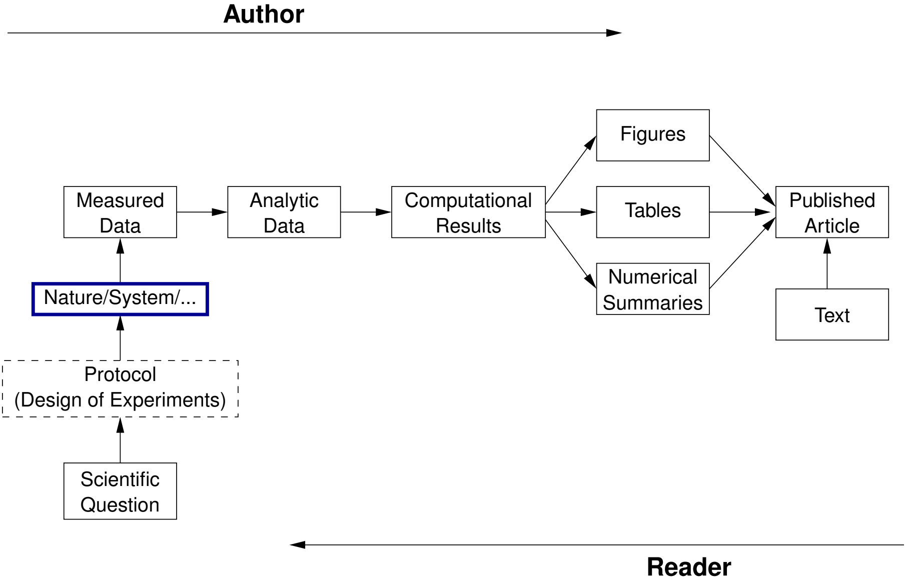
Reproducible Research on Grid'5000

Lucas Nussbaum

and many others, including Olivier Richard, Cristian Ruiz, Tomasz Buchert,
the Grid'5000 architects committee and the Grid'5000 technical team

Distributed computing: a peculiar field in CS

- Most contributions are validated using experiments
  - Very little formal validation in distributed computing
  - Even for theoretical work $\leadsto$ simulation (SimGrid)
- Performance and scalability are central to results
  - But depend greatly on the testbed (hardware, network, software, etc.)
  - Many contributions are about *fighting* the platform (load balancing, fault tolerance, middlewares/runtimes, etc.)
- Experimenting is difficult and time-consuming
- Shifts the scope for reproducible research:
  - **How can one perform "good" experiments?**
  - Very similar to (not computational) biology or physics



Author

Scientific Question

Protocol (Design of Experiments)

Nature/System/...

Measured Data

Analytic Data

Computational Results

Figures

Tables

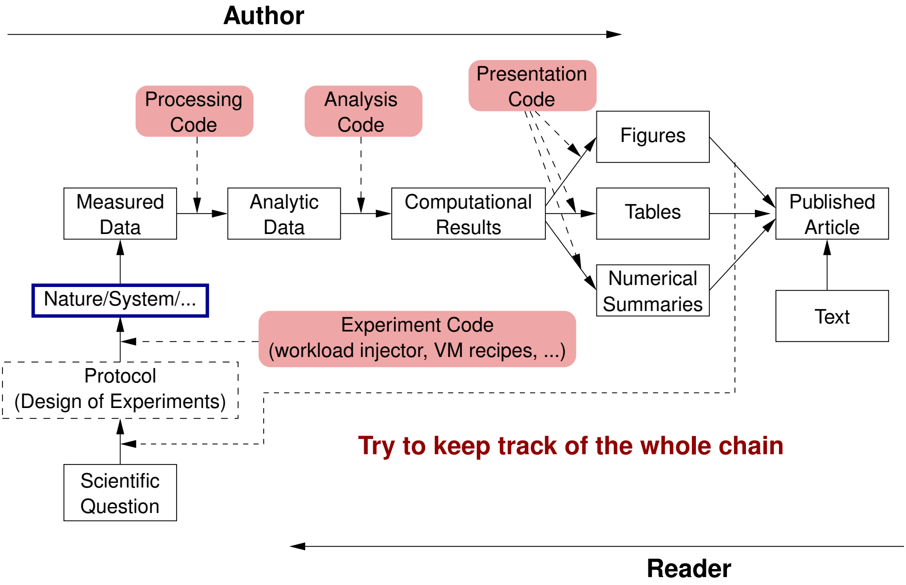
Numerical Summaries

Text

Published Article

Reader

Inspired by Roger D. Peng's lecture on reproducible research, May 2014
Improved by Arnaud Legrand



Author

Processing Code

Analysis Code

Presentation Code

Measured Data

Analytic Data

Computational Results

Figures

Tables

Numerical Summaries

Published Article

Text

Nature/System/...

