Realistic simulation of large scale distributed systems using monitoring

Ciprian Dobre*, Corina Stratan*, Valentin Cristea*
** Faculty of Automatics and Computer Science, University Politehnica of Bucharest, Romania*
E-mails: {cipsm, corina, valentin}@cs.pub.ro

Abstract

In this paper we present an extension to MONARC, a generic simulator for large scale distributed systems, which allows realistic evaluation of various actual distributed system technologies based on real-world monitored data supplied by MonALISA. The field of modelling and simulation was long-time seen as a viable alternative to develop new algorithms and technologies and to enable the development of large-scale distributed systems, where analytical validations are prohibited by the nature of the encountered problems. 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 the design of realistic scenarios. This paper presents a novel approach to combining two distributed systems domains, monitoring and simulation, highlighting a realistic solution to the problem of accurately evaluating various distributed systems technologies using simulation. We also present a simulation study which demonstrates the interoperability between the simulation framework and the monitoring instrument, demonstrating important properties of the US LHCNet research network in the context of the LHC experiments in CERN.

Keywords: *Modeling and simulation, monitoring, large scale distributed systems, performance analysis*

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, 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 [1]. The simulation model being proposed by MONARC provides the mechanisms to describe concurrent network traffic and to evaluate different strategies in data replication or in the job scheduling procedures.

One of the biggest challenges in developing a realistic simulation is represented by the complexity of distributed systems, in which a great number of users perform simultaneously their computation tasks or data transfers. Since estimating the impact of all these simultaneous operations on the system is difficult or even impossible, a more convenient approach is to incorporate in the simulation a set of performance data obtained by monitoring real systems. In this paper we present the solutions adopted in MONARC to design realistic simulation experiments based on real-world data collected by one of the most important monitoring frameworks for large scale distributed systems, MonALISA [2]. We emphasize on the extensions to the MONARC architecture that allows the design of modeling experiments based on real-world conditions.

The paper is organized as follows. Section 2 introduces the general simulation model proposed by MONARC. Section 3 describes the MonALISA monitoring framework. In section 4 we present the design considerations that allow the use of monitoring data as input for modeling experiments. In section 5 we present a simulation experiment designed to evaluate the interoperability between the simulation framework and the monitoring instrument, demonstrating important properties in case of the US LHCNet research network in the context of the LHC experiments at CERN. In section 6 we present related

work in this field and in section 7 we give the conclusions.

2. The simulation framework

MONARC 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, which contains a farm of processing nodes (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. Any regional center can dynamically instantiate a set of users or activity objects, which are used to generate data processing jobs based on different simulation scenarios. Inside a regional center different job scheduling policies may be used to distribute jobs to corresponding processing nodes.

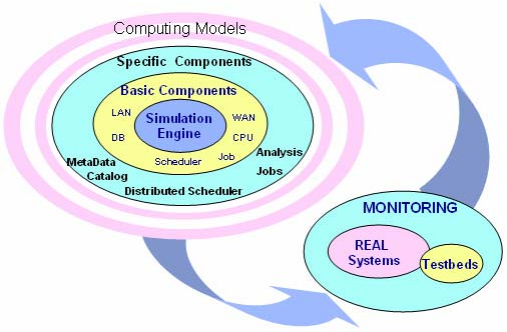

Figure 1. MONARC 2 layers.

One of the strenghts of MONARC 2 is that it can be easily extended, even by users, and this is made possible by its layered structure. The first two layers contain the core of the simulator (which we call "simulation engine") and models for the basic components of a distributed system (CPU units, jobs, databases, networks, job schedulers etc.); these are the fixed parts on top of which some particular components (specific for the simulated systems) can be built. The particular components can be different types of jobs, job schedulers with specific scheduling algorithms or database servers that support data replication. The diagram presented in Figure 1 represents the MONARC layers and the way they interract with a monitoring system.

The maturity of the simulation model was demonstrated in previous work. For example, a number of data replications experiments were conducted in [3], presenting important results 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 [3], [4] and [5]. In [5] we successfully evaluated a simple distributed scheduler algorithm, proving the optimal values for the network bandwidth or for the number of CPUs per regional centre. In [4] 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 [6]. 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 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. 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.

