

A Simulation Model for Large Scale Distributed Systems

Ciprian M. Dobre^{*} and Valentin Cristea^{**}

^{*} Politechnica University of Bucharest, Romania, e-mail: cipsm@cs.pub.ro

^{**} Politechnica University of Bucharest, Romania, e-mail: valentin@cs.pub.ro

Abstract

The use of discrete-event simulators in the design and development of large scale distributed systems is appealing due to their efficiency and scalability. Their core abstractions of process and event map neatly to the components and interactions of modern-day distributed systems and allow designing realistic simulation scenarios. MONARC 2, a multithreaded, process oriented simulation framework designed for modelling large scale distributed systems, allows the realistic simulation of a wide-range of distributed system technologies, with respect to their specific components and characteristics. In this paper we present the design characteristics of the simulation model proposed in MONARC 2. We demonstrate that this model includes the necessary components to describe various actual distributed system technologies, and provides the mechanisms to describe concurrent network traffic, evaluate different strategies in data replication, and analyze job scheduling procedures.

1. Introduction

The design and optimisation of large scale distributed systems requires a realistic description and modelling of the data access patterns, the data flow across the local and wide area networks, and the scheduling and workload presented by hundreds of jobs running concurrently and exchanging very large amounts of data.

MONARC 2, a simulation framework designed for large scale distributed computing systems, provides the necessary components to design realistic simulations of large-scale distributed systems and offers a flexible and dynamic environment to evaluate the performance of a wide-range of possible data processing architectures. The simulation model being proposed by MONARC 2 provides the mechanisms to describe

concurrent network traffic and to evaluate different strategies in data replication or in the job scheduling procedures.

In this paper we present the design considerations of the simulation model, as it incorporates all the necessary components and characteristics that allow the complete and accurate design of realistic simulation experiments of complex Grid architectures, consisting of many resources and various technologies, ranging from data transferring to scheduling and data replication, working together to provide a common set of characteristics.

The paper is organized as follows. Section 2 introduces the general simulation model proposed by MONARC 2. Section 3 describes some Grid characteristics that influenced the design of the simulation model. In section 4 we present a series of simulation results to demonstrate the capabilities of the model to represent a wide-range of Grid systems and their applications. In section 5 we present related work in this field and in section 6 we give the conclusions.

2. The simulation framework

MONARC 2 is built based on a process oriented approach for discrete event simulation, which is well suited to describe concurrent running programs, network traffic as well as all the stochastic arrival patterns, specific for such type of simulation. Threaded objects or "Active Objects" (having an execution thread, program counter, stack...) allow a natural way to map the specific behavior of distributed data processing into the simulation program.

In order to provide a realistic simulation, all the components of the system and their interactions were abstracted. The chosen model is equivalent to the simulated system in all the important aspects. A first set of components was created for describing the physical resources of the distributed system under simulation. The largest one is the regional center (see Figure 1), which contains a farm of processing nodes

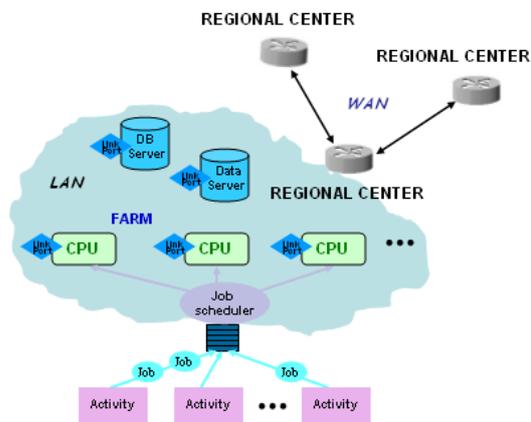


Figure 1. The Regional center model.

(CPU units), database servers and mass storage units, as well as one or more local and wide area networks. Another set of components model the behavior of the applications and their interaction with users. Such components are the “Users” or “Activity” objects which are used to generate data processing jobs based on different scenarios. The job is another basic component, simulated with the aid of an active object, and scheduled for execution on a CPU unit by a “Job Scheduler” object.

With this structure it is possible to build a wide range of models, from the very centralized to the distributed system models, with an almost arbitrary level of complexity (multiple regional centers, each with different hardware configuration and possibly different sets of replicated data). The analysis of the characteristics of various Grid architectures was essential in the design process of the simulation model. It influenced the decision on the type of components and interactions required to completely and correctly model various Grid related experiments.