Experiment Code (workload injector, VM recipes, ...)

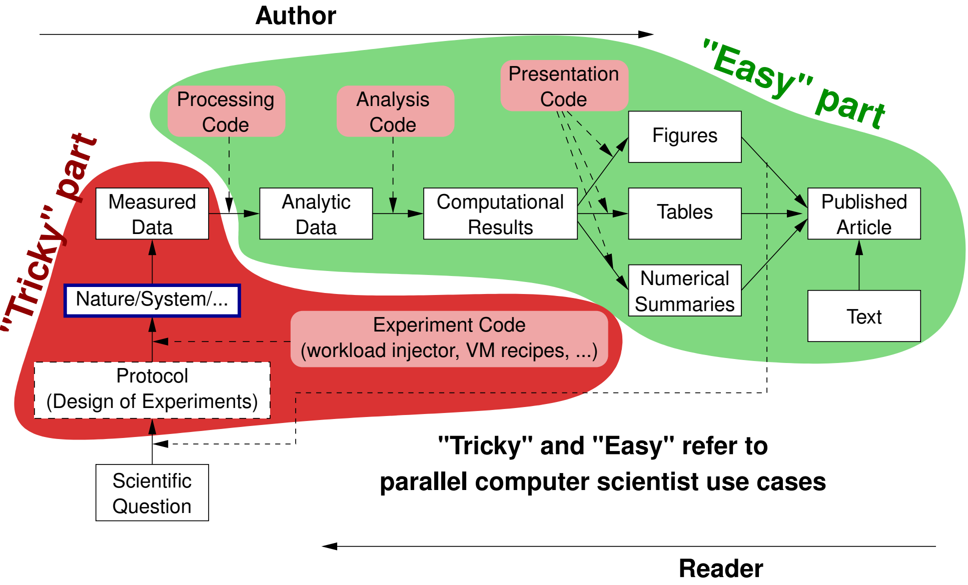
Protocol (Design of Experiments)

Scientific Question

Try to keep track of the whole chain

Reader

Inspired by Roger D. Peng's lecture on reproducible research, May 2014
Improved by Arnaud Legrand



Author

"Easy" part

"Tricky" part

Processing Code

Analysis Code

Presentation Code

Measured Data

Analytic Data

Computational Results

Figures

Tables

Numerical Summaries

Published Article

Text

Nature/System/...

Experiment Code (workload injector, VM recipes, ...)

Protocol (Design of Experiments)

Scientific Question

Reader

"Tricky" and "Easy" refer to parallel computer scientist use cases

Inspired by Roger D. Peng's lecture on reproducible research, May 2014
Improved by Arnaud Legrand

Description and verification of the testbed

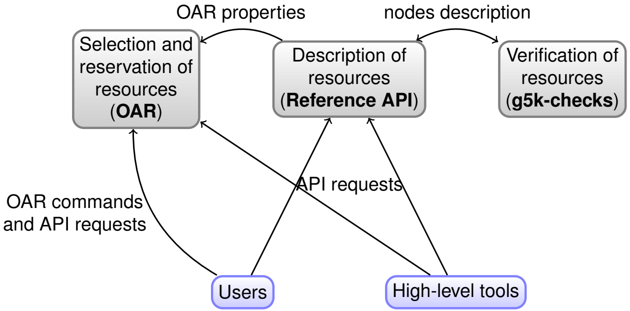
Typical needs:

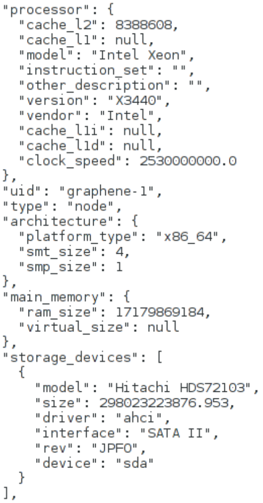
- Find suitable resources for my experiment
- Ensure that the resources match their description
- Find the reference of the disk on nodes used six months ago

Description and verification of the testbed

Typical needs:

- Find suitable resources for my experiment
- Ensure that the resources match their description
- Find the reference of the disk on nodes used six months ago


