Simbatch: an API for simulating and predicting the performance of parallel resources managed by batch systems

Yves Caniou and Jean-Sébastien Gay

Université de Lyon, LIP / ÉNS Lyon

This work is supported by
- the LEGO project ANR-05-CIGC-11
- the REDIMPS project JST-CNRS
- the cluster Rhône Alpes

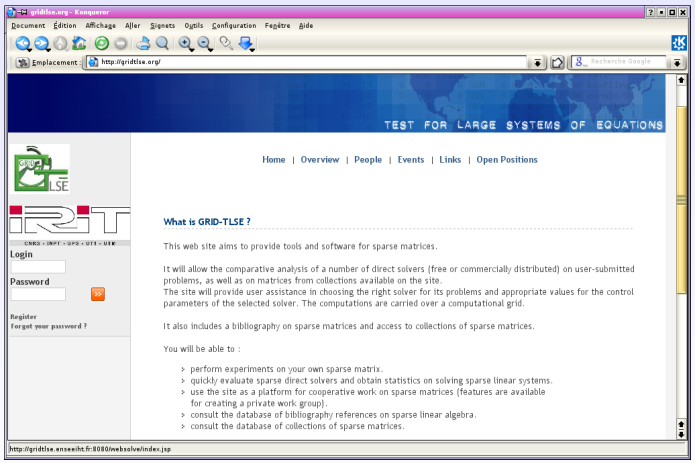
SGS'08, August 25, 2008

Goals of TLSE

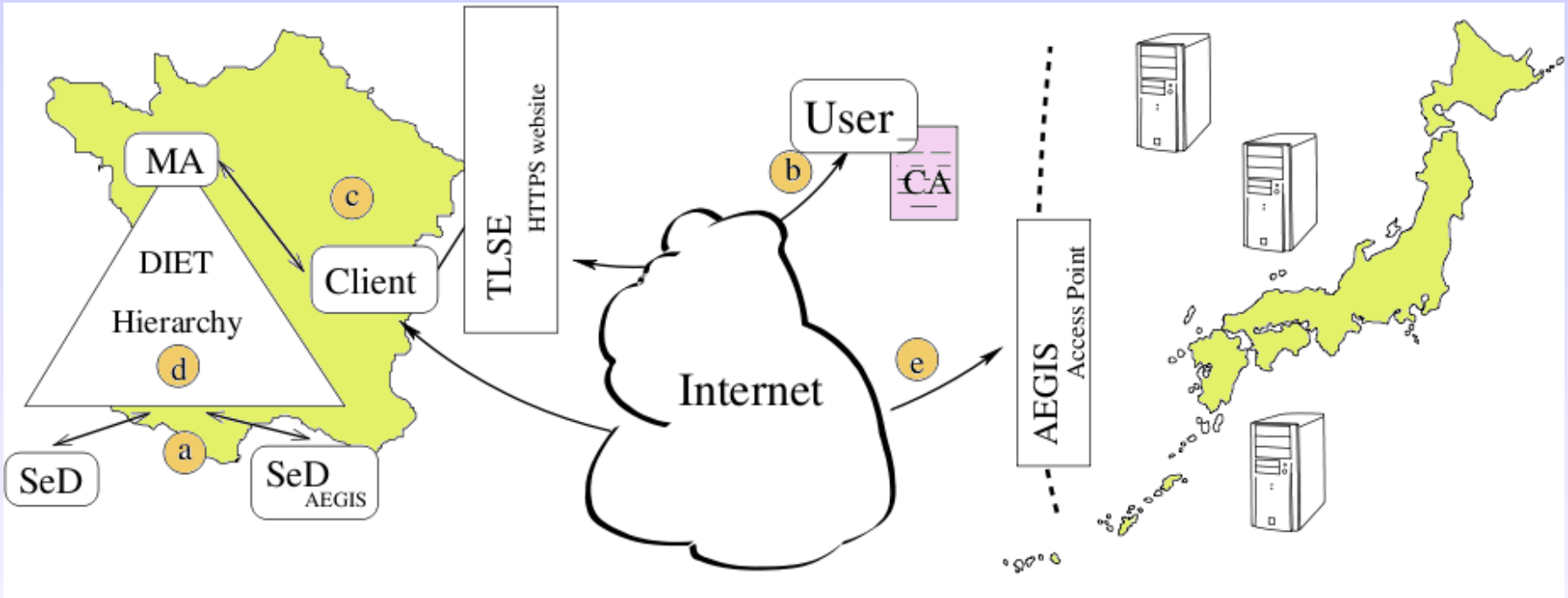
Test for **L**arge **S**ystems of **E**quations

