A Study of a New Dynamic Load Balancing Approach in Cloud Environment

Sanjay Chakraborty*, Nilotpal Choudhury

Department of Computer Science & Engineering, Institute of Engineering & Management, Kolkata, India

Copyright©2016 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract Distributing workloads across multiple computing resources are one of the major challenges in a cloud computing environment. This paper is being discussed over the basic obstacles of load balancing in cloud environment. The paper looks beyond the problems faced by the cloud system to overcome those through probable improvised techniques. This is a paper over solving the problems exist in the present days by logically analyzing and presenting in an algorithmic format. This approach is mainly focused on an effective job queue making strategy which is suitably allocated the various jobs to CPUs based on their priority or without priority. It also deals with some of the major problems of load balancing in cloud environment like, timeout. Finally, it shows how this approach is fitted in famous AWS and GAE cloud architecture partially. This article will provide the readership an overview of various load balancing problems in cloud environment while also simulating further interest to pursue more advanced research in it.

Keywords Cloud, Load Balancing, Migration, Job Sequence, Timeout, Super Node

delay. CPU located in different places have different task loads. Again every CPU has different speed. So the task completion rate of every CPU changes. To make the system efficient, the system must have prepare a process where the CPUs work in such way that the collective speed of the system is increased. So, one of the important issues in cloud computing is to balance these loads. And it is too much difficult to manage cloud computing without balancing loads. There are several other resources which will be load balanced, like

- Network edges and facilities such as DNS, FTP, and HTTP
- Making Connections through intelligent switches
- Processing through computer system task
- Storage resources right of entry to application instances.

The rest of the paper is organized as follows: Section 1.1 discusses about an overall architectural components and various architectures of cloud system. In the 1.2 and 2 sections, various difficulties of cloud systems including load balancing are discussed and a solution based approach is provided in section 3. Section 4 discusses about the QTS service in cloud. Finally, section 5 highlights some research issues and section 6 gives the conclusion of this paper.

1. Introduction

Cloud is a large distributed computing system which shares resources, software and information on-demand, like public utility to a large number of users. Cloud computing is an evolution of evolution of Virtualization, Utility computing, Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS). Load balancing is the process of reassigning the total loads to the individual nodes of the collective system to make the best response time and also good utilization of the resources. Cloud system is the collection of many heterogeneous systems [5,6]. The systems are actually consists of servers and clients. Clients' request for the resources must be provided as soon as possible. The CPU of the server must process the client's request without too much

1.1. Cloud Architecture

Cloud service models are commonly divided into SaaS, PaaS, IaaS and Data as a Service (DaaS) that exhibited by a given cloud infrastructure [16, 17, 18].

A. Software as a Service (SaaS)

Cloud consumers release their applications in a hosting environment, which can be accessed through networks from various clients (e.g. Web browser, PDA, etc.) by application users. Cloud consumers do not have control over the cloud infrastructure that often employs multi-tenancy system architecture, namely, different cloud consumers' applications are organized in a single logical environment in the SaaS cloud to achieve economies of scale and optimization in

terms of speed, security, availability, disaster recovery and maintenance. Examples of SaaS include Salesforce.com, Google Mail, Google Docs, and so forth.

B. Platform as a Service (PaaS)

PaaS is a development platform supporting the full “Software Lifecycle” which allows cloud consumers to develop cloud services and applications (e.g. SaaS) directly on the PaaS cloud. Hence, the difference between SaaS and PaaS is that SaaS only hosts completed cloud applications whereas PaaS offers a development platform that hosts both completed and in-progress cloud applications. This requires PaaS, in addition to supporting application hosting environment, to possess development infrastructure including programming environment, tools, configuration management, and so forth. An example of PaaS is Google AppEngine.

C. Infrastructure as a Service (IaaS)

Cloud consumers directly use IT infrastructures (processing, storage, networks and other fundamental computing resources) provided in the IaaS cloud. Virtualization is extensively used in IaaS cloud in order to integrate/decompose physical resources in an ad-hoc manner to meet growing or shrinking resource demand from cloud consumers. The basic strategy of virtualization is to set up independent virtual machines (VM) that are isolated from both the underlying hardware and other VMs. Notice that this strategy is different from the multi-tenancy model, which aims to transform the application software architecture so that multiple instances (from multiple cloud consumers) can run on a single application (i.e. the same logic machine). An example of IaaS is Amazon's EC2.

