

# State of Simulators in Computational Grid System

Surendra Kumar Patel  
Department of Information  
Technology and Computer  
Application

Dr. C.V.Raman University, Kota  
Bilaspur, Chhattisgarh, India

Anil Kumar Sharma  
Department of Information  
Technology. And Computer  
Application

Dr. C.V.Raman University, Kota  
Bilaspur, Chhattisgarh, India

Gupteshwar Gupta  
Department of Mathematics  
And Information Technology,  
Govt. N.P.G. College of  
Science Raipur, Chhattisgarh,  
India

## ABSTRACT

The development of computational grids and the associated middleware has been actively pursued in recent years to deal with the emergence of greedy applications of large computing tasks and amounts of data. Simulation of the Grid environment is important to evaluate the impact of potential data handling and application before being deployed on the Grid, in this evolving research field. There is always an overriding concern when using simulation that the results may not reflect accurate behavior. It is therefore essential to know the strengths and weaknesses of these simulators. This paper provides a comprehensive survey and comparisons of various popular Grid Computing simulators with a view to help researchers choose the best simulator available for a particular application environment.

## Keywords

Simulation tools, Parallel Computing, Distributed Computing, and Grid Computing.

## 1. INTRODUCTION

Grid computing technologies are increasingly being used to aggregate computing resources that are geographically distributed across different locations. Commercial networks are being used to connect these resources, and thus serve as a fundamental component of grid computing. Since these grid resources are connected over a shared infrastructure, it is essential to consider their effect during simulation [1].

Simulation is the limitation of the operation of a real-world process or system over time.[2] The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviors/functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time [3].

Nowadays a lot of grid infrastructures exist and are extensively used for scientific as well as industrial applications. Due to their complexity, grid applications need to be tested before running on an appropriate infrastructure. However, access to such infrastructures is not feasible or available to everyone. Therefore, various simulation tools were developed to ease the cost and complexity of testing grid applications.

Version 2 [6] adds a new software layer to provide high-level abstractions. SimGrid Version 3 [6] has three main layers-Base, Simulation and Programming.

Grid simulation tools can either be simulators or emulators. Simulators model the logic of specific systems and can execute faster than the real execution would be. The results of the simulation are saved to support post-simulation analysis of the system behavior. An emulator, on the other hand, acts like an actual system. Emulators are used to mimic the execution of grid applications on a real infrastructure without actually deploying the applications on a real testbed [4].

## 2. RELATED WORK

Grid simulators are in use for several years now. Some important simulators are regularly used. In this section we provide a brief description of simulators.

### 2.1 SimGrid

The Simgrid toolkit [5], developed in the University of California at San Diego (UCSD), is a C language based toolkit for the simulation of application scheduling. It supports modeling of resources that are time-shared and the load can be injected as constants or from real traces.

#### 2.1.1 Features

- Fast and accurate simulation capabilities (SURF).
- Ability to run the same code in full or partial simulation mode or in real-world mode (GRAS, SMPI).
- An API for rapid application prototyping to test and evaluate distributed algorithms (MSG).
- An API for application development to obtain fast, robust and portable application code (GRAS).
- An API for MPI application simulation to study the effect of platform heterogeneity (SMPI).
- Task dependencies.
- Performance prediction error simulation

#### 2.1.2 SimGrid Version

The SimGrid version 1 [6] is a discrete event simulation toolkit. It provided a set of core abstractions and functionalities that can be used to easily build simulators for specific application domains and/or computing environment topologies. However, this first version lacked a number of abstractions (e.g. routing, scheduling agents). SimGrid

#### 2.1.3 Components of the SimGrid

The components of the SimGrid [7, 8] toolkit are given in Figure 1 and are described below:

XBT - The eXtended Bundle of Tools provides portable base functionalities like logging, exception and configuration support.

