Research online Publishing
www.ronpub.com

Open Access

Open Journal of Cloud Computing (OJCC)
Volume 2, Issue 2, 2015
http://www.ronpub.com/ojcc
ISSN 2199-1987

A Trust-Based Approach for Management of Dynamic QoS Violations in Cloud Federation Environments

Manoj V. Thomas, K. Chandrasekaran

Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, Karnataka, India-575025, {manojkurissinkal, kchnitk}@gmail.com

Abstract

Cloud Federation is an emerging technology where Cloud Service Providers (CSPs) offering specialized services to customers collaborate in order to reap the real benefits of Cloud Computing. When a CSP in the Cloud Federation runs out of resources, it can get the required resources from other partners in the federation. Normally, there will be QoS agreements between the partners in the federation for the resource sharing. In this paper, we propose a trust based mechanism for the management of dynamic QoS violations, when one CSP requests resources from another CSP in the federation. In this work, we have implemented the partner selection process, when one CSP does not have enough resources, using the Analytic Hierarchy Process (AHP) and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) methods, and also considering the trust values of various CSPs in the federation. We have also implemented the Single Sign-On (SSO) authentication in the cloud federation using the Fully Hashed Menezes-Qu-Vanstone (FHMQV) protocol and AES-256 algorithm. The proposed trust-based approach is used to dynamically manage the QoS violations among the partners in the federation. We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results are also given.

Type of Paper and Keywords

Regular research paper: *cloud federation, dynamic QoS violation, QoS, Single Sign-On, trust calculation, trust management*

1 Introduction

The widespread acceptance of Cloud Computing has contributed to the design and development of Cloud Federation [1][19][24]. Cloud Federation is an association of different Cloud Service Providers. In the standard Cloud Computing model, a client gets the required services from a single Cloud Service Provider, and this approach has several challenges associated with it. Due to some reasons, if a CSP cannot handle the service requests initiated from the cloud customers, it can leave several customers who depend on that service provider, without access to the required resources and services. Also, depending on a single cloud service provider, sometimes makes it difficult to ensure the adequate responsiveness and QoS to the clients.

In reality, the Cloud Service Providers have finite amount of resources with them. Also, a CSP cannot lose an important customer because of the lack of available resources at the moment, and thereby not being able to cater to the needs of that customer. To overcome these limitations, CSPs got together as a federation. For many service providers, in order to meet the dynamic and unpredictable user requirements, cooperation with other

service providers is an option. This cooperation can be utilized to access resources and services from other partners in the federation to deliver the required QoS to the customers. The CSPs in the federation can share the cloud infrastructure among them in order to have better resource utilization and improved QoS to the cloud consumers.

Thus, this collaboration ensures the support in terms of information and resource sharing among the partners in the cloud federation environment, and the improved QoS in terms of availability, reliability and response time of the services delivered by the various cloud service providers. Hence, the primary reasons for the formation of Cloud Federation are better resource utilization and the increased revenues for the CSPs, and the availability of reliable cloud services with no vendor lock-in for the cloud consumers.

1.1 Need for the Management of Dynamic QoS Violations in the Cloud Federation

Even though the cloud computing paradigm promises to offer infinite resources, in reality, the resources with each and every Cloud Service Provider are finite. Sometimes, there could be requests from the cloud users for rapid increase in the usage of their computing, memory or network resources due to reasons such as failure of a server or data centre or to meet the sudden request made by their own clients. In this case, when the Cloud Service Provider runs out of resources, the Service Provider can get the required services from partners in the Cloud Federation or the Inter-Cloud.

Generally, there will be Service Level Agreements (SLAs) between the partners in the Cloud Federation to share the resources. Due to the dynamic nature of customer requirements, sometimes a CSP in a federation may urgently need some resources from other CSPs in the federation to meet the customer requirements, as the requested resources are unavailable with the CSP at that time. Since the CSPs in the cloud federation operate by the Service Level Agreements among them, a CSP can get the services as per the QoS agreement in the SLA. Normally, the process of SLA renegotiation is carried out among the CSPs in order to modify the QoS parameters of the services agreed among them. Now, if a request comes to a CSP from another CSP in the federation for some resources whose QoS features are not as per their prior agreement, how to deal with such a request in the federation dynamically without the time consuming SLA renegotiation at that time is an issue to be considered.

QoS/SLA violation in the Cloud Federation occurs when one CSP requires some service from another CSP whose QoS features differ from what have been agreed in the SLA between them. Suppose that there is an SLA agreed between CSP-A and CSP-B in the cloud federation. Also, assume that as per the SLA, CSP-B has agreed to give the service consisting of a maximum of n number of VMs of type 'small' to CSP-A. Now, imagine that CSP-A makes a service request of m VMs ($m > n$). Also the type of the VMs requested is 'large'. This is an example of the QoS/SLA violations between the CSPs. Even though this example is simple, we have considered this just to show the working of our approach. Hence, in order to make the best use of the federation, we need a dynamic management of this possible QoS violations among the partners in the federation so that the mutual benefits of the CSPs in the federation, in terms of reliability, reputation and the economic benefits are improved.

1.2 Role of Trust in the Cloud Federation Environments

The effective management of trust among the entities is significant for the service computing environment as it is for the activities involving human beings. Human beings trust others depending upon the environment or contexts, and this trust values change from time to time. According to Azzedin and Maheswaran [2], trust is defined as: "trust is the firm belief in the competence of an entity to act as expected such that the firm belief is not a fixed value associated with the entity, but rather it is subject to entity's behaviour and applies only within a specific context at a given time". Therefore, trust is a dynamic aspect of an entity which varies from very trustworthy to very untrustworthy.

The trust value of an entity is derived based on the previous experience with that entity in a specific context. Also, the trust value associated with a particular entity is not the same always as it varies from time to time. One entity can trust another entity in a system also based on the reputation of that particular entity. In this way, the reputation of an entity can be effectively used for building the trust [3][4]. Azzedin and Maheswaran [2] define reputation of an entity as: "the expectation of its behaviour based on other entities' observations or information about the entity's past behaviour at a given time".

For an entity, there can be either direct or indirect experience with another entity. Direct experience shows that the entities have had some direct interactions between them in the past, and how one entity learns about the behaviours of the other entity using this interactions. Indirect experience of an entity is developed based on the recommendations given by other trusted members in the community. Hence, while calculating the trust value of an entity, it considers the reputation of that entity also. Thus, reputation has direct effect on the trust of an entity. That means, a good trust value of an entity results in

good reputation of that entity and vice versa [5].

In a multi-cloud environment, it requires the association among multiple clouds and the effective establishment and management of trust among them is of paramount importance [6]. In the multi-cloud environment, the partner clouds are independent and loosely coupled which makes the trust management a challenging one. In this environment, the trust management system should help in distinguishing various entities as trustable or not so that effective cooperation of the CSPs are ensured. Although it has been discussed that efficient resource allocation and utilization requires a high degree of trust values [7], to the best of our knowledge, the issue of how to solve the dynamic QoS violations in a cloud federation environment has not been addressed in a satisfactory manner.

