ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

A Load Balanced Greening Approach for Proficient Resource Allocation with Cloud Partitioning

R.Divya[1], S. Premkumar deepak[2]

P.G Student, M.E CSE, Parisutham Institute of Technology and Science, Thanjavur, T.N, India[1]

Guide, Parisutham Institute of Technology and Science, Thanjavur, T.N, India[2]

ABSTRACT— Cloud computing is mainly used in load balancing technique. It has improved performance and user satisfaction. Cloud providing users with new type of services. Here different type of strategies can be used. Previously they have used ANT colony optimization and Game theory method. Here job arrival pattern is not predictable. In this method not flexible and also not efficient. Load balancing schemes depending on either static or dynamic. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy. The load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions, the load balancing then starts. When a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition. When the node is idle, the data will be directly shared to another node inside the partition by using green computing concept. Normally, cloud computing is a cost per usage. When transfer the image to the cloud, the size of the image can be reduced by the decomposition technique. This whole project mainly deals with reducing data size and cloud space.

KEYWORDS: Cloud Partitioning, Load Balancing, Green computing.

I. INTRODUCTION

The cloud will bring changes to the IT industry. The cloud is changing our life by providing users with new types of services. NIST gave a definition of cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The load balancing model given in this paper is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. When a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

The load degree results are input into the Load Status. Tables created by the cloud partition balancers. Each balancer has a Load Status Table and refreshes it each fixed period time(T). The table is then used by the balancers to calculate the partition status. Each partition status has a different load balancing solution. When a job arrives at a cloud partition, the balancer assigns the job to the nodes based on its current load strategy. This strategy is changed by the balancers as the cloud partition status changes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service, for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs while applications are running. However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink.

The given set of files and store them across the servers in a space and access-efficient manner. In the process, we reduce the network-overhead for file access. Our main goal is to minimize the space required by the access-efficient SAP solution. For this small example, we are able to determine the optimal partition simply by inspection. For a general set of files it is a very computationally expensive task to compute all possible partitions, and then identify the ones which capture file similarity better. In this paper, we present polynomial time algorithms to partition the given set of files and store them across the servers in a space and access-efficient manner. A space-efficient solution, in which we identify unique chunks across all the files and distribute them arbitrarily across the servers (chunk-distribution), may require communication with more than one server to access any file. On the other hand, an access efficient solution in which we randomly partition the files across the servers, and then store their unique chunks on each server may not exploit the similarities across files to reduce the space overhead.

In the context of grayscale images, we consider the 8 connected pixel adjacency graph as a simplifies complex. It is obvious that using the pixel values as a Morse function gives valid segmentations only in special cases, as it is only capable of isolating darker regions separated by lighter boundaries (or vice versa). Thus, as it is common with watershed techniques, we use the absolute value of the image. For color images we have implemented the method of the original, inspired by the mean shift methods. We first build a nearest neighbor graph from the point cloud representing the image in color space. It is widely known, those Euclidean distances in common color spaces like RGB do not correspond well with perceived color differences, and so, the image is first transformed into the more suitable color space. Then, the so-called Vietoris-Rips graph is generated by connecting points that are closer to each other than some given parameter d. The segmentation is produced by clustering the point cloud.

II. RELATED WORK AND EXISTING MODEL

Using the methodology is ANT COLONY optimization. In this method only used for check performance. Client send data to receiver. But it will not check the receiver is free to execute or not. In the natural world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. Thus, when one ant finds a good (i.e., short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads to all the ants' following a single path. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

