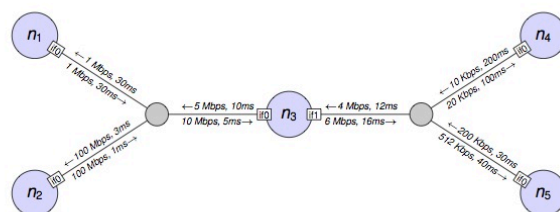
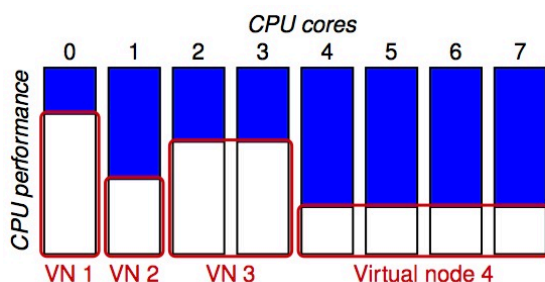
- **On Grid'5000**
 - Operating System reconfiguration with Kadeploy
 - Customize networking environment with KaVLAN



Changing experimental conditions



- **Reconfigure experimental conditions with Distem**
 - Introduce heterogeneity in an homogeneous cluster
 - Emulate complex network topologies



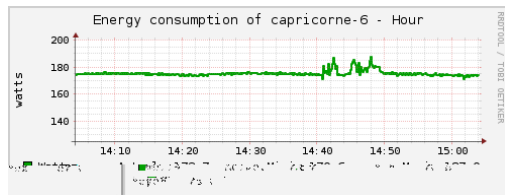
- **What else can we enable users to change?**
 - BIOS settings
 - Power management settings
 - CPU features (Hyperthreading, Turbo mode, etc.)
 - Cooling system: temperature in the machine room?



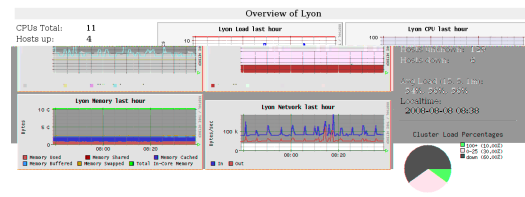
Monitoring experiments



Goal: enable users to understand what happens during their experiment



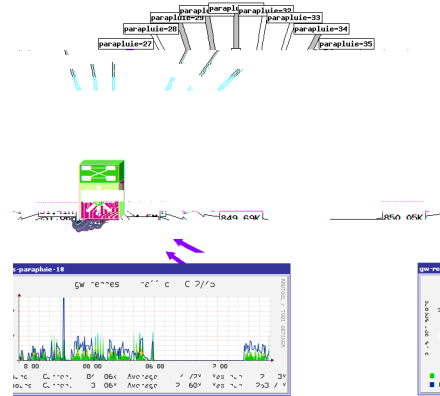
Power consumption



CPU – memory – disk



Network backbone



Internal networks



Controlling advanced CPU features



Modern processors have advanced options

- Hyperthreading (SMT): share execution resources of a core between 2 logical processors
- TurboBoost: stop some cores to increase frequency of others
- C-states: put cores in different sleep modes
- P-states (aka SpeedStep): Give each core a target performance

These advanced options are set at a very low level (BIOS or kernel options)

- Do they have an impact on the performance of middleware ?
 - Do publications document this level of experimental setting?
 - Does the community understand the possible impact ?
- How can this be controlled/measured on a shared infrastructure ?

