SIMGRID_Scheduler

A Simulated Scheduling System for GRID Environment

Marcello CASTELLANO, Giacomo PISCITELLI and Nicola SIMEONE
Department of Electrical and Electronic Engineering, Politecnico di Bari
Domenico DIBARI, Eugenio NAPPI INFN Bari
Goal
Evaluation of super-scheduling algorithms

Why Simulate?

• We haven’t a working GRID system yet
• We can test scheduling algorithms without occupying physical resources
• The simulated environment acts like the physical one should do (no bugs and failures)
• Simulation is faster (we don’t have to wait the real execution of the job)
Identifying the Broker/Scheduler System

The Grid Architecture

Users → Workload Management System → Grid Information Space → Resources
Identifying the Broker/Scheduler System

Ref: Integrating GRID tools to build a Computing resource broker: activities of DataGrid WP1
Identifying the Broker/Scheduler System

Broker/Scheduler System Architecture

**SCHEDULER**
- Job specifications
- Job specifications

**BROKER**
- Job specifications
- Job specifications
- Job specifications + Resource Specifications
- Job Status

Resource Info
Formalizing the Model

```
user
- maxcount
- maximinmem
- maxmaxmem
- maxduration
- maxdelay

- generate_request(maxcount,maximinmem,maxmaxmem,maxduration,maxdelay)

scheduler
- scheduling_policy

+ submit(count,minmem,maxmem,duration,delay)
+ generate_index(scheduling_policy,count,minmem,maxmem,duration,delay)

requestbuffer

+ insert(index,count,minmem,maxmem,duration,delay)
+ pop(count,minmem,maxmem,duration,delay)

broker
- brokering_policy

information_space

+ query(count,minmem,maxmem)
+ update(name,jobsinqueue)
```

resource
- name
- maxtime
- maxjobsinqueue
- maxcount
- totalnodes
- freenodes
- maxtotalmemory
- maxsinglememory
- jobsinqueue
- refresh

+ run(count,minmem,maxmem,duration,delay)

arrow

arrow
Formalizing the Model: How it works
Formalizing the Model: Configuration Files

• There are 3 configuration files:
  – CL.INI (Computing Level configuration file)
  – RG.INI (Request Generator configuration file)
  – SG.INI (Simulator configuration file)
Formalizing the Model: Computing and Storage Resources

An Example - The Configuration File - CL.INI

<table>
<thead>
<tr>
<th>#</th>
<th>Max Time</th>
<th>Max Jobs in Queue</th>
<th>Max Count</th>
<th>Total Nodes</th>
<th>Free Nodes</th>
<th>Max Total Memory</th>
<th>Max Single Memory</th>
<th>Refresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>64</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>64</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>64</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>7</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>64</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>64</td>
<td>256</td>
<td>5</td>
</tr>
</tbody>
</table>
Formalizing the Model: Users

An Example - The Configuration File - RG.INI

%Final Time [s]: 3000;
%Request Rate (probability of 1 request in 1 sec) [%] : 10;
%Max number of executions of the same job requested (maxcount) : 3;
%Min memory needed by job [Mb] (maxmaxmem) : 16;
%Max memory needed by job [Mb] (maxminmem) : 128;
%Max Duration Time for a job [s] (maxduration) : 500;
%Max Delay Time for a job [s] maxdelay : 100;

<table>
<thead>
<tr>
<th>Job Specifications</th>
<th>How we generate them</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of executions of the same job</td>
<td>Uniformly between 1 and maxcount</td>
</tr>
<tr>
<td>Minimun Memory Needed</td>
<td>Uniformly between 1 and maxminmem</td>
</tr>
<tr>
<td>Maximun Memory Needed</td>
<td>Uniformly between minmem and maxmaxmem</td>
</tr>
<tr>
<td>Duration (foreseen)</td>
<td>Uniformly between 1 and maxduration</td>
</tr>
<tr>
<td>DelayTime</td>
<td>Uniformly between 1 and maxdelay</td>
</tr>
</tbody>
</table>
Formalizing the Model: Information Space

- It reports the state of the resources
- It is updated by resources every \textit{refresh} seconds or by an event
Formalizing the Model: Scheduler

An Example - The Configuration File - SG.INI

%Broker Rate (number of request managed by broker in 1 s) [s]: 4;
%Scheduling policy 0=EDF 1=FCFS 2=SJF: 0;

- The scheduler calculates the index using the chosen algorithm and puts the user request in the buffer

```
user
- maxcount
- maxminmem
- maxmaxmem
- maxduration
- maxdelay

- generate_request(maxcount,maxminmem,maxmaxmem,maxduration,maxdelay)
```

```
scheduler
- scheduling_policy

+ submit(count,minmem,maxmem,duration, delay)
+ generate_index(scheduling_policy,count,minmem,maxmem,duration, delay)
```

```
requestbuffer
+ insert(index,count,minmem,maxmem,duration, delay)
+ pop(count,minmem,maxmem,duration, delay)
```
Formalizing the Model: Broker

- The Broker gets the first request in the buffer
- Queries the Information Space
- Dispatch the Job
- If it doesn’t find the suitable resource, it resubmits the request to the scheduler
Metrics

Cumulative value [0..100]
Missed deadlines [per cent]
System Usage [per cent]
Average Queue time [per cent of total time]
Average Execution time [per cent of total time]

\[
\text{Cumulative Value} = \sum_i \frac{\text{value}_i (\text{stop time}_i)}{\text{total execution time}_i}
\]
Status Report

• We have implemented 3 scheduling algorithms
  – FCFS First Come First Served
  – EDF Earliest Deadline First
  – SJF Shortest Job First

More algorithms will be added

• We have developed a first level simulation

• Brokering strategies need to be improved
Technologies

• UML is used to analyze and design the system model and the simulation software

• C++ is used to develop software so that we can easily:
  – reuse the same code to implement a real superscheduler
  – use code developed by others (WP1)

• Telelogic Tau
FCFS SIMULATION RESULTS

Cumulative value \([0..100]\) : 79
Missed deadlines \([\text{per cent}]\) : 23
System Usage \([\text{per cent}]\) : 65
Average Queue time \([\text{per cent of total time}]\) : 13
Average Execution time \([\text{per cent of total time}]\) : 87
Final Time : 10895
EDF SIMULATION RESULTS

Cumulative value [0..100] : 85
Missed deadlines [per cent] : 16
System Usage [per cent] : 65
Average Queue time [per cent of total time]: 10
Average Execution time [per cent of total time]: 90
Final Time : 10998
SJF SIMULATION RESULTS

Cumulative value [0..100] : 85
Missed deadlines [per cent] : 15
System Usage [per cent] : 65
Average Queue time [per cent of total time]: 10
Average Execution time [per cent of total time]: 90
Final Time : 10913
Next Step

• to consider the HEP computing model (Ex. Monarc project)
• to choose a class of applications (Ex: ALICE experiment apps)
• to determine an appropriate scheduling algorithm
• to evaluate the right parameters

in order to design and develop a real

“Community Broker Scheduler”

using the DATAGRID tools available