EDGE and Grid'5000 Day at LORIA

Martin Quinson <martin.quinson@loria.fr>

LORIA

16 mai 2011

Welcome to this EDGE day

Thanks for your participation

- Thanks for participating to this workshop
- Another similar day is expected in Fall

Presentation Objectives and Agenda

- Present some motivation for the instrument (why)
- Explain the scientific objective and their implementation (what)
- Highlight some usage direction of used technologies (how)

Grid'5000 Day @LORIA

- Motivation: why Grid'5000
- Grid'5000 Design
- State of Grid'5000
- Some Results
- Conclusion

A bit of Epistemology

unavoidable to speak of experiment, as we will

Usual Classification of Sciences

- Natural science: knowledge accumulated through experiences
- Speculative science: Truth determined through analyze of its objects
- Lot of discussion about this *A posteriori / A priori* distingo (Kant. . .)

Computers mandates to review this all

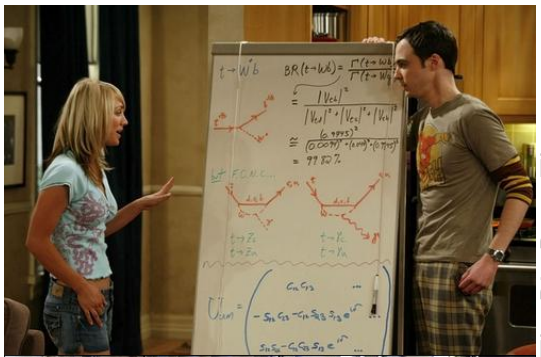
- 100th decimal of $\pi \rightsquigarrow$ analytical (no experiment) but *a posteriori*
- Philosopher usually trust their reasoning over their senses
  - Computer constitute an external extension to your reasoning
  - Prosthesis computing faster than brain ($\approx$ Galileo's telescope for eyes)
- Computers are usually not considered as part of the nature:
  - We built it entirely and know it all
  - Contains only what we've put in it
  - There is no need to study it as per

How does Science work?

Proposed theories remain valid until proved false (or better proposed)

Classical approaches in science and engineering

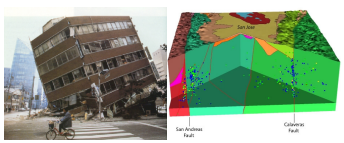
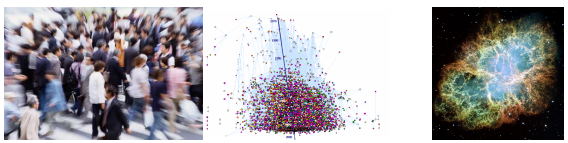
1. Theoretical work: equations on a board
2. Experimental study on an scientific instrument


The Big Bang Theory

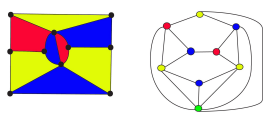

Large Hardron Collider

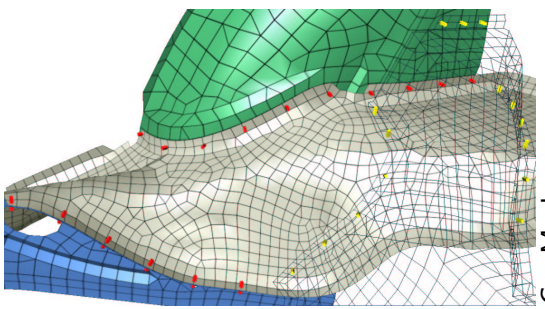
That's not always desirable (or even possible)

- Some phenomenons are intractable theoretically
- Experiments too expensive, difficult, slow, dangerous

The third scientific way: *Computational Science*

3. Use computers (*in silico* study)
   - Modeling / Simulation of the phenomenon
   - Data Mining to find interesting subject of studies
   - Automated theorem proving 


Car Mesh

Computers, Science and Computer Science

> *Computer Science is not more related to computers than Astronomy to telescopes.* – *Dijkstra*

Computer Science

- Fundamental concepts: algorithm; information; machine; language
- The technical aspect is not the central concept of our science...
- ...but our science allow us to be the experts of the computer systems

Computer Systems considered here

- Objects: (Distributed) Software Systems
- Metrics: Scalability, Robustness, Security, Fairness, Resource consumption, ...

Computers widely used in Science (one of the root motivation to computers)

Q: How can we use computers to assist our Computer Scientist work?

A: through Grid'5000 of course

We, as computer scientists, don't need that, you say?

(some) Challenging Computer Systems

Cloud Computing

- Large infrastructures underlying commercial Internet (eBay, Amazon, Google)
- Main issue: Optimize costs; Keep up with the load (flash crowds)

High Performance Computing and Exascale

- Have the world's biggest computer, to lead CS and IT world's research
- Main issues: do the biggest possible numerical simulations [justify investment]

Grid Computing

- Infrastructure for *computational science:* lot of sequential simulation jobs
- Main issues: compatibility, virtual organizations (trust and accountability mgmt)

Peer-to-peer Systems (P2P)

- Exploit resources at network edges (storage, CPU, human presence)
- Main issues: Intermittent connectivity (churn); Network locality; Anonymity

Systems already in use, but characteristics hard to assess

- Performance: everyone want to maximize it, but definition differs
- Correction: absence of crash, race conditions, deadlocks and other defects

Studying Computer Systems

Computers are eminently artificial artifacts

- Humans built them completely, they contain only what we've put in it

$\Rightarrow$ Theoretical (maths) methodology to study it

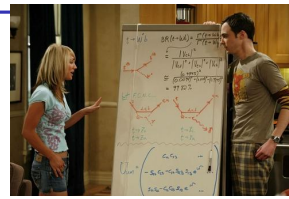
Computer systems complexity getting tremendous

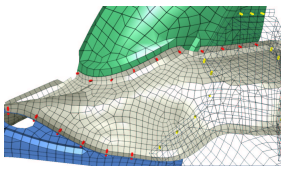
- Heterogeneity of components (hosts, links)
  - Quantitative: CPU clock, link bandwidth and latency
  - Qualitative: ethernet *vs* myrinet *vs* quadrics; amd64 *vs* ARM *vs* GPU
- Dynamicity
  - Quantitative: resource sharing $\rightsquigarrow$ availability variation
  - Qualitative: resource come and go (churn, failures)
- Complexity
  - Hierarchical systems: grids of clusters of multi-processors being multi-cores
  - Deep software stacks: Middleware, Web Services, mashups
  - Multi-hop nets, high latencies; Interference comput./comm. (disk/memory)

Computer Systems as Natural Objects

- The complexity is so high that we cannot understand them fully anymore
- Frankenstein effect, but allows to use computers to understand computers

Assessing Distributed Applications Performance





Classical Scientific Pillars Apply

- Theoretical Approach: Mathematical study of algorithms
- Experimental Science: Study applications on scientific instrument
- Computational Science: Simulation of a system model

Performance Study $\rightsquigarrow$ Experimentation

- Theory still mandatory, but everything's NP-hard
- Experimental Facilities: Real applications on Real platform *(in vivo)*
- Emulation: Real applications on Synthetic platforms *(in vitro)*
- Simulation: Prototypes of applications on system's Models *(in silico)*

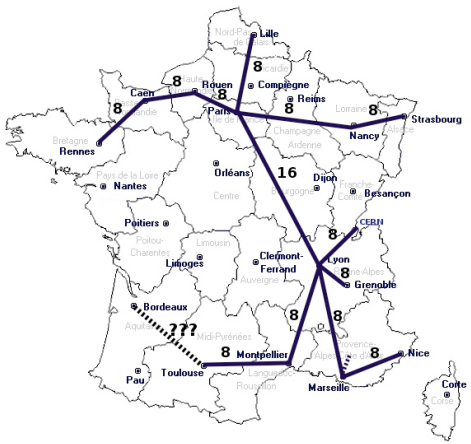
	Experimental Facilities	Emulation	Simulation
Experimental Bias	☺☺	☺	☹
Experimental Control	☹☹	☺	☺☺
Ease of Use	☹	☹☹	☺☺

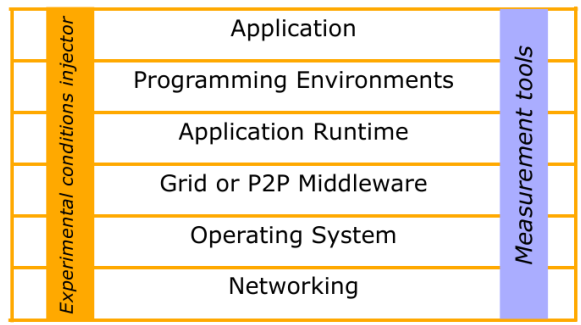
In vivo approach: Direct Experimentation

- Principle: Real applications, Real environment (with reduced external noise)
- Challenges: Not trivial nor immediate. Experimental control? Reproducibility?

