Load Sharing for Cloud Processors using RPF and Genetic algorithm Concepts

R.Sunitha , Assistant Professor, CMS College of science and commerce,Coimbatore,Scholar of Bharathiar university , and
Dr. Antony Selvadoss Thanamani, Head and Reader in Computer Science, NGM College , Pollachi.

Abstract

Load-sharing methods are the task of broadcasting Internet traffic over multiple links or paths within the network. We proposed a novel schema for a packet transaction in which the router is used, which provides load sharing among cloud processors for various traffic patterns. In cloud computing mainly the virtual machine (VM) is used for scheduling the resources at cloud environment solves imbalance load sharing problem. In general cloud computing environment the scheduled target is VM resources granularity is large and the transferred data is large, deployment time T of VMs, the time of scheduling algorithm can almost be unused. In solving the load balancing problem in VM resource scheduling, initially the load is transferred to the virtual machine (VMS), the scheduling process is done. In this case the scheduling consumes high memory capacity as it is not assigned, according to the low memory processor. To overcome these problems we use Load balancing with the relative performance score. In Relative performance score is calculated only once in the whole of the process, optimization result is less, we enhance our work to a Genetic algorithm to find the best fitness value for the number of the repeated iterations in a cloud processor to allocate the resources to each virtual machine.

Index Terms—*load sharing, Virtual Machines, Cloud Processor.*

1. Introduction

Cloud computing, a framework for changing convenient, and on demand network access to a shared pool of computing resources [1] is emerging as a new paradigm of large-scale distributed computing [2]. In most cloud processor systems, the work generated at a processor is processed there; trivial sharing of computational resources is allowed, in such systems it is possible for some processor to be hard loaded while others are thin loaded, resulting in poor overall system performance [9]. The aim of load sharing is to better performance by spreading the workload among the processor. The load sharing policies with the largest potential benefit are adaptive in the sense that they react to changes in the system state. Adaptive policies can range from easy to complex in their learning and use of system state information. Load Sharing tries to mend the performance of a cloud processor system by using the processing power of the whole system to “smooth out" periods of high over-crowding at single nodes [10]. This acts of moving a few of the workload of a closed node to other nodes for processing. The possible attraction of load sharing is increased by factors such as the increasing size of cloud processor systems, the expend of shared file servers, the presence of pools of computation servers, and the development of clean communication protocols [1]. The two significant elements of a load sharing policy are the transfer policy, which finds whether to process a task locally or remotely, and the location policy, which finds to which node a task selected for transfer should be sent. Apply policies that use only information about the average behavior of the system, dismissing the current state, are termed static policies. Static policies may be either deterministic or probabilistic. Policies that respond to the system state are named adaptive policies. In the primal establishment of the problem it was accepted that information about the average execution times and intercommunication requirements of all tasks were known. We propose a scheduling scheme on load balancing of VM resources based on optimization algorithm. According to historical data on various traffic patterns in cloud processor and through genetic algorithm proposed strategy achieves the best load balancing and reduces or avoids the dynamic migration. This scheme clears the problem of load inequality and huge movement cost by traditional algorithms after scheduling. The tested results show the performance of the system in terms of the cost, computation time and load. Tested results show that this method is able to understand load balancing and sensible resource usage of both when system load is stable and variant.

Consider a script where we only have one web server in operation to deal all incoming requests to your company website. When the business is being based, it may be potential to handle the volume of traffic your site receives with one web server. However, as the business grows, the one the server will no longer be sufficient [17]. If we don't add new web server instances our WebPages will load slowly and you will have users waiting till the server is free to process client requests. This is not a good thing as people are not very good at playing the waiting game. In these days where the market is very competitive customers service is hugely important or potential customers will move to the competition. When you have several web servers in your server group, the approaching traffic can be evenly shared among the different servers. However, it will only appear to the client as one server only rather than several. A case in point is the internet browser. The purpose includes:

• To reach the load between a number of machines/locations

• To supply redundancy in case one machine/server fails

• To provide zero downtime during patch installations on servers or updates to applications on the server [17].

