SimBatch : une API pour la simulation et la prédiction de performances de systèmes batch

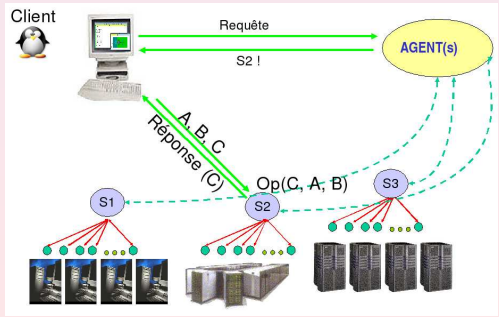
Jean-Sébastien Gay

LIP, ENS Lyon

18 mai 2006

Introduction

- “Interconnection hétérogènes de ressources hétérogènes”
- Métacomputing à l’aide d’intergiciel
- Modèle GridRPC
- Plusieurs utilisateurs
- Ordonnancement à 2 niveaux
  - Intergiciel de grille
  - Système de réservation batch

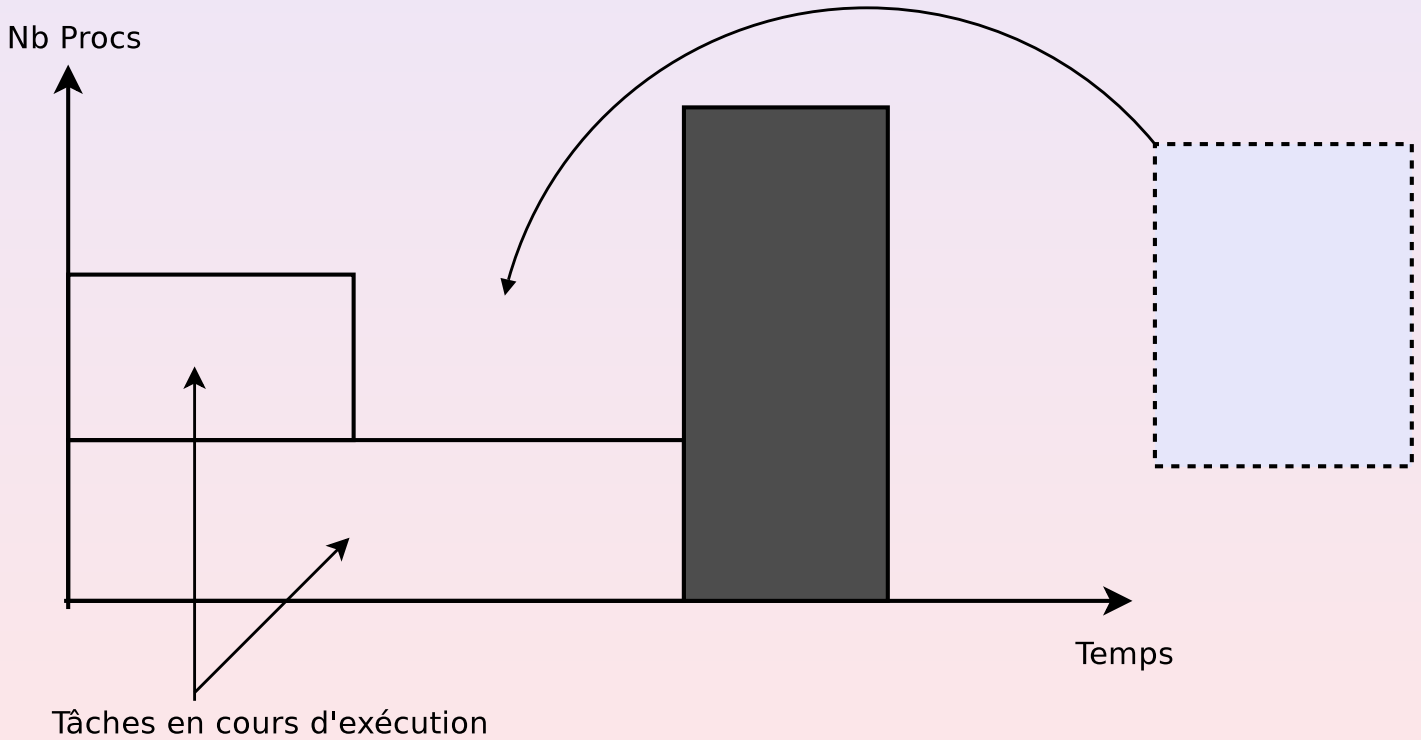


Algorithmes

- Round Robin
- First Come First Served (FCFS)
- Conservative BackFilling (CBF)
- Easy
- Agressive BackFilling

