A Survey on Grid Simulators

Mrs. R. Kingsy Grace
Assistant Professor
kingsydhas@gmail.com
Computer Science and Engineering
Sri Ramakrishna Engineering College
Coimbatore-22

S. Swathi Priya, S. Surya
Final Year B.E. Students
swathipriya.91@gmail.com
suryaswamynathan@gmail.com
Computer Science and Engineering
Sri Ramakrishna Engineering College
Coimbatore-22

Abstract

Simulation is "Attempting to predict aspects of the behavior of some system by creating an approximate (mathematical) model of it". In large and distributed environment interconnected by networks it is not possible to evaluate the performance of systems in different scenarios. Hence simulators are used for modeling the real world processes in various areas. The performances are evaluated based on various metrics. This paper gives a brief study of various simulation tools such as GridSim for simulation of heterogeneous resources, OptorSim for managing storage, SimGrid for scheduling applications, GridNet for managing replicas of data in network, and the GangSim simulator. A comparison of various simulators is also studied.

Keywords: Distributed Computing, peer-to-peer (P2P) networks

I. INTRODUCTION

Simulation is used for modeling real world processes in many different application areas, including manufacturing, construction and computer science. It is used for modeling large complex systems and understanding their behavior, especially in the area of parallel and distributed systems. It provides the study of various issues, such as feasibility, behavior and performance without building the actual system, thus saving time, cost and effort. Several tools are constructed for simulation of grid computing environments since it is difficult to have easy access to ready-built grid platforms to evaluate the concepts and algorithms involved It is almost impossible to evaluate their performance under different scenarios, given the varying number of resources whose availability changes with time and users with different requirements in a grid computing environment. Simulation tools support the creation of a repeatable and controllable environment for performance evaluation. This facilitates grid computing researchers, educators, and students to conduct effective research, teaching, and learning with ease.

Simulation has various advantages which includes (1) No need to build a real system, nor the full-fledged application (2) Ability to conduct controlled and repeatable experiments (3) Almost no limits to experimental scenarios (4) Possible for anybody to reproduce results

The characteristics of a modeling tool are (1) Provide capabilities for constructing a representative Grid model (2) Provide capabilities for easily constructing a representative application model (3) Metrics for performance (4) Easy to use (5) Efficient.

II. GRIDSIM

Clusters, grids, and peer-to-peer (P2P) networks are a form of parallel and distributed computing. They are used for solving large-scale problems in science, engineering, and commerce. The resources are usually distributed across various domains interconnected by networks. This introduces a number of challenges in resource management, scheduling, security, resource and policy heterogeneity and fault tolerance. The performance needs to be evaluated under different scenarios such as varying number of resources and users with different requirements which is almost impossible in a grid environment as the resources and users are distributed across multiple organizations with their own policies. To overcome this limitation, *GridSim* [1], a Java-based discrete-event grid simulation toolkit was developed.

GridSim is a toolkit for modeling and simulation of heterogeneous resources, users, applications, brokers and schedulers in a Grid computing environment. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management.

A. System Architecture

GridSim adopts the multi-layered design architecture as shown in Figure 1. The first bottom layer is the Java interface and the JVM [10]. The second layer is SimJava [1], which provides an event-driven discrete event simulation package on top of JVM to drive the simulation for GridSim. The third layer is the GridSim toolkit that provides the modeling and simulation of Core Grid entities, such as resources and Grid Information Services, using the events of the second layer. The simulation of resource aggregators called Grid brokers [1] or schedulers is provided by the fourth layer.