2. Related Works

Cloud computing, a framework for changing convenient, and on demand network access to a shared pool of computing resources [1] is emerging as a new paradigm of large-scale distributed computing [2]. It has universally been approved by the industry, still there are various real problems according to Energy Management, Virtual Machine Migration, Server Consolidation, Load Balancing, etc. that are not totally addressed [3]. Important to these problems is the issue of load balancing that is a mechanism to assign the dynamic workload equally to all the nodes in the whole cloud to reach a high user satisfaction and resource utilization ratio [1]. It serves in keeping bottlenecks of the system which may occur due to load inequality. When one or two many components of any benefits fail, load balancing helps continuing of the benefits of achieving fair-over, i.e. its support in provisioning and deprovisioning of instances of applications beyond fail. It additionally assures that whole computing resource is shared well and enough [3] [8]. Resources consumption and power conservation is not constantly a best center of analysis of cloud computing. However, resource consumption can be stored to a minimal with meet load balancing which not only use in reducing costs but producing enterprises greener [4] [7]. Scalability, one of the perfect significant appearance of cloud computing, is also approved by load balancing. Hence, developing resource use and the performance of a distributed system in such a way will reduce the energy consumption and carbon footprints to achieve Green computing [5] [6].

Virtual machine allows the abstraction of an Operating System and Application flows on it from the hardware. The inner hardware infrastructure services completing to the Clouds are designed in the simulator by handling the service requests for Datacenter element. The application elements are requests these sandboxes within VMs, which require to be allocated a Datacenter's host component to share of processing power. The data center management activities are managed by Datacenter object such as VM creation and deletion and does the routing of user requests received from User Bases via the Internet to the VMs the Data Center Controller [11], uses a VmLoadBalancer to determine which VM should be assigned the next request for processing. The majority of common Vmloadbalancer is limited and active monitoring load balancing algorithms [13].

Cloud vendors are established for automatic load balancing services, which permitted entities to gain the number of CPUs or memories with the gained demands for their resources to scale [14]. This service is optional and depends on the entity's business needs. Therefore load balancers served two significant necessitates, primarily to promote the availability of cloud resources and secondarily to promote performance. Allowing to the previous section Cloud computing will apply the dynamic algorithm, which permits cloud entities to advertise their existence to presence servers and also supplies the means of communication between concerned parties [15].

Load balancing is a comparatively new technique that helps for networking and resources are provided with a maximum throughput and minimum response time [15]. Separate the traffic between the servers, data can be sent and received without delay. To serve the traffic loaded between available servers, there are different kinds of algorithms are available. The sample of load balancing in our day to day life can be linked to websites. Absents of load balancing means the users could know the delays, timeouts and long time to taken for system responses. Load balancing is mainly answers to the redundant servers which serve to improve the distribution of communication traffic, so that the website availability is positively decided [15].

Load Balancing is a procedure for disseminating the workload throughout more than one network interface, servers, hard drives, or other computing resources. Usually the data center utilization relies on large, powerful (and expensive) computing hardware and network infrastructure, which are the fields are having the common risks associated with any physical device, containing power and/or network interruptions, hardware problem, and resource limitations in times of high demand. Load balancing in the cloud distinct from classical thought of load-balancing architecture and implementation by utilizing specialty servers to execute the load balancing [16]. This supplies a new chance and economies-of-scale, as well as introducing its own unique set of challenges. Balance the load distribution is can move the load from the origin nodes to the relative easily loaded destination nodes. Appling the load balancing concepts in runtime, it is known as dynamic load balancing, this can be made for direct or repetitive way allowing for the performance node selection:

• In the repetitive methods, the destination node is decided by the various iteration steps.

• In the direct procedure, the final destination node is selected in one step.

Another form of Load Balancing procedure can be used for the Randomized Hydrodynamic Load Balancing method, its hybrid method that carries the advantage of both direct and repetitive methods [16].

3. Proposed System

Adaptive load sharing schema doesn't support various traffic patterns which influence the performance of the load-sharing scheme. So we proposed a novel schema for a packet transaction in which the router is used, which provides load sharing among cloud processors. In cloud computing mainly the virtual machine (VM) is used for scheduling the resources at cloud environment solves imbalance load sharing problem. In general cloud computing environment the scheduled target is VM resources granularity is large and the transferred data is large, deployment time T of VMs, the time of scheduling algorithm can almost be unused. In solving the load balancing problem in VM resource scheduling, initially the load is transferred to the virtual machine (VMS), the scheduling process is done. In this case the scheduling consumes high memory capacity as it is not assigned, according to the low memory processor. To overcome these problems we use Load balancing with the relative performance score. In Relative performance score is calculated only once in the whole of the process, optimization result is less, we enhance our work to a genetic algorithm to find the best fitness value for the number of the repeated iterations in a cloud processor to allocate the resources to each virtual machine. However, since adaptive policies must collect and react to system state information, they are necessarily more complex than static policies. The adaptive policies that have been examined in the literature collect considerable state information and attempt to make the "best" choice possible based on that information. Consider the applicable level of involvement of adaptive load sharing policies, we revolve a set of abstract policies that shows only the fundamental aspects of load sharing, and we research these policies working simple analytic models. Our aim is not to resolve the absolute performance of specific load sharing policies, but rather to evaluate the respective advantages of varying degrees of sophistication. By representing only the important appearance of load sharing and removing secondary details, we are better able to interpret the results of our comparative analysis and so build our intuition. Fig 1 shows the load balancing in normal cloud processing.

