Towards Profile-Based Task Management in Grids

Antonio R. Mury,
Bruno Schulze,
Antônio T.A. Gomes
ComCiDis - Distributed Scientific Computing Group
National Laboratory for Scientific Computing
Petrópolis – Brazil
{aroberto, schulze, atagomes}@ lncc.br

Summary

- Paper Proposal
- Treating Submissions
- Standard Tasks and Building Profile
- Task Queues Load Balancing
- Fault Detection and Recovery
- Performance Analysis
- Load Balancing Simulation
- Summary and Outlook

Paper Proposal

- *A procedure for initial task submission in a Grid Computing environment*

- *The presented model proposes:*

  - *A profiled-based task management*
    - *Pre-treatment of the submissions,*
    - *Load balancing the distribution of the tasks*
    - *Fault detection recovery, and*
    - *Performance analysis of the grid.*

Paper Proposal

- *The main difficulties regarding task management in grids include:*
  - *The decentralized nature of grids,*
  - *The heterogeneity and dynamism of their resources, and*
  - *High communication costs as compared with local clusters.*

Paper Proposal

- *Many approaches have been proposed in the context of task scheduling.*
  - *They have focused on adaptive strategies that try to predict either resource or task loads.*
  - *The load prediction in these approaches disregard the characteristics of the tasks from a specific user or group of related users.*

Paper Proposal

- *This paper proposes a distribution model for tasks in grids that takes into account:*
  - *The expected time of execution of a task from a particular user/group in a particular resource,*
  - *This expected time is refined as the user submits new tasks to the grid, allowing the maintenance of a user profile that links the user, her task, and the used resources, and*
  - *This profile may then be used for the purposes of both load balancing and fault recovery.*

Treating Submissions

- *In a grid, it is important to monitor not only the status of the resources but also the time spent for a task completion in a specific resource.*

- *This monitoring is intended to verify the availability and the level of usage of the resource at the time of the task submission.*

Treating Submissions

- *This approach allows:*
  - *SLA - meeting the minimum requirements for the task and calculating the expected run time of a task in that resource.*
  - *Fault detection recovery- it is possible to check if the resources chosen by the user are active and the extent of usage of such resources at that time.*

Treating Submissions

- *It is important to determine and keep control of the total estimated run time of a task.*
- *This time is started at the moment of the submission until the end of the task.*
- *This information is stored, monitored and estimated continuously.*

Treating Submissions

- *The initial estimate is extremely difficult given the heterogeneity of the resources*

- *It is based on the parameters of the tasks obtained from the user and the extent to which the user submits the tasks in the grid, this estimate will be refined on every new usage.*

Standard Tasks

- *The initial estimates could be obtained through the use of a set of standards tasks, with different complexities, submitted to the grid, from time to time.*

The usage of standards tasks can be extended to:

- *The analysis of performance of the resources in the grid,*
- *For comparison between resources,*
- *To measure the quality of services offered, and*
- *The determination of problems.*

Standard Tasks

- *These standard tasks, in case of a specific user, may also help in determining the set of resources that better fits his needs.*

- *Parameters of physical capacity and processing are important in determining the minimum requirements for the execution of a given task, but the use of standards tasks for the analysis of performance allows to estimate the actual behavior of the system given a specific demand.*

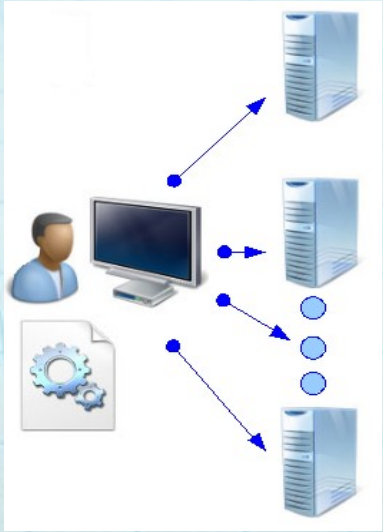
Building Profiles

- *From the previously mentioned it is possible to create a profile for the user and for the task, based on the estimated total time of execution, helping in the selection of the resources by the user.*

