International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.4, August 2013, pp. 313~320
ISSN: 2089-3337

Dynamic Resource Provisioning in Cloud based on Queuing Model

Sandeep K. Sood
Department of Computer Science & Engineering, Guru Nanak Dev University Regional Campus Gurdaspur, India

Article Info

Article history:

Received Jun 18th, 2013
Revised Jul 10th, 2013
Accepted Jul 30th, 2013

Keyword:

Cloud computing
Data centers
Service level agreements
Resource provisioning
Queuing model

ABSTRACT

One of the main aim of cloud computing is to provide bigger data center that will carter the storage needs of end user. In a data centre, server clusters are used to provide the required processing capability to get acceptable response time for interactive applications. Managing many applications on consolidated resources is difficult and complex. Deadlock can occur which effects all other running applications. In this paper, an interactive system based on queuing model is presented in which the cloud customer (CC) initially establishes the session to access the resources. The proposed model uses banker's algorithm for resource allocation due to which deadlock for resource allocation among various processes is not possible. Moreover, by putting restriction on number of login users, resources are not choked out even in case of heavy demand of resources. The concept of resource allocation matrix helps the cloud service provider to predict the resource requirements in advance. Resources are dynamically allocated according to the requirements of the user. The results obtained are accurate in terms of predicting the minimum number of processor nodes required to meet the performance goal of an interaction application.

Corresponding Author:

Author,
Department of Computer Science & Engineering,
Guru Nanak Dev University Regional Campus Gurdaspur, India
Email: san1198@yahoo.co.in

1. INTRODUCTION

Cloud computing is a paradigm shift where details are abstracted from the users who no longer have need of expertise or control over the technology infrastructure. It describes a new supplement, consumption and delivery model for information technology services based on Internet. This concept involves a computing capability that provides an abstract between the computing resources such as server, storage, networks, applications, services and its underlying technical architecture. It requires the provisioning of dynamically scalable virtualized computing resources as a service over the internet.

It is a pay-per-use model for providing available, convenient, on-demand network access from a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. The main elements of cloud computing are on demand self- service, broad network access, resource pooling, rapid elasticity and other required services by the end user [1]. Cloud computing provides virtually unlimited computational resources to its users, while letting them pay only for the resources they actually use at any given time. Cloud service providers offer resources to the users on demand using various pricing schemes based on on-demand instances, reserved instances and spot instances.

A data center in cloud is a large distributed computing environment consisting of heterogeneous computing resources including individual servers, server clusters and databases. Due to the decreasing cost of hardware and the advancement in Internet technologies, data centers are increasingly used to host diverse services by web and application service providers. Resource management in large data centers is an important issue, especially for the next generation of data centers [2]. Traditionally this is done manually,

which takes a long time and often results in poor resource utilization. Autonomic resource provisioning is an effective approach for resource management in data centers. Its objective is to automate the dynamic allocation of resources by minimizing the mean amount of resources used and meet the performance goals of individual applications as specified in their service level agreements.

Amazon played a key role in the development of cloud computing by modernizing their data centers. Amazon EC2 is a commercial web service that allows small companies and individuals to rent computers on which they run their own computing applications. Amazon EC2 is an elastic cloud computing service that offers complete control over web computing resources. Its main feature is pay per use as per minute usage basis. Amazon offers other web services that allow developers to easily utilize different degrees of infrastructure and software as service (SaaS) needs on an as needed basis. The salesforce.com launched force.com as platform as a service (PaaS) in which companies developers build, store and run all of the applications and websites in the cloud they needed to run their business. In 2010, salesforce.com introduced the cloud based database at database.com for developers for the development of could computing services on any device which can run on any platform and can be written in any programming language.

Microsoft windows azure is a cloud computing platform used to build, host and scale web application through Microsoft data centers. The platform consists of various on-demand services hosted in Microsoft data centers. These are windows azure (an operating system providing scalable compute and storage facilities), SQL azure (a cloud based version of SQL Server) and windows azure appfabric (a collection of services supporting applications both in the cloud and on premise). There are a lot of commercial cloud service providers who provide their cloud services on demand. Google offers Google Apps as SaaS and Google App Engine as PaaS. Google Apps allow user to create and store documents entirely in the cloud. The salesforce.com is a leader in SaaS. Microsoft's azure specializes in PaaS. Microsoft also provides free storage space on Sky Drive.