Grid'5000 project: world leading scientific instrument for dist. apps

- Instrument for research in computer science (*deploy* your own OS)
- 9 sites, 1500 nodes (3000 cpus, 4000 cores); dedicated 10Gb links





Other existing platforms

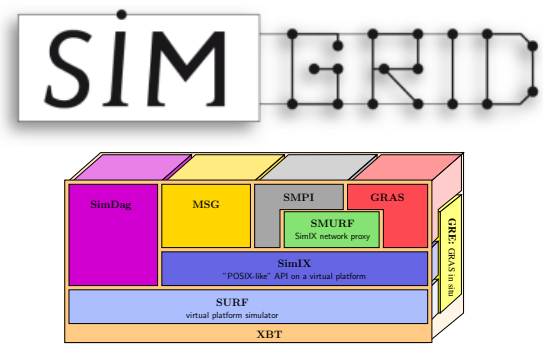
- PlanetLab: No experimental control ⇒ no reproducibility
- Production Platforms (EGEE/EGI): must use provided middleware
- FutureGrid: US experimental platform loosely inspired from Grid'5000

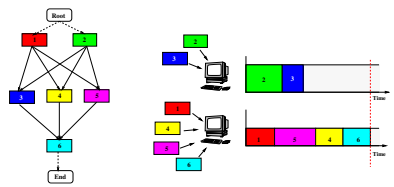
In silico approach: Simulated Experiments

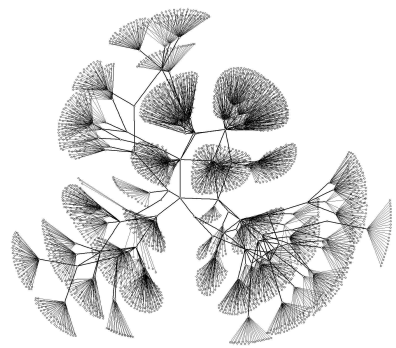
- Principle: Prototypes of applications, models of platforms
- Challenges: Get realistic results (experimental bias)

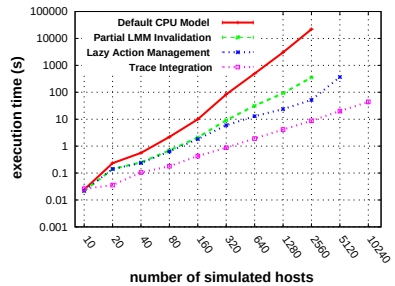
SimGrid: generic simulation framework for distributed applications

- Scalable (time and memory), modular, portable. +70 publications.
- Collaboration Loria / Inria Rhône-Alpes / CCIN2P3 / U. Hawaii









Other existing tools

- *Large* amount of existing simulator for distributed platforms:
  GridSim, ChicSim, GES; P2PSim, PlanetSim, PeerSim; ns-2, GTNetS.
- Few are really usable: Diffusion, Software Quality Assurance, Long-term availability
- No other study the validity, the induced experimental bias

Emulation as an Experimental Methodology

Execute your application in a perfectly controlled environment

- Real platforms are not controllable, so how to achieve this?
- Let's look at what engineers do in other fields

Emulation as an Experimental Methodology

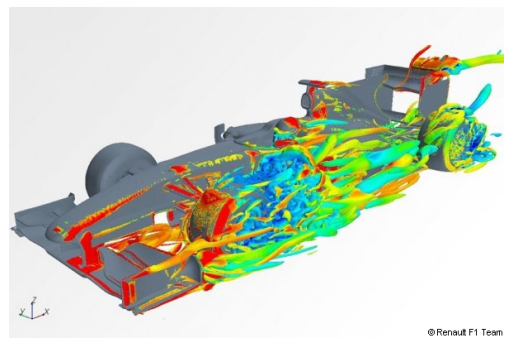
Execute your application in a perfectly controlled environment

- Real platforms are not controllable, so how to achieve this?
- Let's look at what engineers do in other fields

When you want to build a race car...

...adapted to wet tracks

...in a dry country ...



...you can simulate it.

Courtesy of Lucas Nussbaum

But then, you have

- To assess models
- Technical burden
- No real car

Why don't you... just control the climate?

Emulation as an Experimental Methodology

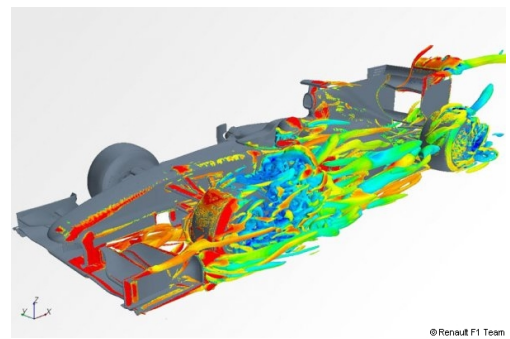
Execute your application in a perfectly controlled environment

- Real platforms are not controllable, so how to achieve this?
- Let's look at what engineers do in other fields

When you want to build a race car...

...adapted to wet tracks

...in a dry country ...



...you can simulate it.

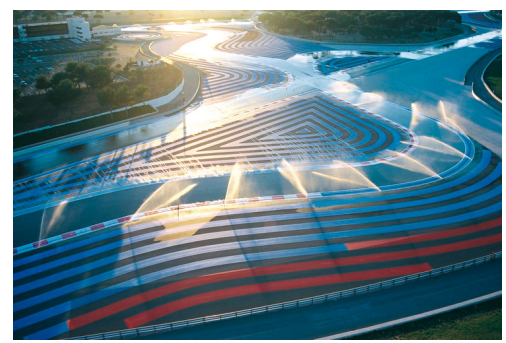
Courtesy of Lucas Nussbaum

But then, you have

- To assess models
- Technical burden
- No real car

Why don't you...

just control the climate?



That's Emulation

Emulating Distributed Systems

Such *Emulation through Degradation* is quite classical

- Degrade the performance of the host platform (CPU burners, Network capping)
- WrekAvoc (LORIA – Lucas Nussbaum *et Al.*) works this way

☺ Real application, controlled environment

☹ Complex technologies, heavy infrastructures, tedious tool assessment

☹ Reeeeally hard to emulate faster/larger platforms than host platform

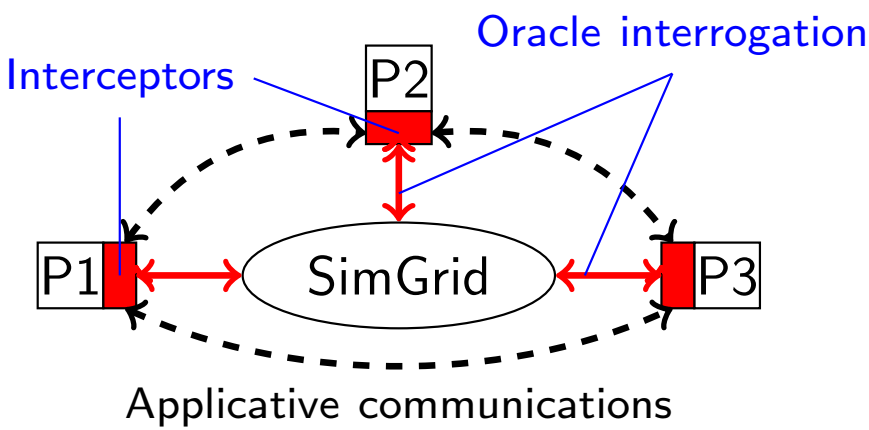
Emulating Distributed Systems

Such *Emulation through Degradation* is quite classical

- Degrade the performance of the host platform (CPU burners, Network capping)
- WrekAvoc (LORIA – Lucas Nussbaum *et Al.*) works this way
- ☺ Real application, controlled environment
- ☹ Complex technologies, heavy infrastructures, tedious tool assessment
- ☹ Reeeeally hard to emulate faster/larger platforms than host platform

Another approach: Emulation through Simulation

- Intercept what the application does, Compute answer by simulation, Apply it
- This is what you do to assess the breaking system of your car



This is SimTerpose

Toward the Best Methodology

	Experimental Facilities	Emulation	Simulation
Experimental Bias	☺☺	☺	☹
Experimental Control	☹☹	☺	☺☺
Ease of Use	☹	☹☹	☺☺

Best approach: use them all through the lifecycle of your ideas

- None of these approaches is sufficient, each have advantages/drawbacks

	Whiteboard	Simulation	Experimental Facility	Emulation	Production Platforms
Idea	☺☺				
Algorithm	☺	☺☺			
Prototype		☺	☺☺		
Application			☺☺	☺	
Product					☺

One could determine the differing ages of a science by the technic of its measurement instruments.

– Gaston Bachelard, La formation de l'esprit scientifique.

