© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

An Efficient and Trustworthy Resource Sharing Platform for Collaborative Cloud Computing

Malvika K, Nikita T K, Aravindhan B
Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai, India

ABSTRACT

The purpose of this project is to provide information about available resources and reputations to provide services to the customers in Collaborative Cloud Computing (CCC). In this project, we address the problem of a single cloud which may not be able to provide sufficient resources effectively using a single QoS during a peak time for an application. A Collaborative Cloud Computing (CCC) called Harmony is being developed which effectively combines resource management and reputation management in a harmonious manner. Harmony platform incorporates three key components. The components are integrated multi-faceted resource/reputation management, Multi-QoS-oriented resource selection and Price-assisted resource/reputation control. Compared to the existing system, the proposed system enables a client to perform resource selection that offers the highest QoS with different priorities using Harmony platform. Moreover Harmony enables a node to locate its desired resources and also find the reputation of the located resources. The existing system fail to exploit node reputation by always selecting the highest-reputed nodes in resource selection to fairly utilize resources in the system to meet different users QoS demands. The proposed system exploits node reputation by always selecting the highest-reputed nodes and priority consideration of multiple QoS in resource selection to utilize resources.

Keywords: Distribution systems, Reputation management, Resource management, Distributed hash tables, Cloud computing.

I. INTRODUCTION

Cloud computing has become a popular computing paradigm, in which cloud providers offer scalable resources over the Internet to customers. The demand for scalable resources in some applications has been increasing very rapidly. A single cloud may not be able to provide sufficient resources effectively using a single QoS during a peak time. Thus, advancements in cloud computing are inevitably leading to a promising future for collaborative cloud computing (CCC), where globally-scattered distributed cloud resources belonging to different organizations or individuals are collectively pooled and used in a cooperative manner to provide services.

We propose a Collaborative Cloud Computing (CCC) called Harmony, which integrates resource management and reputation management in a harmonious manner. A CCC platform interconnects physical resources to enable resource sharing between clouds and provides a virtual view of a tremendous amount of resources to customers. This virtual organization is transparent to cloud customers. In addition, Harmony can deal with the challenges of large scale and dynamism in the complex environment of CCC.

Our Contributions– Summarized as below:

1) Preliminary study on real trace and experimental results:
   We analyzed the transaction and feedback rating data we collected from an online trading platform. We found that some sellers have high QoS in providing some merchandise but offer low QoS in others, and buyers tend to buy merchandise from high-reputed sellers.
2) Integrated multi-faceted resource/reputation management:
   Harmony offers multi-faceted reputation evaluation across multiple resources by indexing the resource information and the reputation of each type of resource to the same directory node.

3) Multi-QoS-oriented resource selection: Unlike previous resMgt approaches that assume a single QoS demand of users, Harmony enables a client to perform resource selection with joint consideration of diverse QoS requirements, such as reputation, efficiency, distance, and price, with different priorities.
4) Price-assisted resource/reputation control: In a resource *t*ransaction, a resource requester pays a resource provider for its resource. The transactions are conducted in a distributed manner in Harmony.

II. METHODS AND MATERIAL

A. Related Work

The already proposed methods can be classified into:

a) Integrated multi-faceted resource/reputation management
b) Multi-QoS-oriented resource selection and price-assisted control.

Integrated Multi-faceted Resource/Reputation Management

Relying on a distributed hash table overlay (DHT), Harmony offers multi-faceted reputation evaluation across multiple resources by indexing the resource information and the reputation of each type of resource to the same directory node. In this way, it enables nodes to simultaneously access the information and reputation of available individual resources. Zol is a top online trading platform in China similar to Amazon and eBay. Zol is chosen for market data analysis because neither Amazon nor eBay provides the historical rating record of each transaction. Zol provides the historical reputation record of each transaction, which enables to calculate the reputation for each type of a seller's merchandise for the multi-faceted reputation study. We collected trace data including 1,562,548 transaction records from Zol covering the period from 9/20/2006 to 6/26/2010. In addition to the overall reputation values of sellers, Zol provides the ratings within [0,100] for five QoS attributes for each transaction: 1) price, 2) distance, 3) quality, 4) service, and 5) efficiency. Thus, if the resource a node possesses is limited, the highest-reputed nodes can easily become overloaded. Therefore, a seller's individual reputation cannot reflect its QoS for each QoS attribute, which confirms the need to consider multiple QoS attributes in selecting resources.

