Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.243 – 250

RESEARCH ARTICLE

Design and Implementation of Job Scheduling in Grid Environment over IPv6

Akshay A. Bhoyar[1]

M.Tech-Second year, Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India

Akshaybhoyar21@gmail.com

Prof. R. C. Dharmik[2]

Asst. Prof Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India

raj_dharmik@yahoo.com

Abstract: ***In this paper we are going to design and implement the job scheduling algorithm for Grid environment using IPv6. We are establishing a cross regional scheduling mechanism to accomplish the job scheduling and job management, that chooses the operating location and mode automatically through the scheduling system and users request, due to which in grid computing the resources of will be used more efficiently and more reasonably The idea is that Grid technology managing this large and heterogeneous environment will allow an easy access to its resources for various users, by means of allowing them to submit their jobs into the system, guaranteeing them nontrivial Quality of Service (QoS) while hiding the complexity of the system itself by providing powerful but simple interfaces for the end user of the Grid. A grid environment is of two types: Data grids and Computing grids. Load Balancing is a technique in which the workload is distributed equally across multiple computers so that resource utilization is enhanced and the response time in grid environment get reduced. Main goal of load balancing is balancing load across all the processors which improves the throughput of grid resources. A good Scheduling algorithm works as it should balance the system load and assign jobs to resources efficiently. Hierarchical Load Balanced Algorithm is used to solve the problem in grid environment. The proposed system Enhanced Hierarchical Load Balance Algorithm is designed to schedule the jobs and also to improve the overall performance of the system in terms of resource utilization and user satisfaction. We will be using First Come First Serve(FCFS) approach so as to achieve the most efficient and optimized solution for our problem definition.***

Keywords: Grid computing, IPv6, Open Message Passing, Message Passing Interface

I. INTRODUCTION

A grid is defined as a group of nodes that are connected to each other through a number of connecting paths for communication. Grid computing combines computers from multiple heterogeneous domains to solve a single task. Grid is a type of parallel and distributed system that allows the aggregation, sharing and selection of geographically distributed resources dynamically depending on their availability, user QoS requirement, performance, capability, and cost during run time.

Grid computing is a type of Distributed Computing in which a complex problem is to be solved by group of independent computer systems. Big task is divided into number of sub tasks and allocating to each resource that solves a part of solution and then combine the result from all computers. These loosely coupled systems together working for a common goal. Grid computing allows a cluster of loosely coupled computers to perform large tasks or tasks that generally consume more resources and time than is feasible for a single system. This is a parallel computing technology in which the computers involved, called "nodes", are almost completely independent of each other in terms of resources, timing and synchronization aspects. These nodes are more heterogeneous than those present in other distributed systems. The nodes involved in the process are voluntary, and the interaction and mediation between the nodes of the grid are performed through a management interface. All transactions in the grid are performed through the management interface. The growth in the amount of data to be computed, computational problems are getting more and more complex and costly in terms of time needed. Grid computing can integrate and utilize heterogeneous computing resources which are connected through networks without the limitation of geography. Thus grid computing is widely used to solve large-scale computational problems. Unlike traditional cluster computing, computational capabilities of resources in grid computing environments are usually different. In order to fulfil the user expectations in terms of performance and efficiency, the Grid system needs efficient load balancing algorithms for the distribution of tasks. A load balancing algorithm attempts to improve the response time of user's submitted applications by ensuring maximal utilization of available resources.

Need of Grid Computing:

Grid enable the sharing, aggregation and selection, of a various types of resources including storage systems , supercomputers, data sources, and specialized devices that are geographically distributed and owned by different organizations for solving large-scale computational and data intensive problems in science, engineering, and commerce[4].

Grid applications (typically multi-disciplinary and large-scale processing applications) that may be globally located for other practical reasons, or often couple resources that cannot be replicated at a single site. These are some of the driving forces behind the foundation of global Grids. In this light, the Grid allows users to solve larger-scale problems by gathering together resources that could not be easily coupled before. Hence, the Grid is not only a computing infrastructure, but also a technology that can bond and unify remote and diverse distributed resources ranging from meteorological sensors to data vaults and from parallel supercomputers to personal digital organizers. As such, it will 10 provide pervasive services to all users that need them.

