Novel Statistical Load Balancing Algorithm for Cloud Computing

Shekhar kumar Swarnkar
(Scholar)
Department of Computer Science & Engineering
Kautilya Institute of Technology & Engineering,Jaipur,Rajasthan,India
Shekharsn19@gmail.com

Chetan Kumar
(Associate Professor)
Department of Computer Science & Engineering
Kautilya Institute of Technology & Engineering,Jaipur,Rajasthan,India
chetanmnit@yahoo.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract: Load balancing in a cloud computing environment has an important impact on performance. Good load balancing makes cloud computing more efficient and improves user satisfaction. Because enterprises are located near users and have the least distribution, load balancing in cloud computing becomes more and more important. The load balancing of cloud computing provides a good and efficient strategy for multiple queries residing in a centralized cloud computing environment. Complete balance must directly consider two tasks; one will be resource supply and resource allocation, as well as the task plan in the entire distributed system. The round-robin algorithm can be the simplest algorithm shown so far, and can help distribute the population between nodes. Therefore, it is usually the first choice when implementing a simple scheduler. One of the reasons for this simplicity is that the only information required is the node list. This work identified the drawbacks of implementing existing load balancing algorithms in a cloud computing environment, and designed a solution that uses priority as a load balancing parameter in a given environment

Key Words: Cloud Computing, Load Balancing, iDR Simulation tool

I. Introduction

The term cloud computing [1] [2] [3] has changed the classic computing environment of the IT industry. It has powerful features such as virtualization and on-demand resource allocation (dynamic), so it is the most emerging and popular technology in the IT and research fields. Now, due to the increased use of the Internet, related resources are rapidly increasing within a few days, which generate a high workload. A simple cloud network is shown in Figure

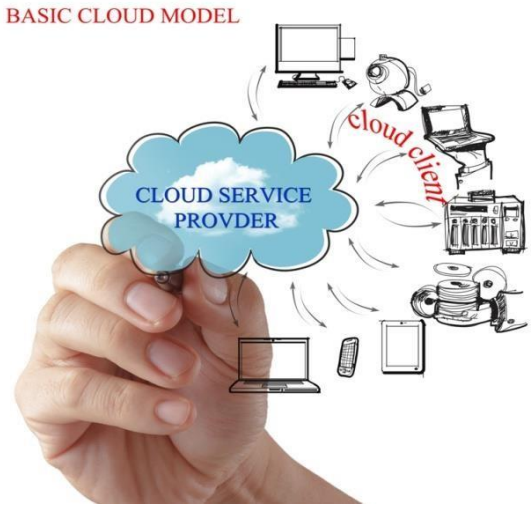

Fig 1 - Basic cloud model [4]

In order to use QOS [5] to provide reliable services to clients, in a cloud environment, a load balancing mechanism must be used to prevent system overload and crashes, and an automatic expansion mechanism must be provided based on applications and incoming user traffic. The load balancing mechanism provides load distribution among one or more nodes. In order to achieve an efficient service model, you can also enable automatic scaling through the load balancer to handle excess load. Auto-scaling can dynamically expand and shrink the platform according to the incoming traffic from the client, thereby saving money and physical resources. Delay-based routing is a new concept in cloud computing. It provides DNS delay-based load balancing for global clients by mapping the Domain Name System (DNS) [10] in different hosting areas.

Cloud computing can be used as three mode and 4 deployment model, three modes are

- SasS
- IasS
- PasS

Four deployment models are

- Public cloud
- Private cloud
- Community cloud
- Hybrid cloud

II. Characteristics of Cloud Computing

- **Elasticity:** it is the core feature of cloud system and confines the potency of the infrastructure to handle the changing dynamically. In simple elasticity can be defined as the how fast a system will respond to the change by providing and removing resource in dynamic manner according to the demands this ensure the current available resources are enough to handle the current demands. Main aim of the elasticity is to match the current demands of services to the resources such as what amount of resource the service need to perform the operation and handle the no of user . let's compare this aspect with traditional computing in which when the request from users increased from the capacity of the server a new physical server need to install to handle the requests but establishing a new server will require time and money .this effect the efficiency of system but in cloud as the no of user request exceed from the capacity of the system it will automatically initiate a new virtual server to handle the extra request it will save both money and time and thus improve the efficiency of the system. Elasticity also handles the issue of over provisioning and under provisioning by monitoring the allocated VMs (virtual machine) and the incoming request. Elastic property allocates and reallocates virtual machines according to the request of application.

