Modeling Large Scale Systems and Validating their Simulators

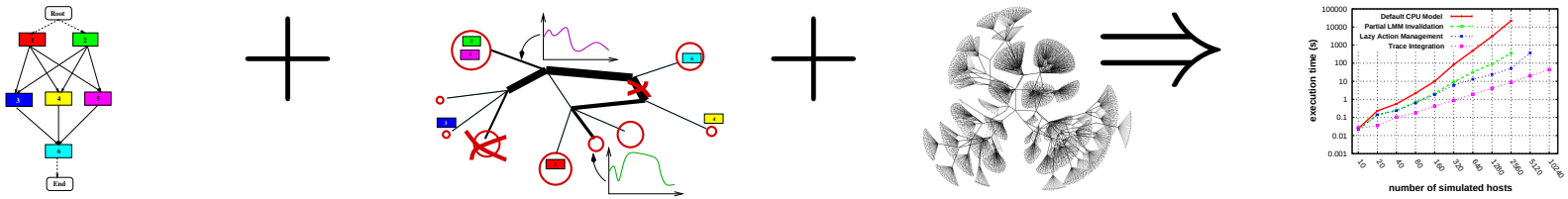
Martin Quinson, Arnaud Legrand
(and everyone else)

Hemera Evaluation
February 11, 2013

SimGrid: Simulator of Distributed Applications

Scientific instrument for the study of large scale distributed computing

Idea to test + Experimental setup + Model ⇒ Scientific Results



Main Features

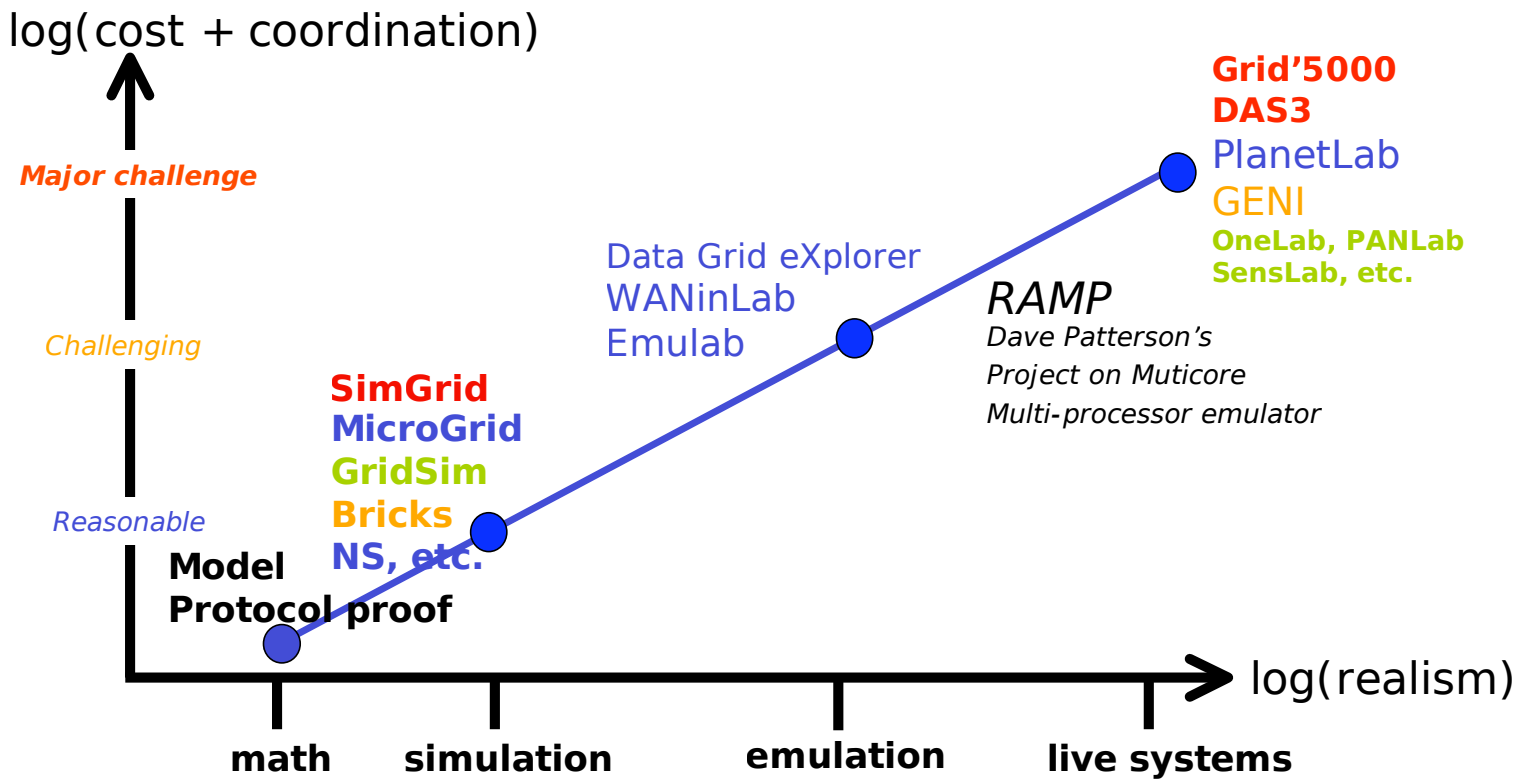
- Versatile: Grid, P2P, HPC, Volunteer Computing, Clouds, ...
- Valid: Accuracy limits studied and pushed further for years
- Scalable: 3M chord nodes; 1000× faster than other (despite precise models)
- Usable: Tooling (generators, runner, visu); Open-source, Portable, ...

