ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 15, March 2016

An Efficient Cross Cloud Composition Plan for Big Data Applications Using HIRESOME-II

Ms. Arunthathi.M. Ms. Shanmuga priya M.E. M.B.A,

Pg Scholar, Computer Science and Engineering Department, Anna University Regional Campus, Keelakuilkudi, Madurai-625019 INDIA; E-Mail:arunthathimb@gmail.com).

Professor, Computer Science and Engineering Department, Anna University Regional Campus, Keelakuilkudi, Madurai-625019 INDIA; E-Mail:vijaylaya2000@gmail.com).

Abstract: In recent years, Cloud Computing and big data receives enormous attention internationally due to various business-driven promises and expectations such as lower upfront IT costs, a faster time to market, and opportunities for creating value-add business. Cloud computing promises a scalable infrastructure for processing big data applications such as medical data analysis. Cross-cloud service composition provides a concrete approach capable for large-scale big data processing. However, the complexity of potential compositions of cloud services calls for new composition and aggregation methods, especially when some private clouds refuse to disclose all details of their service transaction records due to business privacy concerns in cross-cloud scenarios. Moreover, the credibility of cross-clouds and on-line service compositions will become suspicion, if a cloud fails to deliver its services according to it ''promised'' quality. In view of these challenges, we propose a privacy-aware cross-cloud service composition method, named HireSome-II (History record-based Service optimization method) based on its previous basic version HireSome-I. In our method, to enhance the credibility of a composition plan, the evaluation of a service is promoted by some of its QoS history records, rather than its advertised QoS values. Besides, the k-means algorithm is introduced into our method as a data filtering tool to select representative history records. As a result, Hire Some-II can protect cloud privacy, as a cloud is not required to unveil all its transaction records. Furthermore, it significantly reduces the time complexity of developing a cross-cloud service composition plan as only representative ones are recruited, which is demanded for big data processing. Simulation and analytical results demonstrate the validity of our method compared to a benchmark.

Keywords-HIRESOME-II; K-means; Cross-cloud.

I INTRODUCTION

In practice, to satisfy different security and privacy requirements, cloud environments usually consist of public clouds, private clouds and hybrid clouds, which lead a rich ecosystem in big data applications. Generally current implementations of public clouds mainly focus on providing easily scaled-up and scaled-down computing power and storage. If data centers or domain specific services center tend to avoid or delay migrations of themselves to the public cloud due to multiple hurdles, from risks and costs to security issues and service level expectations, they often provide their services in the form of private cloud or local service host . For a complex web-based application, it probably covers some public clouds, private clouds or some local service host. For instance, the healthcare cloud service, a big data application illustrated in, involves many participants like governments, hospitals, pharmaceutical research centers and end users. As a result, a healthcare application often covers a series of services respectively derived from public cloud, private cloud and local host. In practice, some big data centers or software services cannot be migrated into a public cloud due to some security and privacy issues. If a web-based application covers some public cloud services, private cloud services and local web services in a hybrid way, cross-cloud collaboration is an ambition for promoting complex web based applications in the form of dynamic alliance for value-add applications. It needs a unique distributed computing model in a network-aware business context. Cross-cloud service composition provides a concrete approach capable for large-scale big data processing. Existing (global) analysis techniques for service composition, however, often mandate every participant service provider to unveil the details of services for network-aware service composition, especially the QoS information of the services. Unfortunately, such an analysis is infeasible when a private cloud or a local host refuses to disclose all its service in detail for privacy or business reasons .In such a scenario, it is a challenge to integrate services from a private cloud or local host with public cloud services such as Amazon EC2 and SQS for building scalable and secure systems in the form of smashups. As the diversity of Cloud services is highly available today, the complexity of potential cross-cloud compositions requires new composition and aggregation models. On the other hand, as a cloud often hosts a lot of individual services, cross-cloud and on-line service composition is heavily time-consuming for big data applications. It always challenges the efficiency of service composition development on Internet. Besides, for a web service which is not a cloud service and its bandwidth probably fails to match to the cloud, it is a challenge to trade off the bandwidth between the web service and the cloud in a scaled-up or scaled-down way for a cross-cloud composition application. Here, the time cost is heavy for cross-platform service composition. Wit these observations, it is a challenge to tradeoff the privacy and the time cost in cross-cloud service composition for processing big data applications. In view of this challenge, an enhanced History record-based Service optimization method named HireSome-II is presented in this paper for privacy-aware cross-cloud service composition for big data applications. In our previous work, a similar method, named HireSome (could be treated as HireSome-I) has been investigated, which aims at enhancing the credibility of service composition. HireSome-I is incapable of dealing with the privacy issue in cross-cloud service composition. Compared to HireSome-I, HireSome-II greatly speeds up the process of selecting a (near-to-) optimal service composition plan, and protects the privacy of a cloud service for cross-cloud service composition. The remainder of the paper is organized as follows. Section 2 presents some preliminary knowledge. Section 3 investigates a benchmark for evaluating a history record based service composition. Section 4 elaborates HireSome-II for cross-cloud service composition. Section 5 demonstrates comprehensive simulation experiments to evaluate the efficiency and effectiveness of our method. Section 6 presents conclusions. Finally, Section 7 addresses our references.

