Multi Agent System for Cloud Service with Fidelity

Deepu S, Anu VR, Joyal Ulahannan

Abstract— Cloud computing is a virtualization concept, which provides large amounts of on-demand computation power, such as computing, storage and software. All these kind of on-demands computation power provided by commercial service providers such as Amazon, Google, Microsoft, etc. when an organization or a client using the cloud service, some time they need to face some unpredictable resource requirement and need to access another service based on the new requirement. But selecting a new service from the cloud is not a simple task because the users need to assure that the selected service can achieve our requirement and the selected service should be a secure one. In this system we use a reputation aware software service which is a self-adaptive and a feedback mechanism is used to ensure the trustworthiness of services. Thus the client will get appropriate service based on the Qos requirement along with quality, cost, and trust.

Index Terms -Fidelity in cloud services, Self-Adaptivity, Secure cloud selection, QoS monitoring, Feedback computation.

1 INTRODUCTION

In the last few years computing resources are easily available with the help of cloud computing. Many organizations use the cloud infrastructure to do their tasks, because with the help of cloud they can easily complete their task and do not need to spend more money to collect all the computing resources. We know that infrastructure providers manage cloud platform and lease resources and service providers rent resources from infrastructure providers to serve the end users. To select a service from the cloud the end user should confirm the following things.

1 The service can achieve the QoS requirement.
2 The service should be a secure one.

After selecting a service, applications need some additional resources and need to adapt to a new service. A self-adaptive [1] service based application should be able to change one of its services to another service that is functionally the same, but delivers better QoS. It integrated the decentralized service evaluation and using selection mechanism, and a price adjustment technique to allow for QoS constraint satisfaction. In self-adaptive system an application is selected in a decentralized manner. This cause to the application to exhibit to increase in performance, throughput or lower latency, as the situation warrants. But selecting a new trusted service from cloud is not a simple task, because the number of cloud service is large and some of them are malicious and we can't assure that the service should satisfy all the QoS based on the SLA. The Service Level Agreement (SLA) is a contractual basic between users and service provider. So before selecting a service we need to ensure the trustworthiness of the service. One's a user can find that the particular service is trusted he can easily select that service without any fear.

The current implementation of the Decentralized Self – Adaption mechanism [1] does not deal with the trustworthiness of the services. It only deals with how we can continuously select services when QoS changes. This paper mainly concentrated on how clients can achieve a service based on QoS and trustworthiness. So here we use the feedback and the reputation mechanism to ensure the trustworthiness of services.

2 PROBLEM STATEMENT

In the earlier stage of the service selection mechanism is done by combining web services together. Composite web services are services that are composed by other web services. In web service composition each service should be effective for achieving the QoS required by the users. The experiments have also shown that the performance of the method is affected by the difficulty of the composition problem. To avoid the complexities in the service selection self-adaptive systems [1] was developed. We know that one of the important issues in the cloud computing is the security. All data stored in a third party server and user need full trust in them. So the infrastructure provider must achieve confidentiality and audit ability. When selecting a new service from the cloud we should ensure the security and the trustworthiness of the service which we are going to select. In self –adaptive system [1], service providers are collected to a single platform and the services are selected according to the QoS requirement, but it does not deal with the trust and security of the system. Our contribution of this paper is the design of feedback based mechanism for trust and security certain tools are used to ensure the reliability, security, response time etc.

Disadvantages:

Lack of optimality- The self-adaptive mechanism [1] does not achieve optimum, either from the individual perspective or from the aggregate perspective.

Lack of trust- In service selection mechanism there is no trust between the user and the service provider.

3 OUR APPROACH

In artificial intelligence, we know that the agent takes appropriate action when circumstances changes. Likewise, in the cloud service, selection we can also use the agent's behavior when QoS requirements changes. In cloud services, Multi-Agent system [6] that allow an application to self-Adapt on the basis of their QoS and budget. It allows applications to select services, in decentralized [3] manner. Service based applications can change their constituent services at runtime. This means that they have the ability to change both, their functionality and their QoS attributes dynamically. A self-adaptive service based [1] application should be able to change one of its services to another when a new service with the same functionality and different QoS will obtain. This would lead to the application being able to access better performance, higher throughput or lower latency, as the situation warrants. The proposed service rating allows feedback to be assigned to a delivered service that objectively reflects the satisfaction or dissatisfaction with the rendered performance and quality. It helps consumers predict the credibility of the service offer, and the trustworthiness of the service provider prior to reaching an agreement. So in the decentralized self-adoption [1] system we can add the feedback mechanism to improve the trustworthiness of the users.

