



ISSN 2278 – 0211 (Online)

A Throttled Load Balancing for Cloud Infrastructure

Kashmira Jagtap

Research Scholar, Department of Information Technology, BVUCOE, Pune, Maharashtra, India

P. A. Jadhav

Professor, Department of Information Technology, BVUCOE, Pune, Maharashtra, India

Dr. S. D. Joshi

Professor, Department of Computer Engineering, BVUCOE, Pune, Maharashtra, India

Abstract:

Cloud computing is attraction-gaining technology now days because of its great functioning. Day by day, there is tremendous increase in utilization of cloud functions by various users' raises issues of workload management. Work Load can be a many types like user's requests, processors work, and size of memory. Therefore, to manage such load to maintain stable system is necessary. Due to its feature cloud achieve scalability but at the same time not able to maintain stability because of coming multiple requests in few seconds. Due to such scenarios nowadays, workload management for cloud is popular among researchers. In cloud request is not arrived in same pattern also, resources are not available in one place so to manage workload properly is quite problematic in cloud. So good workload management is necessary to maintain stability and improve user's satisfaction. In this paper, we focused on workload management concept. Here we are implementing a work load management architecture for cloud based on throttled method to solve issue of load management in cloud.

Keywords: Workload management, Cloud Computing, Throttled workload management, Job scheduling, clustering.

1. Introduction

Cloud computing is now a popular because of its great functioning and having best facilities for utilization in online environment. Actually in this technology data saved from anywhere of world, user does not aware of where he actually stores his data and who can use it, how it will be managed. Again, requests are coming from so many places of world in few seconds it will put on cloud resources. However, this technology having number of advantages and due to we can store unlimited data users are using this technology without taking care of workload [1]. Therefore, this will degrade the working of cloud system. All the resources of cloud situated far from its database and for using those resources internet is must so congestion is one problem.

Cloud is wide by nature but cloud request is not arrived in same pattern also, resources are not available in one place so to manage workload properly is quite problematic in cloud. So good workload management is necessary to maintain stability and improve user's satisfaction [1].

Work Load management depends on system types whether is dynamic or static. In static type do not utilize any status statistics form so difficult to manage load in this scheme. Other side in dynamic case it is efficient manage load because it use system status but result in more cost. For our approach, we are using dynamic scheme

Architecture requires balancer and controller to analyze and collect the data. In dynamic approach, it does not disturb other busy nodes. Then status statistics gives an option to assign exact load management strategy. In this project, work our focus on, to give better workload management strategy in context of cloud infrastructure. In this section, we are going to focus on workload management and will take overview of static and dynamic algorithms. Let see workload management.

1.1. Work Workload Management

Workload management is one of the main problems related to cloud computing. The load can be in many types like processor or memory and devices or resources in cloud. To increase working of resources there is need to distribute this load in between available all servers. These things can boost performance where nodes are loaded. Therefore, workload management is a mechanism of distributing work among all available resources according its capacity. Need of a workload management [2] is for increase in

performance of system, for user satisfaction, fault resistance system, to achieve stable system. Generally there are two schemes of workload management as follow...

1.1.1. Dynamic Algorithm

In this scheme light load server selected to adjust load from a framework. Status table used to decide light loaded server

1.1.2. Static Algorithm

In this scheme, require information of all resources in system according to that load distinguished in available servers. So here, status statistics not required. If system always requires low work then this is proper mechanism.

In this project, our aim is to give proper mechanism to balance workload in cloud infrastructure based on throttled mechanism. To achieve our goal we first partition public cloud in cluster so in case of big organizations, it will do balancing easily by based on dynamic approach. Reminder of this paper articulate the design section II gives overview of previous techniques, section III depicts proposed architecture. Section IV provides result analysis. Section V concludes our discussion.

