

Experiment Driven Research

Emmanuel Jeannot

INRIA – LaBRI

Ker Lann visit

Bordeaux, January 2011

My solution is better than yours

Efficiency	Workload A	Workload B	Workload C
My Solution	10	20	40
Your Solution	40	20	10

My Solution/Your Solution:

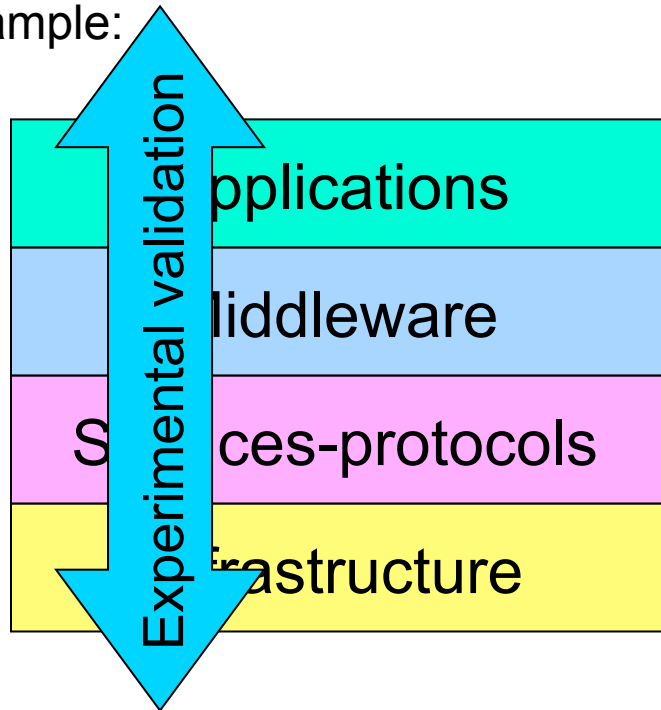
- Workload A: 0.25
- Workload B: 1
- Workload C: 4

Average: $5.25/3 = 1.75$

On average, My Solution is 1.75 more efficient than Your Solution

Environment Stack

•Example:



Research issues at each layer of the stack

- algorithms
- software
- data
- models
- ...

Problem of experiments

- Testing and validating solutions and models as a scientific problematic
- Questions:
 - what is a good experiments?
 - which methodologies and tools to perform experiments?
 - advantages and drawbacks of these methodologies/tools?

Outline

- Importance, role and properties of experiments in computer science
- Different experimental methodologies
- Grid'5000

The discipline of computing: a science

“The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis design, efficiency, implementation and application”

Peter J. Denning et al. COMPUTING AS A DISCIPLINE

- **Confusion:** computer science is not only science it is also engineering, technology, etc (≠biology or physic)
- Problem of **vulgarization**
- Two ways for **classifying knowledge:**
 - Analytic: using mathematics to tract models
 - Experimental: gathering facts through observation

The discipline of computing: an experimental science

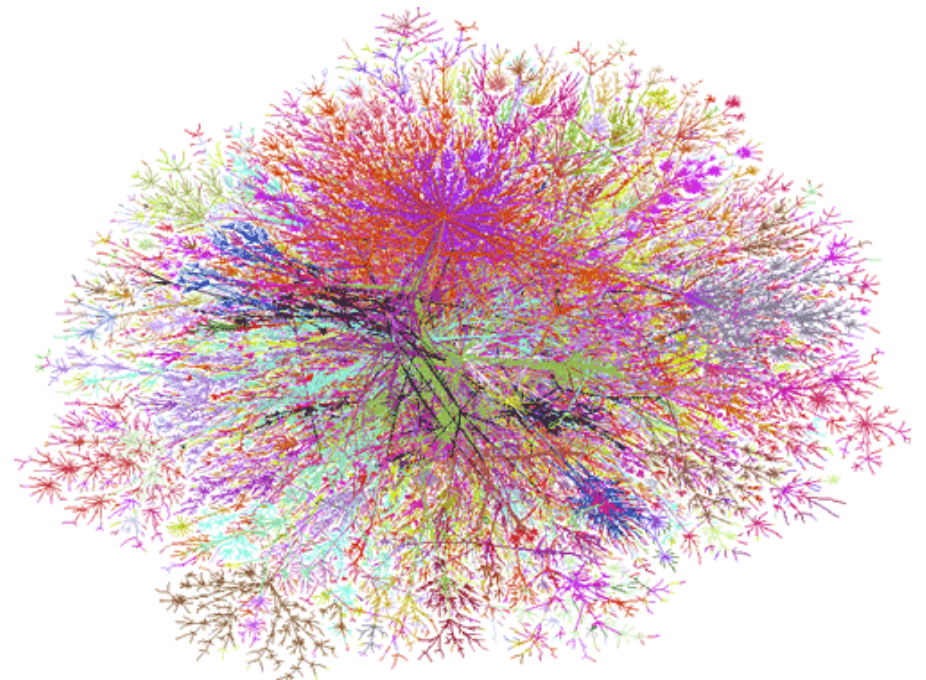
The reality of computer science:

- information
- computers, network, algorithms, programs, etc.

Studied objects (hardware, programs, data, protocols, algorithms, network): more and more complex.

Modern infrastructures:

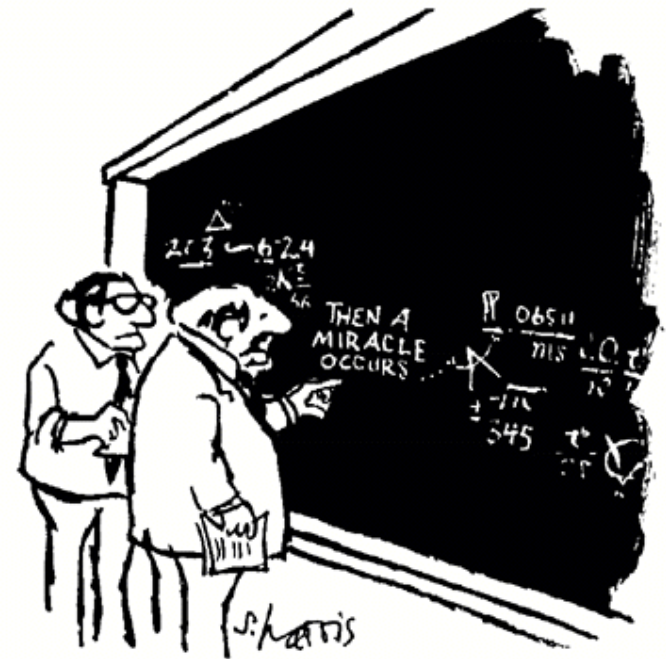
- Processors have very nice features
 - Cache
 - Hyperthreading
 - Multi-core
- Operating system impacts the performance (process scheduling, socket implementation, etc.)
- The runtime environment plays a role (MPICH≠OPENMPI)
- Middleware have an impact (Globus≠GridSolve)
- Various parallel architectures that can be:
 - Heterogeneous
 - Hierarchical
 - Distributed
 - Dynamic



Analytic modeling

Purely analytical (math) 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 culture: great successes

Experimental computer science at its best [Denning 1980]:

- Queue models (Jackson, Gordon, Newel, '50s and 60's). Stochastic models validated experimentally
- Paging algorithms (Belady, end of the 60's). Experiments to show that LRU is better than FIFO

ACM President's Letter

Performance Analysis: Experimental Computer Science at Its Best



Peter J. Denning

What is experimental computer science? This question has been widely discussed ever since the Feldman Report was published in 1979 [13]. Surprisingly, computer scientists disagree on the answer. Some believe that it is large system development projects, i.e., computer and software engineering. Some believe that it is all the nontheoretical activities of computer science, especially those conferring "hands-on" experience. Some believe that computer science is still too young and no operational definition is yet available.

I disagree. There are well-established standards for experimental science. The field of performance analysis meets these standards and provides the best examples of experimental computer science.

Hypotheses, Apparatus, and Tests

Science classifies knowledge. Experimental science classifies knowledge derived from observations. The experimenters set up an apparatus, use it to collect data about a phenomenon, and then analyze the data to sustain or refute hypotheses about

¹Part of this essay is based on my editorial, "What is Experimental Computer Science?" in *Communications of the ACM*, October 1980, pp. 543-544. The rest is based on my speech at the ACM SIGMETRICS Symposium on Performance Modeling, September 15, 1981.