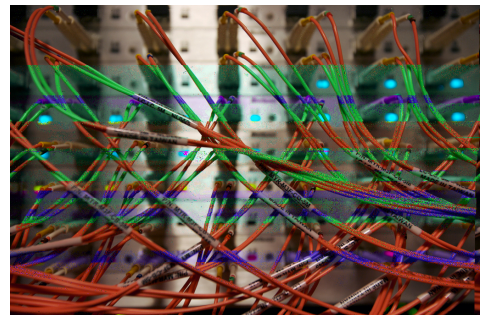


On Grid'5000

- Recent management cards can control low-level features (BIOS options to be short) have their options set through XML descriptions (tested on DELL's IDRAC7)
- Kadeploy inner workings support
 - Booting a deployment environment that can be used to apply changes to BIOS options
 - User control over cmdline options to kernels

Ongoing work

- Taking BIOS (or UEFI) descriptions as a new parameter at kadeploy3 level
- Restoring and documenting the standard state

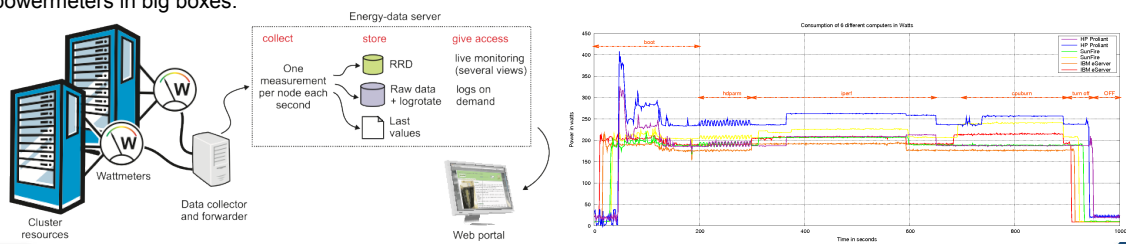


GRID'5000 EXPERIMENTS

Energy efficiency around Grid'5000



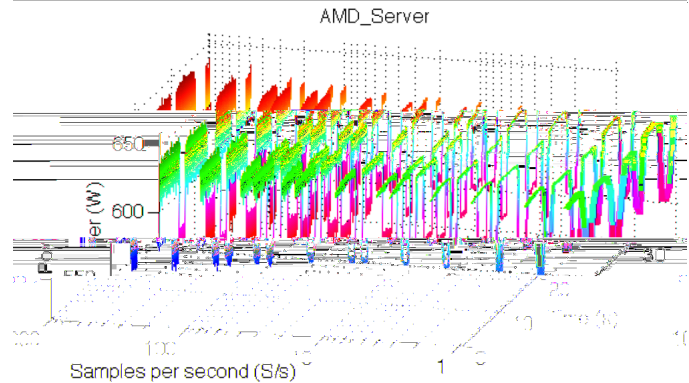
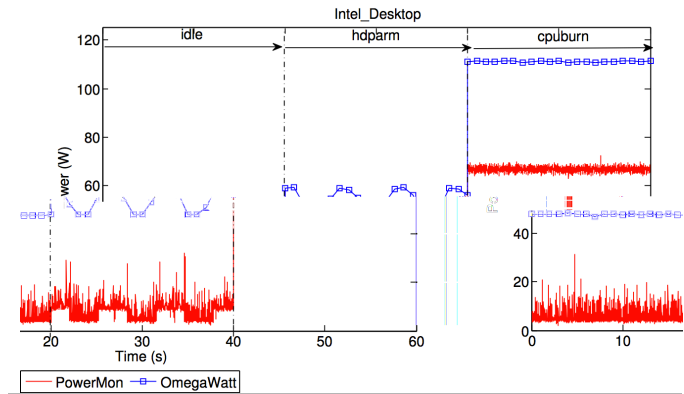
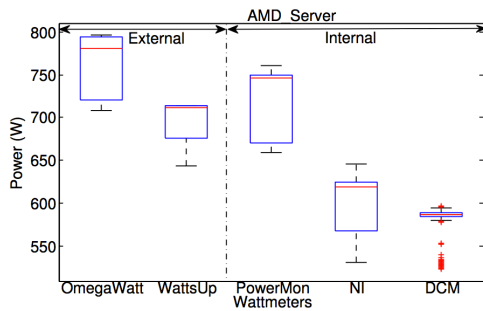
- IT – 2-5% of CO2 emissions / 10% electricity
- Green IT → reducing electrical consumption of IT equipments
- Future exascale/datacenters platforms → systems from 20 to 100MW
- How to build such systems and make them (more) energy sustainable/responsible ? Multi dimension approaches : hardware, software, usage
- **Several activities around energy management in Grid'5000 since 2007**
 - 2007: first energy efficiency considerations
 - 2008: Launch of Green activities...
 - 1st challenge: finding a First (real) powermeter
 - SME French company Omegawatt
 - Deployment of multiple powermeters: Lyon, Toulouse, Grenoble
 - 2010: Increasing scalability: Lyon Grid5000 site - a fully energy monitored site > 150 powermeters in big boxes.