Multi-QoS-oriented Resource Selection and Price-assisted Control

A multi-QoS-oriented resource selection algorithm that enables clients to choose resources based on their priority considerations of the different QoS attributes in a resource transaction, a resource requester pays a resource provider for its resource. The transactions are conducted in a distributed manner in Harmony. Harmony employs a trading model for resource transactions in resource sharing and leverages the resource price to control each node's resource use and reputation. It enables each node to adaptively adjust its resource price to maximize its profit and maintain a high reputation while avoiding being overloaded, in order to fully and fairly utilize resources in the system.

Simply combining multi-resMgt and repMgt will lead to a few problems. First, resMgt and repMgt always have their own infrastructures. Second, most resMgt approaches are driven by either efficiency or security through choosing the highest-reputed or highest-capacity node. Hence, a direct combination will lead to contradictory behaviors. Third, since a node refers to the overall reputation in selecting individual resources, it may receive incorrect guidance because a high overall-reputed node may provide low QoS for individual resources.

The simulation assumed three types of nodes: altruistic, neutral and egotistic, each of which provides its service successfully with a probability 1, 0.5, and 0.1, respectively. Each node is randomly assigned to one of the three types, and every request has 10 servers that are able to satisfy the request. The utilization of a node is defined as the ratio of its load to its capacity. The maximum utilization value of each node during the experiment was recorded. The 99.9th percentile value in this group of maximum utilization values is referred to as the 99.9th percentile maximum utilization. A service failure occurs when the selected node is unwilling to provide a service. It shows that the efficiency-oriented policy, MaxCap, performs best in controlling node utilization but incurs a high service failure rate; in contrast, the trust-oriented policy, MaxTrust, performs the best in controlling service failure rate but leads to high node utilization. The fundamental reason for these severe problems is the neglect of the interdependencies between repMgt and resMgt; the uncoordinated deployment of either one will exhibit contradictory behaviors and significantly reduce the effectiveness of both. Therefore, a method is needed to jointly consider efficiency and trust in resource selection, and also need a method to avoid overloading nodes with resource requests.

B. Proposed Methodology

We propose a CCC platform, called Harmony, which integrates resource management and reputation management in a harmonious manner. Harmony incorporates three key innovations: integrated multi-faceted resource/reputation management, multi-QoS-oriented resource selection, and price-assisted resource/reputation control. The trace data we collected from an online trading platform implies the importance of multi-faceted reputation and the drawbacks of highest-reputed node selection. Simulations and trace-driven experiments on the real-world Planet Lab test bed show that Harmony outperforms existing resource management and reputation management systems in terms of QoS, efficiency and effectiveness.
The advantages are as follows:

- Here all the clouds resources will be utilized.
- User can get the more efficient and trusted output.
- We can improve the number of the user.
- We can improve the QoS.

i) Integrated Multi-Faceted Resource/Reputation Management

In this mechanism, nodes update their neighbors periodically and transfer resource and reputation information based on the DHT key assignment policy when joining or leaving. Also, before a node departs from the system or after a node joins in the system, it notifies the directory nodes that store *Ir* of its resources. Thus, a node's *Ir* is always stored in its directory nodes even in dynamism, and the *Lookup* (*ID*) requests can always be forwarded to the directory nodes. In a very large-scale distributed system, directory nodes may become bottlenecks. Our experimental results show that Harmony achieves a more balanced load distribution among directory nodes than the previous DHT-based resource management systems due to its two-layer hierarchical resource information distribution. When a directory node is overloaded, it can use the load balancing algorithm introduced to move its load to a lightly loaded node and maintain an index to the node. Harmony's resource/reputation management also features three characteristics: (1) locality-awareness and dynamism-resilience; (2) low maintenance overhead by relying on a single DHT for both resource and reputation management; and (3) multi-resource management using one DHT.