D. Data as a Service (DaaS)

The delivery of virtualized storage on demand becomes a separate Cloud service - data storage service. Notice that DaaS could be seen as a special type IaaS. The motivation is that on-premise enterprise database systems are often tied in a prohibitive upfront cost in dedicated server, software license, post-delivery services and in-house IT maintenance. DaaS allows consumers to pay for what they are actually using rather than the site license for the entire database. In addition to traditional storage interfaces such as RDBMS and file systems, some DaaS offerings provide table-style abstractions that are designed to scale out to store and

retrieve a huge amount of data within a very compressed timeframe, often too large, too expensive or too slow for most commercial RDBMS to cope with. Examples of this kind of DaaS include Amazon S3, Google BigTable, and Apache HBase, etc.

1.2. Difficulties of Cloud Systems

Cloud computing can provide infinite computing resources on demand due to its high scalability in nature, which fulfills the needs of the large number of customers. There are several difficulties associated with the services of cloud computing. They are listed below,

I. Security & Privacy Issues

There are several common security threats in cloud computing paradigm. This category includes organizational and technical issues related to keeping cloud services at an acceptable level of information security and data privacy. This includes ensuring security and privacy of sensitive data held by banks, medical and research facilities [17].

II. Infrastructure

This deals with the issues relating with hardware layer of cloud services along with the software used to operate this hardware. Our proposed issue belongs to this category.

III. Data Handling

This category deals with the data storage type of problems like, data segmentation and recovery, data resiliency, data fragmentation and duplication, data retrieval, data provenance, data anonymization and placements etc.

In this paper, we mainly focus on the symmetric distribution of workloads among processors in cloud system. It is commonly found that the CPUs are not working with proper distribution of workloads. Sometimes a set of CPUs stay idle or less loaded while others are too much loaded. Though the system has the high functioning CPUs, but this type of behavior of the system makes the performance too much degraded from the actual or theoretical value. So several cloud systems face the challenge of being maintaining its speed by sharing loads or tasks to the idler or less busy CPUs.

Figure 1. Basic Cloud Architecture

2. Problems of Load Balancing

For a cloud system the tasks are distributed over the system and every system has two parts (a) Task Reception and (b) Task Migration. These two are also known as 'Node States Information' [1]. Here node is actually the system CPU or server. For a general task given to the node the expected waiting time is (at node i)

$$W_i(t) = q_i(t) \cdot t_s \quad (1)$$

Where, $W_i(t)$ is the waiting time for task, q_i is the queue length of the task and t_s is average time or completion of a task.

Now the load balancing concept says that there are communicating nodes in between two local domains to communicate the data with the global domain. These nodes are actually the nodes which are overlapped in more than one domains. In this type of load sharing N : Number of nodes, $V = \{1, 2 \dots N\}$ a set of nodes in a system.

Here we calculate a maximum load to be executed in the node and then proceed to share the load to the other nodes.

For the algorithm the node's speed and task completion rate are also calculated. But in this instance some possible drawbacks are not calculated.

Here those drawbacks are,

- 1) If the communicating node fails then the whole load balancing will be stopped or harmed partially.
- 2) For two domains there is a possibility of communication delay.
- 3) Communication overhead problem.

This paper focuses on this particular problem of waiting time of the tasks or jobs. The given approach is to solve waiting problem as well as simplistic approach to the communication delay [8, 9, 10, 15].

3. Proposed Approach

The proposed approach for effective load balancing is divided into three sub-sections, they are as follows,

3.1. Queue Making and Job Making Processes

For every CPU there is a speed limitation. This speed is function depending on variable like instruction set, cache, clock speed, bandwidth, generated heat and heat dissipation. Depending on all these variables the clock speed is calculated and then the average task completion time is also calculated.

According to the task completion average time there is a queue for every CPU. This queue is actually a list of tasks which also contains a number called "threshold". This threshold is a general indicator which tells the task that beyond this point the tasks have to wait longer than the expectation. Thus the task is ready to be migrated to the node which is idler. Using these three processes will increase the speed of the system. It is also converted the system into a

backup planned system which is adaptive to the speed of the nodes and also to the queue and server status.