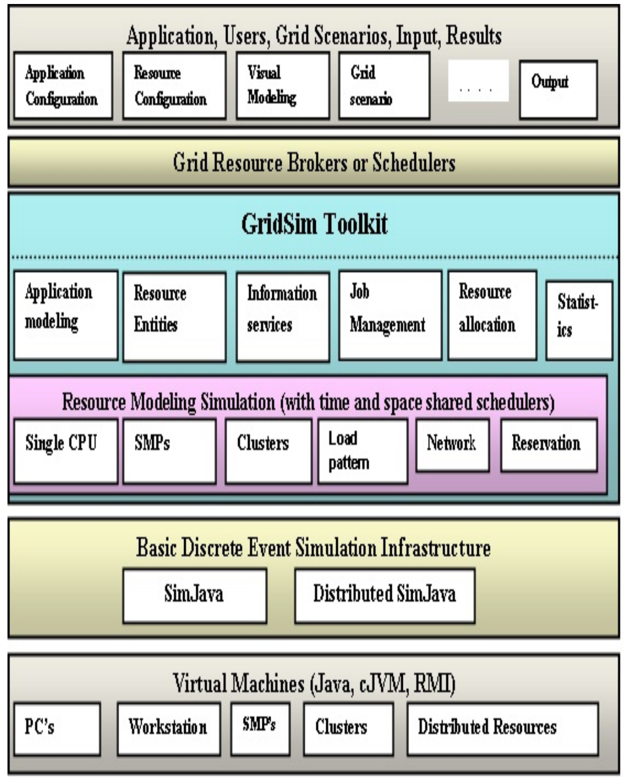

Figure.1: GridSim Architecture

The top layer focuses on application and resource modeling with different scenarios to evaluate scheduling and resource management policies, heuristics and algorithms. *Alea* Simulator is based on the latest GridSim 5 simulation toolkit which provides a simulation environment that supports simulation of varying Grid scheduling problems.

B. Features

Salient features of the GridSim toolkit include the following:

- It allows modeling of heterogeneous types of resources.
- Resources can be modeled operating under space- or time -shared mode.
- Resources can be located in any time zone.

- Multiple user entities can submit tasks for execution simultaneously for the same resource.
- It supports simulation of both static and dynamic schedulers.

III. OPTORSIM

Grid technology provides solution for data handling and storage problems. It is important to make the best use of a grid's resources. A typical Grid job requires three types of resources: computing facilities, data access and storage, and network connectivity. The Grid must make scheduling decisions for each job based on the current state of these resources. Complete optimization is achieved when the combined resource impact of all jobs is minimized, allowing jobs to run as fast as possible. In large-scale grids, the replication of files to different sites and the appropriate scheduling of jobs are important mechanisms which can reduce access latencies and give improved usage of resources such as network bandwidth, storage and computing power. To provide an optimal scheduling and replication strategy, the grid simulator OptorSim [3] was developed as part of the European Data Grid project.

A. Architecture

OptorSim's architecture [3] is shown in Figure 2. It is based on that of the European Data Grid data management components. Computing and storage resources are represented by Computing Elements (CEs) and Storage Elements (SEs) respectively, which are organized in Grid Sites. CEs run jobs by processing data files, which are stored in the SEs. Users submit jobs to the grid according to the submission pattern chosen, and then a Resource Broker (RB) schedules the jobs to Grid Sites. When a job is being processed at a CE, it will go through its list of files to process according to the chosen access pattern. If a file is required which is not present on the execution site, it must either be replicated or read remotely. Each site handles its file content with a Replica Manager (RM), within which a Replica Optimizer (RO) contains the replication algorithm which drives automatic creation and deletion of replicas.

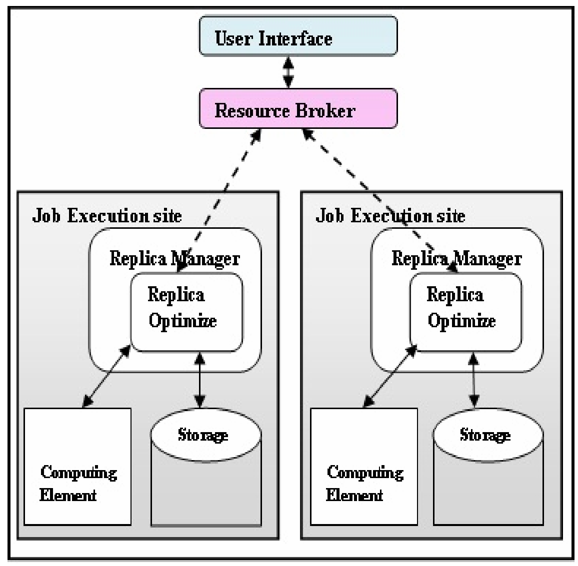

Figure 2: OptorSim Architecture

B. Optimization Algorithms

There are two different types of optimization which may be investigated using OptorSim [5]: the scheduling algorithms used by the RB to allocate jobs, and the replication algorithms used by the RM at each site to decide when to replicate a file, which file to replicate and which to delete. The time taken to complete the job must be reduced with the best utilization of grid resources .In the short term, an individual user wants their job to finish as quickly as possible, but in the long term the goal is to have the data distributed in such a way as to improve job times for all users, thus giving the greatest throughput of jobs.