2008-2013 Facts

- 63 publications (98 distinct authors, 8 Inria teams and 4 continents), 4 PhD
- 4 EPIs contributed (+ 4 EPIs just joined through ANR); 1 ODL + 1 ADT
- Open-Source project: 26 distinct commiters, 5 "unaffiliated contributors"

The Accuracy vs. Speed tradeoff

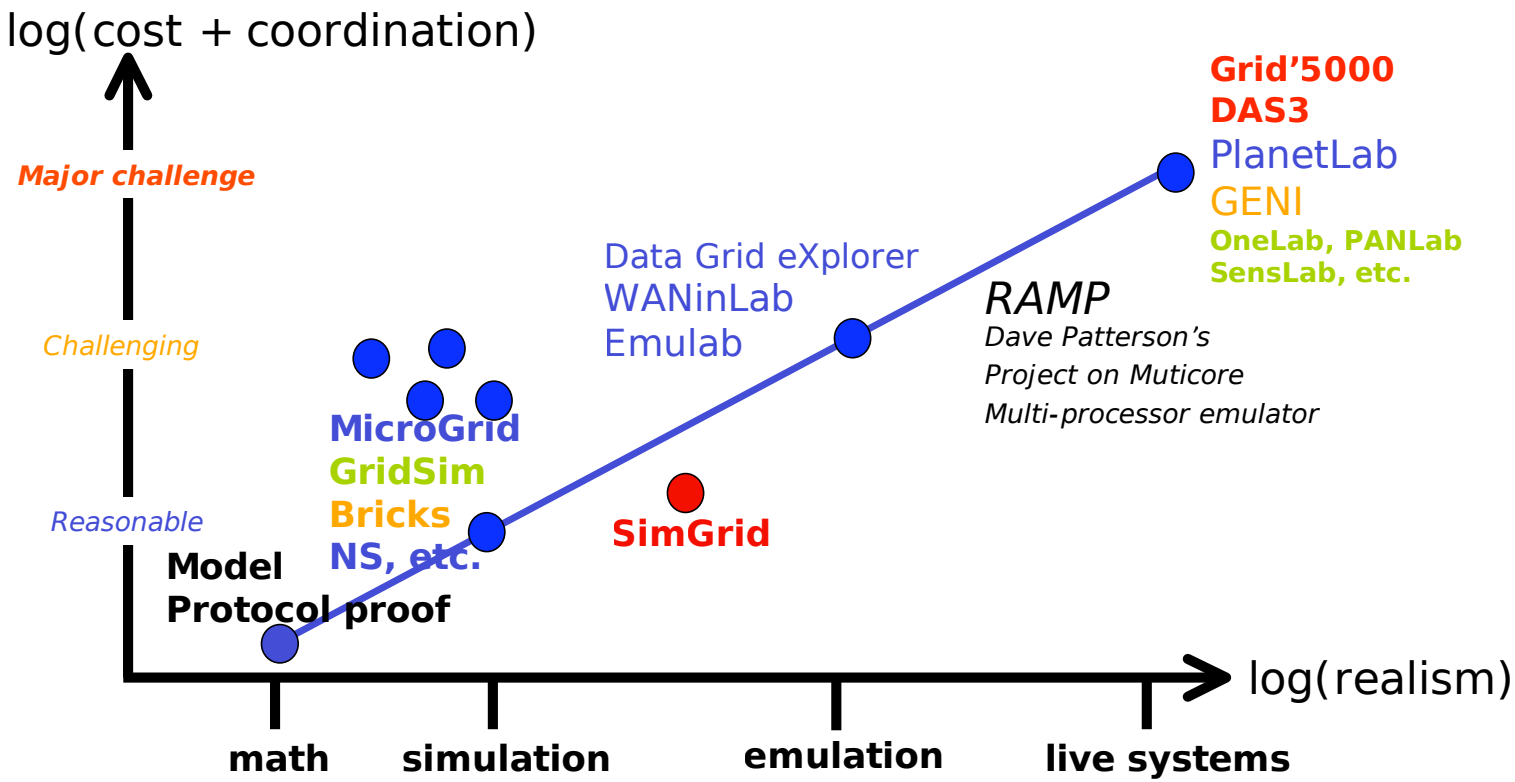
- Common Belief in 2008: Simulation as a toy methodology



Courtesy of Franck Cappello (Gri5000 keynote @ EGEE, Feb 2008 :)

The Accuracy vs. Speed tradeoff

- Common Belief in 2008: Simulation as a toy methodology
- Consensus in 2013: SimGrid as a scientific instrument (w/ Grid'5000)



Purpose of this Talk

How did we turn **Simulation** into a
Reliable and Versatile Scientific Instrument
for Research in Distributed Computing?

- A Fast and Versatile Simulation Kernel
  - ⇝ Using Grid'5000
- Validating our Models in a Wide Range of Settings
  - Simulating Real MPI Applications (beyond prototypes)
  - Simulating Map Reduce
  - ⇝ On Grid'5000 (and could not have been done elsewhere)
- Toward a Coherent Workbench for Distributed Applications
  - ⇝ With Hemera members

Simulating MPI Applications

Many advantages:

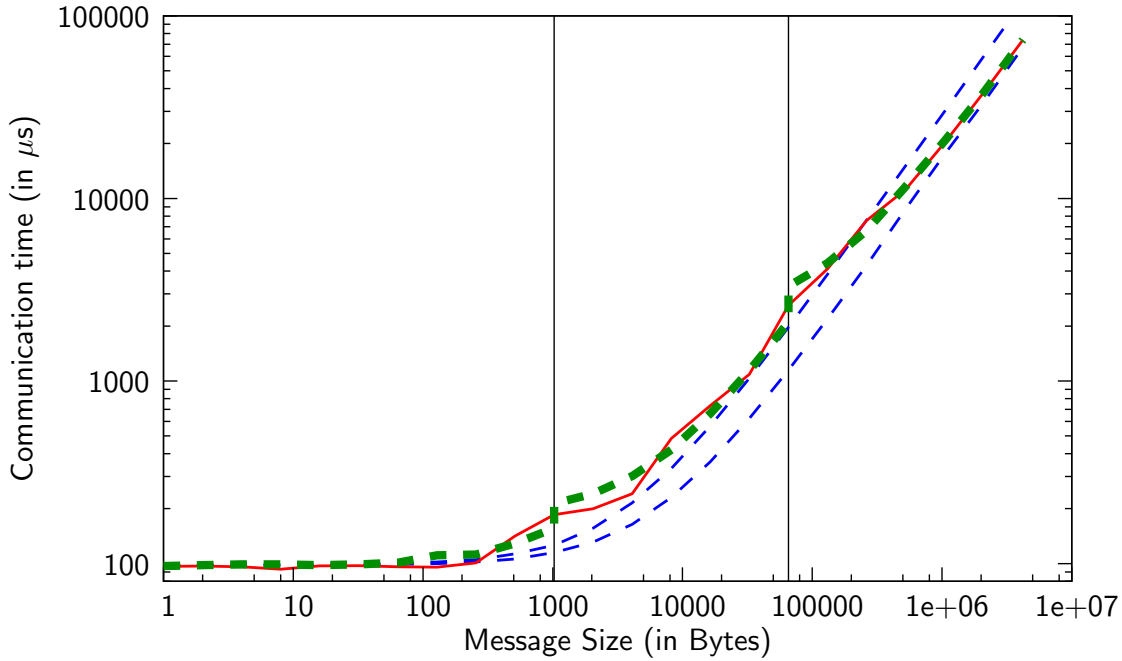
- No need for a real cluster/supercomputer to get a feeling on outcomes
- Skip computation intensive parts of the code
  $\leadsto$ quickly evaluate the influence of code modification
- Reproducible tests over a wide range of scenarios
- What-if study the impact of heterogeneity, variability, resilience, bw, ...
- Stop wasting computing hours & Watts to test a tiny modifications
- Easily conduct scalability studies
- Trace without intrusivity (bye bye heisenbugs)
- ...

Many other tools with similar goals but often limited or difficult to use.

Perfect workbench to validate the simulation models

Modeling of Point-to-Point Communications

Comparing average point-to-point comms time with linear approximations



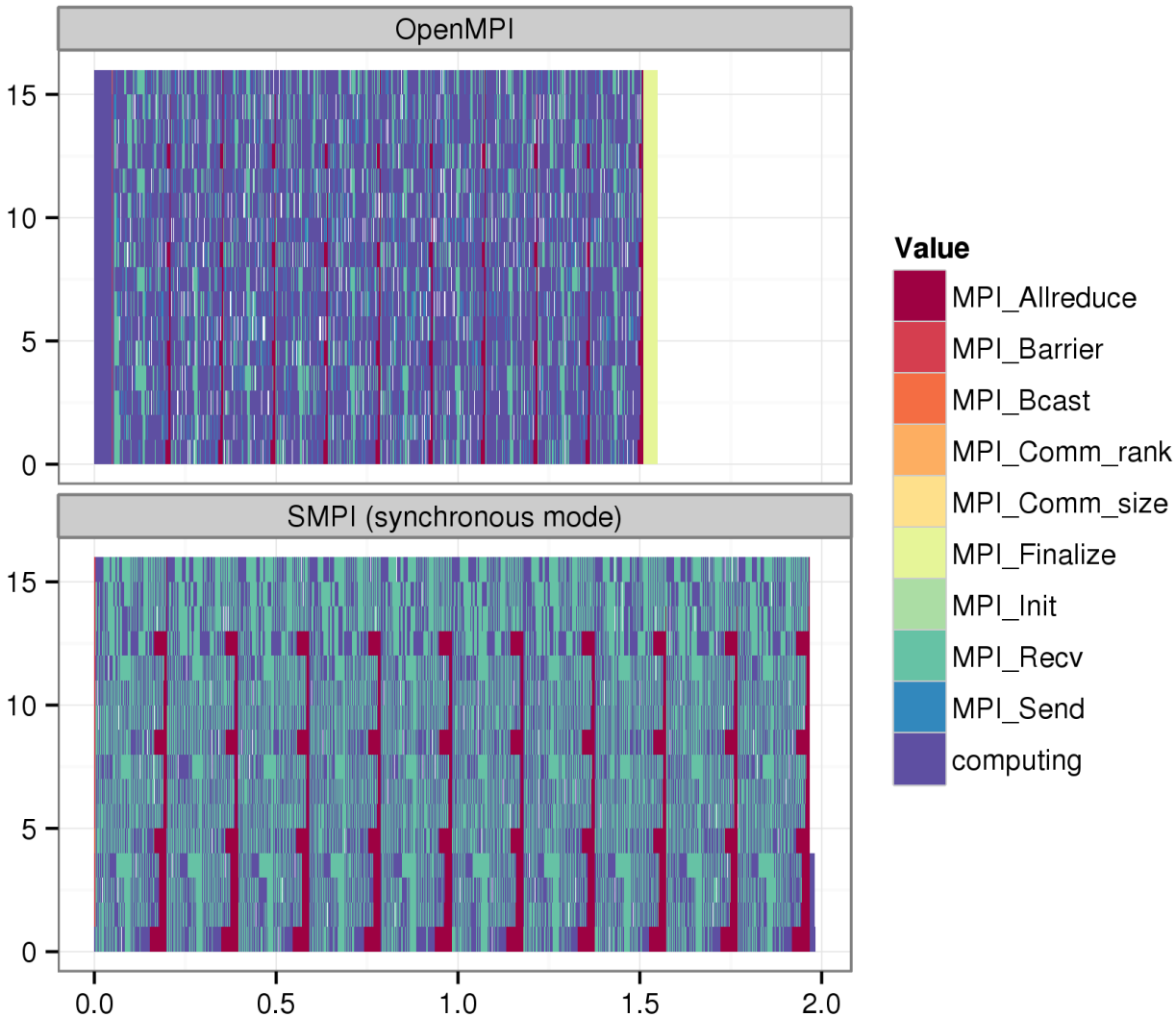
- No linear function matches the real measurement on the whole interval
- Piece-wise modeling is essential
- That lead to a very realistic modeling of point-to-point communications