II RELATED WORK

In this section, we briefly introduce some preliminary knowledge about objective function and utility function leveraged to guide the selection of service composition.

A.DECENTRALIZED SELF-ADAPTATION MECHANISM

This thesis presents a Cloud-based-Multi-Agent System (Clubman) that uses multiple double auctions, to enable applications to self-adapt, based on their QoS requirements and budgetary constraints. We design a marketplace that allows applications to select services, in a decentralized manner.

B.SECURE ERASURE CODE-BASED CLOUD STORAGE SYSTEM

Cloud storage is a service model in which data in maintained, managed and backed up remotely and made available to user over a network. Having your data stored offsite in the cloud makes it accessible from anywhere without the hassle of maintaining your own local storage and file-serving systems.

C.CLOUD COMPUTING AND EMERGING ITS PLATFORMS

With the significant advances in Information and Communications Technology (ICT) over the last half century, there is an increasingly perceived vision that computing will one day be the 5th utility (after water, electricity, gas, and telephony).

D.BERKELEY VIEW OF CLOUD COMPUTING

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased.

E.DECLARATIVE RECOMMENDER SYSTEM

The cloud infrastructure services landscape advances steadily leaving users in the agony of choice. Therefore, we present Cloud Recommender, a new declarative approach for selecting Cloud-based infrastructure services.

III EXISTING SYSTEM

In practice, some big data centers or software services cannot be migrated into a public cloud due to some security and privacy issues. If a web-based application covers some public cloud services, private cloud services and local web services in a hybrid way, cross-cloud collaboration is an ambition for promoting complex web based applications in the form of dynamic alliance for value-add applications. It needs a unique distributed computing model in a network-aware business context. Cross-cloud service composition provides a concrete approach capable for large-scale big data processing. Existing (global) analysis techniques for service composition,

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 15, March 2016

however, often mandate every participant service provider to unveil the details of services for network-aware service composition, especially the QoS information of the services. Unfortunately, such an analysis is infeasible when a private cloud or a local host refuses to disclose all its service in detail for privacy or business reasons. In such a scenario, it is a challenge to integrate services from a private cloud or local host with public cloud services such as Amazon EC2 and SQS for building scalable and secure systems in the form of smashups. As the diversity of Cloud services is highly available today, the complexity of potential cross-cloud compositions requires new composition and aggregation models. On the other hand, as a cloud often hosts a lot of individual services, cross-cloud and on-line service composition is heavily time-consuming for big data applications. It always challenges the efficiency of service composition development on Internet. Besides, for a web service which is not a cloud service and its bandwidth probably fails to match to the cloud, it is a challenge to trade off the bandwidth between the web service and the cloud in a scaled-up or scaled-down way for a cross-cloud composition application. Here, the time cost is heavy for cross-platform service composition.

THREAT MODEL

We consider data confidentiality for both data storage and data forwarding. In this threat model, an attacker wants to break data confidentiality of a target user. To do so, the attacker colludes with all storage servers, non target users, and up to (t _ 1) key servers. The attacker analyzes stored messages in storage servers, the secret keys of non target users, and the shared keys stored in key servers. Note that the storage servers store all re-encryption keys provided by users. The attacker may try to generate a new re-encryption key from stored re-encryption keys. We formally model this attack by the standard chosen plaintext attack1 of the proxy re-encryption scheme in a threshold version.