Grid'5000 Day @LORIA

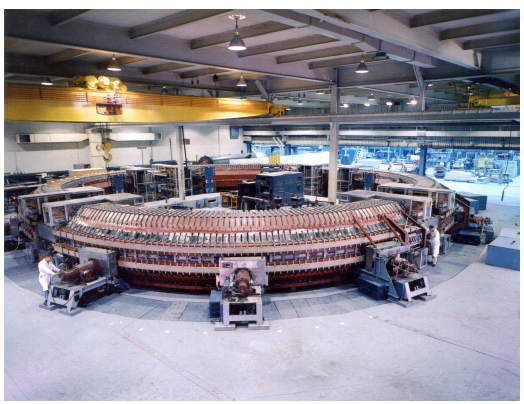
- Motivation: why Grid'5000
- Grid'5000 Design
- State of Grid'5000
- Some Results
- Conclusion

The Grid'5000 Project

Build a National Experimental Facility (and mutualize resources)

- 9 sites distributed in France (country-wide plaform)
- Each site hosts between 100 and 1000 cores
- Sites are interconnected by RENATER (french NREN)
- Robust and reproducible experiments on distributed apps *in real settings*

Original Vision: Analogy with Physics Instruments



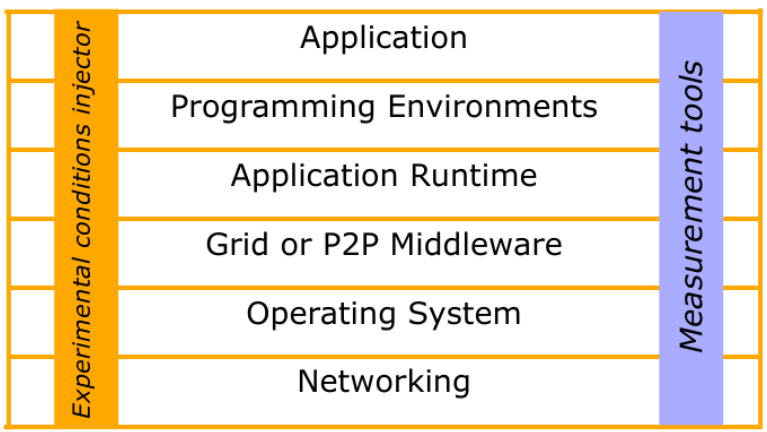
The Cosmotron, 1953

- Experiment isolation
- Capability to reproduce experimental conditions
- High degree of flexibility
- Strong control of experiment preparation and running
- Precise measurement methodology
- Deep on-line monitoring

Typical Experiments Expected on Grid'5000

Experiment in every layer of software stack

- Networks: next generation long distance protocols, HPC/SAN
- Systems: HPC and distributed Operating Systems
- Middleware/Runtime/Apps: scalability, robustness, efficiency, correctness



Highly Reconfigurable Platform

- No limit! You deploy your own OS
- Deploying G5K within G5K [soon] possible
- Almost no noise onto experiments
  (but little experimental control per se,
  unless you use an emulator in addition)

Grid'5000 Key Design Decisions

Build experimental facility bottom-up

- KISS while providing robust platform for large variety of experiments
- Offer low level software functionalities first, and build upon it
- Allow experiments in every layer of the software stack

Experimental Reproducibility is our graal

- Users re-run experiments on the same hardware
- Experiments are as isolated as possible

Security: keep Internet safe from Grid'5000 (and also the contrary ;)

- From Internet: only some specific gateways
- To Internet: no connexion – some rare sites white-listed (Linux distributions)

Experimental Model

Grid'5000 as a IaaS Cloud

- Reserve resources $\rightsquigarrow$ OAR
- (Deploy your own operating system); boot the nodes $\rightsquigarrow$ KaDeploy
- Enjoy

The Grid'5000 Software Ecosystem

- TakTuk: control of a large number of nodes
- KaVlan: network isolation of experiment through VLAN manipulation
- Kapower: shut down and start up machines remotely
- WrekAvoc: emulation through degradation
- FAIL: fault injection
- Kastafior: data broadcasting
- Kaspied: usage spying
- Restful interfaces (status, reservation, reconfigure, measures)
- ScaLab: framework for experiment industrialization
- Dynamic Energy Consumption Monitoring

Grid'5000 Day @LORIA

- Motivation: why Grid'5000
- Grid'5000 Design
- State of Grid'5000
- Some Results
- Conclusion

Reserving nodes on Grid'5000: OAR

Resources of each clusters are managed through OAR

OAR is a classical *batch scheduler*

- Several queues, with differing priorities
- Reservations in advance, back-filling

Grid Mode for reservations spanned on several sites

- Requests are submitted to local schedulers
- If a request fails, the others are canceled

Reconfiguring nodes

- A specific queue allows to reconfigure the nodes (reinstall an OS)
- The user gains the root rights on this machine (they are revoked afterward)

Using idle machines: the best effort mode

- Resource usage limited to ensure that everyone gets what he needs
  https://www.grid5000.fr/mediawiki/index.php/Grid5000:UserCharter
- Best effort: get all free resources (+be kicked when other reclaim resources)

Reconfiguration on Grid'5000: Kadeploy

Kadeploy is a tool to deploy disk images

Deployment

- Possible only on machines reserved through the `deploy` queue
- Reboot on a specific kernel to prepare the machines
- User's OS (kernel+disk) copied in pipeline on machines
- Post-installation scripts (hostname, SSH keys, etc)
- Reboot on user's OS

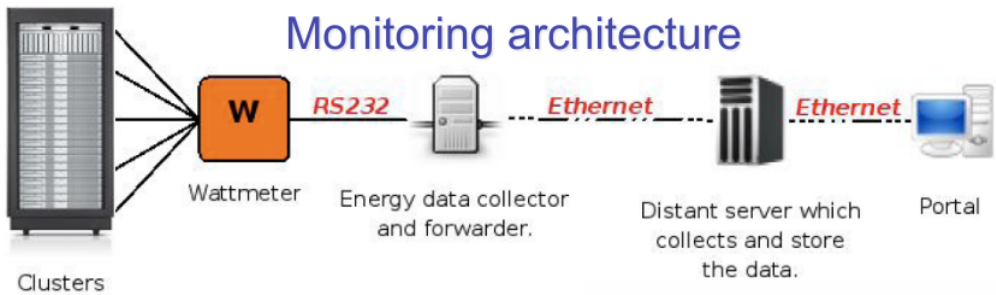
Preparing Disk Images

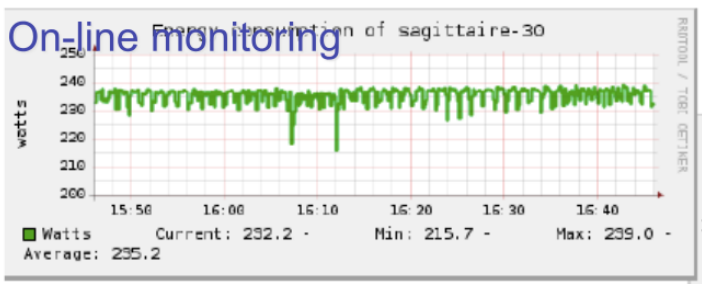
1. Deploy an existing image
2. Tune and Adapt it
3. Save it (it's stored in a DB and copied on servers)

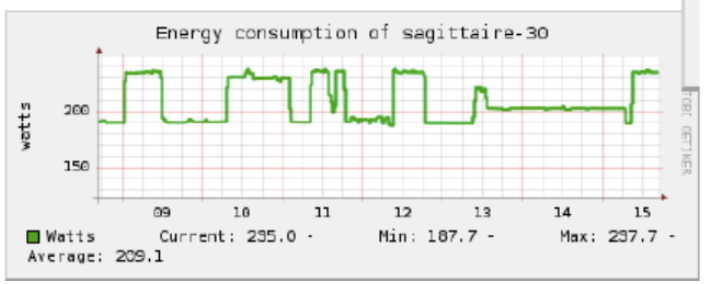
Dynamic Energy Consumption Monitoring

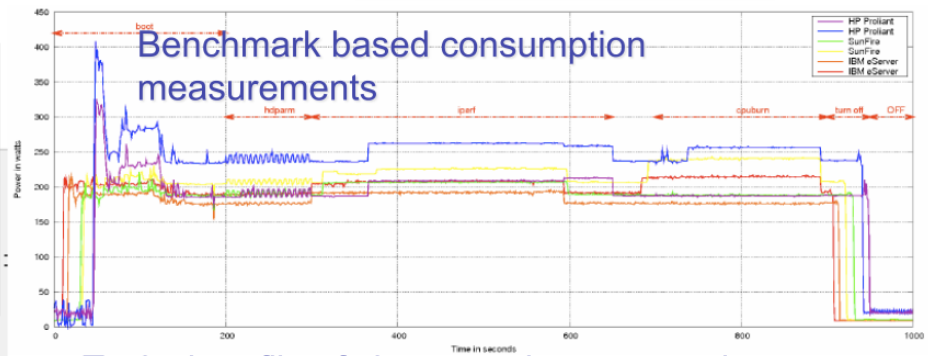
Controllable and external energy sensors

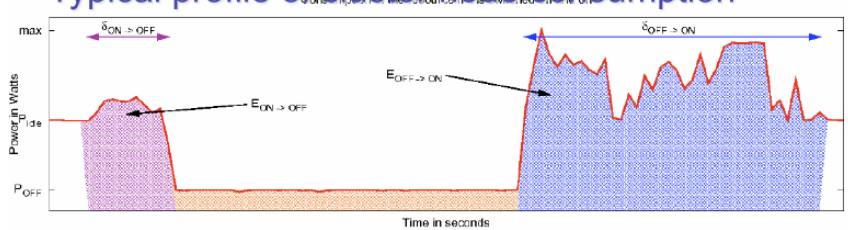
- On-line energy consumption measurements
- Allows to develop energy saving algorithms