725

that phenomenon. The result of one line of investigation may be a model that becomes the apparatus for a later line of investigation.

The experimental apparatus may be a real system, a subsystem, or a model. The hypothesis may concern a law of nature, characteristics of people, design principles of computers, or the quality of models.

The key affects of experimental science are an apparatus for collecting data, a hypothesis, and systematic analysis to see whether the data supports the hypothesis. There is considerable latitude in the types of hypotheses and apparatuses that may be used.

Two Examples

It is no accident that the best examples of experimental computer science can be found in the field called performance analysis. The primary aim of this field is the construction, validation, and evaluation of computer-system models that are robust enough to be used for prediction.

I will cite two examples of experimental science in this field. The first is the M44/44X project at the IBM T.J. Watson Research Lab in the middle 1960s; this project evaluated concepts of time sharing, especially

memory policies and program behavior, by implementing and measuring them. This example illustrates experimental work in computer systems architecture. The second example is the study of queueing network models since 1971; this line of investigation illustrates how strong interaction between theory and experiment can lead to a conceptually simple model that may serve as the starting point for future lines of investigation. This example illustrates how yesterday's theorems can become tomorrow's definitions.

The M44/44X Project

The M44/44X project was conducted at the IBM Research Center in Yorktown Heights, N.Y., in the middle 1960s. Its purpose was to evaluate the emerging concepts of time-sharing systems by reducing them to practice and measuring them. The central principle of its architecture was a set of virtual machines, one for each user. The main machine was an IBM 7044 (M44 for short) and each virtual machine was an experimental image of the 7044 (44X for short). Virtual memory and multiprogramming were used to implement the address spaces of the 44Xs in the memory hierarchy of the M44. This machine served as the

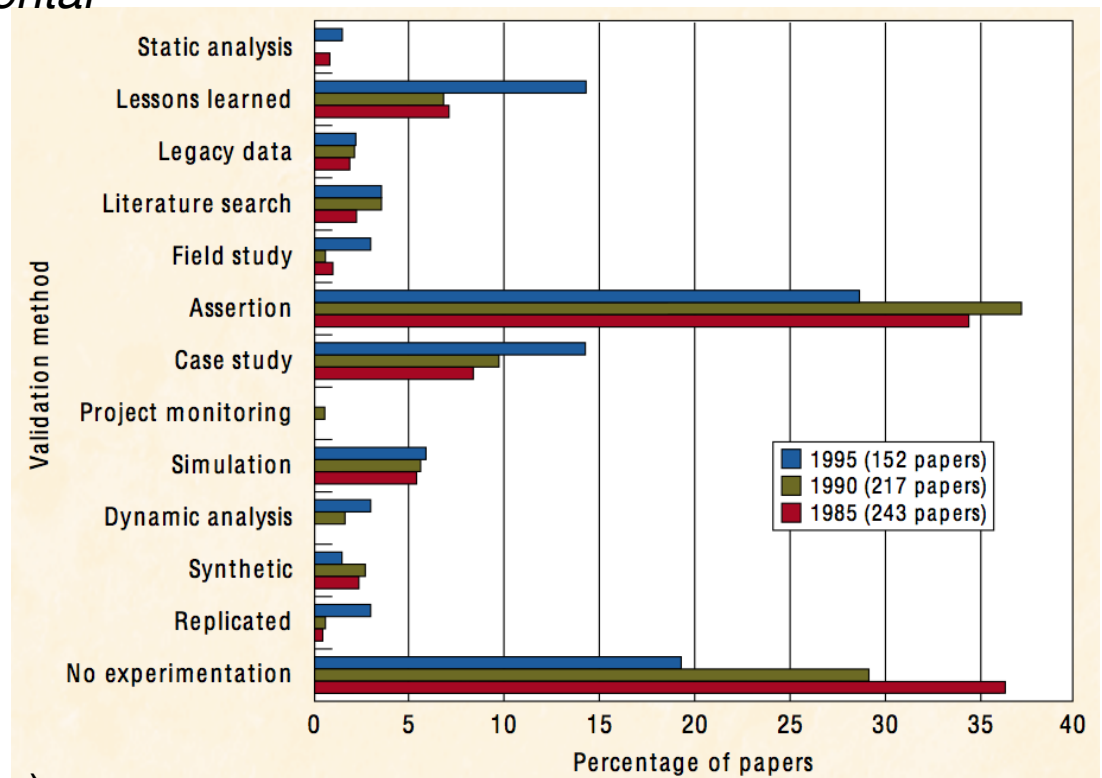
Communications
of
the ACM

November 1981
Volume 24
Number 11

Experimental culture not comparable with other science

Different studies:

- In the 90's: between 40% and 50% of CS ACM papers requiring experimental validation had none (15% in optical engineering) [Lukovicz et al.]
- “*Too many articles have no experimental validation*” [Zelkowitz and Wallace 98]: 612 articles published by IEEE.
- Quantitatively more experiments with times



M.V. Zelkowitz and D.R. Wallace. Experimental models for validating technology. *Computer*, 31(5):23-31, May 1998.

Computer science not at the same level than some other sciences:

- Nobody redo experiments (no funding).
- Lack of tool and methodologies.

Error bars in scientific papers

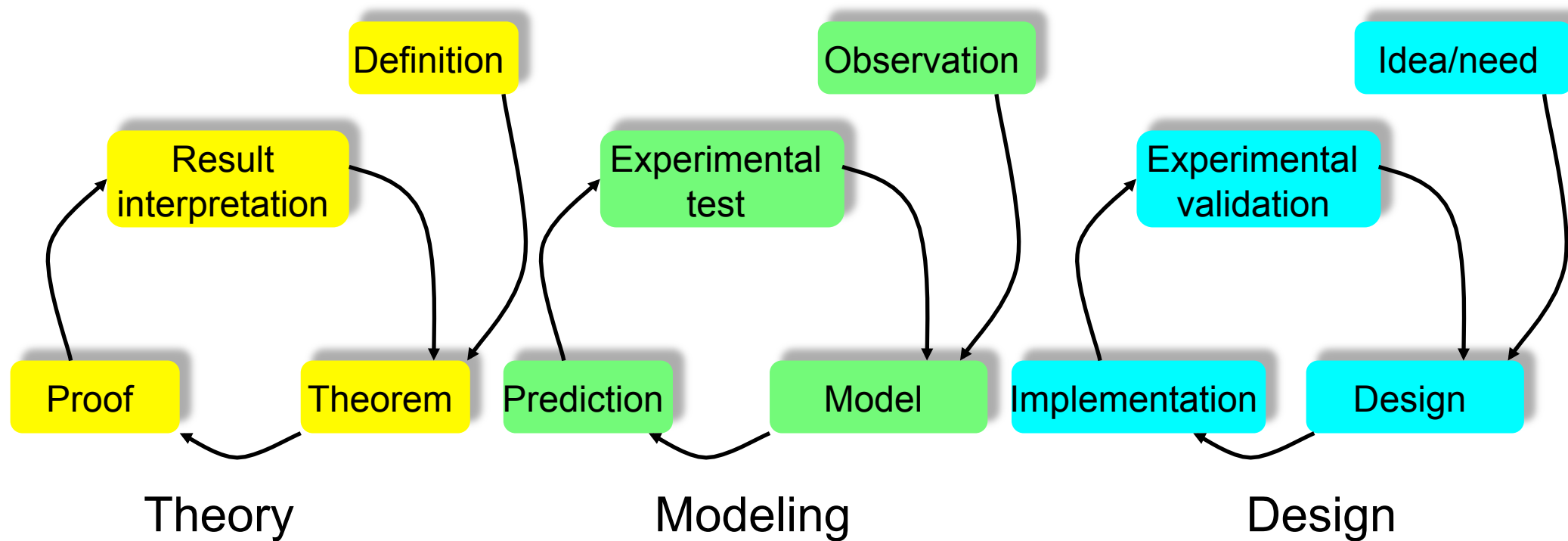
Who has ever published a paper with error bars?

