International Journal of Scientific & Engineering Research, Volume 5, Issue 3, March-2014
ISSN 2229-5518

Performance Evaluation of Load Balancing Algorithms on Cloud Data Centers

Soumya Ranjan Jena, Sudarshan Padhy, Balendra Kumar Garg

Abstract— Cloud computing is the state-of-the-art of research and challenge and one of the recent research emerging trends in the field of computer science and engineering. This work is moreover an extension of the work [1] that Soumya Ranjan Jena and Zulfikhar Ahamad have previously performed. Basically Cloud computing provides services that are referred to as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). It has many advantages along with some crucial issues to be resolved in order to improve reliability of cloud environment. These issues are related with the load management, fault tolerance and different security issues in cloud environment. Load balancing is one of the essential factors to enhance the working performance of the cloud service provider. Since, cloud has inherited characteristic of distributed computing and virtualization there is a possibility of occurrence of deadlock. The aim of this paper is to demonstrate and discuss the critical role of load balancing of resources that plays in improving and maintaining the availability in cloud systems.

Index Terms— Cloud computing, Efficient load balancing, Round robin load balancing algorithm, Active monitoring load balancing algorithm, Throttled load balancing algorithm, Cloud analyst, IBM SPSS Amos, Regression analysis.

1 Introduction

CLOUD data centers are the foundations to support many internet applications, enterprise operations, and scientific computations. Data centers are driven by large-scale computing services such as web searching, online social networking, online office and IT infrastructure outsourcing, and scientific computations. A data center can run a large variety of applications and services, and a smart routing protocol should guarantee the performance of each application by efficiently utilizing the link capacity, e.g., distributing the traffic among the links inside the data center as evenly as possible.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. It promises to eliminate the need for maintaining expensive computing facilities by companies and institutes alike. Through the use of virtualization and resource time sharing, clouds serve with a single set of physical resources a large user base with different needs. Moreover, the use of virtualization and resource time sharing may introduce significant performance penalties for the demanding scientific computing workloads. In this context we evaluate the performance of three different load balancing algorithms through regression analysis and find the efficient among them having least significant error. The load balancing algorithms automatically move load between servers so that most of the hardware resources are effectively utilized and to avoid any resource overloading situations.

- *Soumya Ranjan Jena, Teaching Associate, Department of CSE, AKS University, Satna, M.P., India.*
- *Dr. Sudarshan Padhy, Director, IMA, Bhubaneswar, Odisha, India.*
- *Belaendra Kumar Grag, Teaching Associate, Department of CSE, AKS University, Satna, M.P., India.*

2 Characteristics of Cloud Computing

Cloud computing has the following five essential characteristics [2]

Fig.1 Cloud Characteristics

1) *On-demand self-service*: A consumer can practically provision computing capabilities, such as server time and network storage, when needed automatically except requiring human interaction with each service's provider.
2) *Broad network access*: Capabilities are available over the network and it has accessibility through standard mechanisms which promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

3) *Resource pooling*. For serving multiple consumers using a multitenant model, the provider's computing resources are pooled to serve, with different physical and virtual resources dynamically assigned and reassigned as per consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

4) *Rapid elasticity*. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

5) *Measured Service*. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

3. EFFICIENT LOAD BALANCING ALGORITHMS

Load balancing is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded [1].

We are basically going to analyze three basic load balancing algorithms. They are:

- *Round Robin (RR)*: It is one of the simplest scheduling techniques that utilize the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or time interval i.e. it utilizes the principle of time scheduling. Each node is given a quantum and in this quantum the node will perform its operations. The resources of the service provider are provided to the requesting client on the basis of this time slice.

- *Active Monitoring (AM)*: It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines. The load balancer maintains a queue of the jobs that need to use and are currently using the services of the virtual machine. The balancer then continuously scans this queue and the list of virtual machines. If there is a VM available that can handle request of the node/client, the VM is allocated to that request. If however there is a VM that is free and there is another VM that needs to be freed of the load, then the balancer distributes some of the tasks of that VM to the free one so as to reduce the overhead of the former VM.