3. MonALISA

MonALISA (MONitoring Agents in a Large Integrated Services Architecture) is a global distributed monitoring system developed as a result of a

collaboration between Politechnica University of Bucharest, the European Center for Nuclear Research – CERN, in Geneve, Switzerland, and the California Institute of Technology [2].

The MonALISA system is designed as an ensemble of autonomous multi-threaded, self-describing agent-based subsystems which are registered as dynamic services, and are able to collaborate and cooperate in performing a wide range of information gathering and processing tasks. These agents can analyze and process the information, in a distributed way, to provide optimization decisions in large scale distributed applications. An agent-based architecture provides the ability to invest the system with increasing degrees of intelligence, to reduce complexity and make global systems manageable in real time. The scalability of the system derives from the use of a multithreaded execution engine to host a variety of loosely coupled self-describing dynamic services or agents and the ability of each service to register itself and then to be discovered and used by any other services, or clients that require such information. The system is designed to easily integrate existing monitoring tools and procedures and to provide this information in a dynamic, customized, self describing way to any other services or clients.

The MonALISA architecture, presented in Figure 2, is based on four layers of global services. The network of JINI - Lookup Discovery Services (LUS) provides dynamic registration and discovery for all other services and agents. Each MonALISA service can execute many monitoring tasks through the use of a multithreaded execution engine and to host a variety of loosely coupled agents that analyze the collected information in real time. The collected information can be stored locally in databases. The layer of Proxy services, shown in the figure, provides an intelligent multiplexing of the information requested by the clients or other services and is used for reliable communication between agents. It can also be used as an Access Control Enforcement layer.

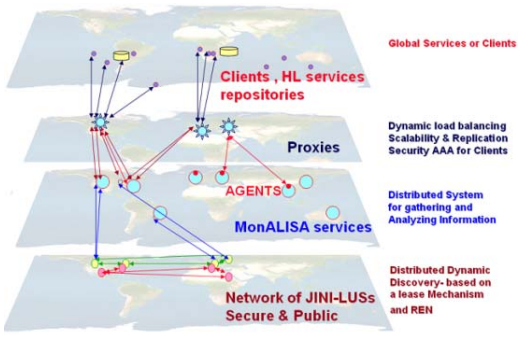

Figure 2. The MonALISA Architecture.

The MonALISA framework is currently being used to monitor many grid sites in the U.S., Europe and Asia. MonALISA provides complete information, in near real-time, about resource utilization and job execution. The framework also provides information about network traffic in major networks and connectivity between different grid sites. Currently MonALISA is running in more than 250 computing sites around the world. Overall, the system is collecting approximately 250.000 parameters in near real-time with a rate of 25.000 updated parameters per second, collecting data related to approximately 12.000 computers, 100 WAN links and thousands of jobs running concurrently in Grid systems. Some major Grid communities rely on MonALISA, among which are OSG, CMS, ALICE, D0, STAR, VRVS, LCG Russia, SE Europe GRID, APAC Grid, UNAM Grid, ABILENE, ULTRALIGHT, GLORIAD, LHC Net, and RoEduNet.

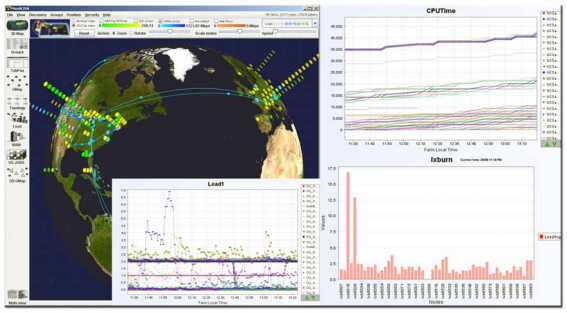

Figure 3. Monitoring Grid sites facilities, traffic and Jobs.

MonALISA represents an important instrument that was extensively used in the design and validation of MONARC. In order to obtain a robust simulation model, every component of the simulation framework was designed based on several real-world observations provided by MonALISA. Starting from the types of components that are specific to various Grid systems and ranging to the functional characteristics and types of input parameters of the various components incorporated in the simulation model, all this information was obtained by analyzing various monitoring data provided by MonALISA.

