Expo : Toward a Framework to conduct and control Experiments

The Experiment Execution Engine

Brice Videau and Olivier Richard

Laboratoire d'Informatique de Grenoble

13 février 2008

Outline of the Talk

Context and Objectives

Problematic : Large Scale Experiments

Large scale platforms

- Study the performances of distributed applications
- On a large number of nodes
- Running on several geographic locations

Efficient experiments

- Obtain **accurate** and **reproducible** results
- Obtain a maximum of information
- In a minimum of time

To Conduct Experiments is a Difficult Process

Many parameters

- Nodes are not as homogeneous as they should be
- Nodes status can vary between nodes and through time
- Environment is ever-changing, other experimenters are working
- Managing large experiments is tedious

How to ?

- Account for nodes heterogeneity
- Bring nodes in a known state
- Check the environment
- Plan and run experiments

Objectives : Methodology and Tools to Solve this Problematic

A methodology that guarantees :

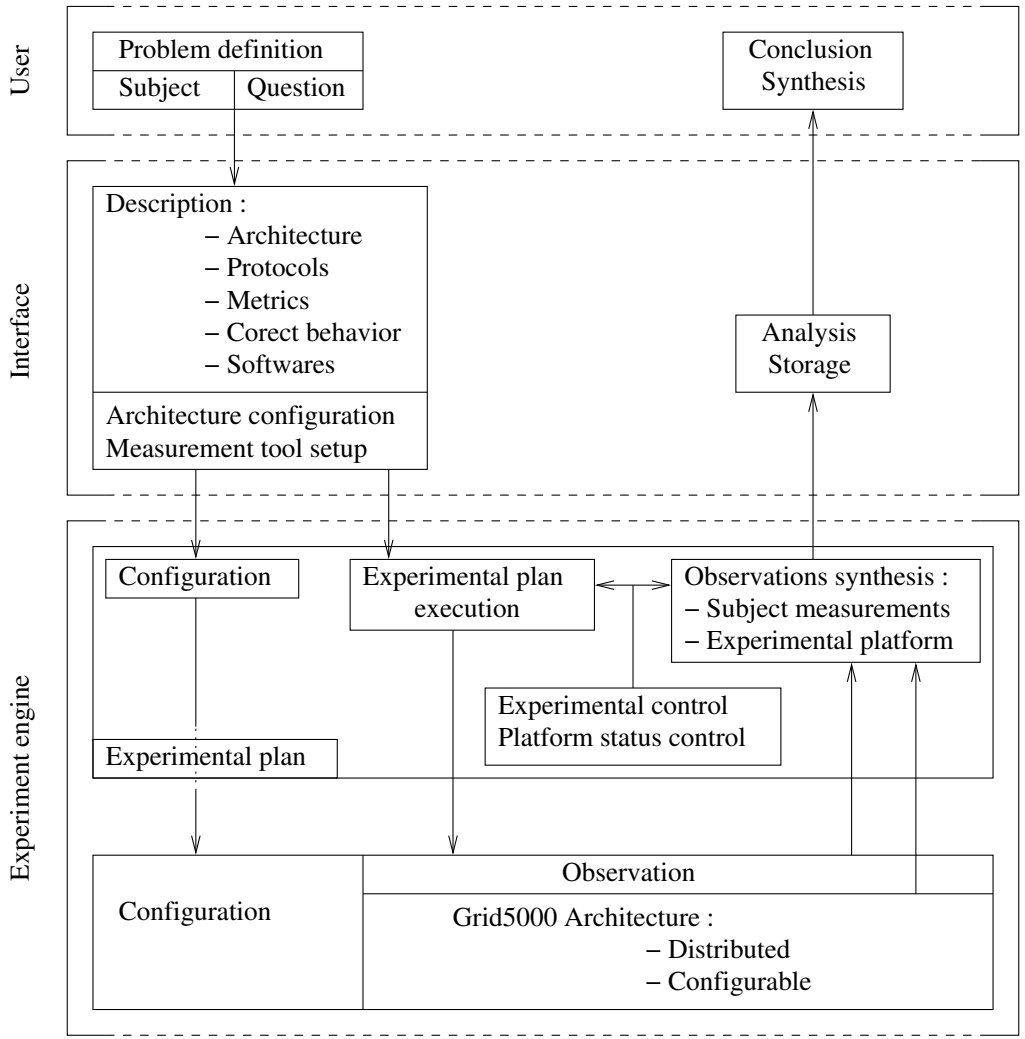
- **Reproducibility** of measurements
- **Reproducibility** of measurements
- **Efficiency** of the process
- **Automation** of the process

