Grid Resource Computing Environment Simulation Using Gridsim Toolkit

Pradeep Singh Rawat
Dehradun Institute of Technology
Dehradun, INDIA
Email Id: pradeep_kec09@yahoo.com

Anuj Kumar Yadav
Dehradun Institute of Technology
Dehradun, INDIA
Email Id: anujbit@gmail.com

Varun Barthwal
Hemwanti Nandan Bhuguna
Garhwal University
Srinagar, INDIA
Email Id: varuncsed1@gmail.com

Abstract*—To study scientific application different computing paradigm used. Simulation of tool base Grid infrastructure plays its roll for study of grid base computation. We introduce the here Grid computing paradigm for resource coordination over the global computation world. Resources management and scheduling of applications in such large-scale distributed systems is a complex undertaking in case of Grid computational environment. It is hard and even impossible to perform scheduler performance evaluation in a repeatable and controllable manner as resources and users are distributed across multiple organizations using their own policies. To overcome this limitation, we used a Java-based discrete-event grid simulation toolkit called GridSim. The toolkit supports modeling and simulation of heterogeneous grid resources (both time- and space-shared), users and application models. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management. To demonstrate suitability of the GridSim toolkit, we have simulated a Grid environment. We deploy the Gridlet on grid resources which include machine properties e.g. mips rating. We simulated the resource entities user and router connected via network Topology using Gridsm. Simulation results are obtained for different grid resource which are distributed across the globe in a computation world.

Keywords-MIPS, Gridlet, Gridsim, Topology, Simulation

I. INTRODUCTION

In the presence of the Internet and the availability of powerful computers and high-speed networks are changing the way we do large-scale parallel and distributed computing. The interest in coupling geographically distributed (computational) resources is also growing for solving large-scale problems, leading to what is popularly called the Grid [1] and peer-to-peer (P2P) computing [2] networks. These enable sharing, selection and aggregation of suitable computational and data resources for solving large-scale data intensive problems in science, engineering, and commerce. The Grid consists of four key layers of components: fabric, core middleware, user-level middleware, and applications [3]. The Grid fabric includes computers (low-end and high-end computers including clusters), networks, scientific instruments, and their resource management systems. The core Grid middleware provides services that are essential for securely accessing remote resources uniformly and transparently. The services they provide include security and access management, remote job submission, storage, and resource information.

The user-level middleware provides higher-level tools such as resource brokers, application development and adaptive runtime environment. The Grid applications include those constructed using Grid libraries or legacy applications that can be Grid enabled using user-level middleware tools. The user essentially interacts with a resource broker that hides the complexities of Grid computing [4,5]. The broker discovers resources that the user can access using information services, negotiates for access costs using trading services, maps tasks to resources (scheduling), stages the application and data for processing (deployment), starts job execution, and finally gathers the results.

It is also responsible for monitoring and tracking application execution progress along with adapting to the changes in Grid runtime environment conditions and resource failures. The computing environments comprise heterogeneous resources (PCs, workstations, clusters, and supercomputers), fabric management systems (single system image OS, queuing systems, etc.) and policies, and applications (scientific, engineering, and commercial) with varied requirements (CPU, input/output (I/O), memory and/or network intensive). The users: producers (also called resource owners) and consumers (also called end-users) have different goals, objectives, strategies, and demand patterns. More importantly both resources and end-users are geographically distributed with different time zones. In managing such complex Grid environments, traditional approaches to resource management that attempt to optimize system-wide measures of performance can not be employed. This is because traditional approaches use centralized policies that need complete state information and a common fabric management policy, or decentralized

978-9-3805-4416-8/15/$31.00 ©2015 IEEE

consensus based policy. The GridSim toolkit supports modeling and simulation of a wide range of heterogeneous resources, such as single or multiprocessors, shared and distributed memory machines such as PCs, workstations, SMPs, and clusters with different capabilities and configurations. In large-scale Grid environments, it is impossible to define an acceptable system-wide performance matrix and common fabric management policy. Apart from the centralized approach, two other approaches that are used in distributed resource management are: hierarchical and decentralized scheduling or a combination of them [6].