3. The Grid influences on the design of the simulation model

Historically, the most important Grid architecture is that described in [5] known as the protocol oriented architecture. According to this view the Grid computing focuses on large-scale, multi-institutional, resource sharing to deliver high performance. Within a virtual organization (VO), participants belonging to member organizations are allocated shared resources based on request priorities. For the interoperability among potential participants in a VO the authors proposed the use of “protocols defining the basic mechanisms by which VO users and resources

negotiate, establish, manage, and exploit sharing relationships”. The protocol architecture is based on layers, as presented in Figure 2.

The figure shows the relation between the protocol layers and the components of the simulation model. The protocol architecture is structured on specific layers and follows the so-called “hourglass model”, connecting different Grid enabled applications with the resources needed to run them. At the lowest level of the hierarchy is the Fabric, consisting of resources (desktop PCs, batch farms, SMPs, storage media such as disk or tape, network resources, sensors, etc.) being contributed to the Grid, and the local resource management. It includes operating systems, local queuing systems, and libraries of software available at each site. The simulation model incorporates components specific to this layer. The simulation equivalent of the computational resource is the processing unit, having a specified amount of processing power and a maximum amount of memory (the simulation model incorporates even a paging algorithm to incorporate elements specific to the operating systems). The storage element is modeled by database servers and tape servers. The data is organized into databases, consisting of multiple containers. The containers can be used to simulate various data organization technologies. For example, the containers can be seen as files, while the databases can be seen as the directory structure of the system; also the containers can represent the i-nodes of a Unix-like file system, and the databases can model the logical filename. The range of scenarios can even comprise the simulation of real database technologies, where the containers are the equivalent of the database segments. The computational and storage elements are linked together by simulated network entities. A complete host server is simulated by incorporating, in the same local network or farm, of at least one computational unit and a database server. So, grouping the elements in order to achieve higher-level architectures is also possible. In the same time, the regional center, consisting of multiple computational resources, is the simulation model equivalent of a cluster system. The regional center also incorporates a local job scheduler, allowing the simulation of local queuing systems. The scheduler also follows the object-oriented architecture of the simulation model, allowing to easily incorporate various user-defined scheduling algorithms.

The Connectivity layer defines the base communication and authentication protocols required for Grid-specific network transactions. The simulation model incorporates networking stacks (IP, TCP, and

UDP) that facilitate the data exchange among the

various components. The authentication protocols

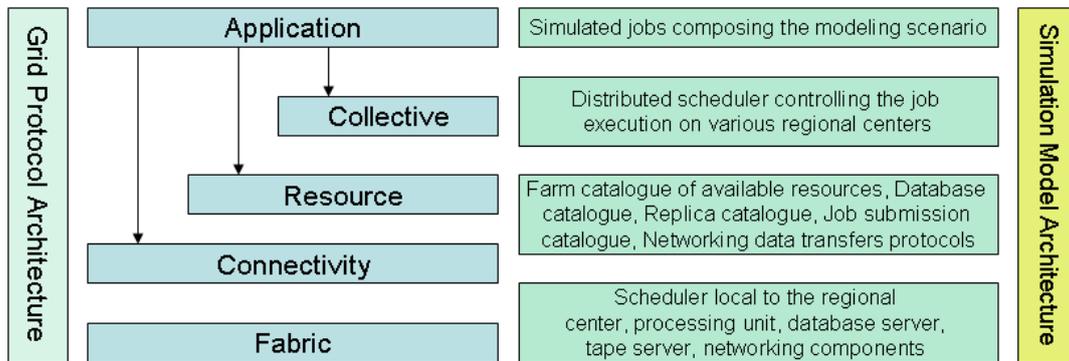


Figure 2. Grid layered architecture and its influence of the components comprising the simulation model.

are not yet part of the simulation model. However, future extensions of the simulation model will include various security models.

The Resource layer defines protocols (and APIs and SDKs) for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. The simulation model incorporates components that are specific to this layer of the architecture. The simulated regional centre contains a farm catalogue of available resources that is used by the meta-scheduler components. A metadata catalogue is responsible with the management of the data, while the database catalogue is responsible with the data location functionality. The status of the submitted jobs is centrally monitored for failures.

The Collective layer contains protocols and services that are not associated with specific resources, but with the interactions across collections of resources. The Collective components can implement a wide variety of sharing behaviors without placing new requirements on the resources being shared. At this layer the simulation model provides a meta-scheduling functionality, allowing the execution of simulated jobs in a distributed manner. The meta-scheduler can incorporate a wide-range of user-defined scheduling algorithms.

The last layer, Application, contains the user applications that operate within a VO environment; the applications may themselves define protocols, services, and/or APIs and have a high degree of complexity. At this layer, the user can describe various jobs that model the behavior of the application being tested. The jobs can be data-intensive, computational-intensive, or a combination of both. In this way a wide range of application can be simulated. The user can easily

extend the framework to incorporate new applications in the simulation model.