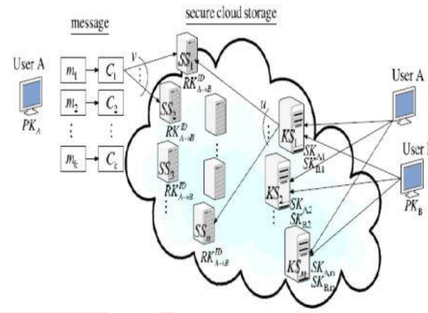

Fig 1.Threat Model

IV PROPOSED SYSTEM

We propose a privacy-aware cross-cloud service composition method, named HireSome-II (History record-based Service optimization method) based on its previous basic version HireSome-I. In our method, to enhance the credibility of a composition plan, the evaluation of a service is promoted by some of its QoS history records, rather than its advertised QoS values. Besides, the k-means algorithm is introduced into our method as a data filtering tool to select representative history records. As a result, HireSome-II can protect cloud privacy, as a cloud is not required to unveil all its transaction records. Furthermore, it significantly reduces the time complexity of developing a cross-cloud service composition plan as only representative ones are recruited, which is demanded for big data processing. Simulation and analytical results demonstrate the validity of our method compared to a benchmark.

ADVANTAGES

1. It can effectively promote cross cloud service composition in the situation where a cloud refuses to disclose all details of its service transaction records

2. Our method significantly reduces the time complexity as only some representative history records are recruited, which is highly demanded for big data applications.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 15, March 2016

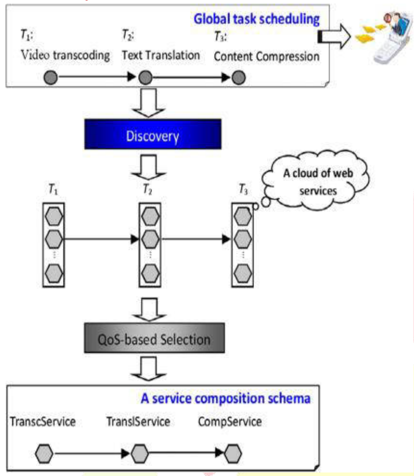

Fig 2.Service composition scheme

HIRESOME-II: TOWARD PRIVACY-AWARE CROSS-CLOUD SERVICE COMPOSITION

In our method, a tree structure is recruited to specify the service composition context. Concretely, a Task-Service Tree is defined (See Definition 4), to incorporate a task and a group of candidate services into an integrated applicationcontext. Here, the candidate services are the qualified services that can fulfill the task execution's specification infunctional and non-functional properties.

Definition 1 (Task-Service Tree). For a task and the candidate services that can fulfill the task execution's specificationin functional and non-functional properties, a Task-Service tree is a two-level tree structure that consists of a main rootnode and a group of leaf nodes, where the main root node is instantiated by the task and the leaf nodes are instantiated by the candidate services. For example, Fig. 3 illustrates two Task-Service trees respectively initiated by task T1 and task T2, in which, T1 _ WS1 and T1 _ WS2 are the candidate services that can fulfill T1's execution's specification in functional and non-functional properties, and T2 _ WS1, T2 _ WS2 and T2 _ WS3 are the candidate services that can fulfill T2's execution's specification in functional and non-functional properties. In a Task-Service tree, for a candidate service, the QoS's history records associated with its non-functional properties reflected in its past executions will be divided into two clusters by taking advantage of the well-known k-means clustering algorithm introduced in Section 2. Here, the k-means clustering algorithm is put into practice with k ¼ 2 in our method. With these processes, two peer clusters and their representative history records can be selected respectively from these two clusters.