II. RELATED WORK

Simulation has been used extensively for modeling and evaluation of real world systems, from business process and factory assembly lines to computer systems design. Accordingly, over the years, modeling and simulation has emerged as an important discipline and many standard and application- specific tools and technologies have been built. They include simulation languages (e.g. Simscript [12]), simulation environments(e.g. Parsec [13]), simulation libraries (SimJava [14]), and application specific simulators (e.g. OMNet++ network simulator [15]). While a large body of knowledge and tools exists, there are very few tools available for application scheduling simulation in Grid computing environments. The notable ones are: Bricks [16], MicroGrid [17], SimGrid [18], and our GridSim toolkit. The Bricks simulation system [16], developed at the Tokyo Institute of Technology in Japan, helps in simulating client-server like global computing systems that provide remote access to scientific libraries. The MicroGrid emulator [17], undertaken at the University of California at San Diego (UCSD), is modeled after Globus [19]. It allows execution of applications constructed using the Globus toolkit in a controlled virtual Grid emulated environment.

The results produced by emulation can be precise, but modeling numerous applications, Grid environments, and scheduling scenarios for realistic statistical analysis of scheduling algorithms is time consuming as applications run on emulated resources. Also, scheduling algorithms, designers generally work with application models instead of constructing actual applications. Therefore, MicroGrid's need for an application constructed using Globus imposes significant development overhead. However, when an actual system is implemented by incorporating scheduling strategies that are evaluated using simulation, the MicroGrid emulator can be used as a complementary tool for verifying simulation results with real applications. The SimGrid toolkit [18], developed at 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. It is a powerful system that allows creation of tasks in terms of their execution time and resources with respect to a standard machine capability. Using SimGrid APIs, tasks can be assigned to resources depending on the scheduling policy being simulated. It has been used for a number of real studies, and demonstrates the power of simulation. However, because SimGrid is restricted to a single scheduling entity and time-shared systems, it is difficult to simulate multiple competing users, applications, and schedulers, each with their own policies when operating under a market-like Grid computing environment, without extending the toolkit substantially. Also, many large-scale resources in the Grid environment are space-shared machines and they need to be supported in simulation. Hence, our GridSim toolkit extends the ideas in existing systems and overcomes their limitations accordingly. Finally, we have chosen to implement GridSim in Java by leveraging SimJava's [14] basic discrete event simulation infrastructure. This feature is likely to appeal to educators and students since Java has emerged as a popular programming language for network computing.

III. GRIDSIM

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 domain distributed computing systems such as clusters and Grids. Application schedulers in the Grid environment, called resource brokers, perform resource discovery, selection, and aggregation of a diverse set of distributed resources for an individual user. This means that each user has his or her own private resource broker and hence it can be targeted to optimize for the requirements and objectives of its owner. In contrast, schedulers, managing resources such as clusters in a single administrative domain, have complete control over the policy used for allocation of resources. This means that all users need to submit their jobs to the central scheduler, which can be targeted to perform global optimization such as higher system utilization and overall user satisfaction depending on resource allocation policy or optimize for high priority users. Resource simulation and performance evaluation are highlighted in the next two section. Using Gridsim toolkit simulation results and scenario are highlighted in the section VI and VII.

IV. CASE STUDY

A. Simulation of gridsim based application

We are simulating one Grid resource with three Machines that contain one or more PEs. Deploy application on real grid infrastructure so discrete event simulation plays its role. Scientific application may benefit from Grid because they typically present non-uniform usage patterns across the grid infrastructure. We employed a layered and modular architecture for Grid simulation to leverage existing technologies and manage them as separate components. Create user and resource entities connected via network topology, user and router using Gridsim toolkit to evaluate performance. Here we setup the grid infrastructure which includes two grid users and 5 Gridlet which are deployed on a grid infrastructure having 3 machines. Each machine have same MIPS rating i.e. 377 and each machine have 2 to 4 no of PEs. Each machine follow the same system architecture i.e. Sun Ultra and same operating system i.e. Solaris. We get the simulation results for different scenario of Grid resource entity, Grid resource properties shown in table 1 and 2.

V. GRID SIMULATION INPUT PARAMETERS

TABLE I. GRID RESOURCE ENTITY

Number of grid user	Baud rate (bit/second)	Propagation delay(ms)	Mtu (byte)
2	1000	10	1500
	10000	10	2000
	100000	10	2500
	1000000	10	3000
	10000000	10	3500
	1000000000	10	4000

Above Table I include the Grid resource entity. Entity includes the number of grid user, propagation delay, Baud rate and Mtu. For same number of grid user we use different scenario for grid resource.