- **Reliability** is very much essential and required in all cloud environments to provide QOS. No of user is increasing day by day on cloud and the reliability is the main aspect to ensure steady operation of cloud system without distraction. Fault tolerance technique is introduced by service provider to maintain reliability For example no data loss, session reload, no shut down during the operations, and this can be achieved in the cloud two approaches is used software and hardware. In hardware the redundant & distributive resource utilization in software the duplicity in file system is used.
- **Quality of service** it is the factor by which some required specification of service is decided and must be fulfilled to the outsourced services. it is very essential to met these specific requirement to the service .for example performance QOS is response time must be fast and the throughput of the system must be good same as QOs in result is defined as data freshness, data should be correct. Reliability is the another factor of the QOS
- **Agility and adaptability** this essential feature is relate to dynamic capabilities. It provides the on-time decision to react on the request of changing amount of resources.
- **Availability** for cloud service providers it is very important to make sure that services is available to client at any time. This is core aspect of cloud and abilities to handle services and data failure by introducing the redundancy and as the load and traffic is increasing availability can be achieved by introducing efficient load balancing and scaling technique.

III. Literature Review

Borja Sotomayor, Ruben S. Montero, Ignacio M. Llorente, Ian Foster, etc. [7] In this paper, they define a bee feeding mechanism that uses random stealing techniques to improve load balance. In order to find the state of the virtual machine, calculate the deviation of the virtual machine load and check whether it is limited within the threshold condition set. Using the random stealing method, when the virtual machine is in an idle state, tasks will be stolen from a random virtual machine. Therefore, it saves the idle time of processing elements in the virtual machine. Use cloudSim for performance evaluation. The simulation results show that the improved random bee stealing and foraging technology reduces the execution time of the algorithm, balances the system load, and reduces the idle time of the virtual machine

Aarti Singh, Dimple Juneja, Manisha Malhotra, etc. [8] in order to provide valuable information and influence the decision-making process of the load balancer in order to maintain the best load balance in a hosted (or cloud) environment. The information from the computer system or the network part is insufficient from the external load balancer.

Sasmita Parida, Suvendu Chandan Nayak and others. [9] The author of this article discussed the deployment model of cloud computing. In cloud computing, fault tolerance is a major issue and one of the most important indicators, because resource failures will affect job execution, throughput, response time and performance. System and network. The fault tolerance of load balancing is one of the main challenges of cloud computing. The workload must be evenly distributed among all nodes, faults are detected and eliminated from the network, and then the workload is shared among all nodes to improve the cloud performance of the network.

Suguna R, Divya Mohandass, Ranjani R, etc. [10] In this article, the author defines a two-level centralized scheduling model, where the upper level is the global centralized scheduler (GCS), and the next level is the local centralized scheduler (LCS), which overcomes the high-level problems of distributed algorithms. . The communication cost and single point of failure of the centralized algorithm. Energy-saving load balancing technology can be used to balance the workload on all nodes in the cloud and maximize the use of resources to improve cloud computing performance, thereby reducing energy consumption and carbon emissions to a certain extent, which will help achieve green environmental protection.

IV. Load balancing

Load balancing is a technology that redistributes the total load among the nodes of the cloud aggregation system to improve response time and resource utilization.

The load balancer can be cooperative or non-cooperative. In cooperative load balancing, all nodes of the system must work together to optimize response time. In the absence of collaboration, each node can freely process a single task to improve the response time of the local task.

a. Various existing load balancing algorithms

Load balancing is the process of evenly distributing load among nodes in the cloud to achieve higher user satisfaction, effectively utilize resources and improve job response time [2]. Two types of load balancing algorithms are: static load balancing algorithms and Dynamic load balancing algorithms. With prior knowledge of system tasks and resources, static load balancing algorithms can work in this situation. These designs are simple and require little execution time [5]. Examples include round-robin, weighted round-robin, least connection scheduling algorithm, etc. [6]. In the dynamic load balancing algorithm, there is no need to know the tasks and available resources in advance. They are flexible, reliable and able to handle a large number of user requests. These algorithms depend on the current state of the system and are most suitable for changing environments [7]. Examples include bee foraging behavior, biased random sampling, active clustering, etc. [8].