Aggressive ON/OFF is not always

To understand energy measurements : take care of your wattmeters !

Frequency / precision



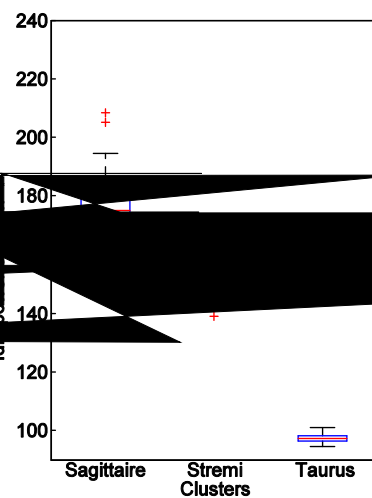
M. Diouri, M. Dolz, O. Glück, L. Lefevre, P. Alonso, S. Catalan, R. Mayo, E. Quintan-Orti. **Solving some Mysteries in Power Monitoring of Servers: Take Care of your Wattmeters!**, *EE-LSDS 2013 : Energy Efficiency in Large Scale Distributed Systems conference*, Vienna, Austria, April 22-24, 2013



Homogeneity (in energy consumption) does not exist !

- Depends on technology
- Same flops but not same flops per watt
- Idle / static cost
- CPU : main responsible
- Green scheduler designers must incorporate

this issue !



Mohammed el Mehdi Diouri, Olivier Gluck, Laurent Lefevre and Jean-Christophe Mignot. **"Your Cluster is not Power Homogeneous: Take Care when Designing Green Schedulers!"**, *IGCC2013 : International Green Computing Conference*, Arlington, USA, June 27-29,



Improving energy management with application expertise

- Considered services : resilience & data broadcasting
- 4 steps: Service analysis

The logo for India, featuring the word "India" in a red, cursive script font, positioned on a white rectangular background with rounded corners. This logo is part of a dark blue horizontal bar that spans the width of the page and has a curved end on the right side.