Load balancing is an important issue that is recently studied very much because of need of high performance computing. The computation grid involves high performance platforms to heterogeneous dynamic and shared environment. Thus how to adapt load balancing schemes for Grid becomes the focus of this research area. For parallel applications, load balancing attempts to distribute the computation load across multiple processors or machines as evenly as possible with objective to improve performance.

A load balancing scheme consisting of three phases: information collection, decision making and data migration. During the information collection phase, load balancer gathers the information of workload distribution and the state of computing environment and detects whether there is a load imbalance. The decision making phase focuses on calculating an optimal data distribution, while the data migration phase transfers the excess amount of workload from overloaded processors to under loaded ones.

The rest of paper is organized as follows: Section 2 focuses on related work; Section 3 discusses the proposed algorithm with results; Section 4 we are coming up with future scope.

II. LITERATURE SURVEY

Jun Chen,You Zou, Zhongyu Liu,Qing Wu [1] says that in grid computing the biggest problem that is data transmission , communication efficiency and stability is solved. Grid Computing composed of small local area network(LAN), thus the problem can be solved by properly building LAN with the use of IPv4 network or high-speed computing network such as Infiniband. However, during the process of distribution of computer resources in different physical regions, data transmission and communication can be affected factors like time period and load. So the speed of cross-regional data transmission is not optimistic and the stability of transmission cannot be guaranteed. While using IPv4 network to accomplish the resources sharing process, there will be two difficulties: One is the limitation of address space, as the current IPv4 address is extremely rare, so it is impossible to allocate an independent IPv4 address for all nodes in the grid,; secondly the instability of transmission speed of IPv4 on which most of the current network applications are based which create full load condition on IPv4 network.

According to authors to solve the problem related to load, data transmission and instability in grid computing, they suggested to make use of IPv6 instead of IPv4.The address capacity of IPv6 can be upto 2^128 which is enough for all computing nodes that is to be allocated to an independent IPv6 address in grid; with this the other advantage of IPv6 is, load on IPv6 is lower as compared to IPv4 network, so the data transmission and exchange are more efficient.

The author proves speed test of IPv6 in which Three files in different sizes are used to test the transmission speed.

The author experiences the transmission period of IPv6 network and can use its high-speed bandwidth to accomplish some applications; they also introduces the advantage of IPv6 by testing the transmission speed.

Location	**50MB**		**150MB**		**200MB**	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
1	464.8	686.6	763.3	684.4	482.6	667.6
2	1336.2	1877.4	568.3	906.3	700.8	1365.8
3	693.6	1797.4	666.9	1927.6	673.6	1950.9
4	1985.9	4205.1	2230.6	4076.3	2158.7	4117.4
5	2068.5	4082.3	1238.9	4228.5	2063.6	4214.8

Table I: Data Transmission Results(KB/s)of IPv4 and IPv6

R L Warrender, J Tindle, D Nelson discusses some of the technical challenges encountered when attempting to use software intended for workstation use within a semi-automatic batch cluster (HPC) environment. High performance clusters (HPC) are constructed typically from a collection of computer boxes (or nodes) interconnected by a high speed, high capacity TCP/IP network, each box containing multiple processors (or cores). Assuming the software has been designed using the most efficient algorithms and parallel techniques, it follows that the highest output of useful work will be obtained if we can ensure every core in every node is running continually at 100% utilization for the entire duration of the batch run. These paper include:

Use of Symmetric Multiprocessing (SMP).

Use of Message Passing Interface (MPI) or equivalent.

Use of graphic processor units.

Use of a scheduler.

Feeding files to and from nodes for processing.

Assessment of batch results.

The writers have shown that by concentrating on job efficiency rather than absolute time to execute individual jobs, the time taken to process a batch of Gaussian09 jobs is reduced[2].

Jagdish Chandra Patni,Dr. M.S.Aswal, Om Prakash Pal,Ashish Gupta discuss about the load balancing strategies that control the load of the entire system. Load balancing algorithms in classical distributed systems, which usually run on homogeneous and dedicated resources. The structure of the Grid comprises characteristics of homogeneous as well as heterogeneous systems, loosely coupled as well as tightly coupled systems. Load balancing strategies aim to adapt the load optimally to the environment. The author conclude that, they addressed the problem of load balancing in large scale distributed systems. They study various load balancing strategies based on a tree representation of a Grid. They defined a hierarchical load balancing strategy that privileges local balancing in first (load balance within groups without communication between groups). After load balancing at group level (if load is not balanced at group level) it will be balanced at region level and after balancing at region level (if load is not balanced at region level) it will be balanced at grid level.