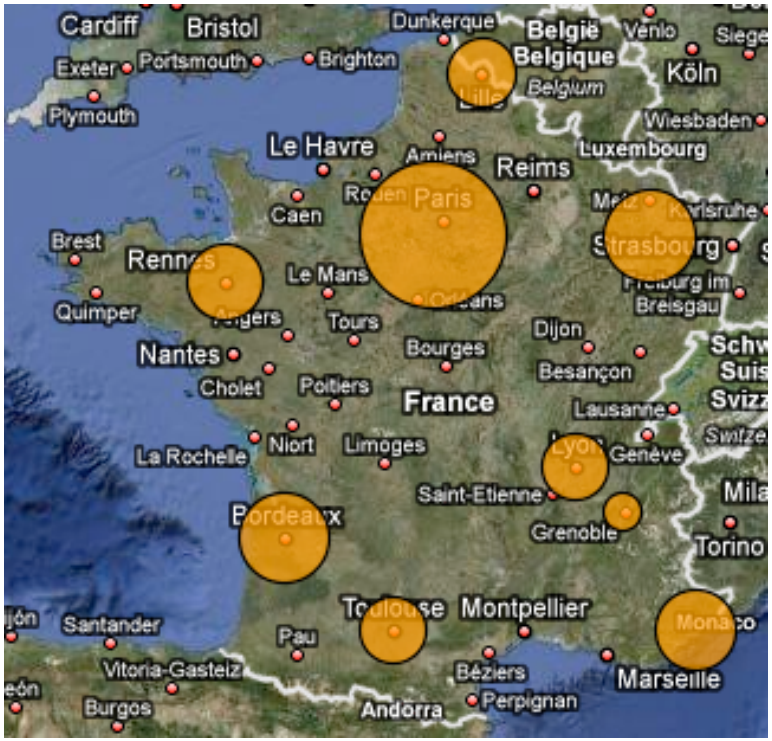




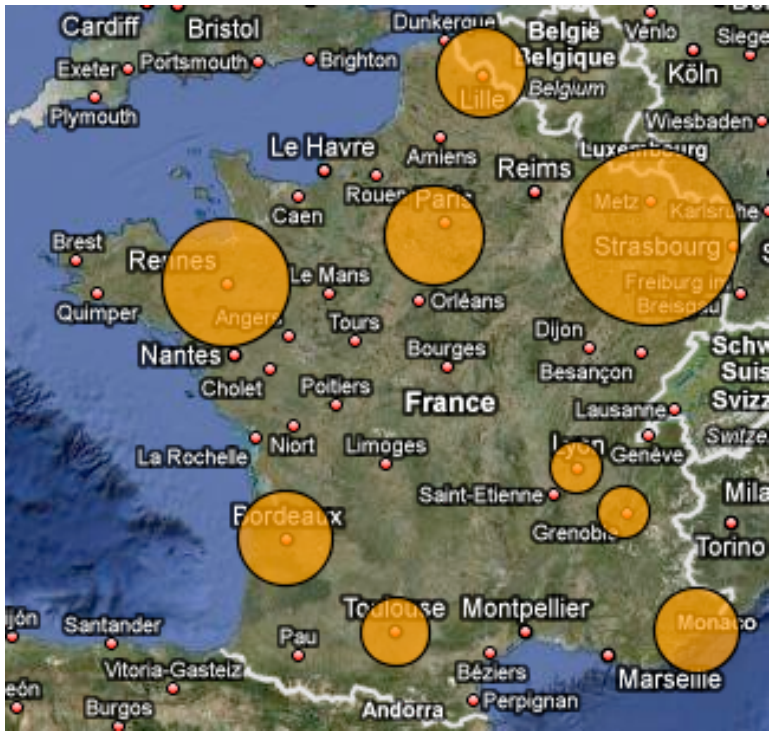




Grid'5000 Map



1500 machines

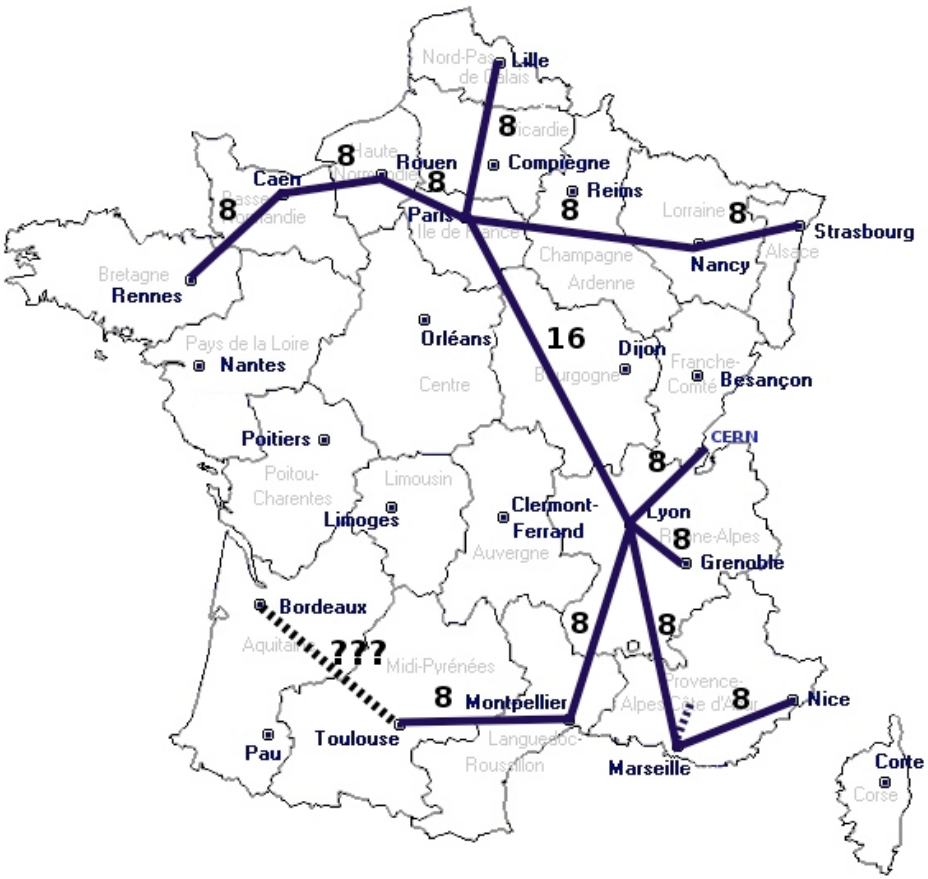


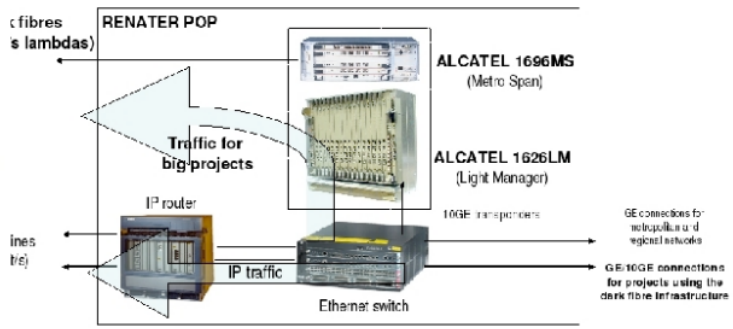
7244 cores

- 4 sites have difficulties to renew their hardware:
  Lyon and Orsay (nothing new since 2006); Toulouse and Bordeaux (2007)
- Hardware on the 5 other sites were recently replaced
- 3 nodes in Luxembourg, Reims and Bresil currently integrated
  that's administratively and technically far from being trivial

