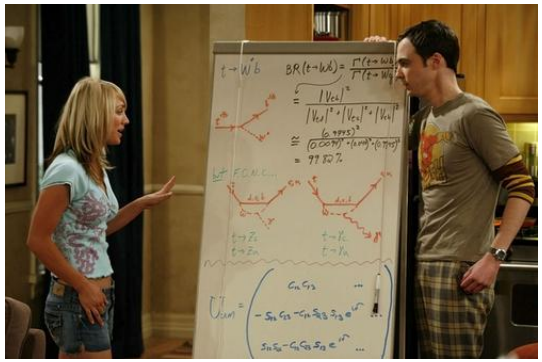
Supporting Experimental Science in Distributed Systems Research

Lucas Nussbaum

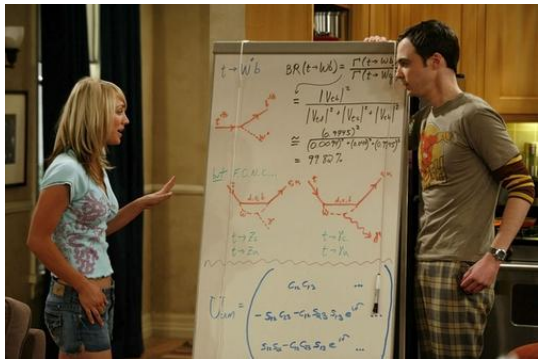
EPI ALGORILLE

Validation in (Computer) Science

- Two classical approaches for validation:
  - Formal: equations, proofs, etc.
  - Experimental, on a scientific instrument
- Often a mix of both:
  - In Physics
  - In Computer Science



Validation in (Computer) Science



- Two classical approaches for validation:
  - Formal: equations, proofs, etc.
  - Experimental, on a scientific instrument
- Often a mix of both:
  - In Physics
  - In Computer Science
- Very little formal validation in distributed systems research
  - Counter-examples:
    - Worst-case analysis of allocation/scheduling heuristics
    - Properties of algorithms (e.g. deadlock-free)
  - Our scientific objects are often intractable theoretically: too complex, dynamic, heterogeneous, large

(Poor) state of experimentation in CS

- 1994: survey of 400 papers[1]
  - *among published CS articles in ACM journals, 40%-50% of those that require an experimental validation had none*
- 1998: survey of 612 papers[2]
  - *too many papers have no experimental validation at all*
  - *too many papers use an informal (assertion) form of validation*
- 2009 update: *situation is improving*[3]

[1] Paul Lukowicz et al. “Experimental Evaluation in Computer Science: A Quantitative Study”. In: *Journal of Systems and Software* 28 (1994), pages 9–18.

[2] M.V. Zelkowitz and D.R. Wallace. “Experimental models for validating technology”. In: *Computer* 31.5 (1998), pages 23 –31.

[3] Marvin V. Zelkowitz. “An update to experimental models for validating computer technology”. In: *J. Syst. Softw.* 82.3 (Mar. 2009), pages 373–376.

(Poor) state of experimentation in CS (2)

- Most papers do not use even basic statistical tools

Papers published at the Europar conference[4]

Year	Tot. papers	With error bars	Percentage
2007	89	5	5.6
2008	89	3	3.4
2009	86	2	2.4
2010	90	6	6.7
2011	81	7	8.6
2007-2011	435	23	5.3

- 2007: Survey of simulators used in P2P research[5]
  - Most papers use an unspecified or custom simulator

[4] Study carried out by E. Jeannot.

[5] S. Naicken et al. “The state of peer-to-peer simulators and simulations”. In: *SIGCOMM Comput. Commun. Rev.* 37.2 (Mar. 2007), pages 95–98.

State of experimentation in other sciences

- 2008: Study shows lower fertility for mices exposed to transgenic maize
  - AFSSA report[6]:
    - *Several calculation errors have been identified*
    - *led to a false statistical analysis and interpretation*

[6] Opinion of the French Food Safety Agency (Afssa) on the study by Velimirov et al. entitled "*Biological effects of transgenic maize NK603xMON810 fed in long-term reproduction studies in mice*"

State of experimentation in other sciences

- 2008: Study shows lower fertility for mices exposed to transgenic maize
  - AFSSA report[6]:
    - *Several calculation errors have been identified*
    - *led to a false statistical analysis and interpretation*
- 2011: CERN Neutrinos to Gran Sasso project: faster-than-light neutrinos
  - 2012: caused by timing system failure

[6] Opinion of the French Food Safety Agency (Afssa) on the study by Velimirov et al. entitled "*Biological effects of transgenic maize NK603xMON810 fed in long-term reproduction studies in mice*"

State of experimentation in other sciences

- 2008: Study shows lower fertility for mices exposed to transgenic maize
  - AFSSA report[6]:
    - *Several calculation errors have been identified*
    - *led to a false statistical analysis and interpretation*
- 2011: CERN Neutrinos to Gran Sasso project: faster-than-light neutrinos
  - 2012: caused by timing system failure
- ☹ Not everything is perfect
- ☺ But some errors are properly identified

[6] Opinion of the French Food Safety Agency (Afssa) on the study by Velimirov et al. entitled "*Biological effects of transgenic maize NK603xMON810 fed in long-term reproduction studies in mice*"

Axes of improvement for experiments

- Improve quality
  $\rightsquigarrow$ more trustworthy results
  - Testbed description
  - Experiment description
  - Control of XP conditions
  - Automatize experiments
  - Monitoring & measurement
- Improve scope & scale
  $\rightsquigarrow$ more interesting results
  - Handle large number of nodes
  - Automatize experiments
  - Handle failures
  - Monitoring & measurement

