Grid Computing and GridSim Toolkit: An overview

Tushina Bedwal
Assistant Professor
IMS Engineering College
Ghaziabad

Radhika Tayal
Assistant Professor
IMS Engineering College
Ghaziabad

Anjali Batra
Assistant Professor
IMS Engineering College
Ghaziabad

ABSTRACT
In Grid Computing technology the fundamental concept that is used is to connect the geographically distributed resources. Commercial networks have been used for this purpose due to which some network effects get introduced in grid computing. This paper gives an introduction to grid computing technology. In this paper some of the problems of resource management and scheduling have also been addressed which includes the management of resources and scheduling of applications according to the requirements of consumers and owners. The GridSim toolkit has been developed to overcome the above problems. An introduction to this toolkit is also given in this paper.

Keywords
Grid Computing, Grid Scheduling, Resource management, Job Scheduling, GridSim

1. INTRODUCTION TO GRID & GRID COMPUTING
Buyya et. al. [7] defined grid as follows: "It a type of parallel and distributed system that enables the sharing, exchange, selection and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users quality-of-service requirements." The aim of a computational grid, is to allow the users to access the IT assets and aggregated processing powers, whenever they need them.

The most important resource included in a grid is a processor but grids also include other resources such as storage, data repositories, sensors, graphics and terminal devices, equipment, instrumentation, applications and so on.

Grid computing combines computers from multiple domains to reach a common goal i.e.to solve a single problem.

-It creates an environment that provides the ability to share and transparently access geographically distributed heterogeneous resources in a consistent manner for solving large scale data intensive problems in science, commerce and engineering.

To achieve this, grid computing needs the technology to virtualize IT resources and standards in the areas of security, scheduling, systems management, accounting, and so on. Grid computing uses middleware to divide and apportion pieces of a program among different computers.

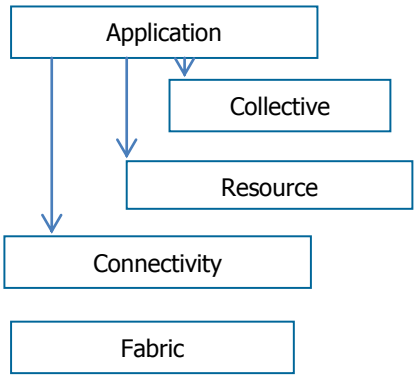

Fig.1 Architecture of Grid Computing

Grids provide services at five different layers as presented in the architecture.

- **Fabric layer:** Provides the shared resources such as computer, network resource, storage and data repository, etc.
- **Connectivity layer:** Contains the communication and authentication protocols that help for easy and secure grid-specific network functions.
- **Resource layer:** Defines protocols, APIs for secure negotiations, initiation, monitoring control, accounting and payment of sharing operations on individual resources.
- **Collective Layer:** Contains protocols and services that specify interactions among a collection of resources, discovery of VO resources and directory services.
- **Application Layer:** Contains user applications that are built on top of APIs and operate within VO environment.

Following advantages of Grid Computing have been identified:

Efficient use of underutilized resources-
Using load sharing software, it distributes the load evenly on servers on basis of resources and policies.

Modular grid environment-
Grid environments are modular in nature and don't have point of failures. If any of the servers in grid fails, other server can pick up the load.

Virtual resources and virtual organizations for collaboration.

Scalability-
Resources can be added when required by simply installing grid client on additional servers.

Parallel job execution-Performance can be speed up by allowing jobs to be executed in parallel on many servers.

Can solve larger, difficult problems in less time.

1.2 RESOURCE MANAGEMENT AND SCHEDULING

The concept of Grid computing has evolved as a distributed computing technology that provides access to geographically dispersed heterogeneous resources for solving large scale problems.

Since computational resources are heterogeneous in nature, the allocation of available resources to the applications is a complicated task in grid computing environment. This problem of allocating resources (Grid scheduling) requires a model that allows local and external schedulers to communicate with each other in order to achieve an efficient management of the resources.

These distributed resources are under ownership of different organizations and they have their own access policy, mechanism and cost. The resource Owners and consumers have different goals, strategies, and demand patterns.

