IJCSEC- International Journal of Computer Science and Engineering Communications.

Vol.3, Issue 2, 2015, Page.903-909, ISSN: 2347–8586

Secure and Efficient Sharing of Resources in Dynamic Harmony Platform for Collaborative Cloud Computing

Senthilvadivu.K[1], Rasi.D[2]

[1] PG Student, [2] Assistant Professor

Department of Computer Science and Engineering,

Hindusthan College of Engineering and Technology, Tamilnadu, India.

[1]senthilvadivu.kk@gmail.com,[2]priyamudanrasi@gmail.com

ABSTRACT

The recent advances in Web technology has made it easy for any user to provide and consume content of any form. This has called for a paradigm shift in the computing architecture and large scale data processing mechanisms. It leads to a advancements of cloud computing which is a promising future for collaborative cloud computing (CCC), where Globally- scattered distributed cloud resources belonging to different organizations or individuals are collectively used in a cooperative manner to provide services. For successful implementation of Collaborative Cloud Computing resource management and reputation management need to be jointly addressed. An integrated platform called Harmony is used to deploy Collaborative Cloud Computing. Harmony integrates resource management and reputation management in a harmonious manner. It enables a node to locate desired resource and also find the reputation of the located resources. The reputation management system assigns one reputation value for each node for providing all of resource type in that node. Reputation of each type of resource is measured, so that a client can choose resource providers not only by resource availability but also by the provider's reputation. By this harmony technique outperforms existing systems by efficiency and quality of service.

Keywords *– Resource management, reputation management, Load balancing Quality of service(QoS), distributed hash table, distributed system.*

1 INTRODUCTION

The concept of cloud computing has been evolved from grid and utility computing. Now a days the cloud computing is widely used, in which cloud providers offer scalable resources to customers over the Internet. The various cloud providers are Amazon's EC2, Google's AppEngine, Microsoft's Azure, and IBM's Blue Cloud. The cloud customers are charged by the actual usage of the resources. The cloud refers to the hardware and software of datacenter that supports client needs, often in the form of data storage [1]. These infrastructures cut the cost for companies by eliminating the need for physical hardware, allowing companies to outsource data and computation resources on demand [2]. The user requested resources may not be provided by using single cloud. So multi cloud architecture can be built using virtual lab environment for petascale supercomputing capabilities. By this way idle resources can be fully utilized.

Thus advancement in cloud computing provides a way for promising future called Collaborative Cloud Computing (CCC). CCC technology interconnects physical resources and enables resource sharing among clouds, provide tremendous amount of resources to users virtually.

While many technologies are important to achieving the objective of efficient and trustworthy resource sharing, perhaps two of the most essential issues to address are resource management (resMgt) and reputation management (repMgt). ResMgt involves resource discovery and allocation for high system efficiency. A repMgt system computes each node's reputation value based on evaluations from others about its performance in order to provide guidance in selecting trust-worthy nodes for high system reliability and security [5]. However, these two issues have typically been addressed separately, despite the significant interdependencies between them; resMgt needs repMgt to provide a cooperative environment for collaborative resource sharing and in turn facilitates repMgt to evaluate multi-faceted node reputations for providing various resources.

Importance of Resource and Reputation Management:

CCC operates in a large-scale environment involving thousands or millions of resources across disparate geographically distributed areas, and it is also inherently dynamic as entities may enter or leave the system and resource utilization and availability are continuously changing [3],[4]. This environment makes efficient resource management (resMgt) a non-trivial task. Further, due to the autonomous and individual characteristics of entities in CCC, different nodes provide different Quality of Service (QoS) in resource provision [5],[6]. A node may provide low QoS because of system problems or because it is not willing to provide high QoS in order to save costs. Also, nodes may be attacked by viruses and Trojan horse programs. This weakness is revealed in all the cloud platforms. Thus, resMgt needs reputation management (repMgt) to measure resource provision QoS for guiding resource provider selection.