In computer science, very few papers contain error bars:

Euro-Par	Nb Papers	With Error Bar	Percentage
2007	89	5	5.6%
2008	89	3	3.4%
2009	86	2	2.3%
3 last conf	264	10	3.8%

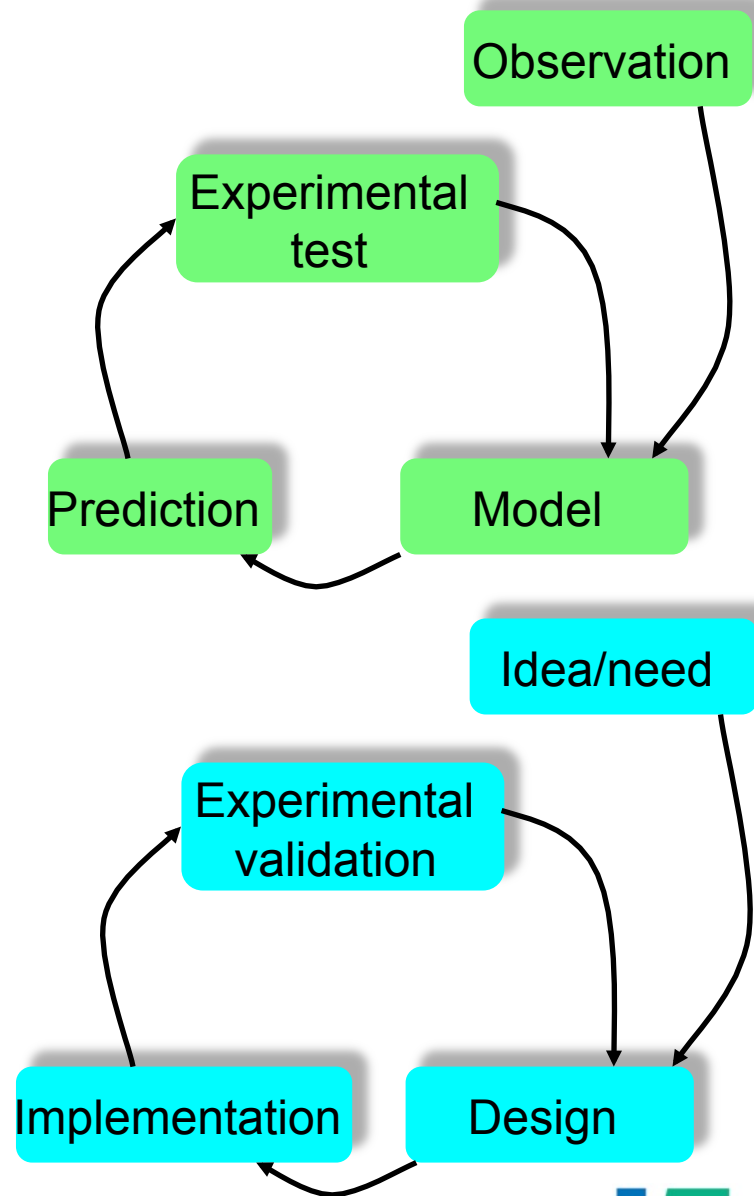
Three paradigms of computer science

Three feedback loops of the three paradigm of CS [Denning 89], [Feitelson 07]



Two types of experiments

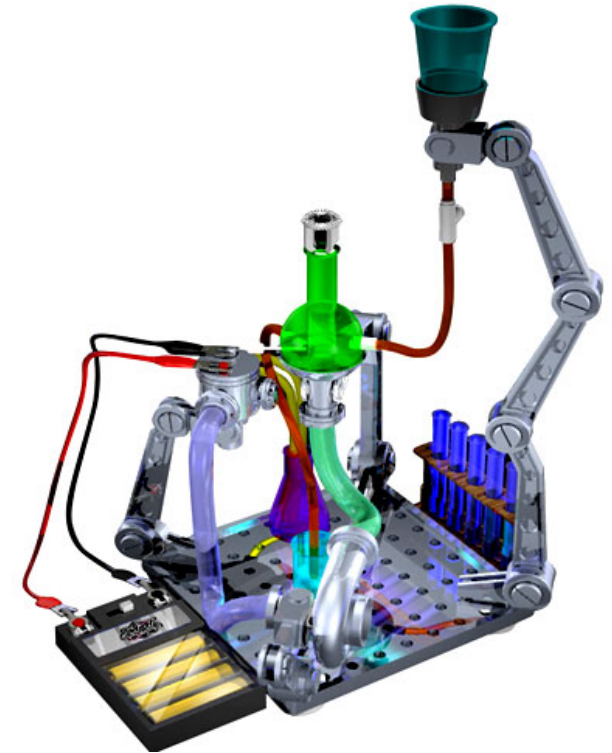
- Test and compare:
 1. Model validation (comparing models with reality)
 2. Quantitative validation (measuring performance)
- Can occur at the same time. Ex. validation of the implementation of an algorithm:
 - is the grounding modeling precise?
 - is the design correct?