Now, let's review the existing load balancing algorithms

Throttling load balancing algorithm

This is a dynamic method. In this case, the user request will be submitted to the data center controller (DCC). The restricted VM Load Balancer is responsible for keeping the list of virtual machines and their status (available or busy). First, set the status of all virtual machines to available. When DCC receives a user request, it will ask VM load balancer for information about the virtual machine. VM load balancer checks the index table from the start, and sees which virtual machine can handle the specific load or scans the index table completely. If a suitable virtual machine is found, only the VM load balancer returns the ID of the specific virtual machine to the DCC. Then, DCC assigns the request to the specific virtual machine identified by the ID. After the VM load Balancer updates the allocation table, if no suitable virtual machine is found, the VM load Balancer returns a value of -1 to DCC, and DCC puts the request into the queue. After completing the processing of the allocation request, Xiaoyun is sent to DCC, and DCC returns to send a notification to cancel the allocation.

ESCE load balancing algorithm

ESCE stands for Equally Splitting Current Implementation. It is also called the active VM load balancing algorithm. As the name suggests, it evenly distributes the workload on each VM in the data center [8]. ESCE VM Load Balancer (VMLB) maintains a list of virtual machines and the number of requests that have been allocated to that particular virtual machine. When DCC receives a new client request, it will ask ESCE VM Load Balancer for information about the next virtual machine allocation. ESCE VM Load Balancer scans the index table from startup and searches for the appropriate virtual machine with the least load. If there are many virtual machines with the least load, the first virtual machine is selected and its ID is sent to the DCC. The DCC assigns the client request to the virtual machine identified by the specific ID. After that, VM load Balancer updates the table by increasing the number of allocations for that particular virtual machine. After completing the processing of the allocation request, Xiaoyun will be sent to DCC, and DCC will send back the cancellation notice [10]. At the same time, VMLB checks for overloaded VMs. If any virtual machine is found to be overloaded, VMLB will move some of the load to idle or under-loaded virtual machines to reduce the load of some overloaded VMs. The main disadvantage is high computational overhead

FCFS load balancing algorithm

In terms of formal or artificial fairness, FCFS scheduling is fair, but in terms of long jobs executing short job waits instead of important jobs executing important job waits, FCFS scheduling is unfair. FCFS is more predictable than most other schemes because it provides time. When arranging interactive users, the FCFS scheme is useless because it cannot guarantee a good response time. FCFS scheduling code is easy to write and understand. One of the main disadvantages of this scheme is that the average time is usually very long. The First-Come-First-Served algorithm is rarely used as the main solution in modern operations

Minimum connection

When distributing requests, neither Round Robin nor weighted Round Robin will consider the current server load. The least connection method does not consider the current server load. The current request will go to the server with the least activity at the current time.

Token Routing

The main goal of the algorithm [11] is to minimize the process cost by transmitting your own token to the system. But inside a good scalable cloud technology, the agent cannot provide you with enough detailed information currently.

Random

Random algorithms can be static types with natural properties. In a specific algorithm [12], this process can be handled by a private node n with probability p. For each independent processor associated with the remote processor allocation, your current system allocation order will be retained.

LBHM Hybrid Load Balancing Algorithms

LBHM have the characteristics of Throttled Load balancing Algorithm and Equally Spread Current Execution Algorithm with an additional feature of Threshold Limit associated with each VM. The value of Threshold for each VM is different and based on the size and capacity of each VM [13]. VM having high Threshold Value will execute task more efficiently then compare to other VM. It is a non- preemptive load balancing algorithm. Here balancer maintains a allocation table with five fields: VM id, VM status, number of active task on that VM, threshold limit of that VM and difference count of VM (difference of threshold limit and number of active task on that VM).LBHM works in such a way that the numbers of active tasks on each virtual machine should not exceed the threshold limit of that VM at any time instant. In the beginning all VM status is available. Active task is 0. Threshold limit is size and capacity of VM. First free VM gets the task. If no VM is found free then task is assigned to VM with maximum value of difference count. In both cases allocation table values are updated. Allocation table values are also updated when any VM gets free. If there is no task in waiting queue and any VM gets free then reshuffling of task is done between free VM and VM with minimum value of difference count.

V. Proposed Algorithm Model

Based on our study we observed all the algorithms till now have very less effort on statistical analysis and they work on VM schedulers in different logics, with these efforts the task scheduling cannot happen in tandem with task execution and thus load balancing exercise increase lot of time for task allocation.