2 PREVIOUS METHODOLOGIES

However, though many distributed resMgt and repMgt systems for grids have been proposed previously, and cloud resource orchestration (i.e., resource provision, configuration, utilization and decommission across a distributed set of physical resources in clouds). These two issues have typically been addressed separately. Simply building and combining individual resMgt and repMgt systems in CCC will generate doubled, prohibitively high overhead. Moreover, most previous resMgt and repMgt approaches are not sufficiently efficient or effective in the large-scale and dynamic environment of CCC [7],[8].

Previous repMgt systems neglect resource heterogeneity by assigning each node one reputation value for providing all of its resources [8],[9]. For example, a person trusts a doctor for giving advice on medical issues but not on financial issues. Similarly, a node that performs well for computing services does not necessarily perform well for storage services. Thus, previous repMgt systems are not effective enough to provide correct guidance for trust-worthy individual resource selection. In task (1), RepMgt needs to rely on resMgt for reputation differentiation across multiple resources.

Previous resMgt approaches only assume a single QoS demand of users, such as efficiency or security [8],[9]. Given a number of resource providers (i.e., servers), the efficiency-oriented resMgt

policy would choose the one with the high-est available resource, while the security-oriented repMgt policy would choose the one with the highest reputation. The former may lead to a low service success rate while the latter may overload the node with many resource requests. Thus, uncoordinated deployment of repMgt and resMgt will exhibit contradictory behaviors and significantly affect the effectiveness of both, finally leading to degraded overall performance.

3 PROPOSED METHODOLOGY

In proposed system there are four components are used to perform three process namely resource location, resource selection and finally resource transaction is shown in figure 3.1. For *resource selection* component called *integrated multi-Faceted Resource or Reputation* is used which identifies the resource availability in distributed datacenter. Next process is *resource selection* is achieved by two component *multi QoS oriented resource selection* and *Dynamic Load Balancing*, here the provider are selected based on user priority and load of the selected provider. Final process is *resource transaction* is done based on *Price assisted resource control*

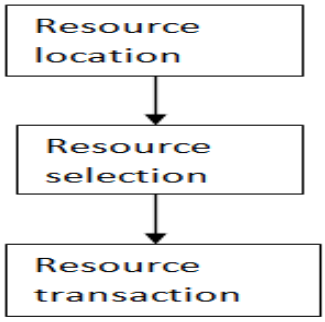

Figure 3.1 Process of Harmony

3.1Integrated Multi-Faceted Resource or Reputation Management

Harmony leverages the Cycloid hierarchically structured P2P overlay for its substrate in figure 3.2 describes the structure of Harmony. All nodes are grouped into different clusters, which are identified by $a_{d-1}a_{d-2}.....a_0$. Within a cluster, the nodes are differentiated by *k*. Cycloid assigns a key to a node whose ID is closest to the key's ID. It provides two main functions: *Insert* (*ID, object*) and *Lookup*(*ID*), to store an object in its owner node and to retrieve the object, respectively.

The resource types, such as CPU, bandwidth and memory, are globally defined and known by every node. The resource information (denoted by I_r) includes the resource owner's IP address, the resource type, the available amount, the resource physical location, the price, etc. As shown in Figure 1, Harmony proactively collects all resource information I_r of each resource type in a cluster. Thus, to search for one resource, a node only needs to probe nodes in one cluster rather than executing system-wide probing.

