ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

Design of Decentralized Load Balancing Algorithm for Cloud Environment

Sarika Vasantrao Bodake[1], Dr. Radhakrishna Naik[2]

ME CSE Student, CSE Department, MIT, Aurangabad, India[1]

Professor, CSE Department, MIT, Aurangabad, India[2]

Abstract: Cloud computing is that the next generation of computation. Probably people will have everything they have on the cloud. Cloud computing provides resources to shopper on demand. The resources are also software package resources or hardware resources. Cloud computing architectures square measure distributed, parallel and serves the requirements of multiple purchasers in numerous situations. This distributed design deploys resources distributive to deliver services expeditiously to users in numerous geographical channels. Purchasers in a very distributed setting generate request haphazardly in any processor. Therefore the major disadvantage of this randomness is related to task assignment. The unequal task assignment to the processor creates imbalance i.e., a number of the processors square measure over laden and a few of them square measure beneath loaded. The target of load balancing is to transfer the load from over laden method to beneath loaded method transparently. Load balancing is one in all the central problems in cloud computing. To realize high performance, minimum interval and high resource utilization magnitude relation we want to transfer the tasks between nodes in cloud network. Load balancing technique is employed to distribute tasks from over loaded nodes to beneath loaded or idle nodes. In following sections we tend to square measure discuss concerning cloud computing, load balancing techniques and also the planned work of our load balancing system.Proposed load balancing algorithm is simulated on Cloud Analyst toolkit. Performance is analyzed on the parameters of overall response time, data transfer, average data center servicing time and total cost of usage. Results are compared with three existing load balancing algorithms namely Round Robin, Equally Spread Current Execution Load, and Throttled. Results on the basis of case studies performed shows more data transfer with minimum response time.

Keywords: Cloud Computing; Load balancing; Load balancing Algorithms; IaaS.

I. CLOUD COMPUTING

There is no correct definition for cloud computing, we will say that cloud computing is assortment of distributed servers that provides services on demand [8]. The services are also software package or hardware resources as shopper would like. Primarily cloud computing have 3 major parts [9]. Initial is shopper; the tip user interacts with shopper to avail the services of cloud. The shopper is also mobile devices, skinny purchasers or thick purchasers. Second element is information centre; this can be assortment of servers hosting totally different applications. This might exist at an oversized distance from the purchasers. Currently days a thought known as virtualization [6] [7] is employed to put in software package that permits multiple instances of virtual server applications. The third element of cloud is distributed servers; these square measure the elements of a cloud that square measure gift throughout the web hosting totally different applications. However as exploitation the applying from the cloud, the user can feel that he's exploitation this application from its own machine.

Cloud computing provides 3 varieties [5] of services as software package as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). SaaS provides software package to shopper which require to not installing on purchasers machine. PaaS provides platform to make associate degree applications like information. IaaS provides procedure power to user to execute task from another node.

II. LOAD BALANCING

In cloud system it's attainable that some nodes to be heavily loaded and alternative are gently loaded [9]. This example will result in poor performance. The goal of load balancing is distribute the load among nodes in cloud setting. Load balancing is one in all the central problems in cloud computing [6].

For higher resource utilization, it's fascinating for the load within the cloud system to be balanced [9] equally. Thus, a load balancing formula [1] tries to balance the whole system load by transparently transferring the employment from heavily loaded nodes to gently loaded nodes in a shot to make sure smart overall performance relative to some specific metric of system performance. Once considering performance from the purpose of read, the metric concerned is usually the interval of the processes. However, once performance is taken into account from the resource purpose of read, the metric concerned is total system outturn [3]. I n distinction to interval [2], outturn cares with seeing that every one users square measure treated fairly which all square measure creating progress.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

To improve the performance of the system and high resource allocation magnitude relation we want load balancing mechanism in cloud. The characteristics of load balancing square measure [1] [5]:

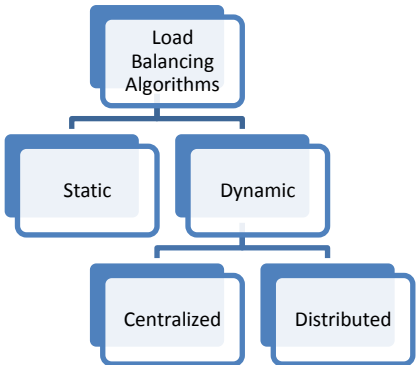
- Distribute load equally across all the nodes.
- To realize a high user satisfaction.
- Improving the performance of the system.
- To scale back interval.
- To reach resource utilization magnitude relation.