- Provide access to a range of sparse linear algebra tools
- Assist users in selecting
  - the most appropriate solver and
  - appropriate values of control parameters for their problem
- Matrix collections available
- Scenario of experiments available

http://gridtlse.org/



Grid architecture and involved mechanisms



AEGIS-DIET Grid protocol interoperability!

In the point of view of performance...

TLSE

- helps in predicting the best choice
- enables faster and more accurate execution of applications
- increases efficiency of problem solving in physical and engineering research

But...

- Better performance if Grid scheduling
- Scheduling if performance estimations
- And architecture composed of parallel resources which rely on different batch reservation systems
  - “Simple” scheduling similar to Minimum Completion Time
  - Rescheduling, task migration
  - ... and maximize overlap of waiting queue with data transfer!

→ Need for a tool to simulate such Grid elements

Outline

Defining the needs

2 main requirements:

- Grid point of view
  - Realistic predictions
  - Fast, easy to deploy (with the Grid middleware)

  → Use scripts, use DRMAA?
- Reusable code
  - Scalable experiments are hardly reproductible

  → Have a tool that helps to simulate a real Grid!

Goals:

- Accurately simulate clusters and parallel jobs execution
- Accurately simulate batch reservation systems

State of the Art

Data replication

- ChicSim (I. Foster)
- OptorSim (W.H. Bell and R. Cameron): Java

Grid Economy

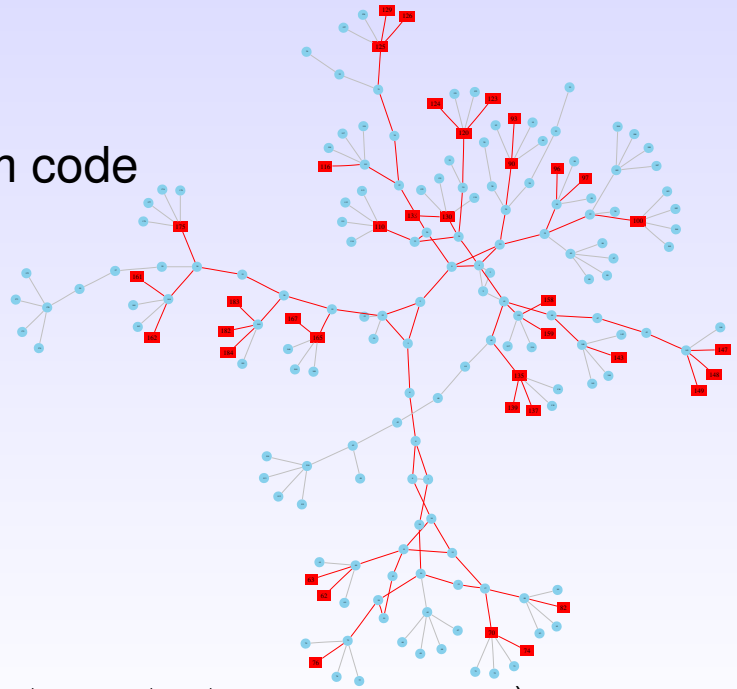
- GridSim (R. Buyya): Java

Distributed scheduling algorithms study

- Simgrid (H. Casanova, A. Legrand, M. Quinson)