The static load balancing involves partitioning into subdomains. The subdomains can then be distributed over the processors and in parallel the calculation is carried out. Different partitions may result in different times to completion for the calculation. So it is necessary to examine the quality of the partitioning based on its effect on the application code.
Dynamic Load Balancing (DLB) provides application level load balancing for individual parallel jobs.
It decides that the load submitted by the DLB environment are distributed such that the overall load of system is balanced and the maximum benefit of available resources should br gain by the application programs.[3]

III. PROPOSED SYSTEM AND RESULTS

In order fulfill the user requirements related to performance and efficiency, for the proper the distribution of tasks in Grid system needs efficient load balancing algorithms. The algorithm attempts to improve the response time of user's submitted requirement by utilizing maximum available resources.

Grid set up on LAN

There are several ways through out of which grid set up is developed. Many of the organization are having their own created grid and cluster set up. Before implementation there is need of that environment to implement the algorithm and generate the results. As stated in previous chapter, this project doesn't uses gridsim toolkit all the results has been calculated on real time set up. Here we are focusing on Grid Environment structure. In Grid Environment the machines which are connected by LAN can communicate with each other, which involves control server, 5 server and 4 clients shown in fig1. So, our main objective is to schedule the tasks to the server depending on load on server and improve the performance by reducing the process execution time. To show the efficiency of our proposed load balancing algorithm, several parameters have been considered [4], which the following are:-

Avg. CPU Utilization: Percentage of CPU utilization involved at the time of job execution.

Available Heap Memory: Available heap memory used to execute the particular job.

Maximum Heap memory: Maximum heap memory availability in the main memory.

CPU Idle time: The current CPU idle time of during execution.

Available Processor: For particular job execution, list of processors available at that instance.

Node id: Indicates the unique identification of each local and remote node.

N: {n1, n2…} is the total number of local and remote nodes.

J: {j1, j2…..}

$\sum_{i=1}^{m} \boldsymbol{Exe\ Ti}$: Total execution time of all jobs.

3.1 Grid Environment setup

The fig1 depicts set up for grid environment. It shows that there are total 10 nodes in the grid environment. The IP address that used is real time addresses in LAN.

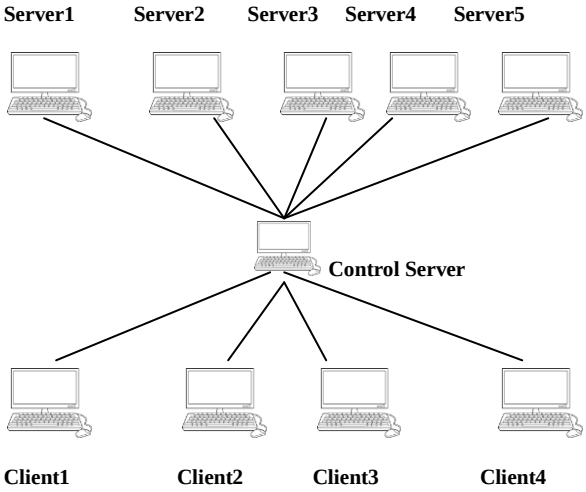

Fig.1 Grid Environment setup

In grid computing environment, a number of machines are connected to each other as workstations as shown in fig1.3. It is a structure which is grouped together with the help of LAN. The Grid consist of 1 control server, 5 servers and 4 clients. Each and every node in the structure will communicate with the LAN connectivity.

The control server is the heart of the Grid environment, to which the servers and the clients are connected. The client sends the request to the control server for resource utilization ; the control server thereafter checks the load on the server and forward the clients request to the server having minimum load along with the execution time required by server to complete the required request.

A Grid structure can have more number of servers and clinets i.e. nodes that can be formed by joining many numbers of computing nodes. This can be implemented in two possible ways. One is, to create the hierarchy logically on one machine by using Grid Simulator or any middlewar. Second is, to implement it physically on LAN by using as many nodes as possible.