Comparison on a Real Application

Methodology to compare a real execution with a simulation

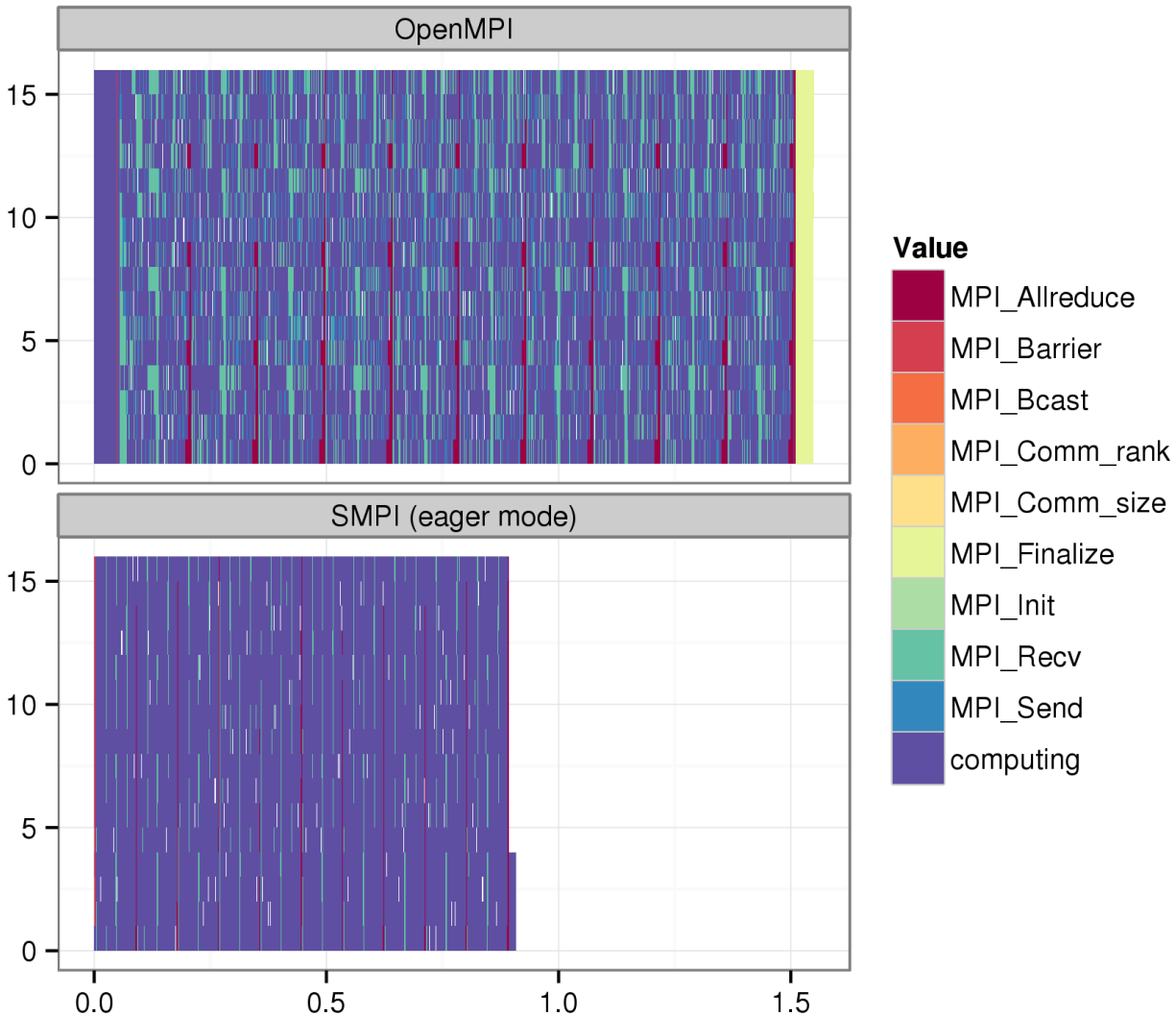
- Sweep3D: many small messages with a complex communication pattern (both point-to-point and collective operations)
- Griffon (Nancy), OpenMPI, TCP, Gigabit Ethernet
- Instrument real application with Tau or Akypuera
- Compare trace of real system with trace of SMPI using R