Even though cloud federation offers various advantages, establishment of trust among the partners in the federation is a challenging issue [8]. In order to make the best use of cloud federation in terms of resource management, there should be an efficient mechanism for the establishment and evaluation of the dynamic trust among the CSPs in the federation [9]. Researchers have been working on various trust models in the Cloud Computing domain that evaluates the trust of various CSPs [10][11]. Majority of these trust models focused on evaluating and managing the trust between cloud users and the CSPs. Very few of the proposed trust models focuses on effective trust management in the Inter-Cloud domain and hence, the present Cloud Federation scenario requires effective trust management approaches. Considering the future scope of cloud federation and also the role of an effective trust management system in the domain, we are proposing our approach in this paper.

The major contributions of this paper are:

- Design and implementation of a partner selection approach in the cloud federation environment, using the Analytic Hierarchy Process (AHP) [28] and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [29][30] methods, that can be used by a CSP, when it does not have enough resources to meet the clients' requirements.

- Design and implementation of a SSO mechanism [31] in the cloud federation, using AES-256 algorithm [33] and the FHMQV protocol [32].

- Design of the Trust-Based mechanism for dealing with the dynamic QoS violations in the cloud federation environment.

- Implementation of the proposed approach using the CloudSim toolkit [35].

- Discussion of the results obtained highlighting the advantages and the disadvantages of the proposed approach.

The rest of the paper is organized as follows: Section II describes the work done in the area of resource management in the cloud federation environments, highlighting the merits and demerits of various approaches. Section III presents the trust-based approach for the management of dynamic QoS violations in the cloud federation environments. Section IV discusses the workflow of the proposed trust-based approach. Section V presents the results and analysis of the proposed approach with pointers to the future works, and Section VI concludes the paper.

2 Literature Review

In this section, we throw light on the relevant research activities in the area of resource management in the cloud federation or inter-cloud domain, analyzing the works carried out by the researchers. The research in the field of effective resource management in the Inter-Cloud environment is still in its nascent stage, and some of the relevant approaches proposed by the researchers in this area are discussed in this section.

Goiri et al. [12] use the characteristics of cloud providers in a federated cloud such as the resource utilization level of a CSP, pricing of the VMs, capital costs, operational costs etc. while handling the resource requests of the cloud customers. Depending on the above characteristics, upon receiving a resource request, a CSP takes the decision regarding allocation within itself, outsourcing to other federated clouds, insourcing from other federated clouds, turning on/off various nodes in the data centres. But in this work, the major focus is on the cost aspect of the resource management, and also the management of QoS violations is not discussed.

Wu et al. [13] provide a QoS-based research component composition architecture for research collaboration using distance based evolutionary algorithm. The aim of the algorithm is to compose and optimize research components according to multi-QoS attributes. A game-theory based distributed resource management mechanism for data intensive IaaS cloud providers in a federation environment is proposed by Hassan et al. [14]. These works also lack the management of QoS violations among the CSPs.

Hassan et al. [15] propose a multi-objective optimization model for partner selection in a market-oriented dynamic collaboration (DC) platform of Cloud Service Providers, to minimize the conflicts among the providers during their negotiation. The price and QoS of the service offered by the various CSPs, and also the success of

any previous association among them in the past are considered. For this multi-objective optimization, they developed multi objective genetic algorithm. In this case, they have considered the group of CSPs satisfying the users' request, and the group bid of different CSPs is treated as a single bid. But, our work focuses on how each CSP handles the dynamic QoS violations so that the best possible service is offered to the CSPs without requiring the SLA renegotiation at that time.

Kertsz et al. [16] discuss the integrated federated management and monitoring approach for the autonomous service provisioning in the federated clouds. The users submit the service requests to the brokering component called the Generic Meta Broker Service (GMBS). The service provider's information and health metrics are stored in a Global Service Registry (GSR). The GMBS matches the service request with the information stored in the GSR and selects the suitable cloud broker. In this approach, every CSP has a broker for dealing with the users' requests.

Chen et al. [17] proposed a game-theoretic approach to solve the service selection problem in the cloud environment. The proposed solution is based on game-theory and for each provider; they merge the consumer's game with the provider's game. Based on the responses arising from the interaction among the client and other CSPs, a CSP can select the suitable cloud provider as the business partner. In [18], Stihler et al. propose the architecture for Federated Identity Management in a scenario similar to the Inter-Cloud environment. The work focuses on sharing of information or resources across all the three cloud service models such as SaaS, PaaS and IaaS.

The works carried out by Celesti et al. in [19][20][21] and Tusa et al. in [22] present a heterogeneous horizontal cloud federation model, for CLoud-Enabled Virtual EnviRonment (CLEVER). These works use the concept of a middleware component called the Cross-Cloud Federation Manager (CCFM) that could be integrated into the Cloud Manager component of the Cloud Service Provider. This helps the participating clouds to be a part of the cloud federation. The CCFM consists of three subcomponents, called the Discovery Agent, Match-Making Agent and Authentication Agent, and they are responsible for performing the required functions for the cloud federation. This work does not discuss the issue of the management QoS violations in the cloud federation scenario.

Bernstein et al. presented a blueprint for the design of Inter-cloud in [23][24] and [25]. This blueprint is designed considering the interoperability factor among the various Cloud Service Providers and is focused at the Internet scale. In this work also, the dynamic QoS violations are not discussed. Goiri et al. carry out an analysis of the cost benefits of resource sharing in cloud federation in the work presented in [26]. The Cloud Scheduler project explained in [27] by Armstrong et al. focuses on developing a model for resource provisioning and sharing among the various participating Clouds. In this work, the authors concentrate more on the scheduling of applications among the partners in the federation, and not on establishing the federation.

Based on the literature review, it is seen that the issue of resource management in the cloud federation environments lacks effective solutions to meet the requirements of the present day cloud federation environments, which emphasizes the need for further research in this domain. We also note that there are few works that deal with the dynamic QoS violations in the inter-cloud environment. Even though exiting works are complementary to the approach discussed in this paper, to the best of our knowledge, this is the first work that attempts to discuss the dynamic management of the QoS violations considering the trust and reputation of various CSPs in the cloud federation environment. The proposed mechanism effectively incorporates the trust management in the cloud federation to deal with the management of QoS violations, and thereby improving the reliability and efficiency of the CSPs in the cloud federation.

3 Trust-Based Management of Dynamic QoS Violations in the Cloud Federation

In the Cloud Federation, as already mentioned, whenever a CSP runs out of resources, it can get the resources from other CSPs in the federation. Normally, there will be SLAs between the CSPs in a Cloud Federation regarding the details of the services agreed among them. In order to change the QoS of the agreed service, SLA-renegotiation is required between the participating CSPs in the Federation. In this section, we propose a trust-based mechanism to deal with the QoS or SLA violation among the CSPs in the federation so that without the SLA renegotiation at that time, a CSP can get the required services from other CSPs in the federation, even though the QoS of the service requested is not exactly as per the SLA agreed between the CSPs. Hence, this approach enables the CSPs to improve their profits and the reputation in the Cloud Federation environment.