Let us take associate degree example for higher than sited characteristics:

Suppose we've got developed one application and deploy it on cloud. Mean whereas this application is extremely common. Thousands of individuals square measure exploitation our application. Suppose many users exploitation this application at constant time from single machine and that we didn't apply load balancing approach to our application. This point the actual server is extremely busy to execute the user's tasks and alternative server's square measure gently loaded or idle. The users didn't satisfy as a result of low response and performance of the system.

If we tend to apply load balancing on our application, we are able to distribute some user's tasks to alternative nodes and that we will get the high performance and quicker interval. During this method we will reach higher than characteristics of load balancing.

A. Taxonomy of Load-Balancing Algorithms



There square measure main 2 classes of load balancing [3] [4]. They're

i) Static load balancing and ii) Dynamic load balancing.

Static algorithms works statically and don't contemplate the present state of nodes. Dynamic algorithms [4] work on current state of node and distributes load among the nodes. Static algorithms use solely info concerning the common behavior of the system, ignoring the present state of system. On the opposite hand, dynamic algorithms react to the system state that changes dynamically.

Static load balancing [4] algorithms square measure less complicated as a result of there's no got to maintain and method system state info. However, the potential of static formula is proscribed by the very fact that they are doing not react to the present system state. The attraction of dynamic algorithms that they square measure doing reply to system state therefore are higher able to avoid those states with unnecessarily poor performance. Attributable to this reason, dynamic policies have considerably bigger performance edges than static policies. However, since dynamic algorithms [5] should collect and react to system state info, they're essentially a lot of complicated than static algorithms.

III. RELATED WORK

Numerous researchers have proposed load balancing algorithms [2], [12], [13] for parallel and distributed systems, as well as for cloud computing setting [14]. For a dynamic load-balancing algorithm, it's unacceptable to oftentimes exchange state information due to the high communication overheads. In order to scale back the communication overheads, Martin et al. [23] studied the results of communication latency, overhead, and information measure in a very cluster design to observe the impact on application performance. Anand et al. [2] planned associate calculable load data programming algorithm (ELISA), and Mitzenmacher [24] analyzed the usefulness of the extent to that previous data are often used to estimate the state of the system. Arora et al. [21] proposed a localized load-balancing formula for a Grid setting. Though this work tries to incorporate the communication latency between 2 nodes throughout the triggering method on their model, it didn't contemplate the actual price for employment transfer. Our approach takes the duty migration price under consideration for the load-balancing call. In [15], [16], and [18], a sender processor collects standing information concerning neighboring processors by communication with them at each load-balancing instant. This will lead to frequent message transfers. For a large-scale cloud environment wherever communication latency is extremely giant, the standing exchange at every load-balancing instant will lead to giant communication overhead. In our approach, the problem of frequent exchange of knowledge is mitigated by estimating the load, supported the system state data received at sufficiently giant intervals of your time. We have planned algorithms for a cloud setting that area unit supported the estimation approach as dole out in the design of enzyme-linked-sorbent serologic assay [2]. In ELISA, load equalization is carried out supported queue lengths. Whenever there's a distinction in queue length, jobs are going to be migrated to the gently loaded processor, ignoring the duty migration price. This cost becomes a vital issue once the communication latency is extremely Gantt like for a Grid setting and/or the job size is Gantt. Each of our algorithms balance the load by considering the duty migration price, that is primarily influenced by the out there information measure between the sender and receiver nodes.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

IV. PROBLEM DEFINITION

In today's competitive market, activity application success as "user interface" alone is not any longer enough. Poor convenience prices revenue, loyalty and whole image. Application leaders' square measure shifting business-centric metrics to service level management (SLM) [8] to bring IT nearer to business.

