RESEARCH ARTICLE OPEN ACCESS

Partitioning based Approach for Load Balancing Public Cloud

[1]T.V V S S C Divya,[2]V.Sireesha ,[3]S.k.Chaitanya,[4]N.Baswanth
[1, 2, 3,4] *Pursuing B.Tech (CSE) from St. Ann's College of Engineering. & Technology. Chirala, Andhra Pradesh -, 523 167 INDIA*
[5]Dr. P Harini working as Professor & HOD(CSE) in *St. Ann's College of Engineering. & Technology. Chirala, Andhra Pradesh -, 523 167 INDIA*

ABSTRACT
Load Balancing Model Based on Cloud Partitioning for the Public Cloud environment has an important impact on the performance of network load. A cloud computing system which does not use load balancing has numerous drawbacks. Now-a-days the usage of internet and related resources has increased widely. Due to this there is tremendous increase in workload. So there is uneven distribution of this workload which results in server overloading and may crash. In such systems the resources are not optimally used. Due to this the performance degrades and efficiency reduces. Cloud computing efficient and improves user satisfaction. This project is a better load balance model for public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applies the game theory for load balancing strategy to improve the efficiency in the public cloud environment.

I. INTRODUCTION

Cloud Computing is a concept that has many computers interconnected through a real time network like internet. cloud computing means distributed computing. Cloud computing enables convenient, on-demand, dynamic and reliable use of distributed computing resources. The cloud computing model has five main characteristics on demand service, broad network access, resource pooling, flexibility, measured service. Cloud computing is efficient and scalable but to maintain the stability of processing many jobs in the cloud computing is a very difficult problem. The job arrival pattern cannot be predicted and the capacities of each node in the cloud differ. Hence for balancing the usage of internet and related resources has increased widely. Due to this there is tremendous increase in workload. So there is uneven distribution of this workload which results in server overloading and may crash. In such the load, it is crucial to control workloads to improve system performance and maintain stability. The load on every cloud is variable and dependent on various factors. To handle this problem of imbalance of load on clouds and to increase its working efficiency, this paper tries to implement "A Model for load balancing by Partitioning the Public Cloud". Good load balancing makes cloud computing more efficient and also improves user satisfaction . This project is aimed at the public cloud which has numerous nodes. A system having main controller, balancers, servers and a client is implemented here. It introduces a switch mechanism to choose different strategies for different situations. This paper divides the public cloud into cloud partitions and applies different strategies to balance the load on cloud.

The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs. The relationship between the balancers and the main controller Assigning jobs to the cloud partition When a job arrives at the public cloud, the first step is to choose the right partition.

The cloud partition status can be divided into three types:

(1) Idle: When the percentage of idle nodes exceeds ¸, change to idle status.

(2) Normal: When the percentage of the normal nodes exceeds ˇ, change to normal load status.

(3) Overload: When the percentage of the overloaded nodes exceeds change to overloaded status.

II. RELATED WORK

II.I Existing System with Drawbacks

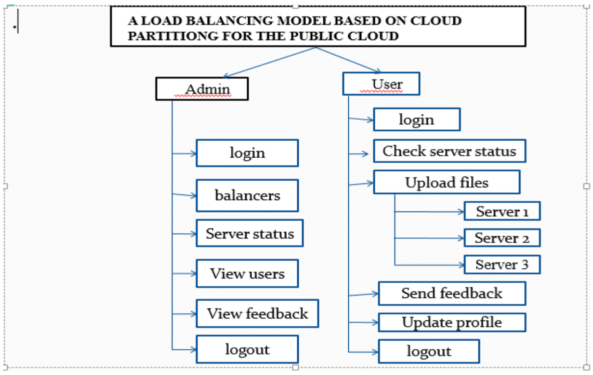
Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. Since the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is. crucial to improve system performance and maintain stability. Load balancing schemes depending on whether the system dynamics are important can be either static and dynamic . Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility.

II.II Proposed System with Features

Load balancing schemes depending on whether the system dynamics are important can be either static or dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility. The model has a main controller and balancers to gather and analyze the information. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right load balancing strategy.

The load balancing model given in this article is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. 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.

III. SYSTEM ARCHITECTURE



IV MODULES

The Following modules are :

- **User Module**
- **Admin Module**

User Module:-

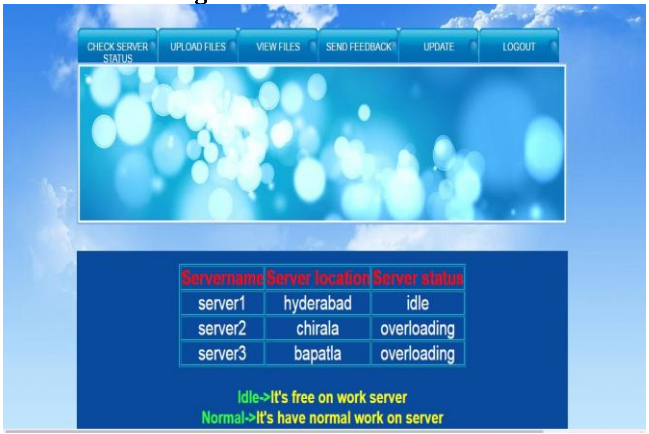
In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first.

Admin Module:

In cloud computing model admin can login into system and they can manage the servers in the module. Admin can check server status and they can balance the severs. Admin can accept the permissions which are send from user.