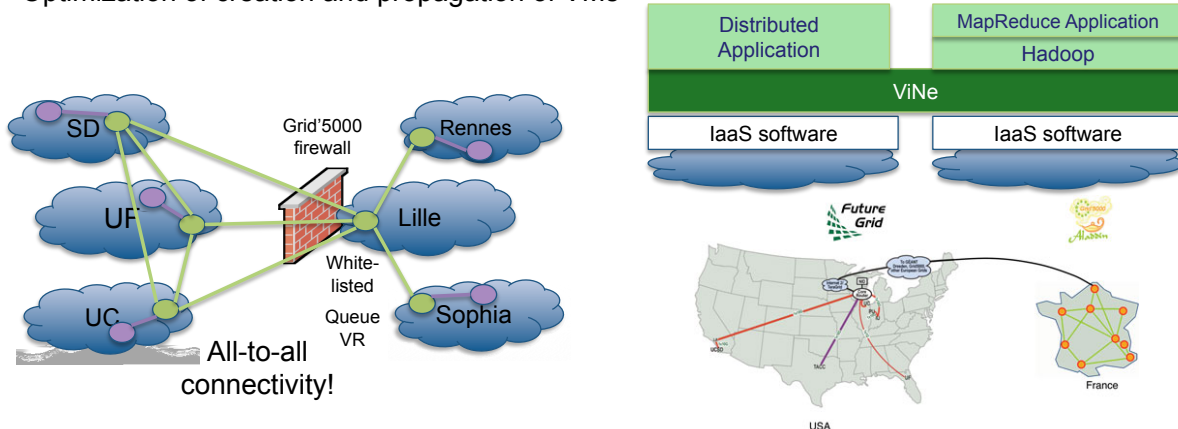
India

GRID'5000, Virtualization and Clouds: Sky computing use-case



Experiments between USA and France

- Nimbus (resource management, contextualization)/ViNe (connectivity)/Hadoop (task distribution, fault-tolerance, dynamicity)
- FutureGrid (3 sites) and Grid'5000 (3 sites) platforms
- Optimization of creation and propagation of VMs



Large-Scale Cloud Computing Research: Sky Computing on FutureGrid and Grid'5000, by Pierre Riteau, Maurício Tsugawa, Andréa Matsunaga, José Fortes and Kate Keahey, ERCIM News 83, Oct. 2010.

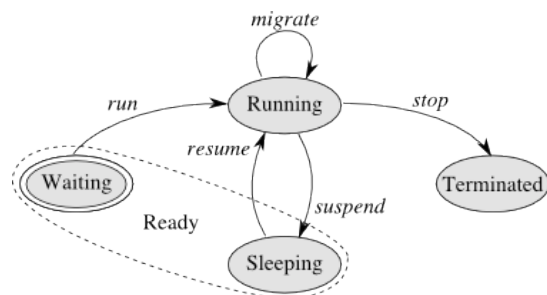


GRID'5000, Virtualization and Clouds: Dynamic VM placement use-case



2012 - Investigate issues related to preemptive scheduling of VMs

- Can a system handle VMs across a distributed infrastructure like OSES manipulate processes on local nodes ?
- Several proposals in the literature, but
 - Few real experiments (simulation based results)
 - Scalability is usually a concern
- Can we perform several migrations between several nodes at the same time ? What is the amount of time, the impact on the VMs/on the network ?

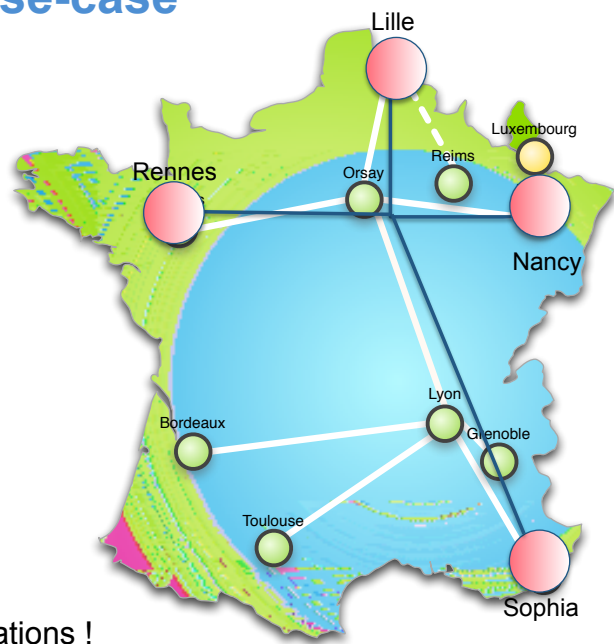


GRID'5000, Virtualization and Clouds: Dynamic VM placement use-case



Deploy 10240 VMs upon 512 PMs

- Prepare the experiment
 - Book resources
 - 512 PMs with Hard. Virtualization
 - A global VLAN
 - A /18 for IP ranges
 - Deploy KVM images and put PMs in the global VLAN
- Launch/Configure VMs
 - A dedicated script leveraging Taktuk utility to interact with each PM
 - G5K-subnet to get booked IPs and assign them to VMs
- Start the experiment and make publications !