Our aim is to develop a scalable CLOUD resolution [6] that is capable of delivering desires of Stock Broking firm while not compromising on performance, measurability and price.

A. Features

We will be showing load balancing exploitation following options

1. User Level Load balancing on stock application

2. Cloud setup and application readying [8]

3. Obtaining Cloud statistics and performance analysis of every node

4. Resource watching [5] of Cloud Nodes

5. Deploying associate degree application war file on cloud nodes considering their processor, RAM Usage exploitation cloud controller.

V. SYSTEM DESIGN AND IMPLEMENTATION

In this system the dynamic cloud computing environment is used, the intermediate node is used to monitor the load of each VM in the cloud pool. In this approach the user can send the request to the intermediate node. It is responsible for transfer the client request to the cloud. Here, the load is considered as in terms of CPU load with the amount of memory used, delay or Network load.

A. Architecture

Proposed Load Balancing in Cloud Computing contains User, Server, Load Balancer, and Stock Application.

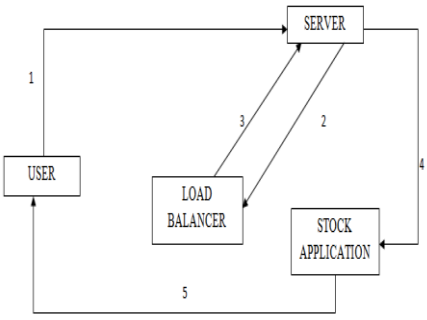


Fig. 2 System Architecture

- User: End users interact with the servers to manage information related to the cloud. Users are assigns task to the servers on which stock application is running.
- Server: On requesting of user the distributed server will send the request to load balancer to check whether any node is available or not. After getting response from load balancer server will migrate task coming from user to node on which stock application is running.
- Load Balancer: The load balancer monitors all nodes in cloud environment on which stock application is running. It calculates free RAM, free CPU and response time of each node. Then it selects one node who's RAM and CPU is less utilized and response time is very low, and sends migration link to server.
- Stock Application: After selecting proper node for execution of users' assigned task that node will send response to user's query.

B. System flow

In figure 3 we are proposing the flow of our system

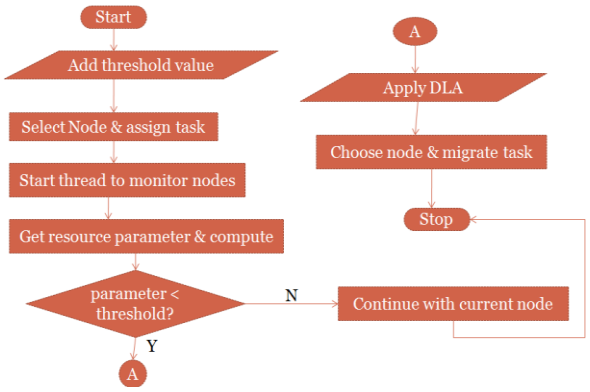


Fig. 3 System Flow

C. Dynamic Load Algorithm (DLA)

The DLA algorithm is used in this current project. The algorithm uses the six phases for load balancing as under

1) Get Load Status of All the Nodes*:* Here, we set a scheduler which contains a Monitor to gain and read load status, and also a Database to store the load status and work request historical data of user access to the server. Most of the current methods of nodes load status collection divided the system resource into several types: CPU utilization, Memory, Disk I/O and network bandwidth etc. But with different size of servers or provide different services we cannot propose a unified set of those parameters.

2) Evaluate the Status Of nodes*:* We set a threshold that when the resource utilization beyond the threshold, we can consider compute as an over-load node, also if the resource utilization is under the threshold we know that

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

the node is in a light-load status use and to represent those two statuses.

3) Predict the Future Load Flow: Based on the statistics, system's load status could show seasonal changes, which help to predict future, load of nodes.

4) Benefit Estimates: When a load status of N is signed as which caused by transient spike, in this condition we cannot make the decision that whether we should perform migration.