ii) Multi-Qos-Oriented Resource Selection

After a directory node locates the resource providers that have the required reputation, available amount, and price, it needs to choose provider(s) 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). When choosing from a number of providers, most previous approaches rigidly consider a single QoS demand at a time. However, different tasks have different requirements. For time-critical tasks, distance should be given priority. For a large computing task, efficiency should be the main deciding factor. Further, a server's distances to different clients are different. This means a server's final QoS for client *i* does not necessarily represent its QoS for client *j*. Also, a task or user may have multiple demands with different priorities. The output of the neural network is the overall QoS. Finally, the directory node determines the server(s) with the highest overall QoS value. Thus, Harmony jointly takes into account the client's considered priority on different attributes and the influence weights of attributes on the final rating in server selection.

iii) Price-Assisted Resource/Reputation Control

The price-assisted resource/reputation control scheme prevents uncooperative behaviors, encourages nodes to provide high QoS, and allows nodes to adaptively adjust their load to offer 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. As a result, all resources in the system are fully and fairly utilized, nodes are not overloaded, and a node's reputation can truly reflect its QoS in offering resources without the influence of the

overloaded status. To address the problem of low reputation for newly joined nodes, Harmony assigns the nodes a certain amount of starting virtual credits that can be used for building initial reputation.

III. RESULTS AND DISCUSSION

Performance Evaluation on PlanetLab

In order to show the importance of integrating resource management and reputation management, we first evaluated Harmony in comparison with Harmony without reputation management (denoted by resMgt) and the PowerTrust reputation management system. To make the methods comparable, we use Harmony's structure and resource discovery algorithms for PowerTrust. These methods are only different in resource selection. After locating resource providers, Harmony chooses a lightly loaded provider with the highest individual reputation, PowerTrust chooses the provider with the highest overall reputation, and resMgt randomly chooses a provider among lightly loaded nodes.

We compared Harmony with a Single-DHT method and a Multi-DHT method in order to show Harmony's higher efficiency in the complex environment of large-scale and dynamic CCC. Single-DHT relies on a single DHT, in which a resource ID owner is the directory node for all information of this resource. Multi-DHT relies on multiple DHTs, in which one DHT is responsible for one resource type, and resource information is distributed among all DHT nodes based on their resource values.

A. Integrated Multi-Faceted Res/Rep Management

Resource management needs reputation management to providea cooperative environment for resource sharing. Otherwise, a node cannot know which resources are trustworthy. Resource management in turn facilitates reputation management to evaluate multi-faceted node reputation in providing different resources. Therefore, resMgt may choose nodes unwilling to provide services and generate many service failures. PowerTrust may choose overloaded nodes that cannot process requests successfully. By integrating both resource management and reputation management, Harmony enables joint consideration of node reputation and load status and chooses lightly loaded high-reputed nodes that can always process requests successfully.

Trustworthy Resource Sharing

In order to see the effect of reputation management alone, we assumed that nodes do not drop requests when overloaded but queue the requests for processing later on. A request is successfully resolved if the selected server has provided its requested resource. We see that Harmony achieves a success rate of over 96%, while PowerTrust achieves around 73% and resMgt achieves around 44%. ResMgt selects a resource without considering reputation and may choose a resource provider with low reputation for the requested resource, leading to a low success rate. PowerTrust always selects the highest overall- reputed provider. As verified by the trace, a node with a high overall reputation may provide low QoS for another resource due to either unwillingness or overloaded status. Therefore, Harmony significantly outperforms PowerTrust by always selecting the supplier with the highest individual, rather than overall, reputation. It shows the average individual reputation of every group of 500 selected resource providers for the requested resources, which follows Harmony>PowerTrust>resMgt. The experimental results confirm the effectiveness of multi-faceted reputation management and its importance in guiding trustworthy resource selection. The experimental result also follows Harmony>PowerTrust>resMgt, which is consistent with the success rate result in Figure 9(c). Also, the result of each method is lower. This is caused by the same reasons for the differences between which confirm the importance of considering individual reputation in selecting resource providers, especially for multi-resource requests.