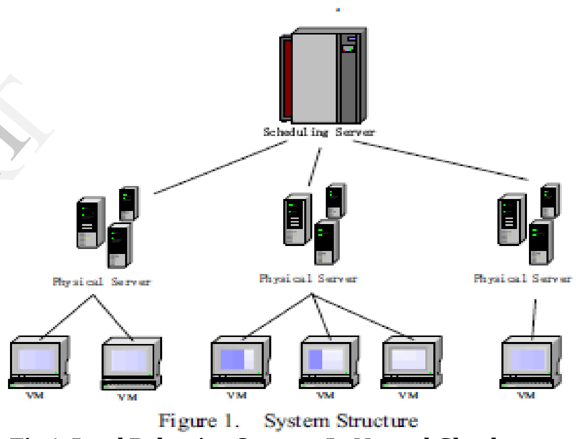

Fig 1. Load Balancing Statergy In Normal Cloud Processing

4. Load sharing among cloud processors – RPS

Adaptively balancing the workload between the router and the cloud processor with the relative performance score. Fig 2 shows the adaptive load sharing for cloud processor.

A Relative Performance Score for a given application is a measure of the relative distance of the application's performance. RPS of the cloud processor will depend on each particular workload. RPSs are calculated for both transactional applications and long-running jobs in our system. The performance of the system can then be measured as an ordered vector of application performance values.

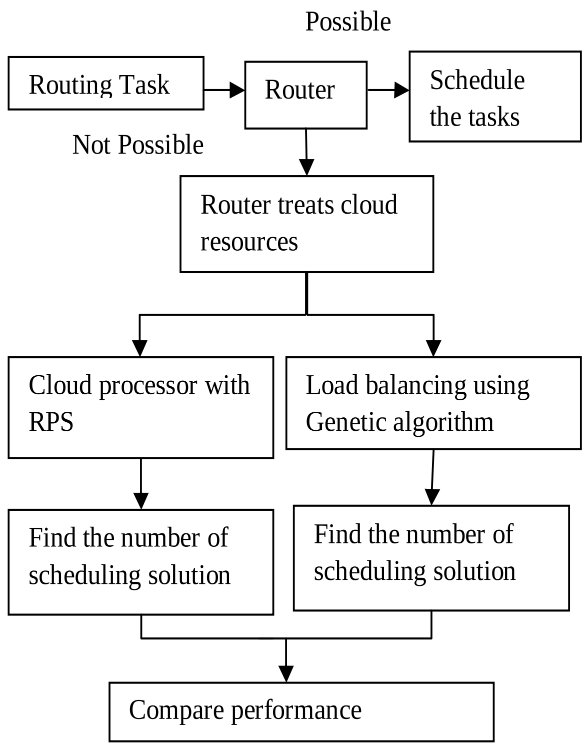
Possible
Routing Task
Router
Schedule the tasks
Not Possible
Router treats cloud resources
Cloud processor with RPS
Load balancing using Genetic algorithm
Find the number of scheduling solution
Find the number of scheduling solution
Compare performance

Fig 2. Process of Load Sharing

(i) Steps to calculate RPS

1. Initialize the application placement matrix P and a load distribution matrix L, a relative performance score value can be calculated for each application.

2. Find the best possible new placement of applications as well as a corresponding load distribution that maximizes objective.

$$\max_{m} \min \mu_m \omega_m \ \ s.t$$
$$\forall_m \ \omega_m = \textstyle\sum_n L_{m,n},$$
$$\forall_n \sum_m L_{m,n} \le \Omega_n,$$
$$\forall_n \sum_m P_{m,n} \gamma_m \le \Gamma_m,$$

3. Calculate response time t_m with each transaction application and we evaluate the system performance with respect to objective function u_m

$$\mu_m(t_m) = \frac{\tau_m - t_m}{\tau_m},$$

4. Define resource requirements of a job and is given at job submission time in the real system

5. The resource usage profile is estimated based on historical data. Each job m consists of a sequence of N_m stages, $s_1; \ldots ; s_{N_m}$, where each stage s_k.