5) Choose Receiver Nodes: We use the forward probability method to help us to choose a receiver host, every candidate nodes' probability to receive a job or VM is mainly depends on the result of load status evaluation.

6) Migration: Helps migration of the heavily loaded nodes to the lighter ones.

The pseudo code is given in figure 4.

Pseudo Code for DLA

Input : *Cloud, VM nodes, Task and allocate the server to the client, Threshold value*
Output: *Solution(s) i.e. Task Completion*
Begin
Assign task to one of the node$_i$. Where i=1, 2, 3…..
Estimate the service time for assigned task. Also determine the CPU and RAM utilization of node.
If *CPU and RAM utilization is below threshold and response time is slow* ***then***
Start DLA
Calculate the utilizations of all nodes in cloud and service time.
Compare all three components i.e. CPU and RAM utilization and service time.
Select node that's response time is faster nd utilization is above threshold.
Transfer *the task to selected node.*
Else
Continue execution of current task to assigned node.

Fig. 4 DLA

VI. PERFORMANCE EVALUATION AND COMPARISONS

For this project there's would like of load testing tool to live performance of user's request. This project is predicated on cloud setting we want cloud load testing tool. There square measure several tools accessible on-line to live load on cloud nodes. For this project we tend to square measure exploitation Load cloud load testing tool.

Load Storm is on-line testing tool. The summery of load testing result's given below.

TABLE 1: SUMMARY OF THE RESULT

	Requests	**Response (average s)**	**Response (max s)**	**RPS (average)**	**Throughput (average)**	**Total Transfer**
HTML	288	0.6	1.01	0.24	23.05 kB/s	0.03 GB
Other *	1531	0.22	0.72	1.28	10.31 kB/s	0.01 GB
Total	1819	0.28	1.01	1.52	33.36 kB/s	0.4 GB

*Other includes javascript, css, images, pdf, task migration etc. (any content sort except hypertext mark-up language and xml)

Th The result's shown below pictures.

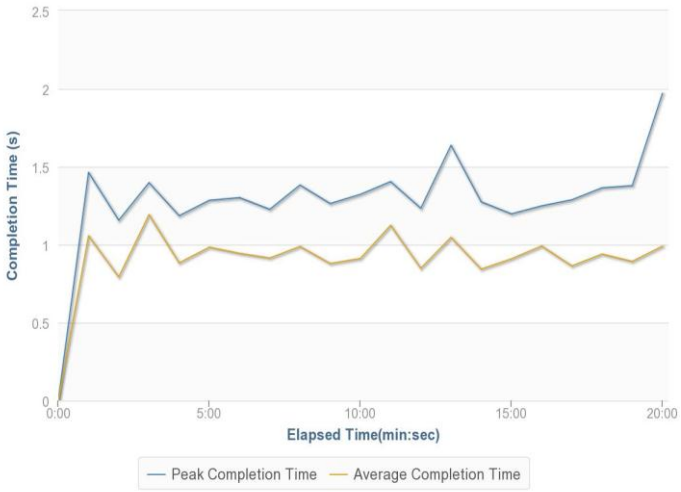


Fig. 5 All Pages Completion Time

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

Some of the load balancing techniques are tested and monitored on cloud analyst. Here Round Robin, Equally Spread Current Execution Load, and Throttled algorithms are used. These algorithms are tested on cloud analyst simulation tool and the result is given below table.

TABLE 2: COMPARISON RESPONSE TIME

Sr. No.	Name of Algorithm	Overall Response Time in ms		
		Avg.	Min	Max
1	DLA	280	33.36	1010
2	Round Robin	292.79	39.33	607.82
3	Equally Spread Current Execution Load	292.84	37.83	608.77
4	Throttled	292.79	38.51	597.84