4. Development of realistic simulation experiments in MONARC based on real-world monitoring data

In order to design realistic simulation experiments we extended the default simulation model in MONARC to allow for the incorporation of real-world monitored input trace data in various simulation

scenarios. This aspect is important when designing simulation scenarios based on hypothetical conditions starting from real-world running conditions of various large-scale distributed systems.

In order to obtain the input trace data from MonALISA we added a novel functionality to its repository. This functionality allows the user to select the data of interest for a simulation experiment using the web page of a repository and then to download it. The format of the input data files specify two meaningful values: the time a particular value was registered by the monitoring services and the effective value.

With the provided data, we then extended the default job functionality in MONARC in order to make full use of the real-world running conditions of Grid systems. Within the simulation model of MONARC, the users can define different types of jobs to model common types of actions that occur in any distributed systems: processing, data transfer (both pull and push models are supported), database handling, etc. The model is not limited to a specific defined activity, the user having the possibility to easily incorporate new advanced job behaviors, as specified by the simulation scenario being executed.

Due to the openness and modularity characteristics of the simulator we were able to extend the default job model with two new sets of jobs, for modeling realistic data transfers and for simulating data processing.

The transfer data job uses the input trace data to simulate realistic networking transfers occurring in a simulation experiment. This job waits until the simulation clock reaches the time a particular amount of data was collected by the monitoring framework as being the result of the amount of data that was transferred in real-world on a particular data link. When the time is reached, the job proceeds to model its characteristic behavior - the transfer of the specified amount of data. Because of the way MONARC considers the modeling of dependencies among simulated jobs, this job behavior is also useful in modeling background traffic, such as in case of the experiment presented in the next section.

The second extended job considers the monitored data regarding the real-world CPU utilization. This job models the loading of a simulated processor with a particular value, exactly as encountered in real-world monitored situation.

In the next section we present a simulation experiment that demonstrates the capability of MONARC to incorporate input trace data obtained from real-world using monitoring.

5. US LHCNet – a simulation study involving monitoring

In this section we present a simulation study which demonstrates the interoperability between the simulation framework and the monitoring instrument. The simulation experiment is based on input trace data that is represented by real-world monitoring information collected by MonALISA. The scenario was designed to realistically test the behavior of the system being monitored under the influence of several hypothetical added conditions. It is a simulation experiment in which we tried to answer questions such as "what if?" on a real testbed.

This simulation experiment tested the capability of the US LHCNet research network to sustain the data traffic that will be generated by LHC physics analysis. The experiment was based on the real-data monitored information provided by MonALISA, together with the data traffic patterns described in the computing models of ATLAS and CMS. The objectives of the scenario were:

1) Test the capacity of the US LHCNet network to sustain the traffic that will be generated by the LHC experiments under the traffic values that are currently being observed on the provided links.

2) Demonstrate the use of monitoring data as trace input for the simulation scenario depicted in the experiment.

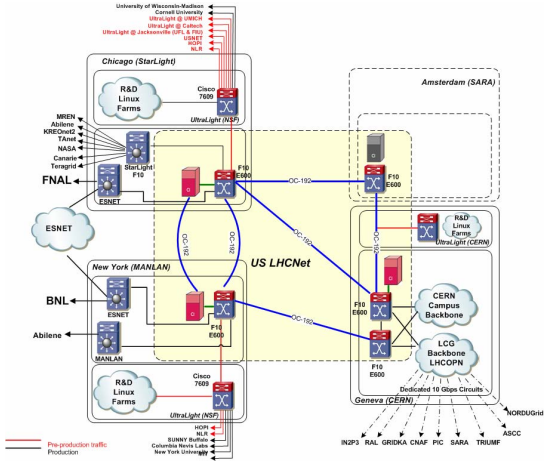

Figure 4. The US LHCNet current network topology.

The US LHCNet transatlantic network is part of the ensemble of networks used by the HEP community, and an essential resource for US participation in the LHC. The network has been architected to ensure efficient and reliable use of the 10 Gbps bandwidth of each link, up to relatively high occupancy levels, to cover a wide variety of network tasks, including: large

file transfers, grid applications, data analysis sessions involving client-server software as well as simple remote login, network and grid R&D-related traffic, videoconferencing, and general Internet connectivity. The current topology of the network is presented in Figure 4.