Vol.3, Issue 2, 2015, Page.903-909, ISSN: 2347–8586

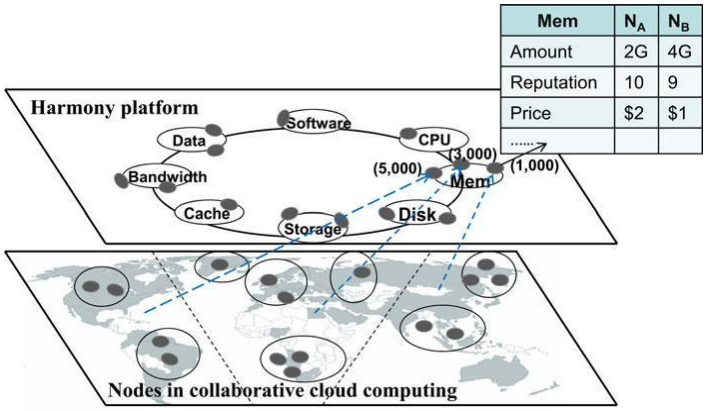

Figure 3.2 Design of Harmony

Harmony distinguishes the reputation feedback of a resource provider by resource types and maps the reputation feedback of one resource type to the corresponding resource cluster. For example, the reputation feedback of a node's "Mem" resource provision is mapped to the cluster for the "Mem" resource. As a result, for each specific resource type, Harmony pools together the resource information on available resources and the reputation of resource providers to facilitate resource searching. According to the reputation and resource information, a resource requester can select a resource provider.

To enable users to find physically close resources within a cluster, Harmony further groups the resource information of physically close nodes into one node, called a *directory node*. Directory nodes periodically collect I_r and requests, and function as matchmakers between resource requesters and providers. Harmony adopts the hash-based proximity clustering approach that relies on DHT functions to collect the information of physically close nodes together [11]. The clustering approach introduces the *Hilbert number* (*H*) that represents a node's geographical location so that the distances between nodes can be inferred from the distance between the Hilbert numbers. Physically close nodes have closer *H* values.

3.2Multi-QoS-oriented Resource Selection

After a directory node locates the providers that have the required reputation, available amount, and price, it needs to choose a provider for the requester. The final QoS offered by a provider is determined by a number of factors such as efficiency, trustworthiness, distance, security, and price. We call these factors QoS demands or attributes. A challenge is how to consider individual or combined QoS attributes with a user's desired priorities of the attributes in provider selection. Harmony solves this problem by unifying all attribute values and a client's consideration priority of different attributes to a profit metric. eBay's reputation system asks users to provide feedback on different aspects such as item description and shipping charge. Similarly, Harmony utilizes a list of QoS attributes. It requires nodes to give ratings for each QoS attribute and for overall QoS in addition to a reputation for the server. As with the reputation feedback, the QoS ratings are collected at the directory node for the provided resource of the server. The overall QoS is actually a result of the combined influence from the QoS attributes. However, it is not easy to detect how the different attributes influence the overall QoS. Harmony depends on a neural network to find out the influence weight of each attribute on the overall QoS value, and further considers users' attribute consideration priority [12].

Each directory node builds a neural network model. Its inputs are the ratings of QoS attributes such as reputation, available resource amount, distance, price and so on, and its output is the overall QoS value for a provider as a combined effect of the different attributes. The training process of the neural network is the process of determining the weight of influence of each attribute on the QoS. Suppose the attribute influence weights are $\{w_1, w_2, \cdots, w_5\}$ after training. These weights reflect the normal influence of different QoS attributes on the overall QoS.

3.3 *Dynamic Load Balancing*

The load balancing technique used to make sure that none of the node is in idle state while other nodes are being utilized. In order to balance the lode among multiple nodes you can distribute the load to another node which has lightly loaded. Thus distributing the load during runtime is known as Dynamic Load Balancing technique. Some of the goals of Load balancing are

1. To progress the performance substantially.
2. To have a backup preparation in case the system fails.
3. To maintain the system stability.
4. To hold future alteration in the system.

Before allocating the provider to the customer for service the load balancer will check the memory and the CPU usage [10]. If the load is very high is very high then the client request will be transfer to the next available node with user specified QoS. For that *Round Robin Algorithm* is used. It use random sampling method which means the main controller select the balancer randomly to allocate the load in case of some balancer is heavily loaded .When compared to other algorithm the complexity of round robin algorithm is less

3.4 Price assisted resource control

In managing decentralized resources, Harmony employs a resource trading model which is recognized as an effective way to provide incentives for nodes to provide high QoS In the model, a node pays credits to a resource provider for offered resources, and a resource provider specifies the price of its resources. The credits could be either virtual money or real money. The price is the amount of credits to use one unit of resource for one time unit. Consequently, in order to use others' resources, a node must provide its resources to others or pay real money[13],[14],[15].