Both goals raise similar challenges

Related to the Reproducible Research movement

- Mostly in computational sciences
- Explores tools and methods (provenance, executable papers, etc.)
- Different types of experimental reproducibility[7]:
  - *Replications that vary little or not at all with respect to the reference experiment*

    same method, environment, parameters $\rightarrow$ same result
  - *Replications that do vary but still follow the same method as the reference experiment*

    same method, but different {env., params} $\rightarrow$ same conclusion
  - *Replications that use different methods to verify the reference experiment results*

    different method $\rightarrow$ same conclusion

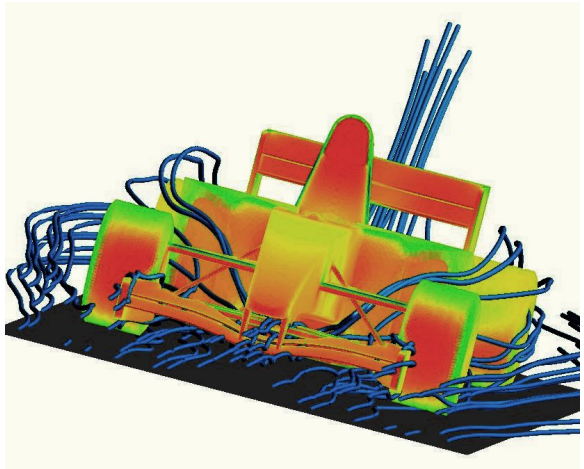
[7] Omar S. Gómez et al. “Replications types in experimental disciplines”. In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM ’10. 2010.

Outline

Experimentation methodologies in Hemera

Experimentation methodologies in Hemera

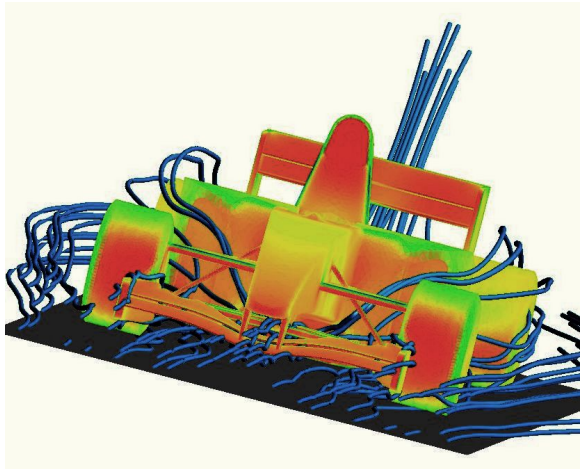
Simulation



1. **Model** application
2. **Model** environment
3. **Compute** interactions

Experimentation methodologies in Hemera

Simulation



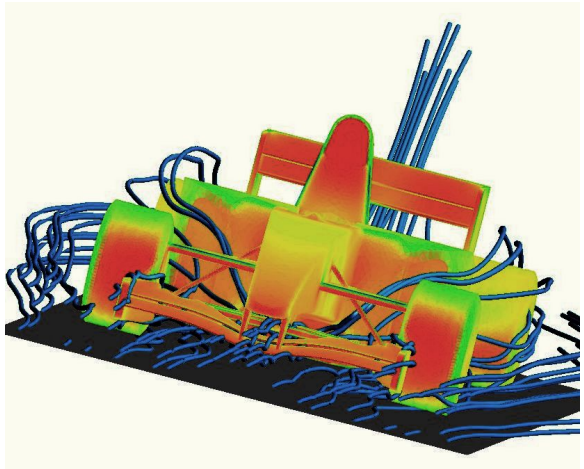
1. **Model** application
2. **Model** environment
3. **Compute** interactions