The research network relies on MonALISA to monitor and manage all network links. For that purpose MonALISA provides monitoring information such as the traffic throughput or the amount of data being transferred, as well as the current topology of the network links in use and their health status. The experiment used the provided information in order to test how the network links would support an addition of data traffic such as the one depicted in the LHC computing models. The simulation experiment tried to find an answer to the posed problem of what would happen if the LHC experiment were to start now. We used the monitored data from the last six month for this experiment. A global view of the links currently monitored by MonALISA is presented in Figure 5.

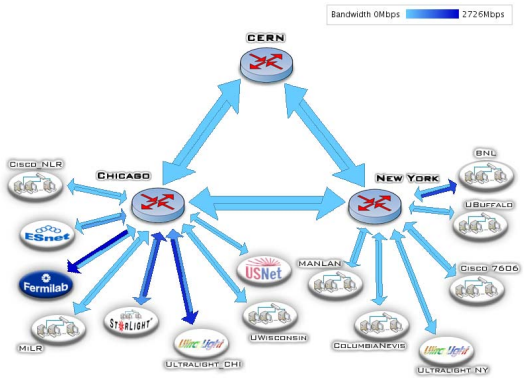

Figure 5. The monitored network links.

Figures 6 and 7 present the monitored data used as input trace in the simulation experiment, the real-world data gathered by the MonALISA monitoring framework.

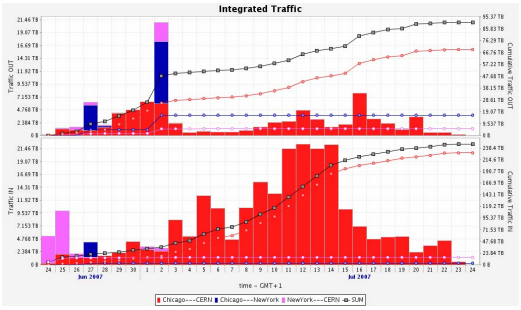

Figure 6. The integrated traffic as collected by the MonALISA monitoring framework.

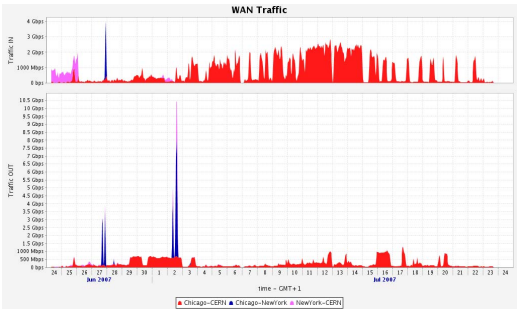

Figure 7. The monitored data of the major links that was used in the simulation experiment.

The monitored trace input data was used in conjunction with the LHC traffic values proposed in the LHC computing models. Of particularly importance are the two Tier1 centers in US, namely FNAL and BNL. The traffic values that were used in the simulation experiments, based on the computing models, are presented in Table 1 (the values are in Mbps).

Table 1. The data traffic needs as described in the LHC computing models.

Activity	T0 -> T1	T1->T2	T2->T1	T1<-> T1
Centre	**Data Taking (Mbps)**	**User needs (Mbps)**	**Simulation (Mbps)**	**Scheduled Reprocessing (Mbps)**
FNAL	110	415.0	52.6	417.0
BNL	186.5	137.7	24.8	358.0

For this experiment we used the network topology presented in Figure 8. This topology represents a simplification of the real-world conditions, being based on the layout presented in Figure 4.

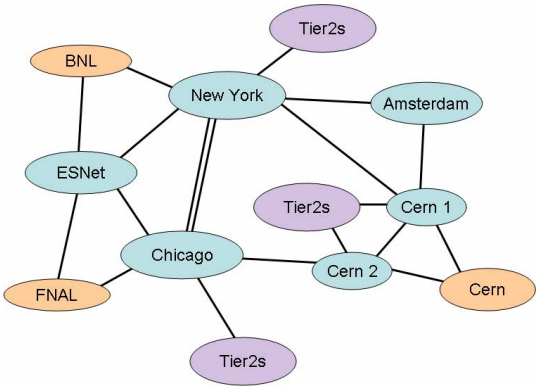

Figure 8. The network topology used in the simulation experiment.