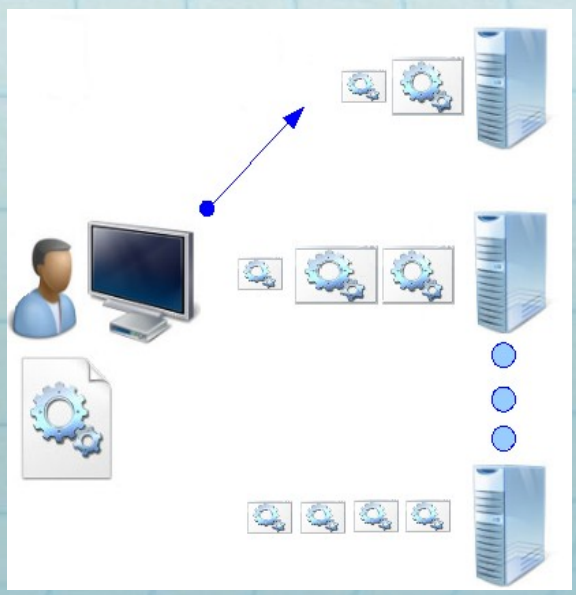
- *The immediate benefits from this control are:*
  - *Inform the user at the submission of a task the estimate of when the work / tasks will be finished;*
  - *Initial distribution of resources by the grid, and*
  - *Identification of deficiencies in the grid.*

Tasks Queues Load Balancing

- *The typical model of submission in grids is similar to the multiple-server queue model.*

