Journal of Theoretical and Applied Information Technology
30th April 2014. Vol. 62 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL APPROACH FOR DYNAMIC CLOUD PARTITIONING AND LOAD BALANCING IN CLOUD COMPUTING ENVIRONMENT

[1]SUGUNA R, DIVYA MOHANDASS, RANJANI R
[1]Professor and Dean, Department of CSE, SKR Engineering College, Chennai, India.
[2]Professor, Department of CSE, SKR Engineering College, Chennai, India.
[3]PG Scholar, Department of CSE, SKR Engineering College, Chennai, India.
Email : [1]dean.cse@skrenggcollege.org , [2]divya.skrec@gmail.com , [3]ranjanisureshkumar@gmail.com

ABSTRACT

Cloud Computing is an attracting technology in the field of computer science. Cloud Computing involves sharing of resources. In the event of processing many jobs, the load balancing becomes essential for efficient operation and to improve user satisfaction. This paper introduces the strategic model that performs load balancing as well as dynamic partition of the nodes of different cloud. Different Load balancing strategy is applied for different situations. The algorithm applies game theory to load balancing strategy to improve the efficiency in the cloud environment.

Keywords—*Load Balancing; Dynamic Partition, Game Theory, Cloud Partitioning*

1. INTRODUCTION

Cloud Computing describes a variety of computing concepts that involves a large number of computers connected through an internet. Load balancing aims to optimize resource use, maximize throughput, maximize response time, and avoid overload of any resources. Cloud computing is efficient yet load balancing is major issue to address upon [2] [3].

Load balancing is a computer networking scheme for distributing workloads across numerous computing resources, such as computers, a cluster, network links, central processing units or disk drives. Load balancing aims to improve resource usage, maximize throughput, minimize response time, and avoid overload of any one of the resources. Usage of multiple components with load balancing as an alternative for a single component may enhance reliability through redundancy.

The purpose of load balancing is to make every processor or machine to perform the same amount of work throughout the network which helps in increasing the throughput, minimizing the response time and reducing the number of job rejection [14]. The scalability of network is generally marked by efficient usage of the resources available. This could only be achieved when the load in each and every node is maintained with stability.

The job arrival pattern in the cloud cannot be predictable and its complexity is also more. Load balancing scheme depending on whether the system are important can be either static or dynamic [6]. Static scheme do not use the system information and are less complex while dynamic schemes may incur additional cost but they are more flexible.

The model includes the partition manager and job distributor that gathers and analyses the status information of each node in the partition. In this model the cloud is divided into several partitions thus, it may simplify the handling of load. Depending on the situation the nodes of different partitions could be combined to increase the number of partitions. The partition manager chooses the suitable partition and may create new partition for arriving jobs while the job distributor chooses the best load balancing strategy. The allocation of job to a particular partition is carried out by the Partition manager and load distribution in that single partition is the responsibility of the job distributor. This distribution may adopt any of the existing algorithms such as Round Robin algorithm, First Come First Serve or even active clustering and of its similar kind to perform the balancing mechanism. The approach that has been

Journal of Theoretical and Applied Information Technology
30th April 2014. Vol. 62 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

adopted for implementation purpose is Honeybee Foraging Behavior.

Generally static partitions are adopted for load balancing which usually increase the number of job rejections. This issue is resolved by allowing dynamic partition for new services thereby improving the scalability.

2. RELATED WORKS

There have been many studies of load balancing for the cloud environment. Load balancing in cloud computing was described in a white paper written by Adler [7] who introduced the tools and techniques commonly used for load balancing in the cloud. However, load balancing mechanism adopted in the cloud is still needs better steps to adapt hasty changes [4][5] . Chaczko et al. [8] described the part that load balancing plays in improving the performance and stability.

There are numerous load balancing algorithms, such as Round Robin algorithm, Equally Spread Current Execution Algorithm, and Ant Colony algorithm. Nishant et al. [9] used the ant colony optimization method in nodes for load balancing. Randles et al. [10] gave a comparative analysis of some algorithms in cloud computing by checking the performance cost and time. Each algorithm has its own pros and cons. Their study reveals that the ESCE algorithm and throttled algorithm are enhanced than the Round Robin algorithm.