F. Quesnel, D. Balouek, and A. Lebre. **Deploying and Scheduling Thousands of Virtual Machines on Hundreds of Nodes Distributed Geographically**. In IEEE International Scalable Computing Challenge (SCALE 2013) (colocated with CCGRID 2013), Netherlands, May 2013



GRID'5000, Virtualization and Clouds



- More than 198 publications
 - Three times finalist of the IEEE Scale challenge (2nd prize winner in 2013)
- 
- Tomorrow - Virtualization of network functions (Software Defined Network)
 - Go one step ahead of KaVLAN to guarantee for instance bandwidth expectations
HiperCal, Emulab approaches
Network virtualisation is performed by daemons running on dedicated nodes
⇒ Do not bring additional capabilities (close to the Distem project)
 - By reconfiguring routers/switches on demand
⇒ Required specific devices (OpenFlow compliant)
 - Which features should be exported to the end-users ?
 - Are there any security concerns ?



Riplay: A Tool to Replay HPC Workloads



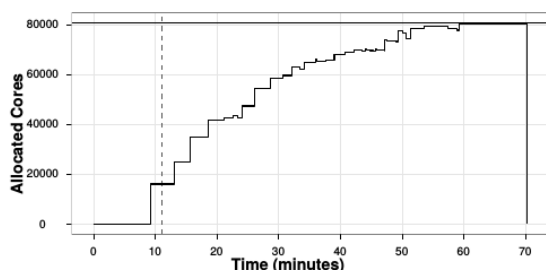
- **RJMS : Ressource and Job Management System**
 - It manages resources and schedule jobs on High-Performance Clusters
 - Most famous ones : Maui/Moab, OAR, PBS, SLURM
- **Riplay**
 - Replay traces on a real RJMS in an emulated environment
 - 2 RJMS supported (OAR and SLURM)
 - Jobs replaced by *sleep commands*
 - Can replay a full or an interval of a workload
- **On Grid'5000**
 - 630 emulated cores need 1 physical core to run
- **Curie (rank 26th on last Top500, 80640 cores)**
 - Curie's RJMS can be ran on 128 Grid'5000 cores



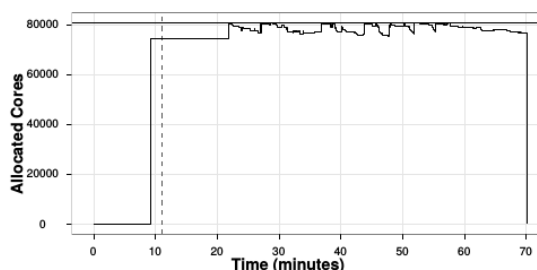
Riplay: A Tool to Replay HPC Workloads



- Test RJMS scalability
 - Without the need of the actual cluster.
 - Test a huge cluster fully loaded on a RJMS in minutes.



OAR before optimizations



OAR after optimizations

Large Scale Experimentation Methodology for Resource and Job Management Systems on HPC Clusters, Joseph Emeras, David Glesser, Yiannis Georgiou and Olivier Richard



HPC Component Model: From Grid'5000 to Curie SuperComputer



Issues

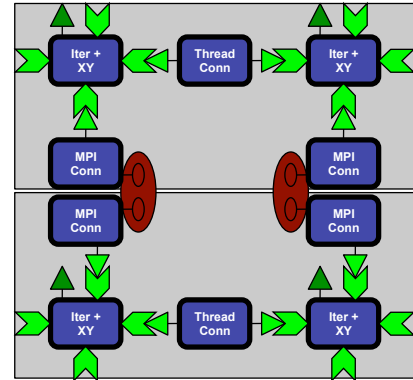
- Improve code re-use of HPC applications
- Simplify application portability