5.1 The amount of CPU cycles consumed as a stage, $\propto_{k,m}$

5.2 The maximum speed with which the stage may run $\omega_{k,m}^{max}$

5.3 The minimum speed with which the stage must run, whenever it runs $\omega_{k,m}^{min}$.

5.4 The memory requirement $\gamma_{k,m}$.

6. If they τ_m should be greater than the job's desired start time, τ_m^{start} the job is completed.

7. RPS that maps actual job completion time t_m to a measure of satisfaction by using $\mu_m(t_m)$

$$\mu_m(t_m) = \frac{\tau_m - t_m}{\tau_m - \tau_m^{start}}$$

8. Estimate the completion time needed to achieve a relative performance score you

$$\omega_m(u) = \frac{\alpha_{N_{m,M}}^{cr}}{t_m(u) - t_{now}}$$

$\alpha_{N_{m,M}}^{cr}$ as the remaining work to complete all stages up to the stage $N_{m,M}$

9. The relative performance of u for all jobs, the aggregate allocation of all jobs must be

$$\omega_g = \sum_m \omega_m(u)$$

$\omega_m(u)$ For various values of u and interpolate values between the sampling points.

10. In each round first computes the maximum total application demand at step 2 that can be satisfied with the current placement solution.

11. Repeatedly and incrementally optimizes the placement solution in multiple rounds.

5.Genetic Algorithm based Load Sharing

We proposed a scheduling strategy on load balancing of VM resources based on genetic algorithm. According to historical data on various traffic patterns in cloud processor and through genetic algorithm, proposed strategy achieves the best load balancing and reduces or avoids the dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling. The experimental results show the performance of the system in terms of the cost, time and load [3]. Experimental results prove that this method is able to realize load balancing and reasonable resource utilization both when system load is stable and variant. Workload requires different control mechanisms for managing. In cloud computing transactional workloads are managed by using flow control, load balancing, and application placement.

Genetic algorithm (GA) is a computational model inspired by evolution. This algorithm encodes a specific problem with several attributes on a simple chromosome-like data structure, and generates a population of chromosomes randomly as the initial colony. Then it applies operators - selection operator, crossover operator and mutation operator - to these structures to get an optimal solution evaluated by a fitness function.

VM resource scheduling in cloud computing with the genetic algorithm. Proposed system performs a balanced scheduling strategy of VM resources based on genetic algorithm. Starting from the initialization in a cloud computing environment, Find the best scheduling solution by genetic algorithm in every scheduling. VM resources in the whole system use the algorithm to choose the scheduling solution according tothe computed selection probability; with the increase of VM resources and the increase of running time. VM service resources arranged to every physic node, and then choose the best solution. The workflow of GA is shown in Fig 3.

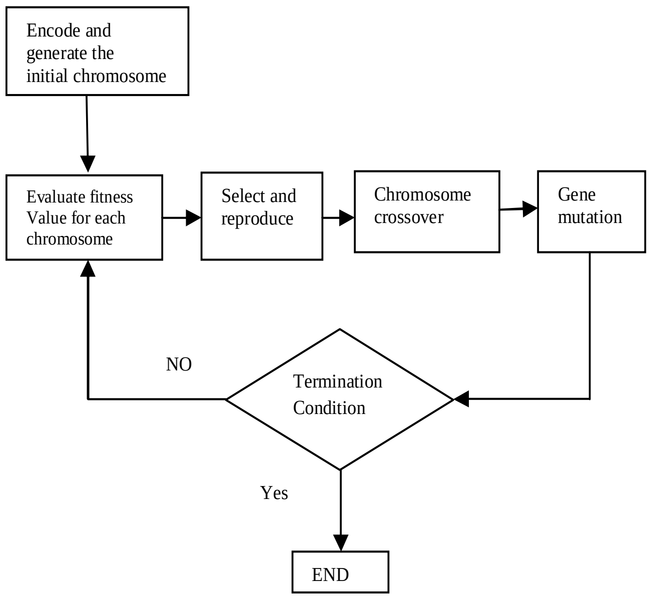
Encode and generate the initial chromosome
Evaluate fitness Value for each chromosome
Select and reproduce
Chromosome crossover
Gene mutation
NO
Termination Condition
Yes
END

Fig 3. Genetic Algorithm Workflow

(i) STEPS TO PERFORM BALANCED SCHEDULING STRATERGY USING GENETIC ALGORITHM

Step 1: In initialization, there are not any VM resources in the system so there is no historical information. When there are VM resources to be scheduled, based on the algorithm randomly chooses the free physical machine and starts scheduling;

Step 2: Increase of VM resources in the system and the increase of running time, according to historical information and the current state. We compute the selection probability of the individuals according to their fitness values.