Grid'5000 Network

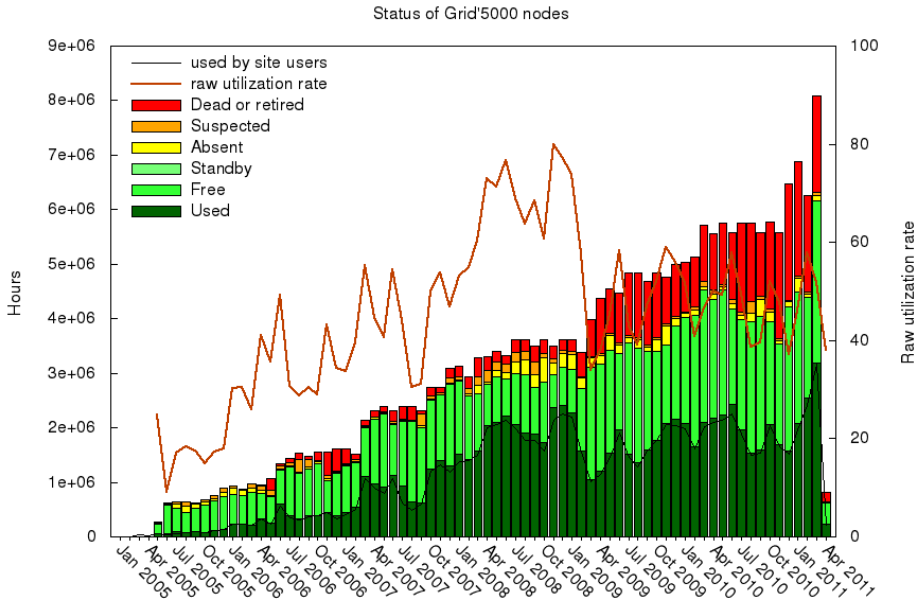
Strong implication of Renater

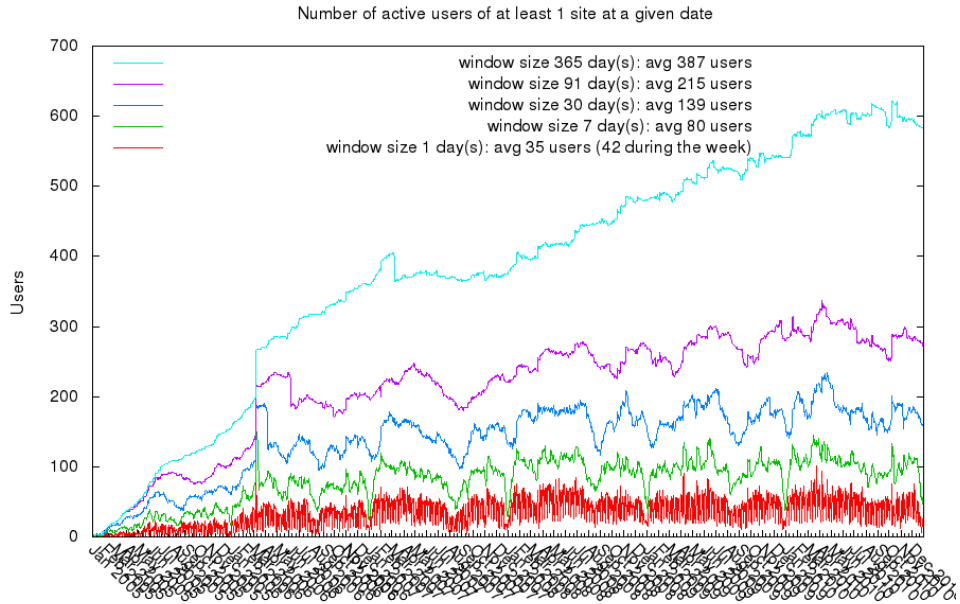


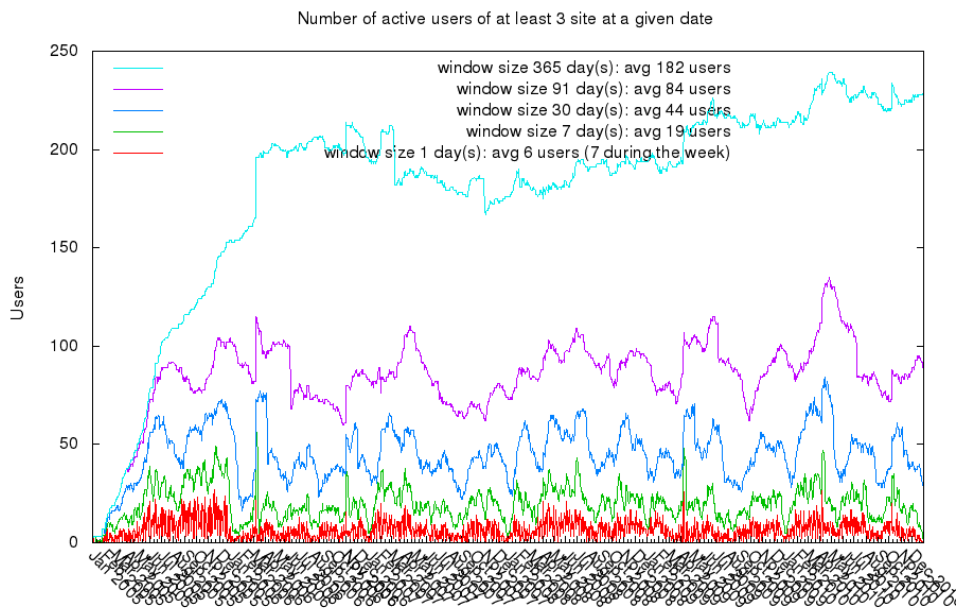


- v1: 1Gbps (IP premium QoS)
- v2: 10Gbps (dedicated lambda)
- v3: x10Gbps
  - dedicated lambda + traffic isolation
  - Resources reserved on demand

Resource Usage since 2005







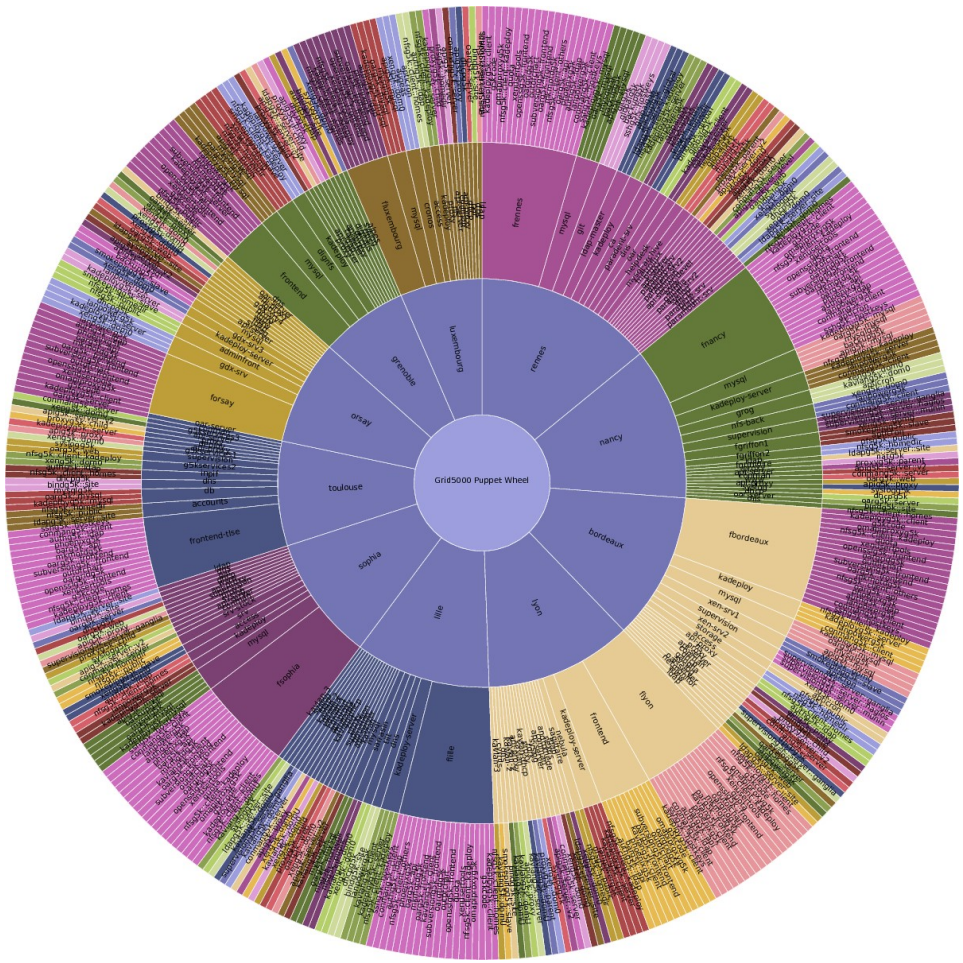
Estimations of costs (raw estimates)

What	Annual Cost	Who pays?
Network Backbone	1 720 000	RENATER
Hardware	1 000 000	INRIA + CPER
Personal	500 000	INRIA, University Rennes 1
Energy	450 000	INRIA, CNRS Universities hosting clusters
Missions/Animation	50 000	INRIA

3 million euros per year is quite an amount of money, but...