4 RELATED WORKS

There have been a lot of researches on the optimal cloud service selection

4.1. Web service composition

Web service composition [2] means combining web services together to get an optimal service. In web service composition we need to ensure the performance of the services which we are going to combine. When the request for a new service is occur the system need to find the appropriate service from the UDDI (Universal Description Discovery and Integration) registry and the composition manager combine the appropriate services. The effectiveness of the new service is depending on the quality of the individual services. When the number of services is increasing the system becomes complex because finding an appropriate service from a pool of service is difficult. We know that nowadays the number of cloud services is in a large quantity and the web service composition is not becoming a suitable method for a cloud service selected

4.2 Decentralized Self- Adaption

Self-Adaption in the computer system [1] is a concept in which the system will do appropriate action when circumstances changes. This adaptability becomes the key feature of several software systems. Commonly adaptability is based on a feedback loop and accomplished through sensors and actuators. The sensors decide what to do and take appropriate action to the system through actuators. Self-Adaptivity in the cloud service selection is shown in fig1.

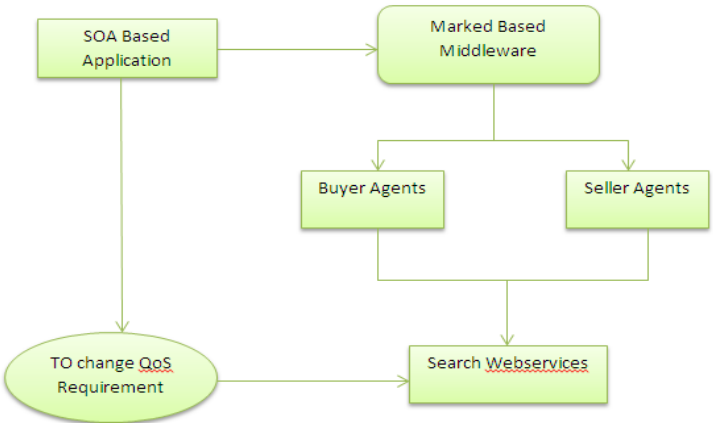

Fig 1: Self-Adaptivity in cloud

In the self-adaptive the system is considered as a market place [7] where service providers can register their services. The buyer agent buys services and generates bids. The seller agent sells services and generates Asks. When the Qos requirements changes a new service is selected through self -Adaption.

5 SERVICE SELECTION WITH FIDELITY

To select a service from the cloud the user has to trust that service because several types of services are available in the cloud with the same functionality and some of them may be malicious. So we need to avoid the malicious services to be selected.To ensure the trustworthiness in the service selection a feedback mechanism can be used. This means that the service, selection depends on the feedback from the users.

5.1 Feedback Computation

Feedback is computed on the basis of past feedbacks. It helps users to predict the credibility of the service and the trustworthiness of the service. Past feedback shows the past behaviour of a service and may give an indication of its future behaviour. Thus, it is necessary that the automated rating process provides feedback that corresponds to the level of satisfaction/dissatisfaction with service delivery. To achieve this we use stochastic work flow reduction [1] mechanism. By using stochastic workflow reduction a rank is calculated for each attribute such as feed aback, response time, security, etc... After finding the rank list an overall mark is calculated and selects a service which has highest mark based on the QoS requirement. To achieve this we consider that all service providers are grouped into one market and register all the services into the market. In a service based application QoS monitoring [4] [5] is important in the selection of a particular service. So in this system application agent do periodic checking of reliability, response time, reliability, latency security values of services and update the results. When the user request for a particular service, the service selection depends on the feedback and the Qos requirement about that service . Thus the client can assure the trustworthiness of that service in the service selection.

5.2 Structure Of Service Selection

To select a service from the cloud the user can mention the type of service they want and the QoS requirement. Each request from the user is considered as a bid. Service selection contains accepting bid from buyers and matching the bid with Asks. When a request for a new service is arriving that particular bid is entered into an order book and a matching is done to find corresponding Asks from the order book. To select a service all attributes of a service (QoS) such as feedback, reliability, response time are checked and when a service with the required QoS is obtained, then that service is allocated to that user. By using the feedback mechanism the client can assure the credibility of the service. After the transaction is completed the bid Ask is clear from the order book. The fig 2 shows the architecture of the service selection and it shows that how the service is selected based on the feedback.