3.1 Access Control Framework

The overview of the access control framework dealing with the dynamic QoS violations as implemented in our work is shown in the Fig. 1. In our implementation, in order to meet the resource requirements of a user, when the local resources are unavailable, a CSP (CSP-1) selects the most suitable CSP in the federation using

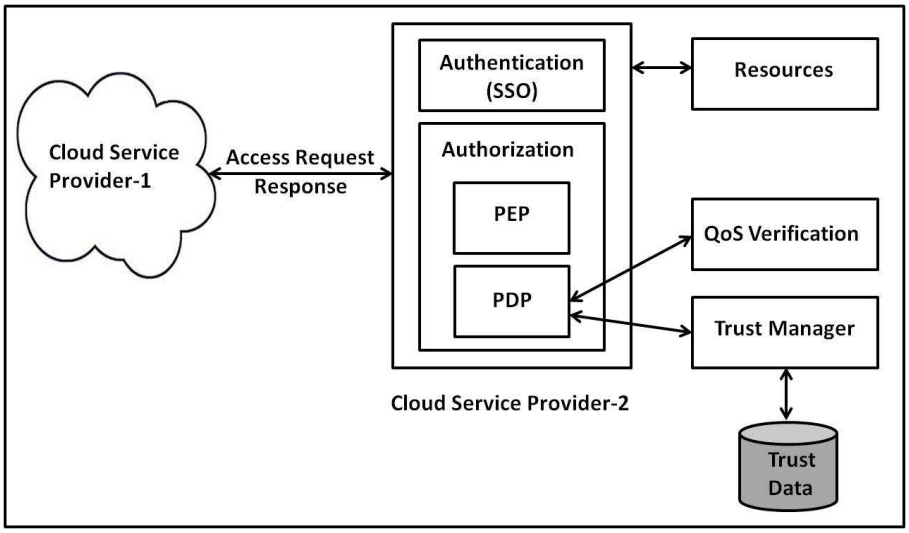

Figure 1: Overview of the Access Control Framework

the AHP [28] and the TOPSIS methods [29][30]. Now with the selected CSP, if there is a QoS violation when it gets the access request from the CSP-1, our proposed approach comes into action. Hence, the access control framework of the selected CSP takes the access decision as to whether the access request should be permitted or not considering the trust and reputation of the requesting CSP (CSP-1). Thus, the various components in this framework as shown in the Fig. 1 are:

3.1.1 Authentication

This module deals with the authentication of the requesting user. In our implementation, we have implemented the Single Sign-On (SSO) [31] for the authentication of the requesting users at different CSPs in the federation. In our implementation, an access request of the user is passed from one CSP to another, in case a CSP runs out of resources at a particular time. We have implemented in this work, the SSO approach in the cloud federation environment using AES-256 algorithm [33] and the FHMQV protocol [32]. Every user needs to be authenticated before availing services from the CSPs in the federation.

3.1.2 Authorization

When a CSP gets service request from other CSPs in the federation, it takes the access decision dynamically considering various factors. Hence, this module verifies the access rights of the requesting CSP. This module of a CSP has two components, PEP (Policy Enforcement Point) and PDP (Policy Decision Point).

1. PEP-The PEP contacts the PDP for access decision and implements the access decision taken by the PDP.

2. PDP-Whenever a CSP receives a service request from another CSP, this component verifies the request and takes a decision as to whether the request should be permitted or not. As shown in the Fig. 1, this component contacts the QoS-Verification module for verifying the QoS terms of the agreed service with the requesting CSP. In case a CSP requests some services whose QoS features do not exactly match with that mentioned in the SLA, the PDP contacts the Trust Manager module for calculating the trust value of the requesting CSP. Trust Manager calculates the local trust value, by accessing the trust data stored locally, and the recommended trust value (reputation) by contacting other trusted CSPs in the federation. If the final trust value of the requesting CSP is above the trust threshold, the resource request from the requesting CSP (CSP-1) is accepted, otherwise rejected.

3.2 Single Sign-On (SSO)

Single Sign-On (SSO) is a mechanism used for authentication in which a service consumer is required to be authenticated only once while accessing various services from multiple service providers [31]. The process of SSO involves the association among the following entities: Cloud Service Consumer (CSC), Relying Party or Cloud Service Provider (CSP) and the Identity Provider (IdP). The CSP and the IdP have mutual trust established between them. That is, IdP offers Identity Management functions to the CSP. Before accessing the services from