2. Related Work

Workload management is important aspect regarding cloud infrastructure for effectiveness and scalability, up to today there are many researchers have been proposed mechanism to achieve workload management. In 2010 author Randles et al. [4] presented comparison of some existing algorithms based on cost and its performance. Author mentioned that the throttled algorithm ESCE algorithms are works properly compare to Round Robin algorithm in such scenarios. In 2011 Chaczko et al. [5] presented paper on availability and work load management in cloud they mentioned that how work load management plays is important in cloud to maintain stability and to improve performance. The study proposed in 2012 [6] makes conclusion that existing mechanisms just have give emphasis on lessen overhead related to work, resource utilization time and increasing work measurement etc in this paper author observed previous techniques on various parameters. In same year, Adler published a white paper on work load management in cloud computing [7], where he overviewed some techniques and tools for workload management. However, workload management still a challenge to resolve and require efficient architectures. In their study Nishant et al. [8] proposed indentifying method based on ant mechanism in nodes work load management.

In 2013, research [9] proposed a load-balancing model for a public cloud. In this model with distributed resources there are number of nodes in different geographical areas. Therefore, this public cloud divided in to partitions to simplify the workload management in large and complex cloud infrastructure. However, in this method nodes are far from each other so difficult to maintain it. In same year there had been proposed a review [10] on various workload management techniques for cloud computing. Author also focused on optimized matrix based on virtualization for workload management.

In 2014, Palivela et al. [11] proposed an improved workload management model for cloud infrastructure by cloud separation idea with a transfer mechanism to choose distinct ideas for different environments. In same year Rajkumar et al. in their research work [12] mentioned hybrid approach for workload management. They combined features of round robin and throttled algorithm to achieve workload management. In 2015 AartiSinghaet. Al. presented work on Autonomous workload management mechanism in Cloud Computing based on agent [13]. This mechanism provides dynamic workload management for heavy loaded cloud environment. Author contributed proactive work calculation of VM in a DC and whenever load of a VM reaches near threshold value, load agent initiates search for a candidate VM from other DCs. Keeping information of candidate VM beforehand, reduces service time. Result obtained through implementation proved that this algorithm works satisfactorily.

3. Proposed Work

In this project, our motive is to maintain performance and stability of cloud environment to achieve this aim we designing a workload management architecture. The basic idea behind our proposed system is we are maintaining job arrival by adjusting it on less loaded nodes. For this design, we are first dividing total nodes of a network in different clusters and status table is been maintained to show capacity of each node. Here when request come controller decides best cluster for it and then balancer decide proper workload management mechanism. Workload management is been achieved through threshold parameter, arrived file is first compare to threshold value then if load is greater than threshold it get divides in to chunks. Load balancer then refresh the status table. The balancer in each cluster collects the status statistics from each available node and then selects the correct way to distribute the jobs. Therefore, we maintain two-status tables one shows information of each node and second show information after workload management for particular node, which is result status table. To achieve this we are using throttled workload management mechanism.

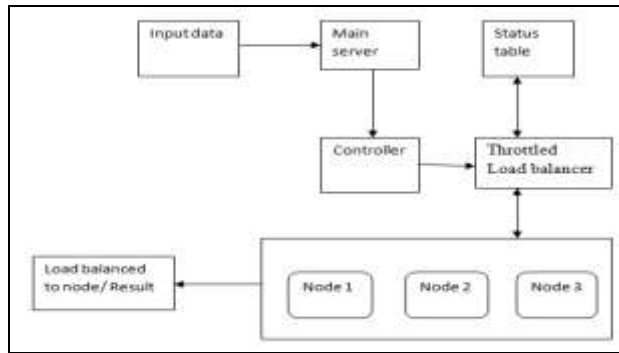


Figure 1: Architecture for workload management in cloud.

3.1. Throttled Algorithm

Throttled scheme is an algorithm for load management using virtual machine concept in this concept first user have to requests balancer to select correct VM which is able to manage load and perform according to user’s request. If number of jobs is there in compare to VMs in data stack then incoming requests get queued.

4. Result

We designed cloud-computing system where various organizations place their data. When any organization uploads their files, system automatically saves it on proper node according to threshold. If file is greater than threshold, it divided in to chunks so workload management achieved. Let see view of a system.

4.1. Registration Page for Organization

New organization registers their information by selection of Add new organization tab.



Figure 2

4.2. Organizations Available in System

View organization tab of a cloud system shows organizations using this system.



