An Effective Selection Method for Scheduling of Gridlets among Heterogeneous Resources with Load Balancing on GridSim

Deepak Kumar Patel *
Dept. of Computer Science & Engineering
Veer Surendra Sai University of Technology
Burla, Odisha, India
patel.deepak42@gmail.com

Prof. Chita Ranjan Tripathy
Dept. of Computer Science & Engineering
Veer Surendra Sai University of Technology
Burla, Odisha, India
crt.vssut@yahoo.com

Abstract—**Since the GridSim toolkit is used to simulate the Grid computing environment, it has been widely used in the study of Grid. In this paper, the proposed work presents a load balancing scheme called LBLC for GridSim. Here, the load balancing is performed by providing an effective selection method for scheduling of Gridlets among heterogeneous resources with load consideration which maximizes the utilization of the resources and increases the efficiency of the Grid system. The performance is evaluated under various cases by using GridSim. From the simulation results, our load balancing scheme is shown to be quite efficient in maximizing the finished Gridlets and minimizing the execution time, unfinished Gridlets.**

Keywords— Selection Method, Scheduling, GridSim, Load Balancing

I. INTRODUCTION

GridSim [1] is a widely used Grid simulation tool. The GridSim toolkit provides a comprehensive facility for simulation of parallel and distributed computing systems such as Grids. GridSim is constructed on a java based simulation package known as SimJava [5]. The structure of Grid on GridSim is shown in Fig. 1. The Grid Broker is maintaining the overall scheduling of Gridlets on Gridresources and also the load balancing of all Gridresources present in the Grid system. The GridResource is maintaining the overall scheduling of Gridlets on machines and also the load balancing of its machines. Machine is maintaining the scheduling of Gridlets on processing elements (PEs) and also the load balancing of its PEs [2,3,11].

Improving the performance of a Grid system by an appropriate distribution of the workload among the resources is commonly known as load balancing [4]. The various load-balancing techniques proposed by many researchers for Grid, cannot be used in GridSim due to the structural differences of Grid. GridSim represents the tree based structure of a Grid. For a Grid, there exist several load balancing approaches, but the load balancing techniques on GridSim are classified under tree based load balancing approach of Grid. Therefore, there is a need to develop an effective and efficient load balancing scheme for GridSim.

The rest of the paper is organized as follows. The load balancing techniques on GridSim are presented in Section 2. In Section 3, the load balancing technique LBLC is proposed. In Section 4, the simulated results are presented. Finally, the paper is concluded in section 5.

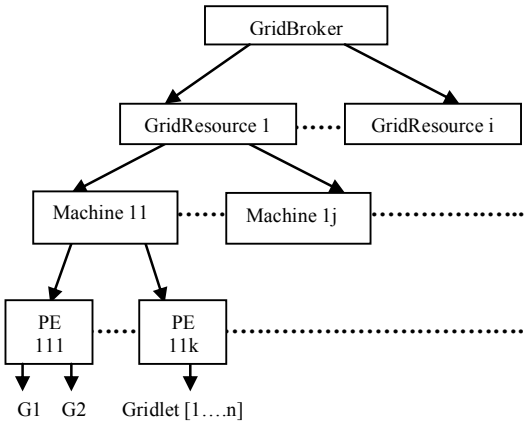

Fig. 1 The Grid Structure on GridSim

II. LOAD BALANCING TECHNIQUES ON GRIDSIM

The proposed load balancing techniques on GridSim include the WLB [2,3], LBEGS [9] and EGDC [10]. In WLB [2,3], the GridSim simulates the execution and allocation of Gridlets to the processing elements (PEs). In LBEGS [9], the load balancing is done for GridBroker, Gridresource and machine level of GridSim which is an improvement over WLB [2,3] and LBGS [6,7,8].

The load balancing mechanism EGDC [10] pays attention towards the deadline of tasks which is an improvement over WLB [2,3] and LBEGS [9]. Further, the EGDC [10] is found to be the best among all the stated schemes, it suffers from serious drawbacks. Here, for the scheduling of Gridlets among heterogeneous resources, there is no selection method used. There is no management for the scheduling of critical Gridlets to the best underloaded resources. Therefore, more Gridlets cannot be executed on these heterogeneous resources. Hence, it needs further improvement.