Objective

- Validate L2C, a low level HPC component model
 - Jacobi¹ and 3D FFT² kernels

Roadmap

- Low scale validation on Grid'5000
 - Overhead wrt "classical" HPC models (Threads, MPI)
- Study of the impact of various node architecture
- Large scale validation on Curie
 - Up to 2000 nodes, 8192 cores



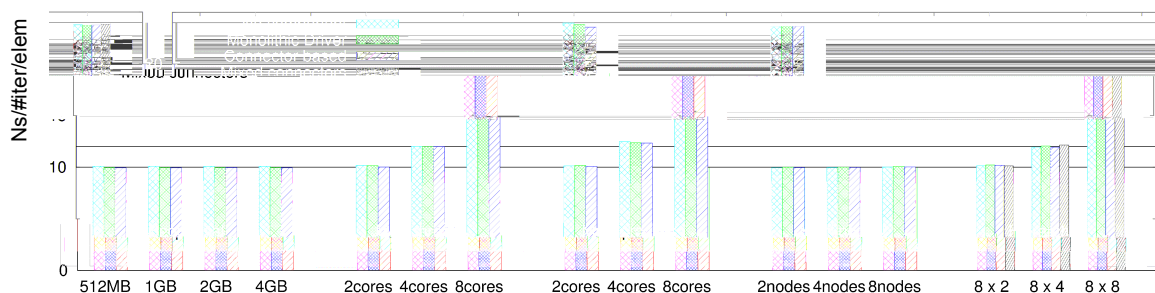
1: J.Bigot, Z. Hou, C. Pérez, V. Pichon. **A Low Level Component Model Easing Performance Portability of HPC Applications**. Computing, page 1–16, 2013, Springer Vienna
 2: On going work.



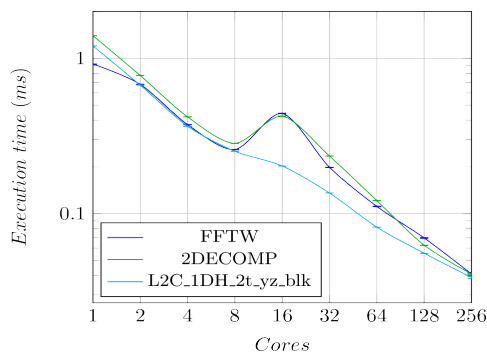
HPC Component Model: From Grid'5000 to Curie SuperComputer



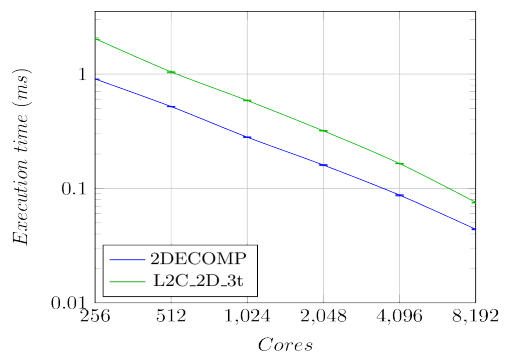
Jacobi: No L2C overhead in any version



3D FFT



256³ FFT, 1D decomp., Edelp+Genepi Cluster (G5K)



1024³ FFT, 2D decomp., Curie (Thin node)

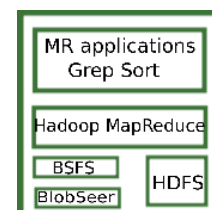
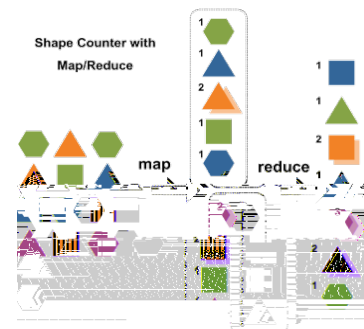