Some of the classical loads balancing methods are similar to the allocation method in the operating system such as Round Robin algorithm and the First Come First Served (FCFS) rules.

There are approaches inspired by Honeybee Foraging Behavior, Active Clustering and Biased Random Sampling methods [8]. In Honeybee Foraging Behavior mechanism an advert is build that stores the status of each node, thus when a job has to be assigned it is done using the advert. The advert is updated periodically. In Active Clustering the nodes are combined for the execution of jobs. When Biased Random Sampling is considered the status report is generated from collected random samples. The Honeybee Forging Behavior is used for the proposed work because it is fairly efficient.

3. SYSTEM MODEL

A cloud will include many nodes and the nodes may be in diverse geographical locations. Cloud partitioning is used to manage this hefty cloud. A cloud partition is a subarea of the cloud with divisions based on the geographic locations.

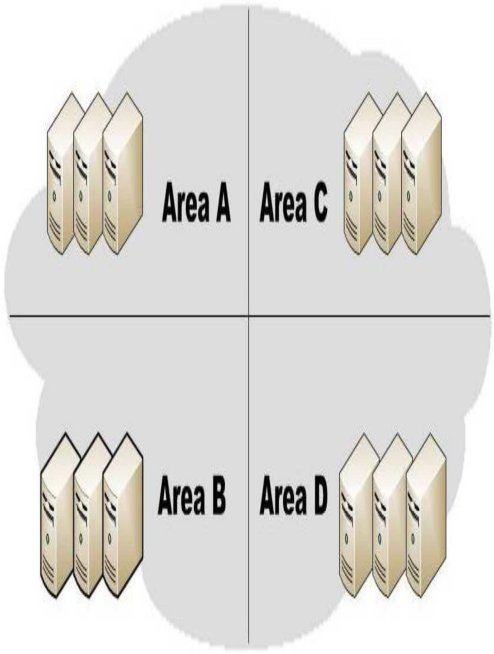

Figure 1: Typical Cloud Partitioning

The typical cloud partitioning is depicted in Figure 1. The division is mainly based on the geographical separation of the nodes. Each location with set of nodes is considered as separate partition and is managed by job distributors.

The load balancing approach is based on the cloud partitioning concept. After creating the cloud partitions, then load balancing the procedure begins: The job arrives and the partitions manager decides in which partition the job has to be executed and check is there any requirement for creation of new partition if required new partition is created or else in the existing partition the job is submitted.

Then the job distributor decides how to assign the jobs to the individual nodes in that partition. The choice of using the load balancing strategy may depend on the job distributors. Even multiple strategies could be combined and used for favorable situations.

Journal of Theoretical and Applied Information Technology
30th April 2014. Vol. 62 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The complete process in the system model is shown in Figure.2

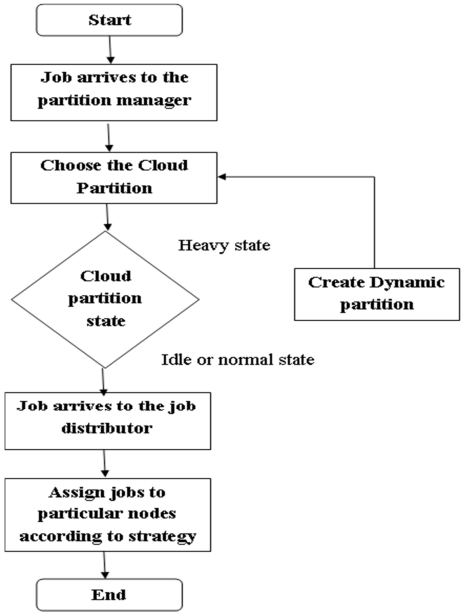

Figure 2: System Model

3.1 Partition Manager And Job Distributor

The load balance solution is done by the partition manager and the job distributors. The partition manager first assigns jobs to the suitable cloud partition and then communicates with the job distributors in each partition to refresh this status information. Since the partition manager deals with information from each partition, smaller data sets will lead to the high processing rates. The job distributors in each partition gather the status information from every node and then choose the right strategy to distribute the jobs.