Step 3: The algorithm computes the load and variance of every physical machine in every solution from the scheduling solution set *S*.

Step 4: The algorithm uses genetic algorithms to compute the best mapping solution for every solution in *S* . The best solution refers to the one in which the variance meets the predefined load constraints;

Step 5: The algorithm computes respectively the costs or cost divisors of every solution in *S* to achieve the best mapping solution;

Step 6: According to the cost divisor of every solution, the algorithm chooses the one with the lowest cost as the scheduling solution and completes the scheduling;

Step 7: New VM resources need scheduling, then go back to step 2

6. Experimental Setup

The proposed algorithm implemented through simulation packages like CloudSim and cloud sim based tool. Java language is used for implementing the VM load sharing algorithm. Parameter Values are as under in Table 1.

Table 1: Parameter Value

Parameter	Value
Data Center OS	Linux
Data Center Architecture	X86
Service Broker Policy	Optimize Computation Time
Physical H/W units (physical hosts)	3
No. of VMs	6

Initially in the physical host we create 6 numbers of VMs and Each VMs configuration as shown in Table 3.

Detailed Results with the existing and proposed system are shown in Table 2.

Table 2. Detailed Results

Computation Time With Adaptive Load Sharing Algorithms		
Computation Time	Proposed System (ms)	Existing System (ms)
	450	700
Cost with Adaptive Load Sharing Algorithm		
Cost	Proposed System	Existing System
	22130.2	25630.2
Load with Adaptive Load Sharing Algorithm		
Load	Proposed System	Existing System
	96977.2	109387.912

Table 3. VMs Configuration

Id	Memory (MB)	Storage (MB)	Available BW (MB)	No.of processors	Processor Speed (MIPS)	VM Policy
0	512	1048576	700	1	400	TIME SHARED
1	3056	4194304	1000	2	500	TIME SHARED
2	1024	2097152	1000	1	350	TIME SHARED
3	512	1048576	800	1	250	TIME SHARED
4	3056	4194304	1250	3	400	TIME SHARED
5	1536	2597152	700	2	250	TIME SHARED

7. Simulate Experiment

Comparing the results of relative performance score and ALS (GA) in Computation Time, Load and Cost.

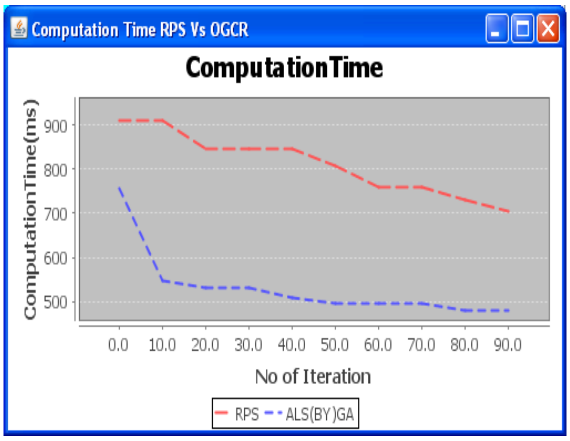

Fig 4. Computation Time of RPS Vs ALS (GA)

In this graph we measure the computation time for RPS and ALS(GA) system , when compared to computation time results the ALS(GA) system is lower than the RPS.

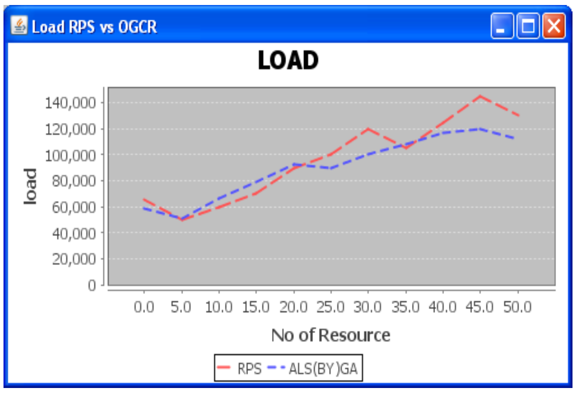

Fig 5. Load of RPS Vs ALS (GA)

In this graph we measure the load for RPS and ALS(GA) system , when compared to load results the ALS(GA) system is lower than the RPS.