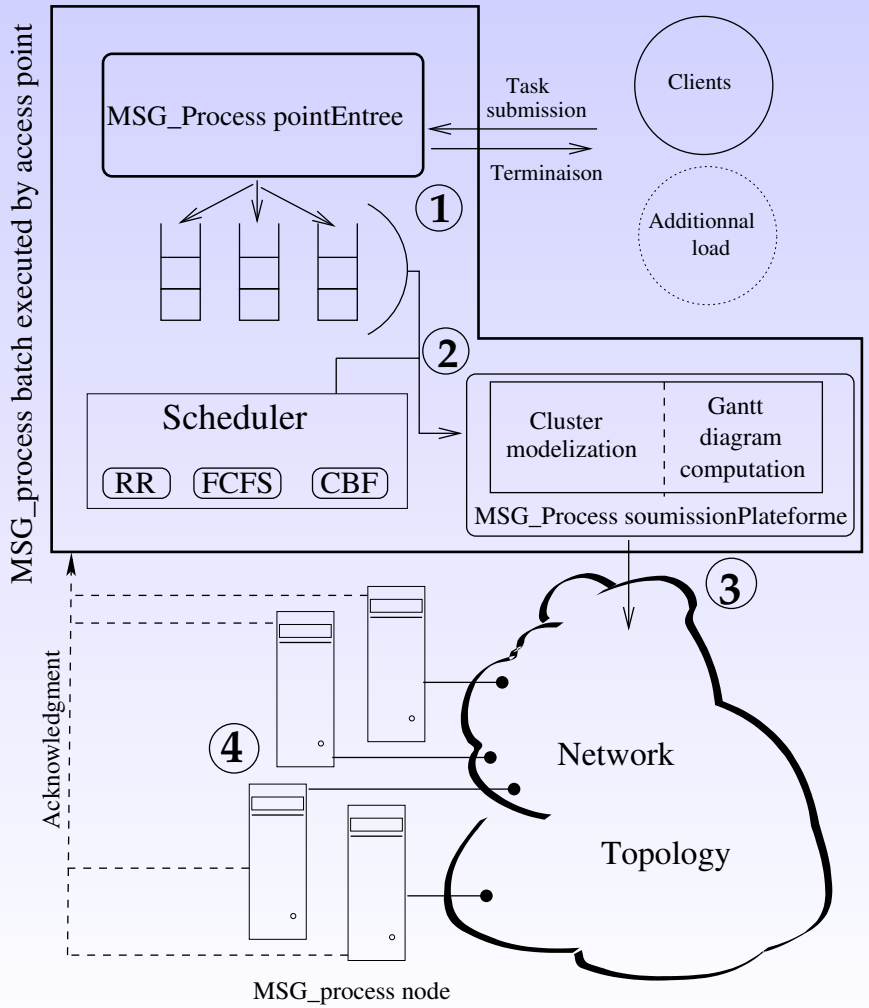
Simgrid

- “SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments”
- 4 layers
  - SURF: Simgrid 3.0 kernel
  - XBT: similar to glib
  - MSG: meta SimGrid
  - Gras: distributed and reusable program code
- 3 main and mandatory structures
  - MSG_host
  - MSG_process
  - MSG_task
- Some functionalities
  - Simulation of Lan, Wan ...
  - Grid description with XML files
  - Compatible with Tiers (`http://www.isi.edu/nsnam/ns/ns-topogen.html`)



Models and involved mechanisms

- Parallel resources
  - Access point
  - Computing resources
  - Topology
- Parallel task: Simbatch metatask
  - Input and output size
  - Submission date
  - Time of execution and walltime
  - Number of computing resources



Built-in

- Parallel resource models
- Batch reservation system scheduling models
  - RR
  - FCFS (PBS)
  - CBF (OAR, MAUI, openPBS)
- Use adapted Simgrid XML files
- 2 APIs

Programmer API

Example of functionalities

- Get holes, characteristics provided or not

```
hole_t *
find_first_hole(cluster_t c, int nb_nodes,
                double start_time, double duration);
```

- Empty Gantt and reschedule

```
void
generic_reschedule(cluster_t c,
                   void (*schedule)(cluster_t c,
                                    m_tast_t task));
```

User API

Use 4 XML files

- Platform description with Simgrid XML files
- Use of Simbatch API in deploiement.xml
  - Creation of a batch process on the Access Point

```
<process host="Frontale" function="batch" />
```

- Simulated batch system configuration file
- Additional intern load for a batch system

Differents kind of experiments – 1/2

2 goals

- Validation of the correctness of simulations
  - Different topologies, scheduling algorithms, etc.
- Validation of the accuracy of simulation

  → Comparison with identical real life experiments
  - Computing tasks
  - Communicating taks