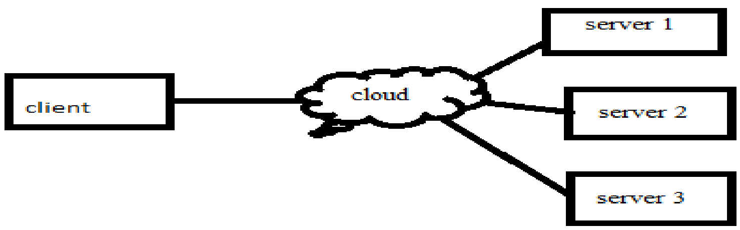

Fig 2.1:Existing Model

Performance Analysis

In compared analysis of some algorithms in cloud computing by checking the performance time and Cost. ANT COLONY optimization compared to Game theory it has less efficiency. Game theory is more flexible and maintains user satisfaction

Problem statements

In Existing job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is crucial to improve system performance and maintain stability. Cloud computing is efficient and scalable but maintaining the stability of processing is very difficult here. So many jobs in the cloud computing environment are a very complex problem with load balancing.

III. PROPOSED MODEL

Objective

Using algorithm is SAP. The load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions the load balancing then starts. When a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

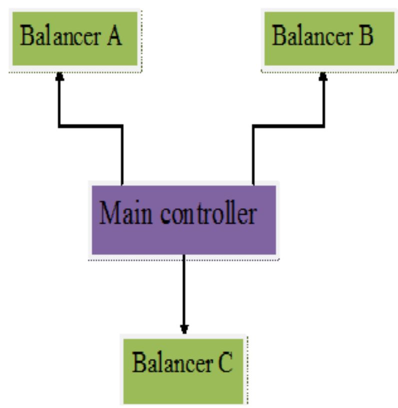

Fig 3.1:Proposed Model

Similarity aware partitioning algorithm

We consider a novel problem of storing them across a set of distributed servers in a manner that is: (i) access-efficient, (ii) space-efficient. (i) They should be access-efficient in terms of minimizing network accesses to read or write to the files in the system. This is a crucial factor with increasing network congestion in the underlying data center networks.(ii) They should be space-efficient to manage the high volumes of data. Given a set of files that show a certain degree of similarity or redundancy, we consider the problem of storing them across a set of distributed servers and present solutions that are both access-efficient and space-efficient.In this paper, we first show that finding an access-efficient, space optimal solution is an NP-Hard problem. Following this, we present the similarity-aware-partitioning (SAP) algorithms that find access-efficient solutions within polynomial time complexity and guarantees bounded space overhead for arbitrary files. Space efficiency is often achieved by the process of deduplication, which splits all the files in the system into chunks and maintains only a unique copy of each chunk. Typically, most deduplication solutions have focused on reducing the space-overhead within a single server. These techniques do not consider the problem of distributed deduplication due the cost of network accesses and file maintenance in a distributed solution. However, files with a high degree of similarity may be deduplicated on different servers. In this paper, we show that it is possible to achieve both space and access-efficiency in a distributed solution. In general, this may not be the case and we may have additional redundancy in the SAP approach. However,the given set of files and store them across the servers in a space and access-efficient manner. In the process, we reduce the network-overhead for file access. Our main goal is to minimize the space required by the access-efficient SAP solution. For this small example, we are able to determine the optimal partition simply by inspection. For a general set of files it is a very computationally expensive task to compute all possible partitions, and then identify the ones which capture file similarity better. In this paper, we present polynomial time algorithms to partition the given set of files and store them across the servers in a space and access efficient manner. A space-efficient solution, in which we identify unique chunks across all the files and distribute them arbitrarily across the servers (chunk-distribution), may require communication with more than one server to access any file. On the other hand, an access-efficient solution in which we randomly partition the files across the servers, and then store their unique chunks on each server may not exploit the similarities across files to reduce the space overhead.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Range partitioning

A range partitioner divides a data set into approximately equal size partitions based on one or more partitioning keys. Range partitioning is often a preprocessing step to performing a total sort on a data set. This topic describes the range partitioner, the partitioner that implements range partitioning. It also describes the write range map operator, which you use to construct the range map file required for range partitioning, and the stand-alone make range map utility.The range partitioner guarantees that all records with the same partitioning key values are assigned to the same partition and that the partitions are approximately equal in size so all nodes perform an equal amount of work when processing the data set. The diagram shows an example of the results of a range partition.