TABLE II. GRID RESOURCE PROPERTIES

Total Gridlet (grid user entities)	Machine id	No of PEs	Mips Rating	system architecture	operating system	time zone this resource located
5	0	4	377	Sun Ultra	Solaris	9.0
	1	4	377	Sun Ultra	Solaris	9.0
	2	2	377	Sun Ultra	Solaris	9.0

Above Table II include the Grid resource properties i.e. number of Gridlet, machine identification, each machine have number of PEs with unique MIPS rating. SunUltra system architecture is used for deployment of application. Resoures are allocated in a time zone 9.0. Grid infrastructure is allocated in a fixed geographic region.

TABLE III. NETWORK TOPOLOGY AMONG ENTITIES

Entity	Connection band width(Mbps)	Link	Source/ Destinaton
User(s)	1	User - - r1	User(s)/ GridResource(s)
r1	10	r1- - r2	
r2	1	r2 - - GridResource(s)	

Above Table III include the physical structure used for connection among the Grid entity. Connection bandwidth indicates the quality of service and link utilization. Transmission link may be in between internetworking hardware or source and destination. In this scenario source entity is user and destination entity is Grid resource.

VI. SIMULATION RESULTS

TABLE IV. PING INFORMATION FOR USER_0

Entity Name	Entry Time	Exit Time	Bandwidth	Number of Hops	Bottleneck Bandwidth Bits/second	Round Trip Time (second)
User_0	2211.7226	3683.000	1000.000	3	1000.0	1769.660
Router1	3683.010	3685.410	10000.000			
Router2	3685.420	5444.220	1000.000			
Res_0	5444.230	5448.230	1000.000			
Router2	5448.240	5448.640	10000.000			
Router1	5448.650	5452.650	1000.000			
User_0	5452.660	N/A	N/A			

Above Table IV shows the simulation results for each user_ 0 assocated with Grid computation. We got the ping information of user_0. There are maximum 3 hops for Each Grid entity. Each entity have unique entry and exit time. For different scenario entry and exit time is continuously increasing along with ping information of user_0. Bottleneck bandwidth is same for all entity and round trip delay is recorded 1769.660 second. While deploying the Gridlet on Grid infrastructure the ping information may very across the entities.

TABLE V. PING INFORMATION FOR USER_1

Entity Nme	Entry Time	Exit Time	Bandwidth	Number of Hops	Bottleneck Bandwidth(bits/second)	Round Trip Time(second)
User_1	2603.020	4295.000	1000.000	3	1000.0	2085.660
Router1	4295.010	4295.410	10000.000			
Router2	4295.420	6372.220	1000.000			

Res_0	6372.23 0	6376. 230	1000.00 0			
Router2	6376.24 0	6376. 640	10000.0 00			
Router1	6376.65 0	6380. 650	1000.00 0			
User_1	6380.66 0	N/A	N/A			

Above Table V shows the simulation results for each user_ 1 associated with Grid computation. We got the ping information of user_1. There are maximum 3 hops for Each Grid entity. Each entity have unique entry and exit time. For different scenario entry and exit time is continuously increasing from 2603.02 second to 6380.66 second along with ping information of user_0. Bottleneck bandwidth is same for all entity and round trip delay is recorded 2085.660 second. While deploying the Gridlet on Grid infrastructure the ping information may very across the entities starting from source to destination. We are simulating one Grid resource with three Machines that contains one or more PEs.

VII. EXPERMENTAL RESULTS FOR VARING PARAMETER ACROSS GRID ENTITY ASSOCIATED WITH GRID INFRASTRUCTURE

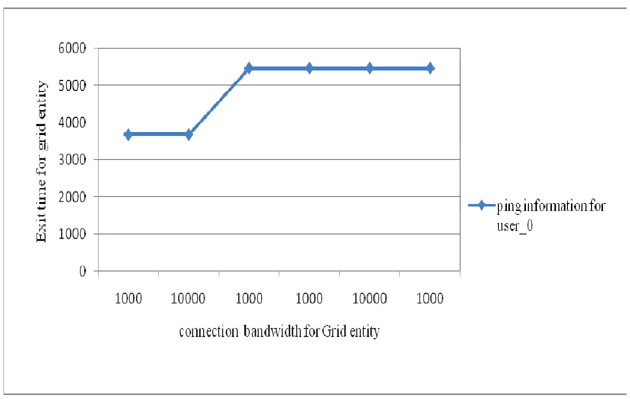

Fig. 1. Exit time versus connection bandwidth for Grid Entity associated with grid user_0

As shown in the above diagram variation of exit time along with connection bandwidth. For first two grid resource entity user_0 and router1 is small for bandwidth variation ten times. For the range of 10000 to 1000 Mbps variation of exit time of entity increases which is considerable. I t is clear that for high quality of service for grid user range of connection band width should be 1000 to 10000.