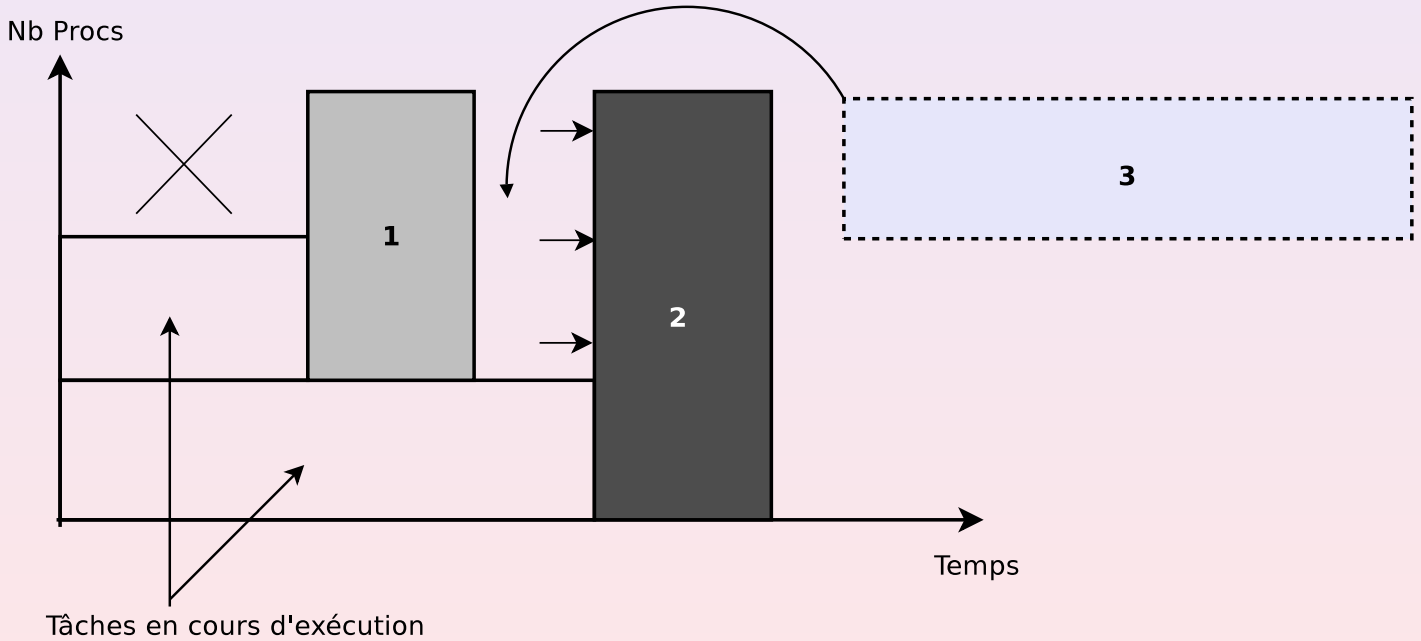
Conservative BackFilling

- Principe : aucune Tâche ne peut être retardée



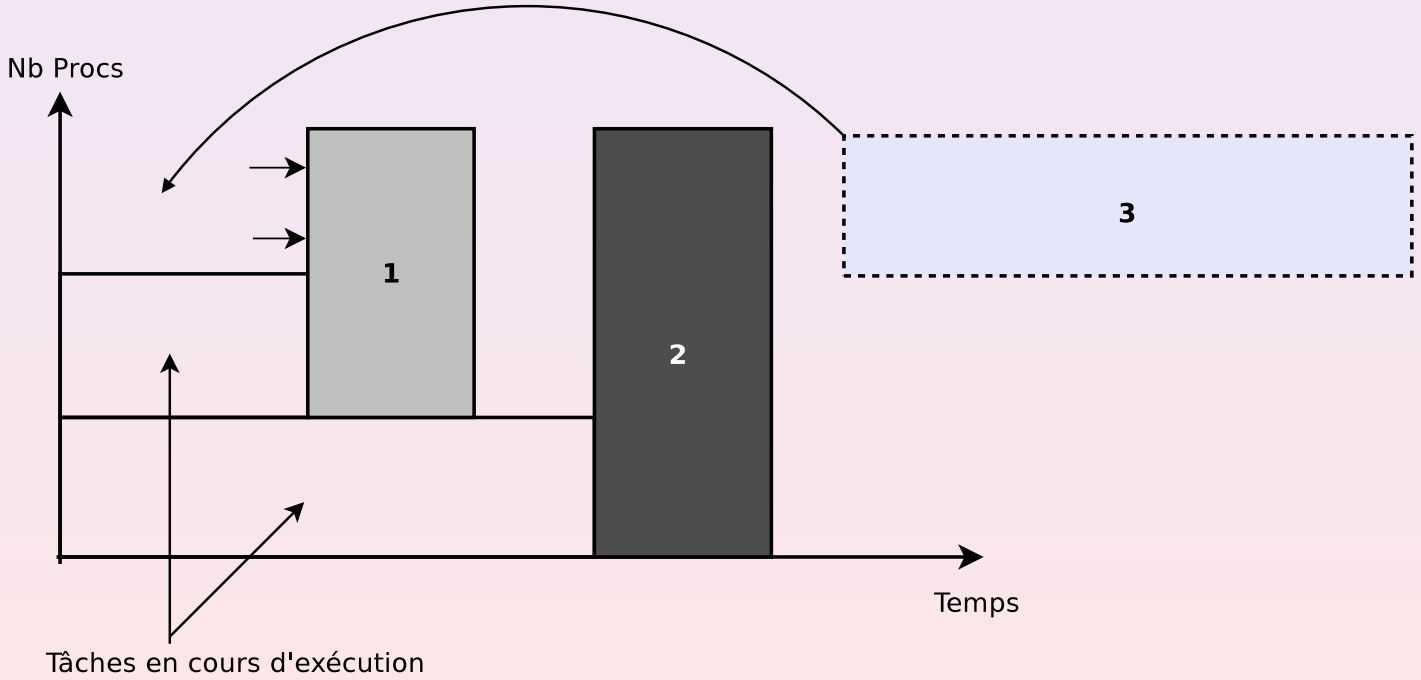
Easy

- Principe : seule la plus ancienne tâche dans la file ne peut être retardée



Agressive BackFilling

- Principe : toutes les tâches en attente peuvent être retardées



OAR

- Développé à l'IMAG (Grenoble)
- Utilisé principalement sur la plate-forme Grid'5000
- Algorithme utilisé : CBF
- Programmé en perl et MySQL
- `http://oar.imag.fr/`

PBS

- Portable Batch System
- Développé par la NASA au milieu des années 90
- Implante FCFS
- 2 versions :
  - OpenPBS (cf Torque)
  - PBS pro (version payante)
- `http://www.openpbs.org/`

Autres

- Torque + Maui
  - Torque = fondé sur *PBS (FCFS)
  - Maui = ordonnanceur amélioré
  - Gratuit
- MOAB cluster suite (Maui amélioré et payant)
- Sun Grid Engine (SGE) développé par Sun Microsystem
- LoadLeveler développé par IBM

