

Modified Active Monitoring Ant Clustering based Load Balancing over Public Clouds

Sonam Raghuwanshi
Department of CSE
Samrat Ashok Technological Institute
Vidisha, (M.P.), India

Rashmi Nigoti
Assistant Professor, Dept of CSE
Samrat Ashok Technological Institute
Vidisha, (M.P.), India

ABSTRACT

Cloud computing is a technique of sharing resources over Internet, where Users Can Share or Store Resources and Data over Data Centers processed by Virtual Machines. During the Access of Resources over Data Centers requires load to be balanced by Virtual machines. The Existing Naïve Bayes Clustering is an efficient technique for Load Balancing over Cloud, but the existing methodology takes low Throughput and Make Span Time. Hence a new and efficient technique is implemented for Load Balancing over Public Clouds using Modified Active Monitoring based Ant Clustering. The Proposed Methodology implemented provides High Throughput and Make Span time as well as low Standard Deviation in Comparison with the Existing Naïve Bayes Load Balancing.

Keywords

Cloud Computing, Virtual Machines, Load Balancing, Naïve Bayes Clustering, Ant based Clustering, Active Monitoring.

1. INTRODUCTION

Cloud computing provides qualified rapid growing development in modern years by contribution extremely accessible and scalable services. On the other hand, growing customer resource demand and viable market power the suppliers to make bigger their data centers that show the way to huge power consumption or concern more reasonable resource provision maps that degrades feature of service. For this reason, the cloud providers require the solutions to reduce energy costs and get better resource conditioning. Within the cloud infrastructure the speed at which this is implemented is considerable. It makes available repairs of infrastructure, platform and software for users and supplies the on-demand services to consumers through Internet [1]. Figure-1 features a general idea of cloud computing from the viewpoint of the three principle layers: the infrastructure layer; the platform layer; and the software layer. The infrastructure layer submits to the “nuts and bolts” layer of the cloud consisting of the virtual servers the network communications and the block storage methods. Within the structure of the infrastructure cloud one can reproduce their in-house server circumstances, structure an array of virtual servers and storage explanations to sustain their daily process. The consumer can then right to use their data, applications and resources anywhere in the globe across a series of devices (laptops, tablets, smart phones). Load complementary is a scheme of allocating the entirety weight to the disconnect nodes of the mutual group to the permit networks and assets to make progress the comeback with time of the job with maximum throughput in the scheme [2].

Infrastructure as a Service (IaaS) clouds rely on economies of range to distribute computational resources to users in a cost efficient method.