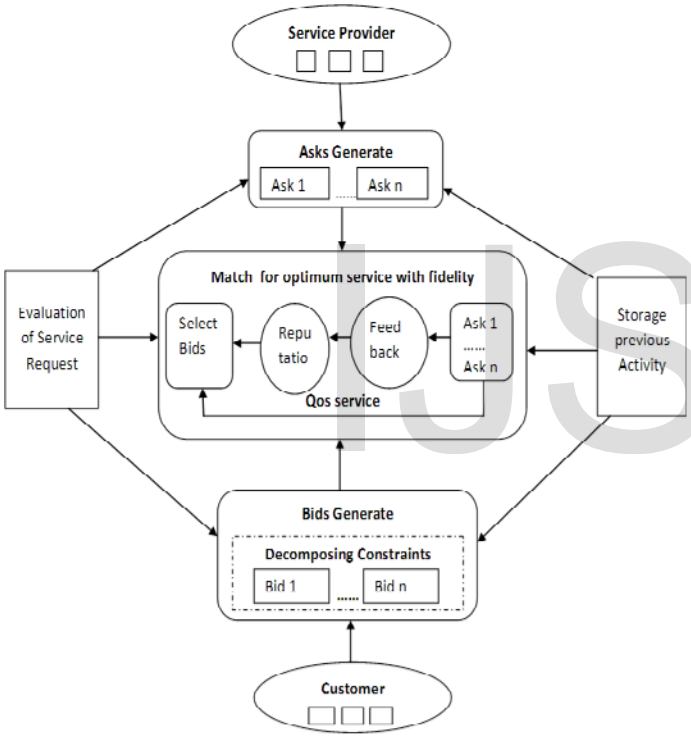

Fig 2 Feedback based service selection

1. User $u_i$where i=1 to n send request R_i to a cloud service system. Services are selected on a particular type T and it should satisfy a function f to meet the user requirement Total services in the cloud system can be represented by
$TS_i = \{TS_i^1, TS_i^2 \ldots TS_i^n\}$

2. Here we assume that the user u_i can provide q type of requirements is $req_i = \{Q_i^1, Q_i^2 \ldots Q_i^n\}$
Users u_i assign weights to all attributes $w_i = \{w_i^1, w_i^2 \ldots w_i^n\}$

3. Find the trust evidence of all services TS_i^x in TS_i.

4. Calculate the reputation value based on the trust evidence.

5. Calculate the aggregate trustworthiness of services TS_i.

6. Select appropriate service TS_i^x from TS_i which satisfies the user requirement.

7. Collect feedback from the active user to ensure the trustworthiness to others.

6 Experimental Results

We simulate the system with 100 applications. Each application trying to achieve their QoS requirement based on the feedback and cost. Because of the absence of a deterministic algorithm we evaluate that the performance of the system on the basis of adaption using feedback. We measure how many trading rounds needed for an application take to be get satisfied. After achieving the required QoS, that application removed from the trading rounds. Every application has a separate workflow and, different number of abstract services. The graph shows that time taken for adaption with customer satisfaction is increased along with the number of services increased. Our experiments have shown that the performance of the system is increased and all applications can select the services based on feedback in a self-adaptive manner.

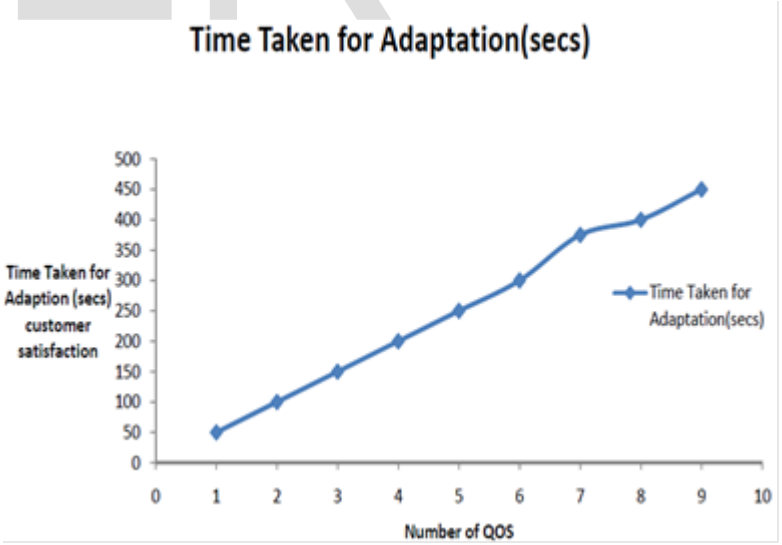

Fig 3 Performance graph

7 Conclusion

In this paper, we develop a new mechanism to select cloud services in a trusted manner, which help the users to assure the trustworthiness of cloud services. By using the feedback mechanism services are selected in a trusted manner with self-

Adaptivity. Trust related problem in the cloud service selection can be avoided through our system. One of the other features of this system is the periodic checking of the QoS .This also helps the users to achieve better service selection. So by the development of the stem users can select the services in a trusted manner.

REFERENCES

[1] Vivek Nallur and Rami Bahsoon, Member, IEEE. " A Decentralized Self-Adaptation Mechanismfor Service-Based Applications in the Cloud",2013

[2] Mohammad Alrifai,Dimitros Skoutas,Thomas Risse, "Selecting skyline services for QoS-based web service composition",2010.

[3] Luciano Baresi,SamGuinea and giodano Tamburrelli, "Towards decentralized self-adaptive component-based system" 2008.

[4] Anton michlmayr,Florian Rosenberg,Philipp Leitner,Schahram Dustdar, "Comprehensive QoS Monitoring of web Services and Event-Based SLA violation Detectiuon,November 2009".
Parallel and Distributed Systems, vol. 13, no. 3, pp. 193-211, Mar. 2002.
[5]JCardoso,ASheth,and J.Miller,"Workflow Quality of Service ",technical report,March 2002 Ptems,

[6] G. Di Marzo Serugendo, M.-P. Gleizes, and A. Karageorgos, "Self-Organization in Multi-Agent Systems", June 2005
[7] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," Proc. 10th IEEE Int'l Conf. High Performance Computing and Comm., pp. 5-13, Sept. 2008..