- That's only about 5000 euros/active user/year, ie about 2 servers
  Not counting the energy nor the administration effort
- That's under 0.06 euros per hour (Amazon is at $0.085 per hour)

System Administration Challenges



Goals and means

- Automating to factorize
  From 12 to 6 people
- Unique domain
  Intervention range unlimited
- Receipts in a central git
  - Puppet for servers
  - Chef for images
- Capistrano to push configs

Results

- Most know how encoded in receipts (young engineers-friendly)
- Hard to handle HPC hosts this way

Grid'5000 Day @LORIA

- Motivation: why Grid'5000
- Grid'5000 Design
- State of Grid'5000
- Some Results
- Conclusion

ScaLab: industrializing the experiments

Layer 3: Experimental methodology

- Run several experiments to reach conclusions

Layer 2: Orchestration

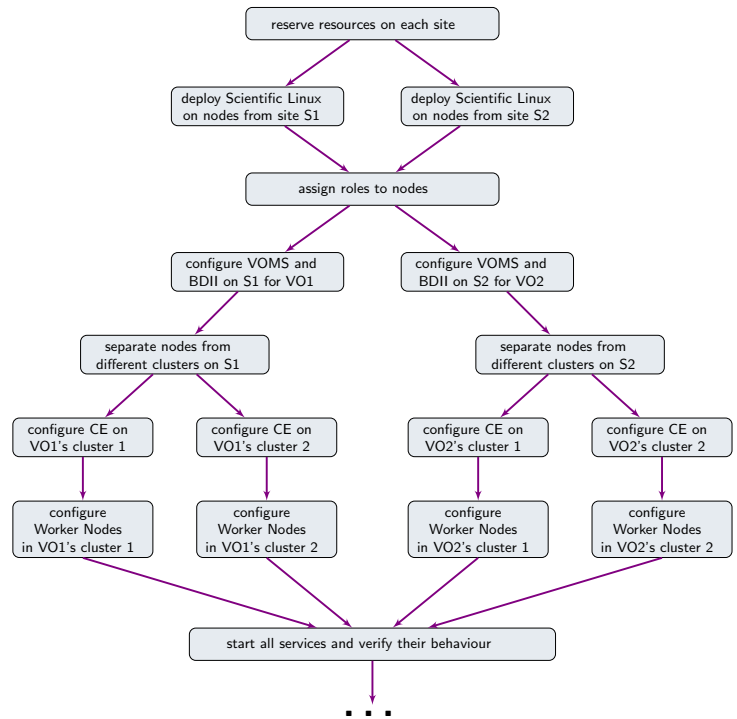
Run each experiment

- Run unattended, Handling of faults
- Composition of simpler experiments

Layer 1: Building blocks

Experiment Components

- Parallel Launcher; Data management
- Emulation; Fault / Load injection
- Monitoring / instrumentation



Layer 0: Experimental testbed (Grid'5000 or others)

Provides hardware and network

Quelques pistes de recherche autour de Grid’5000

Applications

- Simulations stochastiques multi-paramétriques intensives pour l’hydrogéologie
- Simulation électromagnétique
- Calcul à grande échelle pour les problèmes d’optimisation combinatoire
- Métaheuristiques hybrides parallèles sur grilles de calcul. Application au Q3AP et au problème des règles de Golomb
- Cryptanalyse de primitives fondamentales en cryptologie asymétrique, et étude de l'apport de Grid'5000 pour les calculs d'algèbre linéaire induits par ces algorithmes

Grilles/Desktop CP

- Expérimentation du système XtreemOS à très large échelle
- Robustness of large systems in presence of high churn (P2P-ch)
- Profiling énergétique pour les applications à grande échelle
- Gestion de la sécurité dans les grilles de calcul
- Application Autonome sur Grille

Réseau

- Analyser et comprendre le traffic

Clouds

- Traitement distribué extensible utilisant le paradigme MapReduce
- Gestion de données partagées sur des infrastructures de type cloud
- Virtualisation et cloud computing dans les infrastructures distribuées à grand échelle

OS Grid/Cloud-aware

Des systèmes et des plates-formes hétérogènes
    Grappes, grilles, clouds

Une utilisation compliquée
    intergiciels multiples, OS différents, gestion de ressources, de données, fichiers, tolérance aux pannes, sécurité, ...

Vers des OS pour les grilles et les clouds ?

Des challenges !
    Maîtrise de la grande échelle

- Nombre de ressources
- Sites et domaines multiples

    Dynamicité

- Charge, pannes, ajouts de ressources

    Difficulté à prédire le comportement des plates-formes et de leurs utilisateurs





XtreemOS

http://www.xtreemos.eu

Un système d'exploitation distribué pour les grilles

- Support des VOs multiples
- Ensemble de services coopérants
- Interface Posix/Unix
- Basé sur Linux
- API SAGA (OGF) pour les applications

Extensibilité

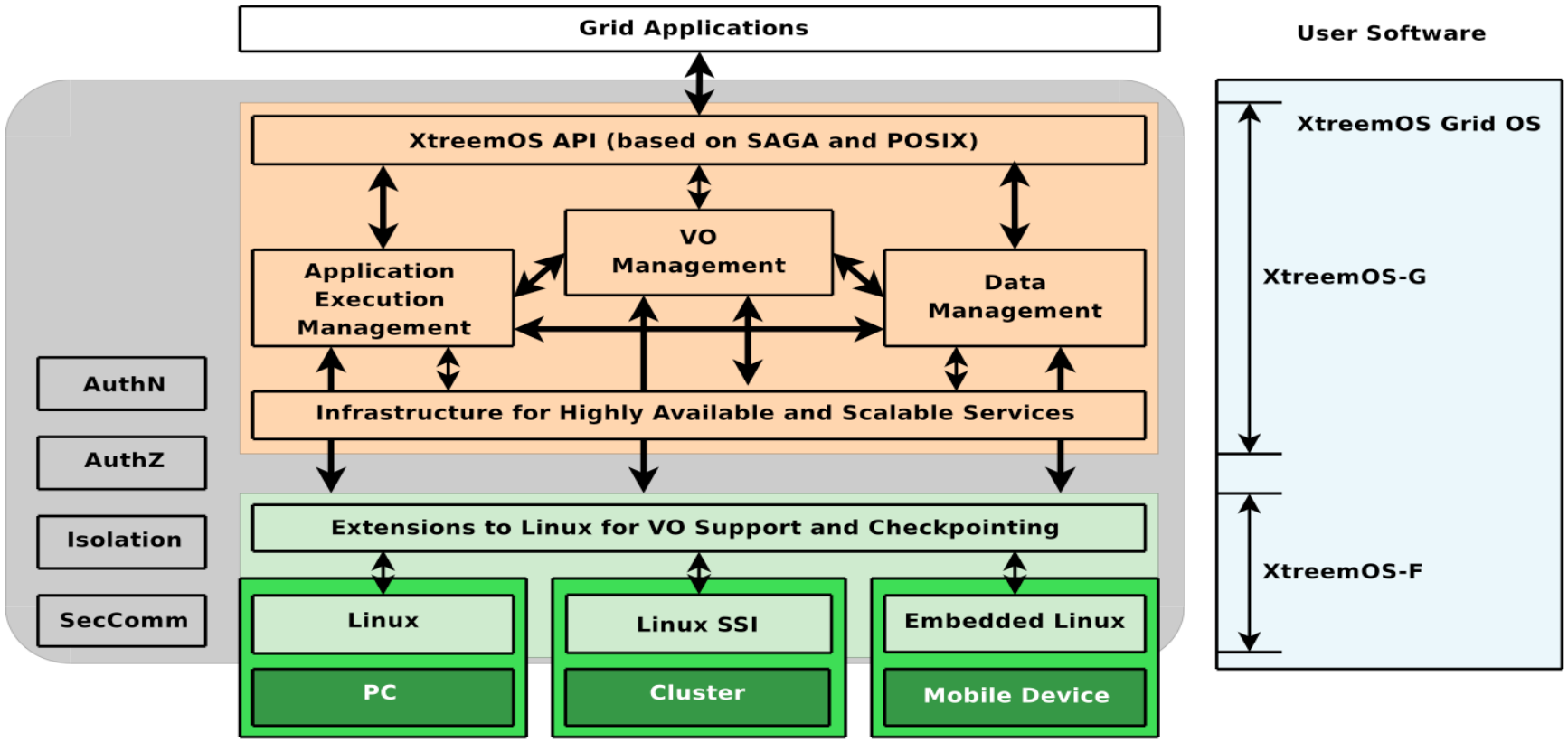
- Gestion de la grande échelle et des domaines administratifs différents
- Distribution, réplication et migration des services XtreemOS

Quelques fonctionnalités

- Gestion de VOs extensible, système de fichier grille (XtreemFS), monitoring, single-sign-on, checkpointing générique, outils pour l'auto-configuration et le déploiement automatique, découverte de ressources décentralisée, support pour les travaux interactifs, ...

