ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

MULTICLOUD MULTI-ACCESS SMART CARD

[1] M.Sujatha [2] E.Evangelin [3]M.Jothi [4]M. Amsa
[1]Assistant professor, Dept of CSE, Kings Engineering College,
[2, 3,4] UG Students ,Dept of CSE, Kings Engineering College.

Abstract—Cloud computing promises a scalable infrastructure for 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 its ''promised'' quality. In view of these challenges, we propose a multi cloud Multi access smart card is a RFID card where we access two cloud Server and a big data DB. We integrate Ration, Hospital and passport application for user access .user personal authentication includes username, password, Primary key and RFID card. All these data are stored and verified in mongo DB. The data is splitted and stored in two cloud server sequentially. User request is handled by first cloud and the application are handled by second cloud.

I. INTRODUCTION

services 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 [1], [2], [3]. As the latest computing paradigm, cloud is characterized by delivering hardware and software resources as virtualized services by which users are free from the burden of acquiring the low level system administration details [5]. Cloud computing promises a scalable infrastructure for processing big data applications such as the analysis of huge amount of Medical data . Currently, Cloud providers including Amazon Web Services (AWS), Salesforce.com, or Google App Engine, give users the options to deploy their application over a network of resource pool [6]. By leveraging Cloud services to host Web, big data applications can benefit from cloud advantages such as elasticity, pay-per-use, and abundance of resources with practically no capital investment and modest operating cost proportional to actual use [1], [8],[9], [10].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 [11], [12], [13]. 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 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 [2]. For a complex web-based application, it probably covers some public clouds, private clouds or some local service host [10], [14].For instance, the healthcare cloud service, a big data application illustrated in [14], involves many participants like governments, hospitals, pharmaceutical research centres 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 [14], [15]. 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

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

[16]. 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 [17]. 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 mashups. 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 [18], [19], [20], [21],[22]. 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. With these observations, it is a challenge to trade off 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 Hire Some-II, is presented in this paper for privacy-aware cross-cloud service composition for big data applications. It 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.

1. Taking Advantage of the Preliminary Knowledge of utility function and objective function, each service composition instance' utility value can be calculated.
2. For Service composition class CS, an average value of it service composition instances' utility value can be computed, which will be treated a CS's utility values
3. The Service composition class that own the largest utility value will be treated as the optimal service composition plan

Fig.1 Specification of a Benchmark for achieving an optimal service composition plan

II. A BENCHMARK FOR EVALUATING A HISTORY RECORD-BASED SERVICE COMPOSITION PLAN

Benchmark for History Record-Based Service Composition Evaluation Current service optimization approaches often assume that the quality delivered by service providers does not change over time [22]. However, this assumption is often spoilt by the dynamic network environment [19]. Therefore, a service provider may fail to deliver their services with ''promised'' quality. It often causes some fluctuations in service performance, and the evaluation of a service will be suspicional, if the evaluation is scored by the QoS values' 'promised'' in advance. In [27], the ''promised'' QoS values are treated as tentative QoS values. To address this challenge, we leverage history records associated with a service's past transaction to reflect the service's quality in statistical probability. We aim at filter out the latent mendacious information. Technically, our work starts from a referred service composition evaluation method, which will play as a benchmark in this paper. For simplifying our discussion, a service's history transaction records specified by it QoS values will be indicated as ''a service's history records'' in short.

A. Definition 1 (Service Composition Class):

For two given tasks T1 and T2 engaged in a task will be selected as an optimal

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

global task scheduling. Fig. 1. Specifications of a benchmark for achieving an optimal service composition plan. scheduling, let P1 _ W Service ¼ ðWS11; WS12; . . .; WS1nÞ a n d P2 _ W Service¼ ðWS21; WS22; . . .; WS2mÞ be two service pools, in which P1 _ W Service consists of the candidate services for executing T1 and P2 _ W Service consists of the candidate web services for executing T2.