We hereby propose a new mechanism for load balancing using poison distribution and compare the same with existing Round Robin Algorithm on following Parameters:

1. Virtual Machine and Data Center Vs Jobs Allocated
2. Virtual Machine and Data Center Vs CPU Utilization
3. Virtual Machine and Data Center Vs RAM Utilization
4. Virtual Machine and Data Center Vs Network Utilization
5. Time Stamp Vs VM Utilization
6. Time Stamp Vs Data Center Utilization

Poison Distribution

In statistics, the Poisson distribution is a probability distribution that can be used to show how many times an event may occur in a specified time period. In other words, it is a count distribution. The Poisson distribution is often used to understand independent events that occur at a constant rate within a given time interval. It is named after the French mathematician Siméon Denis Poisson.

The Poisson distribution is a discrete function, which means that the variable can only take certain values in a (possibly infinite) list. In other words, the variable cannot take all values in any continuous range. For Poisson distribution (discrete distribution), variables can only take values 0, 1, 2, 3, etc., without decimals.
The Equation for Poison Distribution is:

$$f(x) = \frac{\lambda^x}{x!}e^{-\lambda}$$

Where:

- e is Euler's number (e = 2.71828...)
- x is the number of occurrences
- x! is the factorial of x
- λ is equal to the expected value of x when that is also equal to its variance

Simulation Environment

To design & analyze the proposed model we used iDR simulation tool as discussed in chapter 4. Where in we created

1. Data Center 3
2. Each Data center is having 10 Virtual Machine
3. Each Virtual Machine has 5 services
4. We created 20 Jobs to be executed amongst the 3 Data centres
5. These 20 jobs are being raised by 10 Users

We used same simulation environment for both algorithm, existing Round Robin Load balancing Algorithm and new proposed max Resource –Min Time Load Balancing Algorithm

Proposed Algorithm

Step 1**:** Create Simulation Environment
Step 2: Initiate Job ID
Step 3: Initiate Service ID
Step 4: Define CPU Utilization per Job
Step 5: Define RAM utilization per Job
Step 6: Define Network Usage per Job
Step 7: Loop: Time stamp 1 to 500
For 1st DC
Step 7.a Check CPU availability, if yes mark available
Step 7.b Check RAM availability, if yes mark available
Step 7.c Check Network availability, if yes mark available
If No availability inform to DC & search next DC
Step 8: Load balancing for Resource Allocation
Step 8.a Using Poison Distribution we distribute Jobs to nearest available DC and its VM
Step 8.b. Continue till all task are completed
Step 9: After Completing of all task, release all resources
Step 10: End