Ouverture vers les Clouds (IaaS)

Architecture XtreemOS



Gestion optimisée de l'énergie

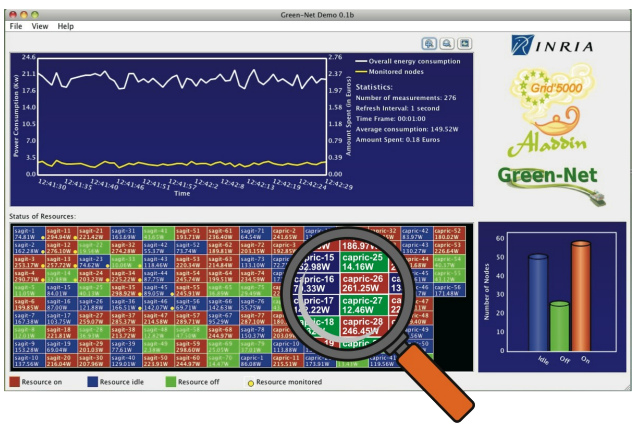
Les grilles et les Clouds participent au changement climatique !
Approches « vertes » pour les grilles

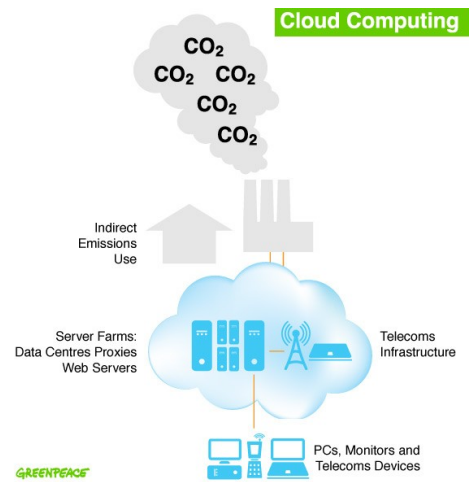
Optimisation : améliorer la conception des matériels et logiciels pour réduire leur consommation d'énergie

Eteindre / ***Shutdown*** : réduire le nombres de ressources et d'équipements alimentés et inutiles : nœuds de calculs, de stockage, de communication, périphériques, …

Adaptation / **Slowdown** : adapter la vitesse des ressources à l'usage réel : DVFS, ALR, …

Coordination : proposer des solutions à grande échelle afin de bénéficier de leviers de réduction énergétique plus importants





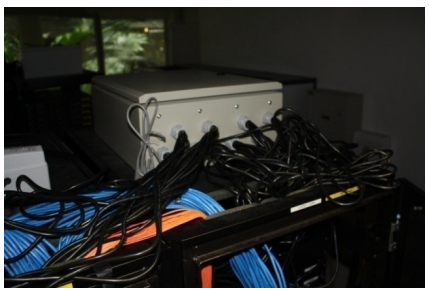
GreenIT logiciel

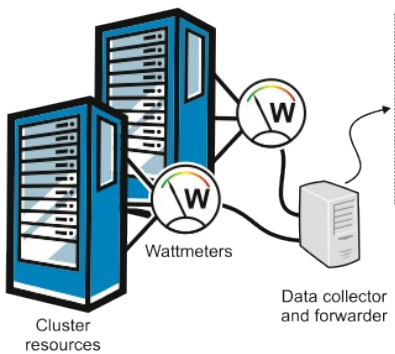
Etapes

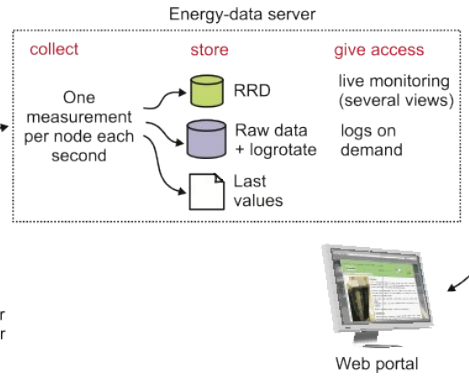
- Mesurer / collecter des informations sur la consommation électrique
- Injecter dans les systèmes d'information et composants
- Définir des environnements logiciels sensibles à la consommation électrique (protocoles, services, applications)
- Sensibiliser les utilisateurs et fournir des stratégies d'usage raisonnées / plus vertes

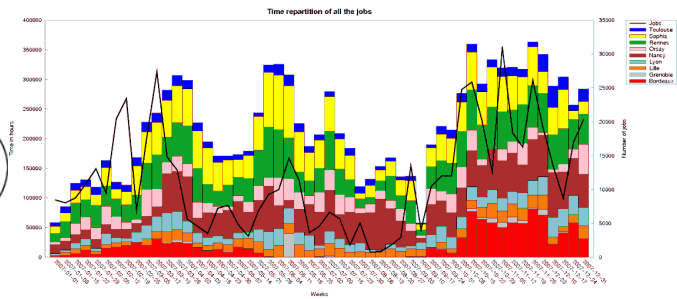
Exemple sur Grid'5000

- Observation et contrôle de 160 nœuds
- Déploiement de wattmètre pour chaque équipement





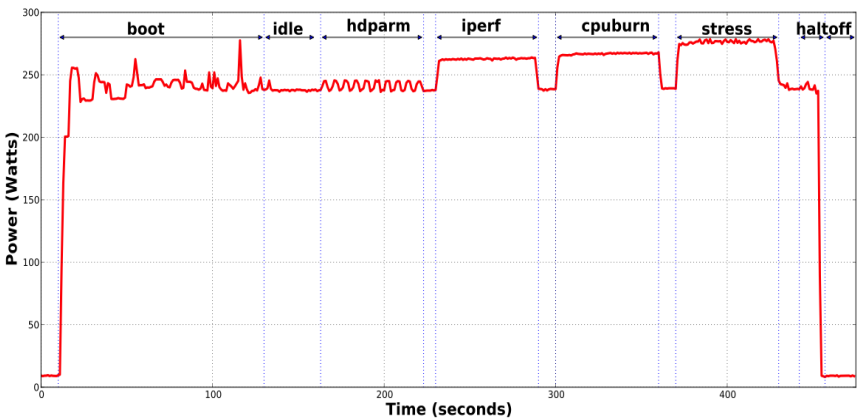


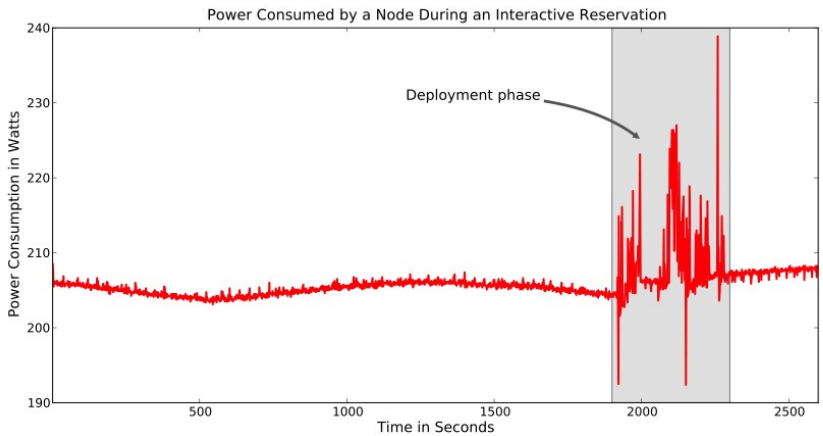


Crédits: Laurent Lefèvre (INRIA/LIP)

Vers des grilles « vertes »

L’énergie est un paramètre incontournable des infrastructures distribuées à grande échelle
Mesurer/collecter/exposer la consommation énergétique des systèmes, des services et des applications est le premier pas indispensable
Donner la possibilité aux utilisateurs/administrateurs des Grilles d’exprimer des compromis Energie/performance/réactivité…
Ajouter le contexte énergie dans le composants logiciels des grilles (ordonnanceurs, gestionnaire de ressources…)
Proposer des leviers de réduction énergétique aux utilisateurs





Crédits: Laurent Lefèvre (INRIA/LIP)

Gestion de la virtualisation à grande échelle

Concept de virtualisation

Le système d'exploitation n'est plus central et est un logiciel comme un autre !

Le concept de machines virtuelles consiste à recevoir des instances de systèmes.

Les ressources physiques sont partagées par plusieurs machines virtuelles

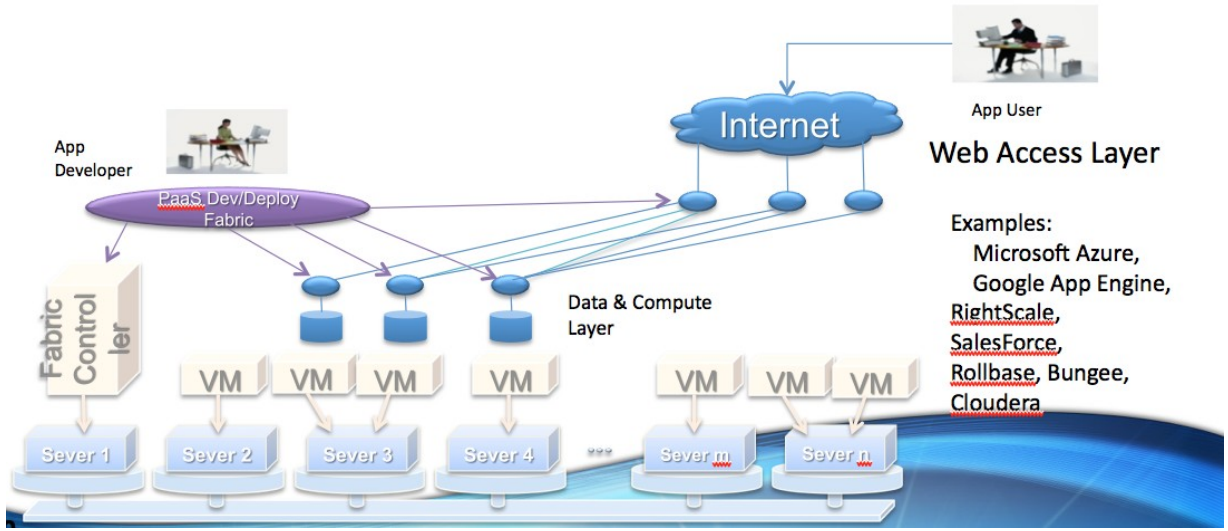
Propriétés

Isolement

Portabilité

Suspend/restart

Composant essentiel des Clouds



Gestion de la virtualisation à grande échelle

Quelques problématiques liées à l'utilisation de machines virtuelles à grande échelle

Gérer les tâches dynamiquement (**SALINE**)

- Utiliser les capacités des machines virtuelles pour suspendre une tâche et l'exécuter (potentiellement) ailleurs
- Sauver des snapshots et gérer leur sauvegarde de manière extensible

Combiner la virtualisation des ressources et du réseau (**HiperNet**)

- Donner à l'utilisateur l'illusion qu'il utilise un système privé

Un langage de reconfiguration pour les infrastructures virtualisées (**VMScript**)

- pour décrire les jobs, les VOs, les architectures physiques

Migrer une image virtuelle ou un cluster entier entre des datacenters (**Shrinker**)

- Réduire le coût de déplacement (ne transférer que le strict minimum)
- Tirer partie de la bande-passante d'un réseau WAN

Gestion de ressources à grande échelle

Comment ordonnancer les tâches à grande échelle

Workflows, tâches hétérogènes, liens avec la gestion de données, la réplication, modèles énergétiques, …

Comment gérer l'allocation de machines virtuelles

Modéliser les plates-formes, prédire?