Main advantages

[Tichy 98]:

- Experiments : testing hypothesis, algorithms or programs help construct a database of knowledge on theories, methods and tools used for such study.
- Observations: unexpected or negative results \Rightarrow eliminate some less fruitful field of study, erroneous approaches or false hypothesis.



“Good experiments”

A **good experiment** should fulfill the following properties [*Algorille project 05*]

- **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

- Importance, role and properties of experiments in computer science (and in grid computing)
- Different experimental methodologies
 - Simulation
 - Emulation
 - Benchmarking
 - Real-scale
- Tools for performing experiments

Experimental Validation

A good alternative to analytical validation:

- Provides a comparison between algorithms and programs
- Provides a validation of the model or helps to define the validity domain of the model

Several methodologies:

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

Properties of methodologies

Enabling good experiments:

Control:

- essential to know which part of the model or the implementation are evaluated
- allows testing and evaluating each part independently

Reproducibility:

- base of the experimental protocol
- ensured by experimental environment

Realism:

- Experimental condition: always (somehow) synthetic conditions
- Level of abstraction depends on the chosen environment
- Three levels of realism:
 1. **Qualitative:** experiment says $A_1 \geq A_2$ then in reality $A_1 \geq A_2$
 2. **Quantitative:** experiment says $A_1 = k * A_2$ then in reality $A_1 = k * A_2$
 3. **Predictive.**
- Problem of validation

Simulation

Simulation [Quinson 08]: predict parts of the behavior of a system using an approximate model

- Model = Collection of attributes + set of rules governing how elements interact
- Simulator: computing the interactions according to the rules

Models wanted features:

- Accuracy/realism: correspondence between simulation and real-world
- Scalability: actually usable by computers (fast enough)
- Tractability: actually usable by human beings (understandable)
- *“Instanciability”*: can actually describe real settings (no magic parameters)

⇒Scientific challenges



Emulation

Emulation: executing a real application on a model of the environment



Two approaches:

- Sandbox/virtual machine: confined execution on (a) real machine(s). syscall catch. Ex: MicroGrid
- Degradation of the environment (to make it heterogeneous): direct execution. Ex: Wrekavoc

Benchmark

Synthetic application:

- Test workload
- Model of a real application workload
- Shared by other scientists
- Do not care for the output (e.g. random matrix multiplication).

Classical benchmark:

- NAS parallel benchmarks (diff. kernels, size and class).
- Linpack (Top 500)
- SPEC
- Montage workflow
- Archive:
 - Grid Workload archive (GWA)
 - Failure trace archive (FTA)