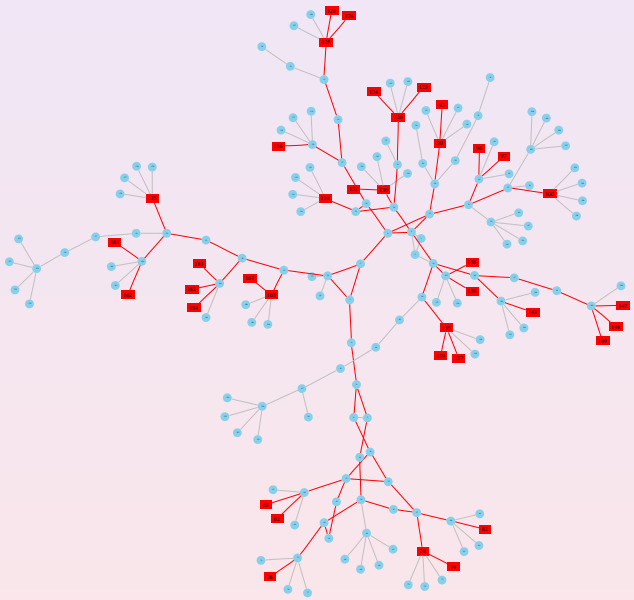
Simulateurs

- Réplication de données
  - ChicSim
    - Développé par I. Foster
    - PARallel Simulation Environment for Complex Systems
  - OptorSim
    - Développé par W. H. Bell, D. G. Cameron, R. Carvajal-Schiaffino en Java
- Modéliser les algorithmes de GRID-ECONOMY
  - GridSim
    - Développé par R. Buyya en Java
    - Même modèles que Simgrid
- Évaluation d'algorithmes d'ordonnancement distribué
  - Simgrid

Présentation

- "SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments"
- 4 couches
  - SURF : noyau de Simgrid 3.0
  - XBT : équivalent de la glib ou d'APR
  - MSG : Méta SimGrid
  - Gras : écriture de programme distribué
- 3 structures de données essentielles
  - MSG_host
  - MSG_process
  - MSG_task

Topologie



- Prise en compte des réseaux Lan, Wan ...
- Description à l'aide de fichiers XML
- Compatibilité avec Tiers [a]

[a] http://www.isi.edu/nsnam/ns/ns-topogen.html

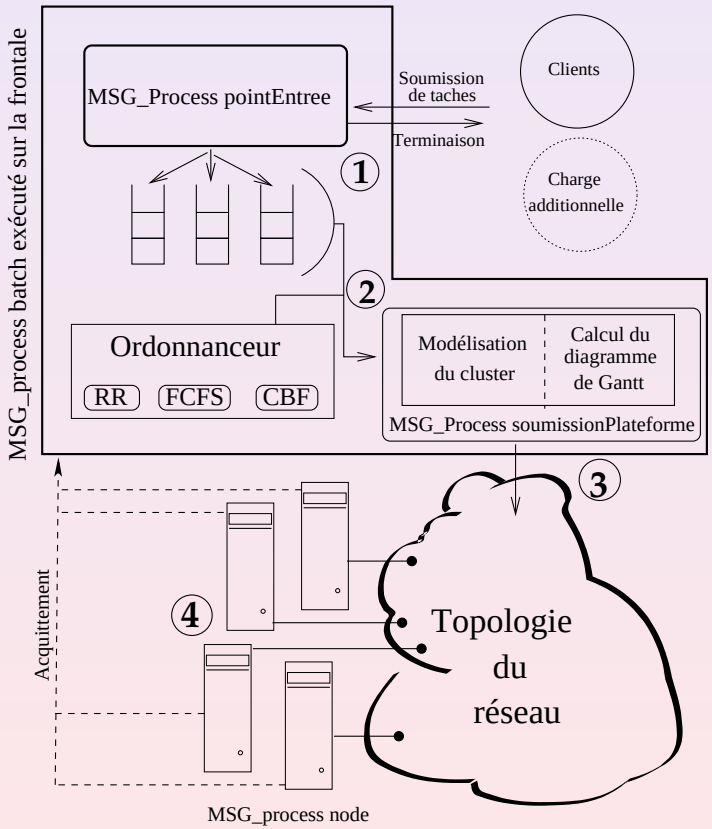
Buts

- Réalisme de la modélisation des grilles
- Via une API intégrée à SimGrid
  - Simulation de grappes de PCs
  - Simulation de systèmes de réservation Batch
- Conception d'algorithmes d'ordonnancement
- Évaluation d'algorithmes d'ordonnancement
- Prédiction de performances au sein de DIET

Modèles utilisés

- Grappe de PCs
  - Une frontale
  - Des ressources de calcul
  - Une topologie
- Méta-tâche Simbatch
  - Taille des entrées
  - Date de soumission
  - Durée d'exécution
  - Durée de réservation
  - Nombre de ressources de calcul
  - Taille des sorties

Modélisation d'une grappe et fonctionnement



1. Réception et placement des tâches parallèles dans la file d'attente appropriée
2. Attribution d'une date de passage et réservation des ressources nécessaires
3. Envoi des tâches à la date déterminée sur les ressources attribuées
4. Exécution des tâches puis envoi d'un acquittement

Présentation de l'API

- Modélisation de grappes de PCs
- Modélisation de ressources de calcul
- Fournis des algorithmes d'ordonnancement
  - Round Robin
  - FCFS
  - CBF
- Modélisation de système de réservation batch
  - OAR
  - PBS