SURF - It is a very low level simulator and serves as a core layer for the higher programming layers. SURF [7] provides several types of resources like workstation, network, CPU and time resources

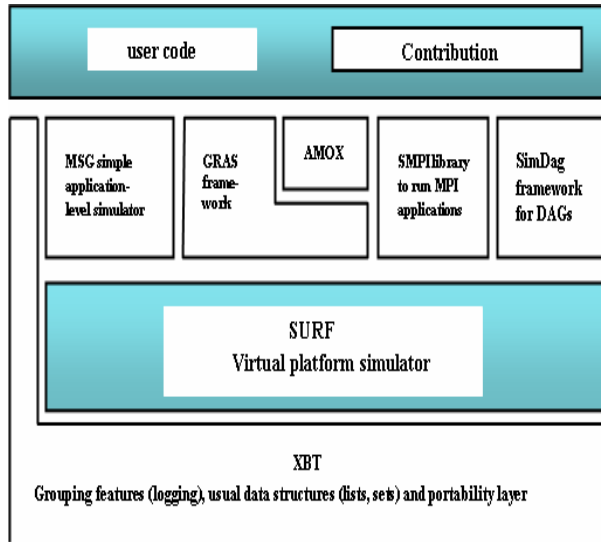


Figure 1: SimGrid Components

GRAS - The Grid reality and simulation API is a programming environment to develop real distributed application for large-scale heterogeneous platforms.

AMOK - The Advanced Metacomputing Overlay Kit is a new SimGrid module. It is a toolbox for applications built on top of GRAS, providing a set of features needed by most GRAS applications.

SMPI - This programming environment will allow you to simulate any MPI application without having to modify them for the simulator. SimDag - It expresses your application as a collection of interdependent parallel & distributes tasks. SimDag [7] is a new implementation of DAGs handling and it is built on top of SURF

#### 2.1.4 Resource Model

Hosts and links are modeled in SimGrid as simple resources. A host is a computational resource which is defined by computational speed and CPU availability. A link is a network resource which is defined by latency and bandwidth [4].

SimGrid models the task execution time according to following formula:

$$T_{\text{Task}} = \frac{\text{Computational cost}}{\text{Computational Speed}}$$

Computational Cost represents the task complexity and Computational Speed defines the computational speed of the CPU. The data transfer time is modeled in the following way:

$$T_{\text{Data}} = \text{Latency} + \frac{\text{DataSize}}{\text{Bandwidth}}$$

The values of Latency and Bandwidth for each link can either be constant or varying values from traces.

## 2.2 GridSim

GridSim is a toolkit for modeling resources and simulating network connectivity with different configurations. It is implemented as a Java API published under the GNU GPL and is part of the GridBus Project [9].

The GridSim toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers. It can be used to simulate application schedulers for single or multiple administrative domains distributed computing systems such as clusters and grids. Application schedulers in grid environment, called resource brokers, perform resource discovery, selection, and aggregation of a diverse set of distributed resources for an individual user [10].

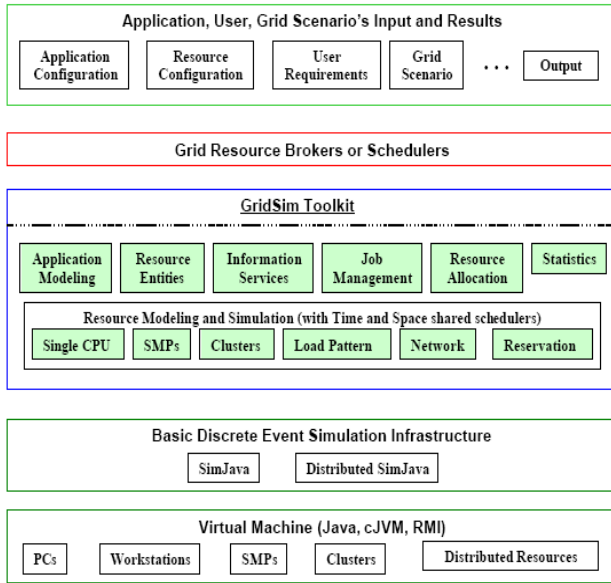
### 2.2.1 Features

- Supports heterogeneous resources
- Space- or time-shared resource scheduling
- Resource speed in MIPS
- Resources support time zones
- Time settings to model non-grid workload.
- Resource reservation supported
- Tasks can be CPU and I/O intensive
- No task limit on resources
- Multiple user can submit tasks
- Network speed between resources can be specified
- Supports static and dynamic schedulers
- Statistics can be recorded
- Cross-platform (Java-based)