Real-scale experiments

Execute the **real** application on **real** machines

Experimentation methodologies in Hemera

Simulation



1. **Model** application
2. **Model** environment
3. **Compute** interactions

SIMGRID

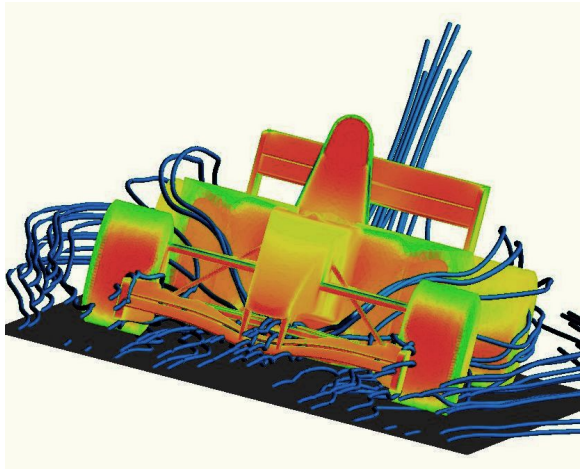
Real-scale experiments

Execute the **real** application on **real** machines

Grid'5000

Experimentation methodologies in Hemera

Simulation



1. **Model** application
2. **Model** environment
3. **Compute** interactions

Real-scale experiments

Execute the **real** application on **real** machines

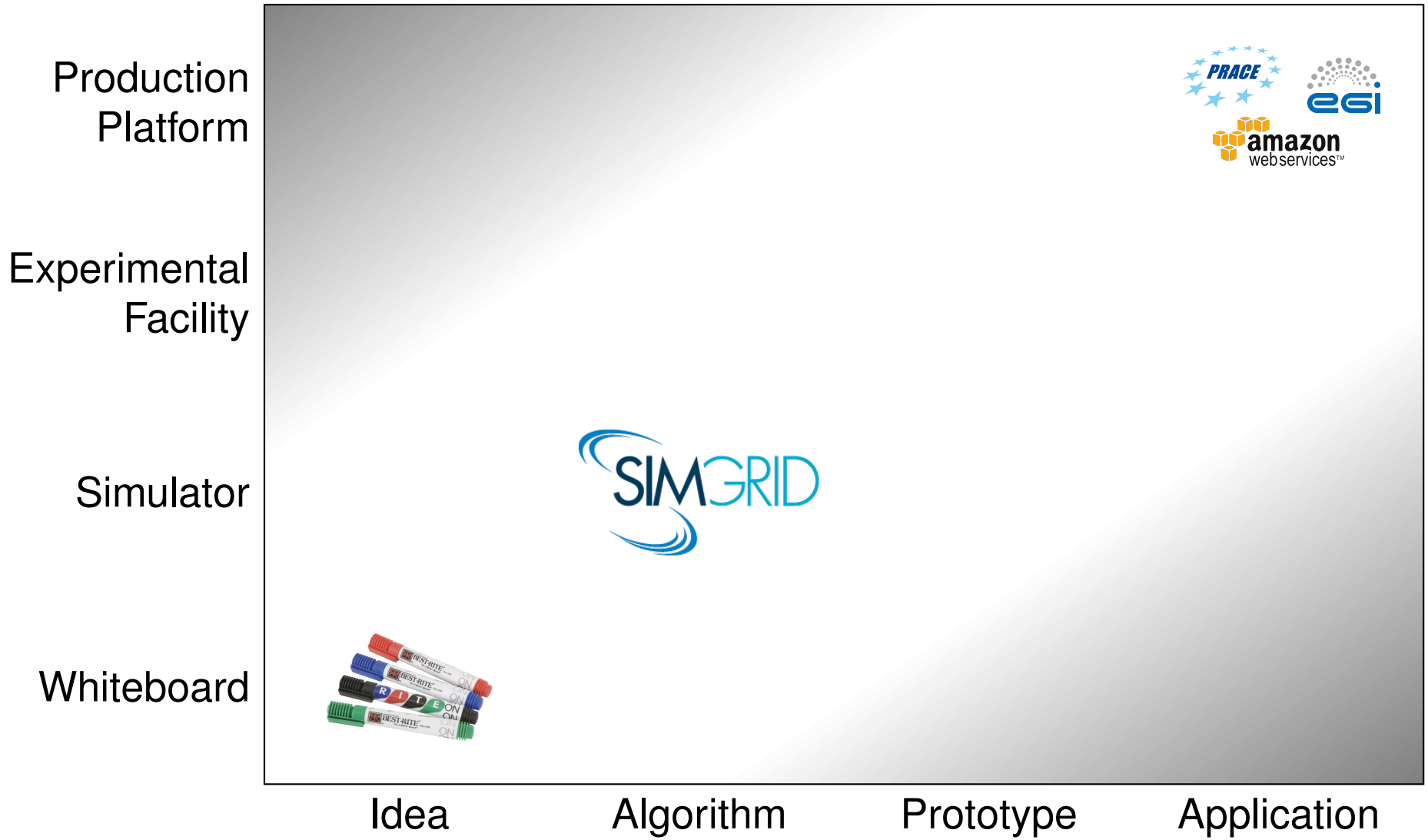
Complementary solutions:

☺ Work on algorithms
☺ More scalable, easier

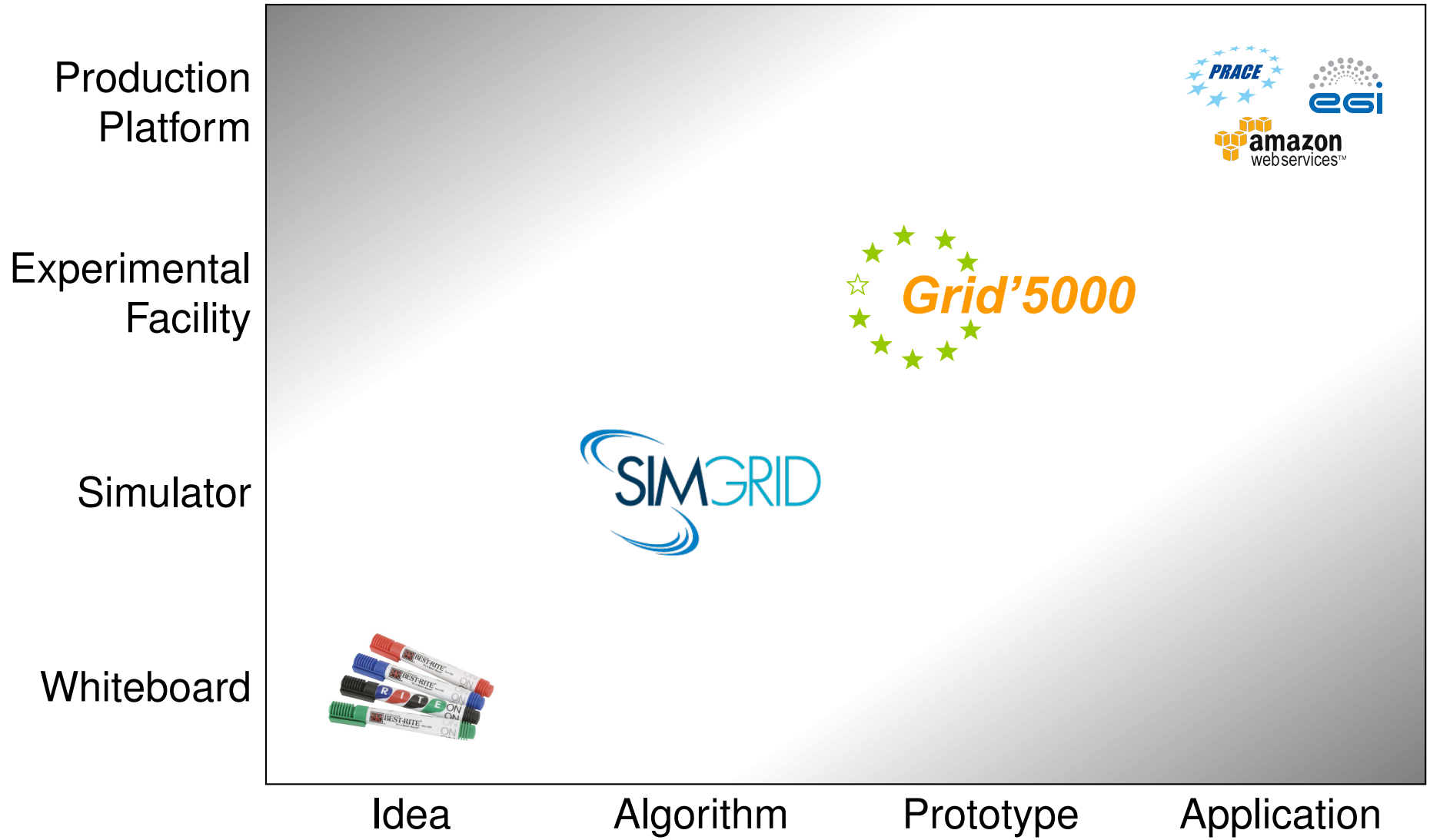
☺ Work on applications
☺ Perceived as more realistic