* *Corresponding Author*

978-1-5386-2529-3/17 $31.00 © 2017 IEEE
DOI 10.1109/CINE.2017.17

In this paper, a load balancing scheme called LBLC is proposed for GridSim. Here, our priority is always to assign the most critical Gridlets to the best underloaded resources. Therefore, we perform load balancing by providing an effective selection method for scheduling of Gridlets among heterogeneous resources with load consideration which maximizes the utilization of the resources and increases the efficiency of the Grid system. The performance is evaluated under various cases by using GridSim. From the simulation results, our load balancing scheme is shown to be quite efficient in maximizing the finished Gridlets and minimizing the execution time, unfinished Gridlets.

III. THE POPOSED WORK: LOAD BALANCING ON ENHANCED GRIDSIM WITH LOAD CONSIDERATION (LBLC)

A: Notations and Terminologies

Table 1: Notations

Notation	Definition
RL	resource load
ML	machine load
PEL	PE load
PEC	PE capacity
MC	machine capacity
RC	resource capacity
PE	no. of PEs on a machine
M	no. of machines on a resource
GL	Gridlet length
GD	Gridlet's deadline
rt	resource threshold
rb	machine threshold
UResourcelist	underloaded resources's list
OResourcelist	overloaded resources's list
NResourcelist	normallyloaded resources's list
Gridletlist	Waiting Gridlets's list for execution

B: Proposed Algorithm

Begin

*/*Estimate the PE load*/*
PEL = GL / (GD * PEC);

*/*Estimate the machine load*/*
MC = PEC * PE;
ML = GL / (GD * MC);
ML = GL / (GD * PEC * PE);
ML = PEL * PE;

*/*Estimate the resource load*/*
RC = PEC * PE * M;
RL = GL / (GD * RC);
RL = GL / (GD * PEC * PE * M);
RL = PEL * PE * M;
RL = ML * M;

for all resources do
calculate RL;
if (RL < rb)
submit this underloaded resource to UResourcelist;
else if (RL > rt)
submit this overloaded resource to OResourcelist;
else
submit this normallyloaded resource to NResourcelist;
end for

While (OResourcelist and UResourcelist is not empty)
Insert the Gridlets's list into Gridletlist which is need to move from OResourcelist to UResourcelist;
end while

Sort the Gridlets of the Gridletlist according to the Gridlet's largest size;

Sort the underloaded resources of the UResourcelist in ascending order of load;

while (UResourcelist and Gridletlist is not empty)
for every Gridlet g in Gridletlist
for every resource r in Resourcelist
factor1 = RL + ((GL) / (GD *PEC*PE*M));
if (factor1 <= rb)
Gridlet g is executed on resource r;
elseif (rb <= factor1 <= rt)
Gridlet g is executed on resource r;
else
Execute the Gridlet g on the next underloaded resource of UResourcelist;
end if
end for
end for
end while

End

C: Description

In GridSim, when a Gridlet is assigned to one of the resources by the GridBroker, then the Gridresource submits that Gridlet to one of its machine and the machine submits that Gridlet to one of its PE for execution. To estimate a proper load for a Gridresource, we should estimate the load of the machines and PEs present under that Gridresource. Therefore, our proposed scheme shows the load estimation method for

every levels of GridSim. To check all the resource states, the GridBroker estimates the load of all resources.

Then, the GridBroker checks all the recent resource states. If the resource load is greater than rt, then the resource state is overloaded and if the resource load is less than rb, then the resource state is underloaded. If the resource load is present in between rt to rb, then the resource state is normally loaded. The Grid Broker inserts the underloaded resources in a queue known as UResourcelist.

If a resource state is overloaded, then the GridBroker has to transfer some Gridlets from that overloaded resource and assign that Gridlets to one of the underloaded resources. The unfinished Gridlets which need to transfer from all the overloaded resources are stored in a queue known as Gridletlist for subsequent scheduling.

Now, the Question is, which Gridlet from the Gridletlist will be assigned by the Grid Broker on which underloaded resources present in the UResourcelist. Our priority is always to assign the most critical Gridlets to the best underloaded resources. Here, we provide an effective selection method for scheduling of Gridlets among heterogeneous resources with deadline consideration so that more Gridlets will be executed with in the assigned deadline.

To develop an effective selection method for scheduling, the GridBroker has to deal with two issues known as Gridlet ordering and resource ordering. Gridlet ordering refers to the order in which all Gridlets will be assigned to resources and resource ordering refers to the order of resources on which each Gridlet will be assigned so that the best underloaded resources can execute the most critical Gridlets within their assigned deadline.

In Gridlet ordering, the GridBroker uses a selection method in which there is one arrangement for Gridlets with the consideration of Gridlet heterogeneity. All the Gridlets present in the Gridletlist should be executed on the underloaded resources by preventing the resource overloading. Therefore, always the priority should be assigning the most critical Gridlets means the Gridlet of largest size to one of the best underloaded resources for scheduling. Here, the GridBroker sorts the Gridlets of the Gridletlist according to the Gridlet's largest size.