If we try to think that there are n numbers of CPUs and k number of tasks with their own different characteristics. So we can get a working algorithm for queue making in the following way:

a) CPU rearrangement:

Arranging all the available CPUs from high to low in according to their functional speed.

b) Rearrange Job requests:

Priority basis

- i. If there is any priority assigned to the jobs then the jobs should be rearranged according to the priority. In this rearranged list the highest priority job will be in the first position and the least one at the last position.
- ii. If any job is not assigned with priority then the job is set to execute with the lowest priority and the priority index is set thus.

Without priority basis:

When there is no priority is assigned for the listed jobs then the jobs are arranged by their estimated completion time. The estimated time depends on code of lines, loops, inner loops and resources required. Jobs are rearranged from highest to lowest order.

c) CPU allocation:

The rearranged CPU list gets the rearranged job list and then the CPUs are given the jobs as like first CPU in the list gets the first job and the last CPU gets the last job until all the CPUs are occupied then the list of jobs start to allocate themselves from the start of the CPU list.

d) Completion time estimation:

The speed of the CPU and the required CPU cycle of the job give the general completion time of the job.

e) Runtime new request allocation:

When a new job is arrived then the job is included to the initial list of jobs that are not being executed by the CPUs. Then the list goes through the step (ii) to (iv). This way a new job is inserted in the existing list of jobs and allocated to the available CPUs.

The above job making and queue making processes are represented in the figure2 and figure3 respectively.

3.2. Timeout Chart

For a particular task queue, one task is migrated to a node but then task is delayed because of longer waiting time. Again this may happen that searching or migration takes so long time that it is actually exceeding the first node completion time. The third scenario may be such that one task is migrated and processed very quickly but the transfer

time for the task is too high to wait or the task. To overcome all these problems there must a timeout chart available which will work as a look ahead table and also to take a decision whether the task should be migrated or not. Now if it is seen that the total system is not working and it is becoming too slow then also this timeout chart work as a very good reference to eliminate or terminate jobs. That is if the job completion is not executing properly then the chart provides us the time of termination and then we can decide which jobs are to be completed within this time and which are to be terminated. This process also checks the priority and job types. As if there is no priority set initially then we can have many different jobs. Among those jobs some are reversible or have roll back option, others may not have that option. In that case if a job without rollback option is being executed then that job will not be terminated before completion.

3.3. Super Node as Proxy Node

All the communicating nodes of the global domain of cloud are connected to one super node with data base of required characteristic of these nodes. In such a case, if a

failure happens in one of these nodes then the super node will provide all the information to make the system work properly. Here this super node actually a proxy node which keeps all the copy of the information and updates it in time to time. But it is not searched in every case as it is part of the large database. But it is updated without delaying the system that the system may have a backup plan at the time of failure. The entire process is shown in figure 4.

4. QTS in Cloud Computing

The term QTS is actually a combination of those three processes which have been discussed previously. In cloud environment two mostly available and used cloud architectures are Amazon Web Services (AWS) [11] and Google App Engine (GAE) [12]. In the proposed model the QTS comes in between web server and VaR Analytic Server (GAE) and EC2 Instance VaR Server (AWS). This is the proposed approach of the paper to improvise the migration process so that the performance gets better or the system.

Figure 2. Job making Process

Figure 3. Queue Making Process

Figure 4. Functionality of Super node

Figure 5. Working process of AWS/GAE without QTS

Figure 6. QTS applied in GAE

Figure 7. QTS applied in AWS

5. Future Research Issues

This paper also tries to highlight the changes happening in the field of load balancing algorithms across the world.

- Maximum resource utilization is one of the main objectives in any distributed load balancing system. In future, this load balancing algorithm can be associated with some optimization techniques (PSO, ACO, GA etc.) to give more flexibility to distributing the loads and improving the performance of the system.
- Besides load balancing, fault tolerant in cloud system needs to be incorporated in order to remove security and usage issues and to design the approach more cost effective and successful.
- The recent cloud load balancing solutions are focusing on Green cloud topic in load balancing mechanisms by considering the challenges like: Reducing the energy and power consumption, Reducing carbon emission and also reducing cost customers as a result of energy efficiency topic.
- There must be devised some dynamic load balancing algorithm which can set the load degree high and low dynamically. In future research issues, some sort of fuzzy logic may be associated with load balancing techniques to make it more flexible.