The protocol-based architectural view of Grids was later augmented by the authors with a service-based view. In this view, the Grid is considered an "extensible set of services that respond to protocol messages" [6]. According to this view the Grid services may be aggregated to meet the requirements of VO. In the simulation model the definition of the

Table 1. The influence of the characteristics of the Grid systems on the simulation model.

Grid characteristic	Influence on the simulation model
Large scale	Use of advanced internal structure allows the modelling of experiments with many incorporated resources.
Geographical distribution	The regional centre architecture of the simulation model.
Heterogeneity	Use of various models for hardware components; software architectures captured in different probability distributions.
Resource sharing	Represented in the network model.
Multiple administration	Inclusion of a distributed scheduler.
Resource coordination	Resource coordination mechanisms.
Dependable access	Implementation of DAG scheduling algorithms.
Consistent access	Use of standard methods to access the resources.
Pervasive access	The scheduling framework detecting faults and taking

	appropriate actions.
--	----------------------

jobs follows specially designed rules. The discovery of the resources is simulated by the resource catalogue, and the components are accessed using well-defined networking service points. The simulation model is general enough to also provide elements specific to the service-oriented Grid architecture. For example, a generic Grid service functionality is described in [7]. In their described example a client accesses a file transfer service to perform actions such as transfer a file from one storage service to another. A simulation scenario that uses the services of a file transfer service, in the form of a data transfer agent, was successfully executed using the modeling framework (see [4]), thus demonstrating the extensibility of the simulation model to make use of various Grid services paradigms.

The Grids are complex systems that present specific characteristics. According to [8] the characteristics of a Grid system can be summarized into 10 main features. A summary of the characteristics and of their influence on the design of the simulation model is presented in table 1.

As presented, the Grid architectures and characteristics are well mapped on the simulation model being proposed. The simulation model allows the realistic simulation of a wide-range of Grid system technologies, with respect to their specific components and characteristics.

4. Simulation experiments

The generic simulation model allowed the testing of various scheduling algorithms, data transport algorithms and infrastructures, data transfer protocols, replication algorithms, all with interesting results that were used in real-world.

A number of data replications experiments were conducted in [1]. The simulation experiments tested a number of replica strategies in the context of the LHC experiments at CERN. In our tests we were interested in the way the data availability influences the performances. We did a number of tests in which we adjusted the amount of replicated data contained in the satellite regional centers and the bandwidth capability of the link leading to the central data storage unit. The obtained results showed that the performance improves when the data is located closer to the jobs, being greatly influenced by the network characteristics. The obtained results showed that the amount of replicated data and the replication algorithm being used have a great impact on the overall performances of the processing data applications. This is particularly

important for the future LHC experiments, which will produce more than 1 PB of data per experiment and year, data that needs to be then processed.

A series of scheduling simulation experiments were presented in [1], [2] and [3]. In [3] we tested the behavior of a simple distributed scheduler. The experiment allowed the analyzing of the efficiency of the scheduling algorithm. In the same time the simulation proved useful for determining the optimal values for the network bandwidth or for the number of CPUs per regional centre. In [2] the simulation model was used to conduct a series of simulation experiments to compare a number of different scheduling algorithms.

Probably the most extensive simulation scenario is the one described in [4]. The experiment tested the behavior of the tier architecture envisioned by the two largest LHC experiments, CMS and ATLAS. The simulation study described several major activities, concentrating on the data transfer on WAN between the T0 at CERN and a number of several T1 regional centers. The experiment simulated a number of

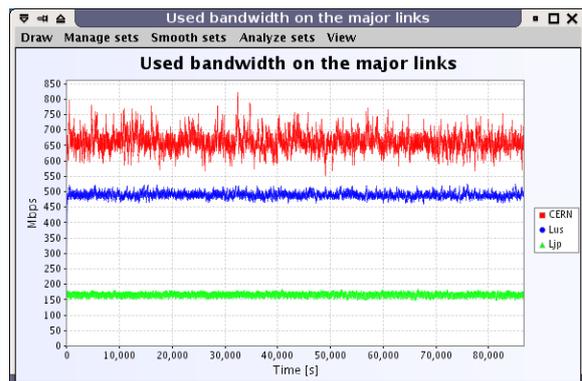


Figure 3. The results obtained in the LHC simulation experiment, using the data transfer agent.