Differents kind of experiments – 2/2

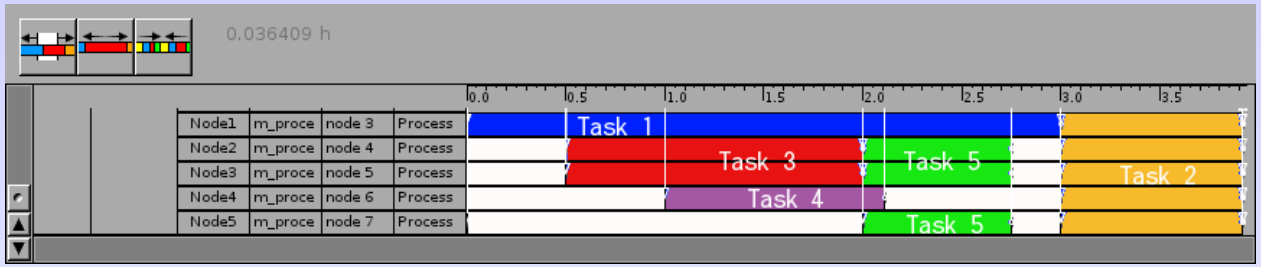
Experimentation protocol

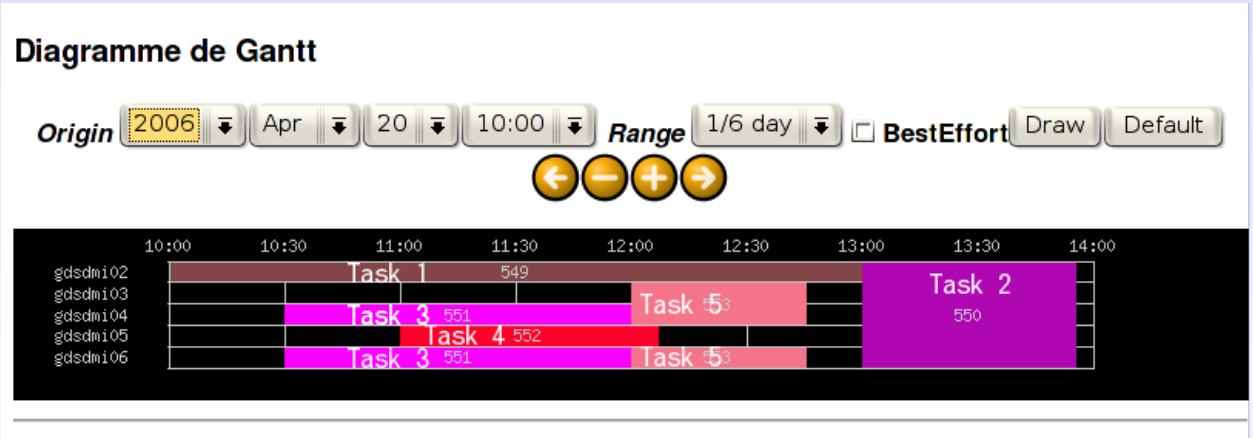
- Task generation
  - Task arrival: Poisson law
  - Number of proc: U(1 ; 7)
  - Execution duration: U(600 ; 1800)
  - Walltime: duration*U(1.1 ; 3)
- Communication tasks
  - U(1 ; 6) for: 1, 2, 5, 10, 15, 20 MB file

Real life platform

- Cluster: Access Point, 7 computing nodes
- Star topology, 100 Mbits/sec switch
- OAR v1.6