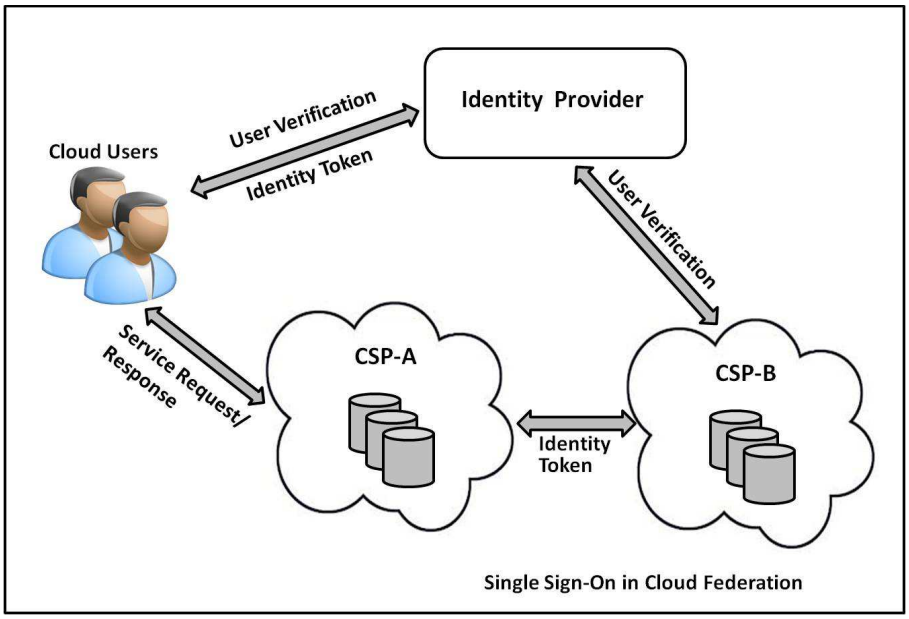

Figure 2: SSO in the Cloud Federation

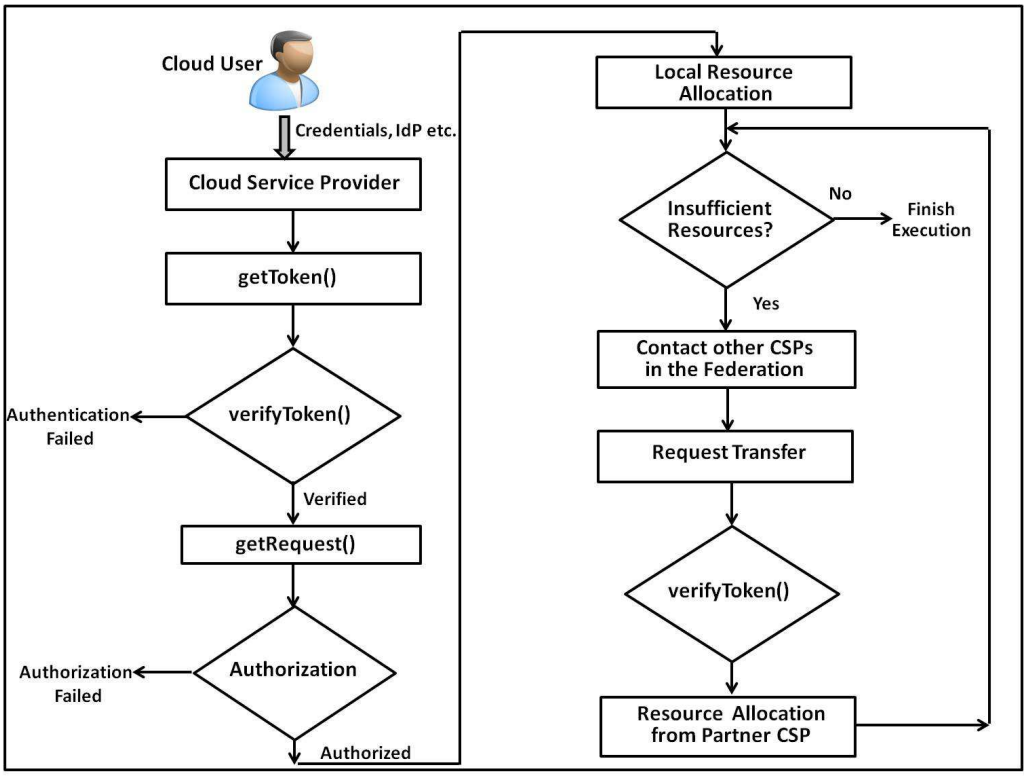

Figure 3: Overall flow of the SSO in Cloud Federation

the CSP, the Cloud Service Consumer has to get authenticated as a valid user from the IdP.

Since the CSP and IdP are part of the association, and they have mutual trust with each other, the user is allowed to access the services from the CSP after successful authentication. Hence, the Identity Federation supports Single-Sign On as the users are able to access multiple services from the different CSPs using the same identity token issued by the Identity Provider. Because of this association, the service providers can concentrate more on their core services, since the identity management operations are taken care of by the Identity Providers. The overview of the SSO in Cloud Federation is shown in the Fig. 2. Thus, in the Single Sign-On (SSO) mechanism in Cloud Federation, a user needs to verify his credentials and get authenticated himself only once during an active session of accessing cloud services. The cloud users are benefitted in such a way that they will be able to access the services offered by different CSPs seamlessly without the need of providing the identity credentials again and again for accessing the services.

The overall flow of the SSO in Cloud Federation implemented in this work is shown in the Fig. 3. As shown in the figure, in order to access the cloud services in the federation, the cloud user submits the credentials and also the details of the Identity Provider (IdP) supported by the CSP. The CSP verifies the identity token by contacting the IdP mentioned (provided that this IdP is trusted by the CSP considered). Upon successful authentication, the user request is processed to verify the access request of the user. If the verification of the authorization is successful, the local resources are allocated to the user.

If the local resources are insufficient to meet the client's request, this CSP contacts other CSPs in the federation (using the Rank Table as explained in the Section 3.3.4) for the allocation of the required resources. Upon receiving the resource request along with the corresponding identity token, the other CSPs in the federation verify the identity of the user by contacting the corresponding IdP. In this case, the user does not need to enter the identity credentials each time he gets resources from the cloud partners in the federation. The identity credentials are submitted only once to the first CSP alone, while making the access request.

3.3 Proposed Approach for the Management of Dynamic QoS Violations

The various functional components in the proposed approach for dealing with the dynamic QoS violations are shown in the Fig. 4. They are discussed in the following sections.

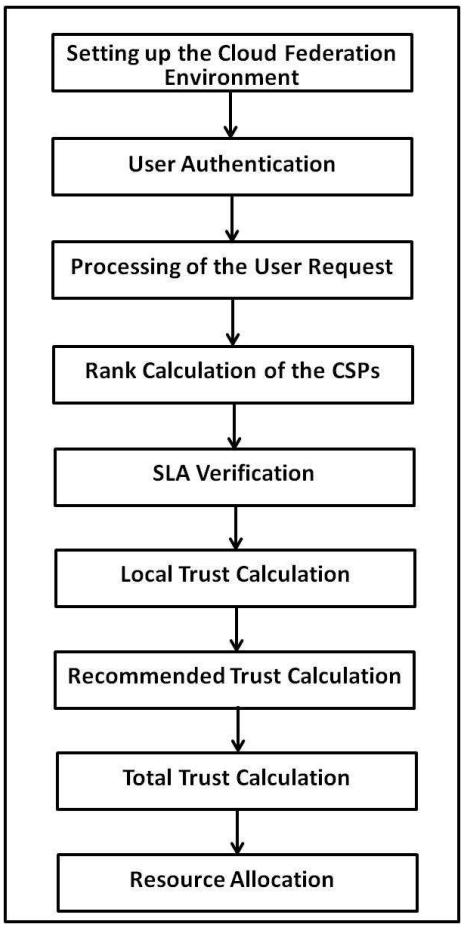

Figure 4: Management of the Dynamic QoS Violations

3.3.1 Setting up the Cloud Federation Environment

The required cloud federation environment needs to be set up in order to implement and test our proposed approach. We have set up a Cloud Federation Environment of 25 CSPs using the CloudSim toolkit [35] to implement our proposed mechanism.

3.3.2 User Authentication

In the cloud federation environment, a user requesting the service needs to be authenticated. When one CSP does not have enough resources, it can transfer the access request of the user to other CSPs in the federation. In order to avail the requested services from that CSP, the user needs to be authenticated there also. We have implemented the SSO authentication as explained in the previous section to facilitate that. In this case, the user need not enter the identity credentials again and again, but only once at the first CSP of the federation.

3.3.3 Processing of the User Request

The user request is analyzed to verify the details of the requested service such as type of VMs, number of VMs etc.

3.3.4 Rank Calculation of the CSP

In our implementation, when a CSP does not have enough resources to meet the requirements of the user, it ranks the various CSPs in the federation so that the best CSP can be selected to transfer the user's request. In the federation, there can be many CSPs offering different types of services with different QoS features such as availability, reliability, uptime, response time, cost etc. Also, a CSP in a Cloud Federation may not have equal trust values towards every other CSP in the federation at a time. Hence, for any CSP in the cloud federation, the selection of suitable CSP(s) for availing the required resources is an important activity in order to increase its business value.

The cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. We have used AHP [28] and the TOPSIS [29][30] methods for the rank calculation of the CSPs in the federation. The various steps in the process of Rank Calculation is shown in the Fig. 5. When a user request is processed by the CSP, the QoS parameters associated with the user request are given suitable weights using the AHP method, and these weights are used in the TOPSIS method to rank the various CSPs in the Cloud Federation according to the user requirements. The calculated rank values are stored into the database for further reference by the CSP. Simulation results show the effectiveness of this approach in order to efficiently select the trustworthy partners in large scale federations to ensure the required QoS to the cloud consumers.

3.3.5 SLA Verification

In our work, when a CSP runs out of resources, it selects a suitable CSP from the federation using the process of Rank Calculation (as explained in the previous section) to transfer the user's resource request. Now, when the selected CSP gets the resource request from a CSP, it verifies the SLA or QoS agreements with the requesting CSP. If the QoS features of the current request match with that present in the SLA, the request is accepted by the CSP and the available resources are given to the requesting CSP. If there is a violation of the QoS agreed between them, the proposed trust-based mechanism is used to deal with the resource request as explained in the following sections.

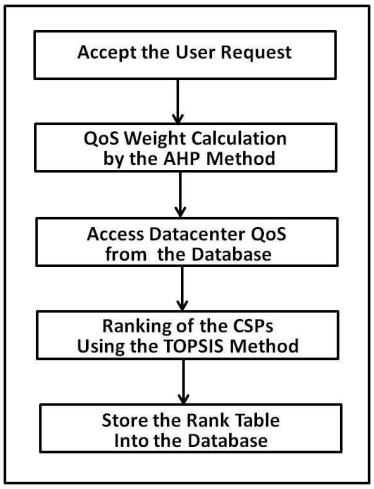

Figure 5: Rank Calculation using AHP and TOPSIS Methods

3.3.6 Local Trust Calculation

In our proposed approach, whenever a CSP gets a service request from another CSP, if there is an SLA or QoS violation, as a matter of mutually improving the economic benefits and the reputation of the CSPs in the federation, the CSP receiving the request calculates the trust value of the requesting CSP locally. If the local trust calculated is above the predefined trust threshold, the service request is accepted. Otherwise, the CSP calculates the recommended trust of the requesting CSP.

3.3.7 Recommended Trust Calculation

This module calculates the trust value of various CSPs in the federation to select the trusted CSPs. Then, the feedback regarding the requesting CSP is collected from those trusted CSPs to calculate the recommended trust of the requesting CSP.

3.3.8 Total Trust Calculation

In our implementation, in order to take a decision on whether to accept or reject the access request made by a CSP, its local trust and recommended trust values are calculated as explained in the preceding sections. Then, the total trust of the requesting CSP is calculated as:

Total Trust of the CSP=(Local Trust + Recommended Trust)/2.

3.3.9 Resources Allocation

Depending on the total trust value of the requesting CSP, the PDP of the CSP, as shown in the Fig. 1, takes a decision to permit the access request if the trust value of the

requesting CSP is above the trust threshold maintained in the system. The access request of the CSP is rejected if the final trust value is less than the trust threshold. Local Trust Calculation and the Recommended Trust Calculation of the requesting CSP are discussed in details in the following sections.

3.4 Local Trust Calculation of the CSP

The calculation of the local trust of the requesting CSP by another CSP in the federation involves the following parameters.

3.4.1 Probability of Success

The Probability of Success of the requesting CSP with any other CSP in the federation shows how many transactions with that CSP were successful in the past. This can be calculated as: Probability of Success=(x/y), where x is the total number of successful transactions and y is the number of total transactions initiated with that CSP.

3.4.2 History of Interaction

This shows the lead of the number of successful transactions over the number of unsuccessful transactions with a particular CSP. It is calculated as: History of Interaction=$(x - y)/z$, where x is the total number of successful transactions, y is the total number of unsuccessful transactions and z is the number of total transactions by a specific CSP with another CSP in the federation.

3.4.3 Existing Trust

This shows the existing trust value of a CSP before the current trust value is calculated. As this factor indicates, a CSP with a higher existing trust value is expected to have a more positive impact on the calculation of the current trust value, than a CSP having a lower existing trust value, or a CSP joined the federation recently.

3.4.4 Degree of Association

For calculating the total trust of a CSP in the federation, the total period of association of the CSP with the federation is taken into account, by considering the date and time of joining of the CSP with the Cloud Federation. Based on the date and time of joining the federation, the Degree of Association is given a value x for the CSP, where $x \in [0, 1]$.

3.4.5 QoS Value

While calculating the trust of the requesting CSP in the federation, the QoS parameters are considered separately to distinguish one CSP from another in the federation. In our work, the QoS parameters considered are availability, reliability, confidentiality, integrity and response time. It shows the details of the previous interaction with that CSP in the past. Hence, this calculation involves the following factors:

Availability Factor: Availability Factor is calculated as (x/y), where x is the total number of times the service from the requesting CSP was available when requested and, y is the total number of service requests made to that CSP.

Reliability Factor: Reliability Factor is calculated as (x/y), where x is the total number of times the service was reliable and, y is the total number of times the service was available from that CSP.

Confidentiality Factor: Confidentiality Factor is calculated as (x/y), where x is the total number of times the confidentiality was intact with the service from the requesting CSP and, y is the total number of times service was available from that CSP.

Integrity Factor: Integrity Factor is calculated as (x/y), where x is the total number of times the integrity was intact with the service from the requesting CSP and, y is the total number of times service was available from that CSP.

Response Time Factor: Response Time Factor is calculated as (x/y), where x is the total number of times the response time was within the promised limit and, y is the total number of times service was available from that CSP.

Hence, the final QoS Value of the requesting CSP in the federation is calculated as:

QoS Value=(Availability Factor + Reliability Factor + Confidentiality Factor + Integrity Factor + Response Time Factor)/5.

Hence, the Local Trust Value of the CSP is calculated as:

Trust Value= (Probability of Success + History of Interaction + Degree of Association + Existing Trust + QoS Values) /5.

QoS values are evaluated while calculating the local trust value of a CSP by another CSP. Generally, there are many CSPs in the cloud federation each with its own business priorities. Also, the trust value of a CSP calcu-

lated by any other CSP is subjective. That means, same CSP may be trusted differently by another two CSPs in the cloud federation. While calculating the QoS values, we have considered five factors such as availability, reliability, confidentiality, integrity and response time factors. In real time implementation, the weightage given for each parameter may be different from one CSP to another depending on their business objectives. Also, it depends on the type of service a CSP offers to other CSPs in the cloud federation. For example, a CSP may require some application with very high response time, another CSP might require a service with a high degree of confidentiality, and a third one might require a service with a high degree of availability etc. Hence, by considering these factors, the various parameters can be given suitable weights by a CSP in the cloud federation. In our prototype simulation, just to show the working of our approach, we have given equal weights to all the parameters. In real time cloud federation environment, it will vary from one CSP to another.