### 2.2.2 Design and Architecture

GridSim is built upon a general-purpose event-driven discrete event simulator called SimJava [11, 12]. SimJava provides a set of classes to support the creation of discrete event simulation (DES) models with Java. The library includes three main packages called simjava, simanim and simdiag. The simjava package incorporates an event-driven DES engine to create discrete event simulations. simanim provides an applet template to create visual animations of the simulation. simdiag enables a graphical representation of the results.

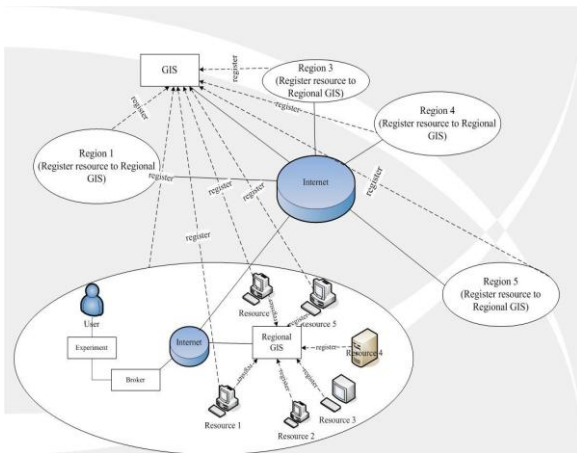
A multi-layer architecture and abstraction for the development of GridSim platform and its applications is shown in Figure 2.



**Figure 2: A modular architecture for GridSim platform and components.**

### 2.2.3 Resource Model

GridSim uses Processing Elements (PEs) to model computational resources. One or more PEs can be put together to model a machine. One or more machines can also be combined to create a grid or Cluster resource. These machines can be scheduled by time- or space-shared algorithms. Workstations are managed by time-shared operating systems using round-robin scheduling with priorities. Cluster resources can be scheduled by space-shared algorithms like first-come-first-served (FIFO) or shortestjob- first (SJF) [4].



**Figure3: GridSim Entities**

## 2.3 OptorSim

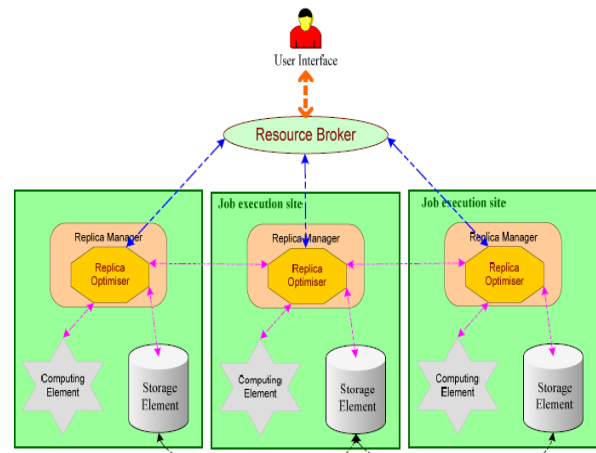
OptorSim [13] was designed to test various replication optimization strategies in a simulated Grid environment before they are deployed in the real Grid, especially simulating data access optimization algorithms. It uses discrete event simulation. This open source simulator has been developed in the framework of the European DataGrid (EDG) as a joint effort of ITC-first, University of Glasgow and CERN [14].

### 2.3.1 System Architecture

OptorSim has a modular architecture that is directly based on the DataGrid project architecture figure:3. OptorSim consists of the following components [4]:

- **Sites** - provides resources for submitted jobs.
- **Resource Broker** - controls the scheduling of jobs to computing resources.
- **Routers or Network Nodes** - without computing or storage resource.