Large amounts of data can be generated by an organization and this data can be so large that it needs to be stored at a data center. This problem can be solved by storing the data on the cloud. Large data centers are used to host multiple application services, whose workloads can vary widely and can be hard to predict. The data center manages numerous resources, including compute servers, database servers, storage devices etc. and serves different users using multiple large-scale applications [3]. This creates challenges for dynamic resource allocation. A solution using queuing network models is presented in this paper to determine the best possible configuration for the data center. The processing of interactive jobs is considered only because these jobs generally have small processing requirements and require good response time.

This paper is organized as follows. Section 2 summarizes the related work for resource provisioning in cloud environment. In Section 3, a model based on queuing theory is proposed. Mathematical and experimental results are shown in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In 2003, Doyle et al. [4] proposed a resource provisioning approach based on a coordinated provisioning of memory and storage resources using cache hit ratio and storage response time as provisioning goals. In 2005, Bennani and Menasce [5] presented a combination of analytic queuing network models and combinatorial search techniques which is used to determine the best possible configuration for the data center. In 2005, Tesauro et al. [6] proposed a utility function driven resource allocation model based on a system of N parallel M/M/1 queues and use these results for the mean response time and throughput to maximize the total utility. In 2005, Urgaonkar and Chandra [7] presented an analytical model based on queuing theory. This model determines the number of servers to be allocated to each tier of a multi-tier application.

In 2006, Woodside et al. [8] proposed queuing based model which presents the effect of workload and system parameters on performance. Our focus on this paper is to develop a model that provides the scalability of resources in cloud. Our proposed model reduces complexity and makes resource provisioning very effective and efficient as number of login user keep on changing to utilize the resources of cloud. In 2007, Zhang et al. [7] developed a statistical regression based analytic model to approximate the CPU demands of different transaction types along all the tiers in the system.

In 2007, Padala et al. [10] presented a system that can meet application level quality of service while achieving high resource utilization. In 2009, Ye et al. [11] presented an efficient and effective algorithm to determine the allocation strategy that results in smallest number of servers required and also developed a scheduling discipline, called probability dependent priority, which is superior to First-Come-First-Serve (FCFS) and head-of-the-line priority in terms of requirement of number of servers. Two server strategies are considered for the resource allocation i.e. shared allocation (SA) and dedicated allocation (DA).

In 2011, Fayoumi [12] presented a discrete event simulation to evaluate the performance with respect to the different load points. The performance metrics were the average waiting time as well as the number of

tasks. In 2012, Wang et al. [13] presented adaptive model-free approaches for resource allocation and energy management under time varying workloads and heterogeneous multi-tier applications. This approach can achieve better effect and efficiency than the model-based approaches in dealing with real-world workloads. In 2012, Grewal and Pateriya [14] proposed a rule based resource manager to scale up private cloud and presented a cost effective solution in terms of money spent to scale up private cloud on-demand by taking public cloud's resources and that never permits secure information to cross the organization's firewall in hybrid cloud. This approach also set the time for public cloud and private cloud to fulfill the request and provide the services in time. In 2012, Wu et al. [15] proposed innovative admission control and scheduling algorithms for SaaS providers to effectively utilize public cloud resources to maximize profit by minimizing cost and improving customer satisfaction level. They also conducted an extensive evaluation study to analyze which solution suits best in which scenario to maximize SaaS provider's profit. In 2012, Koperek and Funika [16] proposed an approach to automatic infrastructure scaling based on observation of business related metrics that enables cloud based systems to seamlessly adjust the constantly changing environment of the Internet.