3.4.6 Trust Decay Factor of the CSP

In the cloud federation domain, the trust value of a CSP is considered to be dynamic and the calculated trust value decays over time. Hence, we have considered the Trust Decay Factor in our implementation, while calculating the local trust value of the requesting CSP in the federation. In our implementation, this decay factor is selected depending on when the requesting CSP had the last transaction with any other CSP in the federation. The decay factor is adjusted in such a way that the trust value gets decremented more when the date and time of the last transaction of a CSP with the requesting CSP becomes older. In our implementation, this decay factor is represented as $1/x$, where $x \in [1, 2]$, depending on the date and time of the last transaction.

Hence, the Final Local Trust Value of the requesting CSP in the federation is calculated as:

Local Trust Value=Trust Value X Trust Decay Factor.

We have selected the decay factor as $1/x$ to show the variation in the trust value of a CSP, where x depends on the time elapsed since the last transaction of the requesting CSP with any other CSP in the federation. In our prototype simulation, the parameter x takes values 1.1, 1.2, 1.4, 1.6, 1.8 and 2 corresponding to six ranges of the elapsed time since the last transaction, such as less than one month, 1-3 month(s), 3-6 months, 6-9 months, 9-12 months and greater than one year respectively. In real time implementation, the parameter x is also CSP-specific. Practically, different CSPs can use different values for x for the same time period. It also depends on how long the cloud federation has been in existence, and also how long the requesting CSP has been a member of this federation. Accordingly, a CSP in the federation can decide the value of x.

3.5 Recommended Trust Calculation of the CSP

In our proposed approach, the recommended trust calculation of the requesting CSP involves the following steps.

3.5.1 Selection of Trusted CSPs

In order to calculate the recommended trust of the requesting CSP, the trusted CSPs in the federation are identified. When a CSP gets a resource request, the CSP calculates the trust value of other CSPs in the federation to identify the trusted CSPs, and from this trusted CSPs, the feedback of the requesting CSP is collected. In order to select the trusted CSPs, the CSP calculates the trust values of other CSPs in the federation considering the parameters such as Probability of Success, History of Interaction, Existing Trust, Degree of Association and QoS Values, and these parameters are calculated as explained in the previous section. Those CSPs with trust values greater than a specific threshold are selected into the list of trusted CSPs.

3.5.2 Recommended Trust Calculation

After selecting the trusted CSPs in the federation, the CSP contacts the CSPs in the list of trusted CSPs regarding the feedback of the requesting CSP. The CSP contacts those CSPs (m out of n CSPs, where $m <= n$) and each of the m CSPs calculates its current trust value of the CSP specified, and communicates that trust value to the CSP that asked for it. The CSP then aggregates the trust values collected from the trusted CSPs to calculate the final recommended trust of the requesting CSP in the federation, and decides to grant or reject the resource request from that CSP, even if there is a QoS/SLA violation at that time.

After calculating the final recommended trust value, the CSP calculate the total trust value of the CSP as:

Total Trust Value=(Local Trust + Recommended Trust)/2.

Based on this total trust value of the requesting CSP, it takes a proper decision regarding the service request.

In our simulation, total trust value of a CSP is calculated as the average of the local trust and the recommended trust values. Local trust value is based on own experience of working with a particular CSP, and the recommended trust is based on the feedback from other trusted CSPs. In our implementation, in order to calculate the recommended trust of a CSP, initially the trusted

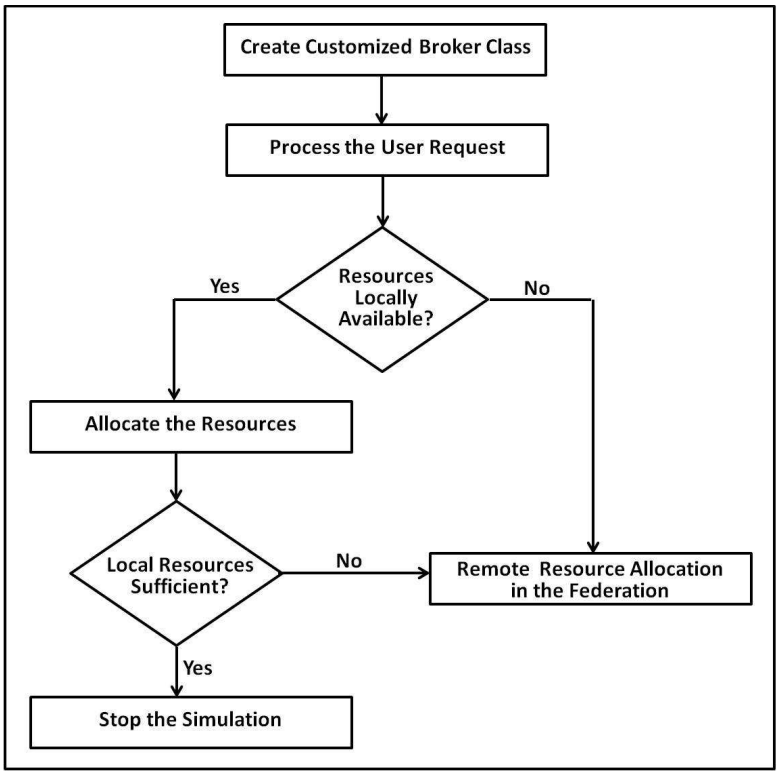

Figure 6: Local Resource Allocation in the Cloud Federation

CSPs are selected. Here also for selecting the trusted CSPs, the trust-threshold used is CSP-specific.

Generally, it can be reasonably high (0.85 in our case). Then, the feedback is collected from these CSPs. Also, any outlier in the feedback is eliminated using the algorithm proposed in [34]. Hence, this recommended trust value also assumes importance in the calculation of the total trust value. That is the reason why we have given equal weightage to local trust and the recommended trust. Again, in the real time cloud federation implementation, a CSP can use different weights such as 0.6 for the local trust value and 0.4 for the recommended trust value. In our prototype simulation, just to show the working of the proposed mechanism, we have used equal weights (0.5) for both the local trust and the recommended trust values.

4 Workflow of the Proposed Approach

The workflow of the proposed approach for the management of dynamic QoS violations in the cloud federation environment is discussed in this section.

4.1 Local Resource Allocation

The Fig. 6 shows the Local Resource Allocation process in our implementation. In our simulation, we have considered the IaaS level of resource management. The Broker class of the CloudSim [35] is extended to deal with the resource allocation process. Upon receiving a resource request from an authenticated cloud user, the CSP checks if the requested resources matching the QoS requirements of the user are locally available with the CSP. If the required resources are available at the moment, it initiates the VM allocation locally at that CSP, otherwise, if the local resources are not sufficient to meet the client requirements, the Remote Resource Allocation process in the federation is initiated.