IV. SIMGRID

Distributed computing comprises of various fields such as cluster computing, computational grids, desktop grids and peer-to-peer (P2P) systems. The study of scheduling algorithms for such application is an active area of research. Given a scheduling problem it is difficult to evaluate and compare their performances under different scenarios. Hence there is a need for simulations, which enable reproducible results and also make it possible to explore wide ranges of platform and application scenarios. To overcome this limitation SimGrid [7] was developed.

The **SimGrid version 1** [5] 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 Version 2 [5] adds a new software layer to provide high-level abstractions.

SimGrid Version 3 [5] has three main layers-Base, Simulation and Programming

A. Components of the SimGrid

The components of the SimGrid [6] toolkit are given in Figure 3 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 [6] provides several types of resources like workstation, network, CPU and time resources

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.

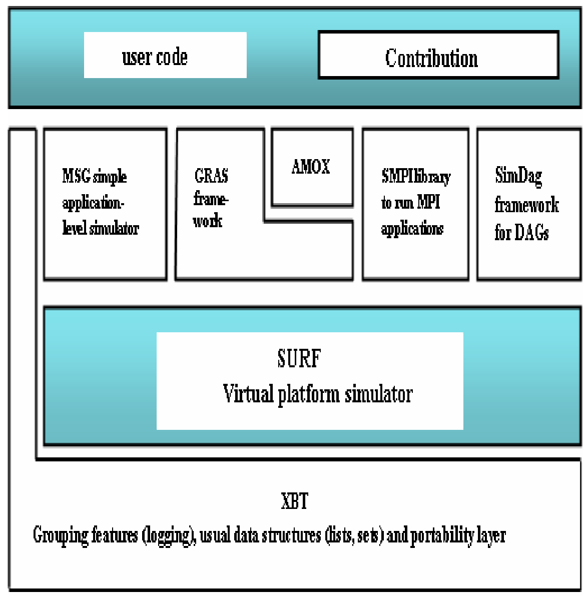

Figure 3: SimGrid Components

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 [6] is a new implementation of DAGs handling and it is built on top of SURF

B.Features

Some of the key features of SimGrid are:

- Resource models for CPUs and network links
- Arbitrary, dynamic, trace-based resource performance metrics
- Resource time-sharing and time-slicing
- Task dependencies

- Performance prediction error simulation
- Simple API

V. GRIDNET

A Data Grid is a collection of geographically distributed computers and storage resources located in different parts of the world that facilitate sharing of data and resources. It is used to solve large scale computational problems. The size of the data in Data Grid is huge and is of the order of Terabytes and Petabytes. Efficient access to such huge and distributed data is hindered by the high latencies in the network. Hence there is a necessarily to replicate the data in order to improve the scalability and the data availability. For this purpose a Grid Simulator-GridNet is developed. It provides a simulation framework which can be used to model different Data Grid configurations and the resource specifications. The goals of replication are to reduce access latency and to improve data availability by dynamically adapting to the changes in the user and network.

The replication management service evaluates the data access gains and the creation and maintenance before moving or copying the data. The replication decisions are made based on various factors like read/write statistics, network latency, bandwidth, and replica size and system reliability.

A. Architecture

Figure 4 shows GridNet architecture. GridNet simulator [11] is a modular simulator built on top of the network simulator NS. Each GridNet node consists of a NS node, a storage element, a replica manager or a monitoring agent and maintains a replica routing table. The replica manager makes decision about when to create and delete replicas. It is based on various statistics such as bandwidth information and storage space. The data is collected by the monitoring module that is responsible for computing the data requests. The collected data is evaluated by the optimizer using cost function Each node maintains a replica routing table that allows it to locate the closest replica site and the node to which it should forward its request. Each time when new replicas are created or deleted; replica routing tables are updated accordingly.