In 2013, Sood [17] proposed a value based prediction model for resource provisioning in which a resource manager is used for dynamically allocating or releasing a virtual machine depending upon the resource usage rate. In order to know the recent resource usage rate, the resource manager uses sliding window to analyze the resource usage rate and to predict the system behavior in advance. By predicting the resource requirements in advance, a lot of processing time can be saved. Earlier, a server has to perform all the calculations regarding the resource usage that in turn wastes a lot of processing power thus decreasing its overall capacity to handle the incoming request. The main features of this model is that a lot of load is being shifted from the individual server to the resource manager as it performs all the calculations and therefore the server is free to handle the incoming requests to its full capacity.

3. Proposed Model

Different approaches have been used in the design of dynamic resource allocation. The work presented in this paper uses queuing model and the resource allocation algorithm i.e. banker's algorithm approach that allocate resources dynamically. The proposed approach can solve performance issues by using mathematical concepts of queuing model and banker's algorithm. The proposed work is divided into two phases. First phase deals with process of session initialization for any incoming request while the second phase deals with the processing of the requests submitted by login users.

3.1. Phase 1: Session Initialization

To access the resources from controlling Authority (CA), the cloud customer (CC) firstly establishes the session. Figure 1. shows the concept of session initialization, where CC submits the request to CA for session initialization. For any arriving request, if the number of login user is already at maximum M, the arriving request is buffered into the waiting queue until session is terminated by one of the logon user. Since new sessions can be created and existing sessions can be terminated, the number of logon users can change over time. The number of login users is increased by 1, each time a new session is created and decreased by 1, each time an existing session is terminated. This information is send to the CA to keep the record of number of login user.

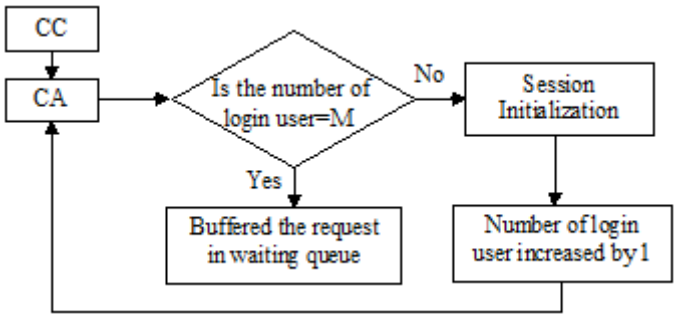

Figure 1. Session initialization

3.2. Phase 2: Request Processing

After initialization of session, the CC submits the request to the Cloud Service Provider (CSP) for processing of the request. Figure 2 shows the processing of incoming requests, where k logon users at their workstations interact with a service facility containing M servers (where k <= M). Jobs submitted by logon

users follows resource allocation and deadlock avoidance algorithm i.e. Banker's algorithm that allocate resources dynamically and tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources. When the system receives a request for resources from login user, it runs the Banker's algorithm to determine if it is safe to grant the request, if so then grant the request otherwise buffer the request. The servers serve requests conservatively i.e. no server stays idle while there is a request waiting in the queue. When a job completes service at a server, there are two choices for the user that whether continue this existing session or terminate it. Probability y represents the termination of a user session and 1-y indicates that the login user will continue with this session.