As shown in the Fig 2 variation of exit time along with connection bandwidth. For first two grid resource entity user_1 and router1 is small for bandwidth variation ten times. For the range of 10000 to 1000 Mbps variation of exit time of entity increases which is considerable. I t is clear that for high quality of service for grid user range of connection band width should be 1000 to 10000. Exit time increase in case of ping request to the user_1. Hence it is quite clear that while pinging the user_1 quality of service decrease. So to improve the utilization of grid resource for same connection bandwidth range we should increase the MIPS rating of PEs.

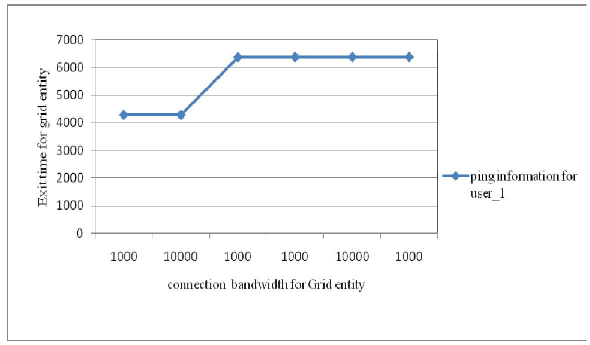

Fig. 2. Exit time versus connection bandwidth for Grid Entity associated with grid user_1

VIII. CONCLUSION

We discussed simulation based on object-oriented GridSim toolkit which is used for resource scheduling in time and space shared and modeling for grid resource, scheduling simulation. GridSim simulates time- and space-shared resources with different capabilities, time zones, and configurations. It supports different application models that can be mapped to resources for execution by developing simulated application schedulers. We have discussed the GridSim toolkit based simulation of Grid infrastructure. GridSim scales with them due to its concurrent implementation. Also, we were able to leverage the existing basic discrete-e vent infrastructure from SimJava while using the GridSim toolkit for sumulation. This helped us in evaluating performance and scalability of our scheduling policies with different Grid configuration such as varying the number of resources, capability, cost, users, and processing requirements i.e. we setup MIPS rating for grid main resource PEs. The results are promising and demonstrate the suitability of GridSim for developing simulators for scheduling in parallel and distributed systems. A better network model is required to support the application model with tasks collaborating and exchanging partial results among themselves in a P2P fashion. Gridlets are allocated to the PE following the time shared multitasking and multiprocessing event scheme. To enable simulation of Grid resource management and scheduling with economic models such as tenders and auctions [24]. We got the simulation results using scenario for time shared and space shared resources in Gridsim. In section VI and VII simulation results indicates that In grid network topology number of hops are same for different scenario but exit time of each grid entity is varying along with bandwidth of transmission link.

REFERENCES

[1] Foster I, Kesselman C (eds.). The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann: San Mateo, CA, 1999.

[2] Oram A (ed.). Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly, 2001.

[3] Baker M, Buyya R, Laforenza D. The Grid: International efforts in global computing. Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, Rome, Italy, 31 July–6 August 2000.

[4] Abramson D, Giddy J, Kotler L. High performance parametric modeling with Nimrod/G: Killer application for the global Grid? Proceedings International Parallel and Distributed Processing Symposium (IPDPS 2000), Cancun, Mexico, 1– 5 May 2000. IEEE Computer Society Press, 2000.

[5] Buyya R, Abramson D, Giddy J. Nimrod/G: An architecture for a resource management and scheduling system in a global computational Grid. Proceedings 4th International Conference and Exhibition on High Performance Computing in Asia- Pacific Region (HPC ASIA 2000), Beijing, China, 14–17 May 2000. IEEE Computer Society Press, 2000.