- *Throttled (TH)*: In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation. The process first starts by maintaining a list of all the VMs each row is individually indexed to speed up the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

4. RELATED WORK

Early work in load balancing is devoted to minimize the response time of different load balancing algorithms in cloud based infrastructure [1]. Ram Prasad et al. [4] have studied divisible load scheduling theory in cloud computing. Kumar Nishant et al. [5] have demonstrated load balancing using Ant colony optimization. In [6] Jasmin James et al. have proposed a better allocation policy called weighted active monitoring load balancing by assigning weights to each VM. Soumya Ray et al. [7] have identified qualitative components for simulation in cloud environment and then based on these components; he has explained execution analysis of load balancing algorithms. Ajith Singh. N et al. [8] have given a semi-distributed load balancing approach in cloud based infrastructure. In [9] authors have demonstrated efficient load balancing in cloud computing using Fuzzy logic. H.Mehta et al. [10] have formulated a new content aware load balancing policy named as workload and client aware policy (WCAP). It uses a unique and special property called UPS that defines the requests as well as computing nodes. USP helps the scheduler to decide the best suitable node for the processing the requests. A. M. Nakai et al. [11] have defined a distributed new server based-load balancing policy for web servers. It helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them. Y. Lua et al. [12] have explained a Join- Idle-Queue load balancing algorithm for dynamically scalable web services which provides large scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. J. Hu et al. [13] have investigated the problem of scheduling on load balancing on VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and

ISSN 2229-5518

reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load balance and high cost of migration thus achieving better resource utilization. A.Bhadani et al. [14] have suitably explained a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment. This policy improves the overall performance of the system but does not consider the systems that are fault-tolerant. A. Singh et al. [15] have proposed a novel load balancing algorithm known as Vector Dot. It handles the hierarchical complexity of the data-center and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies. Y. Fang et al. [16] have studied the problem of two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.S. Wang et al. [17] have formulated a two- phase scheduling algorithm which combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. M. Randles et al. [18] have investigated a distriuted and-scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. He has also have demonstrated [19] Honeyeebee Foraging Algorithm which is derived from the behavior of honey bees for finding and reaping food.

In contrast to the above discussed studies, we discuss three basic efficient and enhanced algorithm of load balancing and show the regression analysis of each for two different cases on cloud data centers.

5. PERFORMANCE EVALUATION THROUGH REGRESSION ANALYSIS

In this section we first calculate the response time of different load balancing algorithms using the tool cloud analyst [10] which is a cloud sim based GUI tool used for modelling and analysis of large scale cloud computing environment. Moreover, it enables the modeller to execute the simulation repeatedly with the modifications to the parameters quickly and easily. The following diagram shows the GUI interface of cloud analyst tool [1].

Fig.2 GUI Interface of Cloud Analyst

Simulation setup and analysis of results are carried out for a period of 60 hrs by taking different numbers of users, 3 data centers i.e. DC1, DC2, and DC3 having 75, 50 and 25 numbers of VMs respectively. The other parameters are fixed according to Table 1 as shown.

Table 1. Setting of Parameters

Parameter	Value Passed
VM-image size	10000
VM-memory	1024 MB
VM-bandwidth	1000
Service broker policy	Optimize response time
Data center architecture	x86
Data center-OS	Linux
Data center-VMM	Xen
Data center- No of VMs	DC1-75 DC2-30 DC3-50
Data center-memory per machine	2 GB
Data center-storage per machine	1 TB
Data center-available bandwidth per machine	1000000
Data center-processor speed	10000
Data center-VM policy	Time shared
User grouping factor	1000

ISSN 2229-5518

Request grouping factor	250
Executable instruction length	250

After performing six different experiments by cloud analyst successfully in two cases we get the overall response time of different load balancing algorithms as given in the Table 2 and Table 4 and overall data center processing time as given in the Table 3 and Table 5.

CASE-I: VMs having Same Number of Processors

In this case we consider all virtual machines having same number of processors i.e. quad core processors.