Several replication strategies have been compared with OptorSim, including: no replication, unconditional replication deleting the oldest files if space is required, unconditional replication LRU (Least Recently Used) and a more sophisticated algorithm based on an economic approach [14].



**Figure 4: The OptorSim Architecture**

### 2.3.2 Applications

With Optor, several replication strategies have been compared: no replication, unconditional replication deleting the oldest files if space is required, unconditional replication LRU and a more sophisticated algorithm based on an economic approach.

## 2.4 GangSim

GangSim [15] is a scheduling algorithm simulator for huge systems with hundreds thousands of computers and storage systems. It simulates components like sites and virtual organizations with different usage policies. This discrete event simulator periodically evaluates the state of all simulated components. GangSim derived from Ganglia, a monitoring toolkit, in the simulation the reports from the monitoring are generated by component models.

### 2.4.1 Component of the GangSim

GangSim models comprise the following real grid elements [16]: a job submission infrastructure, a monitoring infrastructure, and a usage policy infrastructure. As shown in Figure 4, the principal components of GridSim are external schedulers (ESs), local schedulers (LSs), data schedulers (DSs), monitoring distribution points (MDPs), site policy enforcement points (SPEPs), and virtual organization policy enforcement points (V-PEPs).

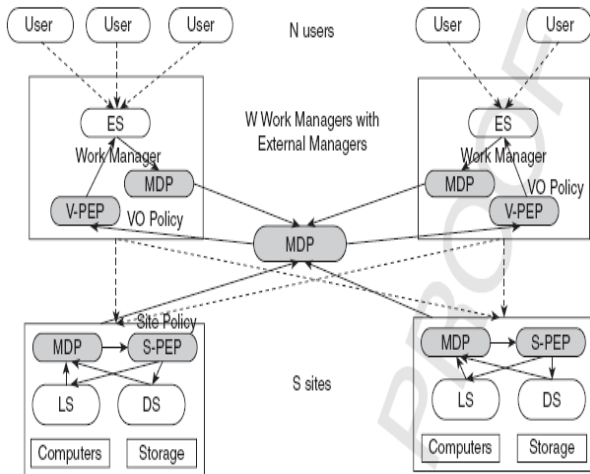


Figure 4: GangSim's components.

### 2.4.2 Applications

GangSim can be used to simulate synchronous and asynchronous workloads. In the former case, all VOs submit their jobs almost at the same time. In the second situation, VOs submit their burst workloads at different moment in time.

## 2.5 GridNet

GridNet [17] is a Data Grid simulation tool for dynamic data replication strategies. It has been developed to evaluate the performance of novel scalable distribution topologies that adapt replica placement to meet the need of a large numbers of users who continuously change their data. GridNet is written in C++ and built on top of the event driven network simulator ns [18]. From ns the basic Grid network specification: nodes, links, and messages have been taken that allow to model different network topologies underlying the simulated environment. The simulator modules are composed of objects that are mapped into ns. Data exchange is defined on the application level and passed down to ns nodes as a stream of packets.

### 2.5.1 System Architecture

GridNet uses the following three node types to define its components: servers, caches, and clients. Server nodes represent main storage sites where grid data are stored (each node may host the whole or just some parts of a data file); cache nodes represent intermediate storage sites (e.g., a regional storage site) to replicate parts of data stored in main storage sites/servers; and client nodes represent sites where data access is requested. Figure 5 shows the simulation architecture of this simulator [19].

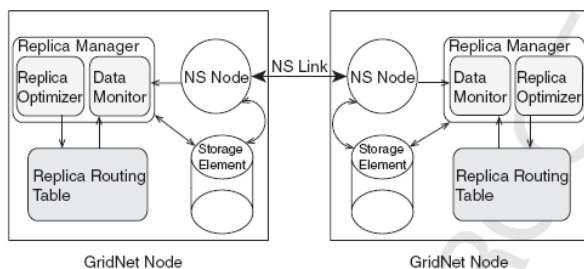


Figure 5 : GridNet's architecture.