- Growing demand from our institutes and users
- Grid'5000 has to evolve to be able to cope with such experiments
- **Current status**
 - home quotas has grown from 2GB to 25GB by default (and can be extended up to 200 GB)
 - For larger datasets, users should leverage storage5K (persistent dataset imported into Grid'5000 between experiments)
 - DFS5K, deploy on demand CephFS
- **What's next**
 - Current challenge: upload datasets (from internet to G5K) / deployment of data from the archive storage system to the working nodes
 - New storage devices: SSD / NVRAM / ...
 - New booking approaches
 - Allow end-users to book node partitions for longer period than the usual reservations

Scalable Map-Reduce Processing

Goal: High-performance Map-Reduce processing through concurrency-optimized data processing

- **Some results**
 - Versioning-based concurrency management for increased data throughput (BlobSeer approach)
 - Efficient intermediate data storage in pipelines
 - Substantial improvements with respect to Hadoop
 - Application to efficient VM deployment
- **Intensive, long-run experiments done on Grid'5000**
 - Up to 300 nodes/500 cores
 - Plans: validation within the IBM environment with IBM MapReduce Benchmarks



- ANR Project Map-Reduce (ARPEGE, 2010-2014)
- Partners: Inria (teams : KerData - leader, AVALON, Grand Large), Argonne National Lab, UIUC, JLPC, IBM, IBCP

Damaris: A Middleware-Level Approach to I/O on Multicore HPC Systems



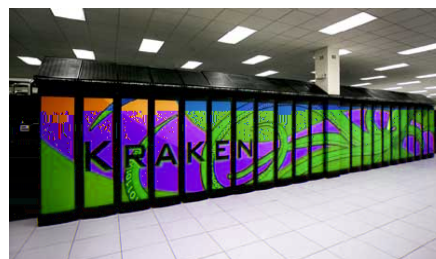
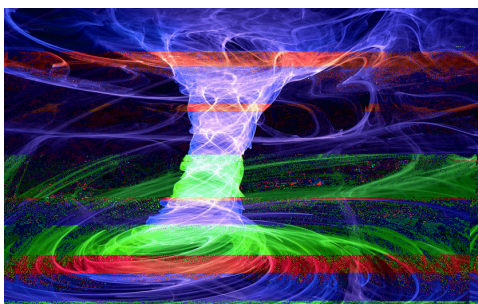
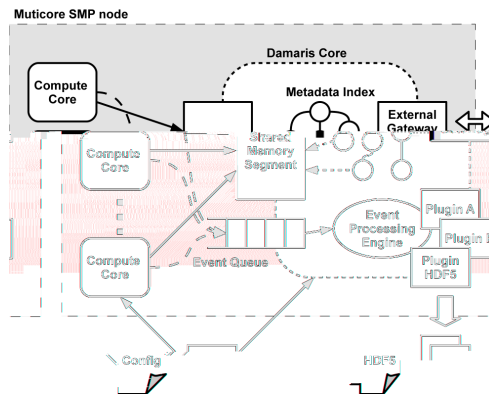
Idea : one dedicated I/O core per multicore node

Originality : shared memory, asynchronous processing

Implementation: software library

Applications: climate simulations (Blue Waters)