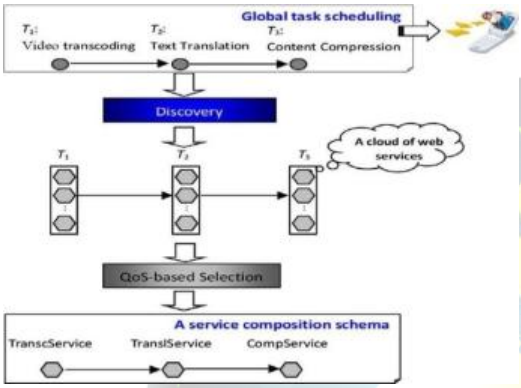

Fig 2 Cross Cloud Service Composition Scenario

In this paper, ðWS1i; WS2jÞ will be treated as a service composition class between P1 _ W Service and P1 _ W Service for orchestrating T1 and T2's later execution, where WS1i 2 P1_ W Service and WS2j 2 P2 _ W Service are held.

From Definition 1, we can deduce that there are n _ m service composition classes between P1 _ W Service and P2 _ W Service for orchestrating T1 and T2's later execution. A service composition class herein specifies a service composition plan associated with a global task scheduling.

B. Definition 2 (Service Composition Instance).

For a service composition class ðWS1i;WS2jÞ, let HR1i be the set of WS1i's QoS history records, and let HR2j be the set of WS2j's QoS history records. In this paper, ðhr1i_h; hr2j_kÞ will be treated as a service composition instance instantiated by service composition class ðWS1i;WS2jÞ, in which hr1i_h 2 HR1i, and hr2j_k 2 HR2j are held.

From Definition 2, we can deduce that if WS1i has M1i QoS history records, and WS2j hasM2j QoS history records, there are M1i _M2j service composition instances instantiated by service composition class ðWS1i;WS2jÞ. Moreover, with Definition 1 and Definition 2, a referred method is specified by Definition 3 for selecting an optimal service composition plan. It plays as a benchmark for history record-based service composition evaluation.

C.Definition 3 (Benchmark for History Record-Based Service Composition Evaluation).

For a service composition class CS: ðWS1i;WS2jÞ, its utility value can be computed by the average utility values of its service composition instances. For a group of service composition classes, the service composition class that has the largest utility value

Fig.2.Cross-cloud service composition scenario. The utility values can be computed according to (1) as specified in Section 2. The benchmark for achieving an optimal service composition plan is specified by Fig. 1.

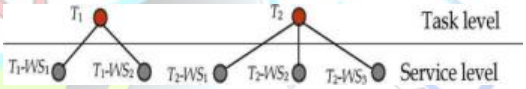

Fig 3.Task Service Tree Instances

D.Definition 4 (Task-Service Tree).

For a task and the candidate services that can fulfill the task execution's specification in functional and non-functional properties, a Task-Service tree is a two-level tree structure that consists of a main root node 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

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

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.

E Definition 5 (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 known k-means ðk ¼ 2Þ clustering algorithm. These two clusters 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-means ðk ¼ 2Þ clustering algorithm into Ti _ WSj'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 6.

F Definition 6 (Task-Service-History Record Tree).

A Task- 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.

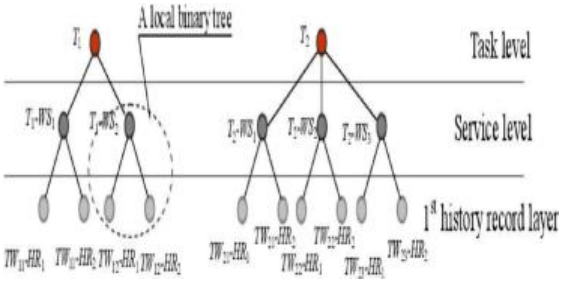

Fig 4.Hierarchical Relationship Among Task

G.Definition 7 (A Representative History Record's Local Contribution Score).

At a history record layer, for a representative history record, the average utility value of the composition instances it is engaged in, will be used as its local contribution score corresponding to this level.

H. Definition 8 (A Composition Class's Local Utility Value)

Corresponding to a history record layer, the average value of a composition class's instances will be used as the is a classic issue in service computing domain. Quality-aware composition class's local utility value.