3.2 Role of Control Server Nodes

At the time of job allocation, control server node distributes the job to total number of server nodes. The job is allocated to the cluster nodes with the available resources. At the same time the job is get divided and distributed to nodes. This job distribution approach is based on multiple different parameters of efficient load distribution. The vital feature of distribution is based on the amount of available resources like CPU idleness, laod on server , heap memory.

3.3 Proposed Algorithm

Step 1: Receive the request from clients.

Step 2: Identify the server having minimum load.

Step 4: Redirect the request on that server using IP address.

Step 5: Update the load for that server accordingly.

Step 5: For every request, allocate the load amongst all the servers in such a way that no server should be overloaded.

Step 6: Once request is served to the server it informs about the execution time required by that server to complete the clients request.

Step 7: For every request served to the server its respective file contents are modified.

3.3.1 Parameters in the Algorithm

However, to satisfy end user expectations with improved performance and efficiency, the Grid computing environment require standard scheduling algorithms for task distribution. A proper scheduling algorithm tries to reduce the response time of end user's submitted jobs by maximizing the utilization of the available resources. Here focus is on grid organizational structure. However, In set up servers at the same level can communicate directly with the control server [11]. Many of the current Grid systems use such type of organizational structure because it is highly scalable. Hence main aim is to improve the performance of the environment by reducing process execution time. To check how efficient our proposed Load balancing algorithm, different parameters are under monitoring [2] which are:-

Idle Server node : Total no. of available server nodes to allocate the job.

Overload Server node: List of overloaded server nodes in the complete set up.

IP Address: Unique identification of each node.

N: Total number of nodes.

J: No. of Jobs.

$\sum_{i=1}^{n} \boldsymbol{Exe\ Ti}$: Total time for execution of all jobs.

On all these parameters load balancing algorithm can be implemented and the results are calculated at the end by adding total latency against the query processing time which is at the shared database.

Result:

With reference to contents included in the paper there is need of a grid interface for an end user to make a request to the control server. For this, a homepage is developed, which provide the end user to create user_id and password. Through this page no. of clients can make request to Control server node. Here main focus is on load balancing there are some of the tests that are included in this section. For this there is need to keep the record of load of all the servers at grid/control server side. Hence a database file or table is used in this project to keep a track of all the servers.

The fig2 shows the login page of the grid environment system which is used by the end users to make request to the grid node.

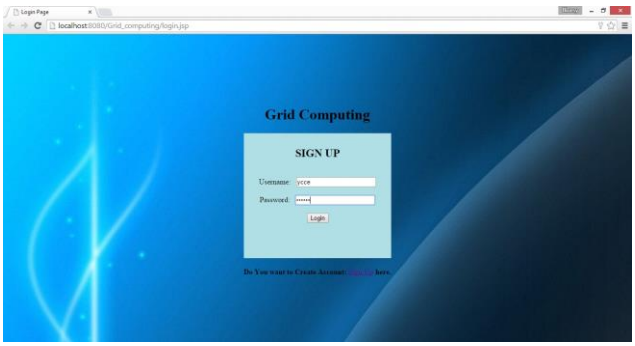

Fig2. Login Page

The client login by entering the user_id and password to access the services ; if the client is new they can can click on sign up option to create new username and password.

Initially load files of all the servers which are at control server are set to zero. As there are total five servers so load on each server is set to zero. Fig3 shows the snapshot of Server Load.

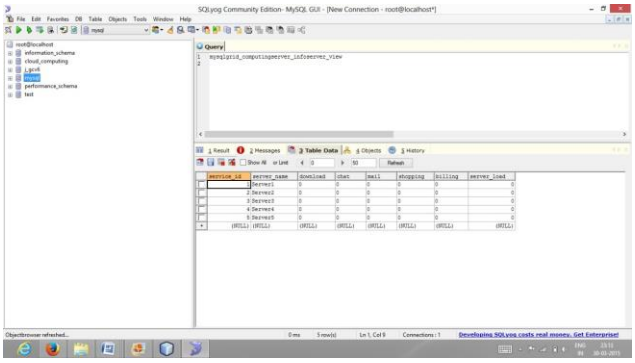

Fig3. Snapshot of Server Load

The client are provided with the services in fig4 after login process.