OAR properties
nodes description
Selection and reservation of resources (**OAR**)
Description of resources (**Reference API**)
Verification of resources (**g5k-checks**)
API requests
OAR commands and API requests
Users
High-level tools

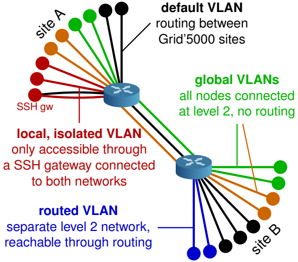


```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "platform_type": "x86_64",
  "smt_size": 4,
  "smp_size": 1
},
"main_memory": {
  "ram_size": 17179869184,
  "virtual_size": null
},
"storage_devices": [
  {
    "model": "Hitachi HDS72103",
    "size": 298023223876.953,
    "driver": "ahci",
    "interface": "SATA II",
    "rev": "JPFO",
    "device": "sda"
  }
],
```

Reconfiguring to meet experimental needs

- Operating System reconfiguration with **Kadeploy**:
  - Provides a *Hardware-as-a-Service* Cloud infrastructure
  - Enable users to get *root* access & deploy their own software stack
  - **Scalable, efficient, reliable and flexible**:
    200 nodes deployed in ~5 minutes (120s with Kexec)

- Customize networking configuration with **KaVLAN**
  - Deploy intrusive middlewares (Grid, Cloud)
  - Protect the testbed from experiments
  - Avoid network pollution
  - By reconfiguring VLANS ⇝ almost no overhead
  - Recent work: support several interfaces



default VLAN
routing between
Grid'5000 sites

global VLANs
all nodes connected
at level 2, no routing

local, isolated VLAN
only accessible through
a SSH gateway connected
to both networks

routed VLAN
separate level 2 network,
reachable through routing

Creating and sharing Kadeploy images

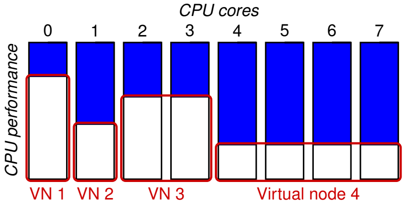
- Avoid manual customization:
  - Easy to forget some changes
  - Difficult to describe
  - The full image must be provided
  - Cannot really be used as a basis for future experiments (similar to binary vs source code)
- Kameleon: Reproducible generation of software appliances
  - Using *recipes* (high-level description)
  - Persistent cache to allow re-generation without external resources (Linux distribution mirror) $\rightsquigarrow$ self-contained archive
  - Supports Kadeploy images, LXC, Docker, VirtualBox, qemu, etc.

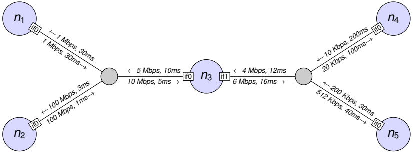
http://kameleon.imag.fr/

PhD of Cristian Ruiz (Hemera PhD)

Changing experimental conditions

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



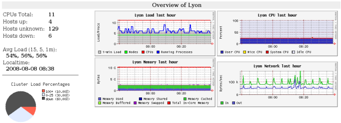


- Collaborations with Trong-Tuan Vu (Hemera PhD, Dolphin team) and Abhishek Gupta (UIUC, Laxmikant Kalé)

http://distem.gforge.inria.fr/

Monitoring experiments

Goal: enable users to understand what happens during their experiment

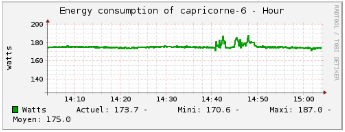


Overview of Lyon

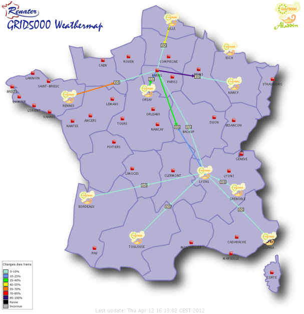
CPUs Total: 11
Hosts up: 4
Hosts unknown: 129
Hosts down: 6