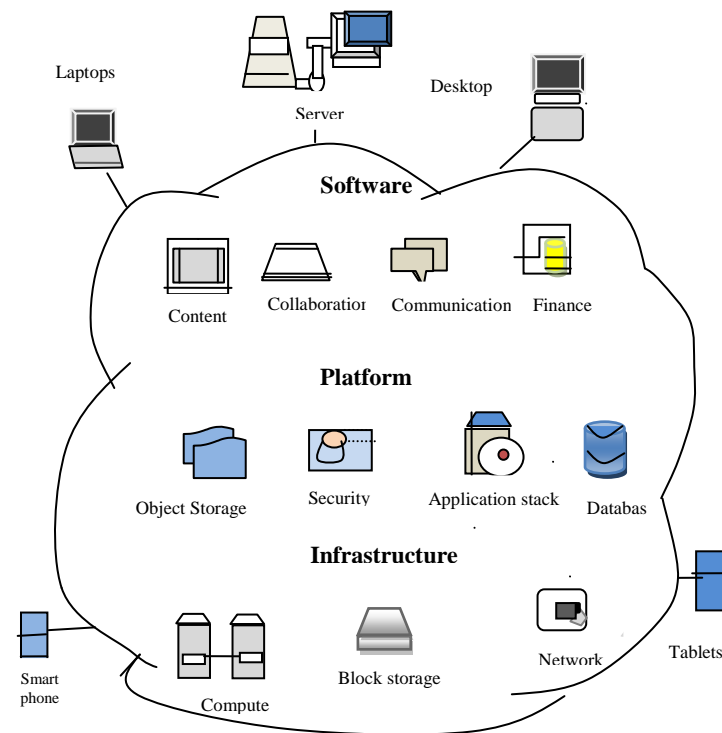


Figure1: Cloud Computing

Comparable to conventional utilities for example electricity and gas [3], consumers characteristically pay only for what they utilize conditioning funds as required in an on-demand method. This flexibility or capability to extent resources as need is one of the standard differences connecting existing computational clouds and earlier effectiveness computing forms for example computational grids and clusters, which need precede reservations. In distributing resources to users, IaaS providers use virtualization technologies for example Xen [4] and VMware [5] to separation a single substantial server into multiple independent Virtual Machines (VMs). These VMs exist in a co-located method and have no visibility or manage over the host environmental pattern or adjacent VMs. Figure-2 shows characteristic virtualization circumstances where various VMs are co-located on a particular physical host server. Depending on its configuration each VM is assigned a portion of the physical host resource i.e. CPU cycles, RAM, Disk and Network bandwidth. A Virtual Machine Monitor (VMM) is installed on the physical host and is responsible for controlling VM access to the host's resources. The VMM efforts to separate entity VMs with respect to security, failure and their relevant situation but not in respect of performance [6-7].

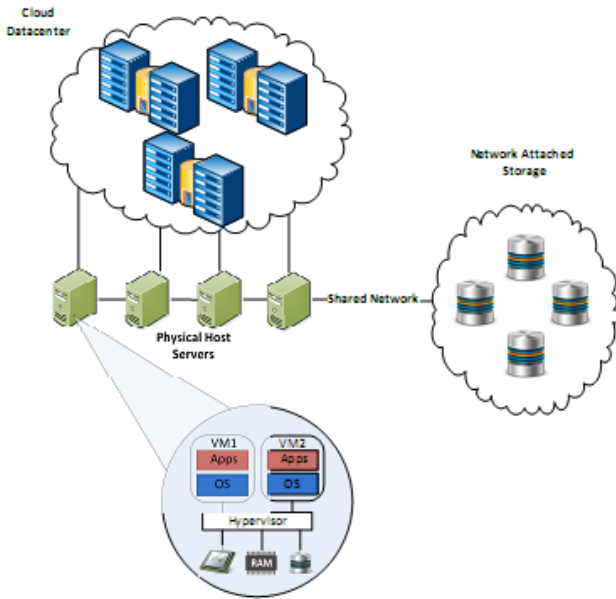


Figure 2: Physical host with three instantiated virtual machines

Infrastructure clouds influence leading progressions made from virtualization technologies to serving up huge physical server machines into smaller virtual servers which inhabit in a co-located method on the similar physical host. This procedure enlarges server exploitation by permitting unusual operating systems, applications and repairs to run side by side in visible separation. A domain level monitor recognized as a hypervisor manages right of entry to the fundamental hardware for example disk, network, CPU and memory communications, sharing them as impartially as possible between the machines. With the several advantages of cloud computing one may be forgive and forget for opinion that the cloud is comparatively difficulty complimentary and the long expected reply to our computing requires conversely there are reasonably a number of undecided problems which are measured barriers to the superior acceptance and enlargement of cloud computing.

Virtual machine movement appointment plan is an extensively utilized approach to accomplish load balancing of cloud computing data centers at present [8], [9]. VMware load balancing explanation is distributed resource scheduling (DRS) [10]. When DRS select the physical host for the virtual machine it will confirm the load position of each physical host and decide the appointment explanation which can recover the taken as a whole load balance amount. And in the procedure of running a virtual machine, DRS will continuously check the load position of the cluster and utilize VMware V Motion technology to achieve live movement of virtual machines between different physical servers. Consequently, it can make sure load balancing and well-organized use of physical resource of the complete cluster. In [11], piao and Yan have suggested a network-aware virtual machine assignment and migration move toward to reducing the data transfer instance use and improving the general application presentation of the cloud data center. On the other hand, this move toward almost certainly effects in a comparatively lower use of resource of physical hosts and increases the process expenditure of the cloud data center.