In managing such complex Grid environments, traditional simple approaches for resource management cannot be used, as these approaches use centralized policies that require complete state information and a common fabric management policy.

In large Grid environments, defining such a common fabric management policy is impossible.

Two more approaches can be used in managing distributed resources: hierarchical and decentralized scheduling or a combination of them.

The motive of resource management is to schedule applications so that available resources of the grid environment can be used efficiently.

The user interacts with a resource broker that hides the complexities of grid computing and presents the user with a single integrated resource. The resource broker acts as a mediator between grid resources and users, it discovers resources that the user can access through grid information server, negotiates with resources using middleware services, binds applications to resources (scheduling), stages the application and data for processing (deployment) and finally collects the results.

It also monitors application execution progress and manages resource failures and changes in the grid infrastructure.

The resource brokers use scheduling algorithms and policies for mapping jobs to resources to optimize system or user objectives. A View of GRID System is shown in Fig.2

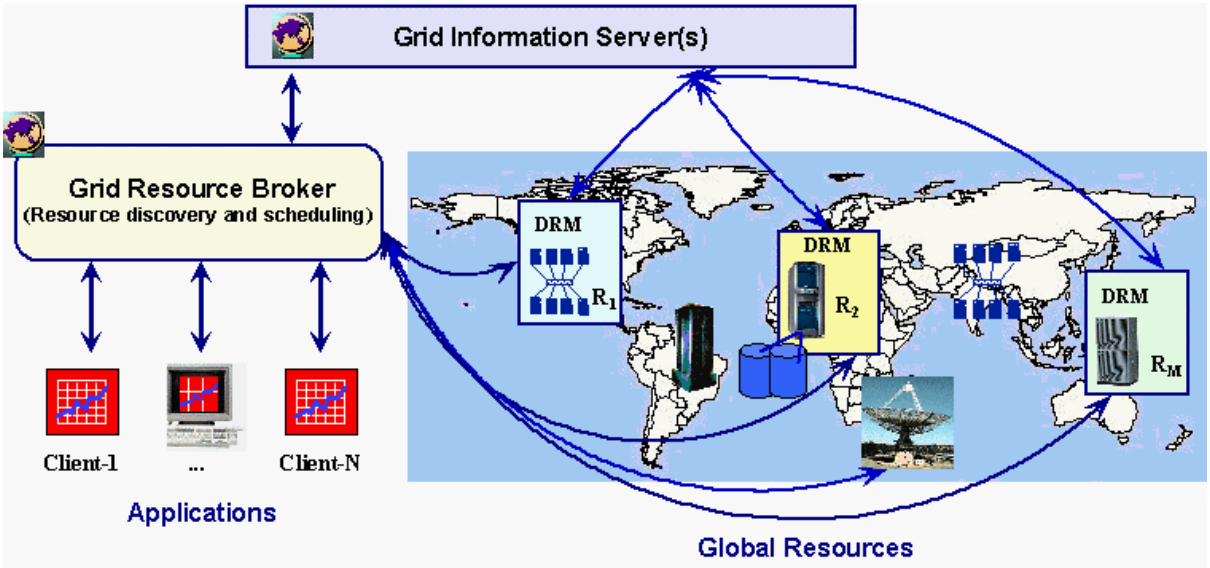

Fig. 2 Grid System

The grid middleware provides services to link a grid user and distributed resources through a resource broker. Such services include as remote process management, co-allocation of resources, information, authentication, security, storage access, and Quality of Service (QoS) for guaranteed availability and minimised computational cost.

1.3 CHALLENGES IN RESOURCE MANAGEMENT AND SCHEDULING

Multiple layers of schedulers -The higher level scheduler has less information about the distributed resources, since local resource managers control the resources actually.

Lack of control over resources-
Grid scheduler does not have ownership over the resources.

Shared resources and variance –
It does not provide dedicated access to resources as resources are shared .which leads to high degree of unpredictability and variance.

Conflicting performance goals- Owners and end users have conflicting preferences. They may have different local policies, cost models, security mechanisms.

2.1 INTRODUCTION TO GRIDSIM TOOLKIT

The GridSim toolkit is a general purpose discrete-event grid simulation package which implemented in Java.