Initial Comparison



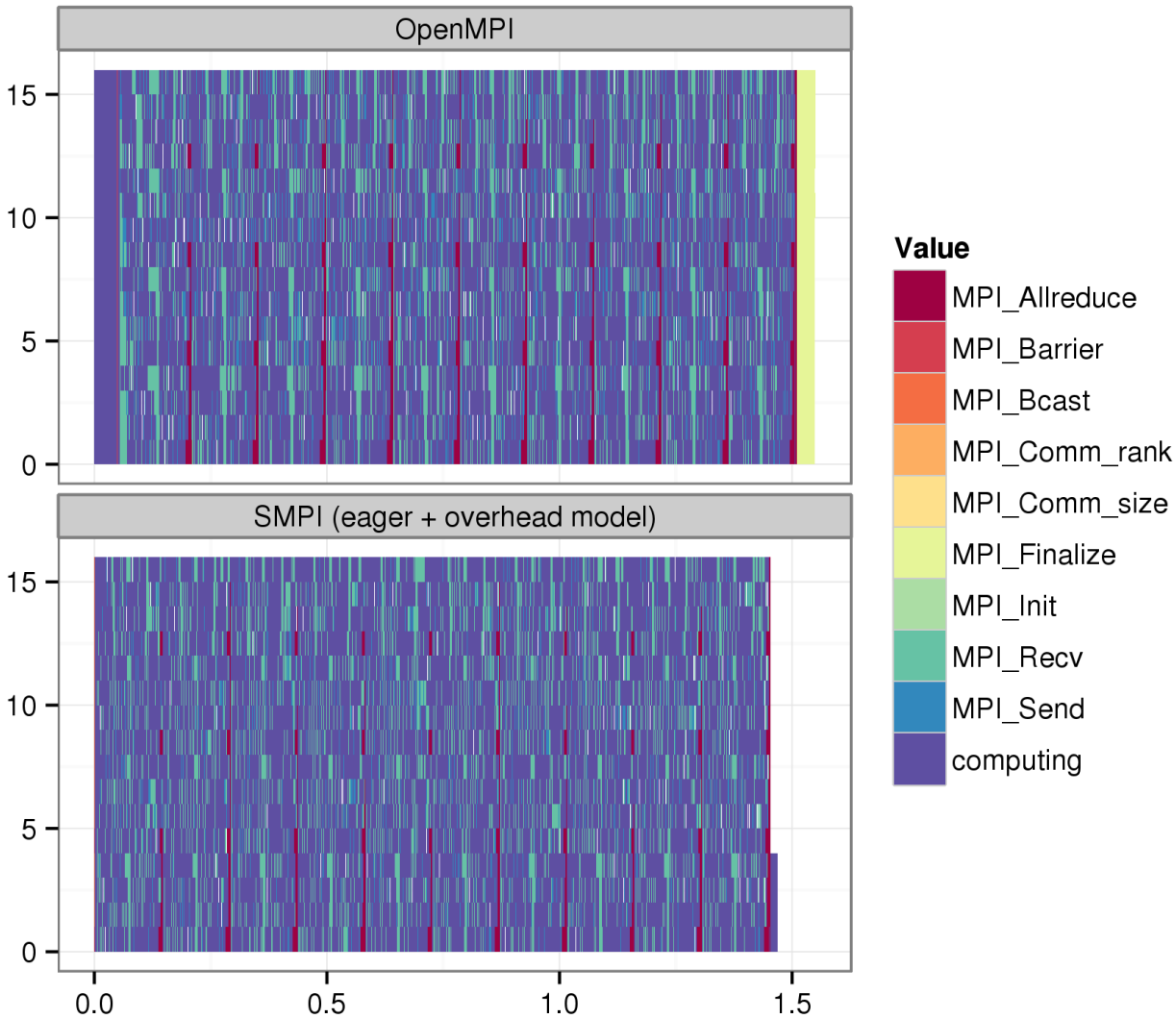
- Pessimistic timings despite accurate estimation of point-to-point comms
- Need to account for **eager** mode!

Accounting for Eager Mode



- Now overly optimistic! `MPI_Send` are not instantaneous...
- The **overhead** of syscalls and memory copies is not negligible.

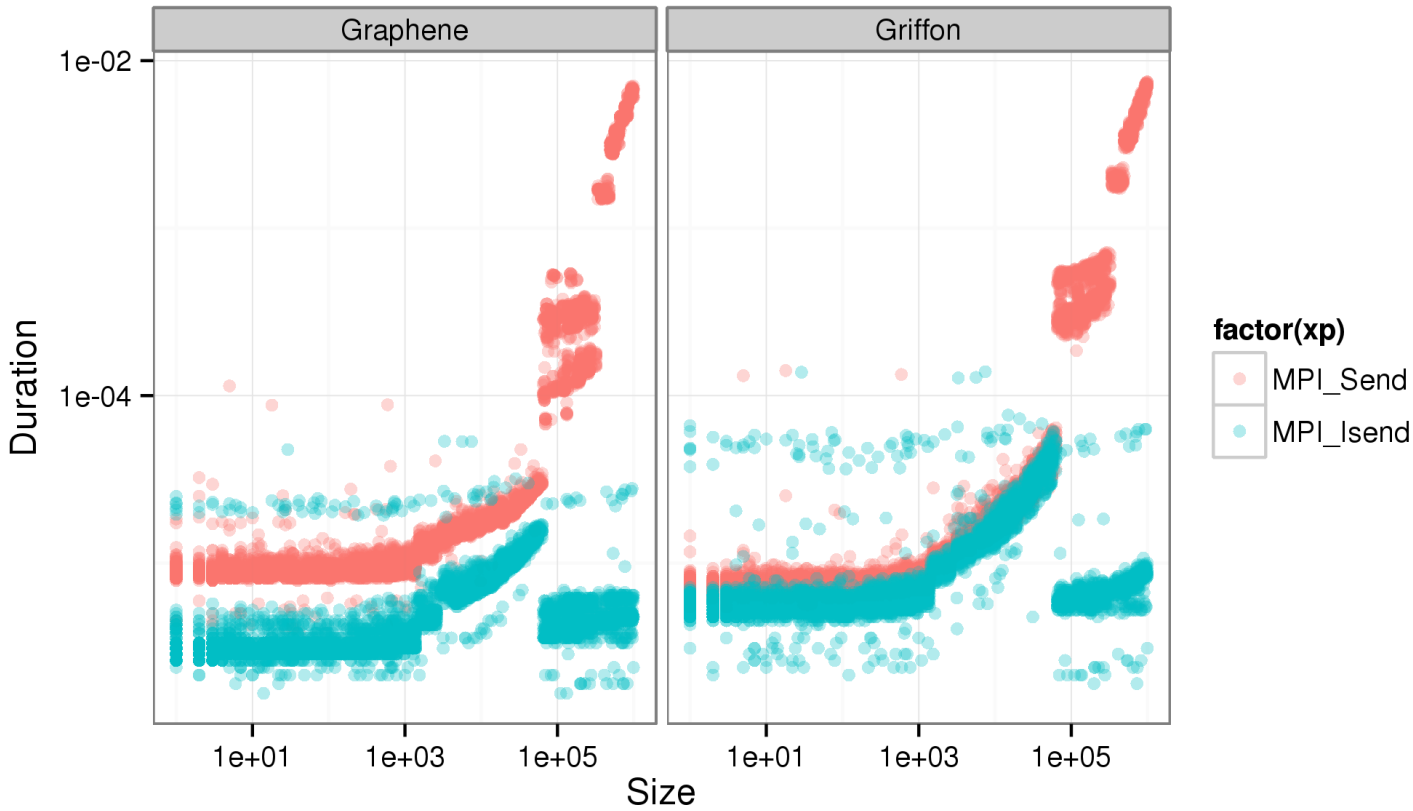
Accounting for Overhead



- Ok now, but beware, simple modeling error $\rightsquigarrow$ gross inaccuracies
- Hidding errors: Consider makespan only; overfit model parameters