## 2.6 Bricks

Bricks, developed by the Tokyo Institute of Technology in Japan, is a helpful tool in evaluating high performance global computing systems like multi-user systems [20]. It provides a virtual high performance grid computing environment for analyzing various scheduling schemes and comparing performance evaluation systems.

### 2.6.1 System Architecture

Bricks allows the simulation of various behaviors such as network topology of client-server, resource scheduling algorithms and processing schemes for network and servers. It is basically an event driven simulator built as a framework of a set of replaceable components [21].

Bricks is implemented in Java and designed using an object-oriented framework. Figure 1 shows the architecture of Bricks which is made up of two main parts: the grid Computing Environment that models the grid computing system and the Scheduling Unit that mimics the scheduling framework for the grid computing system, providing various scheduling algorithms, acts of coordination and components (using Java interfaces).

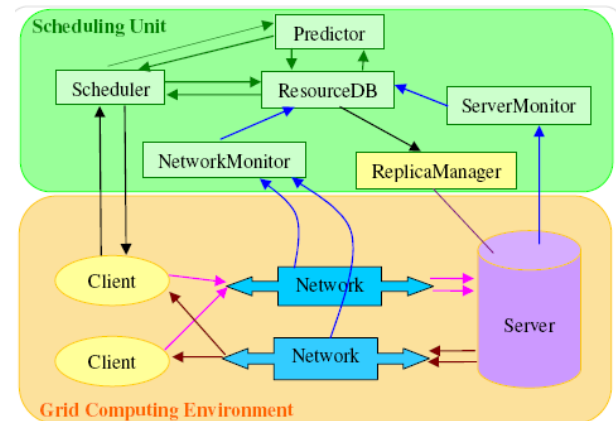


Figure 6: Bricks Architecture

Bricks use Bricks script, a language that enables users to specify network topologies, server architecture, scheduling framework components and communication models.

## 2.7 MicroGrid

The goal of the MicroGrid [22] [23] project is to develop and implement simulation tools that provide a vehicle for the convenient scientific study of grid topologies and application performance issues. The MicroGrid provides a virtual grid infrastructure supporting controlled, repeatable experiments. Computation and applications are emulated; actual code is executed on virtualized resources using some physical CPU. Network is simulated on packet level by discrete events. It has been designed to emulate Globus Grids, all Globus components run on virtual hosts. The emulation runs continuously, however it is possible to manage a virtual time that is passed to the applications through system calls.

### 2.7.1 Components of MicroGrid's

The MicroGrid may conduct experiments by simulating applications on heterogeneous physical resources, as shown in Figure 7 [24].

The MicroGrid simulation comprises of separate local resource simulations which provide a virtual grid resource environment and the network simulator which captures the interaction between these local resources. Each component in the architecture addresses one of key challenges in constructing a highly accurate virtual grid [25]. These challenges are summarized as:

- **Virtualization:** The virtual grid resources (such as networks, host names) are observed only by the application and are utilized independent of the physical resources.
- **Global Coordination:** provides a consistent grid simulation environment with different scenarios of varying virtual resources based on heterogeneous physical resources and coordinates the simulation speed of varying virtual resources.
- **Resource Simulation:** Every element within the simulation is accurately modeled as a virtual resource (such as host, CPU, network, disk and so on).

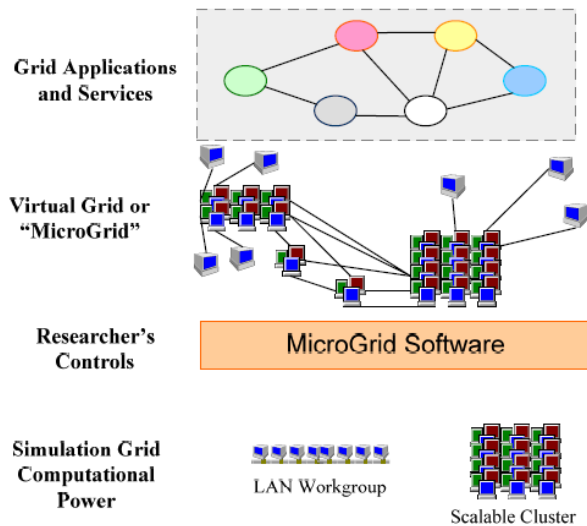


Figure 7: The MicroGrid Architecture

### 3. COMPARISON

The comparison of various simulators that we have surveyed summarized in below table-1.