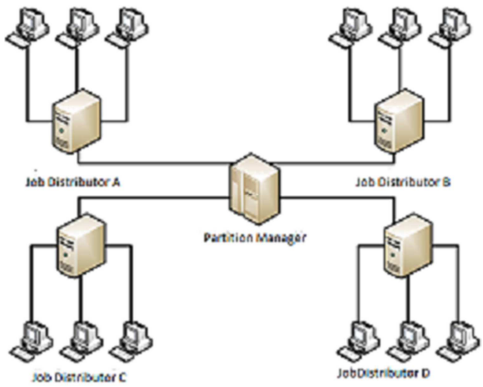

Figure 3: Relationship between Job Distributor and Partition Manager

The relationship between the job distributor and the partition manager is shown in Figure.3. The job distributors in each partition links to the partition manager. This partition manager allocates the jobs to different partition based on the status report it maintains. [1]

3.3 Assigning Jobs To The Cloud Partition

When a job arrives at the public cloud, the first step is to choose the suitable partition. The cloud partition status can be divided into three types:

a. Idle:Load_degree(N) = 0
b. Normal:0<Load_degree(N)<$Load\_degree_{high}$
c. Overload:Load_degree(N)>$Load\_degree_{high}$

The parameters are set by the job distributors. The load degree could be evaluated from the number of CPU cycles involved and also the process execution time involved. The load degree threshold limit is considered based on the current status of the partitions. The partition manager has to communicate with the job distributors frequently to refresh the status information. The partition manager then dispatches the jobs using the following strategy: When a job arrives at the system, the partition manager queries the cloud partition where job is to be allocated.

If status is idle or normal, the job is handled in the same node or else, another cloud partition is found that is not overloaded.

3.3 Assigning Jobs To The Nodes In The Cloud Partition

Cloud partition job distributor gathers load information from every node to evaluate the cloud partition status. This evaluation of each node's load status is very essential. The first task is to characterize the load degree of each node. The node load degree is related to various static parameters and dynamic parameters. The static parameters include the number of CPU's, the memory size, the CPU processing speeds etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc. Based on the parameters the jobs are assigned to the nodes in the partitions.

Journal of Theoretical and Applied Information Technology
30th April 2014. Vol. 62 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.4 Cloud Partitioning Strategy

The Partition Manager and Job Distributor are communicated to determine which partition is capable of handling a particular job. The selection of cloud partition is determined based on the number of jobs running, number of jobs to be allocated, number of nodes available, number of nodes idle.

When a node is continuing execution and a specific type of job is being requested, the cloud has to be reshaped so that the load among the partitions can be distributed effectively. For this purpose partitions are created dynamically. This is done using the honey bee algorithm. The steps involved are

1. All nodes of each partition are monitored and reported to the partition manager.
2. The advert is built based on the reports that are collected.
3. The Partition manager identifies the potential nodes for extraction. When the load on one partition gets high or if in certain partition nodes remains idle for long time then the potential nodes are extracted and a new partition is created using the extracted nodes which gives a new partition which is lightly loaded.

The Job distributor then works on distributing the new requests to the dynamically created node. Mean while the highly loaded node might get unloaded which is monitored again by the Partition manager and marked for extraction and the cycle gets repeated. The method adopted in the job distribution among the nodes of same partition is performed by Map Reduce. This involves the following steps:

Step 1: Job is divided into sub tasks using default class methods and is mapped to different nodes.

Step 2: The mapped task is reduced based on the service implemented

Step 3: Then the results are combined sorted and replied with the correct output for the request initiated.

Step 4: The output is stored in the Distributed File System.

4. IMPLEMENTATION

The framework that has been used in the implementation of the cloud is through Hadoop. A Hadoop cluster is a special type of computational cluster designed specifically for storing and analyzing huge amounts of unstructured data in a distributed computing environment.