Table 2. Overall Response Time for Case-I

No of users	Overall Response Time (in ms)		
	RR	AM	TH
6000	187.41	187.52	187.47
12000	195.63	195.82	195.67
18000	198.19	198.38	198.34
24000	199.50	199.56	199.58
30000	200.23	200.31	200.27
36000	200.87	200.96	200.88
42000	201.04	201.11	201.13
48000	201.43	201.51	201.44

Motivation for Regression Analysis

We perform regression analysis through the tool called "IBM SPSS Amos" which implements the general approach to data analysis known as structural equation modeling (SEM), also known as analysis of covariance structures, or causal modeling [10]. This approach includes, as special cases, many well-known conventional techniques, including the general linear model and common factor analysis.

Here we generally perform the linear regression to find out the degree of correlation. At the time of calculating correlation the correlation value should always be in between -1 to +1; where -1 means perfect negative correlation and +1 means perfect +ve correlation.

In the first case we take number of users as independent variable and the overall response time of Round robin algorithm (RR) as dependent variable. In the second case we take same number of users as independent variable and overall response time of Active Monitoring (AM) as dependent variable. Similar case for the Throttled algorithm (TH). After analysis through IBM SPSS Amos version 22 we get the following observations.

In the line of regression of dependent variable on all requested variable entered in the line which gives the best estimate by dependent variable for any given value of all requested variable. It is also obtained by the principle of least square on minimizing the sum of square of the error parallel to the X- axis in all three algorithms. By starting with the equation of the form:

Dependent variable = A + B (all requested variable) ---------- (i)
and minimizing the sum of squares of errors at estimates of dependent variable, i.e. derivations between the given users and their estimates given by line of regression of dependent variable users on all requested variable, i.e. minimizing.

$$E = \Sigma (x-A-By)^2 \text{ -------------------- (ii)}$$

Where x is the dependent variable and y is the all requested user. We shall get the normal equation for estimating A and B as:

Σ Dependent variable = n A + B Σ all requested variable
And
Σ Dependent variable x all requested variable = A Σ all requested variable + B Σ (All requested variable)^2 --------- (iii)
Now solving (iii) simultaneously for A and B we shall get:
$A = \{(\Sigma(y)^2) (\Sigma x) – (\Sigma y) (\Sigma xy)\} / n\Sigma y^2 – (\Sigma y)^2$
and
$B = n \Sigma xy – (\Sigma x) (\Sigma y)/ n\Sigma y^2 – (\Sigma y)^2$
Substituting these values of A and B in (i) we shall get the required equation of line of regression of dependent variable on all required variable.

Regression Analysis of Round Robin Algorithm

Table 3. Variables Entered/Removed

Model	Variables Entered	Variables Removed	Method
1	Users	-	Enter

a. Dependent Variable: RR

b. All requested variables entered

Table 4. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.832	0.692	0.641	2.81185

a. Predictors: (Constant), users

ISSN 2229-5518

Table 5. Co-efficients

Model	Unstandard-ized Coeffi-cients		Stand-ardized Coeffi-cients		
	B	Std. Error	Beta	T	Sig.
Const	190.86	2.191		87.1	0.000
Users	0.000	0.000	0.832	3.67	0.010

a. Dependent Variable: RR

Curve Estimation of Round Robin Algorithm

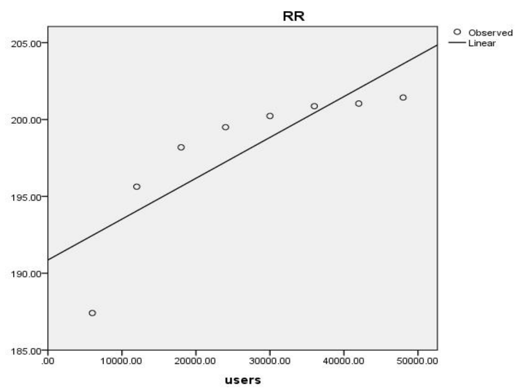

Fig.3 Curve Estimation of RR Algorithm for CASE-I

Regression Analysis of Active Monitoring Algorithm

Table 6. Variables Entered/Removed

Model	Variables En-tered	Variables Re-moved	Method
1	Users	-	Enter

a. Dependent Variable: AM

b. All requested variables entered.