Simulators/ Features	Sim Grid	GridSim	Optor Sim	Gan gSim	Grid Net	Micr oGrid
Network Qos	Not supported	Supported	Not supported	Not supported	Not supported	Not supported
Data Replication	Not supported	Supported	Supported	Supported	Supported	Supported
Disk I/O overhead	Not supported	Supported	Not supported	Not supported	Supported	Not supported
Scheduling user jobs	Supported	Supported	Not supported	Supported	Supported	Supported

CPU reservation of a resource	Not supported	Supported	Not supported	Not supported	Not supported	Not supported
Workload trace-back simulation	Supported	Supported	Not supported	Not supported	Supported	Not supported
Type of Transport Protocols	TCP	Datagram Oriented protocol similar to UDP	Not supported	-		TCP and UDP
Data Packetization	Not supported	Supported	Not supported	Not supported	Supported	Supported
Runtime Network Status	Supported	Supported	Not supported	-	Supported	Supported

Table-1

### 4. CONCLUSION

This paper provides an overview the simulation tools by outlining a few of the tools currently in research and conducts a brief survey on these tools. This paper survey popular simulators namely SimGrid, GridSim, OptorSim, GangSim. Gridnet and MicroGrid that are used for simulating grids and similar distributed systems. Other Grid simulators also exist but still are not presented in this paper.

It is difficult to find a tool that will cover all aspects needed to simulate a Grid; it seems that each one is suited best to work on a specific task. A general rule is that a simulator should be as coarse-grained as possible and concentrate on specific issues since it is not possible to simulate a real Grid. Therefore the selection of the tool depends on what is needed to be analyzed and how many resources are available.

This survey highlighted Grid Computing Simulators preliminary concepts, current developments and future prospect which will be helpful for researchers and developers for inspecting a specific facet of grid systems

### 5. ACKNOWLEDGMENTS

I am thankful to my guide Dr. Gupteshwar Gupta and Mr Anil Kumar Sharma for their encouragement and valuable support.