4.2 Remote Resource Allocation

The Fig. 7 shows the remote allocation of resources in the partner CSPs of the federation, when the local resources are not sufficient to meet the current user requirements. We have assumed that there are SLAs established among the CSPs in the federation to share VMs among them. When a CSP finds that the resource request from a user cannot be met locally, it uses the Rank Table stored locally to identify the CSP(s) in the federation to ask for resources. From the Rank Table, it selects the CSP hav-

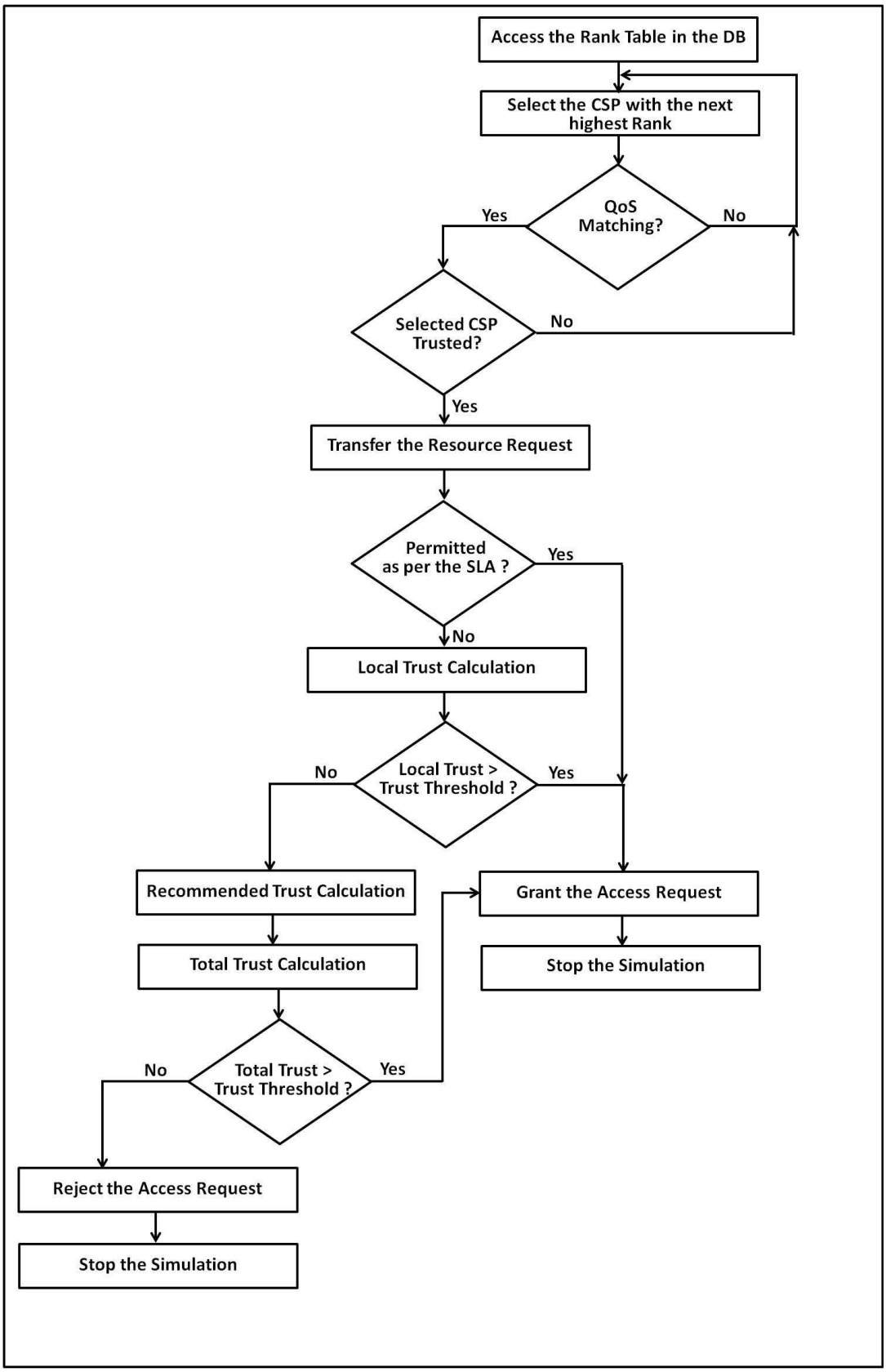

Figure 7: Remote Resource Allocation in the Cloud Federation

ing the best rank to check if the QoS requirements of the user are matching with that of the selected CSP. If it finds that the QoS details are matching, then the selected CSP's current trust value is checked to verify if the current trust value is above the threshold value set by the CSP. If the CSP finds that the selected CSP in the federation is trustworthy, it requests resources from the CSP by transferring the resource request to that CSP.

When a CSP gets the resource request from another CSP in the federation, the CSP verifies the SLA with the requesting CSP to ensure that the QoS requirements of the resource request is agreed by the SLA. If the QoS of the requested service is permitted by the SLA, the service is granted. If there is an SLA or QoS violation, then, the local trust calculation of the requesting CSP is performed. If the calculated local trust value of the requesting CSP is greater than the threshold value, then the access request is permitted. If the trust of the CSP calculated locally is less than the trust threshold, then the recommended trust of the requesting CSP is calculated as explained in the previous section. Then, the CSP calculates the total trust value of the CSP considering both the local and the recommended trust values. If the total trust value of the requesting CSP is greater than the threshold value, then the access request is permitted, in spite of the QoS violations. If the total trust value of the CSP is less than the trust threshold, the resource request of the CSP is rejected.

5 Results and Analysis

5.1 Experimental Setup

We have carried out the simulation experiments on a system with Intel (R) Core (TM) i7-3770, CPU 3.40 GHz, 8.00 GB RAM and 32-bit Operating System (Ubuntu 14.04). Softwares used for the implementation include CloudSim-3.0.3, Eclipse IDE version 3.8, MySQL Workbench Community (GPL) for Linux/Unix version 6.0.8 and Java version 1.7.0_55.

5.2 SSO in Cloud Federation

In our implementation, for securing the data in transit such as during the transfer of the identity tokens of the cloud users between CSPs and also between CSP and IdP, we have used the Symmetric Key Encryption technique using AES-256. Also, we have used Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol [32] for key exchange among the entities in the simulation. AES is a protocol mentioned in the set of standard protocols for security by the National Institute of Standards and Technology (NIST) [33] and the FHMQV protocol has its root in Diffie-Hellman (DH) protocol. The FHMQV protocol defines the Full Exponential Challenge Response (FXCR) and Full Dual exponential Challenge Response (FDCR) schemes which preserve the performance of the (H)MQV protocol, in addition to providing resistance against various attacks such as the Impersonation Attack, Man-in-the-Middle Attack etc.

5.3 Management of Dynamic QoS Violations

In order to show the working of the proposed approach, the use case shown in the Fig. 8 is considered. We assume that CSP-1 has to give the service of 50 VMs to a particular user due to some business reasons. As per the current resource availability of CSP-1, it has the capacity of offering only 6 VMs to the user as shown in the figure. Now, the CSP-1 can get the required resources form other CSPs in the federation. Hence, CSP-1 uses the QoS values offered by the CSPs in the federation to rank the CSPs.