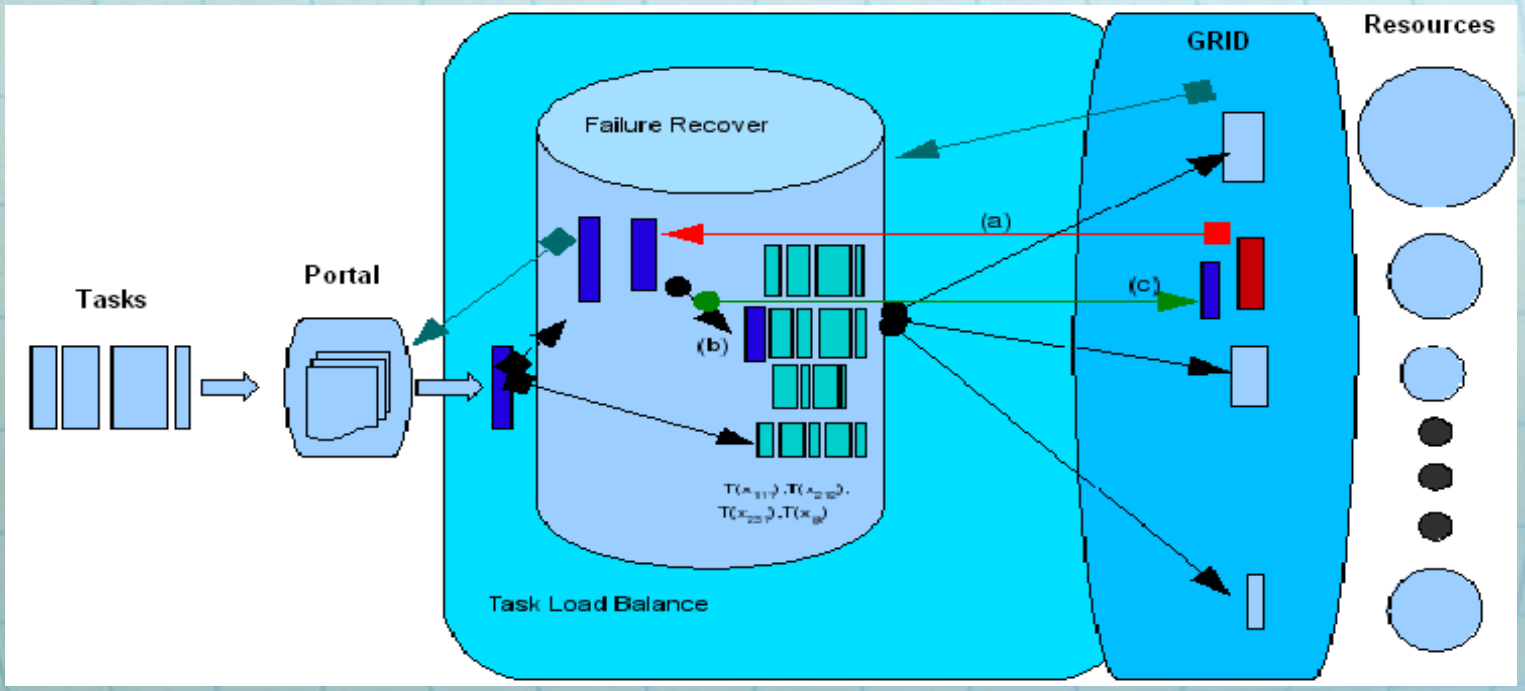


- *The model proposed here is similar to a multiple single-server queue model.*



Tasks Queues Load Balancing

- *This approach is possible because the distribution of each task takes into account the total waiting time on each queue.*



Tasks Queues Load Balancing

- *The distribution of new tasks to the queues will be based on the lower total estimated waiting time, thereby maintaining the balance of the tasks distribution for the queues.*

- *This approach mitigates the problem of choosing a resource already processing a task, while there are other resources available, even with lower performance and a more effective control of detection flaws in the executions of tasks.*

Fault Detection and Recovery

- *Besides providing load balancing among the resources in the grid, the proposed approach allows the detection of faults during task execution.*

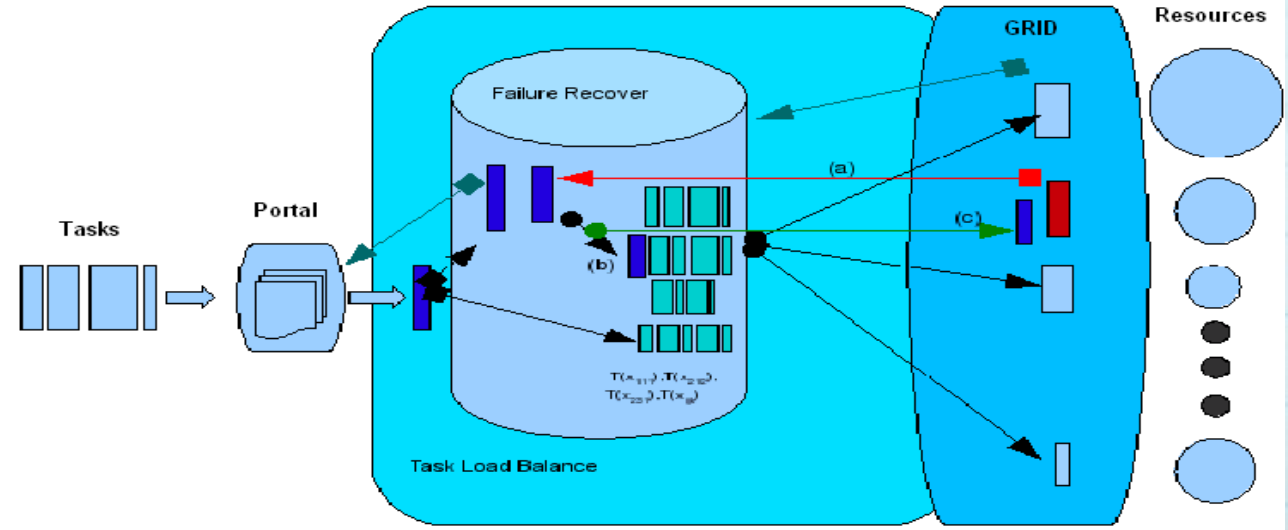
- *Basically, if the scheduled time to the completion of a given task is exceeded then two situations may be possible:*
  - *the task is being executed, but the time for its completion was underestimated;*
  - *the resource has become unavailable during execution.*

Fault Detection and Recovery

- *In the first case, there is a passive solution, by correcting all the estimates made for the tasks already allocated, and in a more extreme situation, by reallocating all the tasks.*

- *In the second case, all the tasks in a resource queue not yet processed will be reallocated.*