- It has been developed to overcome above mentioned problems. This toolkit allows composition of applications, information services for resource discovery, and interfaces for allocating the resources to the applications and then managing their execution.
- It also has the ability to model the characteristics of grid resources and networks with different configurations, capabilities, policies and domains.

- This toolkit provides a facility for simulation of scheduling algorithms in different grid environments having large number of resources.
- A resource can be either single or multi-processor with shared or distributed memory managed by schedulers.
- It provides a collection of classes for simulating discrete events. It can simulate distributed systems, networks, communication protocols and computer architectures. Resource brokers are used to create and aggregate classes of heterogeneous resources.
- It provides a large amount of options like resource capability (in MIPS), location of resources, multiple concurrent task submission, different parallel application models, network speed, resource booking and both dynamic and static schedulers.
- This toolkit has been applied successfully to simulate a grid resource broker like Nimrod-G and to evaluate the performance of cost- and time-constrained optimization scheduling algorithms.

Following are the features of Gridsim toolkit:

- It allows modelling and simulation of heterogeneous resource characteristics and their failure properties.
- It supports reservation based mechanism for resource allocation.
- It can simulate applications with different parallel application models.
- There is no limit on the number of application jobs that can be submitted to a resource.
- It allocates resources to jobs based on space-or time-shared mode.
- It supports simulation of static as well as of dynamic schedulers.
- It allows modelling of several regional Grid Information Service (GIS) components, hence one can simulate with multiple virtual organisations.
- Resources can be located in any time zone.
- Application tasks can be CPU or I/O intensive.
- Statistics of operations can be recorded and can be analysed using GridSim statistics analysis methods.
- Network speed can be specified between resources.
- Resource capability can be defined in MIPS.
- Multiple users can submit jobs for simultaneous execution
  on the same resource which can be time-shared or space-shared.
- It has a network traffic functionality based on a probabilistic distribution which can be used to simulate data-intensive jobs over a congested public network.
- It has well-defined interfaces for implementing different resource allocation algorithms.
- It has a functionality that reads workload traces from supercomputers for simulating a realistic grid environment.

2.2 ARCHITECTURE OF GRIDSIM

GridSim has a modular multi-layered architecture that can be extended to add new layers in Gridsim whenever needed.
A modular architecture for GridSim platform and components is shown in Fig.3

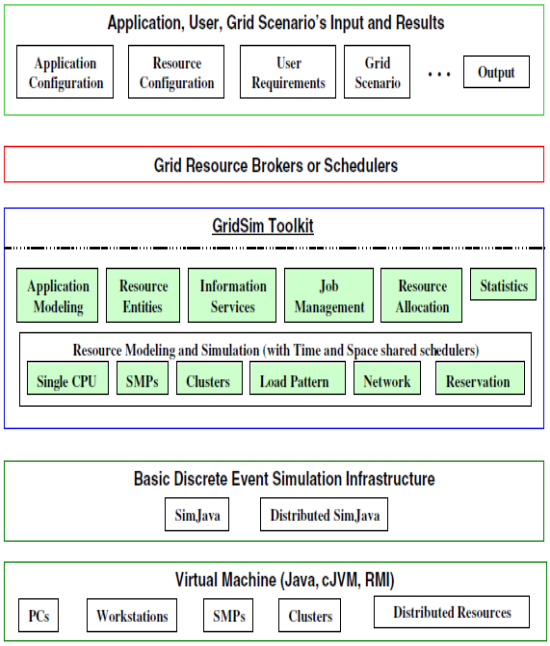

Fig.3 Modular Architecture for GridSim

First layer contains the Java Interface and runtime machinery JVM (Java Virtual Machine).
Second layer has basic discrete-event infrastructure built using the interfaces provided by the first layer. SimJava and distributed SimJava are two such infrastructure implementations.
Third layer deals with the modelling and simulation of Grid entities such as resources, information services, job management, resources allocation etc.
Fourth layer deals with the simulation of resource aggregators called Grid resource brokers.
Topmost layer focuses on resource and application modelling in different scenarios by using the services provided by the two lower-level layers for evaluation of scheduling and resource management policies and algorithms.
GridSim Entities
Each simulated system that interacts with each other is referred to as entity.
Gridsim supports a number of entities
-to simulate hetrogeneous resources in time-or space-shared systems
-to simulate networks used for communication among resources.
A flow diagram in GridSim based simulations is shown in Fig.4.
Simulations using GridSim can contain following entities:
Users- A Grid user is represented by a user entity.
Brokers- Every user is linked to an instance of the Broker entity.
Resource-A Grid resource is represented by a Resource entity.
Grid information service –keeps track of available resources in Grid.
Input &Output- Information flows among the GridSim entities via their Input and Output entities.