The configuration contains the following nodes. The Cern center represents the Tier0 center where the data is first produced. FNAL and BNL are the two Tier1 centers served with priority by the US LHCNet network. The Tier2s nodes are a compact representation of the sites that will process the data from Tier1s. There are two such centers in US, being served with priority from the two Tier1s.

The simulation experiment considered the Cloud Computing Model being introduced by the ALICE experiment at CERN, according to which a Tier2 center can use the data provided by any Tier1 center. In concordance, we considered in the experiment an ensemble of Tier2 centers in Europe that can transfer data from FNAL and BNL (up to 10% of the data being transferred from T1 to T2 and vice versa in the experiment goes to Europe). We considered this to be a more realistic representation of the actual architecture that will be in use. The rest of the nodes are the ones used in the real topology presented in Figure 4. The links connecting the nodes, just like in the real case, are supposed to be of 10 Gbps. There are two actual links connecting New York and Chicago in the experiment.

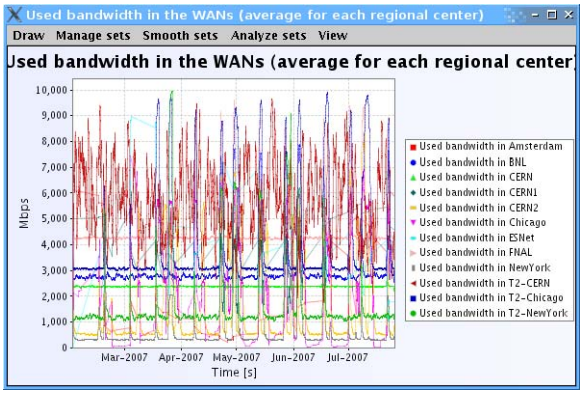

Figure 9. The used bandwidth in WANs.

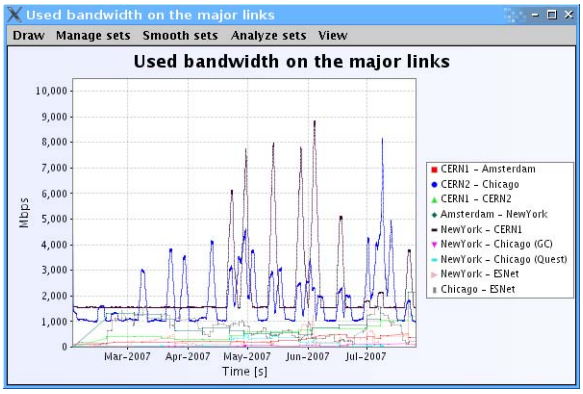

Figure 10. The bandwidth used on the major links.

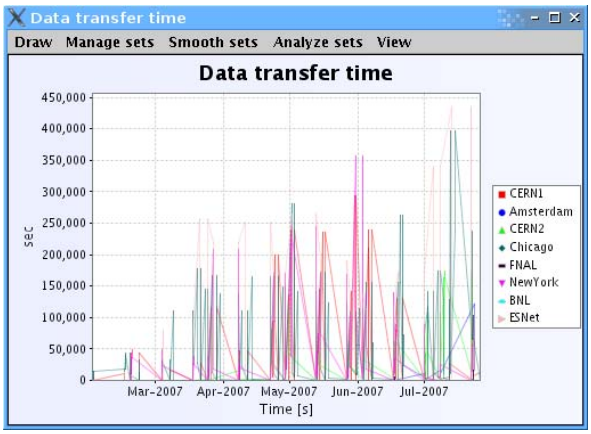

Figure 11. The time needed to transfer the data.

With this topology we first added the trace input data obtained from the monitoring repository. The next input to the simulation was represented by the traffic values proposed for the LHC experiments. The obtained results are presented in the next figures.

Analyzing the bandwidth consumed by the simulated traffic in the major WANs (Figure 9) and on the network links (Figure 10) we conclude that the US LHCNet topology with the bandwidth values it provides are sufficient for handling the assumed amounts of data. The same validation can be observed when analyzing the values of the time needing to transfer the data through the network links. The obtained results for this parameter are represented in Figure 11.