Design tools that :

- Helps the experimenter
- Enforces the methodology
- Can adapt to several platform

Expo Experiment Framework

Framework to Conduct Experiments : a Global Picture



Characteristics :

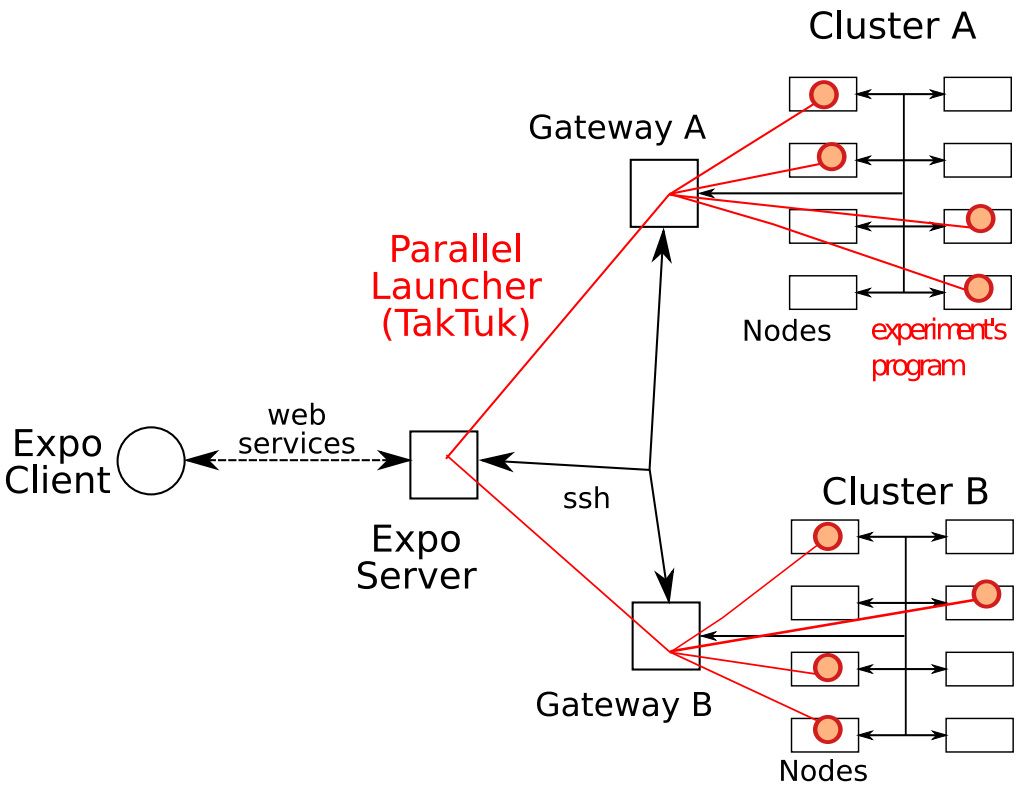
- Experiment description
- Execution engine
- Resources management
- Native archiving
- Fine command control
- Multi-platforms

Related Works :