Avg Load (15, 5, 1m):
54%, 56%, 56%
Localtime:
2008-08-08 08:38

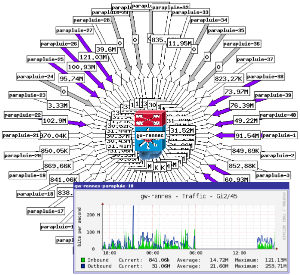
CPU – memory – disk



Power consumption



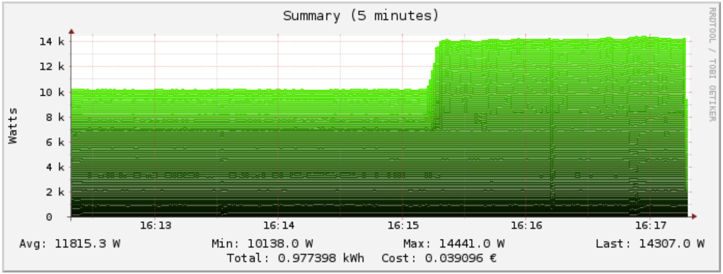
Network backbone



Internal networks

Monitoring experiments (2)

- Current work: high resolution monitoring for energy & network
  - Collaboration between Lyon and Nancy



Improving control and description of experiments

- Legacy way of performing experiments: shell commands
  - ☹ time-consuming
  - ☹ error-prone
  - ☹ details tend to be forgotten over time

- Promising solution: automation of experiments
  ⇝ Executable description of experiments
- Support from the testbed: Grid'5000 RESTful API
  (Resource selection, reservation, deployment)

Tools for automation of experiments

Several projects around Grid'5000 (but not specific to Grid'5000):

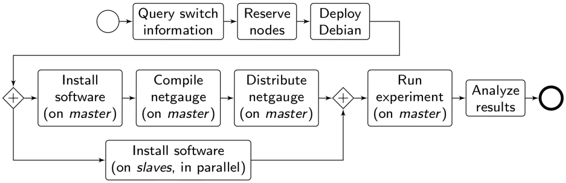
- g5k-campaign (G5K tech team)
- Expo (Cristian Ruiz)
- Execo (Mathieu Imbert)
- XPFlow (Tomasz Buchert)

Features:

- Facilitate scripting of experiments in high-level languages (Ruby, Python)
- Provide useful and efficient abstractions :[1]
  - Testbed management
  - Local & remote execution of commands
  - Data management
- *Engines* for more complex processes

[1] Tomasz Buchert et al. “A survey of general-purpose experiment management tools for distributed systems”. In: *Future Generation Computer Systems* (2015).

XPFlow



Query switch information

Reserve nodes

Deploy Debian

Install software (on *master*)

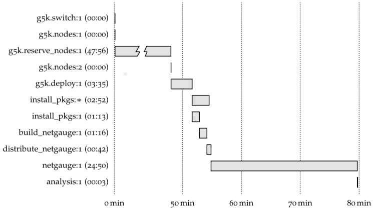
Compile netgauge (on *master*)

Distribute netgauge (on *master*)

Run experiment (on *master*)

Analyze results

Install software (on *slaves*, in parallel)



```
engine.process :exp do |site, switch|
    s = run g5k.switch, site, switch
    ns = run g5k.nodes, s
    r = run g5k.reserve_nodes,
        :nodes => ns, :time => '2h',
        :site => site, :type => :deploy
    master = (first_of ns)
    rest = (tail_of ns)
    run g5k.deploy,
        r, :env => 'squeeze-x64-nfs'
    checkpoint :deployed
    parallel :retry => true do
        forall rest do |slave|
            run :install_pkgs, slave
        end
        sequence do
            run :install_pkgs, master
            run :build_netgauge, master
            run :dist_netgauge,
                master, rest
        end
    end
    checkpoint :prepared
    output = run :netgauge, master, ns
    checkpoint :finished
    run :analysis, output, switch
end
```

Experiment description and execution as a Business Process Workflow

Supports parallel execution of activities, error handling, snapshotting, built-in logging, etc.

soon: automatic provenance collection

What's next?

- Description and verification of the testbed
  - Provide testbed description in other formats (SimGrid) – *80% done*
  - Track testbed's performance history
  - Support for archiving the state of the testbed before experiments
- Reconfiguring the testbed to meet experimental needs
  - Enabling users to change BIOS parameters (power, HT, TB)
  - Providing control over cooling, network and storage systems
- Monitoring experiments, extracting/analyzing data
  - Monitor other pieces of the infrastructure (e.g. storage)
  - Provide long-term archival of experiments and traces
- Control and description of experiments
  - Extend and improve the API (reliability, features)
  - Foster collaboration on XP control tools, and transfer them to users