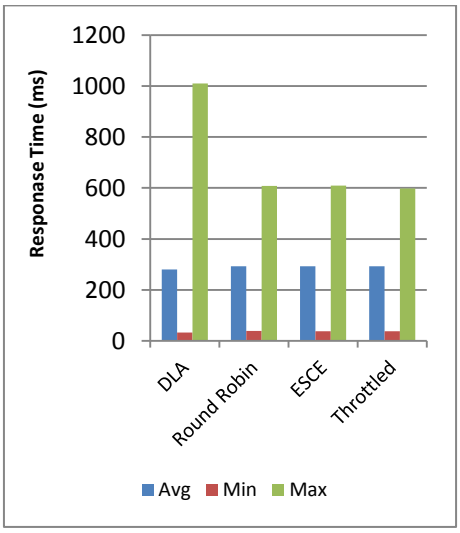


Fig. 6 Comparison Chart Response Time

Total data transferred in between servers and virtual users are given below. The data transfer is given in Giga Byte

TABLE 3: COMPARISON DATA TRANSFER

Virtual Users	ESCEL	RR	Throttled	DLA
5	0.31	0.32	0.32	0.3
10	0.5	0.48	0.51	0.55
15	1.38	1.37	1.38	1.45
20	1.98	1.98	1.96	2

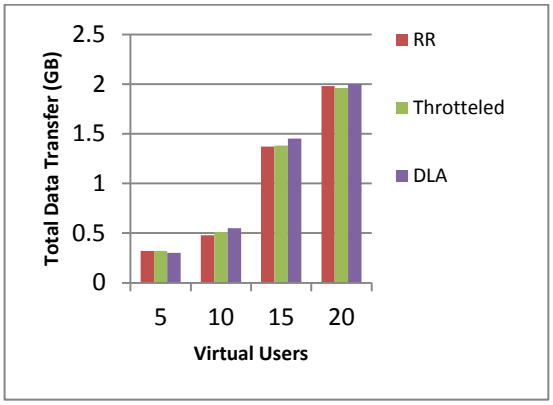


Fig. 7 Comparison Data Transfer

VII. CONCLUSION

Cloud Computing has wide been adopted by the busine although there square measure several subsisting problems like Server Consolidation, Load balancing, Energy Management, Virtual Machine Migration, etc. that haven't been comprehensive addressed . Central to those problems is that the issue of load balancing, that's needed to distribute the surplus dynamic native employment equally to all or any the nodes within the whole Cloud to realize a high used gratification and resource utilization magnitude relation. It nevertheless ascertains that each computing resource is distributed expeditiously and fairly.

Subsisting Load balancing techniques that are studied principally fixate on reducing overhead, accommodation replication time and ameliorative performance etc., however none of the techniques has thought-about the execution time of any task at the run time. Therefore, there's a requisite to develop such load balancing technique that may ameliorate the performance of cloud computing together with most resource utilization.

REFERENCES

[1] Hongbin Liang, Lin X. Cai, Dijiang Huang, Xuemin (Sherman) Shen, and Daiyuan Peng, "*An SMDP-Based Service Model for Inter domain Resource Allocation in Mobile Cloud Networks*", IEEE Transactions on Vehicular Technology, Vol. 61, No. 5, pp. 2222-2232, 2012.

[2] Zhenhuan Gong, Prakash Ramaswamy, Xiaohui Gu, Xiaosong Ma, "*SigLM: Signature-Driven Load Management for Cloud Computing Infrastructures*", IEEE Transactions on Grid Technology, Vol. 60, No. 5, pp. 978-986, 2011.

[3] Jianying Luo, Lei Rao, and Xue Liu, "*Temporal Load Balancing with Service Delay Guarantee for Energy Cost Optimization in Internet Data Centers*", IEEE Transactions on Parallel and Distributed Systems, pp. 1002-1011, 2013.

[4] Tim Dornemann, Ernst Juhnke, Bernd Freisleben, "*On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud*", 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 140-147, 2010.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 2, Issue 5, May 2015

[5] Ruchir Shah, Bhardwaj Veeravalli, and Manoj Misra, "*On the Design of Adaptive and Decentralized Load-Balancing Algorithms With Load Estimation for Computational Grid Environments*", IEEE Transactionson Parallel and Distributed Systems, Vol. 18, No. 12, pp. 1675-1686, 2010.

