RESEARCH ARTICLE OPEN ACCESS

Advanced Gossip-based Approach for Resource Management in Cloud Ambience

Mr. Monilal S

Lecturer, Dept. of computer science and engineering
Govt Polytechnic, Ezhukone, Kollam - India

ABSTRACT
Cloud computing is emerging as a new paradigm of large-scale distributed computing that deciphers daily computing problems, likes of hardware , software and resource availability unhurried by computer users. In this Internet based computing where virtual shared servers provide software, infrastructure, platform, devices and other resources and hosting to customers on a pay-as-you-use basis. With the rapid growth of the Internet, many organizations increasingly rely on Web applications to deliver critical services to their customers. This paper addresses the problem of dynamic resource allocation in large cloud environments by proposing a gossip based design for dynamic load balancing and application placement. Here we deployed site hosting scenario as our test bed. Due to large web traffic, site owners rely on the web servers in cloud. This method can enhance the performance of the entire system and thus provides a reliable web services to the users.

***Keywords*: -** Cloud Computing, Resource Allocation, Dynamic Load Balancing, Application Placement, Distributed approach

I. INTRODUCTION

Cloud computing is a model for enabling 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 [1]. Now a days, cloud data centres runs thousand of machines for hosting large no: of web applications. The demands of application for system resources like memory, CPU etc are fluctuating. So in order to utilize system resources effectively modern web applications rely on dynamic resource allocation to meet their performance goals.

Managing resources at large scale while providing performance isolation and efficient use of underlying hardware is a key challenge for any cloud management software. Most virtual machine resource management, do not currently scale to the number of hosts and VMs supported by cloud service providers. In addition to scale, other challenges include heterogeneity of systems, compatibility constraints between virtual machines and underlying hardware, islands of resources created due to storage and network connectivity and limited scale of storage resources. This paper introduces a middleware component for dynamic resource allocation protocol that dynamically places site requests in web servers with the design goals such as fairness in resource allocation, adaptability and scalability. We considered this work from Platform as a Service (PaaS) perspective. The use case is depicted as shown in fig 1.a).and overall system architecture is depicted in fig 1.b).

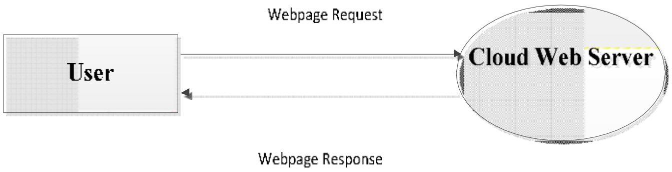
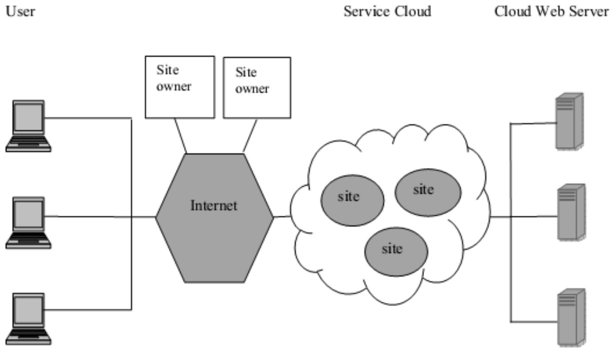


Fig 1. (a) Requestion of site in cloud



(b) System Architecture deployed in cloud site hosting

II. PROBLEM DEFINITION

A successful resource management solution for cloud environments needs to provide rich set of resource controls for better isolation, while doing initial placement and load balancing for efficient utilization of underlying resources. Load balancing is one of the central issues in cloud computing. It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance and resource utility of the system. It also ensures that every computing resource is distributed efficiently and fairly. It further prevents bottlenecks of the system which may occur due to load imbalance. When one or more components of any service fail, load balancing helps in continuation of the service by implementing fair-over, i.e. in provisioning and de-provisioning of instances of applications without fail. It also ensures that every computing resource is distributed efficiently and fairly. Scalability which is one of the very important features of cloud computing is also enabled by load balancing. The important things to consider while developing such algorithm are : estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones . This load considered can be in terms of CPU load, amount of memory used, delay or Network load. A dynamic load balancing scheme need to be proposed for avoiding over-loaded and under-loaded scenarios in cloud system. Thus the problem of dynamic Application placement should be addressed for allocating applications to machines based on their changing demands in cloud environments.Therefore in clouds always a distributed solution is required. Because it is not always practically feasible or cost efficient to maintain one or more idle services just as to fulfil the required demands.

