Understanding and mastering dynamics in computing grids: processing moldable tasks with user-level overlay

Mościcki, J.T.

Citation for published version (APA):
CHAPTER 6

Capability computing case study: ITU broadcasting planning

Grids, as federations of batch systems, are not designed to handle short-deadline, capability-computing applications. In this Chapter\textsuperscript{1} we present a case study of an application for planning of the digital terrestrial broadcasting service for a large number of member states of the International Telecommunication Union (ITU\textsuperscript{2}). We explain how the User-level Overlay was used to implement a dependable, just-in-time service based on the EGEE Grid where fast and reliable response is a key metric of success for a mission-critical application.

6.1 Introduction

The radio-frequency spectrum is a limited natural resource which must be used rationally, efficiently and economically, so that countries or groups of countries may have equitable access to it [96]. The analogue terrestrial broadcasting has been regulated since 1961 by the Stockholm Agreement in Europe (ST61) and since 1989 by the Geneva Agreement for Africa (GE89). The necessity of new regulations has become increasingly important due to a fast development of digital broadcasting technologies in recent years.

From 15 May 2006 to 15 June 2006 a session of the Regional Radiocommunication Conference (RRC06), organized by the International Telecommunication Union (ITU), was held in Geneva. Delegations from 104 countries in Europe, Africa, Middle-East and ex-USSR gathered to negotiate a new frequency plan for the digital terrestrial


\textsuperscript{2}International Telecommunication Union, \texttt{http://www.itu.int}
broadcasting services in UHF (470-862 MHz) and VHF (174-230 MHz) bands. The resulting frequency plan became a part of a new international agreement, the RRC06 Final Acts [2], which enables an efficient move to the new era of digital broadcasting at international level. Fig. 6.1 shows the area covered by the RRC06.

Preliminary analysis indicated that one component of the planning process, the compatibility analysis, was highly CPU intensive. The goal of the compatibility analysis is to evaluate the interference between broadcasting stations to identify those that can share the same channel. The analysis includes several parameters of the broadcasting stations such as the geographic location, the signal strength and other technical characteristics.

Total computing capacity required for the compatibility analysis was estimated at few hundred CPU-days on a high-end 2006 PC. The RRC06 required the output of the compatibility analysis to be delivered at the specified deadline and within few hours of computing time (this corresponds to 100-500 speedup factor). The problem may be therefore described as on-demand, capability computing in the grid.

### 6.2 Broadcasting planning process

The RRC06 planning process consisted of several iterations which interleaved the compatibility analysis step and assessment of the analysis results, shown as a loop in Fig. 6.2. The output of the assessment step was used as an input for the subsequent analysis iteration. One iteration of the loop extended to one full week: the assessment of the analysis
The compatibility analysis was performed in several iterations as a part of the analysis loop. The actual iteration could implement one of the two loop variants (with or without synthesis step).

Results from a previous iteration were performed during weekdays and were followed by the compatibility analysis step (with optional synthesis) performed at weekends.

The assessment step was based on bilateral and multilateral negotiations and coordination discussions between 1200 representatives of the ITU member states at the RRC06 conference. A new, refined version of the frequency plan was produced at the end of each week and was used as an input to the next compatibility analysis iteration.

The workload of one compatibility analysis run at the RRC06 corresponded to several hundred CPU hours to be completed within a few hours. The time constraint was critical: a problem with timely delivery of analysis results would have resulted in a failure of the international negotiations.

### 6.3 Compatibility analysis

The compatibility analysis is a calculation of the interference between digital broadcasting stations and services, using established statistical models of signal propagation such as the ITU-R Recommendation P1546-1 [97]. Radio communication services are described by administrative and technical parameters, so called “broadcasting requirements”\(^3\). For example, administrative parameters include the name of the ITU member state (“notifying administration”), site name, geographic location, site altitude. Technical parameters include the power levels, assigned frequency, network topology, etc. The input data for the compatibility analysis is a set of broadcasting requirements and for the RRC06 consisted of about $95 \times 10^4$ digital requirements, about $95 \times 10^4$ analog TV requirements.

\(^3\)Digital requirements are specified using T-DAB (radio) or DVB-T (television) standards.
requirements and $10^4$ requirements for other services. In addition, a few millions of so-called “administrative declarations” were included to indicate which requirement conflicts might be safely ignored. The analysis was performed for two frequency bands, VHF and UHF, which resulted in six types of analysis tasks: digital versus digital (d2dUHF and d2dVHF), digital versus other services (d2oUHF and d2oVHF) and other services versus digital (o2dUHF and o2dVHF).