In-situ/Real scale

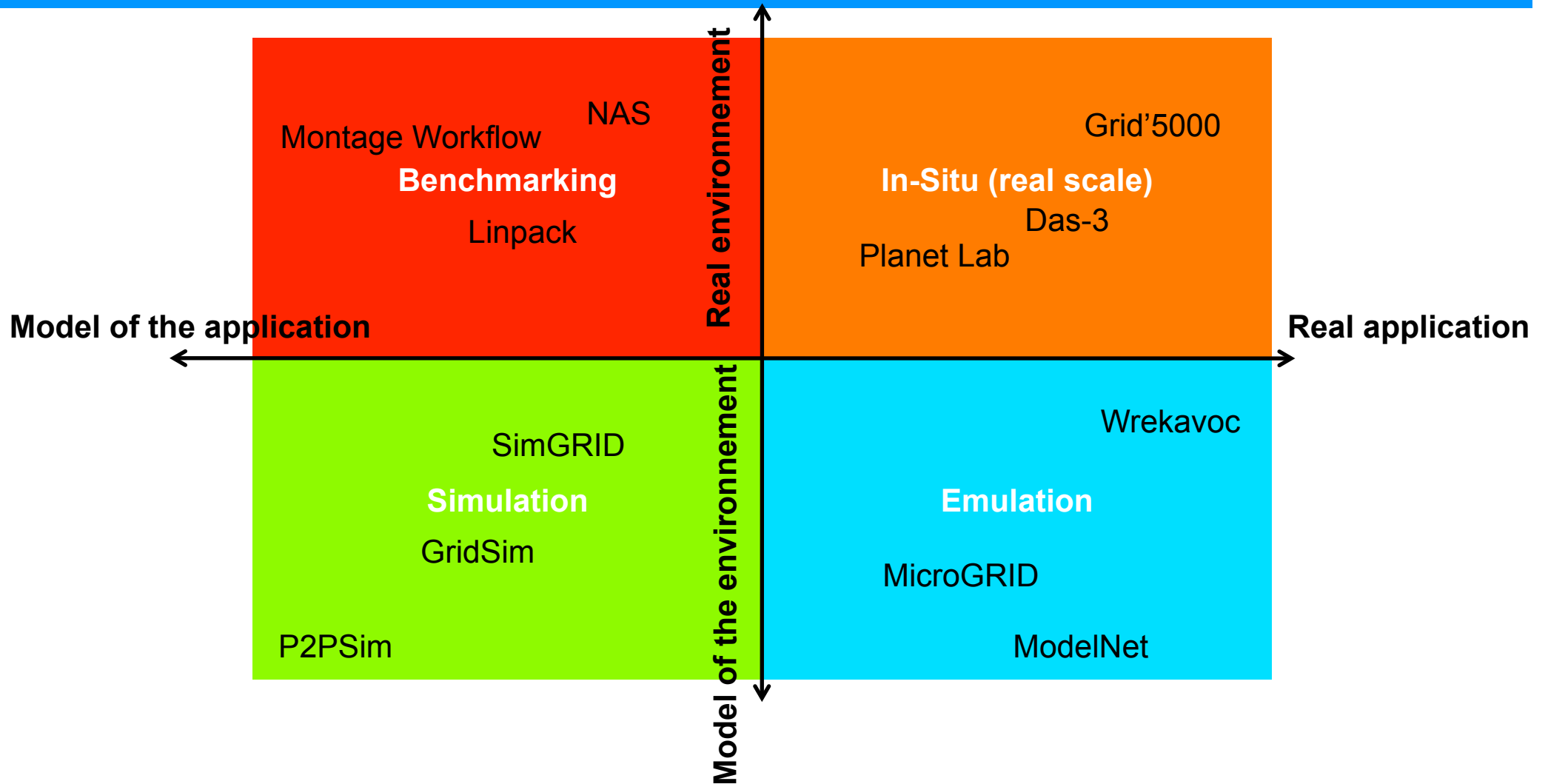
Real application executed on real (dedicated) hardware/
environment

Challenges:

- Configuration
- “*Genericity*”
- Experiment cycle time
- Ease of use, cost



A unified Taxonomy [GJQ09]



Warning: running a benchmark on an emulator is different than doing a simulation

See: J. Gustedt, E. Jeannot and M. Quinson **Experimental Methodologies for Large-Scale Systems: a Survey.** PPL, 19(3):399–418, September 2009