API coté développeurs

- Utilisation de la libschedule.so
- Exemple de fonctions
  - Trouver le premier espace adatpé dans un digramme de Gantt

    ```
    hole_t * find_first_hole(cluster_t c, int
    nb_nodes, double start_time, double
    duration)
    ```
  - Vider les files d'attente et réordonnancer

    ```
    void generic_reschedule (cluster_t cluster,
    void (*schedule) (cluster_t cluster,
    m_task_t task));
    ```

API coté utilisateurs

- Description de la plate-forme à l'aide des fichiers xml Simgrid
- Utilisation de l'API dans le fichier deploiement.xml
  - Création d'un *processus batch* sur l'hôte Frontale

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

- Fichier de configuration propre à un batch
- Possibilité de simuler une charge propre pour chaque système

```
<?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>
```

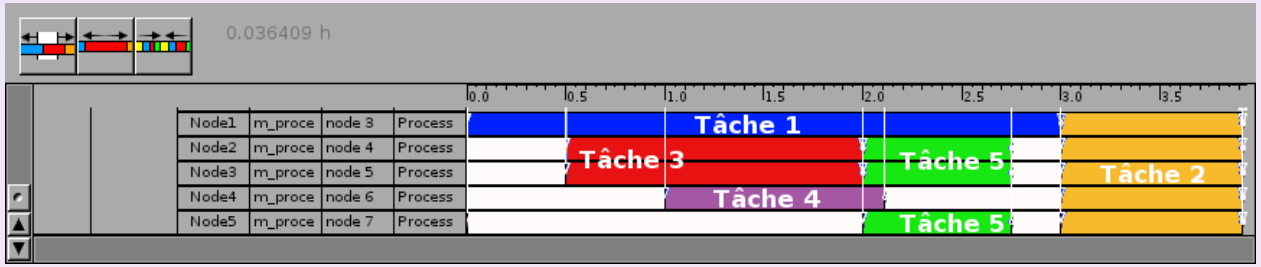
Batchrc

```
# 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"
```

Protocole d'expérimentation

- 2 types d'expériences
  - Validation par simulation : variation des paramètres $\rightarrow$ topologie, algo d'ordonnancement...
  - Comparaison de la modélisation avec une plate-forme d'expérimentation réelle
- Génération de tâches
  - Inter-arrivées : loi de poisson
  - Nombre de ressources : $U(1, 5)$
  - Durée du calcul : $U(600, 1800)$
  - Durée de réservation : durée du calcul x $U(1.1; 1.3)$
- Plate-forme d'expérimentation réelle
  - Cluster de 5 nœuds
  - Topologie en étoile
  - OAR v 1.6

Validation


0.036409 h
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Node1 m_proce node 3 Process
Node2 m_proce node 4 Process
Node3 m_proce node 5 Process
Node4 m_proce node 6 Process
Node5 m_proce node 7 Process
Tâche 1
Tâche 3
Tâche 5
Tâche 2
Tâche 4
Tâche 5