III. EXISTING SYSTEM

Resource Management is critical in cloud computing. With improper resource management, applications might experience network congestion, long time wait, CPU wastage, overused CPU and memory, and security problems. To maximized cloud computing infrastructure utilization and minimize total cost of both the cloud computing infrastructure and running applications, resources need to be managed properly. To overcome this there are kinds of resources in the large-scale computing infrastructure need to be managed, CPU load, network bandwidth, and even type of operating systems. To provide better quality of service, resources are allocated to the users or applications based on their changing demands, via load balancing mechanism and thereby have a better application placement solution. To maximize cloud utilization, the capacity of application requirements shall be calculated so that minimal cloud computing infrastructure devices shall be procured and maintained.

The problem of dynamic application placement in response to changes in application demands have been studied before. they proposes a centralized approach and does not solves the following issues:

(a) Dynamically adapt existing placements in response to a change (in demand, capacity, etc.),

(b) Dynamically scale resources for an application beyond a single physical machine.

(d) Load Balancing approaches consider CPU resources only. Memory and bandwidth requirements of applications were not considered.

(e) Session Affinity problem.

IV. PROPOSED SYSTEM

Cloud computing provides the computing and the storage capacity as service to the users. The proposed system aims at addressing the problem of dynamic resource allocation in cloud. The dynamic allocation should be performed based on the consumers or application demands. This work focuses on dynamic allocation of the resources in the cloud environment that hosted sites. The dynamic allocation should be performed based on the consumers or application demands. It uses a gossip protocol to perform the resource allocation. In order to maximize the cloud utility the allocation process should meet certain objectives like fairness, adaptation to the changes in load and scalability. The dynamic fair allocation should satisfy all users. The requested site should be provided to the users by performing load balancing among servers so that the cloud utility can be maximized. It also solves the problem of session affinity arising during the allocation process shown in fig 3.1. Requests would be distributed among all servers by load balancing among servers and distributed sessions satisfy all user requests.

Session affinity means when a user is routed to the server they stay with that server as long as possible, so the application process have to make an out of process request to retrieve the session object from copy of session object stored in memory or in another server.

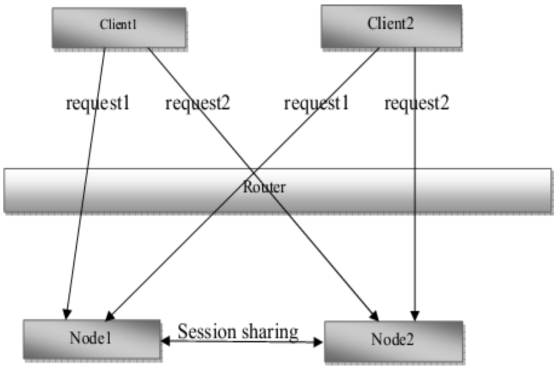


Fig 4.1: Session affinity aware resource allocation

The usual request dispatching performed at the front end level follows a sticky session approach to direct all the requests belonging to the same user session to one server in the application layer that stores the session information. If the request belongs to an existing user session, the dispatching operation selects a server on the basis of a binding table that maps user sessions to servers. If the request belongs to a new session, the dispatching selects a new server that will host the request user session. This creates a burden to the resource allocation and increases the cost of allocation.