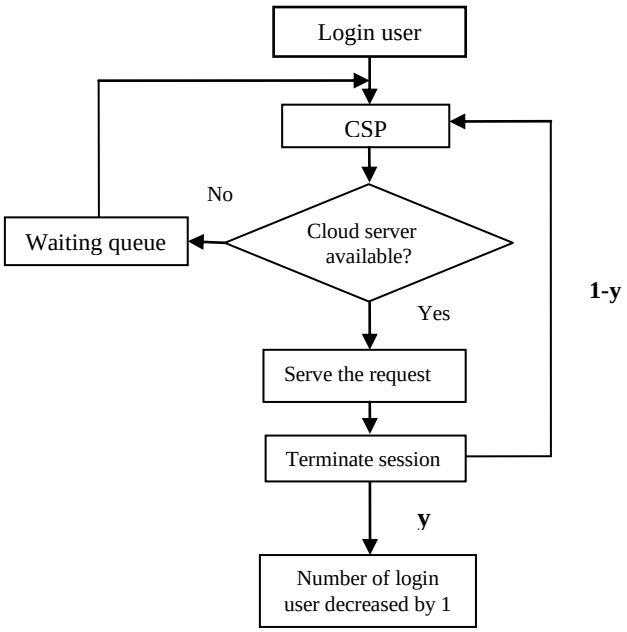

Figure 2. Flow chart of processing incoming requests

The logon user sends the request to CSP in the form of matrix $M_{1,R}=[R_1\ R_2\ R_3$ - - - - - -$R_R]$. In this proposed model, we take a matrix of 6 resources i.e. $M_{1,R}=[R_1\ R_2\ R_3\ R_4\ R_5\ R_6]$. Where, R_1 stands for CPU's processing speed (in Teraflops), R_2 stands for memory units (in Gigabytes), R_3 stands for bandwidth (in GB/sec), R_4 stands for quality of network resources, R_5 stands for storage capacity (in Terabytes), R_6 stands for number of virtual nodes required at the cloud. The CC's request parameters are $\{M_{1,\ 6}$, session initialization time, session termination time$\}$.

Example: Suppose the CC makes a request to CSP as {[233364], 12:00 pm, 12:25 pm}. The first parameter of the user's request is the resource provisioning matrix. Here, the resource provisioning matrix has six elements. 2 in the first column states that 2 teraflops of processing speed is required. The second column of the matrix has value 3 which means 3 GB memory is required. The third column shows the requirement of 3 GB/sec network bandwidth. The quality of network resources should of rating 3 i.e. of medium quality level along with 6 TB storage space. The number 4 in the last column shows that 4 virtual nodes are required at the cloud to accomplish the user's request. The other parameters state that the session initialization for request is 12:00 pm, while the session termination time is 12:25 pm.

In this section, we develop a model for the interactive system in which there are k logon users who interact with a system that consists of M identical processor nodes (where k <= M). A session has been created for each logon user. In this paper, a system consisting of a server cluster having M server in the data center is used in which number of logon users interact with this system. The number of logon users is increased by 1 when a new session is created and decreased by 1 when an existing session is terminated. To avoid degradation in response time, the system places a limit M on the number of logon users because after this limit, even one server can not be allocated to logon user. Generally, the number of login users (k) is comparatively very less than that of total number of servers (M) because more than one server's resources are allocated to each user in real time interactive applications. Arrival of requests for the establishment of a new session is modeled by a Poisson distribution with arrival rate $\lambda_1, \lambda_2, \lambda_3 \ldots\ldots \lambda_k$ respectively. The total arrivals of the requests follow a Poisson distribution with a rate

$$\lambda = \sum_{i=1}^{k} \lambda_i$$

For any arriving request, if the number of logon users is already at the maximum M or resources are not available to process the job belonging to new user, the arriving request is buffered in the waiting queue until the server become free.