Figure 3

4.3. User Login



Figure 4

4.4. Secure Login

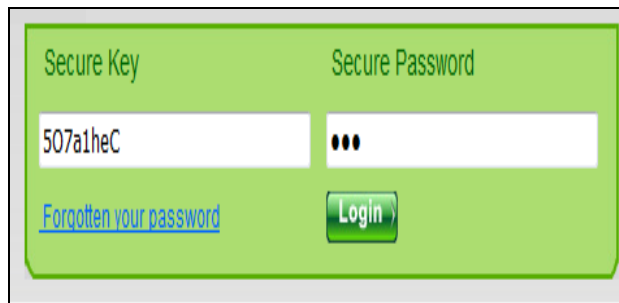


Figure 5

Here user login with secret key.

4.5. User Homepage

Upload, create, Browse tabs are available for organization to handle their data.



Figure 6

4.6. Set Access to Files



Figure 7

Public and private access criteria assigned to users while uploading and browsing of a file.

4.7. Upload Files

If user selects upload option following page will appear according to private or public access.



Figure 8

4.8. Upload Successfully



Figure 9

4.9. Browse Files

Shows list of available files.



Figure 10

4.10. Private Files

User can access private files by using private key.



Figure 11

4.11. Admin Page

All the access to user's public and private files set by administrator so data privacy also maintained.



Figure 12

5. Conclusion and Future Work

This project work is been completed in context of cloud infrastructure where load management is significant issue. Proposed architecture is trying to fulfill requirement of cloud computing in respect of workload management. To complete this work we reviewed some previous techniques with their advantages and disadvantages and designed system to overcome their pitfalls. By setting threshold value load is been managed and if file size exceeds threshold it will divide in chunks. Result analysis shows we can balance load in cloud environment with designed software framework.

In future, there is provision to upgrade this system as real time uploading of files. In addition, now we are selecting files according to its memory size in future we can upload according to processor speed.

6. References

- i. Basic concept and terminology of cloud computing- <http://whatiscloud.com>
- ii. D. Escalante, Andrew J. Korty, "Cloud Services: Policy and Assessment", Educause review July/August 2011.
- iii. Manjunatha H R , Harish H K , " Survey on Workload management in Cloud Computing System", ISRASE ,2015, pp. 1-6.
- iv. M. Randles, D. Lamb, and A. Taleb-Bendiab, A comparative study into distributed workload management algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.
- v. Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, "Availability and workload management in cloud computing", presented at the 2011 International Conference on Computer and Software Modeling, Singapore, 2011.
- vi. Nidhi Jain Kansal* And Inderveer Chana, "Existing Workload management Techniques In Cloud Computing: A Systematic Re-View", Journal of Information Systems and Communication , Volume 3, Issue 1, 2012, pp- 87-91.
- vii. Adler, "Workload management in the cloud: Tools, tips and techniques", <http://www.rightscale.com/info-center/whitepapers/Load-Balancing-in-the-Cloud.pdf>, 2012.
- viii. K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, Workload management of nodes in cloud using ant colony optimization, in Proc. 14th International Conference on Computer Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom, Mar. 2012, pp. 28-30.
- ix. Xu, Gaochao, Junjie Pang, and Xiaodong Fu. "A workload management model based on cloud partitioning for the public cloud." IEEE Tsinghua Science and Technology, Vol. 18, no. 1, pp. 34-39, 2013.
- x. Tushar Desai, Jignesh Prajapati, " A Survey Of Various Workload management Techniques And Challenges In Cloud Computing ", International Journal Of Scientific & Technology Research Volume 2, Issue 11, November 2013, pp. 158-161.
- xi. R. K. Palivela, C. S. Redy, "OAD Balancer Strategy Based On Cloud Computing", International Journal of Research in Computer and Communication Technology, Vol 3, Issue 10, October – 2014
- xii. Rajkumar Somani, JyotsanaOjha' " A Hybrid Approach for VM Workload management in Cloud Using CloudSim ",International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 6, June 2014, pp. 1734-1739.
- xiii. AartiSingha, Dimple Junejab, Manisha Malhotra , "Autonomous Agent Based Workload management Algorithm in Cloud Computing ", Elsevier, Procedia Computer Science 5 (2015), pp. 832 – 841.