- Plush + Nebula
- DART
- Zenturio
- Various Scientific Workflow Engines

Expo : Experiment Execution Engine

Expo : Experiment Execution Engine

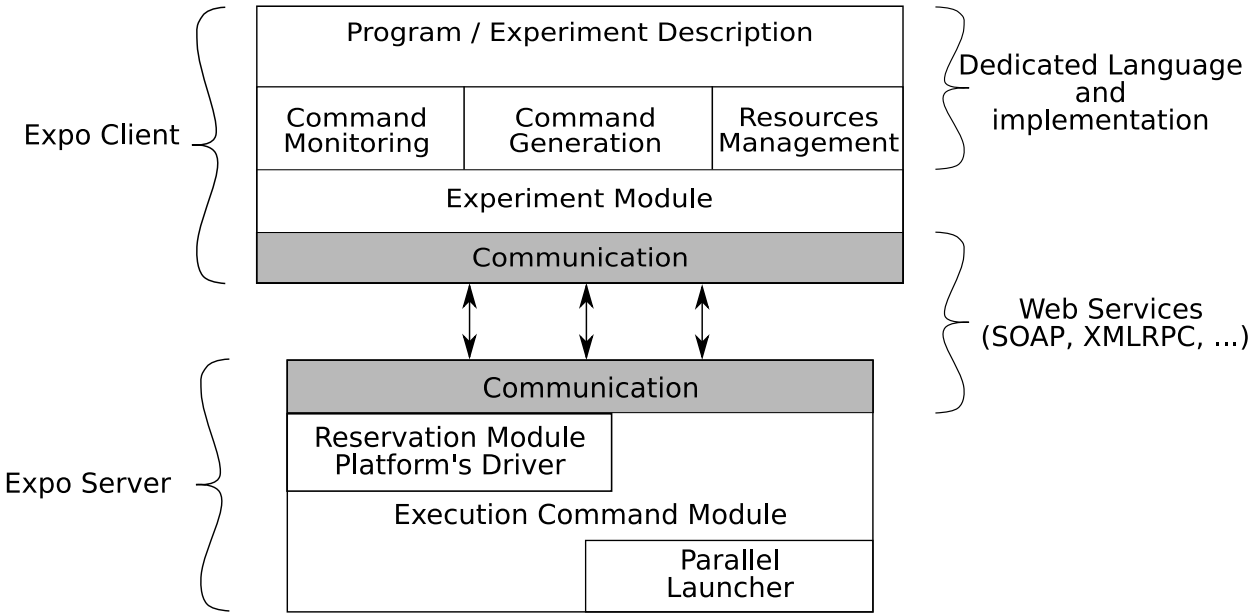


Architecture :

- Client/Server
- Versatile parallel launcher (Taktuk)
-

- Example of experiment on lightweight grid.

Expo : Architecture of Experiment Execution Engine



Characteristics :

- Script Language : Ruby
- Embedded Domain Specific Language to describe experiment
- Operation on Resources Set
- Extensible (driver for new platform, new command...)

Main Expo's Commands

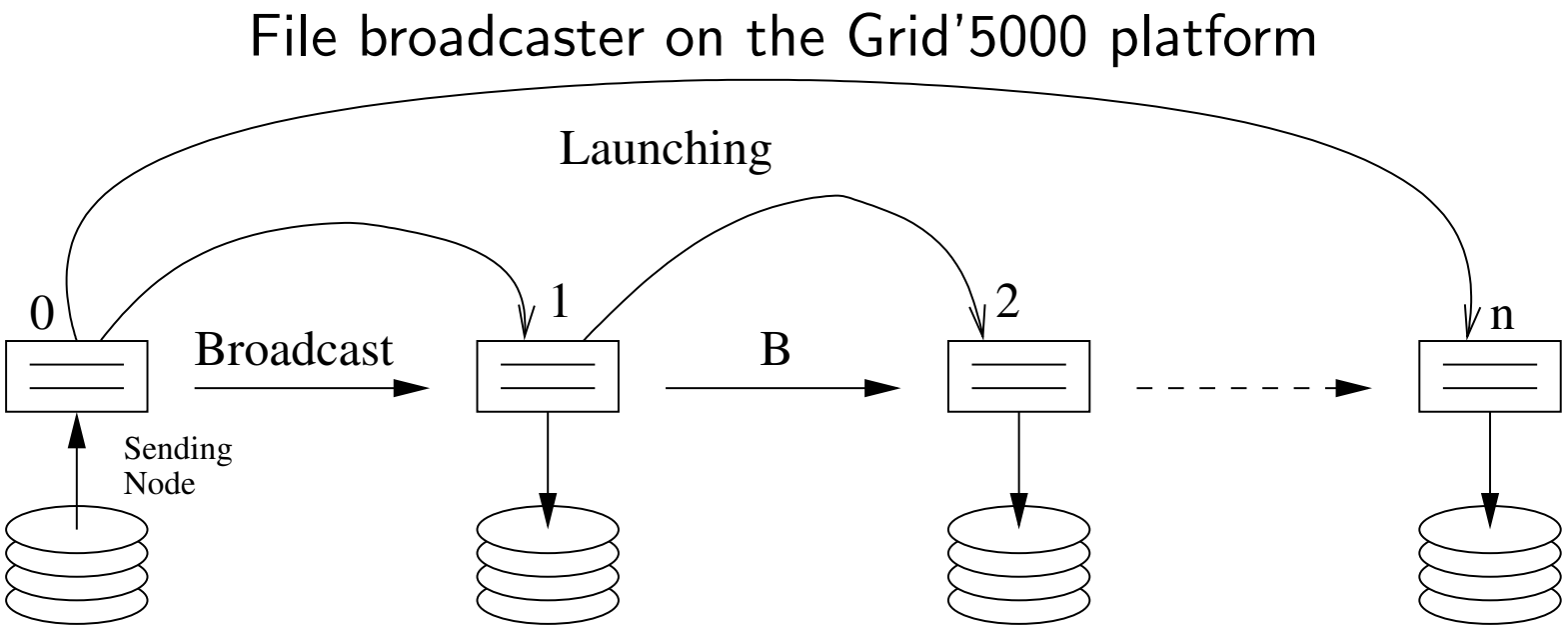
Global Commands	
check	resources (nodes) checking
task	task execution
atask	asychronous task execution
ptask	parallel task execution
patask	asychronous parallel task execution
barrier	waiting of asychronous tasks completion
copy	file copy from one node to another one
Grid'5000's Specific Commands	
oargridsub	resources reservation w/ oargrid
oargridconnect	connection to previous reserved resources
kadeploy	low-level environment deployment thanks to kadeploy
akadeploy	asychronous low-level environment deployment
kadeploy_progress	deployment progression monitoring