V. GOSSIP-BASED APPROACH

Our work contributes a distributed mechanism for resource management. This gossip-based method mainly includes a load balancing component for dynamic inputs and also has a application placement scheme. For simplicity we consider memory requirement of web pages as resource for implementation purposes. The working of this technique consists of four functional phases which is illustrated as follows:

- **Cloud Web server .**The main concept in this work to develop an HTTP web server which manages the client web page requests and retrieves the required pages. Mainly html and JSP web pages were considered here. Page response time for JSP pages is high compared to html pages due to their server side processing. for that, JSP page processing system is created along with the server This server setup can easily adapt into cloud platform for web hosting and web server runs on Active and Passive nodes. This web server is considered as our test bed.
- **Load balancing module.** Load balancing modules consist of two phases:

  a) Load broadcaster: The node in which client sends their request is called active node or active server and they selected a node from network for load balancing is called passive node or passive web server. This module develops a network module for broadcasting current load on each node.

  b) Load balancing algorithm: According to above broadcast information, each active node calculate current load on each module. When a request is coming, active node check whether its current load is higher than passive node, if so allocate current job to passive node. Web page length or size is considered as current load which is different for different web pages.
- **Session Manager .**A session is specific to the user and for each user a new session is created to track the entire request from that user. Every user has a separate session and separate session variable is associated with that session. It is the part of HTTP Server. This module manages session activities. Each node has separate session manager which will cause the problem of session affinity. This problem will be addressed by distributing session between web servers or making global session manager.
- **Load Distribution.** This part of node, transfer HTTP request from active node to passive node. After processing the request it send backs result to the active server. Load distribution phase take place after load balancing step. These two components is the part of web server. Load sharing take place only if the load in passive is less compared to the load in active nodes. Otherwise, that request is handled by active itself. Load in both nodes varies according to the no: of requests given by the user.

VI. EXPERIMENTAL SETUP

In order to demonstrate site hosting environment in cloud, Cloud web server is implemented and their working is shown in prototypic version which is similar to real cloud environment rather than using a simulator. There are no other hardware requirements other than the normal system requirements. Separate web server is maintained and connected via network for the purpose of dynamic load balancing and hence achieve effective resource management. Our work is developed on JAVA platform. JAVA, being platform independent, provide a stable base for this work to be implemented properly. The system is technically compatible. The performance of this system can be expected in real time environment also.

VII. EVALUTION

The performance of gossip based resource management is plotted as graph for HTML pages and JSP pages separately. Performance metrics is depicted in fig 7.1 (a) and (b).

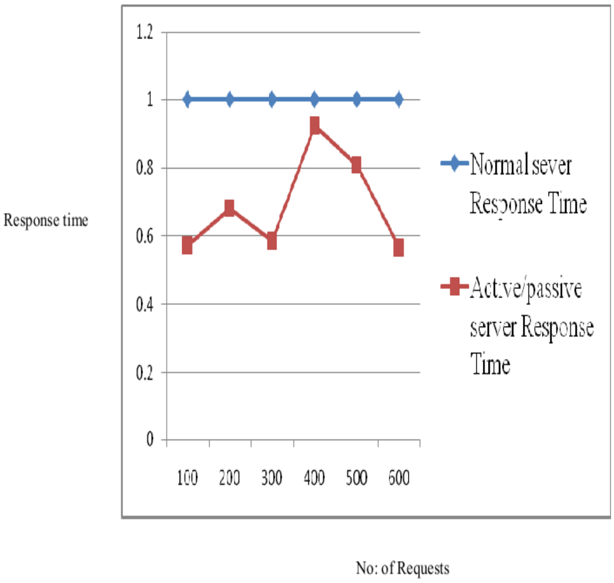


Fig 7.1 a) HTML pages

Here assumes Normal server takes Unit time (say 1sec) for processing each request i.e., for 100 request to normal server takes 100 sec. But in Gossip protocol due to dynamic load balancing some requests to the active server were handled by passive server simultaneously and average response time for each request is significantly reduced (less than 1 sec for a request).

