Meta-Scheduling in Advance using Red-Black Trees in Heterogeneous Grids

Luis Tomás, Agustín Caminero, Carmen Carrión, Blanca Caminero

Dept. of Computing Systems
The University of Castilla La Mancha
Albacete, Spain
OUTLINE

- Introduction
- Meta-scheduling In Advance
- Implementation
- Experiments and Results
- Conclusions
OUTLINE

• Introduction
• Meta-scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions
INTRODUCTION

- The Grid infrastructure must provide the needed services for automatic resource brokerage.
- This infrastructure is named “meta-scheduler”.
- Brokering problem:
  - Heterogeneous and distributed nature of the Grid.
  - Differing characteristics of different applications.
- How to solve this problem:
  - To ensure that a specific resource is available when the job requires it.
  - To reserve or schedule the use of resources in-advance.
INTRODUCTION

• Introduction
• Meta-Scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions

• Advanced reservation:
  – Restrictive or limited delegation of particular resource capacity.
  – Provide some QoS by ensuring that a certain job ends on time.
  – Increase the predictability of a Grid system.

• Disadvantages:
  – Incorporating such mechanisms into current Grid environments is a challenging task due to the resulting resource fragmentation.
  – Reservations may not be always feasible:
    • Not all the LRMS permit them.
    • There are other types of resources which lack a global management entity (bandwidth).
This is the reason to perform **meta-scheduling in advance** rather than advanced reservations to **provide QoS** in Grids.

- **Deadline** is a measure of the QoS required by the user.

**Meta-scheduling in advance:**

- First step of an advance reservation.
- It selects the resource and the time period to execute the job.
- It does not make any physical reservation.

**The main challenge:**

- Without knowing the exact status of the resources at future points in time it is difficult to decide whether a job can be executed fulfilling its QoS.
OUTLINE

• Introduction
• Meta-scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions
META-SCHEDULING IN ADVANCE

• Problems to offer QoS in Grids environments using advanced reservations:
  – They are **not always possible**.
  – Cause severe **performance degradation** because algorithms are complex.
  – They **lack flexibility** as they do not permit graceful degradation in application performance.

• Required features:
  – It must take into account resource **heterogeneity**.
  – It needs to **adapt to** dynamic changes in resource availability and user demand without hurting system and user performance.
  – Algorithms need to be **efficient**.
    • Employing techniques from **computational geometry** to develop an efficient heterogeneity-aware scheduling algorithm.
  – A good **running time prediction** of tasks.
META-SCHEDULING IN ADVANCE

- An **scheduling in advance process** is done following these steps:
  - First, a **“user request”** specifying the job QoS requirements is received.
  - The meta-scheduler executes a **“gap search” algorithm** to obtain the resource and the time interval to execute the job.
    - It keeps track of the meta-scheduling decisions already made in order to make future decisions.
    - It has into account the status reported by the resources.
    - It has into account the QoS requirements of the job.
  - If it is not possible to fulfill the QoS requirements using the resources of its own domain, the **communication with other meta-schedulers** allocated in other domains starts.
  - If it is still not possible to meet the QoS requirements, a **negotiation process with the user** is started to define new QoS requirements.
OUTLINE

• Introduction
• Meta-scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions
IMPLEMENTATION

- Introduction
- Meta-Scheduling In Advance
- Implementation
- Experiments and Results
- Conclusions

**SA-layer**

- Intermediate layer between users and GridWay.
- SA-layer uses functions provided by GridWay.
- Resource usages are divided into time slots of 1 minute.
IMPLEMENTATION

• Introduction
• Meta-Scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions

DATA STRUCTURE:

– Reduces the complexity of algorithms.

– It has influence on how scalable the algorithm is.

• Red black trees.

– Efficiently identify feasible idle periods.
IMPLEMENTATION

GAP MANAGEMENT:

- The way of allocating the jobs influences in how many jobs can be scheduled because of generated **fragmentation**.

- Implementation:
  
  - A **First Fit** policy.
  