Leading experimenters from ideas to applications

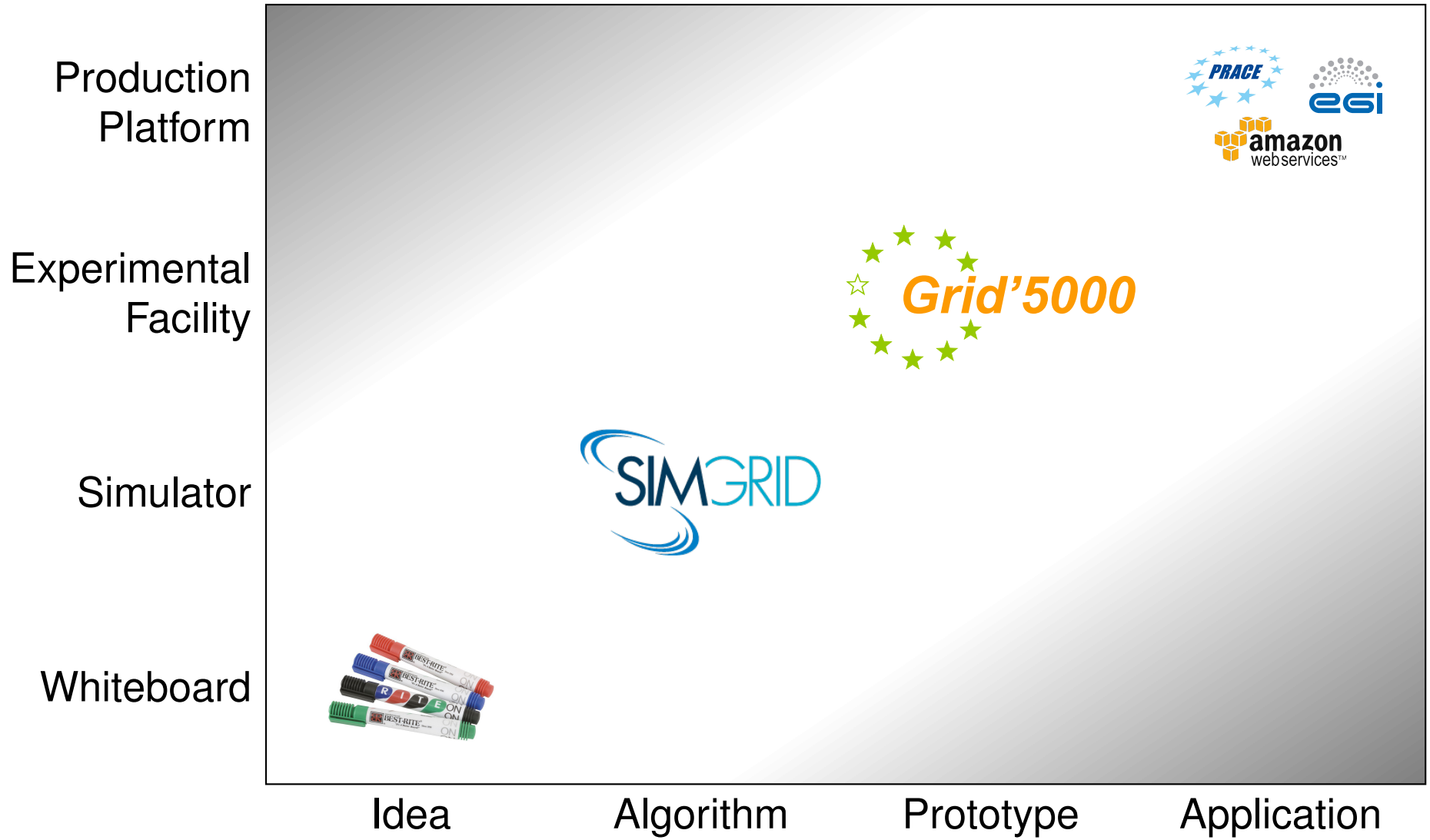
Leading experimenters from ideas to applications