III. RELATED WORKS AND COMPOSITION PLAN

Service-Oriented Computing (SOC) enables the composition of services provided with varying Quality of Service (QoS) levels in a loosely coupled way. Selecting a set of services for a (near-) optimal composition plan in terms of QoS is crucial when many functionally equivalent services are available [16]. Therefore, service composition web services has been fully investigated in [18], [19], [20], [21], [22], to name a few. In [18], [19], [20], the authors propose a per-service-class optimization as well as a global optimization using integer r programming. As opposed to integer programming, in [21], a genetic algorithm based approach is proposed, where the genome length is determined by the number of abstract services that require a choice to be made. GA based approach focuses on dealing with non-linear constraints.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

number of concrete services per abstract service increases. Considering that more and more functionally equivalent services are available on Internet proposes an interesting mechanism for cutting through the search space of candidate web-services, by using skyline queries [22] offline. Skyline queries identify non dominated web services on at least one QoS criteria. A non-dominated web-service means a web-service that has at least one QoS dimension where it is strictly better than any other web service and at least equal on all other QoS dimensions. Technically, linear programming model is often recruited in service composition evaluation [19], [22]. In practice, various composition styles, e.g., sequential parallel , alternative and loops can be engaged in a composition plan. In this paper, we focus on investigating the sequential composition model, as other styles can be reduced or transformed into the sequential model by present mature techniques as mentioned in [19]. Generally speaking, service composition is promoted in an open web environment. For a private cloud, the privacy and security are crucial issues in cloud service access. It often leads to an awkward situation that some QoS information may be unavailable in cross-cloud composition evaluation. It is just the reason that although it is assumed that the history records can be obtained through some monitoring mechanism [30], [31], there is few general QoS dataset widely recruited for testing the performance and accuracy of history record-aware service composition as mentioned in [32].In view of this challenge, an enhanced History record based Service optimization method, named Hire Some-II, is presented in this paper for cross-cloud service composition. This method aims at a tradeoff between the privacy and the time cost in cross-cloud service composition Concretely, our approach differs from the above approaches in three respects:

1. The k-means algorithm is implemented inside a cloud, and only a few representative history records are engaged in composition evaluation. It protect the privacy of a cloud, as the method does not require the cloud to while in [18], [19], [20], etc., the composition approaches did not take into account of the privacy issues, and held the assumption that all the QoS information can be available.

2. The tree mechanism presented in this paper is similar to the tree mechanism presented in [22]. However, our tree mechanism is promoted by imposing k-means algorithm on history records, while the tree mechanism presented in [22] is set up by imposing skyline queries on candidate web services. Besides, in [22], the tree mechanism is initiated by a binary tree; while our tree mechanism is initiated by a Task-Service tree, which is a multifork tree and is more compatible for real life systems.

3. Compared to its previous version investigated in [23], Hire Some-II not only protects the privacy of a cloud service for cross-cloud service composition, but also greatly speeds up the calculating process for selecting a (near-to-) optimal service composition plan with higher optimality and precision. It is suitable for developing a cross-cloud service composition plan over big data of history records with privacy consideration.

IV. CONCLUSION AND FUTURE WORK

In this paper, an enhanced History record-based Service optimization method, named Hire Some-II based on the previous basic one of Hire Some-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.

demonstrated the validity of our method compared to a bench mark. For future work, we plan to apply our method to some specific cloud systems for processing big data applications. Besides, as the privacy preservation for big data analysis , share and mining is a challenging research issue due to increasingly larger volume of datasets in cloud, we also plan to investigate the scalability of privacy preservation in big data applications with cloud service access.

V. REFERENCES

[1] S. Tai, J.Nimis, A. Lenk, and M. Klems, ‘‘Cloud Service Engineering,’’ in Proc. 32nd ACM/IEEE ICSE, May 2010,

[2] V. Nallur and R. Bahsoon, ‘‘A Decentralized Self-Adaptation Mechanism for Service-Based Applications in the Cloud,’’ IEEE Trans. Softw. Eng.

[3] H.Y. Lin and W.G. Tzeng, ‘‘A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding,’’ IEEE Trans. Parallel Distrib. Syst.,

[4] Christo Ananth, G.Poncelina, M.Poolammal, S.Priyanka, M.Rakshana, Praghash.K., “GSM Based AMR”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Issue 4,July 2015, pp:26-28

[5] 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