2. LITERATURE SURVEY

In this paper [12], here author proposed LB-BC method that is based on heuristic job deployment method, which is utilized to organize job requests accepted by the cloud data center into optimal objective physical hosts in the IaaS cloud computing

data center .Its algorithm is what joins Bayes theorem with clustering. It has accomplished the general load balancing of the complete arrangement from the viewpoint of cloud data centers' longtime operations and consequently to get better the presentation and effectiveness. The job deployment procedure of LB-BC is revealed in Fig. 3. Initially, these physical hosts each of which has a superior outstanding resource quantity than the maximum demanded resource quantity of all job requests can be investigated out to represent a novel applicant set to assemble the presentation restriction while manufacture LB-BC have the possible of accomplishing the durable load balancing. Then subsequent, the k physical hosts in the set of physical hosts can be observed as k objects staying for being clustered. Each physical host in the particular set is given to a prior possibility.

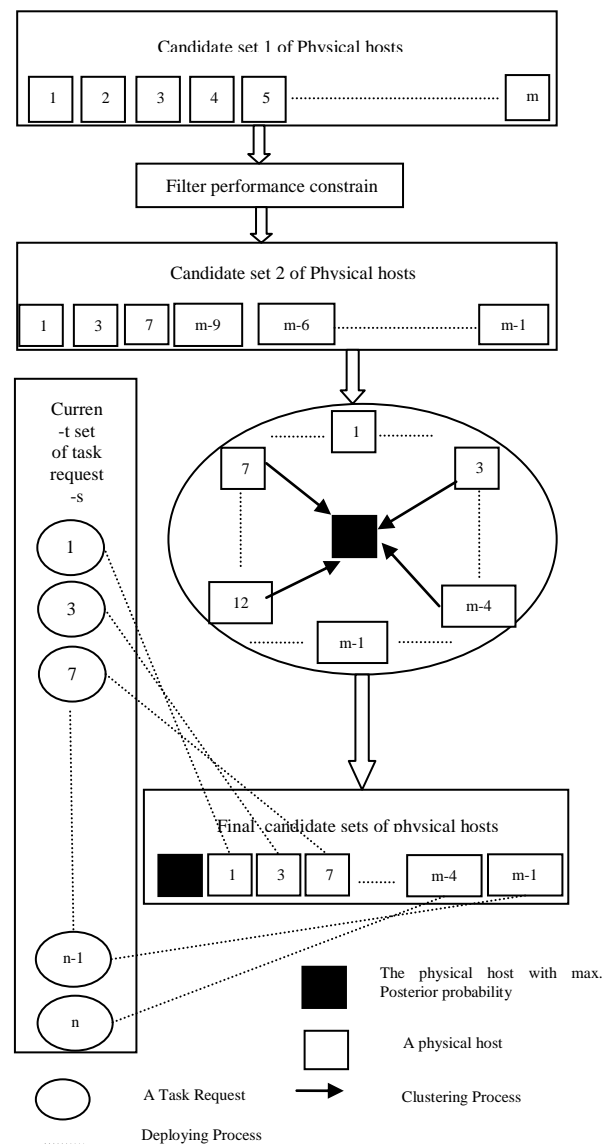


Figure 3: the process of LB-BC task deployment

The subsequent probability of each physical host's management jobs can be computed all the way through Bayes theorem. This possibility can be observed as a characteristic of each object while the continuing CPU resource quantity and the outstanding memory reserve quantity of each physical host can be observed as the other two attributes. The comparison degree assessments between physical hosts are computed according to the three

characteristics of each physical host. A threshold assessment is established on the origin of these comparison degree values. The physical hosts whose comparison degree values between them are surrounded by the given threshold can be observe as the optimal clustering to form the absolute set of applicant physical hosts. to end with, the jobs are positioned on the hosts in the finishing set. And the clustering process of physical hosts in the cloud data center is the progression of discovering the optimal physical hosts for executing jobs.

Another solution, Magnet [13], develops comparable schemes of contribution correlation between the nodes under the bounded node degree statement. On the other hand, Magnet is simply based on a composition place on top and cannot completely confine the correlation between contributions for it is bounded to one dimensional space where the arrangement superimpose is created. Also, Magnet is not as much of robust in volatile situations, such as the Internet. In compare, Visit is not confined to any dimension while capturing the contribution correlation (because clustering is done in an unstructured method) and as they demonstrate in our researches, it is very strong due to the fundamental gossip procedure.