Fault Detection and Recovery



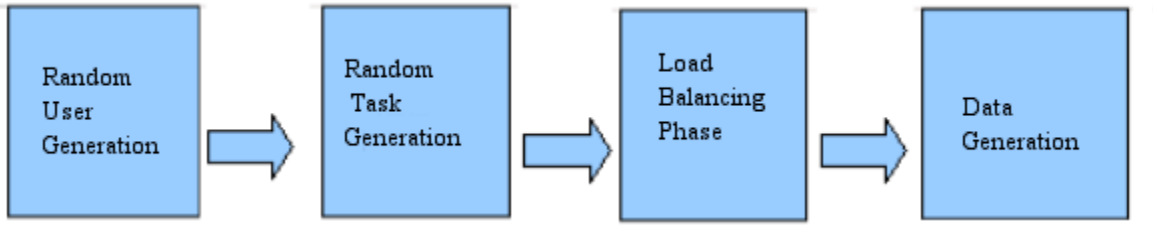
- *At the time of submission, the task will be stored on a temporary database and will remain there until it terminates.*
- *If a resource becomes unavailable (a), resulting in the loss of the tasks allocated to it, the tasks will be submitted to another resource (b) (c).*

Performance Analysis

- *Based on queuing theory, it is possible to establish a minimum level of performance of our system.*
- *This minimum level would be an indicator for the inclusion of new features/resources in the system.*
- *From the analysis of behavior of the tasks submitted to the grid, it is possible to identify those tasks that use more resources in order to identify the resources that better fit such tasks*

Load Balancing Simulation

- *The simulation of the submission process was made in order to evaluate the load balancing of the tasks submitted to the grid.*



- *The random user generation stage and the random task generation stage consist of the random selection of a user for the round, and the random generation of the weight of the task submitted by that user.*

Load Balancing Simulation

- *The weight of the task represents a degree of complexity to a specific resource.*
- *Resource “Server 2” has two times the processing capacity of resource “Server 1” .*

Load Balancing Simulation

- *The average time is calculated and its control is kept associated to the user, to the task and to the resource, i.e., each user has an average waiting time for each resource.*
- *The resource with the lowest total waiting time will be selected.*
- *The task is assigned to this resource.*

Load Balancing Simulation

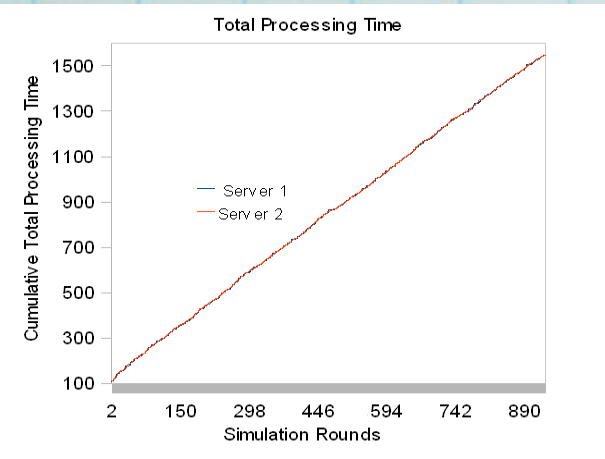
- *After that the real task time for the resource is calculated. This time is calculated based on the weight of the task and the weight of the resource.*
- *The weight of the resource is the processing capacity of this resource in particular.*
- *The value is obtained dividing the weight of the task by the weight of the resource. This value is then added to the value of the waiting queue on that resource.*

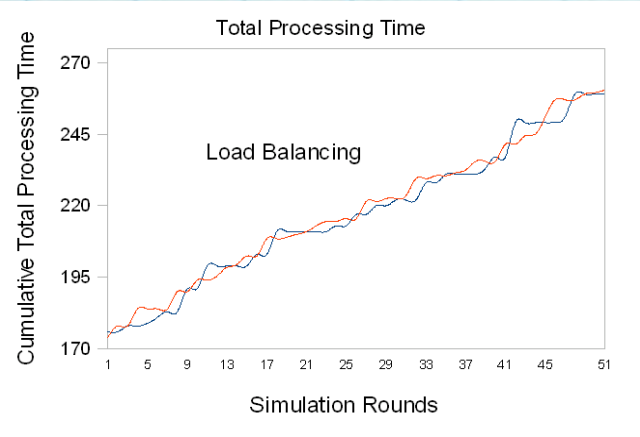
Load Balancing Simulation

- *For each resources in the queue, the value of their weight is then subtracted.*
- *This action represents the processing share completed in that round.*
- *The value obtained is used as the waiting time of the queue on the resource in the next round.*