For resource ordering, the GridBroker uses a selection method in which there is one arrangement for the underloaded resources according to their state of load. All the underloded resources present in the UResourcelist should execute more number of Gridlets by preventing the resource overloading. Here, the GridBroker sorts the underloaded resources present in the UResourcelist in ascending order of load. So, the underloaded resource having the least amount of load gets the first chance to execute the critical Gridlets having largest size. When a underloaded resource is sorted, the GridBroker also sorts the underloaded machines present in an underloaded resource and the underloaded PEs present in an underloaded machine in ascending order of load.

Now, the assigning of the most critical Gridlets from the Gridletlist to the best underloaded resources of UResourcelist is done using the scheduling mechanism.

IV. SIMULATION

A. *Simulation Parameters*

Table 2: Parameter Values

Parameter	Values
Machines present in each resource	2
PEs present in each Machine	5
Resource threshold	0.8
Machine threshold	0.75
PE threshold	0.6
Gridlet Deadline	1-6 seconds
Network Bandwidth	0.5-10 Mbps
Gridlet Length	1-5 Million Instructions
Rating of PE	1-5 Million Instructions per Second

B. *By taking constant no. of resources*

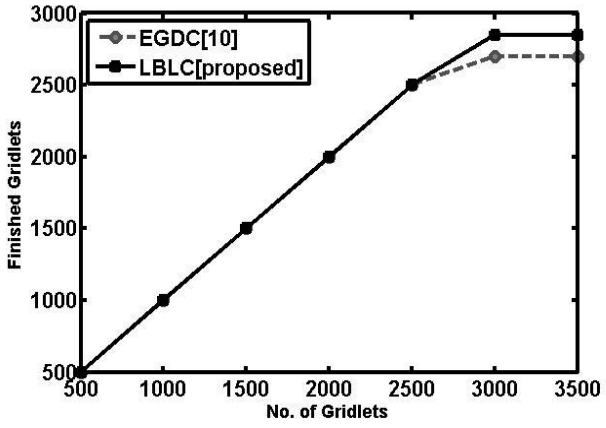

Fig. 2 Finished Gridlets Vs. No. of Gridlets

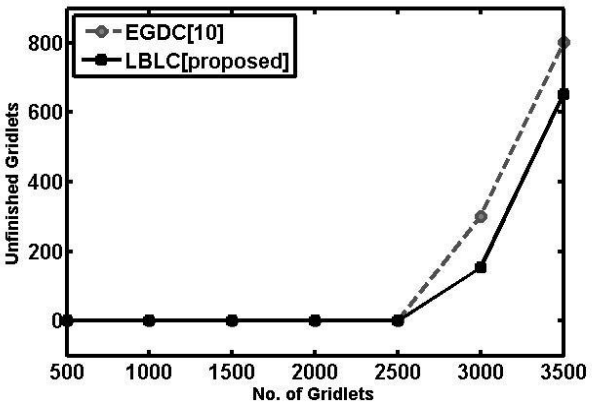

Fig. 3 Unfinished Gridlets Vs. No. of Gridlets

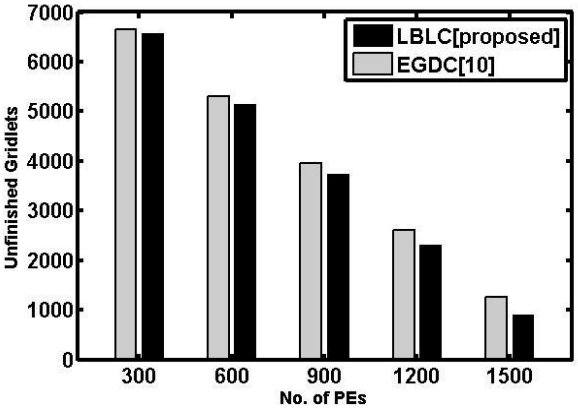

Fig. 6 Unfinished Gridlets Vs. No. of PEs

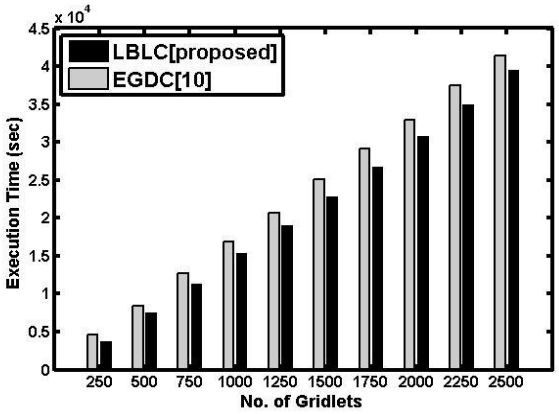

Fig. 4 Execution Time Vs. No. of Gridlets

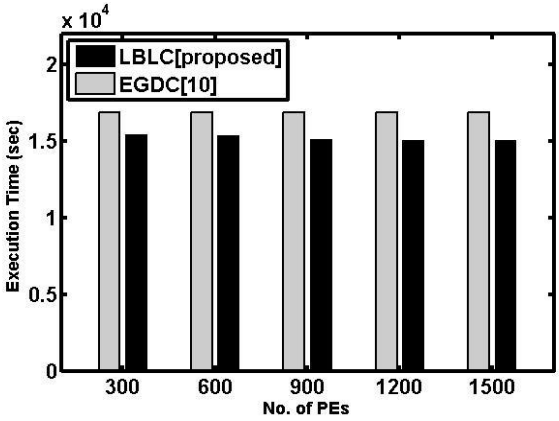

Fig. 7 Execution Time Vs. No. of PEs

C. By taking constant no. of Gridlets

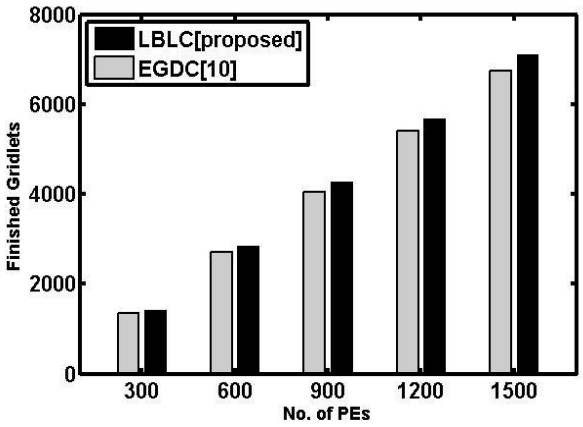

Fig. 5 Finished Gridlets Vs. No. of PEs

Table 3: Simulation Results Summary

Cases	Figures	Performance Metrics	LBLC [proposed] as compared to EGDC [10]
By taking constant number of resources	Fig. 2	Finished Gridlets	Increases
	Fig. 3	Unfinished Gridlets	Decreases