Efficient Resource Sharing

Without resource management, reputation management cannot tell if a resource supplier has sufficient available resources. We then tested the importance of resource management to reputation management. In this experiment, each node maintains a waiting queue; an overloaded resource provider inserts its received request into its waiting queue. The requests staying in the queue for more than 100s are dropped. Therefore, a resource provision failure is caused by either a provider's overloaded status or its unwillingness to provide a service as reflected by its reputation. The Pareto distribution reflects the real world in which the amounts of available resources vary by different orders of magnitude [29, 37–39]. Thus, we used a Pareto distribution in determining a node's capacity for a resource type, a request's requested amount, and time period. We set the shape parameter to 2, and the scale parameters to 100, 40, and 200 for the above three parameters, respectively. We arbitrarily set the values in their reasonable ranges. Different setting values will not change the relative performance differences of the methods. However, resMgt has a higher failure rate due to provider unwillingness than PowerTrust because resMgt only considers node load but neglects reputation. Only Harmony can constrain failures due to either cause, as it jointly considers both load and reputation.

B. Multi-QoS-oriented Resource Selection

We then evaluate the effectiveness of the multi QoS-oriented resource selection scheme. We used 95×12 transaction records for 95 sellers, each with 12 types of merchandise. Each transaction record has 52 transactions. We used 40×12 for training and the remaining 55 × 12 for testing. We regard a node's individual reputation (rating on its transactions for a type of merchandise) as its overall QoS for the merchandise (i.e., resource) and regard its overall reputation as its reputation in Harmony. The inputs of the neural network model include the QoS attributes in each transaction (i.e., price, distance, service, quality and efficiency) and the seller's overall reputation. The output of the model is the seller's overall QoS. Because the real trace does not have users' consideration priorities, we assume that the six QoS attributes have equal priorities. It shows the predicted overall QoS and the real overall QoS for 100 resource requests, both of which almost overlap. Their root mean square error equals 0.95, a very small value. The results show the effectiveness and accuracy of the neural network model in predicting the QoS in individual resource selection.

C. Effect of Queuing Timeout

In order to study the effect of the timeout for dropping requests on the success rate and failure rate, we measured the two metrics with varying timeout values for 5000 requests. It shows the success rates of different methods with different timeout values. We see that Harmony and resMgt generate no delayed requests since they choose lightly loaded nodes, while PowerTrust generates many delayed successful requests since it does not consider node load. We observe that the timeout value does not affect the success rate of Harmony and resMgt. Also, as the timeout value increases, the success rate of PowerTrust for delayed successful requests increases. In conclusion, Harmony and resMgt are not affected by the timeout, while PowerTrust is sensitive to the timeout and a large timeout enables it to resolve more requests. Moreover, Harmony always achieves a higher success rate than the other methods since it considers both reputation and load status.

D. Price-assisted Resource/Reputation Control

We then test the performance on a system with and without the price-assisted resource/reputation control algorithm denoted by *w/Price* and *w/oPrice*, respectively, in a heavy load situation. We randomly chose nodes from the system to generate requests. The request generating rate follows a Poisson process at a rate of 2 requests per second. We set every request to use 40 units of a resource for 200s. We chose these values so that some resource providers have insufficient available resources when requested. We set the price range to [10, 20] credits. All nodes have the same capacity of 200 and reputation of 9. During the simulation period, no resource information is updated in the directory nodes. To choose a resource provider, a requester first identifies the providers with reputation > 8, then identifies the providers with the lowest price, and finally randomly chooses one. Nodes with reputation > 8 have probability 1 to offer the service. A resource request fails only when its provider is overloaded. If a requester receives a successful service, the resource provider's reputation is increased by 0.001. Otherwise, its reputation is decreased by 0.02. In every 10s, each node checks its load; if its load factor $f > 0.8$, it reduces its price by 1; otherwise, it increases its price by 1. These results imply that *w/Price* distributes request load among servers more evenly than *w/oPrice*. *w/Price* makes full use of all available resources while constraining node utilization within 100%; while in *w/oPrice*, some nodes are overloaded or underloaded. The experimental results further verify the effectiveness of the price-based control algorithm in fully utilizing available resources and preventing overloads.