Experiment Description : Simple Example

```
#load specific functions for Grid'5000 platform
require 'expo_g5k'
#reservation on gdx en helios clusters
oargridsub :res => "gdx:nodes=10,helios:nodes=10"
#nodes' checking
check $all
#execution on each nodes
ptask $all.gateway, $all, "date"
#execution on gdx cluster
id, res = ptask $all["gdx"].gateway, $all["gdx"], "sleep 1"
puts "average : " + res.mean_duration
```

Use Case : File Broadcast

Case study : File Broadcast



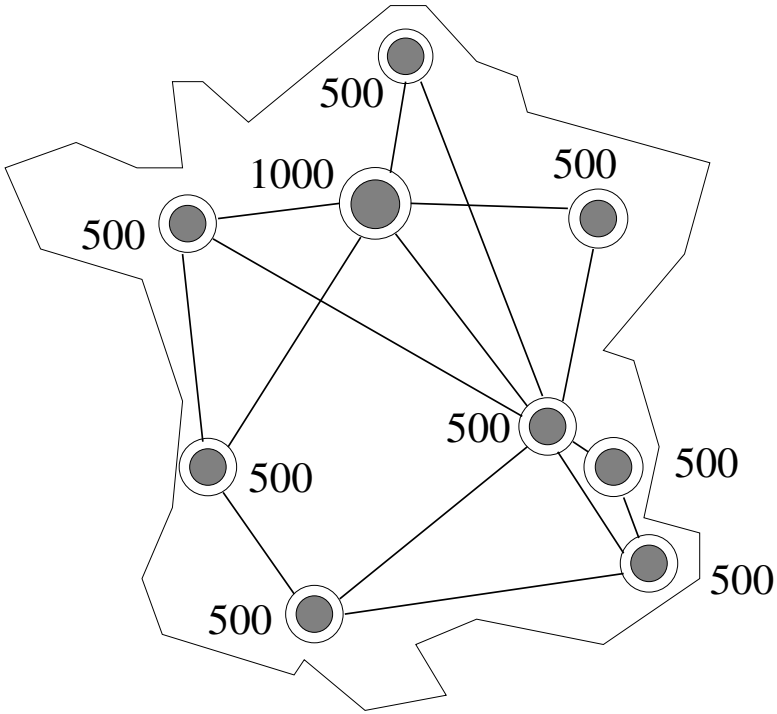
Simple experiment :

- Parallel Launching + File Pipelining

Experiment Description :

```
...
#File to broadcast creation
task $all.first, "./create_files"
#Iteration on node set (set size double at each iteration)
$all.each_slice_power2 do |n|
 #skip first iteration
 next if n.resources.length == 1
 #copy file on sender
 copy n.nodefile, $all.first
 #iterate on file size
 [1024,2048,4096,8192].each do |size|
 10.times do |i| #for statistic
  id,r=task $all.first,"kastafior -#{n.nodefile} -file#{size}
  puts "#{n.resources.length} #{size} #{r.duration}"
 end
end
```