MPI Oddities and Cluster Peculiarities

Instead, we look forward an accurate modeling of important phenomena
Goal: obtain sound predictions over a **wide range of settings**



- Protocol switch (1500, 65k, 327k, . . .),
- Noisy areas and complex synchronization
- New distinctions (e.g., MPI_Send vs. MPI_Isend for small messages) appear when changing cluster

Current Investigation

- Accurate All2All prediction $\Leftarrow$ accurate modeling of contention
- Clusters are generally organized in cabinets and contention may occur within or between cabinets
- To determine switch and link capacity, we need to increase workload up to saturation
- With Grid'5000, we get exactly the right nodes without external noise

Simulating the MapReduce Framework

Many advantages:

Developer Perspective

- Easily prototype new scheduling/file management strategies
- Reproducible tests over a wide range of scenarios
- Study the impact of heterogeneity, variability, resilience, bandwidth

User Perspective

- Find the best map-reduce configuration (mappers, workers,...) for a given setting (cluster×application)
- Model the cost of moving data in and out the cloud
- Determine how many resources to request for a given application

MapReduce-SimGrid has been developed at UFRGS (Brazil).
In the simulation tasks do not execute a real application. Rather, they are defined as a cost measured in FLOPs, hence the need to study and characterize such costs.

Running MapReduce on G5K

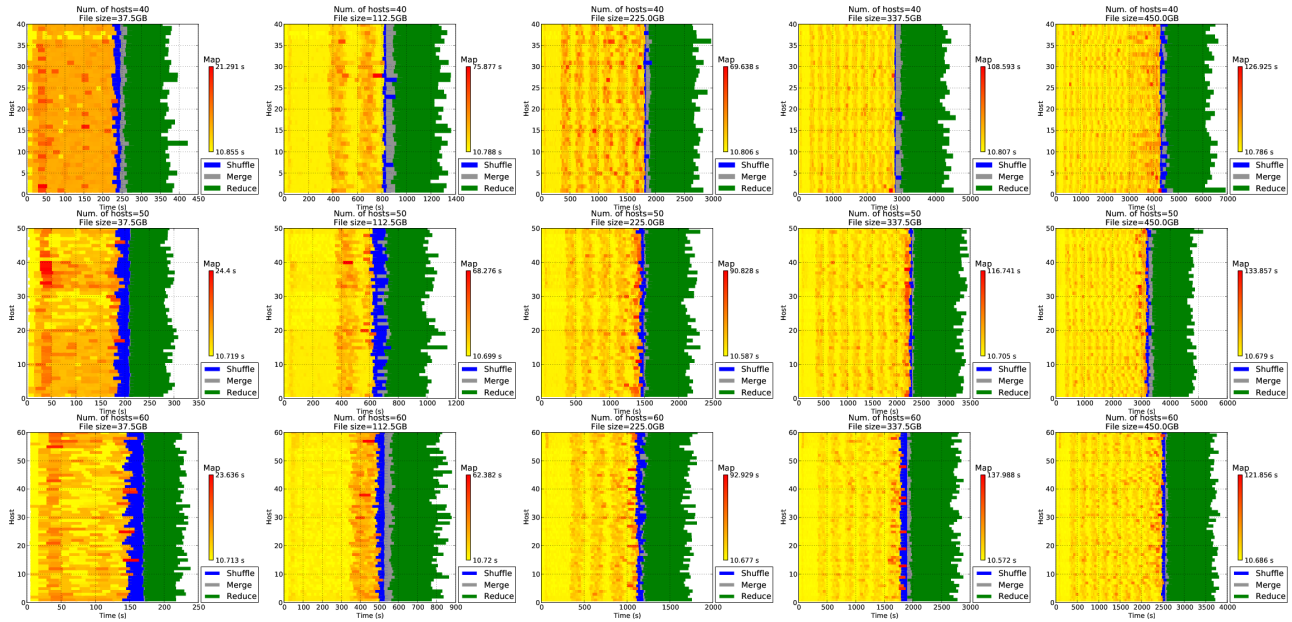
Grid'5000 resources: gdx (Orsay)

Hadoop configuration:

- Number of map and reduce tasks = Number of hosts
- Number of chunk replicas : 1