The obtained results consistently proved that the current network topology could sustain the data traffic generated by the LHC experiments. More importantly, the simulation scenario demonstrated the maturity of the simulation model provided by MONARC to handle real-data as trace input for an experiment.

6. Related work

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. However, 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 and from those only few consider the use of real-world data to design realistic simulation experiments.

SimGrid [8] 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* [9] is a grid simulation toolkit developed to investigate effective resource allocation techniques based on computational economy. *OptorSim* [10] 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* [11] 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.

These simulation instruments are either too focused on particular aspects of grids and distributed systems (such as data replication, scheduling), or not extensive enough, allowing the modeling of a limited number of possible scenarios. They all tend to narrow the range of simulation scenarios to specific subjects, without considering all the characteristics. There is little room for modeling experiments designed to test, for example, a newly proposed job scheduling procedure for tasks involving processing activities, as well as communications among them, as would be encountered in real-world situations [7].

MONARC is more generic than the others. It allows the realistic simulation of a wide-range of distributed system technologies with respect to their specific components and characteristics. As presented in [1], the simulation model of MONARC 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. MONARC can be used to simulate a large number of possible scenarios, ranging from high-granularity network protocols to large-scale distributed systems comprising of many type of resources, to the modeling and simulation of the behavior of various applications. Combined with the power of using a state-of-the-art monitoring framework for designing realistic experiments running under real-world conditions, MONARC proves its value as an instrument to be used for correctly evaluating a wide-range of Grid technologies.

7. Conclusions and Future Work

As a conclusion we demonstrated the capability of using monitoring instruments in the design of realistic experiments using MONARC. MonALISA, a monitoring framework designed for large scale distributed systems, was successfully used in designing realistic modeling experiment such as the one presented in this document. One interesting capability of MONARC that was presented in this paper is that it is capable of incorporating real-world monitored input trace data in various simulation scenarios. This aspect is important when designing simulation scenarios based on hypothetical conditions starting from real-world running conditions. For example, one could easily test the behavior of a real distributed system under the influence of a new scheduling technology. The relationship between the monitoring and simulation technologies clearly proved to be in the best advantage of both of them, as each one greatly benefited from the other one.

In the future we plan to extend the simulation model to also incorporate automatic configuration of parameters for the simulated components. For example, the processor utilization could be adjusted at runtime based on real-world monitored value from various existing large scale distributed systems. Such an extension would made possible the evaluation of various technologies under various real-world distributed architectures without implying the need to actually deploying such technologies in real-world.

7. References

[1] C. M. Dobre, V. Cristea, "A Simulation Model for Large Scale Distributed Systems", *Proc. of the 4th International Conference on Innovations in Information Technology*, Dubai, United Arab Emirates, November 2007.

[2] I. C. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. M. Dobre, "MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications", *Proc. of Computing in High Energy and Nuclear Physics CHEP'04* , Interlaken, Switzerland, 2004.

[3] 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.

[4] F. Pop, C. M. Dobre, G. Godza, V. Cristea, "A Simulation Model for Grid Scheduling Analysis and Optimization", *Parelec* , 2006.

[5] C. M. Dobre, C. Stratan, "MONARC Simulation Framework", *RoEduNet International Conference*, Timisoara, Romania, 2004.

[6] 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.

[7] B. Quetier, F. Capello, "A survey of Grid research tools: simulators, emulators and real life platforms", *IMACS Survey*, 2005.

[8] H. Casanova, A. Legrand, M. Quinson, "SimGrid: a Generic Framework for Large-Scale Distributed Experimentations", *Proc. of the 10th IEEE International Conference on Computer Modelling and Simulation (UKSIM/EUROSIM'08)*, 2008.

[9] R. Buyya, M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", *The Journal of Concurrency and Computation: Practice and Experience (CCPE), Volume 14, Issue 13-15, Wiley Press*, 2002.

[10] W. Venters, *et al*, "Studying the usability of Grids, ethongraphic research of the UK particle physics community", *UK e-Science All Hands Conference, Nottingham*, 2007.

[11] K. Ranganathan, I. Foster, "Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications", *Int. Symposium of High Performance Distributed Computing, Edinburgh, Scotland*, 2002.