Finally, there is modern effort for resource location in clouds [14], which can be understand as a publish/subscribe scheme, although with moderately understandable differences. In [14], nodes query for a reserve with confident attributes and are forwarded to a measurement of the cloud that holds the resources with demanded properties. This effort also uses a peer example repair to build an arrangement and an unstructured superimpose. In the unstructured superimpose, reserves with comparable attributes are positioned secure to one an additional. On the other hand, does not security and actually does not require that all the nodes with the uncertainty properties are establish. However, in V it is, they create confident that all the subscribers are originate and notified of the issued occasion. Additionally, is not appropriate for event distribution for it implements an important load on the nodes in the structured superimpose.

In [15], Wei et al. have utilized the weighted minimum link algorithm which signifies that unusual weights designate the presentation of the physical host. Afterward, the virtual machine will be assigned to the physical host which has the minimum ratio of the existing number and the influence. The benefit of static scheduling algorithms is that it is easy to exploit. But in the significant cloud data centers whose reserve heterogeneity is very well-built and consumer command isn't reliable the load balancing consequence is not perfect. The proposed LB-BC method is consequent the plan of dropping avoidable calculation difficulty just like the subsisting static methods. On the other hand, LB-BC is a well-organized method which has skill in enthusiastically accomplishing the lasting most favorable load balancing outcomes.

In [16], Lau et al. have integrated the two plans of serious load priority and light load priority. They have suggested an adaptive load distribution algorithm to efficiently decrease communication transparency of the load balancing procedure. Utilizing the greedy algorithms can explain the difficulty of load distribution. On the other hand, numerous algorithms above cannot assemble greedy choice presentation and the environment of optimal sub-structure simultaneously. So these load distribution methods often acquire the local optimal solutions. And the consequence of solving the difficulty of load distribution under definite special conditions is not perfect. Cloud datacenter cannot accomplish load balancing of the complete network.

Sonneck et al. in [17], have accessible a decentralized affinity-aware relocation method that fit ins heterogeneity and enthusiasm in network topology and work communication models to assign virtual machines on the accessible physical resources. The method check network affinity between pairs of virtual machines and employs a scattered exchanging algorithm, paired with relocation to enthusiastically regulate virtual machine placement such that communication transparency is reduced and load balancing is accomplished.

3. PROPOSED METHODOLOGY

Ant based Clustering is first proposed by Deneubourget. Al. and according to him the ants can walk randomly on the workspace by picking and dropping of data from one place to another, but the as soon as they move from one place to another a chemical named Pheromone is deposited which attracts other ants to move at the same place. Here we proposed an efficient technique of applying ant based clustering for the Optimization of VM Scheduling in the Cloud Computing Environment.

$T = \{t_1, t_2, \dots, t_n\}$ represents the tasks waiting to be scheduled per unit of time, n is the number of tasks.

$N = \{n_1, n_2, n_3, \dots, n_m\}$ represents the set of nodes in the cloud system, assuming that the cloud system has m nodes. For the cloud computing system, n_i represents the computing resources on the n_i ; For the Cloud Storage System, n_i represents the data on a n_i .

$V = [v_1, v_2, \dots, v_n]$ represents the task scheduling vectors or a scheduling scheme. For Cloud storage system, i v represents the i -th task of data is provided by resources nodes that represented by i v value, and the length of the vector is the total amount of scheduling tasks per unit time.

For example, a task scheduling vector [5, 1, 3, 2, 1, 6], the length of this vector is 6, and represents needs to schedule task number is 6 per unit of time. The value based on the position of No. 1 is 5, represent the data of the task 1 is provided by the system node 5. So, the data of task 2 and 5 are provided by the node 1; the data of task 3 is provided by the node 3; the data of task 4 is provided by the node 2; the data of task 6 is provided by the node 6. For cloud computing system, it is on behalf of a task placed in a node.

The Probability of Pick it up can be given by the following function:

$$P_{pick} = \left(\frac{K^+}{K^+ + f} \right)^2$$

Where, 'f' is an estimation of the fraction of nearby hosts occupied by objects of the same type, and K^+ is a constant. The Probability thus decreases with 'f', from 1 (when $f=0$) to $\frac{1}{4}$ (when $f=K^+$),

The Probability of Drop can be given by the following function; $P_{drop} = \left(\frac{f}{K^- + f} \right)^2$

Where, 'f' is same and K^- is also constant.