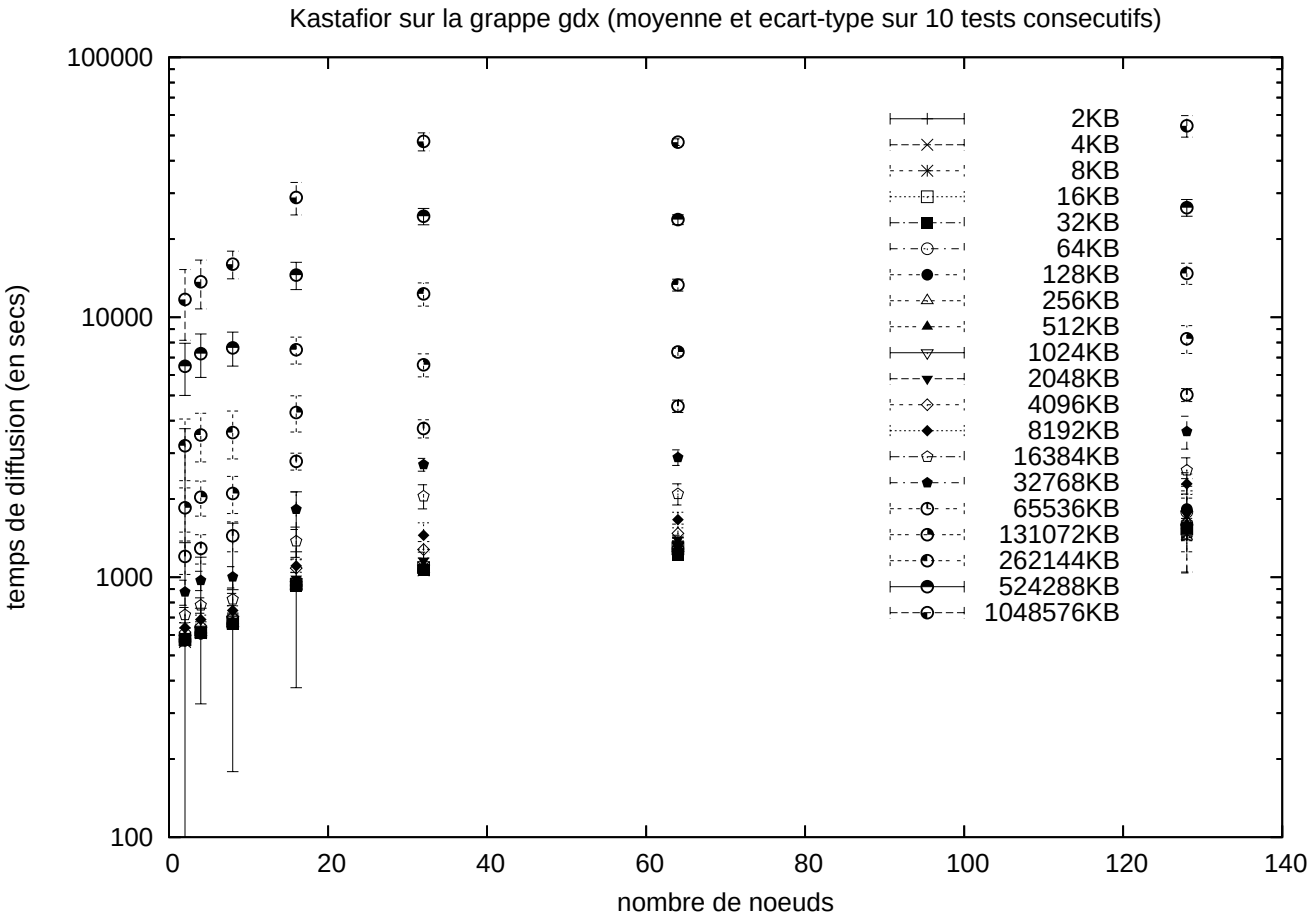
Grid'5000



Some facts :

- 5000 cores
- Nodes : Bi dualcore AMD Opteron @ 2GHz
- 2 GiB of memory
- 1Gib/s Ethernet adapter
- sites interconnected at 10Gib/s

Results



2 parameters (node, file size), 10 iteration by point, 1400 measures, 25 LOC for experiment description

Conclusion

- Large experiments are difficult to conduct
- New tools required
- Expo : toward a Framework to conduct and control experiments
- eDSL approach + versatile parallel launcher

Futur Works

- Set of Resources Enhancements
- Measures and experiments stokage and manipulation
- Beta Release (HTTP ://EXPO.IMAG.FR)
- Test on more platform (PlanetLab, DSLLAB, Emulab, EGEE ?)
- GUI (to launch and monitor experiment)
- More experiment examples

EGEE Perspectives

Expo can be use to

- Study complexe application
- Evalution, Profiling, Debugging...
- Evaluate EGEE middleware (in Grid'5000)

Question

Can we have a Grid'5000 VO ?