Table 7. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Es-timate
1	0.829	0.686	0.634	2.82824

a. Predictors: (Constant), users

Table 8. Co-efficients

Model	Unstandardized Coefficients		Stand-ardized Coeffi-cients		
	B	Std. Error	Beta	T	Sig.
Const	191.028	2.204		86.68	0.000
Users	0.000	0.000	0.829	3.625	0.011

a. Dependent Variable: AM

Curve Estimation of Active Monitoring Algorithm

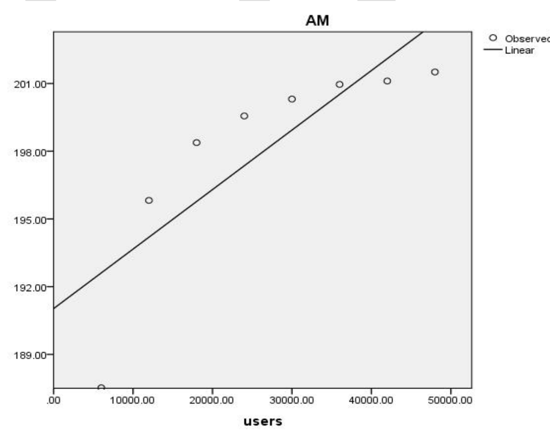

Fig.4 Curve Estimation of AM Algorithm for CASE-I

Regression Analysis of Throttled Algorithm

Table 9. Variables Entered/Removed

Mod-el	Variables En-tered	Variables Re-moved	Method
1	Users	-	Enter

a. Dependent Variable: TH

b. All requested variables entered.

Table 10. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.83	0.688	0.636	2.82836

a. Predictors: (Constant), users

Table 10. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.83	0.688	0.636	2.82836

b. Predictors: (Constant), users

Table 11. Co-efficients

Model	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
Const	190.95	2.204		86.65	0.000
Users	0.000	0.000	0.830	3.639	0.011

a. Dependent Variable: TH

Curve Estimation of Throttled Algorithm

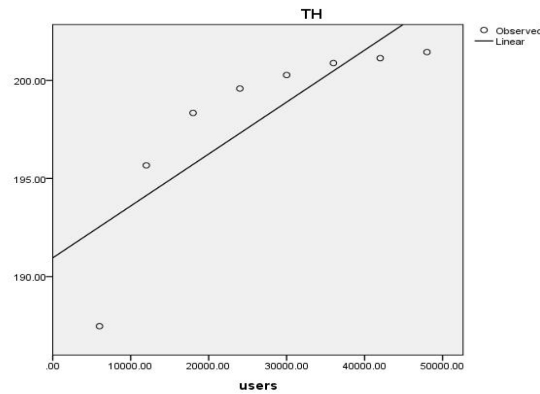

Fig.5 Curve Estimation of TH Algorithm for CASE-I

CASE-II VMs having Different Numbers of Processors

In this case we consider all virtual machines having different numbers of processors i.e. DC1 having the mixture of dual core and quad core processors, whereas DC2 having only dual core processors and finally DC3 have dual core, quad core and hexa core processors.

Table 12. Overall Response Time for Case-II

No of Users	Overall Response Time (in ms)		
	Round Robin	Active Monitoring	Throttled
6000	195.91	192.21	192.75
12000	200.99	197.28	197.29
18000	201.57	199.72	199.14
24000	203.69	199.90	199.92
30000	204.18	200.45	200.43
36000	204.54	200.82	200.84
42000	204.79	201.06	201.06
48000	201.96	201.96	201.28

Regression Analysis of Round Robin Algorithm

Table 13. Variables Entered/Removed

Model	Variables Entered	Variables Removed	Method
1	Users	-	Enter

a. Dependent Variable: RR

c. All requested variables entered.