6. Conclusions

This paper presents the basics of cloud computing along with research challenges in load balancing. It also focuses on merits and demerits of the cloud computing. There is the process of dynamic approach to allocate the jobs to suitable CPUs. Other features are to check the delay and accordingly terminating a process as well as migrating them to other

CPUs so that the server can be utilized more efficiently. This procedure is a theoretical and mathematically evaluated approach. The paper includes an algorithm to support the approach. There is also a manually job migration and allocation through various table and possible dynamic scenario. Although cloud computing is a new and well attractive feature in the computer science area it has some of its own problems. This paper does not try to change any cloud architecture, it is only an add-on to the existing process to improvise its filtration and make it more efficient.

REFERENCES

- [1] P. Neelakantan 'AN ADAPTIVE LOAD SHARING ALGORITHM FOR HETEROGENEOUS DISTRIBUTED SYSTEM', International Journal of Research in Computer Science, ISSN 2249-8265 Volume 3 Issue 3 (2013) pp. 9-15www.ijrcs.org, A Unit of White Globe Publicationsdoi:0.7815/ijrcs. 33.2013.063.
- [2] Mayanka Katyal, Atul Mishra 'A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment', International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013.
- [3] Doddini Probhuling L. 'LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING', International Journal of Advanced Computer and Mathematical Sciences, ISSN 2230-9624. Vol4, Issue3, 2013, pp229-233http://bipublication.com.
- [4] P. Mohamed Shameem, R.S Shaji 'A Methodological Survey on Load Balancing Techniques in Cloud Computing', International Journal of Engineering and Technology (IJET)ISSN: 0975-4024 Vol 5 No 5 Oct-Nov 2013.
- [5] Urjashree Patil, Rajashree Shedge 'Improved Hybrid Dynamic

- Load Balancing Algorithm for Distributed Environment' International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013, ISSN 2250-3153.
- [6] C.H.Hsu and J.W.Liu (2010) "Dynamic Load Balancing Algorithms in Homogeneous Distributed System," Proceedings of The 6th International Conference on Distributed Computing Systems, , pp.216-223.
- [7] Carnegie Mellon, Grace Lewis(2010) "Basics About Cloud Computing" Software Engineering Institute September.
- [8] R.R. Kotkondawar, P.A. Khair, M.C. Akewar and Y.N. Patil, "A Study of Effective Load Balancing Approaches in Cloud Computing", International Journal of Computer Applications, Vol.87, No.8, 2014.
- [9] G. Joshi and S. K. Verma, "A Review on Load Balancing Approach in Cloud Computing", International Journal of Computer Applications, Vol.119, No.20, 2015.
- [10] Amandeep, V. Yadav and F. Mohammad, "Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study", International Journal of Scientific Research Engineering & Technology, Volume 3 Issue 1, April 2014.
- [11] D. Chitra Devi and V. Rhymend Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non-preemptive Dependent Tasks", The Scientific World Journal, Hindawi, Volume 2016.
- [12] Dan C. Marinescu, "Cloud Computing: Theory and Practice", Morgan Kaufmann, ISBN-13: 978-0124046276, 2013.
- [13] Rajkumar Buyya, James Broberg, Andrzej Goscinski, "CLOUD COMPUTING Principles and Paradigms"
- [14] Amazon Web Services Whitepapers (<http://aws.amazon.com/de/whitepapers/>).
- [15] Google Cloud Platform (<https://cloud.google.com/appengine/docs>).
- [16] S. Kumar and R.H.Goudar, "Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey", International Journal of Future Computer and Communication, Vol. 1, No. 4, December 2012.
- [17] Y. Ghanam, J. Ferreira, F. Maurer, "Emerging Issues & Challenges in Cloud Computing— A Hybrid Approach", Journal of Software Engineering and Applications, Vol.5, 2012, pp. 923-937.
- [18] Nidal M. Turab et al., "CLOUD COMPUTING CHALLENGES AND SOLUTIONS", International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.5, September 2013.