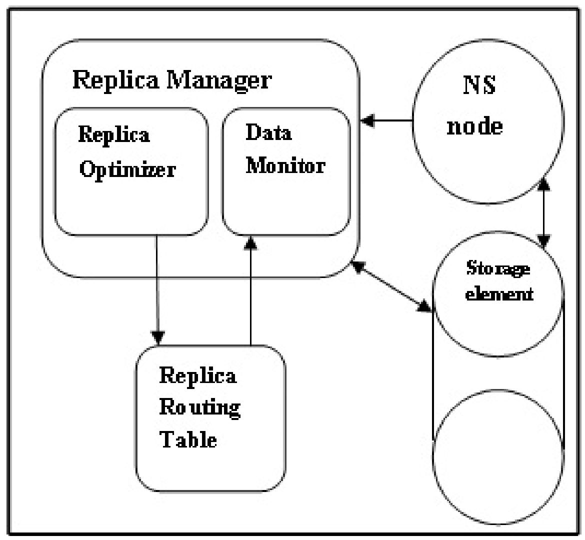

Figure 4: GridNet Architecture

VI. GANGSIM

In Large scale distributed systems, job scheduling is complex due to the system and workload characteristics. A grid environment consists of applications that involve numerous loosely coupled jobs handling large data sets .Scheduling policies for such applications are a complex task. Due to complex parameters modeling this system behavior is very difficult. Hence GangSim [12] simulator was developed with scheduling policies mainly focusing on the interactions between local and community reservation allocation policies. GangSim simulator models not only the sites but also the Virtual Organization (VO) users and planners. It models policies both at the site and the VO levels. GangSim permits parallel execution and can combine simulated components running on real resources. The comparisons of various simulators are given in Table I.

Table I

Functionalities	GridSim	OptorSim	SimGrid	GridNet	GangSim
Data Replication	yes	yes	no	yes	yes
Disk I/O overheads	yes	no	no	yes	no
Scheduling user jobs	yes	no	yes	yes	yes
CPU reservation of a resource	yes	no	no	no	no
Workload trace-back simulation	yes	no	yes	yes	no
Differentiated network QoS	yes	no	no	no	no

VII. CONCLUSION

This paper provides an overview the simulation tools by outlining some of the tools currently in research and conducts a brief survey on these tools. It reinforces the belief that simulation does help in the modeling and development of real-world processes, especially in the area of parallel and distributed systems for high performance computing. Simulation tools are used to model and predict the behaviors of the entities and the interactions among them in a parallel and distributed computing environment. Mainly we have discussed about five simulators namely GridSim, OptorSim, SimGrid, GridNet and GangSim. We have briefly outlined about their purposes, features, and architectures.

REFERENCES

[1] 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.

[2] Anthony Sulistio, Chee Shin Yeo, and Rajkumar Buyya***."A Taxonomy of Computer-based Simulations and its Mapping to Parallel and Distributed Systems Simulation Tools"***, International Journal of Software: Practice and Experience, Volume 34, Issue 7, pp- 653-673, Wiley Press, USA, June 2004.

[3] William H. Bell, David G. Cameron, Luigi Capozza,A. Paul Millar, Kurt Stockinger, Floriano Zini***."OptorSim-A Grid Simulator for Studying Dynamic Data Replication Strategies"***, International Journal of High Performance Computing Applications, Vol.17 (4), pp-403-416, 2003

[4] D. G. Cameron, A. P. Millar, C. Nicholson, Scotland R. Carvajal-Schiaffino, F. Zini, D.Cameron, R.Carvajal-Schiaffino, P.Millar, C.Nicholson, K.Stockinger and F.Zini. ***"Evaluating Scheduling and Replica Optimisation Strategies in OptorSim"***, In 4th International Workshop on Grid Computing (Grid2003), Phoenix, Arizona, USA, 2003.

[5] 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.

[6] 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.

[7] 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 (CCGrid2001), May 15-18, pp - 430, Brisbane, Australia, 2001.

[8] Manzur Murshed and Rajkumar Buyya, ***"Using the GridSim Toolkit for Enabling Grid Computing Education",*** International Conference on Communication Networks and Distributed Systems Modeling and Simulation (CNDS 2002), January 27-31, San Antonio, Texas, USA. 2002.

[9] M. Baker, R. Buyya, and D. Laforenza, ***"The Grid: International Efforts in Global Computing"***, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, July 31 - August 6, 2000, Rome, Italy.

[10] Y. Aridor, M. Factor, and A. Teperman, ***"cJVM: a Single System Image of a JVM on a Cluster",*** Proceedings of the 29th International Conference on Parallel Processing (ICPP 99), September 1999, Fukushima, Japan, IEEE Computer Society Press, USA

[11] Lamehamedi, H,Zujun Shentu , Szymanski. B, Deelman, E.***"Simulation of dynamic data replication strategies in Data Grids",*** Parallel and Distributed Processing Symposium, pp: 22-26 April 2003

[12] C. L. Dumitrescu,I.Foster.***"GangSim: a simulator for grid scheduling studies"*** Proceedings of Fifth IEEE International Symposium on Cluster Computing and the Grid(CGGrid'05),pp:1151-1158,2005