About scheduling

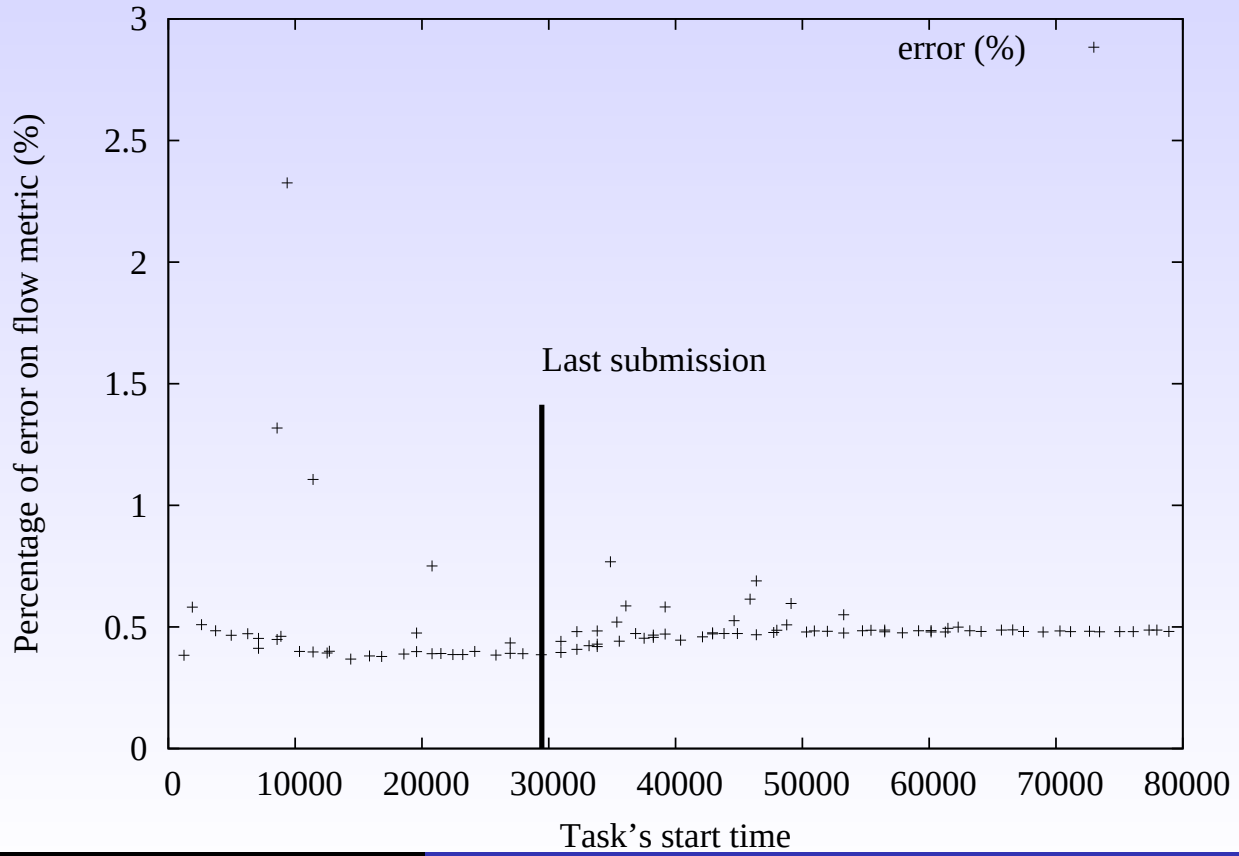




Schedules are similar: same order, different resources

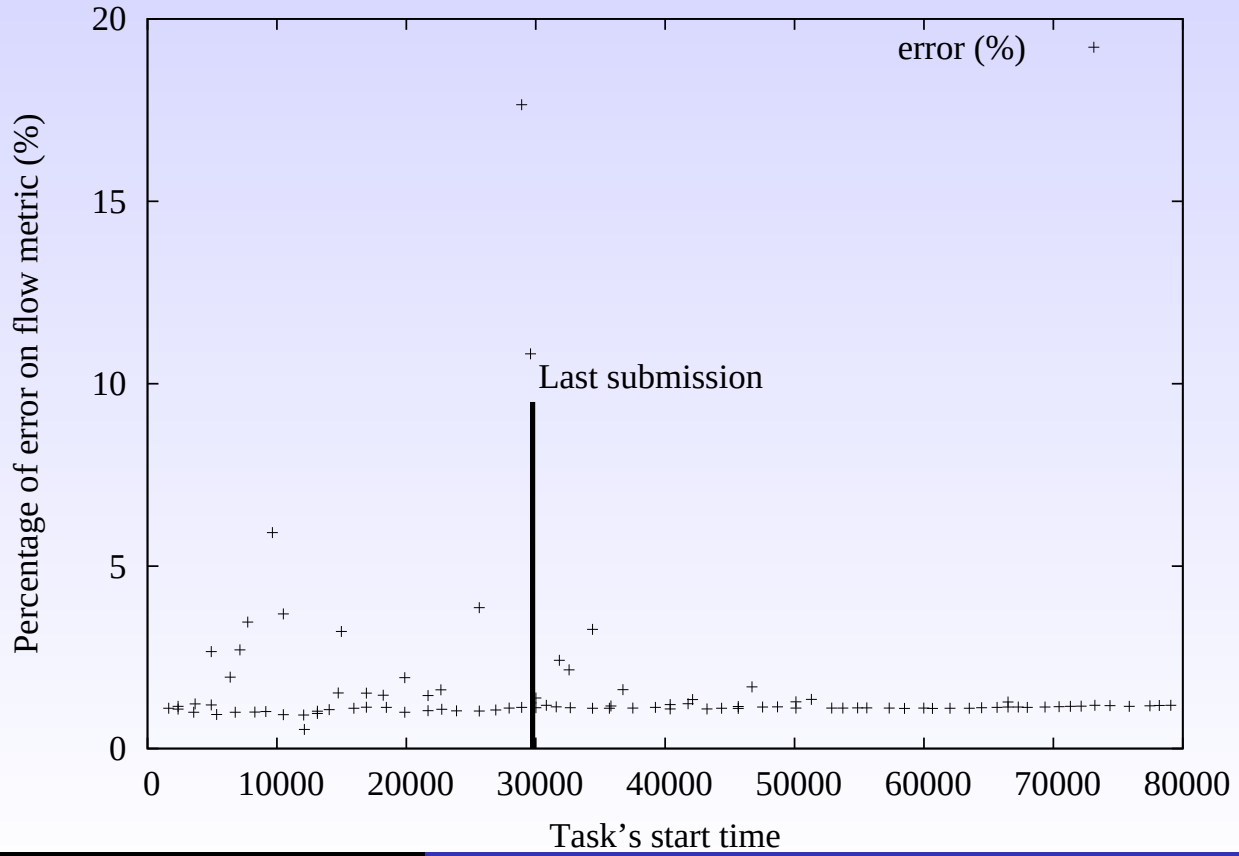
Precision of simulations.. – 1/2

.. with only computing tasks



Precision of simulation.. – 2/2

.. with additional communicating tasks



Future works

- Simbatch
  - Modelization of Grid'5000
  - Size of the files! ... but now corrected
  - Comparison with other batch systems
- Simbatch - DIET
  - Integration of Simbatch calls into DIET
- DIET
  - New scheduling heuristics
  - ... and Grid middleware tunable parallel tasks

  → more generic than for GridTLSE...

Conclusion

- GridTLSE, a secure international expert system for Sparse Linear Algebra
- Simbatch
  - Integrated into Simbatch
  - Provides new models
  - Provides 2 different APIs
  - Fast and Accurate

Still a lot of future work...

deploiement.xml example

```
<?xml version='1.0'?>
<!DOCTYPE platform_description SYSTEM "surfxml.dtd">
<platform_description>
  <process host="Client" function"client">
    <argument value="0" />
    <argument value="0" />
    <argument value="0" />
    <argument value="Frontale" /> <!-- Connection -->
  </process>
  <!-- The Scheduler process (with some arguments) -->