IV. CONCLUSION

In this paper, we propose an integrated resource/reputation management platform, called Harmony, for collaborative cloud computing (CCC).Recognizing the inter dependencies between resource management and reputation management, Harmony incorporates three 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. The price-assisted resource/reputation control component provides incentives for nodes to offer high QoS in providing resources. Also, it helps providers keep their high reputations and avoid being overloaded while maximizing incomes. The components collaborate to enhance the efficiency and reliability of sharing globally-scattered distributed resources in CCC. Simulations and trace-driven experiments on PlanetLab verify the effectiveness of the different Harmony components and the superior performance of Harmony in comparison to previous resource and reputation management systems. The experimental results also show 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, we will 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.

V. REFERENCES

[1] P. Suresh Kumar, P. Sateesh Kumar, and S. Ramachandram. Recent Trust Models In Grid. JATIT, 2011.

[2] J. Li, B. Li, Z. Du, and L. Meng. CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration. In Proc. of SNPD, 2010.

[3] C. Liu, B. T. Loo, and Y. Mao. Declarative Automated Cloud Resource Orchestration. In Proc. of SOCC, 2011.

[4] C. Liu, Y. Mao, J. E. Van der Merwe, and M. F. Fernandez. Cloud Resource Orchestration: A DataCentric Approach. In Proc. of CIDR, 2011.

[5] K. Hwang, S. Kulkarni, and Y. Hu. Cloud Security with Virtualized Defense and Reputation-based Trust Management. In Proc. of DASC, 2009.

[6] H. Shen and G. Liu. Harmony: Integrated resource and reputation management for large-scale distributed systems. In Proc. Of ICCCN, 2011.

[7] H. Shen, Y. Zhu, and W. Li. Efficient and Locality-aware Resource Management in Wide-area Distributed Systems. In Proc. of NAS, 2008.

[8] A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting Scalable Multi-Attribute Range Queries. In Proc. of SIGCOMM, 2004. D. Talia, P. Trunfio, J. Zeng, and M. H¨ogqvist. A DHT-based Peerto- Peer Framework for Resource Discovery in Grids. Technical Report TR-0048, CoreGRID - Network of Excellence, 2006.

[9] H. Shen, C. Xu, and G. Chen. Cycloid: A Scalable Constant-Degree P2P Overlay Network. Performance Evaluation, 63(3):195–216, 2006.

[10] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In Proc. Of STOC, pages 654–663, 1997.

[11] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P systems scalable. In Proc. of SIGCOMM, 2003. [32] M. Mowbray, F. Brasileiro, N. Andrade, and J. Santana. A Reciprocation-Based Economy for Multiple Services in Peer-to-Peer Grids. In Proc. of P2P, 2006.

[12] 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. TPDS, 19(2):204–218, 2008. [37] N. Bansal and M. Harchol-Balter. Analysis of srpt scheduling: investigating unfairness. In SIGMETRICS/Performance, 2001.

[13] X. Zhang, Y. Qu, and L. Xiao. Improving distributed workload performance by sharing both cpu and memory resources. In Proc. Of ICDCS, 2000.

[14] R. Subrata and A. Y. Zomaya. Game-theoretic approach for load balancing in computational grids. TPDS, 2008. I. Konstantinou, D. Tsoumakos, and N. Koziris. Fast and Cost-Effective Online Load-Balancing in Distributed Range-Queriable Systems. TPDS, 2011.

[15] H. Han, Y. C. Lee, W. Shin, H. Jung, H. Y. Yeom, and A. Y. Zomaya. Cashing in on the Cache in the Cloud. TPDS, 2012.