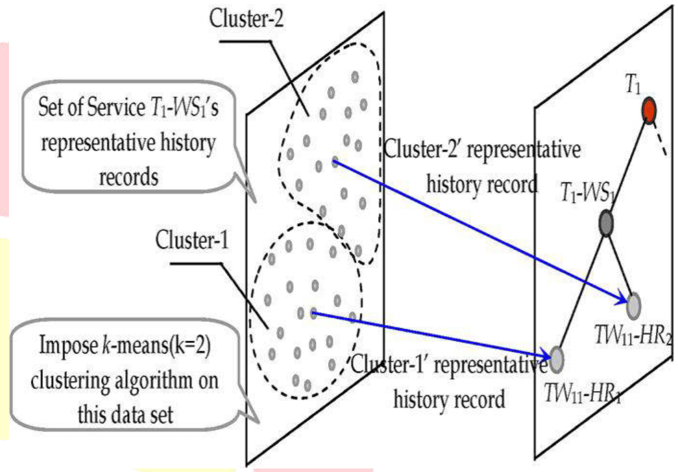

Fig 3.An Instance of Task's tree evaluation

Definition 2 (Peer Cluster and Representative History Record). For a candidate service of a task,

Its history records will be grouped into two clusters by taking advantage of the well-knownk-means eke¼ 2Þ clustering algorithm. These twoclusters will be treated as peer clusters for each other. For these two peer clusters, two representative history records will be selected respectively from these two clusters. Concretely speaking, for a peer cluster, its representative history record is a history record that owns the best utility value in this cluster. Please note that in Definition 5, two representative history records respectively belong to different clusters produced by introducing k-means¼ 2Þ clustering algorithm into Ti _ West's history records' classification. According to k-means clustering algorithm's properties, these two clusters do not overlap. Using the representative history records, a Task-Service tree can evolve into a Task- Service-History Record tree as defined by.

Definition 3 (Task-Service-History Record Tree).ATask- Service-History Record tree is evolved from a Task-Service tree by adding two leaf nodes for each candidate service, where the new leaf nodes are instanced by representative history records of the candidate services. With these scenarios as specified can be evolved into their hierarchical relationship among a task, the task's candidate services, and the candidate services' representative history records. The 1st history record layer consists of all candidate services' representative history records. Compared demonstrates a three-level Task-Service-History Record tree evolved from Fig. 3, by integrating each candidate service's representative history records into the tree structure demonstrated by Fig. 3. In the three-level tree structure, the representative history records of a candidate services are recruited as leaf nodes of the candidates, which makes up of a new level indicated by the 1st history record layer. In Fig. 3, a candidate service and its two representative history records make up of a local binary tree. Algorithm 1 specifies the developing process of the tree evolvement

V IMPLEMENTATION

A.SIMULATING CLOUD FOR BIGDATA

Simulation-based approaches to performance testing and benchmarking offer significant advantages. For example, multiple big data application developers and researchers can perform tests in a controllable and repeatable manner. In addition, it's easier to find performance bottlenecks in a simulated environment than in a real-world test-bed. Simulation-based approaches also simplify experimenting with various hardware resource and big data processing framework configurations and collecting insights about the impact of each design choice on the performance guarantees (service-level agreements). They also let developers and researchers share their simulation datasets and environment setups, leading to better validation of hypothesis and reproducibility of results. Finally, using these approaches, developers and researchers can instantiate multiple big data processing frameworks and diverse workload scenarios. The execution of simulation applications can be computationally demanding requiring special computing resources, such as clusters, grids or clouds. However, it is not only the execution of the simulation that raises technical challenges. Simulation programs typically generate large amount of data that needs to be processed and analyzed.

B.SERVICE HISTORY RECORDS ESTABLISHMENT

QoS denotes the levels of performance, reliability and availability offered by an application and by the platform or infrastructure that hosts it. QoS is fundamental for cloud users, who expect providers to deliver the advertised quality characteristics, and for cloud providers, who need to find the right tradeoffs between QoS levels and operational costs. Any violation of service level agreement (SLA) entails a loss for both cloud providers and cloud users.

C.SERVICE COMPOSITION CLASS, INSTANCE, BENCHMARK

QoS is the key information for distinguishing a service from its non-functional properties. For a group of candidate services that own same functional capabilities, QoS plays a key role in service selection and service composition among these candidate services. In practice, price, duration availability, reliability, reputation, et al.,are typical QoS specification in service delivery.

D.PRIVACY-AWARENESS ESTABLISHMENT

Cloud service composition provides a concrete approach capable for large-scale big data processing. Existing (global) analysis techniques for service composition, however, often mandate every participant service provider to unveil the details of services for network-aware service composition, especially the QoS information of the services. Unfortunately, such an analysis is infeasible when a private cloud or a local host refuses to disclose all its service in detail for privacy or business reasons.