[6] Buyya R, Abramson D, Giddy J. An economy driven resource management architecture for global computational power Grids. Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), Las Vegas, NV, 26–29 June 2000. CSREA Press, 2000.

[7] Buyya R, Giddy J, Abramson D. An evaluation of economy-based resource trading and scheduling on computational power Grids for parameter sweep applications. Proceedings of the 2nd International Workshop on Active Middleware Services (AMS 2000), Pittsburgh, PA, 1 August 2000. Kluwer Academic Press, 2000 Copyright 2002 John Wiley & Sons, Ltd. Concurrency Computat.: Pract. Exper. 2002; 14:1175–1220 1220 R. BUYYA AND M. MURSHED

[8] Buyya R, Stockinger H, Giddy J, Abramson D. Economic models for management of resources in peer-to-peer and Grid computing. SPIE International Conference on Commercial Applications for High-Performance Computing, Denver, CO, 20–24 August 2001.

[9] BuyyaR. The World-Wide Grid.N http://www.csse.monash.edu.au/~rajkumar/ecogrid/wwg/.

[10] Weissman J. Grids in the classroom. IEEE Distributed Systems Online 2000; 1(3). http://www.computer.org/dsonline/archives/ds300/ds3edupr int.htm.

[11] Buyya R (ed.). High Performance Cluster Computing: Architectures and Systems, vol. 1. Prentice-Hall: Englewood Cliffs, NJ, 1999.

[12] CACI. Simscript: A Simulation Language for Building Large-scale, Complex Simulation Models. CACI Products Company: San Diego, CA. http://www.simscript.com/simscript.cfm.

[13] Bagrodia R, Meyer R, Takai M, Chen Y, Zeng X, Martin J, Park B, Song H. Parsec: A parallel simulation environment for complex systems. IEEE Computer 1998; 31(10).

[14] Howell F, McNab R. SimJava: A discrete event simulation package for Java with applications in computer systems modelling. Proceedings of the 1st International Conference on Web-based Modelling and Simulation, San Diego, CA. Society for Computer Simulation, 1998.

[15] Varga A. The OMNeT++ discrete event simulation system. Proceedings of the European Simulation Multiconference (ESM 2001), Prague, Czech Republic, 6–9 June 2001.

[16] Aida K, Takefusa A, Nakada H, Matsuoka S, Sekiguchi S, Nagashima U. Performance evaluation model for scheduling in a global computing system. The International Journal of High Performance Computing Applications 2000; 14(3).

[17] Song H, Liu X, Jakobsen D, Bhagwan R, Zhang X, Taura K, Chien A. The MicroGrid: A scientific tool for modeling computational Grids. Proceedings of IEEE Supercomputing (SC 2000), Dallas, TX, 4–10 November 2000.

[18] Casanova H. Simgrid: A toolkit for the simulation of application scheduling. Proceedings 1st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia, 15–18 May. IEEE Computer Society Press, 2001.

[19] Foster I, Kesselman C. Globus: A metacomputing infrastructure toolkit. International Journal of Supercomputer Applications 1997; 11(2):115–128.

[20] Aridor Y, Factor M, Teperman A. cJVM: A single system image of a JVM on a cluster. Proceedings 29th International Conference on Parallel Processing (ICPP 99),Fukushima, Japan, September 1999. IEEE Computer Society Press, 1999.

[21] Silva L, Buyya R. Parallel programming paradigms. High Performance Cluster Computing: Programming and Applications, vol. 2, ch. 2. Prentice-Hall: Englewood Cliffs, NJ, 1998.

[22] Buyya R, Murshed M. A deadline and budget constrained cost–time optimize algorithm for scheduling parameter sweep applications on the Grid. GridSim Toolkit Release Document, December 2001. http://www.buyya.com/gridsim.

[23] SPEC.CPU2000 Results. http://www.specbench.org/osg/cpu2000/results/cpu2000.ht ml [30 January 2002].

[24] Buyya R, Abramson D, Giddy J, Stockinger H. Economic models for resource management and scheduling in Grid computing. Concurrency and Computation: Practice and Experience 2002; 14(13–15):1507–1542. http://www.buyya.com/papers/emodelsgrid.pdf.

[25] Foundation for Intelligent Physical Agents (FIPA). Interaction and negotiation protocols. http://www.fipa.org/.