ISSN 2229-5518

Table 14. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.706	0.498	0.415	2.23651

a. Predictors: (Constant), users

Table 15. Co-efficients

Model	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
Const	198.42	1.743		113.86	0.000
Users	0.000	0.000	0.706	2.441	0.05

a. Dependent Variable: RR

Curve Estimation of Round Robin Algorithm

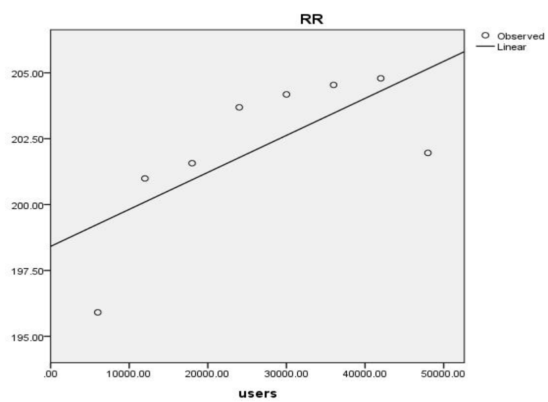

Fig.6 Curve Estimation of RR Algorithm for CASE-II

Regression Analysis of Active Monitoring Algorithm

Table 16. Variables Entered/Removed

Model	Variables Entered	Variables Removed	Method
1	Users	-	Enter

a. Dependent Variable: AM

b. All requested variables entered.

Table 17. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.848	0.719	0.672	1.79231

a. Predictors: (Constant), users

Table 18. Co-efficients

Model	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
Const	194.30	1.397		139.13	0.000
Users	0.000	0.000	0.848	3.917	0.008

a. Dependent Variable: AM

Curve Estimation of Active Monitoring Algorithm

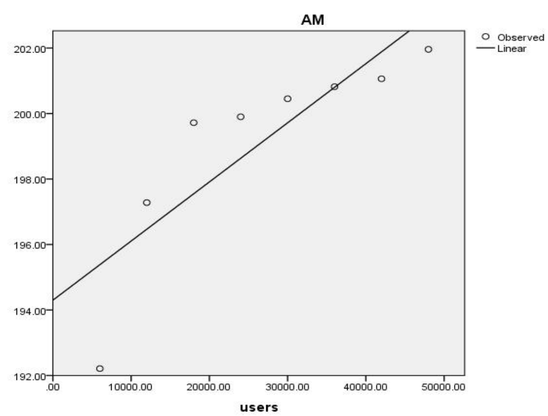

Fig.7 Curve Estimation of AM Algorithm for CASE-II

Regression Analysis of Throttled Algorithm

Table 19. Variables Entered/Removed

Model	Variables Entered	Variables Removed	Method
1	Users	-	Enter

ISSN 2229-5518

a. Dependent Variable: TH

b. All requested variables entered.

Table 20. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.855	0.731	0.687	1.60686

a. Predictors: (Constant), users

Table 21. Co-efficients

Model	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
Const	194.58	1.252		155.4	0.000
Users	0.000	0.000	0.855	4.041	0.007

a. Dependent Variable: TH

Curve Estimation of Throttled Algorithm

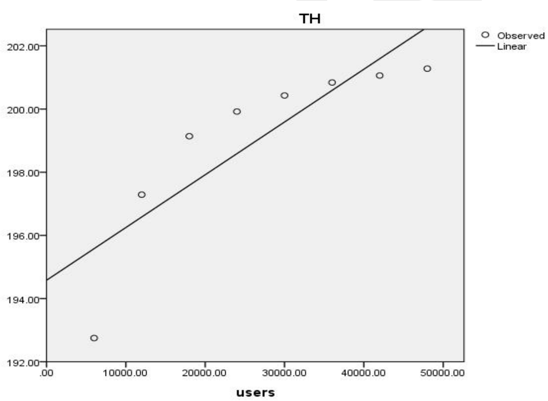

Fig.8 Curve Estimation of TH Algorithm for CASE-II

6. CONCLUSION

In case-1 we find Active monitoring load balancing algorithm and Throttled load balancing algorithm both have same correlation co-efficient value having 0.11 whereas the Round robin has 0.10. Therefore all these three algorithms are efficient where each virtual machine has same number of processors.

On the other hand, when the number of processors per each virtual machine is different then we found that Round robin load balancing algorithm has higher correlation co-efficient (i.e. 0.05) in comparison to Active monitoring (0.008) and Throttled (0.007). Therefore, these two are the efficient algorithms for load balancing in cloud computing environment.