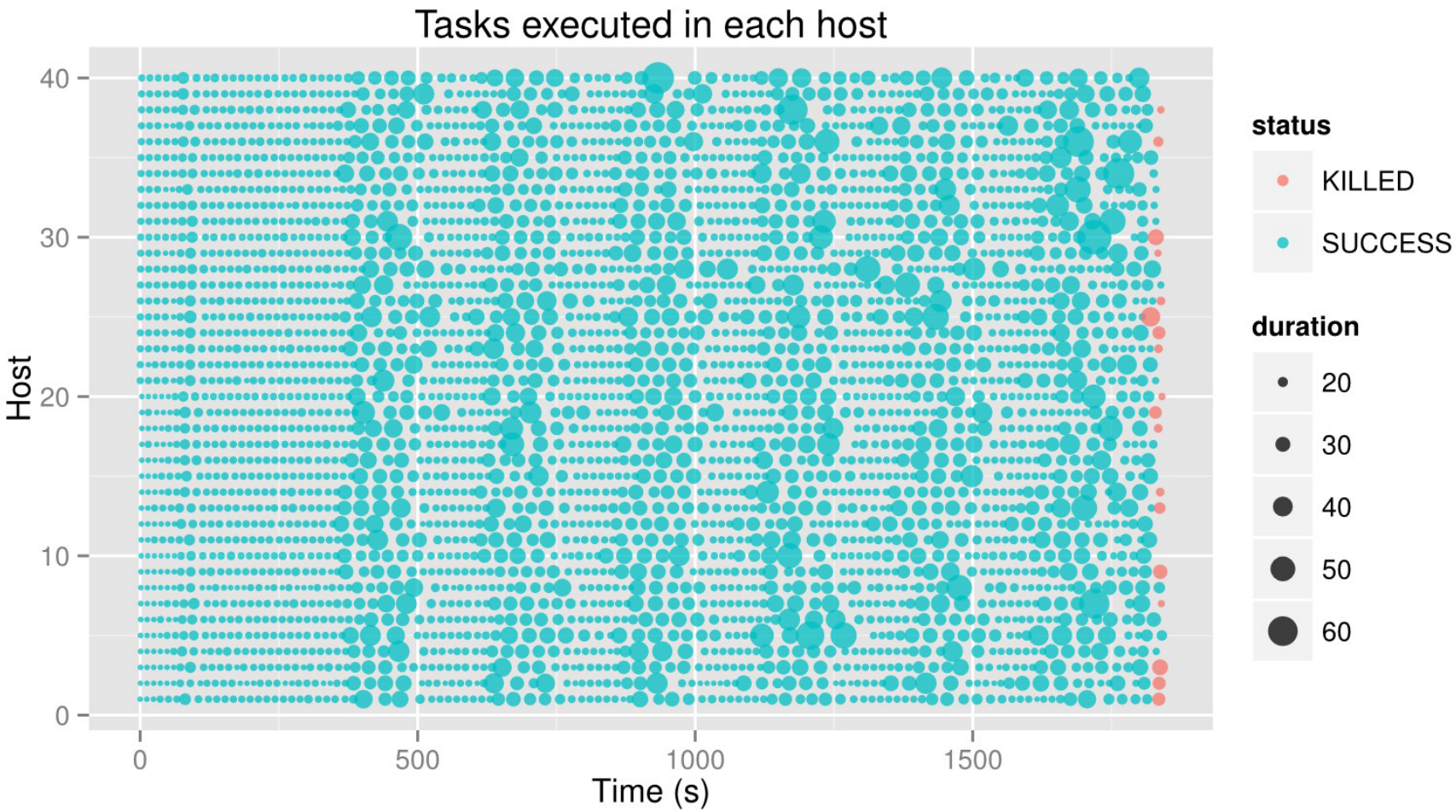
Workload:

- Number of hosts: 40, 50, and 60
- Input file size: ranging from 37.5GB to 450GB
- Application: TeraSort



(Experiment performed by Borja Bergua – 2011)

Closer Look at this Unexpected Behavior



- Map task duration should be roughly the same for all tasks
- Many map tasks suffer the slowdown at the same time
  Even when executing in different hosts

Reproducibility Issues

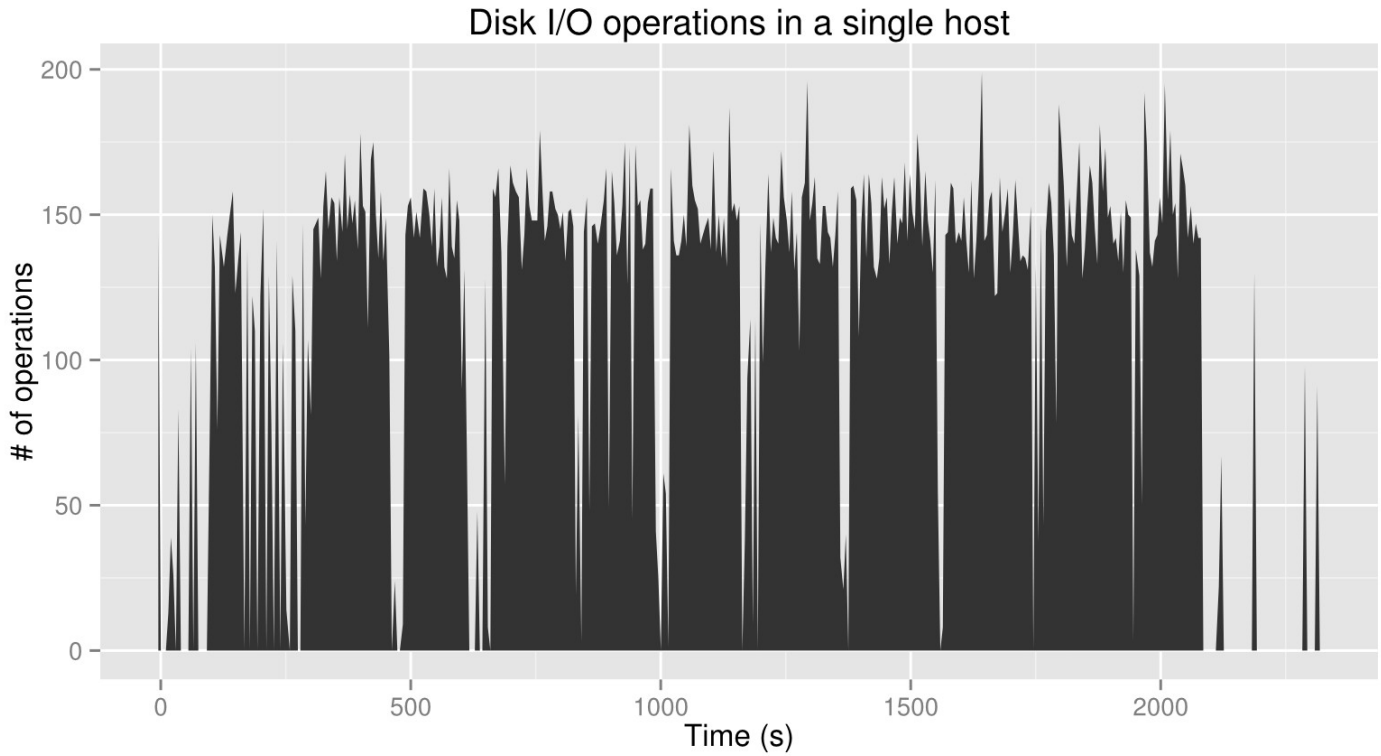
To understand the problem, we tried to reproduce the phenomenon

- At Nancy, with different set of applications
  $\rightsquigarrow$ not a single slowdown "wave" for two months!
- At Nancy with TeraSort
  $\rightsquigarrow$ still so slowdown "wave"!!
- In the meantime Orsay had been retired.
  $\rightsquigarrow$ No way to rerun the experiment in the exact same settings!
- Good thing: Sophia has similar hardware (rather old machines)
  $\rightsquigarrow$ At last, the phenomenon is reproduced!!!