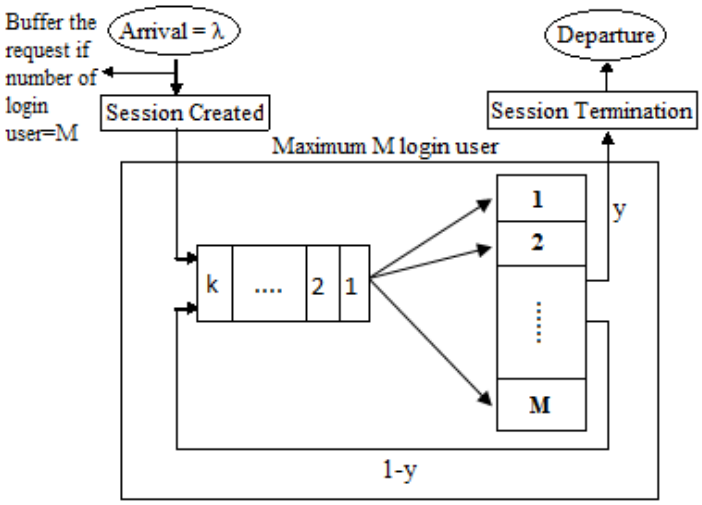

Figure 3. Proposed model based on queue

When a new user (CC) logon to the system, a new session is created. The arrivals of user requests at the data center follow a Poisson distribution with arrival rate λ. For each arriving request for the establishment of a new session, if the number of logon users is already at the maximum M, the arriving request is buffered in the waiting queue otherwise a new session is created and all requests join a single queue. As shown in Figure 3, k login users (CC) at their workstations interact with a service facility containing M servers. A login user at the user terminal submits a job to CSP. Jobs submitted by logon users in the form of matrix follows Banker's algorithm that allocate resources dynamically. When a login user submits a job to the CSP, it declares the maximum number of instances that are needed. This number cannot exceed the total number of resources in the system. If the process can be accommodated based upon the needs of the system, then resources are allocated, otherwise the process must wait. These requests correspond to jobs to be processed by the processor nodes in the server clusters. There are M identical servers, each of which represents a processor node.

Here λ is mean number of arrivals per unit time period and M is number of servers. If $\lambda > M$, then waiting line will be formed and will increase with the passage of time. If $\lambda \leq M$, there will be no waiting line. The server cluster can be viewed as a service facility that has a pool of resources. Suppose there is a probability y of leaving the session, after a job completes service at a server. This represents the termination of a user session and this information is send to the CA that keeps the record of number of login users. It has probability 1-y of returning to the user terminal. That indicates that the login user will continue with his session. The probability of session termination y = 0 means session will not terminate and 1-y i.e. 1-0 = 1, means user wants to continue current session.

The resources under consideration are processor nodes and are referred as server nodes. This service facility provides a full server utility model, where a server node is dedicated to run one application for one user at any instance of time. This system architecture allows resource allocation to be done dynamically. The number of server nodes allocated to an application can change over time in response to changing workload.

4. Mathematical and Experimental Results

In this Section, we show the results for the server utilization of proposed model with respect to request rate. We use three variables to evaluate the system performance. These variables are observation time (T), busy time (B) and completion (C). The observation time (T) is the amount of time that the server is monitored for task. Busy time (B) is the amount of time that the server remains active during the observation time. Completion (C) is the number of requests completed during the observation period. With these three variables, we can calculate the six significant values. i.e.

CPU Utilization (U): The percentage of CPU capacity used during a specific period of time i.e. U =

B/T. Request throughput of the system (X): The average number of request completed during a specified period of time i.e. X = C/T. Average service time (S): The average time to complete a request. i.e. S = B/C. Request capacity of the system (Cp): The number of requests the server can handle. i.e. Cp = 1/S. Average queue length (Q): The average number of transactions in queue i.e. Q = U/ (1-U). Average response time (R): The average time required to respond to a request (R). i.e. R = [(Q-1)*S) + S]/2 = (Q*S)/2

Table 1. Server utilization for different request processing rates

Processing Time (Sec)	Server Utilization	Response Time (Sec)
48	80 %	1.06
50	83.3 %	1.41
52	86.7 %	1.93
54	90 %	2.7
56	93.3 %	4.13

Figure 4 and Figure 5 show the response time and server utilization with respect to request processing time.