Such clusters run Hadoop's open source distributed processing software on low-cost commodity computers. Typically one machine in the cluster is designated as the NameNode and another machine the as JobTracker; these are the masters. The rest of the machines in the cluster act as both DataNode and TaskTracker; these are the slaves. Hadoop clusters are often referred to as "shared nothing" systems because the only thing that is shared between nodes is the network that connects them.

Hadoop clusters are known for boosting the speed of data analysis applications. They also are highly scalable: If a cluster's processing power is overwhelmed by growing volumes of data, additional cluster nodes can be added to increase throughput. Hadoop clusters also are highly resistant to failure because each piece of data is copied onto other cluster nodes, which ensures that the data is not lost if one node fails.

Initially the base cloud setup is established into two distinctive partitions. The Criteria for a partition is a group of nodes controlled by a master node. Based on these criteria two distinct partitions are formed with each containing two slave nodes and a master node. The master node manages and balances the data load and the computation load. The data are distributed across each node and replicated for data recovery if a node fails. This is achieved by the DFS Distributed File System of the cloud and the Name node of the master. The name node manages the distribution of data and replication.

Whenever a file is read or written the master node queries the name node which in turn enquires all the data nodes which are present in each slave. Whenever a job is submitted to the master, the job is maintained in a job queue which is then processed by the job tracker in the master node. The Job Tracker distributes the task to task tracker of the slaves in way that the computation can be done parallel.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The Job Tracker distributes the job based on the load of each node. If one node has higher load then it allocates the job to the node which is capable of handling the job.

The Cloud provides SaaS (Software as a Service) to the user. The SaaS that runs on each partition is capable of counting how many times each word is repeated in the given book. This service uses the Map Reduce Algorithm which divides the task into Maps that are distributed to each slave and a reduce task combines the result form each node and distributes the data across the nodes using the DFS and then the data is replicated. The Word count program is given the input in the form of a text file. The prime number generator is also provided another type of services.

When these services are submitted in the cloud environment for execution map is responsible for mapping the task in different nodes for execution. The reduce part performs the execution to generate the result.

The completion graph in Figure.4 depicts the operation of Map and Reduce operation.

The Map graph shows the number of map task generated and its percentage of completion and the reduce graph shows reduce, sort and copy operation and its percentage of completions.

The dynamic partitions are generated from the status information updated in partition manager. Thus nodes that are idle could be combined and used for providing different service creating a new partition.

The comparison of MapReduce load balancing algorithm with Central Load Balancing Decision Model (CLBDM), Ant colony has been done based on performance parameters is depicted in Table 1. CLBDM is an improvement of the Round Robin Algorithm which is based on session switching at the application layer [7]. Ant Colony algorithm works as follows: once a request is initiated the ants and pheromones are initiated and the ants start their forward path from the 'head' node. A forward movement means that the ant is moving from one overloaded node looking for the next node to check if it is overloaded or not. Moreover, if the ant finds an under loaded node, it will continue its forward path to check the next node. If the next node is an overloaded node, the ant will use the backward movement to get to the previous node. The addition in the algorithm proposed in [9] is that the ant will commit suicide once it finds the target node, which will prevent unnecessary backward movements.