  - Techniques from **computational geometry**.
IMPLEMENTATION

PREDICTOR:

- Extension of algorithm proposed by Castillo et al.:
  - To take into account the heterogeneity of Grid resources.
  - To not need an “a priori” knowledge of the jobs duration into resources.

- The monitoring information collected is kept in databases and reused for the next resource allocation decisions.
IMPLEMENTATION

• Two ways of calculating estimations for job completion times:
  – Based on a linear function (Castillo et al. proposal).
  – Based on executions data log.

• The linear function:
  – Does not take into account the different resource performance.
  – Only the input parameters of the job and the knowledge about its behaviour.
  – All the resources are treated as homogeneous.

• The data logs:
  – The resource heterogeneity is taken into account.
  – The mean of the completion times from previous executions for each type of application is calculated.
IMPLEMENTATION

- Introduction
- Meta-Scheduling In Advance
- Implementation
- Experiments and Results
- Conclusions

- Two applications are considered to belong to the same type when they have the same input and output parameters.

- This mean is calculated for each host separately, taking into account the host where previous executions were performed.

- Predictions on the completion time are calculated for each type of application for each host in the system.
  - These predictions are only calculated when a suitable gap has been found in the host.
OUTLINE

- Introduction
- Meta-scheduling In Advance
- Implementation
- Experiments and Results
- Conclusions
**EXPERIMENTS AND RESULTS**

- **Introduction**
- **Meta-Scheduling In Advance**
- **Implementation**
- **Experiments and Results**
- **Conclusions**

**Testbed description:**

- These machines belong to other users.
- They have their own local background load.
EXPERIMENTS AND RESULTS

Workload:

- 3Node from the GRASP benchmarks.

- Parameterizable options:
  - To make it more computing intensive (compute_scale parameter)
  - To make it more network demanding (output_scale parameter).

- Important parameters of the workload:
PERFORMANCE EVALUATION

• Scheduled job rate
  – Fraction of accepted jobs.

• QoS not fulfilled
  – Number of jobs rejected.
  – Number of jobs that do not meet their deadlines.

• Overlap
  – Minutes that a job execution is extended over the calculated estimation.

• Waste
  – Minutes not used to execute any job because the meta-scheduler thought that jobs would need more time to complete their executions.
EXPERIMENTS AND RESULTS

USERS POINT OF VIEW

Scheduled Jobs

<table>
<thead>
<tr>
<th>% Scheduled Job</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>GridWay</td>
</tr>
</tbody>
</table>

QoS Not Fulfilled

<table>
<thead>
<tr>
<th>% QoS Not Fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>GridWay</td>
</tr>
</tbody>
</table>

Data Log estimations exhibits better performance
EXPERIMENTS AND RESULTS

SYSTEM POINT OF VIEW

Overlap

Waste

Data Log estimations are more accurate
OUTLINE

• Introduction
• Meta-scheduling In Advance
• Implementation
• Experiments and Results
• Conclusions
CONCLUSIONS

• Providing QoS in Grids by means of advanced reservations is not always feasible.

• We proposed scheduling in advance as a possible solution to provide QoS.

• This requires to tackle many challenges.

• It is highlighted the importance of:
  – Using scheduling in advance to meet the QoS requested by users.
  – Taking into account the heterogeneity of Grid resources in the job completion time estimations.

• Meta-scheduling in advance and advanced reservation in Grid environments are open fields that still need research.

• Our work is being carried out in a real Grid environment.
CONCLUSIONS

Future work:

- To differentiate the network transfer time from execution time of the jobs.
- Job rescheduling:
  - It is needed whenever a resource leaves the Grid.
  - It can improve the job scheduled rate by reschedule job already scheduled in order to accept other jobs that have a more restrictive QoS requirements.
Meta-Scheduling in Advance using Red-Black Trees in Heterogeneous Grids

Luis Tomás, Agustín Caminero, Carmen Carrión, Blanca Caminero

Dept. of Computing Systems
The University of Castilla La Mancha
Albacete, Spain