Load Balancing Simulation

- *The cumulative processing time of the load balancing simulation using the proposed algorithm.*

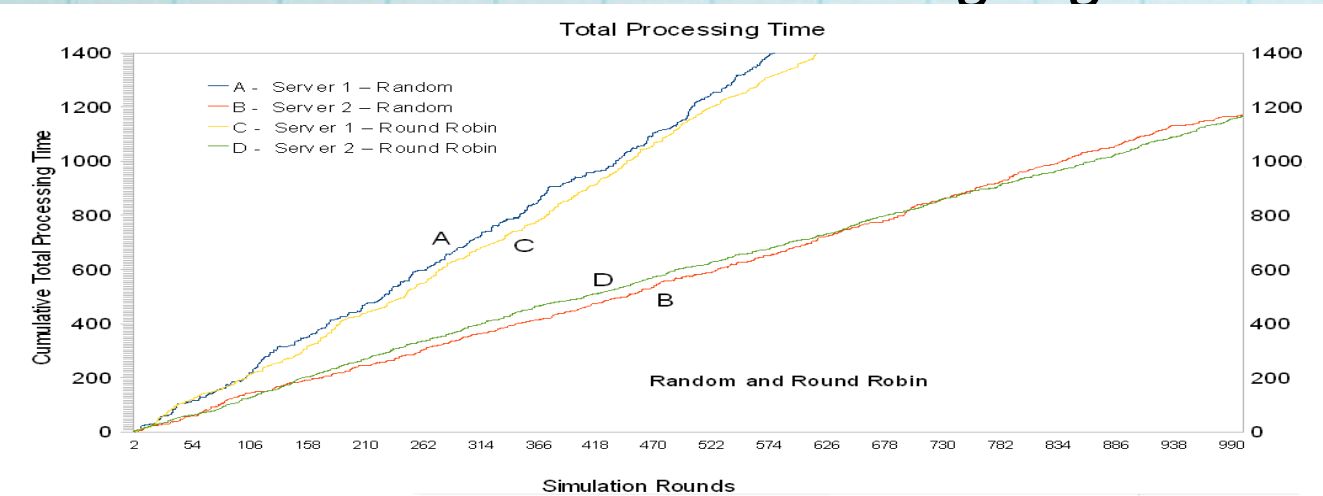




- *It is possible to notice that in both resources (Server 1 and Server 2) the cumulative processing time is equally distributed between them, during all the task submition and*

Load Balancing Simulation

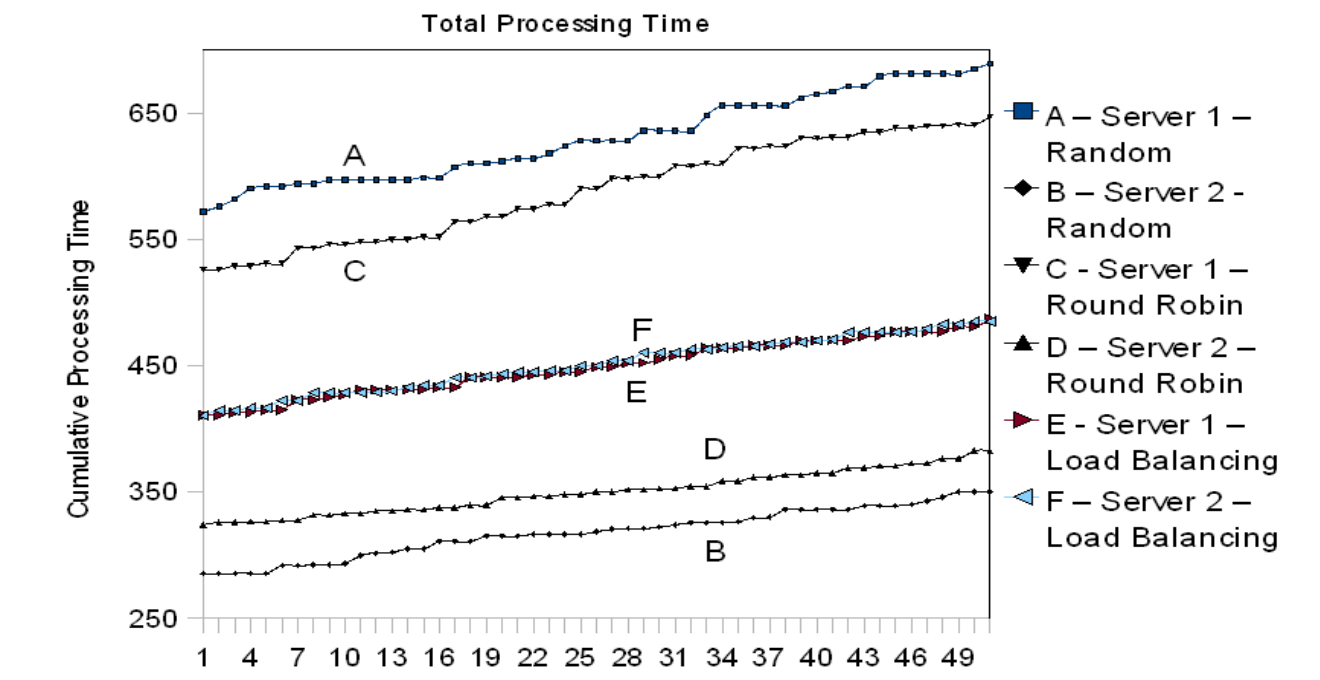
- *The same distribution process was made utilizing a random and round robin scheduling algorithm.*