VI. Results

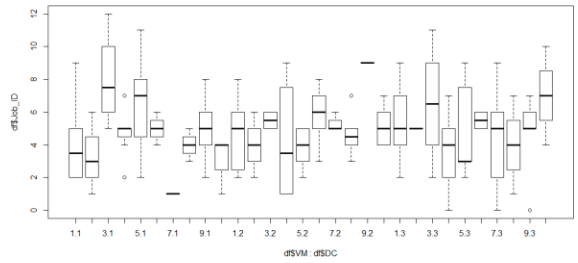

Fig 2 Median Box Plot for VM.DC Vs Job Allocation

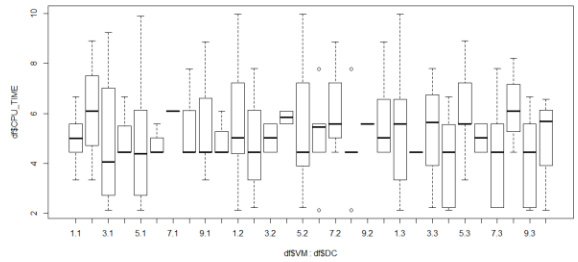

Fig 3 Median Box Plot for VM.DC Vs CPU Time allocation

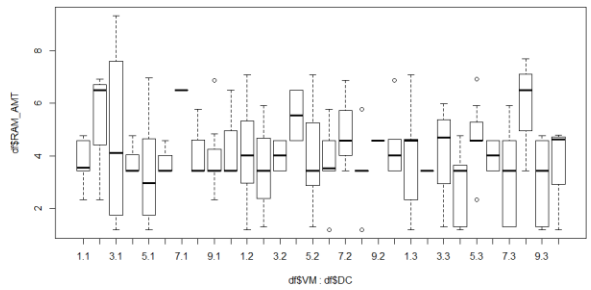

Fig 4 Median Box Plot for VM.DC Vs RAM Utilization

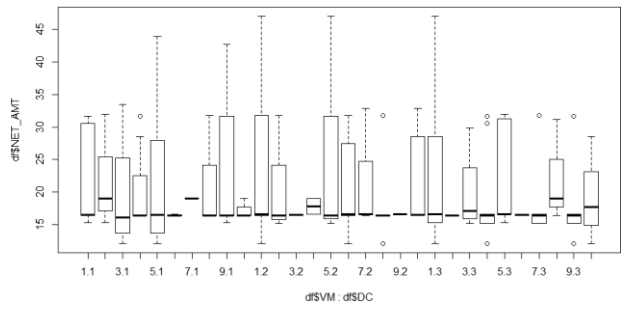

Fig 5 Median Box Plot for VM.DC Vs Network Utilization

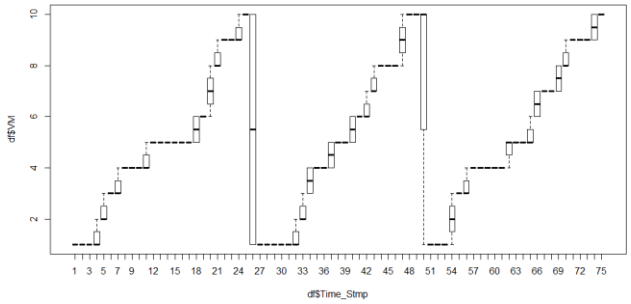

Fig 6 Median Box Plot for Time allocation for each VM

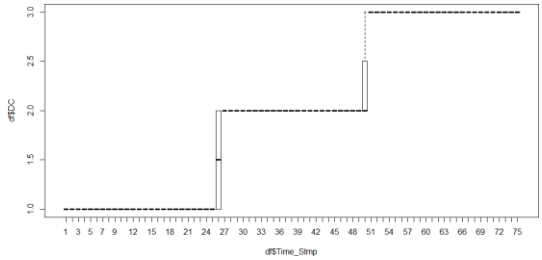

Fig 7 Median Box Plot for Time allocation for each DC

Summary

As clearly visible in above figures the time required for executing all 20 tasks over 3 DC and 10 VM with similar configuration is only 90 Sec for proposed algorithm whereas the time required by Round robin Algorithm is 469 sec. almost 20% only required by new algorithm only. Even the median box bar in most of the graph in proposed algorithm is almost static, except in case of Job allocation, which is smoother in Round robin algorithm, but in case of CPU utilization, RAM utilization and Network utilization the proposed algorithm perform much better.

VII. Conclusion

The current work aims to provide detailed knowledge about the importance of load balancing, automatic scaling, resource monitoring, and latency-based load balancing in a cloud environment. In this study, if all services are used, we will use IDR simulation tools to develop effective load models, as an individual. It is very effective and efficient load balancing as it invest more time on resource utilization and less time on resource allocation. As clearly visible in results the proposed algorithm only requires only 20% of time required by Round Robin Algorithm. Even the median box bar in most of the graph in proposed algorithm is almost static, except in case of Job allocation, which is smoother in Round robin algorithm, but in case of CPU utilization, RAM utilization and Network utilization the proposed algorithm perform much better. Even most of the time RR invests more time in toggling amongst all DC & VM without optimal Utilization

REFRENCES

[1] Eddy Caron: Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds

[2] Miss.Rudra Koteswaramma : Client-Side Load Balancing and Resource Monitoring in Cloud , ISSN: 2248- 9622

[3] N. Ajith Singh, M. Hemalatha, “An approach on semi distributed load balancing algorithm for cloud computing systems” International Journal of Computer Applications Vol-56 No.12 2012.

[4]Nitika, Shaveta, Gaurav Raj, International Journal of advanced research in computer engineering and technology Vol-1 issue-3 May-2012

[5] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, IPCSIT Vol-14, IACSIT Press Singapore 2011

[6] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed

[7]Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems,
IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.
[8]http://www.amazon.com/gp/browse.html?node=201590011
[9] Amazon Elastic Compute Cloud http://aws.amazon.com/ec2/.
[10] Amazon web services cloud watch Web Site, November 2013.
[11] Aws elastic load balancing Web Site, November 2013
[12] R. Ranjan, A. Harwood, and R. Buyya. Peer-to-Peer Based Resource Discovery in Global Grids: A Tutorial. IEEE Communications Surveys and Tutorials, Volume 10, Issue 2, Pages 6-33, IEEE Communication Society, 2008.