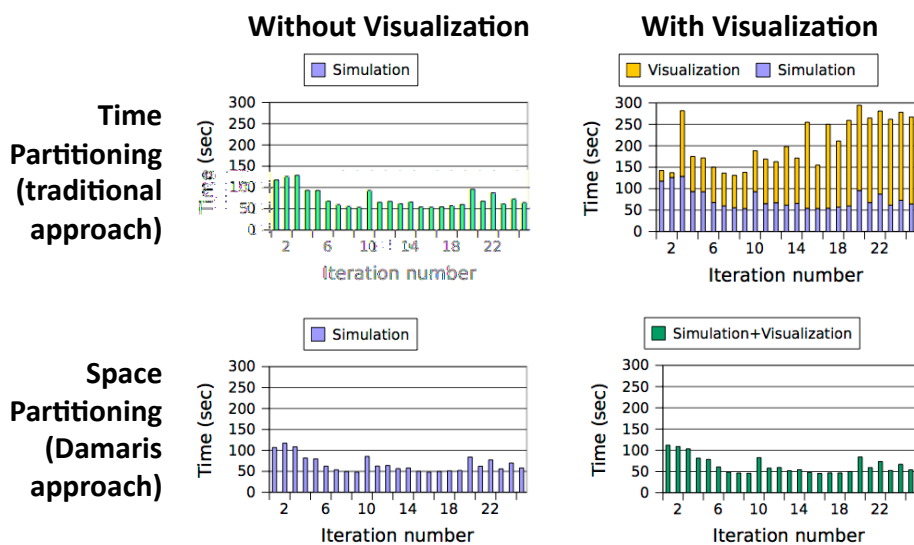
Preliminary experiments on Grid'5000



<http://damaris.gforge.inria.fr/>



Damaris: Leveraging dedicated cores for in situ visualization



Experiments on Grid'5000 with Nek5000 (912 cores)

Using Damaris completely hides the performance impact of in situ visualization



Related Platforms



- **PlanetLab**
 - 1074 nodes over 496 sites world-wide, slices allocation: virtual machines.
 - Designed for experiments Internet-wide: new protocols for Internet, overlay networks (file-sharing, routing algorithm, multi-cast, ...)
- **Emulab**
 - Network emulation testbed. Mono-site, mono-cluster. Emulation. Integrated approach
- **Open Cloud**
 - 480 cores distributed in four locations, interoperability across clouds using open API
- **Open Cirrus**
 - Federation of heterogeneous data centers, test-bed for cloud computing
- **DAS-1..4-5**
 - Federation of 4-6 cluster in Netherland, ~200 nodes, specific target experiment for each generation
- **NECTAR**
 - Federated Australian Research Cloud over 8 sites
- **Futuregrid (ended Sept. 30th)**
 - 4 years project within XCEDE with similar approach as Grid'5000
 - Federation of data centers, bare hardware reconfiguration
- **2 new projects awarded by NSF last August**
 - **Chameleon**: a large-scale, reconfigurable experimental environment for cloud research, co-located at the University of Chicago and The University of Texas at Austin
 - **CloudLab**: a large-scale distributed infrastructure based at the University of Utah, Clemson University and the University of Wisconsin

Conclusion and Open Challenges

- Computer-Science is also an experimental science
- There are different and complementary approaches for doing experiments in computer-science
-

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!

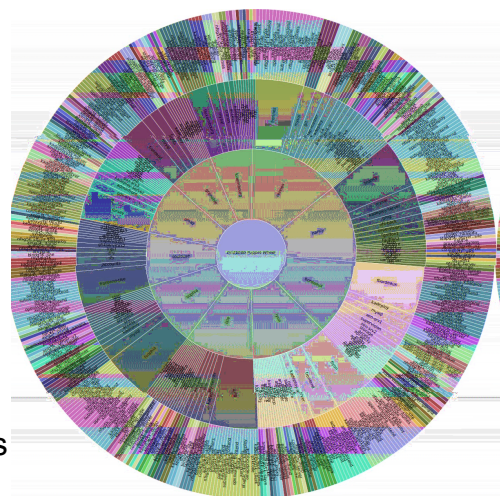
From our experience, experimental platforms should feature

- 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, faults)

Conclusion and Open Challenges, cont



- Testbeds optimized 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
- Paving the way to Open Science of HPC and Cloud – long term goals
 - Fully automated execution of experiments
 - Automated tracking + archiving of experiments and associated data



QUESTIONS ?

Special thanks to
G. Antoniu, Y. Georgiou, D.
Glesser, A. Lebre, L. Lefèvre, M.
Liroz, D. Margery, L. Nussbaum,
C. Perez, L. Pouilloux
and the Grid'5000 technical team

www.grid5000.fr