  <process host="Frontale" function="batch">
    <argument value="0" />      <!-- Number of tasks -->
    <argument value="0" />      <!-- Size of tasks -->
    <argument value="0" />      <!-- Size of I/O-->
    <argument value="Node1" />  <!-- Connections -->
    <argument value="Node2" />
    <argument value="Node3" />
    <argument value="Node4" />
    <argument value="Node5" />
  </process>
  <process host="Node1" function="node"/>
  <process host="Node2" function="node"/>
  <process host="Node3" function="node"/>
  <process host="Node4" function="node"/>
  <process host="Node5" function="node"/>
</platform_description>
```

batchrc example

```
# Configuration File
# Comment or uncomment

################### Batch Behaviour ########################
# Plugin used to schedule
# PLUGIN_SCHEDULER = "../lib/plugins/algo/rrobin.so"
# PLUGIN_SCHEDULER = "../lib/plugins/algo/fcfs.so"
PLUGIN_SCHEDULER = "../lib/plugins/algo/cbf.so"

# Define how many queue you want (5 is higher priority than 1)
NB_QUEUE = "3"

################### Internal Workload ######################
# Comment this if you don't want to use any additionnal
# workload

INPUT_PARSER = "../lib/plugins/input/wld.so"
INTERNAL_WORKLOAD = "workload/wld/ex5.wld"
```