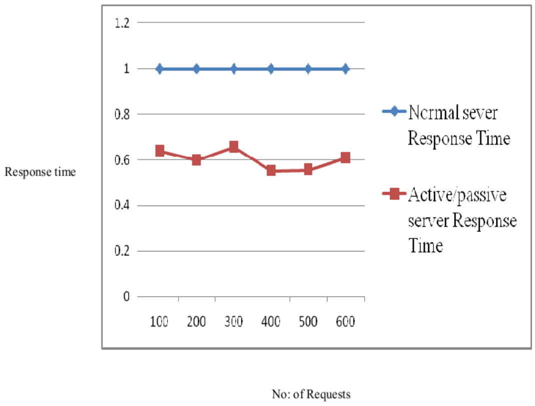


Fig 7.1 b) JSP pages

VIII. CONCLUSIONS

Most of the firms are moving to cloud environment now a days. Moving to cloud is clearly a better alternative as they can add resources based on the traffic according to a pay-per-use model. In this model “customers” plug into the “cloud” to access IT resources which are priced and provided “on-demand”. The major challenge in cloud computing is dynamically allocating resources to the users based on their demands. Resource management is critical component in this paradigm. We proposed a dynamic resource management scheme for large cloud environments that hosted sites. Our design develops a gossip based dynamic load balancing method for effective resource allocation. This approach also addresses session affinity problem in websites and manages the load across the nodes. Our work is more effective than the existing solutions due to improvement in response time. It also provides scalability with the no of applications and no of machines. The advantage of resource management system developed is that it is well suited for all types of cloud environment and hence site hosting platform can be easily deployed.

REFERENCES

[1] Alexa Huth and James Cebula, “The Basics of Cloud Computing”, US- CERT, 2011.

[2] [Fetahi Wuhib, Rolf Stadler, and Mike Spreitzer “A Gossip Protocol for Dynamic ResourceManagement in Large Cloud Environments”, IEEE Transactions On Network And Service Management, June 2012.

[3] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, “A scalable application placement controller for enterprise data centers,” ACM International Conference on World Wide Web,2007

[4] C. Adam and R. Stadler, “Service middleware for self-managing large- scale systems,”IEEE Trans. Network and Service Management, Apr. 2008.

[5] J. Famaey, W. De Cock, T. Wauters, F. De Turck, B. Dhoedt, andP. Demeester,“A latency-aware algorithm for dynamic service placementin large-scale overlays,” in 2009 International Conference on Integrated Network Management, IEEE 2009.

[6] Y. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, “Dynamic resource allocation in computing clouds

using distributed multiple criteria decision analysis,” 2010 IEEE International Conference on Cloud Computing .

[7] D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguade, “Utility- based placement of dynamic web applications with fairness goals,” in2008 IEEE Network Operations and Management Symposium

[8] F.Wuhib , R. Stadler, and M. Spreitzer, “Gossip-based resource man- agement for cloud environments,” in2010 International Conference on Network and Service Management.

[9] Z.Gong, X.Gu, and J.Wilkes, “PRESS: PRedictive Elastic Resource Scaling for cloud systems,” International Conference on Net-work and Service Management, 2010.

[10] S. Voulgaris, D. Gavidia, and M. van Steen, “CYCLON: inexpensive membership management for unstructured p2p overlays,” Network and Systems Management, 2005.

[11] C. Low, “Decentralised application placement,”Future Generation Computer Systems, 2005.

[12] Vladimir V. Korkhov, Jakub T. Moscicki,Valeria V. Krzhizhanovskaya, “Dynamic workload balancing of parallel applications with user-level schedulingon the Grid” Future Generation Computer Systems , Elsevier 2008.

[13] Zenon Chaczko , Venkatesh Mahadevan , Shahrzad Aslanzadeh and Christopher Mcdermid, “Availability and Load Balancing in Cloud Computing”, 2011 International Conference on Computer and Software Modeling.

[14] Zhen Xiao,Senior Member, IEEE,Weijia Song, and Qi Chen, “Dynamic Resource Allocation using VirtualMachines for Cloud Computing Environment” IEEE 2012.

[15] Venkatesa Kumar V,S.Palaniswami, “A Dynamic Resource Allocation Method for Parallel DataProcessing in Cloud Computing”, Journal of Computer Science 2012 .

[16] Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma “Efficient Load Balancing Algorithm in VM Cloud Environment” IJCST 2012.

[17] R. Yanggratoke, F. Wuhib, and R. Stadler, “Gossip-based resource allocation for green computing in large clouds,” International Conference on Network and Service Management, 2011.

[18] Satish, Karuturi S R V, and M Swamy Das. "Quantum Leap in Cluster Efficiency by Analyzing Cost-Benefits in Cloud Computing." In Computer Science and Engineering by Auroras Scientific Technological & Research Academy Hyderabad, vol. 17, no. 2, pp. 58-71. Accessed 2018.