Leading experimenters from ideas to applications



Leading experimenters from ideas to applications

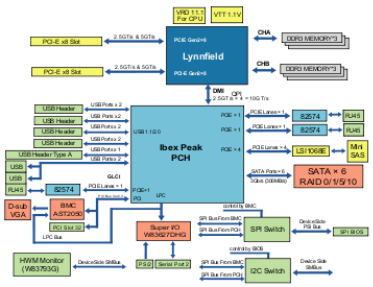


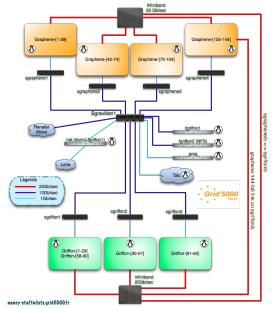
Convergence of methodologies to smoothen transitions

Outline

Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)

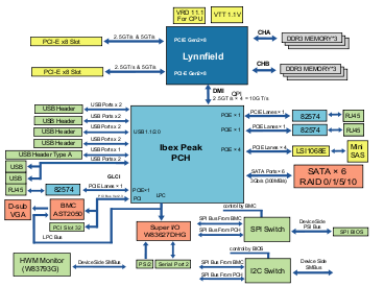


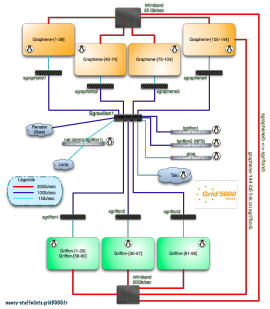


```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "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": "JPF0",
    "device": "sda"
  }
],
```

Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)





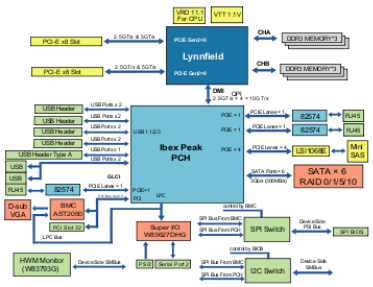

```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "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": "JPF0",
    "device": "sda"
  }
],
```

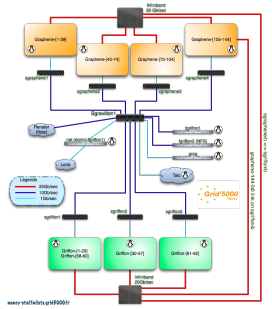

- Selecting resources
  - OAR database filled from JSON

```
oarsub -p "wattmeter='YES' and gpu='YES'"
```

Description, selection, verification of resources

- Describing resources ⇝ understand results
  - Detailed description on the Grid'5000 wiki
  - Machine-parsable format (JSON)





```
"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": "JPF0",
    "device": "sda"
  }
],
```

- Selecting resources
  - OAR database filled from JSON

    `oarsub -p "wattmeter='YES' and gpu='YES'"`

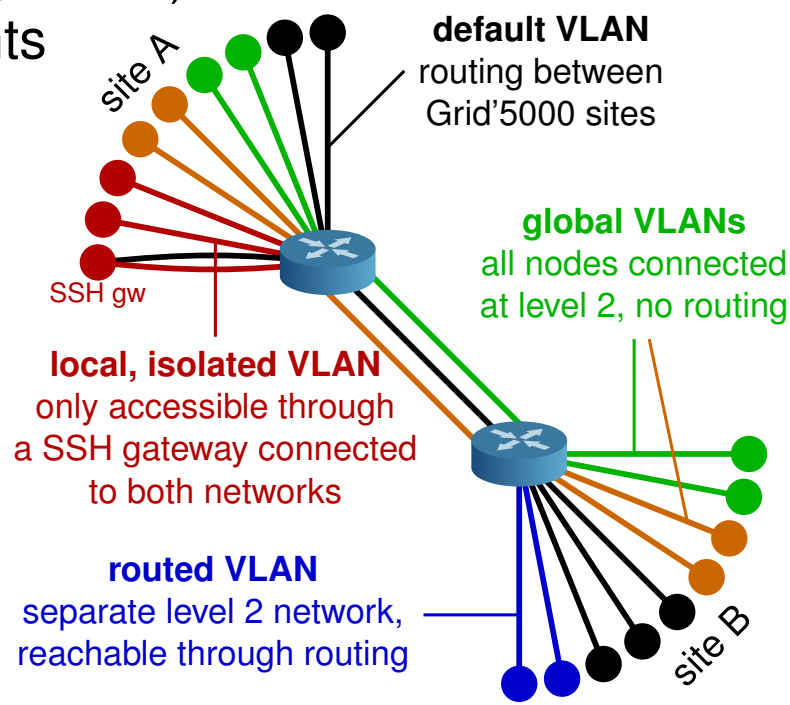
- Verifying resources
  - *G5K-checks*: validates resources against their description (detect hardware failures and misconfigurations at each boot)

Customizing the experimental environment

- Customize software environment with Kadeploy *(ADT 2011-2013)*
  - Enable users to deploy their own software stack & get *root* access
  - Standard environments provided to users
    - ★ Customizations automated using Kameleon (J. Emeras)
  - Re-install 200 nodes in ~5 minutes