## 6. REFERENCES

- [1] Anthony Sulistio\_, Gokul Poduvaly, Constructing A Grid Simulation with Differentiated Network Service Using GridSim, Last accessed 15, April, 2013
- [2] J. Banks, J. Carson, B. Nelson, D. Nicol (2001). Discrete-Event System Simulation. Prentice Hall. p. 3. ISBN 0-13-088702-1.
- [3] <http://en.wikipedia.org/wiki/Simulation> , Last accessed 20, April, 2013
- [4] Survey of Grid Simulators, Network-level Analysis of Grid Applications © Copyright 2008 the Members of the EC-GIN Consortium
- [5] H. Casanova, Simgrid: A Toolkit for the Simulation of Application Scheduling, Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), May 15-18, 2001, Brisbane, Australia, IEEE Computer Society Press, USA.
- [6] Arnaud Legrand, Loris Marchal, Henri Casanova. "Scheduling Distributed Applications: the SimGrid Simulation Framework", Third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03), pp.138, 2003.
- [7] Henri Casanova, Arnaud Legrand, Martin Quinson. "SimGrid: a Generic Framework for Large-Scale Distributed Experiments", Tenth International Conference on Computer Modeling and Simulation, pp.126-131, 2008.
- [8] Mrs. R. Kingsy Grace, S. Swathi Priya, S. Surya ; "A Survey on Grid Simulators" IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 2, No.6, December 2012
- [9] The GRIDS Lab and the Gridbus Project <http://www.gridbus.org/>; Last accessed 20, April, 2013
- [10] R. Buyya and M. Murshed. "GridSim: A Toolkit for the Modelling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, Vol.14, pp. 1175-1220, May 2002.
- [11] Howell F, McNab R. "Simjava: A Discrete Event Simulation Library for Java." Proceedings of the 1st International Conference on Web-Based Modeling and Simulation, San Diego, California, January 11-14 1998.
- [12] Kreutzer W, Hopkins J, Mierlo MV. "SimJAVA - A Framework for Modeling Queueing Networks in Java". In Winter Simulation Conference, pp. 483-488, Atlanta, GA, 1997.
- [13] <http://grid-data-management.web.cern.ch/grid-data> management / optimisation/ptor/OptorSim "OptorSim, a Replica Optimizer Simulation." Last accessed 20, April, 2013
- [14] <http://www.gridpp.ac.uk/papers/OptorSimJune2002.pdf> ; William H. Bell, David G. Cameron, Luigi Capozza, A. Paul Millar, Kurt Stockinger, Floriano Zini ;Simulation of Dynamic Grid Replication Strategies in OptorSim; Last accessed 20, April, 2013
- [15] Catalin L. Dumitrescu, Ian Foster; GangSim: A Simulator for Grid Scheduling Studies <http://people.cs.uchicago.edu/~cldumitr/docs/GangSim.pdf>, Last accessed 20, April, 2013
- [16] C.L. Dumitrescu and I. Foster, "GangSim: A simulator for grid scheduling studies," in IEEE International Symposium on Cluster Computing and the Grid (CCGrid), Vol. 2, pp. 1151-1158, 2005.
- [17] Houda Lamehamedi, Zujun Shentu, and Boleslaw Szymanski1 Ewa Deelman2 Simulation of Dynamic Data Replication Strategies in Data Grids; <http://www.cs.rpi.edu/~szymansk/papers/hcw03.pdf>, Last accessed 25, April, 2013.
- [18] The ns Network Simulator homepage; <http://www.isi.edu/nsnam/ns/> ; Last accessed 25, April, 2013.
- [19] GriPhyN, "Grid physics network in atlas," ••. Available at <http://www.usatlas.bnl.gov/computing/grid/griphyn/>. Last accessed 25, April, 2013.
- [20] Takefusa A, Matsuoka S, Nakada H. "Overview of a Performance Evaluation System for Global Computing Scheduling Algorithms". Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing (HPDC8), Redondo Beach, California, August 3-6, 1999.
- [21] Naqvi S, Riguide M. "Grid Security Services Simulator (G3S) - A Simulation Tool for the Design and Analysis of Grid Security Solutions". Proceedings of the 1<sup>st</sup> IEEE International Conference on e-Science and Grid Computing (e-Science 2005), Melbourne, Australia, December 5-8, 2005.
- [22] Huaxia Xia, Holly Dail, Henri Casanova†, Andrew A. Chien The MicroGrid: Using Online Simulation to Predict Application Performance in Diverse Grid Network Environments [http://grail.sdsc.edu/papers/xia\\_clade2004.pdf](http://grail.sdsc.edu/papers/xia_clade2004.pdf), Last accessed 20, April, 2013
- [23] H. J. Song X. Liu D. Jakobsen R. Bhagwan X. Zhang K Taura A. Chien ; The MicroGrid: a Scientific Tool for Modeling Computational Grids <http://www.sc2000.org/techpap/papers/pap.pap286.pdf>, Last accessed 20, April, 2013
- [24] Song H, Liu X, Jakobsen D, Bhagwan R, Zhang X, Taura K and Chien A. "The MicroGrid: a Scientific Tool for Modeling Computational Grids". Proceedings of IEEE Supercomputing (SC 2000), Dallas, USA, Nov. 4-10, 2000.
- [25] Liu X, Xia H, Chien A. "Network Emulation Tools for Modeling Grid Behavior". Proceedings of CCGrid, May 2003