	Fig. 4	Execution Time	Decreases
By taking constant number of Gridlets	Fig. 5	Finished Gridlets	Increases
	Fig. 6	Unfinished Gridlets	Decreases
	Fig. 7	Execution Time	Decreases

V. CONCLUSION

In this paper, the proposed work presents a load balancing scheme called LBLC for GridSim in which we do load balancing by proposing an effective selection method for scheduling of Gridlets among heterogeneous resources with load consideration which maximizes the utilization of the resources and increases the efficiency of the Grid system. We can extend our proposed load balancing scheme by implementing a fault tolerant scheduling system on GridSim.

REFERENCES

[1] http://www.buyya.com/GridSim/

[2] R. Buyya, M. Murshed, GridSim: a toolkit for the modeling and simulation of distributed management and scheduling for Grid computing, The Journal of Concurrency and Computation: Practice and Experience 14 (2002) 13–15.

[3] M. Murshed, R. Buyya, D. Abramson, GridSim: A toolkit for the modeling and simulation of global grids. Technical Report (2001), Monash, CSSE

[4] S. Penmatsa, A.T. Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel Distributed Computing 71 (2011) 537-555.

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

[6] B. Yagoubi, Y. Slimani, Task Load balancing strategy in grid environment, Journal of Computer Science 3 (3) (2007) 186–194.

[7] B. Yagoubi, Y. Slimani, Load balancing strategy in grid environment, Journal of Information Technology Applications 4 (2007) 285–296.

[8] B. Yagoubi, Y. Slimani, Dynamic load balancing strategy for grid computing, Engineering and Technology (2006) 90–95.

[9] K. Qureshi, A. Rehman, P. Manuel, Enhanced GridSim architecture with load balancing, The Journal of Supercomputing (2010) 1–11.

[10] Y. Hao, G. Liu, Na Wen, An enhanced load balancing mechanism based on deadline control on GridSim. Future Generation Computer Systems 28 (2012) 657-665.

[11] A. Sulistio, G. Poduval, R. Buyya, C.K. Tham, On incorporating differentiated levels of network service into GridSim, Future Generation Computer Systems 23 (2007) 606–615.