In the compatibility analysis each requirement must be run against all the others for six different types of analysis tasks. The term atomic calculations is used to refer to individual, indivisible calculations defined in compatibility analysis datasets. The term task refers to a unit of work which corresponds to a set of atomic calculations. The term job refers to a grid worker agent.

Further investigation showed that a complete static optimization of the workload, i.e. clustering of atomic calculations such that the execution time of each cluster (task) is equal, was not possible. This is because a change of requirements in between the analysis iterations resulted in large and unpredictable changes in the computational

Figure 6.3: Distribution of the number of processed requirements per hour for d2dUHF analysis type for different ITU member states.
Table 6.1: Number of atomic calculations per task (task granularity) for each analysis round and the six analysis types.

<table>
<thead>
<tr>
<th>iteration</th>
<th>d2dUHF</th>
<th>d2dVHF</th>
<th>d2oUHF</th>
<th>d2oVHF</th>
<th>o2dUHF</th>
<th>o2dVHF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

complexity of the atomic calculations.

However, there was clearly a need to create small clusters of atomic calculation for the most CPU demanding type of analysis d2dUHF and d2dVHF minimizing the spread between the shortest and longest tasks. Table 6.1 shows the granularity for the different types of analysis in the RRC06 iterations for the EGEE Grid. The granularity was adjusted manually in between the iterations.

### 6.4 Implementation of grid-based analysis system for the RRC06

The computing support system to perform the compatibility analysis for the RRC06, shown in Fig. 6.4, was composed of two, redundant systems for improved dependability: a dedicated farm of 84 high-end desktop PCs deployed at ITU headquarters and the EGEE Grid infrastructure. Analysis was run by few users simultaneously with the two subsystems. Common monitoring and accounting system was provided by a MonALISA service\(^4\). The ITU system is described in more detail in [141].

Grid-based system was implemented with D\_IANE/GANGA User-level Overlay. Compatibility analysis application modules were developed and plugged in the D\_IANE framework. The job submission was done by GANGA using standard gLite backend plugin. Distribution of input data and software installation was performed by a VO manager in a separate step.

Task were defined by a set of application-specific parameters: a pair of requirement identifiers and an identifier of an analysis type. Task execution consisted of running a standalone executable on a pre-installed input dataset and with appropriate parameters. The task dispatching performed by the RunMaster consisted of selecting a task from the head of the task queue and allocating it to a next available worker by transferring appropriate parameters (c.f. Fig. 4.3). The communication overhead in this case is typically much smaller than in the systems based on checkpointing and task migration and it allows scheduling with a high rate of incoming and outgoing tasks. The D\_IANE RunMaster routinely achieved peaks of 110-120 Hz without observable degradation of the performance. This means that scheduling overhead is negligible for up to \(N \times 120\) worker agents if average task duration is \(N\) seconds.

\(^4\)MonALISA: Monitoring Agents in A Large Integrated Services Architecture, [http://monalisa.cern.ch](http://monalisa.cern.ch)
To improve dependability, some key services, such as the Master servers and critical gLite WMS services were deployed in several instances to increase the redundancy and to allow failover in case of problems. For storing the application output a globally distributed filesystem (AFS) and local filesystem were used simultaneously.

Reliable and fast delivery of the compatibility analysis results was a key to success of the RRC06. Each compatibility analysis run was preceded by an update of analysis software and input data with 2 hour notice from the RRC06 operational team. In this time-window the grid-based system had to be up and ready to start the analysis at full speed. The distribution of updated input data was the first step of compatibility analysis run and it was performed by a VO-manager using a set of grid jobs which downloaded a 100 MB installation package from a central repository, ensuring consistency with MD5 checksums.

The CPU demand for compatibility analysis was estimated at 500-1000 CPUh per iteration what is smaller than for other grid applications. On the other hand the availability of resources within well-defined and strict time constraints was critical. Therefore, high-availability centers\(^5\) in the EGEE Grid were involved. The resources at these centers were not dedicated to the RRC06 activity, however, the job priority parameters were adjusted during short periods of intensive processing of the RRC06 compatibility analysis. On average 300 CPUs were observed to be available at all times with occasional peaks of 600 CPUs.