7. ACKNOWLEDGEMENTS

We would like to thank Mr. Kumar Aashish, Asst. Professor, Faculty of Management Studies, AKS University for his help and co-operation for the completion of the work. We are greatful to Er. Satyabadi Kumar Jena, Mining Engineer, Coal India Limited for his motivation to write this paper.

8. REFERENCES

[1] Soumya Ranjan Jena and Zulfikhar Ahmad, “Response time minimization of different load balancing algorithms in cloud computing environment”, IJCA, Volume-69, No-17, May 2013 edition.

[2] Sherif Sakr, Anna Liu, Daniel M. Batista, and Mohammad Alomari, “A survey of large scale data management approaches in cloud environments”, IEEE Communications Surveys & Tutorials, Vol. 13, No. 3, 2011.

[3] Soumya Ranjan Jena, “Bottle-necks of cloud security- A survey”, Volume 85, No 12, January 2014.

[4] Ram Prasad Padhy and P Goutam Prasad Rao, “Load balancing in cloud computing systems”, Department of Computer Science and Engineering, NIT, Rourkela, Odisha, India, May, 2011.

[5] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kunwar Pratap Singh and Nitin and Ravi Rastogi, “Load balancing of nodes in cloud using ant colony optimization”, 2012, 14th International conference on modeling and simulation.

[6] Jasmin James and Dr. Bhupendra Verma, “Efficient VM load balancing algorithm for a cloud computing environment”, International Journal on Computer Science and Engineering (IJCSE), 2012.

[7] Soumya Ray and Ajanta De Sarkar, “Execution analysis of load balancing algorithms in cloud computing environment”, IJCCSA, Vol.2, No.5, October 2012.

[8] Ajith Singh. N and M. Hemalatha, “An approach on semi -distributed load balancing algorithm for cloud computing system”, IJCA, Volume 56– No.12, October 2012.

[9] Srinivas Sethi, Anupama Sahu and Suvendu Kumar Jena, “Efficient load balancing in cloud computing using Fuzzy logic “, IOSRJEN, ISSN: 2250-3021 Volume 2, Issue 7, 2012.

[10] IBM SPSS Amos 22- User’s Guide- James L. Arbuckle.

[11] H. Mehta, P. Kanungo, and M. Chandwani, “Decentralized content aware load balancing algorithm for distributed computing environments”, Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.

[12] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb and A. Greenber, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services”, An international Journal on Performance evaluation.

[13] J. Hu, J. Gu, G. Sun, and T. Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environ-

ment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, 89-96.

[14] A. Bhadani and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM.

[15] A. Singh, M. Korupolu, and D. Mohapatra, "Serverstorage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

[16] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[17] S. Wang, K. Yan, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, September 2010, pages 108-113.

[18] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, April 2010, pages 551-556.

[19] Martin Randles, Enas Odat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009, Second International Conference on Developments in eSystems engineering.

9. BIOGRAPHIES

Soumya Ranjan Jena is Teaching Associate in the department of CSE at AKS University, Satna. He has completed B.Tech in CSE, M.Tech in IT, and CCNA. He is a member of Cisco Networking Academy. He has published two books and four International research papers in the field of Cloud computing. His research interests include Cloud computing (Load balancing, Security, Moving data to media cloud, Power consumption minimization, etc.), Sensor Networks, Algorithms, Wireless Networks, and Mobile Computing.
Email: *soumyajena1989@gmail.com*.

Sudarshan Padhy, PhD, is Director at Institute of Mathematics and Applications, Bhubaneswar. Having 40 years of teaching and research experience he has guided many PhD and M.Tech students in the field of Numerical Analysis, Parallel algorithm, Computational fluid dynamics, Computational finance, Bioinformatics, etc. Email: *spadhy07@gmail.com*.

Balendra Kumar Garg is currently working as the Teaching Associate in the department of CSE at AKS University. He has completed MCA and M.Phil in Computer Science. He has 5 years of teaching experience. Email: *garg_balendra@yahoo.com.*