- *It is possible to notice how the cumulative time processing are different between the resources in both of them.*

Load Balancing Simulation

- *Sample of the cumulative time processing, comparing the three distributions algorithms (LB- Load Balancing, Random and Round Robin)*



Load Balancing Simulation

- *Number of tasks submitted during simulations and the distribution between the resources utilizing the three algorithms.*
- *It is possible to see how the tasks were distributed and the effect over the processing time with the use of the load balancing algorithm proposed, even with the complexity of the tasks ranging from user to user and from task to task.*

Number of Tasks Submitted per Resource

	Load Balancing		Random		Round Robin	
	Res1	Res2	Res1	Res2	Res1	Res2
User1	150	169	168	174	168	166
User2	122	203	154	174	162	172
User3	96	260	180	158	170	162
P/Ser	368	632	502	498	500	500
Total	1000		1000		1000	

Resource total Processing Time Unit (tu)

1546 tu	1549 tu	2439 tu	1170 tu	2204 tu	1160 tu

Resource 2 has two times the processing capacity of Resource 1
Tasks of User2 has at least two times more complexity than User 1
Tasks of User3 has at least three times more complexity than User 1

Summary and Outlook

- *In this paper was proposed a profile-based task management.*
- *It is obtained by calculating and controlling the estimated time of execution of the tasks of a particular user on a given resource.*
- *This estimated time of execution is used in the grid submission process for the tasks scheduling, for the load balancing, for the fault recovery and for the performance analysis of the grid.*

Summary and Outlook

- *With this approach, the tasks may have their distribution based on the capacity of each resource, helping, mainly, the user with little knowledge of its operation, by allowing the coexistence of manual—as it is typically done today—and automated submissions in the grid.*

- *The main advantage of this type of load balancing strategy is its adaptation to the topology of the grid, assimilating both the entry of new resources and the unavailability of them*

Summary and Outlook

- *From the viewpoint of grid managers, this approach also allows them to have a quality control of existing services in the grid.*
- *Enabling them to detect failures and degradation in resource processing capabilities and network communication.*
- *It allows a grid manager to track the load that the resource is submitted and its variations along the period, getting parameters to determine which corrective actions must be undertaken to maintain and improve the quality of service.*

Summary and Outlook

- *The survey, the control and the analysis of the resources, according to the tasks submitted to them by the grid, is of fundamental importance in the determination of its performance.*

- *This performance indicator is one of the basics parameters for a policy implementation to provide services, that take into account the contribution of the projects to the grid installed capacity*

Summary and Outlook

- *As future work, we intend to use our task distribution model on the implementation of advanced services for the VCG grid, such as advance reservations and project prioritization.*

Acknowledgments

This work has support from the Brazilian National Research Council (CNPq) and the Virtual Community Grid Project.