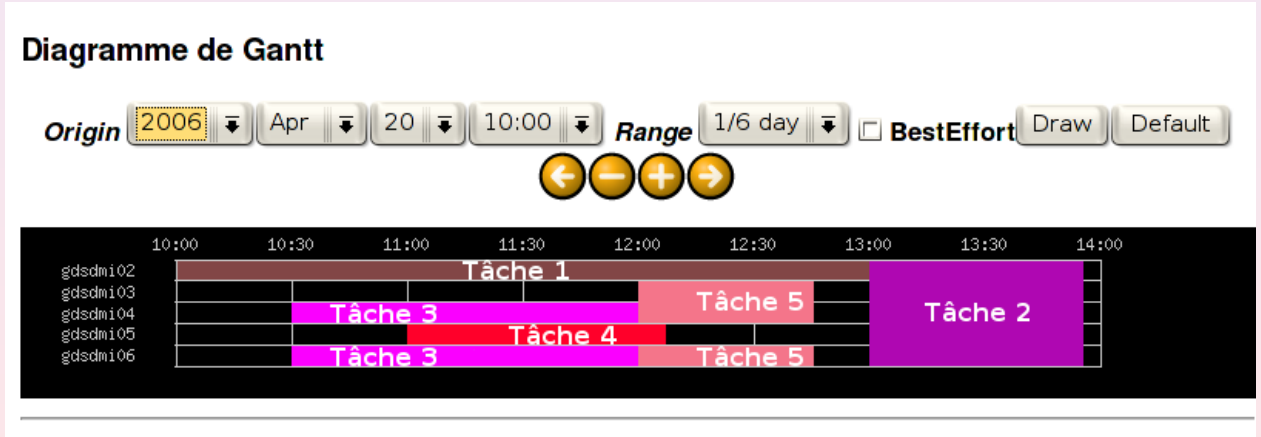
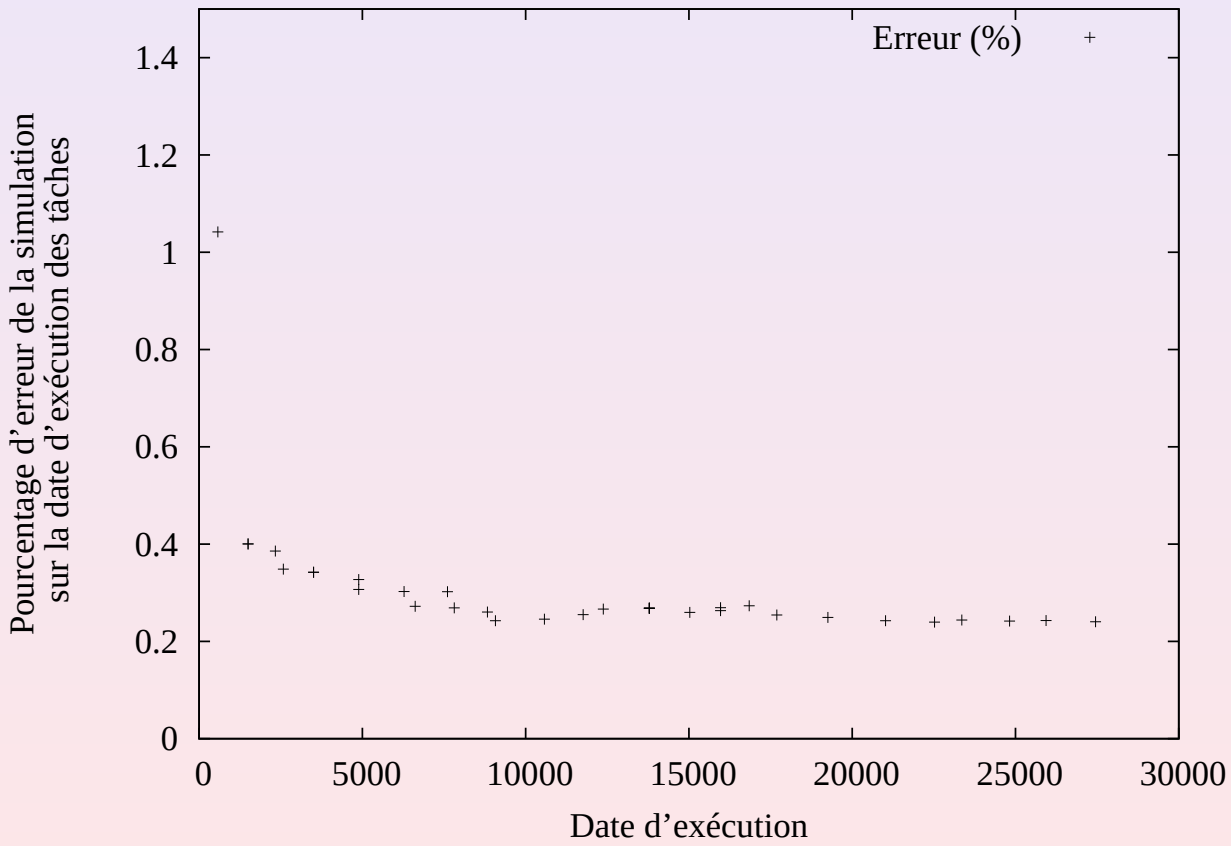
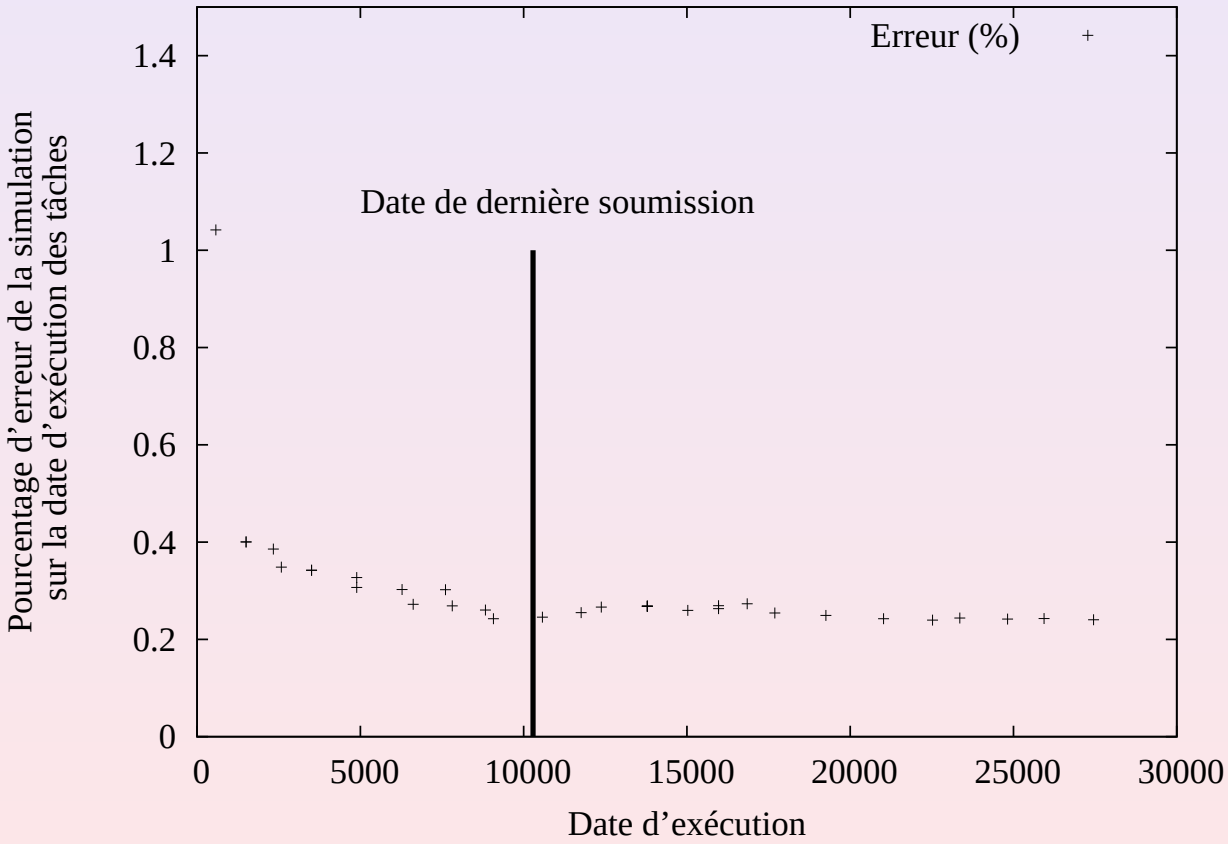


Diagramme de Gantt
Origin
2006
Apr
20
10:00
Range
1/6 day
BestEffort
Draw
Default
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
gdsdmi02
gdsdmi03
gdsdmi04
gdsdmi05
gdsdmi06
Tâche 1
Tâche 3
Tâche 5
Tâche 2
Tâche 4
Tâche 3
Tâche 5

Résultats


Erreur (%)
Pourcentage d'erreur de la simulation
sur la date d'exécution des tâches
Date d'exécution
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
5000
10000
15000
20000
25000
30000

Résultats


Erreur (%)
Date de dernière soumission
Pourcentage d'erreur de la simulation
sur la date d'exécution des tâches
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
5000
10000
15000
20000
25000
30000
Date d'exécution

Conclusion et perspectives

- Présentation des principaux algorithmes d'ordonnacement pour les systèmes de réservation batch
- Présentation de quelques systèmes batch
- Petit tour d'horizon des simulateurs de grille
- Présentation de Simbatch
  - Modélisation facile de grappes de PCs
  - Conception et évaluation d'algorithmes
  - Bonnes performances
- Modélisation de la plate-forme Grid'5000
- Intégration à DIET