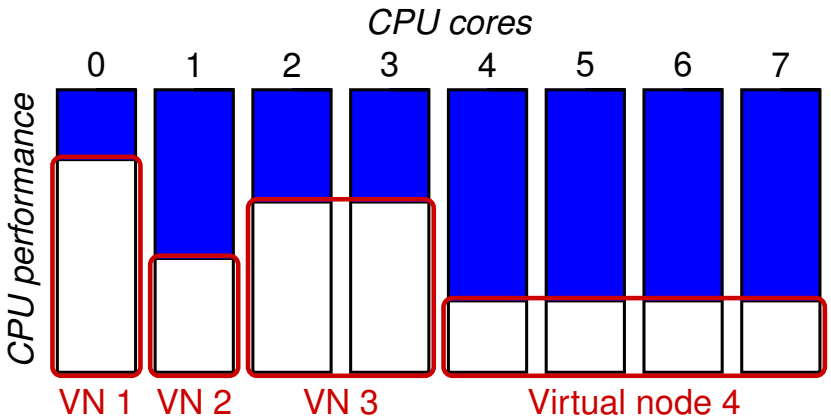
- Customize networking environment with KaVLAN
  - Deploy intrusive middlewares (Grid, Cloud)
  - Protect the testbed from experiments
  - Avoid network pollution

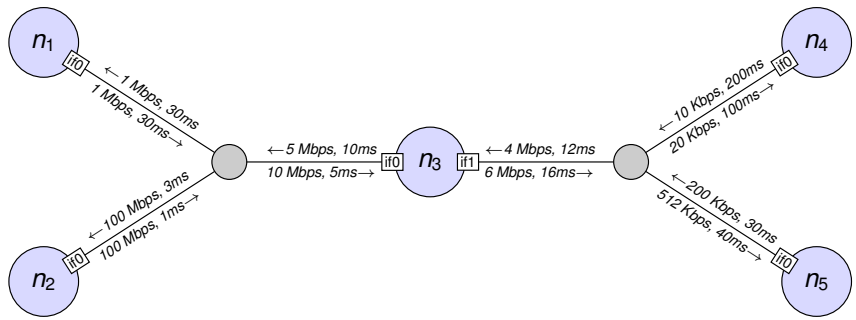
KADEPLOY



Customizing the experimental environment (2)

- Reconfigure experimental conditions with Distem *(ADT Solfege 2011-2013)*
  - Introduce heterogeneity in an homogeneous cluster
  - Emulate complex network topologies





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

Outline

Improving control and description of experiments

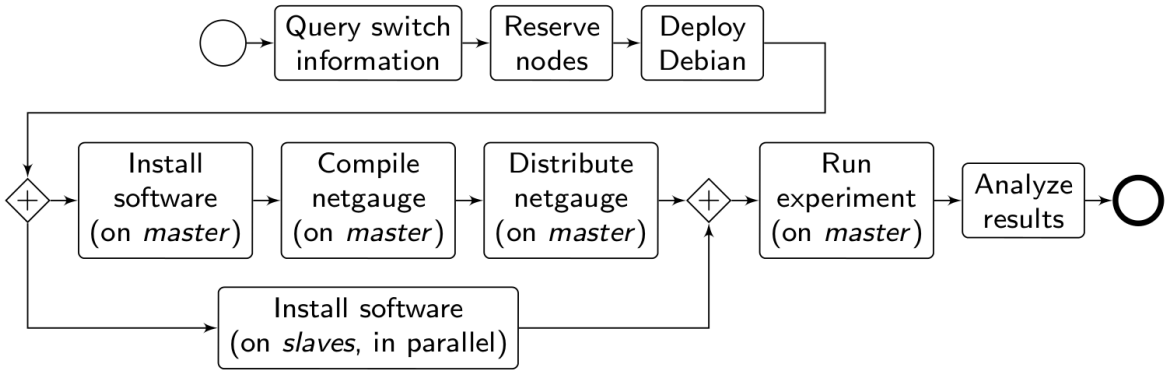
- Legacy way of performing experiments: shell commands
  - ☹ time-consuming
  - ☹ error-prone
  - ☹ details tend to be forgotten
- Promising solution: automation of experiments
  $\rightsquigarrow$ Executable description of experiments
- Support from the testbed: Grid'5000 RESTful API
  (Resource selection, reservation, deployment)

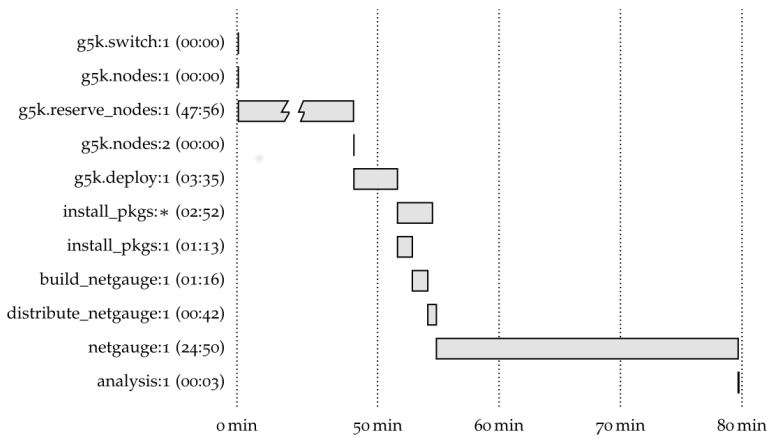
Expo (PhD Hemera Cristian Ruiz)

```
reserv = ExpoEngine::new(@connection)
reserv.site = [ "bordeaux", "lille", "luxembourg", "nancy", "sophia" ]
reserv.resources = [ "nodes=50", "nodes=10", "nodes=4", "nodes=4", "nodes=30" ]
reserv.name = "Expo Scalability"
reserv.walltime = 600
reserv.run!

sizes = [ 10, 20, 40, 50, 80, $all.length ]
$all.each_slice_array(sizes) do |nodes|
  task_mon = Task::new("hostname", nodes, " Monitoring #{nodes.length} nodes")
  10.times do
    id, res = task_mon.execute
    puts " #{res.length} : #{res.duration}"
  end
end

reserv.stop!
```