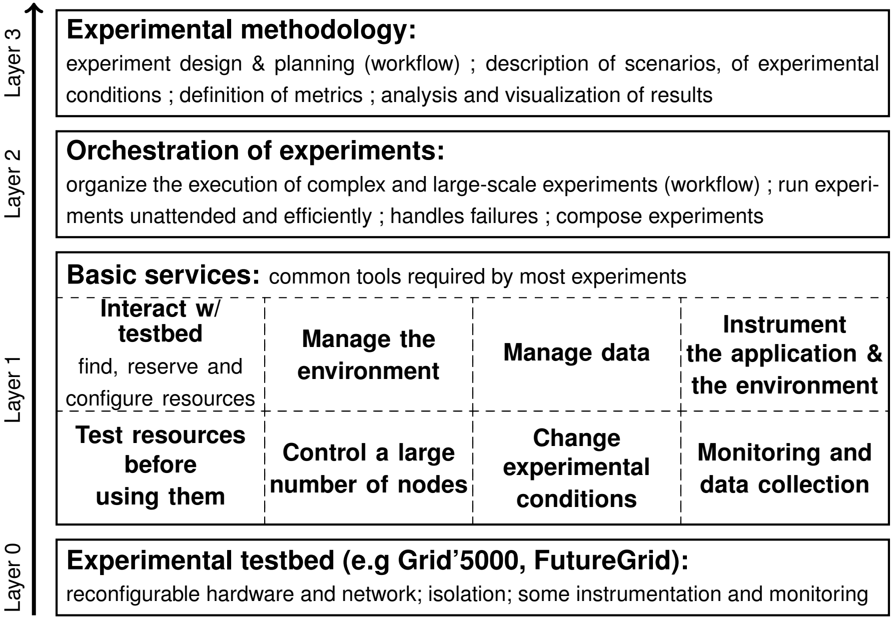
One could determine the age of a science by looking
at the state of its measurement tools.

Gaston Bachelard – *La formation de l'esprit scientifique*, 1938

Bibliography

- Resources management: Resources Description, Selection, Reservation and Verification on a Large-scale Testbed. `http://hal.inria.fr/hal-00965708`
- Kadeploy: Kadeploy3: Efficient and Scalable Operating System Provisioning for Clusters. `http://hal.inria.fr/hal-00909111`
- KaVLAN, Virtualization, Clouds deployment:
  - Adding Virtualization Capabilities to the Grid'5000 testbed. `http://hal.inria.fr/hal-00946971`
  - Enabling Large-Scale Testing of IaaS Cloud Platforms on the Grid'5000 Testbed. `http://hal.inria.fr/hal-00907888`
- Kameleon: Reproducible Software Appliances for Experimentation. `https://hal.inria.fr/hal-01064825`
- Distem: Design and Evaluation of a Virtual Experimental Environment for Distributed Systems. `https://hal.inria.fr/hal-00724308`
- XP management tools:
  - A survey of general-purpose experiment management tools for distributed systems. `https://hal.inria.fr/hal-01087519`
  - XPFlow: A workflow-inspired, modular and robust approach to experiments in distributed systems. `https://hal.inria.fr/hal-00909347`
  - Using the EXECO toolbox to perform automatic and reproducible cloud experiments. `https://hal.inria.fr/hal-00861886`
  - Expo: Managing Large Scale Experiments in Distributed Testbeds. `https://hal.inria.fr/hal-00953123`

A multi-tier challenge



Layer 3

Experimental methodology:

experiment design & planning (workflow) ; description of scenarios, of experimental conditions ; definition of metrics ; analysis and visualization of results

Layer 2

Orchestration of experiments:

organize the execution of complex and large-scale experiments (workflow) ; run experiments unattended and efficiently ; handles failures ; compose experiments

Layer 1

Basic services: common tools required by most experiments

Interact w/ testbed find, reserve and configure resources	**Manage the environment**	**Manage data**	**Instrument the application & the environment**
Test resources before using them	**Control a large number of nodes**	**Change experimental conditions**	**Monitoring and data collection**

Layer 0

Experimental testbed (e.g Grid'5000, FutureGrid):

reconfigurable hardware and network; isolation; some instrumentation and monitoring

Conclusions

- Grid'5000: a testbed for high-quality, reproducible research on HPC, Clouds and Big Data
- With a unique combination of features
  - Description and verification of testbed
  - Reconfiguration (hardware, network)
  - Monitoring
  - Support for automation of experiments
- Paving the way to Open Science of HPC and Cloud – mid term goals:
  - Fully automated execution of experiments
  - Automated tracking + archiving of experiments and associated data

One could determine the age of a science by looking
at the state of its measurement tools.

Gaston Bachelard – *La formation de l'esprit scientifique*, 1938