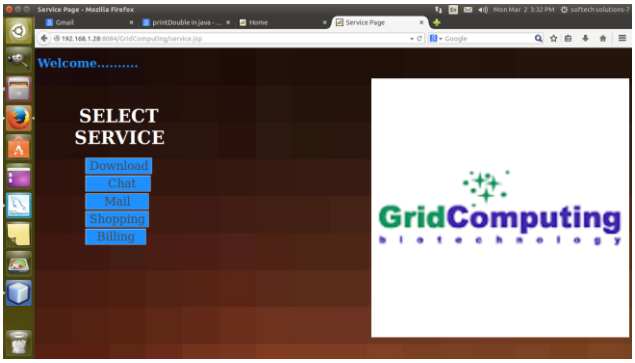

Fig4. Snapshot of Services

As the client select the download service , it get allotted to server1, which is shown in url in fig5. The download services allocated to server1 because the load on server1 is 0 and along with that the execution time for the further service is also displayed as soon as the service is selected.

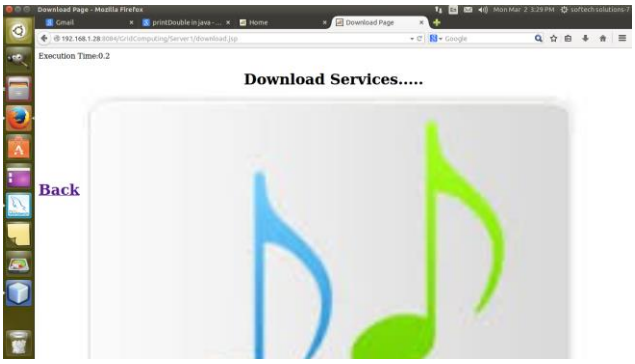

Fig 5 Snapshot of Download service allotted to server1

Fig 6 shows that when the same client or the another client again send the request for the download service to the control server node ; it allocates the load to server2 as the same service is allocated to server1

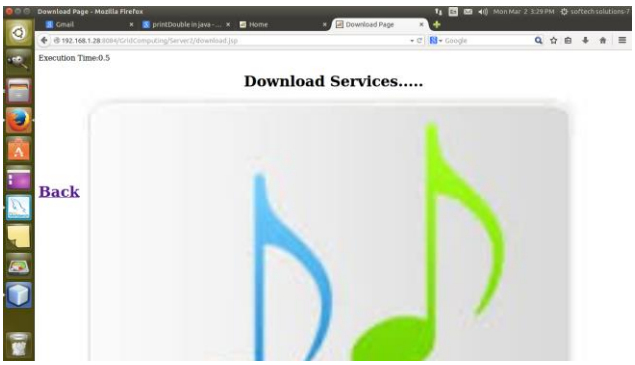

Fig6. Snapshot of Download service allotted to server2

Scheduling the same service i.e. download service to all the server the load on each server and its execution time along with the ip address is modifies in the table shown in fig7.

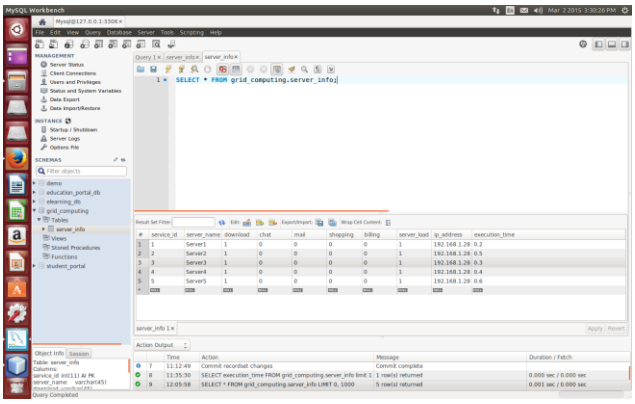

Fig7. Load on server after clients request

IV. CONCLUSION

Paper contains the final results for load balancing and job scheduling in grid environment over IPv6. So, with reference to the results from the above section it is concluded that, there exist two different types of scheduling one is called as static scheduling and other is dynamic scheduling. Dynamic scheduling is more versatile than static scheduling. And grid environment based on LAN is more flexible because it is very easy to add and delete no. of nodes at any instance without disturbing grid set up. Thus we have established a cross regional scheduling mechanism based on IPv6 to accomplish the job scheduling and job management, so that the resources of grid computing will be used more reasonably and more efficiently in grid environment.