Grid'5000

A large-scale experimental platform

Large-scale systems more and more complex

Modeling is hard/intractable/impossible

Test/compare/evaluate solution in this domain

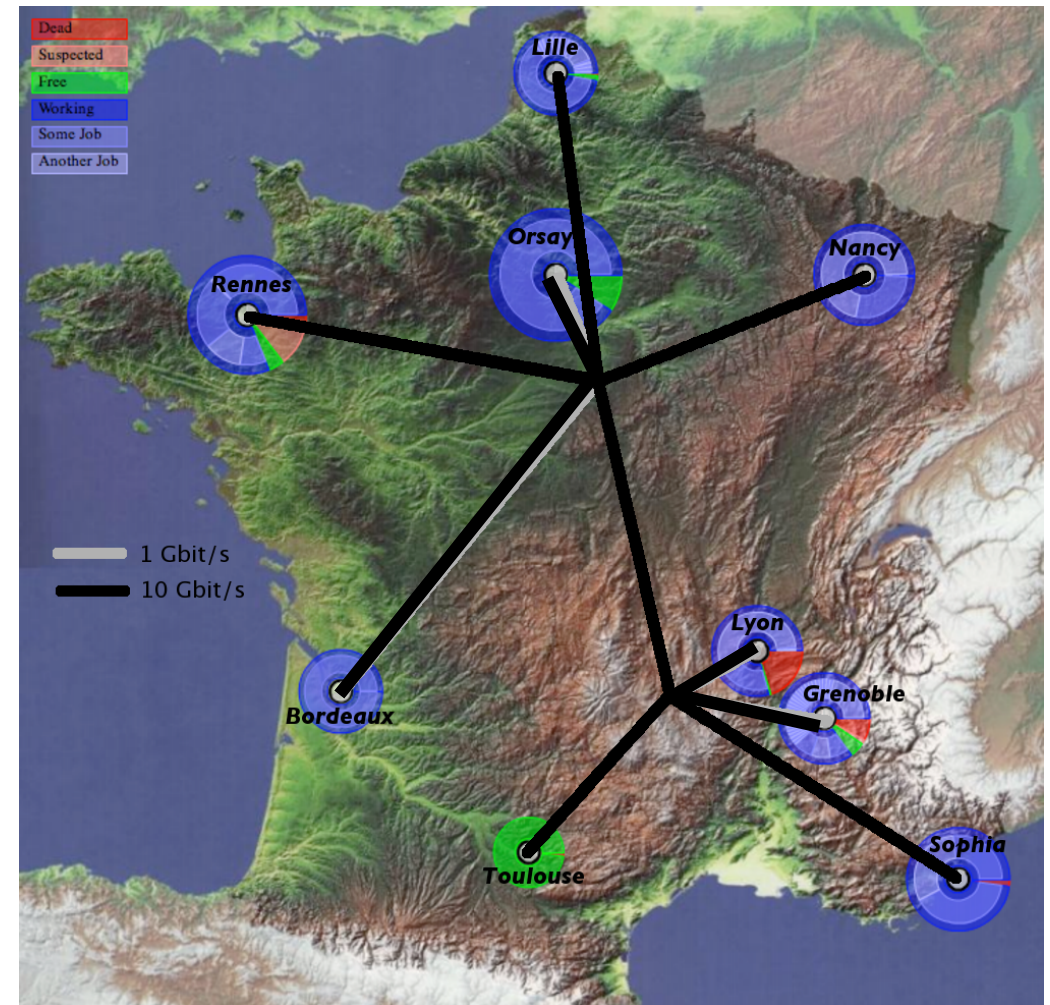
Perform “good experiments” in the domain of parallel and large-scale system.

Grid'5000

- 9 sites across France
- 1 site in Luxemboug
- 1 site in Brazil
- Each sites host one to 3 clusters (3202 proc, 5714 cores).
- Reservation system
- **Possibility to custom its environment, to save it and resume it.**

Different types of experiment (late 2009):

- 108 in network (monitoring, redistribution, QOS, P2P, etc)
- 189 in middleware (JXTA, DIET, fault-tolerance , MPI,...)
- 66 in programming (component, coarse-grain,...)
- 163 in application (bio-info, climate, geology,...)
- 53 divers (tools test (kadeploy), javaspace,)



Kadeploy

Fast and scalable deployment system towards cluster and grid computing.

Tools for:

- cloning
- configuring (post installation)
- managing a set of nodes.



Save, resume and distribute a system image.

Ensures experiment reproducibility.

<http://kadeploy.imag.fr/>

Bordeaux

- 1. History
 - Site open since october 2005.
- 2. Hardware
 - 3 clusters :
 - Bordeplage: 51 INTEL Xeon (bi-pro, 1 core), Infiniband 10G
 - Bordereau: 93 AMD Opteron (bi-pro, dual-core), Gb Ethernet
 - Borderline: 10 AMD Opteron (Quad-pro, Dual-core, 32 Gb/nodes),
 - GB Ethernet + Myrinet 10G + IB 10G
 - Total: 154 nodes
- Link 10 Gbit avec les autres sites par Renater 5.
- 3. Usages
 - 87 users
 - Experiments in netwrok (ex : myrinet) ;
 - High Performance computing(ex : sparse linear algebra) ;
 - application (ex : bioinformatics) ; algorithmic (ex : scotch), etc.
- 4. Perspectives
 - A GPGPU cluster?

Conclusion

Computer-Science is also an experimental science

There are different and complementary approaches for doing experiments in computer-science

Computer-science is not at the same level than other sciences

But, things are improving...