Scripting of experiments with useful & efficient abstractions

XPFlow (PhD Tomasz Buchert)



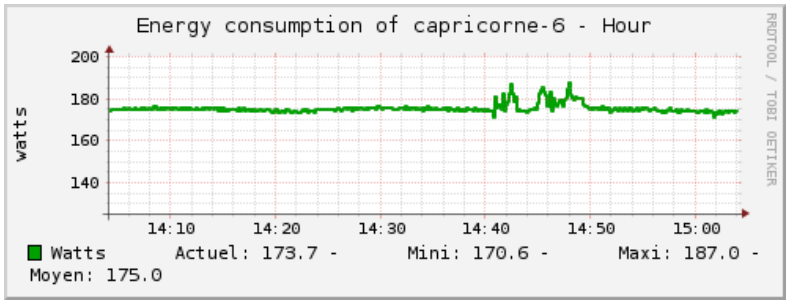



```
engine.process :exp do |site, switch|
    s = run g5k.switch, site, switch
    ns = run g5k.nodes, s
    r = run g5k.reserve_nodes,
        :nodes => ns, :time => '2h',
        :site => site, :type => :deploy
    master = (first_of ns)
    rest = (tail_of ns)
    run g5k.deploy,
        r, :env => '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
```

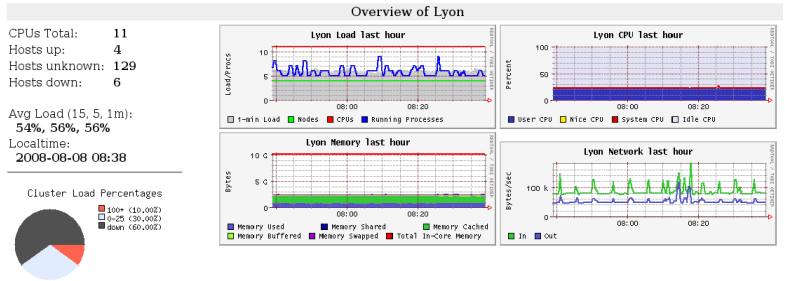
Experiment description and execution as a Business Process Workflow