V. RESULTS

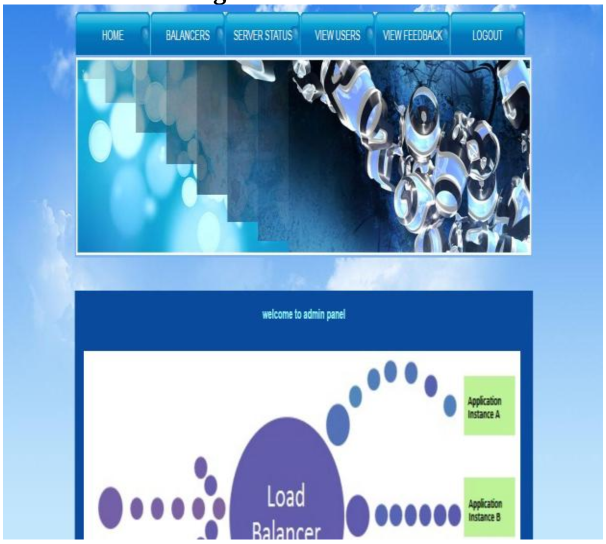
1. User Home Page



Servername	Server location	Server status
server1	hyderabad	idle
server2	chirala	overloading
server3	bapatla	overloading

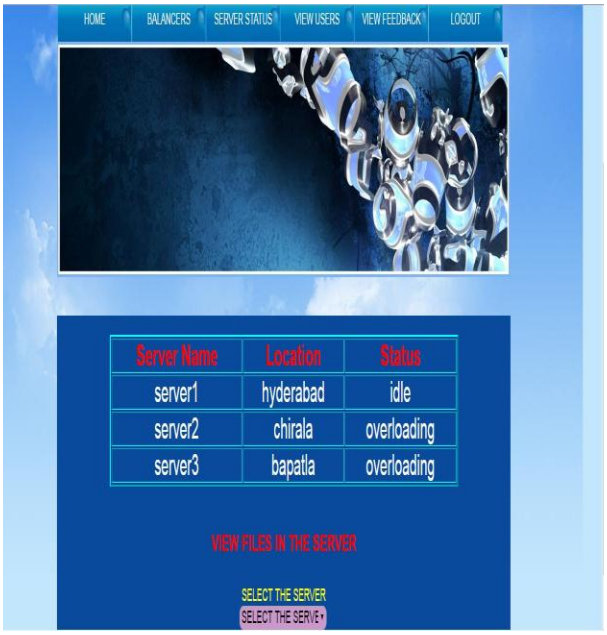
Idle->It's free on work server
Normal->It's have normal work on server

2.Admin Home Page



welcome to admin panel

3. Balancers

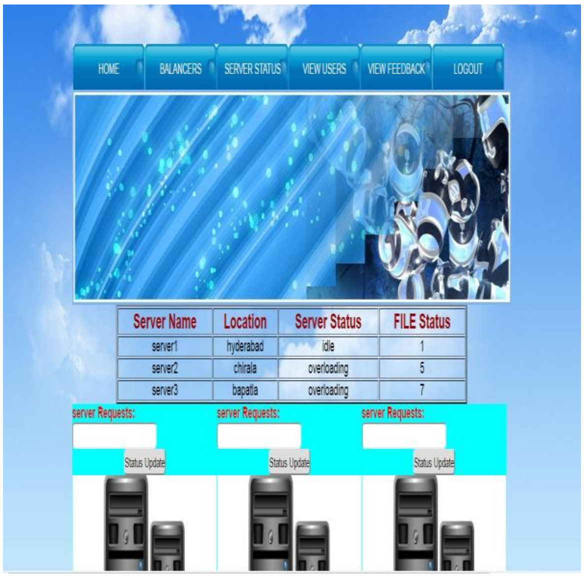


Server Name	Location	Status
server1	hyderabad	idle
server2	chirala	overloading
server3	bapatla	overloading

VIEW FILES IN THE SERVER

SELECT THE SERVER

4. Server Status



Server Name	Location	Server Status	FILE Status
server1	hyderabad	idle	1
server2	chirala	overloading	5
server3	bapatla	overloading	7

server Requests:

server Requests:

server Requests:

CONCLUSION

The overall goal of this project is to balance the load on clouds. Balancing load on the cloud will improve the performance of cloud services substantially It will prevent overloading of servers, which would otherwise degrade the performance

VII. FUTURE SCOPE

Find other load balance strategy: Other load balance strategies may provide better results, so tests are needed to compare different strategies. Many tests are needed to guarantee system availability and efficiency.

REFERENCES

[1] "Comscore," http://www.comscoredatamine.com/.

[2] A. G. Miklas, K. K. Gollu, K. K. W. Chan, S. Saroiu, P. K. Gummadi, and E. de Lara, "Exploiting social interactions in mobile systems," in Ubicomp, 2007, pp. 409–428.

[3] D. Niyato, P. Wang, W. Saad, and A. Hjørungnes, "Controlled coalitional games for cooperative mobile social networks," IEEE Transactions on Vehicular Technology, vol. 60, no. 4, pp. 1812–1824, 2011.

[4] M. Brereton, P. Roe, M. Foth, J. M. Bunker, and L. Buys, "Designing participation in agile ridesharing with mobile social software," in OZCHI, 2009, pp. 257–260.

[5] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in ICDCS, 2010, pp. 468–477.

[6] References R. Hunter, The why of cloud, http://www.gartner.com/ DisplayDocument?doc cd=226469&ref= g noreg