Initial pheromone laying

This is initialization step. Every location (i, j) with an object on the grid will be assigned a pheromone τ_{ij} based on the surrounding. Let $\Delta\tau$ be the amount of pheromone change. The presence of similar objects in the surroundings increases the pheromone trail on the location by $\Delta\tau$ and a dissimilar object decreases the trail by $\Delta\tau$.

Cluster construction

Ants move randomly on the grid. If an unloaded ant meets an object and finds pheromone on that location below the threshold value, it picks it up. If loaded ant comes to a location with pheromone value greater than the threshold and its load matches with the object on that location, it drops in neighborhood of location with $P_{drop} = 2\tau_{ij}$ probability.

Pheromone updation

On a pick/drop action, the pheromone on that location and the surrounding location will be updated. On Pickup, $\tau_{ij}=0$ and pheromone in the surrounding cells containing the similar object will be decreased and containing dissimilar objects will be increased. On Drop, $\tau_{ij} = \Delta\tau$ and pheromone in the surrounding cells containing the similar object will be increased and containing dissimilar objects will be decreased.

1. The Proposed Methodology implemented here contains a table of Virtual Machines id, State (Busy / Available), and the number of requests currently allocated to the VMs. Initially all VMs are in available State.
2. The DCC (Data Center Controller) receives a new request.
3. The DCC then queries the Load Balancer to allocate the upcoming request.

i.	Initialize all the VMs by some Constant Values but should be equal at all VM and probably 0 is best.
ii.	While not stop condition
iii.	Generate m values for a tour
iv.	Perform sorting on values by their length such that
v.	$l_1 \leq l_2 \leq l_m$
vi.	Value at Parse Table of the VM is update is weighted according to the rank R of the VMs
vii.	The n best VMs is chosen based on the rank R
viii.	If W is the weight of the trail level involvement of the best tour length than it should not be exceeded by any other VMs weight.

4. The Load Balancer parses the table from the top and finds least loaded VM whose state is available.
5. If there are more than one
 - a. The Load Balancer checks memory utilization of each VM and finds highest priority virtual machine.
 - b. The Load Balancer returns the VMid of highest priority VM to Data Center Controller.
 - c. DCC notifies the Load Balancer of new allocation.
 - d. Load Balancer updates the information of the table accordingly.
6. Return VMid to Data Center Controller.
7. DCC sends the request to the Chosen VM.
8. After finishing the request by the VM, the DCC receives the response cloudlet and notifies the Load Balancer for the de-allocation of the VM.

9. Whenever the request of the VM is processed the allocation table of VM is decremented by one by the Load Balancer.
10. If there are more requests, go to step 3 else continue from 2.

4. RESULT ANALYSIS

The table shown below is the analysis and comparison of Make Span time between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of requested tasks and hence Make Span time is computed for the existing and the proposed methodology.

Table 1: Analysis of MakeSpan Time

No. of Requested Tasks	MakeSpan (S)	
	Existing Work	Proposed Work
5	280	234
10	330	291
15	380	350
20	390	327
25	740	674
30	985	815

The table shown below is the analysis and comparison of Energy Consumption between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of Hosts and hence Energy Consumption is computed for the existing and the proposed methodology.

Table 2: Analysis of Energy Consumption

No. of Hosts	No. of Vm's	Energy Consumption (Kwh)	
		Existing Work	Proposed Work
10	5	1.52	1.21
10	10	1.55	1.33
10	15	1.9	1.6
10	20	2.32	2.15
10	25	2.55	2.29
10	30	2.76	2.3

The table shown below is the analysis and comparison of Throughput between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of Time and hence Throughput is computed for the existing and the proposed methodology.

Table 3: Analysis of Throughput

Time (S)	Throughput	
	Existing Work	Proposed Work
200	1.3	1.6
400	2.1	2.5
600	2.3	2.7
800	2.4	2.9
1000	2.5	3
1200	2.6	3.1
1400	2.8	3.3

The table shown below is the analysis and comparison of Standard Deviation between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of Time and hence Standard Deviation is computed for the existing and the proposed methodology.

Table 4: Analysis of Standard Deviation

Time (S)	Standard Deviation	
	Existing Work	Proposed Work
100	0.23	0.21
200	0.26	0.22
300	0.24	0.19
400	0.18	0.14
500	0.15	0.1
600	0.16	0.12
700	0.05	0.02
800	0.05	0.03
900	0.05	0.02
1000	0.04	0.01

The Figure shown below is the analysis and comparison of MakeSpan time between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of requested tasks and hence MakeSpan time is computed for the existing and the proposed methodology.