[6] Daniel Warneke, and Odej Kao, "*Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud*", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp. 985-997, 2011.

[7] Mladen A. Vouk, "*Cloud Computing – Issues, Research and Implementations*", IEEE Proceedings of the ITI 30th Int. Conf. on Information Technology Interfaces, pp. 31-40, 2011.

[8] Chun-Cheng Lin, Hui-Hsin Chin, and Der-Jiunn Deng, "*Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System*", IEEE Systems Journal, pp. 1-10, 2013.

[9] Yunhua Deng and Rynson W.H. Lau, "*On Delay Adjustment for Dynamic Load Balancing in Distributed Virtual Environments*", IEEE Transactions on Visualization and Computer Graphics, Vol. 18, No. 4, pp. 529-537, 2012.

[10] Yi Lua, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R. Larus, Albert Greenberg, "*Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services*", Elsevier Publication Performance Evaluation, pp. 1056–1071, 2011.

[11] Jun Wang, Qiangju Xiao, Jiangling Yin, and Pengju Shang, "*DRAW: A New Data-gRouping-AWare Data Placement Scheme for Data Intensive Applications with Interest Locality*", IEEE Transactions on Magnetics, Vol. 49, No. 6, pp. 2514-2520, 2013.

[12] Jiann-Liang Chen, Yanuarius Teofilus Larosa and Pei-Jia Yang, "*Optimal QoS Load Balancing Mechanism for Virtual Machines Scheduling in Eucalyptus Cloud Computing Platform*", IEEE Proceedings 2nd Baltic Congress on Future Internet Communications, pp. 214-221, 2012.

[13] Brighten Godfrey, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp, and Ion Stoica, "*Load Balancing in Dynamic Structured P2P Systems*", IEEE Infocom, pp. 1-8,2004.

[14] Qi Zhang, Lu Cheng, Raouf Boutaba, "*Cloud Computing: State-of-the-Art and Research Challenges*", Springer Publication, pp 7-18, 2010.

[15] Shamsollah Ghanbari, Mohamed Othman, "*A Priority based Job Scheduling Algorithm in Cloud Computing*", Elsevier publication International Conference on Advances Science and Contemporary Engineering, pp. 778-785, 2012.

[16] Giuseppe Aceto, Alessio Botta, Walter de Donato, Antonio Pescape, "*Cloud monitoring: A survey*", Elsevier Publication Computer Networks, pp. 2093–2115, 2013.

[17] Marc Eduard Frincu, "*Scheduling highly available applications on cloud environments*", Elsevier Publication Future Generation Computer Systems, pp. 1-16, 2012.

[18] L.D. Dhinesh Babu, P. Venkata Krishna, "*Honey Bee Behaviour Inspired Load Balancing of Tasks in Cloud Computing Environments*", Elsevier Publication Applied Soft Computing, pp. 1-12, 2013.

[19] Tin-Yu Wu, Wei-Tsong Lee, Yu-San Lin, Yih-Sin Lin, Hung- Lin Chan, Jhih-Siang Huang, "*Dynamic Load Balancing Mechanism based on Cloud Storage*", Proceedings of IEEE, pp. 102-106, 2012.

[20] John Harauz, Lorti M. Kaufinan. Bruce Potter, "*Data Security in the World of Cloud Computing*", IEEE Security & Privacy, Co-published by the IEEE Computer and Reliability Societies, pp. 61-64, 2009.

[21] Yashpalsinh Jadeja and Kirit Modi, "*Cloud Computing - Concepts, Architecture and Challenges*", International Conference on Computing, Electronics and Electrical Technologies, Co-published by the IEEE, pp. 887-880, 2012.

[22] Ramgovind S, Eloff MM, Smith E, "*The Management of Security in Cloud Computing*", Information Security for South Africa (ISSA), Co-published by the IEEE, pp. 1-7, 2010.

[23] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "*Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility*", Future Generation Computer Systems, Volume 25, Number 6, Elsevier Science, pp. 599-616, 2009.

[24] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, "*Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities*", Proceedings of the 7th High Performance Computing and Simulation Conference IEEE Press, pp. 1-11, 2009.