E.PERFORMANCE METRICS ENHANCEMENT

Metrics play an important role, when it comes to measuring a process from various dimensions, assessing it and target improvements. A process can lead to a quality outcome within least time and cost only if the process itself is effective and efficient. The objective of this metric is to track and compare testing progress with the plan, hence enabling teams take early actions upon any indications that the testing activity is lagging behind. When under pressurizing schedules, testing activity is the one, which generally gets impacted the most. With a formal testing progress metric deployed, it is much harder for the team to disregard the problem.

F.TASK-SERVICE TREE'S EVOLUTION

Task Tree is a process attached to an input application. It holds the predefined workflow for a given input. The workflow consists of nodes and connectors. For the service item you have created, create a Task Tree. The two major components of this task are the estimation of the evolutionary tree (branching order), then using the estimated trees as analytical framework for further evolutionary study and finally performing the traditional role of systematic and classification. Using this study a number of interesting facts can be discovered, for example, who are the closest living relatives of humans, who are whales related to, etc.

VI.CONCLUSION

An enhanced History record-based Service optimization method, named HireSome-II based on the previous basic one of HireSome-I, has been developed for privacy-aware cross-cloud service composition for processing big data applications. It can effectively promote cross-cloud service composition in the situation where a cloud refuses to disclose all details of its service transaction records for business privacy issues in cross-cloud scenario. Our composition evaluation approach achieves two advantages. Firstly, our method significantly reduces the time complexity as only some representative history records are recruited, which is highly demanded for big data applications. Secondly, our method protects cloud privacy as a cloud is not required to unveil all of its transaction records, which accordingly protects privacy in big data. Simulation and analytical results have demonstrated the validity of our method compared to a benchmark. Implementation includes proper training to end-users. The implemented software should be maintained for prolonged running of the software. Initially the system was run parallel with manual system. The system has been tested with data and has proved to be error-free and user-friendly. Training was given to end -user about the software and its features.

VII. REFERENCES

M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz,A. Konwinski, L. Gee, D.A. Patterson, A. Rabkin, I. Stoica, andM. Zaharia, ‘‘Above the Clouds: A Berkeley View of CloudComputing,Communications of theACM,’’ Commun.ACM, vol. 53,no. 4, pp. 50-58, Apr. 2010.

R.Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic,‘‘Cloud Computing and Emerging IT Platforms: Vision, Hype,and Reality for Delivering Computing as the 5th Utility,’’ FutureGener. Comput. Syst., vol. 25, no. 6, pp. 599-616, June 2009.

J. Cao, K. Hwang, K. Li, andA.Y. Zomaya, ‘‘OptimalMultiserverConfiguration for Profit Maximization in Cloud Computing,’’IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1087-1096,June 2013.

A.Gambi and G. Toffetti, ‘‘Modeling Cloud Performance with Kriging,’’ in Proc. 34th ICSE, June 2012, pp. 1439-1440.

A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer,and D. Epema, ‘‘Performance Analysis of Cloud ComputingServices for Many-Tasks Scientific Computing,’’ IEEE Trans.Parallel Distrib. Syst., vol. 22, no. 6, pp. 931-945, June 2011.

H.Y. Lin and W.G. Tzeng, ‘‘A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding,’’ IEEE Trans.ParallelDistrib. Syst., vol. 23, no. 6, pp. 995-1003, June 2012.

M. Menzel and R. Ranjan, ‘‘CloudGenius: Decision Support forWeb Server Cloud Migration,’’ in Proc. 21st Int’l Conf. WWW,Apr. 2012, pp. 979-988.

V. Nallur and R. Bahsoon, ‘‘A Decentralized Self-Adaptation Mechanism for Service-Based Applications in the Cloud,’’ IEEE Trans. Softw. Eng., vol. 39, no. 5, pp. 591-612, May 2013.

S. Tai, J. Nimis, A. Lenk, and M. Klems, ‘‘Cloud Service Engineering,’’ in Proc. 32nd ACM/IEEE ICSE, May 2010, vol. 2, pp. 475-476.

M. Zhang, R. Ranjan, A. Haller, S. Nepal, and M. Menzel, ‘‘ADeclarative Recommender System for Cloud InfrastructureServices Selection,’’ in Proc. 9th Int. Conf. GECON, Nov. 2012,pp. 102-113.