So we know it's hardware-related.

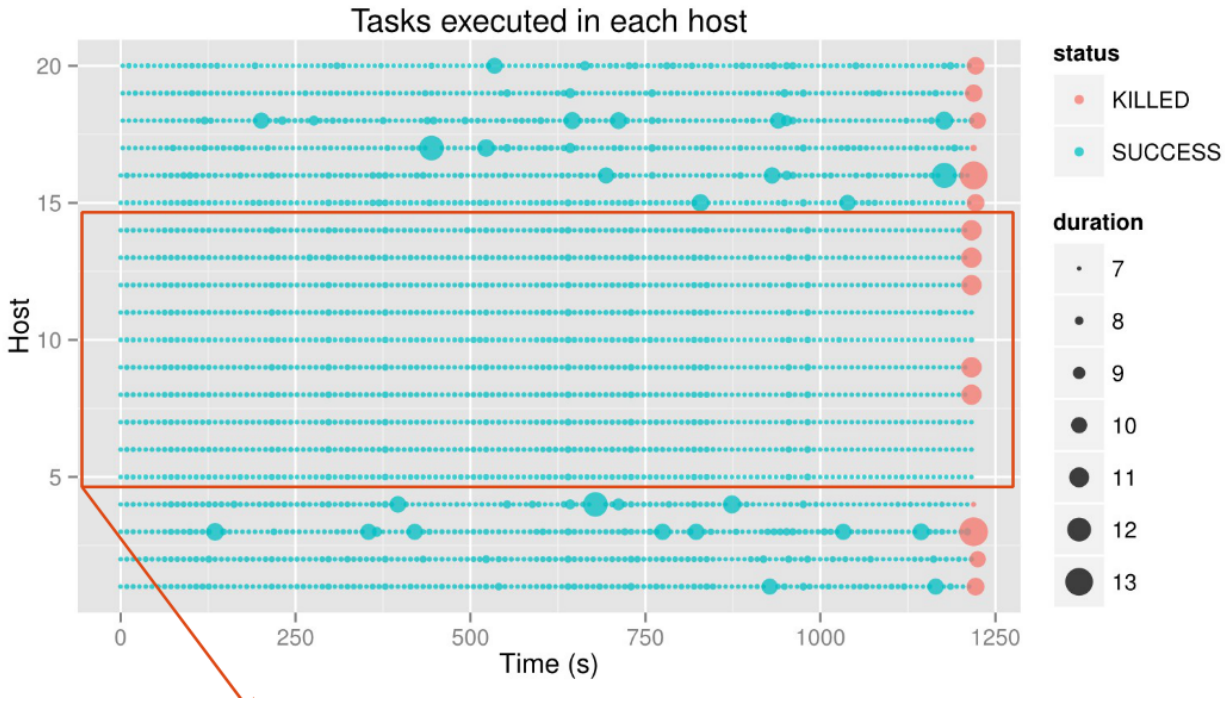
So... What was Causing the Slowdown "Waves"?

- Narrowing the problem, related to the hardware
- Information collected from /proc explained the observed phenomenon



- Further investigation $\rightsquigarrow$ reduce tasks saturate the I/O (prefeching map outputs and spill regularly these inputs on disk)
- That slows down the map tasks, explaining the observed waves!

The Surprising Role of MR Configuration



Hosts without reduce tasks are not affected!

- This phenomenon generally goes unnoticed. Storage difference: SATA (Nice) and SATA II (Nancy)
- How much tasks are slowed down seems difficult to predict
- But we can easily know when the slowdowns will occur and for how long
- Ongoing work to integrate such phenomenon in SimGrid and MRSG

Validation of SimGrid using Grid'5000

Bridging Actual Experiments and Simplistic Simulations

- We push our models to their limits
- Study model validity over a wide range of configurations
- Unveil strange behaviors, hardware or software mis-configurations

$\rightsquigarrow$ Predictive Power: easiest way to run experiments

$\rightsquigarrow$ Better understanding per see: large systems as natural objects

SimGrid as a Scientific Instrument (not only PaperWare)

- Used to **understand** and **optimize** existing production infrastructures
- Application workflows and services deployed on the EGI @ Creatis
- ATLAS Distributed Data Management @ CERN

Share Common Methodologies

- Visualization, Design of XPs, XP Analysis, Methodological Framework
- We scout out these issues in the comfortable settings of the simulator
- Advocate best practices within Hemera (and beyond)

Conclusion

Simulating Large-Scale Applications: Very active, Well funded

- ANR USS-SimGrid: Simulating of P2P scenarios in addition to Grids
- ANR SONGS: Simulating Of Next Generation Systems (Clouds/HPC) 1.8M€, 17 ETP sur 4 ans, 5 laboratoires (all partners are in EPIs)
- Inria: Engineers through ODL & ADT, gforge, pipol & CI
- ERC: application underway (wish me luck)

$\Rightarrow$ Somehow reluctant to use Hemera fundings ;)

SimGrid needs Hemera

- Seeking further collaborations at methodology level
- Joint use with Grid'5000 $\rightsquigarrow$ interesting use cases for us
- Best way to assess your understanding of an object: Simulate It
- Our validation studies depend on the Grid'5000 infrastructure