4 CONCLUSION

In this paper, proposed an integrated resource or reputation management platform, called Harmony, for collaborative cloud computing. Recognizing the interdependencies between resource management and reputation management, Harmony incorporates four innovative components to enhance their mutual interactions for efficient and trustworthy resource sharing among clouds. The integrated resource/ reputation management component efficiently and effectively collects and provides information about available resources and reputations of providers for providing the types of resources. The multi-QoS-oriented resource selection component helps requesters choose resource providers that offer the highest QoS measured by the requesters' priority consideration of multiple QoS attributes. *Dynamic load balancing component* check the load of node the before allocating the provider to the client. If the load has heavy load the next resource available node is selected based on user QoS priority as provider to provide services. *The price-assisted resource control* component provides incentives for nodes to offer high QoS in providing resources.

The components collaborate to enhance the efficiency and reliability of sharing globally-scattered distributed resources in CCC. The superior performance of Harmony in comparison to previous resource and reputation management systems shows that Harmony achieves high scalability, balanced load distribution, locality-awareness, and dynamism-resilience in the large-scale and dynamic CCC environment. In our future work, investigate the optimal time period for neural network training and load factor calculation. We will investigate the challenges of deploying the Harmony system for the real-world applications which involve cooperation between cloud providers.

REFERENCE

[1] P. Suresh Kumar, P. Sateesh Kumar, and S. Ramachandram, "Recent Trust Models In Grid," J. Theoretical and Applied Information Technology, vol. 26, pp. 64-68, 2011.

[2] J. Li, B. Li, Z. Du, and L. Meng, "Cloud VO: Building a Secure Virtual Organization for Multiple Clouds Collaboration," Proc. 11th ACIS Int'l Conf. Software Eng. Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010.

[3] C. Liu, B.T. Loo, and Y. Mao, "Declarative Automated Cloud Resource Orchestration," Proc. Second ACM Symp. Cloud Computing (SOCC '11), 2011.

[4] C. Liu, Y. Mao, J.E. Van der Merwe, and M.F. Fernandez, "Cloud Resource Orchestration: A Data Centric Approach," Proc. Conf. Innovative Data Systems Research (CIDR) , 2011.

[5] K. Hwang, S. Kulkarni, and Y. Hu, "Cloud Security with Virtualized Defense and Reputation-Based Trust Management," Proc. IEEE Int'l Conf. Dependable, Autonomic and Secure Computing (DASC), 2009.

[6] IBM Red Boo. Fundamentals of Grid Computing, Technical Report REDP-3613-00 2000.

[7] L. Xiong and L. Liu, "Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004.

[8] M. Srivatsa, L. Xiong, and L. Liu, "Trustguard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks," Proc. World Wide Web Conf., 2005.

[9] R. Zhou and K. Hwang, "Power Trust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 4, pp. 460-473, 2008. R. Zhou and K. Hwang, "Gossip-Based Reputation Management for Unstructured Peer-to-Peer Networks," IEEE Trans. Knowledge and Data Eng., 2007.

[10] H. Shen and C. Xu. Hash-based proximity clustering for efficient load balancing in heterogeneous DHT networks. *JPDC*, 2008.

[11] M. Mowbray, F. Brasileiro, N. Andrade, and J. Santana, "A Reciprocation-Based Economy for Multiple Services in Peer-to-Peer Grids," Proc. IEEE Sixth Int'l Conf. Peer-to-Peer Computing

(P2P), 2006.

[12] S. Haykin, editor. *Neural Networks: A Comprehensive Foundataion*. Second Edition, Prentice-Hall Publisher, 1999.

[13] S.C.M. Lee, J.W.J. Jiang, D.-M. Chiu, and J.C.S. Lui, "Interaction of ISPs: Distributed Resource Allocation and Revenue Maximization," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 2, pp. 204-218, Feb. 2008.

[14] J. Park and M. van der Schaar, "Pricing and Incentives in Peer-to-Peer Networks," Proc. IEEE INFOCOM, 2010.

[15] X. Zhang and B. Li, "On the Market Power of Network Coding in P2P Content Distribution Systems," Proc. IEEE INFOCOM, 2009.