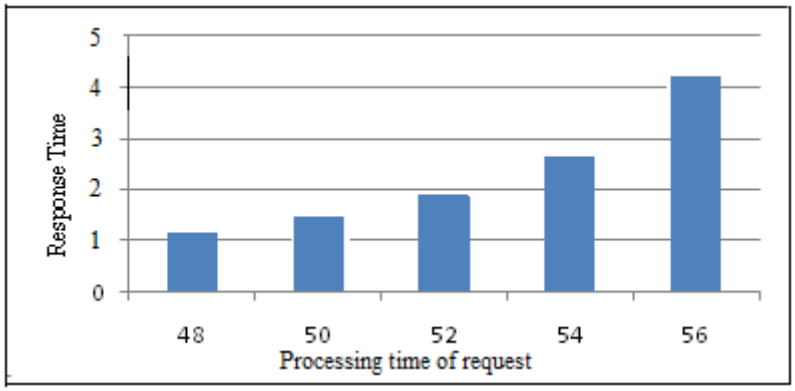

Figure 4. Response time with respect to processing time of request

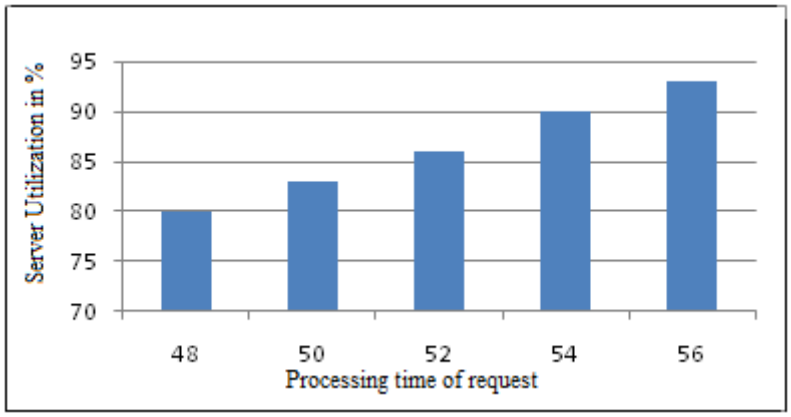

Figure 5. Server utilization with respect to processing time of request

The proposed technique is analyzed and implemented on cloud computing simulator named Hadoop. Figure 4 shows that the status of response time with respect to processing time of request. By performing various calculations, it has been found that the proposed model is very well promising in predicting the actual pattern. Here is an example of how to use these parameters to evaluate server utilization. The server observation time (T) is 60 seconds during which there are 90 completed requests (C) and the server actually busy time for processing the request is 48 seconds (B).

The CPU Utilization (U) comes out to be 48/60 = 80 percent utilization, Request throughput of the system (X) is 90/60 = 1.5 requests/sec, Average service time (S) is 48/90 = .53 seconds, Request capacity of the system (Cp) is 1/.53 = 1.875 requests/sec, Average queue length (Q) is .8/(1 - .8) = 4 requests and Average response time (R) comes out to be (4*.53)/2 = 1.06 seconds.

Table 1 shows the server utilization for different request processing time computed in the same fashion as explained before.

5. Conclusion and Future Work

The work presented in this paper provides a valuable result of resource allocation methodology for a cloud computing infrastructure. A new approach is proposed in which cloud customer (CC) establishes the session to access the resources. The numbers of login users keep on changing over time when a new session is created and existing session is terminated. The proposed model uses banker's algorithm for resource allocation. That mean, there is no possibility of deadlock. Also by restricting the number of login users, resources are not choked out even in case of heavy demand of resources. Moreover, resource allocation matrix specifies the requirement of resources in advance to run that job. Proposed model is an effective model that is efficient from other related existing dynamic resource provisioning model. Our model provides better response time to each request in real time interactive applications. Simulation of proposed model shows that results are good for dynamically allocation of resources. Proposed model is simulated for interactive applications only. Model can be extended to other cloud application. Further, security is very important aspect [18] [19] and can be studied for this model.

Acknowledgment

This work is supported by University Grant Commission, India under major research project entitled ''Trust based proactive resource provisioning in cloud'' wide letter no. 40-255/2011 dated 29 June, 2011.

REFERENCES

[1] A. Rahimli, "Factors Influencing Organization Adoption Decision On Cloud Computing," in *International Journal of Cloud Computing and Services Science (IJ-CLOSER),* vol. 2, no. 2, pp.140-146, 2013.

[2] V. Goswami, et al., "Optimization of QoS parameters through flexible Resource Scheduling in Finite Population Cloud Environment," in *International Journal of Cloud Computing and Services Science (IJ-CLOSER),* vol. 2, no. 3, pp.170-183, 2013.