\(^5\)CERN, CNAF+few other sites(I), PIC(E), DESY(D), MSU(RU), CYFRONET(PL)
Figure 6.5: Workload executed shown as the number of processed broadcasting requirements in the selected EGEE Grid sites and the two ITU desktop farms: RRC06-1.itu.org and RRC06-2.itu.org. Data provided by MonALISA monitoring service.

Fig. 6.5 shows the total workload executed by ITU clusters and the EGEE sites. Fine-grained monitoring of the grid worker nodes and ITU farm nodes to produce in real-time, high-level reports and charts delivered by the MonALISA framework which provides a set of pluggable distributed services for monitoring, control, management and global optimization for large-scale distributed systems.

6.5 Analysis of task processing

The summary of the RRC06 iterations is presented in Table 6.2. For each analysis iteration the total workload consisted of $N_{\text{calc}}$ atomic calculations. The calculations were executed in bunches according to previously defined static clustering (section 6.3). The reliability of DIANE/GANGA system exceeds by several orders of magnitude the reliability of standard job submission: in run 1 less than 10 tasks were lost, in run 2 only one task was lost while in runs 3 and 4 all tasks were successfully completed.

The total CPU demand decreased with each RRC06 iteration. The member states decreased the number of requirements and the number of acceptable channels for each requirement, therefore reducing the total workload at each analysis iteration. As the frequency plan was refined during successful negotiations between the member states, the number of conflicting requirements also decreased.

During pre-conference preparatory planning activities only 34% of requirements were satisfied. At the first iteration of the RRC06 the percentage increased to 64% (UHF) and 74% (VHF), to reach a satisfactory 93% (UHF) and 98% (VHF) for the final plan.
### Table 6.2: Summary of the RRC06 compatibility analysis iterations.

<table>
<thead>
<tr>
<th>iteration</th>
<th>$N_{calc}$ ($\times 10^3$)</th>
<th>$N_{task}$ ($\times 10^3$)</th>
<th>$t_{total}$ (hours)</th>
<th>$t_{worker}$ (hours)</th>
<th>$N_{worker}$</th>
<th>$r_{fail}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>243</td>
<td>26</td>
<td>6.66</td>
<td>425</td>
<td>190</td>
<td>$&lt;3 \times 10^{-4}$</td>
</tr>
<tr>
<td>2</td>
<td>237</td>
<td>23</td>
<td>6.50</td>
<td>332</td>
<td>125</td>
<td>$4 \times 10^{-5}$</td>
</tr>
<tr>
<td>3</td>
<td>224</td>
<td>40</td>
<td>1.58</td>
<td>192</td>
<td>210</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>218</td>
<td>39</td>
<td>1.01</td>
<td>151</td>
<td>320</td>
<td>0</td>
</tr>
</tbody>
</table>

The $N_{task}$ tasks were distributed dynamically to the $N_{worker}$ Worker agents. The Worker agents were submitted as jobs and executed on the grid worker nodes. $t_{total}$ is the makespan or the total time to complete the compatibility analysis. $t_{worker}$ is the integrated elapsed time on the worker nodes. $r_{fail}$ is the reliability of the system and corresponds to the number of failed tasks which could not automatically recover.

### Figure 6.6: Distribution of the wallclock task execution time.

Most of 40000 tasks execute in less than 10 seconds; individual tasks execute in 1000 seconds.
Due to grid dynamics, a different set of worker nodes was used at each iteration, resulting in different CPU and memory characteristics of the worker nodes actually used. Therefore, it is not possible to easily normalize and compare the execution times $t_{\text{total}}$ and $t_{\text{worker}}$. The distribution of wallclock execution time of the tasks is almost exponential and is shown in Fig. 6.6.

### 6.5.1 The 3-phase anatomy of analysis runs

The task processing efficiency depends on the grid job submission latency, efficiency of task scheduling and workload balancing at the end of the run. Fig. 6.7 shows the workload and task processing history for selected runs. Three processing phases may be distinguished. $N_w$ worker agents are submitted at $t_0 = 0$. In the submission phase, $t < t_1$, the throughput of the system is limited by the submission latency. As the pool of worker nodes increases the target of $N_w$ workers is reached at time $t_1$. In the main processing phase, $t_1 < t < t_2$, the pool of worker nodes remains stable and the system throughput mainly depends on the efficiency of scheduling. At time $t_2$ the number of remaining tasks becomes smaller than the number of processors in the pool and the system enters the final phase, in which the execution time is dominated by the workload-balancing effects from the few slowest tasks.