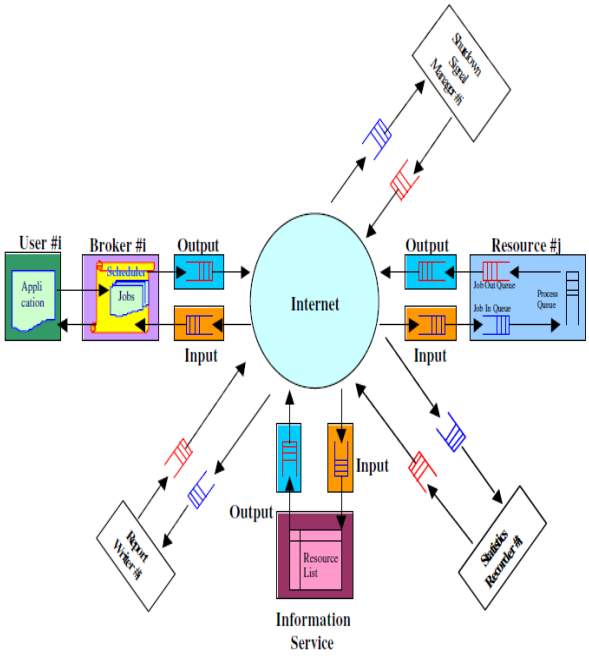

Fig.4 Flow Diagram For GridSim Simulations

3. CONCLUSION

Grid computing is the biggest hope for delivering computing (as utility) to homes and offices.Grids exploit synergies that result from cooperation of autonomous entities:

- Resource sharing, dynamic provisioning, and aggregation at global level can be done with the help of this technology.
- Several open source middleware technologies such as Grid bus exists demonstrating Grid potential.
- Grid computing offers enormous opportunities for realizing e-Science and e-Business at global level**.**
- GridSim toolkit is suitable for application scheduling simulations in Grid computing environment**.**

4. REFERENCES

[1] S. Ayyub, D. Abramson, C. Enticott, S. Garic, and J. Tan. Executing Large Parameter Sweep Applications on a Multi- VO Testbed. In Proceedings of CCGRID ’07, pages 73–82, Washington, DC, USA, 2007. IEEE Computer Society.

[2] R. M. Badia et al. Programming Grid Applications with GRID Superscalar. Journal of Grid Computing, 1(2):151–170, June 2003.

[3] L. Baduel et al. Grid Computing: Software Environments and Tools, chapter Programming, Deploying, Composing, for the Grid. Springer-Verlag, January 2006.

[4] F. Berman and R. Wolski. The AppLeS Project: A Status Report. 8th NEC Research Symposium, Berlin, Germany, May 1997.

[5] R. Buyya, D. Abramson, and J. Giddy. Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. High Performance Computing (HPC) ASIA, 2000.

[6] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In Proceedings of the Super Computing Conference (SC’2000), Nov 2000.

[7] M.-H. Chen, Q.-M. Shao, and J. G. Ibrahim. Monte Carlo Methods in Bayesian Computation. Springer-Verlag, New York, 2000.

[8] J. Dongarra, G. A. Geist, R. Manchek, and V. S. S. m. Integrated PVM Framework Supports Heterogeneous Network Computing. Computers in Physics, 7(2):166–174, 1993.

[9] H. El-Rewini, T. G. Lewis, and H. H. Ali. Task Scheduling in Parallel and Distributed Systems. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[10] S. Fitzgerald. Grid Information Services for Distributed Resource Sharing. In Proceedings of HPDC ’01, page 181, Washington, DC, USA, 2001. IEEE Computer Society.