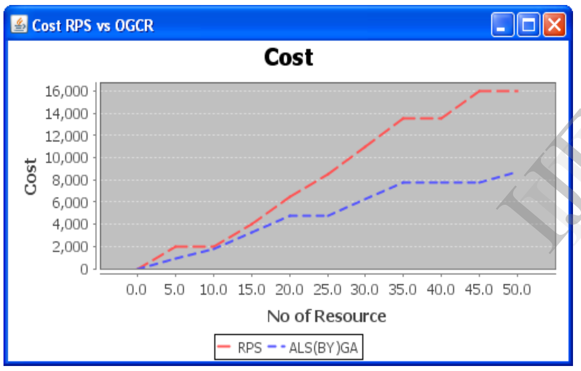

Fig 6. Cost of RPS Vs ALS (GA)

In this graph we measure the cost for RPS and ALS (GA) system , when compared to cost results the ALS (GA) system is lower than the PRS.

8. Conclusion

The proposed system is scheduling strategy on load sharing of VM resources based on genetic algorithm. According to historical data on various traffic patterns in cloud processor and through genetic algorithm, this scheduling strategy computes better than the RPS strategy and proposed strategy achieves the best load balancing and reduces or avoids the dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling.

References

[1] Zhang Z. and Zhang X. (2010) 2nd International Conference on Industrial Mechatronics and Automation, 240-243.
[2] Pallis G. (2010) IEEE J. of Internet Computing, 14(5), 70-73.
[3] Rimal B.P., Choi E. and Lumb I. (2009) 5th International Joint Conference on INC, IMS and IDC, 44-51.
[4] Mata-Toledo R. and Gupta P. (2010) Journal of Technology Research, 2(1), 1-8.
[5] Kabiraj S., Topkar V. and Walke R.C. (2010) International Journal of Managing Information Technology, 2(3), 22-31.
[6] Baliga J., Ayre R.W.A., Hinton K. and Tucker R.S. (2011) IEEE, 99(1), 149-167.
[7] Alakeel A. M. (2010) International Journal of Computer Science and Network Security, 10(6), 153-160.
[8] Rimal B.P., Choi E. and Lumb I. (2010) Computer Communications and Networks, 21-46.
[9**]** Derek L. Eager, Edward D. Lazowska and John Zahorjan," Adaptive Load Sharing in Homogeneous Distributed Systems", Proc. IEEE Transactions on Software Engineering, Vol. SE-12, No.5, pp. 662-675, May 1986.
[10] Anthony Karageorgos and Helen Karatza," Performance Issues of Task Routing and Task Scheduling with Resequencing in Homogeneous Distributed Systems", Proc. IEEE Transactions, pp. 56-63, Mar 2009.
[11] Bhathiya Wickremasinghe, Rodrigo N. Calheiros, Rajkumar Buyya,"CloudAnalyst: A CloudSim-based Visual Modeller for Anal ysi ng Cl oud Comput i ng Envi ronment s and Applications", 20-23, April 2010, pp. 446-452.
[12] Jinhua Hu, Jianhua Gu , Guofei Sun and Tianhai Zhao," A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Proc. IEEE Conference Publications, pp. 89 - 96, Dec 2010.
[13] Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma," Efficient Load Balancing Algorithm in VM Cloud Environment", Proc. IJCST, Vol. 3, pp. 439-441, Mar 2012.
[14] G. Reese. Cloud Application Architectures. O'Reilly Media, 1st Ed., Inc. Sebastopol, CA, US (2009).
[15] Zenon Chaczko , Venkatesh Mahadevan , Shahrzad Aslanzadeh and Christopher Mcdermid," Availability and Load Balancing in Cloud Computing", Proc. International Conference on Computer and Software Modeling, vol.14, pp. 134-140, 2011.
[16] Ratan Mishra and Anant Jaiswal," Ant colony Optimization: A Solution of Load balancing in Cloud", Proc. International Journal of Web & Semantic Technology (IJWesT), Vol. 3, No.2, pp. 33- 50, April 2012.
[17] Venubabu Kunamneni," Dynamic Load Balancing for the Cloud", Proc. International Journal of Computer Science and Electrical Engineering (IJCSEE), Vol-1, Iss-1, pp. 33-37, 2012.