BIBLIOGRAPHY

[13] Dongliang Zhang, Changjun Jiang,Shu Li, "A fast adaptive load balancing method for parallel particle-based simulations", Simulation Modelling Practice and Theory 17 (2009) 1032–1042.
[14] Dhinesh Babu L.D, P. VenkataKrishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing 13 (2013) 2292–2303.
[15] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, "A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers", J. Parallel Distribution Computing. 72 (2012) 1254–1268.
[16] Yunhua Deng, Rynson W.H. Lau, "Heat diffusion based dynamic load balancing for distributed virtual environments", in: Proceedings of the17th ACM Symposium on Virtual Reality Software and Technology, ACM, 2010, pp. 203–210.
[17] Markus Esch, Eric Tobias, "Decentralized scale-free network construction and load balancing in Massive Multiuser Virtual Environments", in: Collaborative Computing: Networking, Applications and Worksharing, Collaborate Com, 2010, 6th International Conference on, IEEE, 2010, pp. 1–10.
[18] B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, I. Stoica, "Load balancing in dynamic structured P2P systems", in: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4, IEEE, 2004, pp. 2253–2262.
[19] P. Sobeslavsky, "Elasticity in cloud computing," Master's thesis, Joseph Fourier University, ENSIMAG, Grenoble, France, 2011.
[20] D. Agrawal, A. El Abbadi, S. Das, and A. J. Elmore, "Database scalability, elasticity, and autonomy in the cloud," in *Proceedings of the 16th Intl. conference on Database systems for advanced applications - Volume Part I*, ser. DASFAA'11. Springer-Verlag, 2011, pp. 2–15.
[21] M. Kupperberg, N. Herbst, J. Kistowski, and R. Reussner, "Defining and quantifying elasticity of resources in cloud computing and scalable platforms," Tech. Rep., 2011. [Online]. Available: http: //digbib.ubka.uni-karlsruhe.de/volltexte/1000023476
[22] L. Badger, R. Patt-Corner, and J. Voas, "Draft cloud computing synopsis and recommendations recommendations of the national institute of standards and technology," *Nist Special Publication*, vol. 146. [Online]. Available: http://csrc.nist.gov/publications/drafts/ 800-146/Draft-NIST-SP800-146.pdf *52th Photogrammetric Week*. W. Verlag, September 2009, pp. 3–12.
[23] Kumar, A., Sharma, G., Jain, P. et al. Virtual environments testing in cloud service environment: a framework to optimize the performance of virtual applications. Int J Syst Assur Eng Manag (2021). https://doi.org/10.1007/s13198-021-01105-y
[24] Rajput, RS and Goyal, Dinesh and Pant, Anjali, The Survival Analysis of Three-Tier Architecture Based Cloud Computing System (2018). International Journal of Advanced Studies of Scientific Research, Vol. 3, No. 11, 2018, Available at SSRN: https://ssrn.com/abstract=3320440
[25]Sharma, Arpita and Kumar Gupta, Amit and Goyal, Dinesh, An Optimized Task Scheduling in Cloud Computing Using Priority (April 20, 2018). Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), 2018 held at Malaviya National Institute of Technology, Jaipur (India) on March 26-27, 2018, http://dx.doi.org/10.2139/ssrn.3166077
[26] Rajput, R S and Goyal, Dinesh and Singh, S. B., Study of Performance Evolution of Three Tier Architecture Based Cloud Computing System (April 21, 2018). Proceedings of 3rd International Conference on Internet of Things and Connected

Technologies (ICIoTCT), 2018 held at Malaviya National Institute of Technology, Jaipur (India) on March 26-27, 2018, http://dx.doi.org/10.2139/ssrn.3166719

[27] Rajput R.S., Goyal D., Pant A. (2019) The Survival Analysis of Big Data Application Over Auto-scaling Cloud Environment. In: Somani A., Ramakrishna S., Chaudhary A., Choudhary C., Agarwal B. (eds) Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics. ICETCE 2019. Communications in Computer and Information Science, vol 985. Springer, Singapore. https://doi.org/10.1007/978-981-13-8300-7_13

[28] Rajput, R. K., & Goyal, D. (2020). Auto-Scaling in the Cloud Environment. In S. Gochhait, D. Shou, & S. Fazalbhoy (Ed.), Cloud Computing Applications and Techniques for E-Commerce (pp. 84-98). IGI Global. http://doi:10.4018/978-1-7998-1294-4.ch005

[29] Rajput,R.S., Goyal, Dinesh ; Hussain, Rashid et. Al.; Provisioning of Virtual Machines in the Context of an Auto-Scaling Cloud Computing Environment, Journal of Computational and Theoretical Nanoscience, Volume 17, Number 6, June 2020, pp. 2430-2434(5)
https://doi.org/10.1166/jctn.2020.8912