Outline

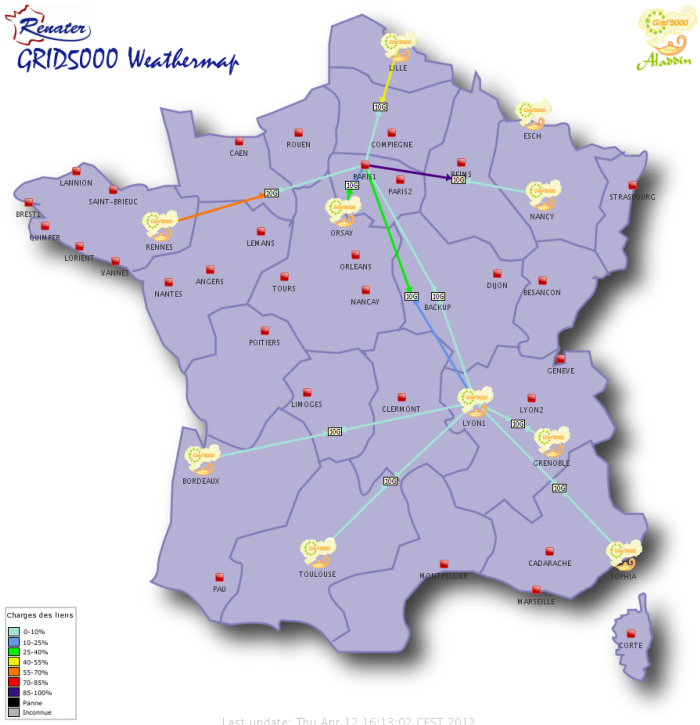
Monitoring experiments



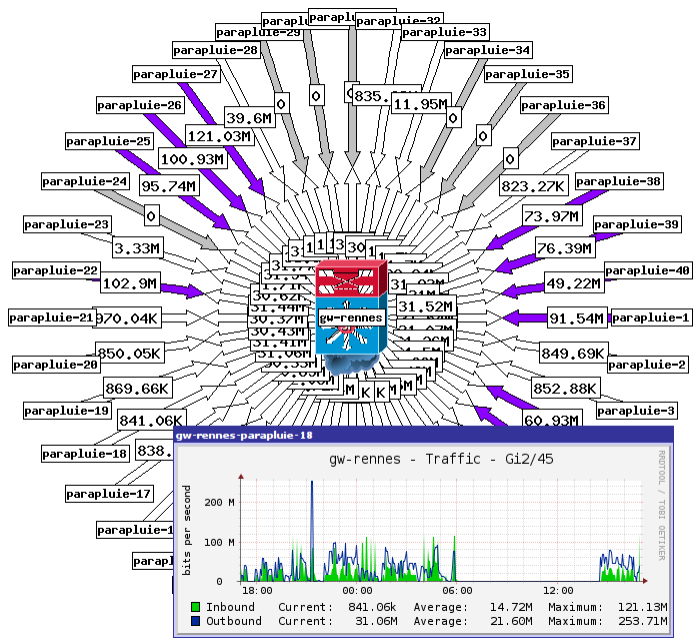
Power consumption



CPU – memory – disk



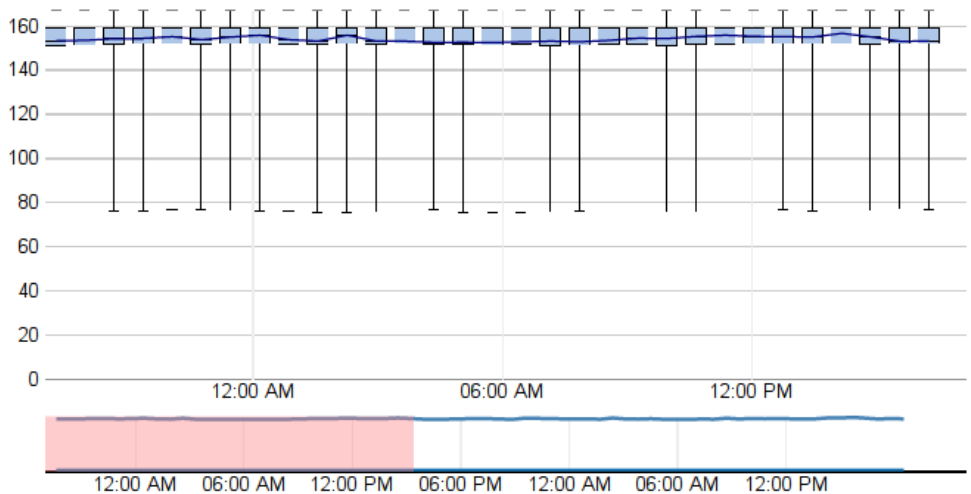
Network backbone



Internal networks

Exporting and analyzing data

- Unified access to monitoring tools through the Grid'5000 API
- Automatically export data during/after an experiment
  - Next step: executable papers?



Outline

Advocating good practices

- In the Grid'5000 community:
  - Best practices BOF during Grid'5000 schools
  - Challenges to demonstrate large-scale experiments

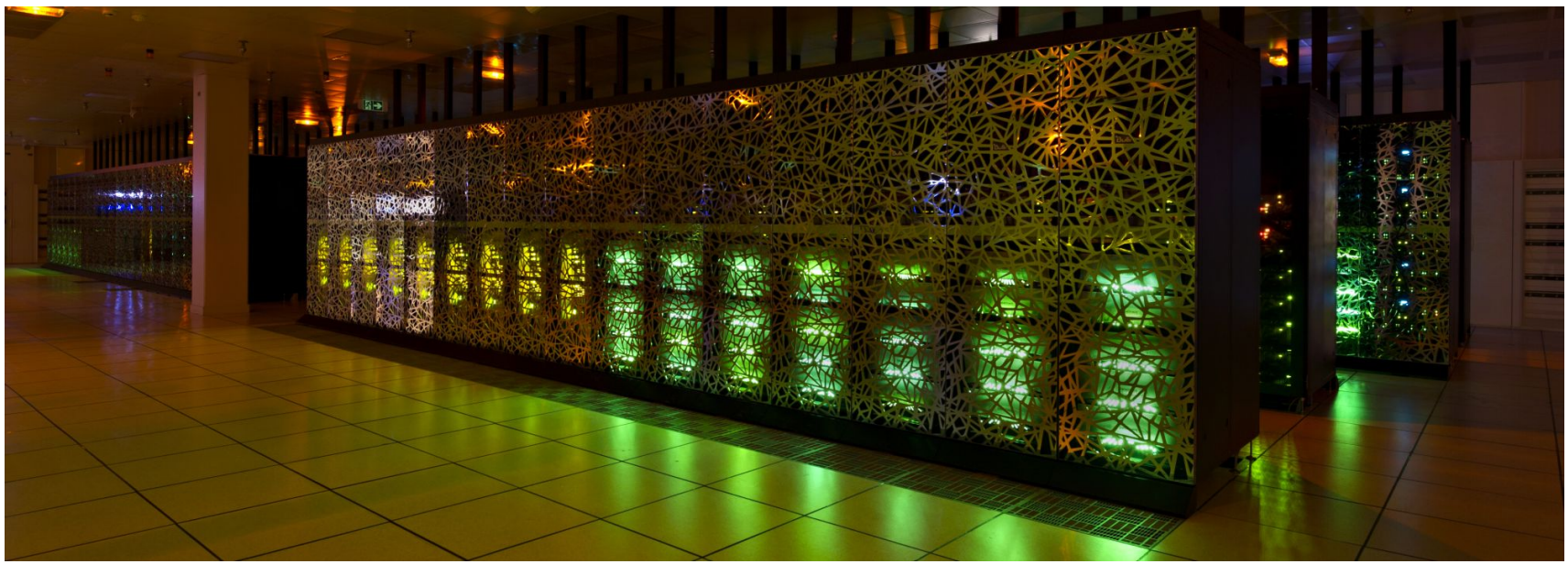
Advocating good practices

- In the Grid'5000 community:
  - Best practices BOF during Grid'5000 schools
  - Challenges to demonstrate large-scale experiments
- In the larger french community: Realis'2013
  - Goal: evaluate reproducibility of articles submitted to ComPAS
  - Process:
    - Authors submit their XP description to Realis
      How to describe an experiment enabling its reproduction?
    - Then (try to) reproduce another article's experiment
  - 9 submissions, 8 could be reproduced (but none without help)

Conclusions

- A lot has been done in Hemera towards high-quality experimental science
- At several levels: testbed, tools, methods
- But still a long way to go!

On pourrait déterminer les différents âges d'une science par la technique de ses instruments de mesure.[8]



[8] Gaston Bachelard, La formation de l'esprit scientifique, Vrin, 1938, p. 216