The number of available worker nodes may vary significantly in the grid from one run to another. The contribution of the job submission latency to the total execution time may be approximated by the area between the target line (requested size of the worker pool achieved when the system enters the main phase) and the curve representing the actual size of the worker pool (see Fig. 6.7) In run 3 the latency of job submission corresponded to 12% of the total execution time, whereas in run 4 it corresponded to 48%: 33% in the submission phase and 15% in the main processing phase.

The integrated difference between the worker pool size and the number of busy workers corresponds to the scheduling overhead. This overhead includes the network latency and throughput as well as the task handling efficiency of the master server. In run 3 the scheduling overhead in the submission and processing phases corresponded to 2-3%. In run 4 the 30% scheduling overhead in the submission phase was observed and 10% in the processing phase.

The unbalanced execution of the slowest tasks in the last phase contributed to 26% of the total execution time in run 3 and to 5% in run 4. In this phase the utilization of available resources was very low, 5% in run 3 and 20% in run 4. The majority of the workers in the pool remained idle while the few remaining tasks were being finished.

In other words in run 3 task scheduling overhead was very small but workload balancing was poor. Conversely, in run 4 task scheduling overhead was large but workload balancing was good.

### 6.5.2 Impact of task ordering on load balancing and efficiency

Why in runs 3 and 4 such radical differences in scheduling overheads and quality of workload balancing were observed? This may be explained by radically different ordering of tasks in both runs, which may be analysed using a workload distribution graph.
Figure 6.7: Workload and task processing history plot for run 3 and 4 showing the size of the worker pool, number of busy workers and number of processed tasks per minute. The solid target line ($N = 212$ for run 3 and $N = 190$ for run 4) indicates the requested worker pool size. Data is sampled in 60 s bins. In run 4 two parallel master servers were used and this figure corresponds to one of the masters hence to a half of the total processed workload in run 4.
Figure 6.8: Workload distribution graphs showing the completion time of tasks for run 3 (random task order) and run 4 (natural task order as defined in the ITU dataset). One point corresponds to one completed task. Higher density of points indicates higher load on the task scheduler.
for a late-binding system introduced in Fig. 3.2 in Chapter 3. The actual workload distribution graph for runs 3 and 4 are shown in Fig. 6.8, where a task completed by a worker $w$ at time $t$ is represented as a point $(t, w)$. In run 4 the tasks were dispatched in the natural order of atomic calculations as defined in the input data set. It happened that the atomic calculations (requirements) were defined country-by-country in the ITU frequency plan. Therefore, clusters of very short tasks could occasionally generate very high load on the server. By a pure chance, the longest tasks were processed in the middle of the run and did not affect the overall load-balancing. On the other hand, in run 3 the tasks were dispatched in a random order by the scheduler. The momentary load on the server was reduced as long and short tasks were more uniformly distributed in time across the entire run. However, there were a few long tasks at the end of the run that resulted in poor load-balancing. This effect is systematic and occurs with a probability proportional to the number of long tasks in the dataset (the phenomenon occurs if long tasks are drawn at the end of a run). Both task ordering methods inevitably result in inefficiencies.

6.6 Summary

The system based on the User-level Overlay in the EGEE Grid contributed to the success of the RRC06 Conference. The GE06 frequency plan is now a part of a new international treaty.

The intrinsic job submission latency in grids makes it hard to run a large number of short jobs in a short time. For the RRC06 using the User-level Overlay allowed to reduce grid overheads and provided efficient management of a large number of tasks. Additionally a runtime workload balancing allowed to evenly distribute the workload without precise, a priori knowledge of the task execution times in the dataset. The overhead reduction and dynamic workload balancing were the crucial factors of the successful usage of the EGEE Grid for the RRC06, however, due to statistical effects of workload distribution the efficiency of task processing may not be always optimal.

A seamless access and integration at the application level of grid resources and corporate infrastructures may be beneficial for other user communities. A typical use-case could include dedicated in-situ resources for fast response and grid resources when facing peak demand. In such a scenario, grids could provide a competitive alternative to traditional procurement of resources. The EGEE Grid delivered dependable peak capacity to an organization which normally does not require a large permanent computing infrastructure. Additionally grid technology was successfully used in a new area to provide a dependable just-in-time service with limited support and training required by the ITU personnel.