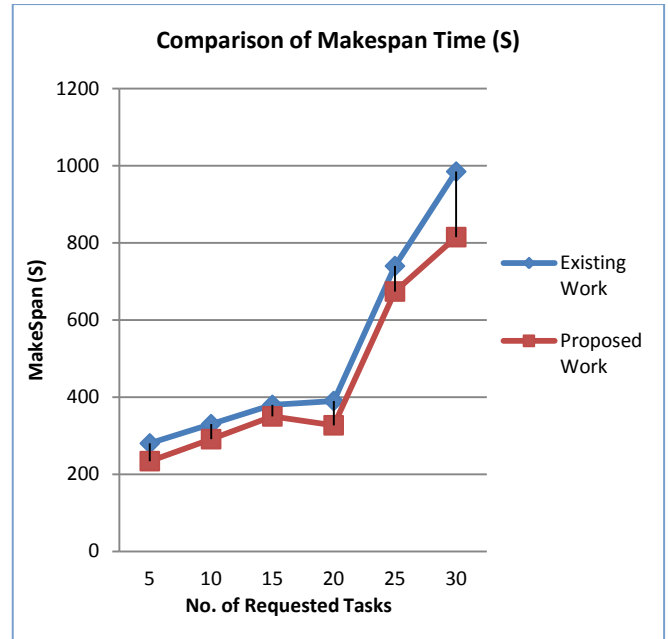


Figure 4: Comparison of Make Span Time

The Figure shown below is the analysis and comparison of Energy Consumption between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of Hosts and hence Energy Consumption is computed for the existing and the proposed methodology.

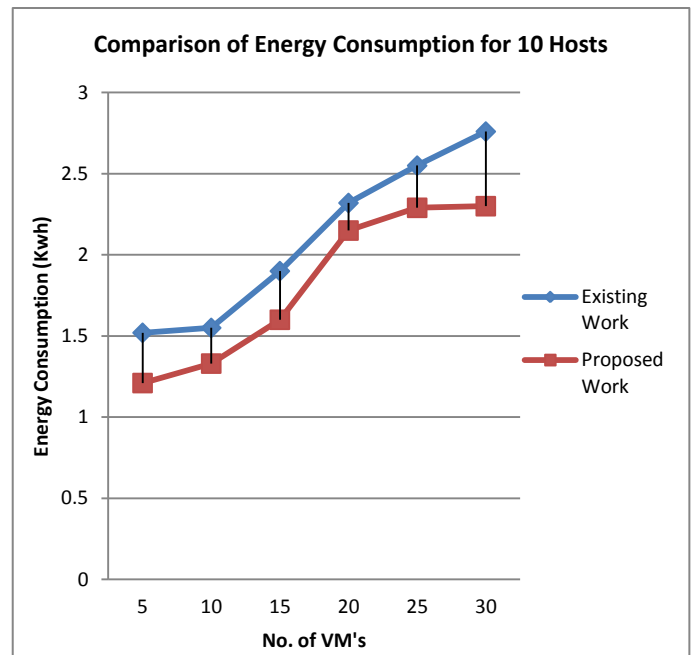


Figure 5: Comparison of Energy Consumption

The Figure shown below is the analysis and comparison of Throughput between the existing Load balancing technique and the proposed methodology. The Analysis is done on the basis of number of Time and hence Throughput is computed for the existing and the proposed methodology.