Property	Value
Number of input data sets	1
Number of output data sets	1
Execution mode	parallel
Partition method	modulus

Table 3.1:Operator Properties

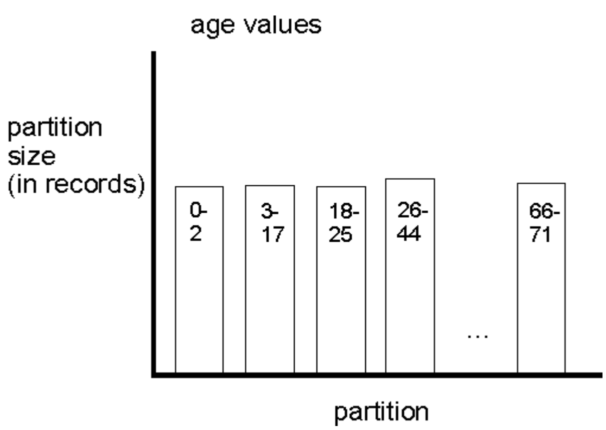

Fig 3.2:Range Partitioning Example

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

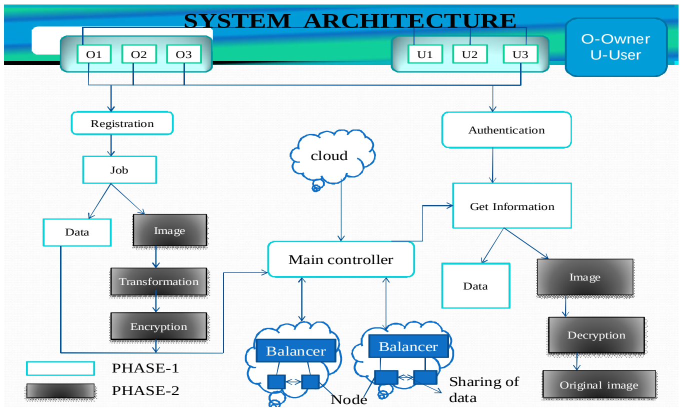

Fig 3.3:System Architecture

All partitions are of approximately the same size. In an ideal distribution, every partition would be exactly the same size. However, you typically observe small differences in partition size. In order to size the partitions, the range practitioner orders the partitioning keys. The range partitioned then calculates partition boundaries based on the partitioning keys in order to evenly distribute records to the partitions.

IV.CONCLUSION

In Formal system I found the problem's are Allocating many jobs is complex, Job arrival pattern is not depends upon priority, degradation of server is not predictable and workload control is crucial. In that, particularly i am focusing on maintaining workload and performance of server. So that i partition the cloud based on capacity of cloud. After that the job could be allocated depends upon the status of the system (System status are idle,normal,overload).If the system is idle it will automatically move to sleep mode. For that purpose green computing can be achieved through cloud computing. To minimizing the workload as well as the cost of cloud i plan to do image decomposition.

V. FUTURE ENHANCEMENT

Parallel and distributed system using decomposition of image plan to implement in cloud computing.It could be used for minimizing the workload as well as very low cost.

REFERENCES

[1] B.Adler,"Load balancing in the cloud: Tools, tips and techniques" ,http://www.rightscale.com/infocenter/whitepapers/Load-Balancing-in-the-Cloud.pdf, 2012

[2] D.MacVittie, 'intro to load balancing for developing the algorithms", https:// devcentral.f5.com/blogs/us/introto-load-balancing-for-developers-ndash-the-algorithms, 2012.

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

[4] 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.

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

[6] 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.

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

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

[9] Rouse," Public cloud", http://searchcloudcomputing. techtarget.com/definition/public-cloud, 2012.

[10] S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems", Journal of Parallel and Distributed Computing, vol. 71, no. 4,pp. 537-555, Apr. 2011.