[3] S. K. Sood, "A Combined Approach to Ensure Data Security in Cloud Computing," in *Journal of Network and Computer Applications*, Elsevier Ltd**,** vol. 35, no. 6, pp. 1831-1838, 2012.

[4] R. P. Doyle, et al., "Model-based Resource Provisioning in a Web Service Utility," in *USENIX Symposium on Internet Technologies and Systems,* vol. 4, pp. 5-5, 2003.

[5] M. N. Bennani and D. A. Menasce, "Resource Allocation for Autonomic Data Centers using Analytic Performance Models," in *International Conference on Autonomic Computing. 2nd IEEE,2005,* pp. 229-240.

[6] G. Tesauro, et al., "Utility-function-driven Resource Allocation in Autonomic Systems*,"* in *2nd International Conference on Automatic Computing*, pp. 342-343, 2005.

[7] B. Urgaonkar and A. Chandra, "Dynamic Provisioning of Multi-tier Internet Applications*,"* in *2nd International Conference on Autonomic Computing*, pp. 217-228, 2005.

[8] M. Woodside, et al., "Service System Resource Management based on a Tracked Layered Performance Model," in *International Conference on Automatic Computing. IEEE*, 2006, pp. 175-184.

[9] Q. Zhang, et al., "A Regression-based Analytic Model for Dynamic Resource Provisioning of Multi-tier Application," in *4th International conference on Autonomic Computing, 2007*, pp. 27-36.

[10] P. Padala, et al., "Adaptive Control of Virtualized Resources in Utility Computing Environments," in *European Conference on Computer Systems. 2007*, pp. 289-302.

[11] H.Ye, et al., "Resource Provisioning for Cloud Computing," in *Conference of Center for Advanced Studies on Collaborative Research. 2009*, pp.101-111.

[12] A. G. Fayoumi, "Performance Evaluation of a Cloud based Load Balancer Severing Pareto Traffic," in *Journal of Theoretical and Applied Information Technology,* vol. 32, no.1, pp. 28-34, 2011.

[13] X. Wang, et al., "An Adaptive Model-free Resource and Power Management Approach for Multi-tier Cloud Environments," in *Journal of System and Software,* vol. 85, no. 5, pp. 1135-1146, 2012.

[14] R. K. Grewal and P. K. Pateriya, "A Rule-based Approach for Effective Resource Provisioning in Hybrid Cloud Environment," in *International Journal of Computer Science and Informatics,* vol. 1, no. 4, pp. 101-106, 2012.

[15] L. Wu, et al., "SLA-based Admission Control for a Software-as-a-service Provider in Cloud Computing Environments," in *Journal of Computer and System Sciences,* vol. 78, no. 5, pp. 1280-1299, 2012.

[16] P. Koperek and W. Funika, "Dynamic Business Metrics-driven Resource Provisioning in Cloud Environments," in *Parallel Processing and Applied Mathematics, LNCS, Springer-Verlag*, vol. 7204, pp. 171-180, 2012.

[17] S. K. Sood, "A Value based Model for Dynamic Resource Provisioning in Cloud Enviournment," in *International Journal of Cloud Applications and Computing*, vol. 3, no. 1, pp. 1-12, 2013.

[18] S. K. Sood, "Secure dynamic identity-based authentication scheme using smart cards," in *Information Security Journal: A Global Perspective,* vol. 20, no. 2, pp. 67-77, 2011.

[19] A. Khalique, K. Singh, S. K. Sood, "Implementation of elliptic curve digital signature algorithm." in *International Journal of Computer Applications,* vol. 2, no. 2, 2010.

BIOGRAPHY AUTHOR

Dr. Sandeep K. Sood done his Phd in Computer Science & Engineering fron IIT Roorkee. He completed his M.Tech, Computer Science & Engineering, from G.J.U, Hisar. He is currently working as Associate Dean (A.A. & S.W), Head & Associate Professor, Computer Science & Engineering, G.N.D.U. Regional Campus, Gurdaspur. He has 14 years of teaching and 6 years of research experience. His research areas are Network & Information Security (Password based Authentication Protocols) and Resource Provisioning & Security in Cloud Computing.