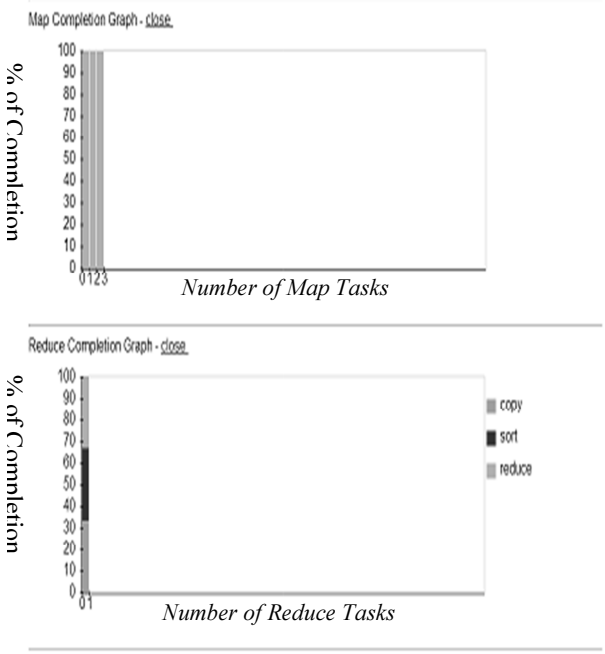

Figure 4: The Completion Graph of Map and Reduce Operation

Table: 1 Comparison Of Load Balancing Algorithm

	CLBDM	Ant Colony	Map Reduce
Replication	Full	Full	Full
Heterogeneity	Yes	No	Yes
Spatially Distributed	Yes	No	Yes
Fault Tolerance	No	Yes	Yes

5. CONCLUSION AND FUTURE WORK

The cloud partitions are individually handled to reduce the complexity of load balancing. The partitions are also customizable i.e., new partitions could be created by using the status report maintained by the job distributor and partition manager. The future work will be to concentrate in creating the higher level of transparency and generating effective technique in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

updating the status report. The time interval should be managed. The balancing technique for each partition could be made dynamic.

A framework will be needed for cloud division methodology as in current work only geographically the divisions are made. There may be some clusters in same geographic location that are still far apart to join the same partition.

REFERENCES

[1] Xu, Gaochao; Pang, Junjie; Fu, Xiaodong, "A load balancing model based on cloud partitioning for the public cloud", *Tsinghua Science and Technology*, Vol.18, No.1, Feb. 2013, pp.34, 39.

[2] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed internet computing for IT and scientific research", Internet Computing, vol.13, no.5, Sept.-Oct. 2009, pp.10-13.

[3] P. Mell and T. Grance, The NIST definition of cloudcomputing,http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2012.

[4]Microsoft Academic Research, Cloud computing,http://libra.msra.cn/Keyword/6051/cloud-computing?query=Cloud%20computing,2012.

[5]GoogleTrends,Cloudcomputing,http://www.google.com/trends/explore#q=cloud%20computing, 2012.

[6] N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.

[7]] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." *in proc.34th International*
Convention on MIPRO, IEEE, 2011.

[8] Martin Randles, David Lamb, A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," *IEEE 24th International Conference on Advanced Information Networking and Applications Workshop*s, 2010, pp.551-556.

[9] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, "Load balancing 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.

[10] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing", *in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications,* Perth, Australia, 2010, pp. 551-556.

[11]S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems", J*ournal parallel and distributed computing*, Vol.71, Apr. 2011, pp. no 537-555

[12] D. Grosu, A. T. Chronopoulos, and M. Y. Leung, "Load balancing in distributed systems: An approach using cooperative games", *in Proc. 16th IEEE Intl. Parallel and Distributed Processing Symp*., Florida, USA, Apr. 2002, pp..No 52-61.

[13] S. Aote and M. U. Kharat, "A game-theoretic model for dynamic load balancing in distributed systems", *in Proc. the International Conference on Advances in Computing, Communication and Control (ICAC3 '09),* New York, USA, 2009, pp. 235-238.

[14] Aarti Khetan, Vivek Bhushan ,Subhash Chand Gupta "A Novel Survey on Load Balancing in Cloud Computing " *International Journal of Engineering Research & Technology (IJERT)* Vol. 2 Issue 2, February- 2013 ISSN: 2278-0181

[15] Nidhi Jain Kansaland Inderveer Chana "Existing load balancing techniques in cloud computing: a systematic re-view" *Journal of Information Systems and Communication ISSN: 0976-8742, E-ISSN: 0976-8750,* Vol. 3, No. 1, 2012, pp- 87-91.

[16] Amandeep Kaur Sidhu,Supriya Kinger "Analysis of Load Balancing Techniques in Cloud Computing", *International Journal of Computers & Technology* Vol. 4 No. 2, March-April, 2013, ISSN 2277-3061.