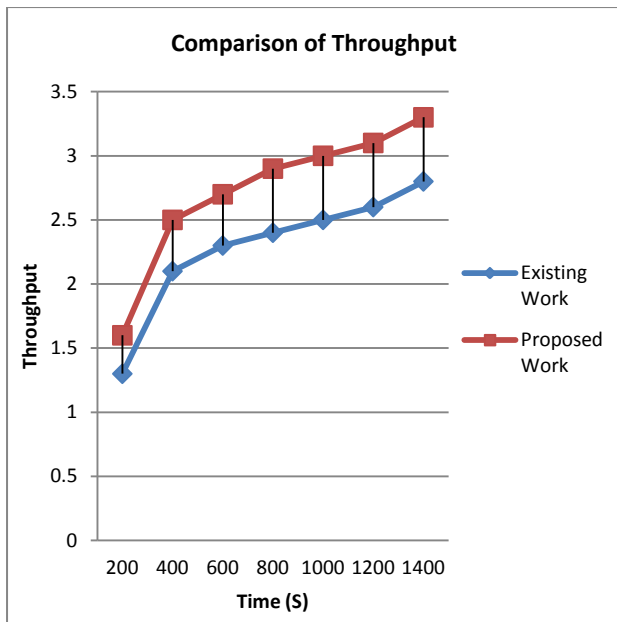


Figure 6: Comparison of Throughput

5. CONCLUSION

The Proposed Methodology implemented here for Load Balancing over Public Clouds using Modified Active Monitoring based Ant Clustering provides efficient Throughput and MakeSpan Time as well as Standard Deviation in comparison with the Existing Naïve Bayes Clustering Load Balancing.

The Proposed Methodology implemented here for Load Balancing using Ant Colony Based technique is efficient in terms of Failure Tasks and Throughput as well as Make span time. The Proposed Methodology when implemented also provides less Energy Consumption and Efficient Load Balancing as compared to the existing methodology implemented for Load balancing over cloud Environments.

6. REFERENCES

- [1] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud based vehicular networks with efficient resource management," *IEEE Netw. Mag.*, vol. 27, no. 5, pp. 48–55, Sep./Oct. 2013.
- [2] Rahman, Mazedur, Samira Iqbal, and Jerry GAO. "Load Balancer as a Service in Cloud Computing." In *Service Oriented System Engineering (SOSE)*, 2014 IEEE 8th International Symposium on, pp. 204-211. IEEE, 2014.
- [3] Rajkumar Buyya. *Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility*. In *CCGRID*, page 1, 2009
- [4] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. *Xen and the art of virtualization*. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003
- [5] VMware. *VMware virtualization software*, 2012.
- [6] Younggyun Koh, Rob Knauerhase, Paul Brett, Mic Bowman, Zhihua Wen, and Calton Pu. An analysis of performance interference effects in virtual environments. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2007.
- [7] Xing Pu, Ling Liu, Yiduo Mei, Sankaran Sivathanu, Younggyun Koh, and Calton Pu. Understanding performance interference of i/o workload in virtualized cloud environments. Pages 51–58, 2010.
- [8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [9] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Key challenges in cloud computing: enabling the future internet of services," *IEEE Int. Comput.*, vol. 17, no. 4, pp. 18–25, Jul./Aug. 2013.
- [10] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, and X. Zhu, "VMware distributed resource management: Design, implementation and lessons learned," *VMware Tech. J.*, vol. 1, no. 1, pp. 45–64, Mar. 2012.
- [11] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Proc. 9th IEEE Int. Conf. Grid Cooperative Comput.*, 2010, pp. 87–92
- [12] Jia Zhao, Kun Yang, "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment", *IEEE Transaction on Parallel and Distributed Systems*, Volume 27, Issue 2, February 2016.
- [13] S. Girdzijauskas, G. Chockler, Y. Vigfusson, Y. Tock, and R. Melamed, "Magnet: practical subscription clustering for Internet-scale publish/subscribe," in *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, ACM, 2010.
- [14] J. Alveirinho, J. Paiva, J. Leitao, and L. Rodrigues, "Flexible and Efficient Resource Location in Large-Scale Systems," *The 4th ACM SIGOPS/SIGACT Workshop on Large Scale Distributed Systems and Middleware*, 2010.
- [15] Q. Wei, G. Xu, and Y. Li, "Research on cluster and load balance based on Linux virtual server," in *Proc. Inf. Comput. Appl.*, 2011, pp. 169–176
- [16] S. M. Lau, Q. Lu, and K. S. Leung, "Adaptive load distribution algorithms for heterogeneous distributed systems with multiple task classes", *J. Parallel Distributed Computing*. vol. 66, no. 2, pp. 163–180, 2006.
- [17] J. Sonnek, J. Greensky, R. Reutiman, and A. Chandra, "Starling: Minimizing communication overhead in virtualized computing platforms using decentralized affinity-aware migration," in *Proc. 39th IEEE Int. Conf. Parallel Process.*, Sep. 2010, pp. 228–237.