Gestion élastique des ressources

Maîtriser les pics de charge

Déplacer les VMs, les tâches

Modèles économiques, énergétiques, …

Méta-schedulers

Ordonnancer et gérer les ressources dans un monde multi-batch et multi-gestionnaires

Pilot jobs

- DIRAC, Condor

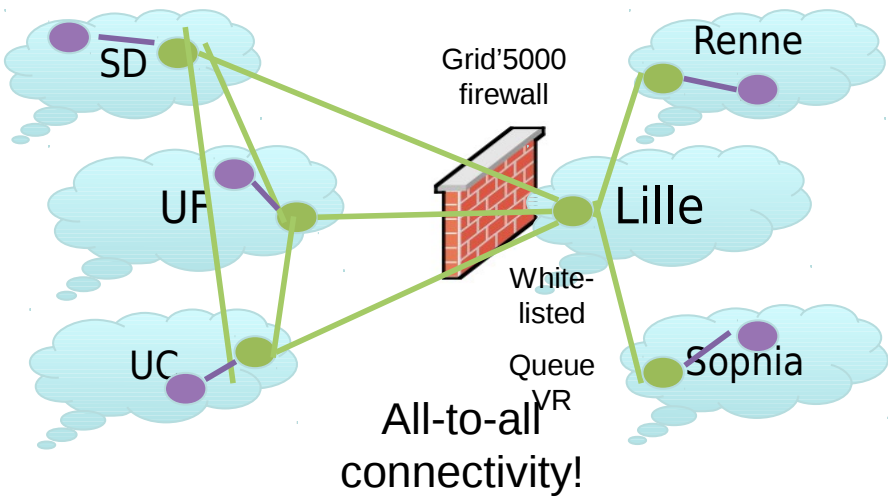
Sky Computing

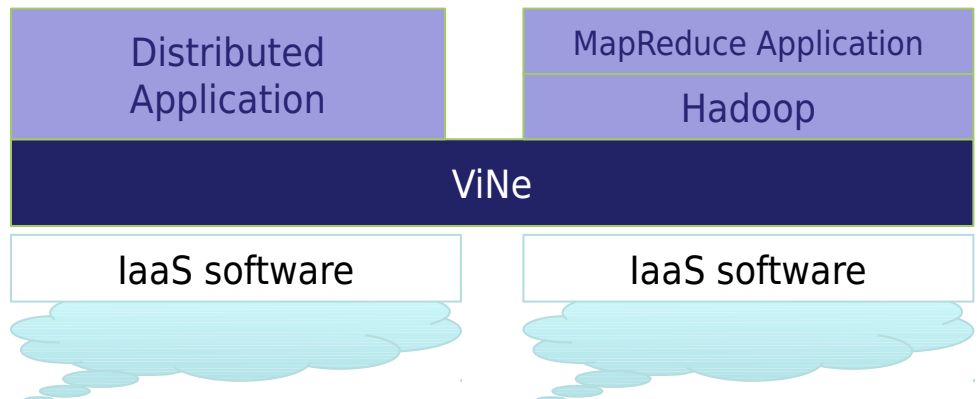
Allowing the execution of applications at large scale over multi-cloud platform
Experiments between USA and France

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



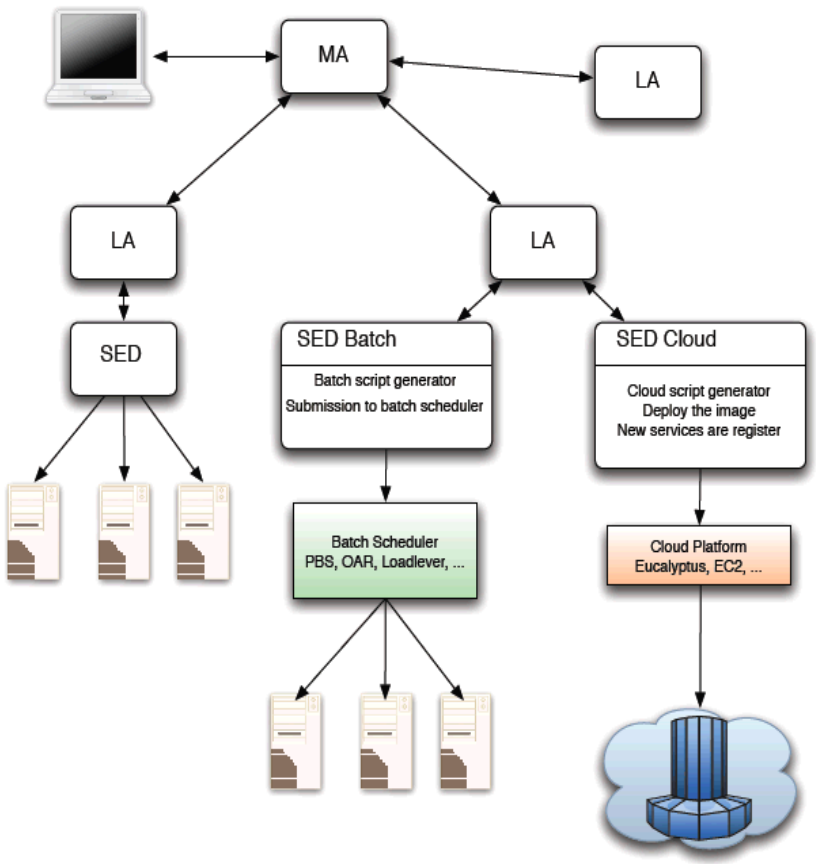


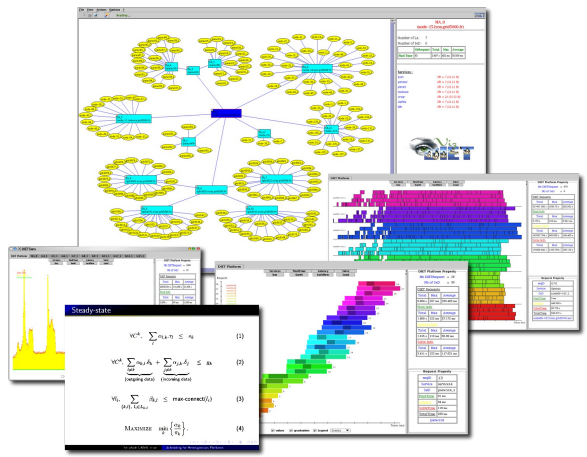
Crédits: Pierre Riteau (IRISA)

Combining Grids and Clouds

Inside the Cloud
+ DIET platform is virtualized inside the cloud. (as Xen image for example)
+ Very flexible and scalable as DIET nodes can be launched
+ Dynamic adaptation % charge

Cloud manager
+ EC2 interface
+ EC2 is treated as a new Batch System
+ Automatic deployment of VMs with associated services





http://www.sysfera.fr/

Grid'5000 Day @LORIA

- Motivation: why Grid'5000
- Grid'5000 Design
- State of Grid'5000
- Some Results
- Conclusion

Conclusion

Scientific Instruments are mandatory in Computer Science

- Even if they are not sufficient (just like in other sciences)
- Grid'5000 is the world leading tool

This scientific instrument is functional

- $\approx$ 7000 cores ($\approx$ 1500 nods), 10Gb Backbones
- Tools for reservation (OAR) and reconfiguration (Kadeploy)

Please use it now...

- Dimensioning experiments ($>$ 1000 hosts, $>$ 3 sites) particularly welcomed

...for computer science experiments

- Grid'5000 is not a production grid
- Bad sign: the computation result matters more than the way to get it
- Production Platforms elsewhere: EGEE, cines, *TALC clusters*

Les rapports utilisateurs (prix d'un accès à Grid'5000)

Fondamentaux: nous permettent de justifier les fonds engagés

Informations demandées

- Main information
- Experiments
- Publications
- Collaboration
- Results
- Benefits

https://www.grid5000.fr/index.php/Special:UserReports