REFERENCES

[1] Jun Chen,You Zou, Zhongyu Liu,Qing Wu:" *Enabling Grid Computing over IPv6 within a Campus Network"*,2011 Fourth International Symposium on Parallel Architectures, Algorithms and Programming.
[2] R L Warrender, J Tindle, D Nelson2011 IEEE:" *Job Scheduling in a High Performance Computing Environment*", 2013 IEEE.
[3] Jagdish Chandra Patni,Dr. M.S.Aswal,Om Prakash Pal,Ashish Gupta:"*Load balancing Strategies for Grid Computing*", 2011 IEEE.
[4] Rency rajan, G.K.Kamalam*:" Priority Based Heuristic Job Scheduling Algorithm For The Computational Grid"* 2012 Fourth International Symposium
[5] R.Manimala,P.Suresh:" *Load Balanced Job Scheduling Approach for Grid Environment*",2011 IEEE.
[6] Xu Yan, "Task Scheduling Algorithm Research in Grid Computing",First National Conference for Engineering Sciences, 2012.
[7] S.K.Karthikumar and M.Udhaya Preethi and P.Chitra, "Fair Scheduling Approach For Load Balancing and Fault Tolerant in Grid Environment", IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology, 2013.
[8] Janhavi B and Sunil Surve and Sapna Prabhu, "Comparison of load balancing algorithms in a Grid",International Conference on Data Storage and Data Engineering, 2010.
[9] R.Venkatesan and Mrs. K. Ramalakshmi and Ms. Arati Patro and Dr. K. Thanushkodi, "Scheduling Framework with Resource Level Load Balance Using Agents in Grid Computing Environments", Akshaya College of Engineering, Coimbatore, Tamilandu, India, 2011.
[10] Neeraj Rathore and Dr. Inderveer Chana, "A Cognitive Analysis of Load Balancing and job migration Technique in Grid", World Congress on Information and Communication Technologies, 2011.
[11] K. Hemant Kumar Reddy and Diptendu Shina Roy, "*A Hierarchical Load Balancing Algorithm for Efficient Job Scheduling in a Computational Grid Test bed*", 1st Int'l Conf. on Recent Advances in Information Technology RAIT, 2012.
[12] S. Friedrich, S. Krahmer, L. Schneidenbach, and B. Schnor, "*Loaded: Server load balancing for ipv6*," in Networking and Services, 2006. ICNS '06. International conference on, july 2006, p. 8.
[13] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, "*Improving mapreduce performance in heterogeneous environments*," in Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 29–42. 1855744.
[14] A. Knoth, C. Kauhaus, D. Fey, L. Schneidenbach, and B. Schnor, "Challenges of mpi over ipv6," *Networking and Services, International conference on*, vol. 0, pp. 242–247, 2008.
[15] H. Li, X. Zhang, and J. Fan, "The safety analysis of ipsec based on ipv6 protocal," *in Cir- cuits,Communications and System (PACCS)*, 2010 Sec- ond Pacific-Asia
[16]Naidila Sadashiv and S. M Dilip Kumar, "*Cluster, Grid and Cloud Computing: A Detailed Comparison*", the 6th International Conference on Computer Science and Education (ICCSE) August 3-5, 2011.
[17] Lalitha Hima Bindu.P. and Venkatesan R. and Ramalakshmi.K, "*Perspective Study on Resource level Load balancing in Grid Computing Environments*", IEEE, 2011.
[18] Augusto Mendes Gomes Júnior and Liria Matsumoto Sato and Francisco Isidro Massetto, "*A parallel application programming and processing environment proposal for grid Computing*", IEEE 15th International Conference on Computational Science and Engineering, 2012.
[19] Rajkumar Buyya and Manzur Murshed; GridSim, "*A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*", Australia, 2003.
[20] https://www.digitalocean.com/community/articles/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu
[21] http://www.openmp.org
[22] http://mitpress.mit.edu/0262533022