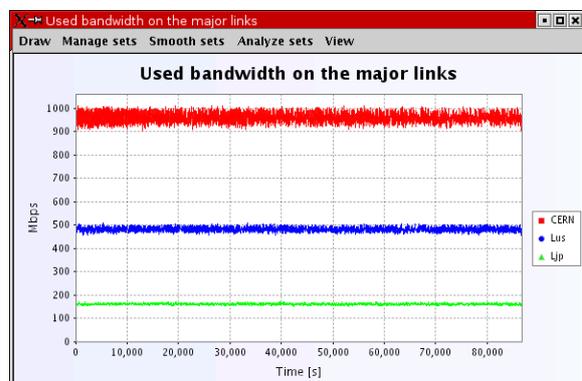


Figure 4. The results obtained in the LHC simulation experiment, without using the data transfer agent.

physics data production specific activities (the RAW data production, Production and DST distribution, the re-production and the new DST distribution and the detector analysis activity). We simulated the described activities alone and then combined.

The obtained results indicated the role of using a data replication agent for the intelligent transferring of the produced data, as presented in Figure 3 and in Figure 4. The obtained results also showed that the existing capacity of 2.5 Gbps was not sufficient and, in fact, not far afterwards the link was upgraded to a current 30 Gbps, based on our recommendations.

5. Related work

SimGrid is a simulation toolkit that provides core functionalities for the evaluation of scheduling algorithms in distributed applications in a heterogeneous, computational Grid environment. It aims at providing the right model and level of abstraction for studying Grid-based scheduling algorithms and generates correct and accurate simulation results. *GridSim* is a grid simulation toolkit developed to investigate effective resource allocation techniques based on computational economy. *OptorSim* is a Data Grid simulator project designed specifically for testing various optimization technologies to access data in Grid environments. *OptorSim* adopts a Grid structure based on a simplification of the architecture proposed by the EU DataGrid project. *ChicagoSim* is a simulator designed to investigate scheduling strategies in conjunction with data location. It is designed to investigate scheduling strategies in conjunction with data location.

Because of the complexity of the Grid systems, involving many resources and many jobs being concurrently executed in heterogeneous mediums, there are not many simulation tools to address the general problem of Grid computing. The simulation instruments tend to narrow the range of simulation scenarios to specific subjects, such as scheduling or data replication. There is little room for modeling experiments designed to test, for example, a newly proposed communication or data-transfer protocol designed for Grid systems, using the described simulation instruments.

6. Conclusions

The MONARC 2 model developed a CPU and code-efficient approach to the problem of simulation of distributed computing systems. The model allows the realistic simulation of a wide-range of distributed systems technologies, with respect to their specific components and characteristics. We presented the design characteristics of the simulation model being proposed by MONARC 2, as it incorporates the necessary components to model various modern-day distributed systems technologies, providing the mechanisms to describe concurrent network traffic and to evaluate different strategies in data replication or in the job scheduling procedures.

The maturity of the simulation model is demonstrated by the number of simulation scenarios that were successfully conducted. The simulation framework allowed the modeling of various Grid systems, with many tasks competing for resources, which tested a wide-range of various technologies, from scheduling and data replication to data transfers and distributed processing.

7. References

- [1] I. C. Legrand, H. Newman, C. M. Dobre, C. Stratan, "MONARC Simulation Framework", International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Tsukuba, Japan, 2003.
- [2] F. Pop, C. M. Dobre, G. Godza, V. Cristea, "A Simulation Model for Grid Scheduling Analysis and Optimization", Parelec, 2006.
- [3] C. M. Dobre, C. Stratan, "MONARC Simulation Framework", RoEduNet International Conference, Timisoara, Romania, 2004.
- [4] I. C. Legrand, C. M. Dobre, R. Voicu, C. Stratan, C. Cirstoiu, L. Musat, "A Simulation Study for T0/T1 Data Replication and Production Activities", the 15th International Conference on Control Systems and Computer Science, Bucharest, Romania, 2005.
- [5] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", Int. J. Supercomp. App., 2001.
- [6] I. Foster, C. Kesselmen, J. Nick, and S. Tuecke, "The Physiology of the Grid - An Open Grid Services Architecture for Distributed Systems Integration", IEEE Computer 35, Open Grid Service Infrastructure WG, Global Grid Forum. 2002.
- [7] I. Foster, C. Kesselman and S. Tuecker, "The Grid 2: Blueprint for a new Computing Infrastructure", Morgan Kaufmann, 2004.

[8] M. Bote-Lorenzo, Y. Dimitriadis, and E. Gomez-Sanchez, "Grid characteristics and uses: a grid definition", Technical Report CICYT, Univ. of Valladolid, Spain, 2002.