The Fig. 9 shows the QoS values offered by the various CSPs in the Federation, as stored by the CSP-1 to which the user has made the resource request. As mentioned earlier, we have used 25 CSPs in our simulation and the various QoS features of the CSPs such as Uptime, Reliability, VM Cost, Response Time, Bandwidth Cost, Instance-Type etc. are stored in the database as shown in the figure. Assume that the QoS requirements of the current request made by the user are Uptime=99.91%, Reliability=99.95%, VM Cost=0.415$, Response Time=6 ms, Bandwidth Cost=0.005$ and the Instance-Type=large.

The CSP-1 now uses the AHP and the TOPSIS methods to rank the 24 CSPs in the federation. The Fig. 10 shows the Weight Table as calculated using the AHP method [28] which includes the different weights for the QoS parameters depending upon the user requirements. The weights are assigned to different parameters in such a way that the sum of the weights of all the parameters is one. These weight values of the QoS parameters of the user request are used in the TOPSIS method to rank the various CSPs in the federation.

The Fig. 11 shows the Rank Table generated by the CSP-1 to which the user has made the resource request.

The ranking of various CSPs in the federation is done using the TOPSIS method [29][30], and this table shows the relative preference of CSP-1 for the selection of partners in the federation, when dealing with the current resource request. In our simulated Cloud Federation environment of 25 CSPs, this Rank Table shows the ranking of 24 CSPs by the CSP-1, and this table is used for the partner selection when the CSP-1 does not have enough resources to meet the user's requirements.

The Fig. 12 shows the Trust Table maintained by the CSP-1 to which the user has made the resource request.

```
------------------------------------------------------------
QoS of User Request is matching with the QoS offered by CSP-1
Number of VMs Requested = 50
CSP-1 has a capacity of executing 6 VMs
------------------------------------------------------------
0.1: Broker: VM #0 has been created in CSP-2, Host #0
0.1: Broker: VM #1 has been created in CSP-2, Host #0
0.1: Broker: VM #2 has been created in CSP-2, Host #0
0.1: Broker: VM #3 has been created in CSP-2, Host #1
0.1: Broker: VM #4 has been created in CSP-2, Host #0
0.1: Broker: VM #5 has been created in CSP-2, Host #1
------------------------------------------------------------
```

Figure 8: User Request to CSP-1

```
------------------------------------------------------------------------------------------------------------------
QoS offered by the CSPs in the Federation
------------------------------------------------------------------------------------------------------------------
CSP_ID  Uptime(%)       Reliability(%)  VM Cost($)      Response Time(ms)       BW Cost($)      Instance Type
1       99.95           99.97           0.395                   6               0.003           large
2       99.988          99.95           0.161                   6               0.003           large
3       99.968          99.953          0.381                   2               0.003           large
4       99.935          99.962          0.084                   3               0.003           large
5       99.988          99.964          0.402                   3               0.002           large
6       99.959          99.954          0.126                   4               0.002           large
7       99.963          99.958          0.222                   6               0.001           large
8       99.939          99.971          0.332                   7               0.004           large
9       99.918          99.975          0.253                   2               0.002           large
10      99.995          99.99           0.308                   7               0.006           large
11      99.958          99.956          0.242                   3               0.005           large
12      99.945          99.971          0.228                   7               0.003           large
13      99.981          99.976          0.067                   7               0.005           large
14      99.911          99.987          0.15                    3               0.004           large
15      99.924          99.983          0.117                   5               0.003           large
16      99.912          99.98           0.248                   5               0.002           large
17      99.948          99.973          0.227                   3               0.003           large
18      99.952          99.967          0.319                   5               0.004           large
19      99.999          99.962          0.239                   2               0.002           large
20      99.944          99.975          0.337                   3               0.002           large
21      99.943          99.972          0.207                   5               0.003           large
22      99.987          99.957          0.196                   5               0.003           large
23      99.959          99.977          0.272                   2               0.003           large
24      99.97           99.952          0.204                   4               0.003           large
25      99.999          99.992          0.254                   5               0.003           large
------------------------------------------------------------------------------------------------------------------
```

Figure 9: QoS offered by the CSPs in the federation

```
------------------------------------------------
Weight Table (Calculated Using the AHP Method)
------------------------------------------------
QoS-Parameter    Weight

Uptime          0.265
Reliability     0.227
VM Cost         0.154
Response-Time   0.191
BW Cost         0.163
------------------------------------------------
```

Figure 10: Weight Table

Rank Table of CSP-1 (Calculated using the TOPSIS Method)

CSP_ID	Calculated Value	Rank
19	0.701	1
4	0.696	2
9	0.693	3
6	0.686	4
23	0.646	5
17	0.641	6
14	0.629	7
20	0.624	8
24	0.62	9
15	0.618	10
7	0.608	11
16	0.604	12
3	0.597	13
5	0.595	14
22	0.592	15
21	0.588	16
2	0.573	17
25	0.572	18
11	0.56	19
12	0.527	20
18	0.519	21
13	0.517	22
8	0.474	23
10	0.437	24

Figure 11: Rank Table

Trust Table of CSP-1

CSP_ID	Trust Value
2	0.519
3	0.544
4	0.684
5	0.663
6	0.572
7	0.806
8	0.783
9	0.452
10	0.48
11	0.571
12	0.588
13	0.599
14	0.483
15	0.455
16	0.845
17	0.828
18	0.501
19	0.534
20	0.635
21	0.513
22	0.58
23	0.577
24	0.504
25	0.839

Figure 12: Trust Table of CSP-1

```
--------------------------------------------------------------------------------
Selection of CSP from Rank Table
--------------------------------------------------------------------------------
CSP_ID          Status          Reason
19              Not Selected    Trust Value less than Trust Threshold(0.65)
4               Selected        QoS and Trust Parameters(0.65) matched
--------------------------------------------------------------------------------
CSP selected from the Federation for meeting the user requirements is CSP-4
The Request for 44 VMs is transferred to the CSP-4
--------------------------------------------------------------------------------
As per the SLA between CSP-1 and CSP-4 ,the Number of VMs agreed is '30';
but the current requirement is '44'.
Also the Instance Type agreed is 'small'; but the current requirement is 'large'.
```

Figure 13: Selection of CSPs in the Cloud Federation

```
####################################
Calculation of Local Trust of the CSPs
####################################
1. Probablilty of Success = 0.916
2. History of Interaction = 0.832
3. Existing Trust         = 0.855
4. Degree of Association  = 1.0
5. QoS Value              = 0.859

Local Trust               = 0.892
Trust Decay Factor        = 0.625
Final Local Trust         = 0.558

Local Trust < Trust Threshold (0.6)
```

Figure 14: Calculation of Local Trust

This table shows the local trust value of every other CSP in the federation as calculated by the CSP-1. The trust values of the CSPs are calculated considering the parameters Probability of Success, History of Interaction, Existing Trust, Degree of Association and QoS Values as explained in the Section 3.4. Every CSP in the federation is assigned a trust value between 0 and 1 which shows how trustworthy that particular CSP is to the CSP-1.

The Fig. 13 shows the selection of CSPs in the cloud federation by the CSP-1 in order to meet the resource requirements of the user. The selection process considers the Rank Table (Fig. 11) and the Trust Table (Fig. 12) created by the CSP-1. As shown in the Fig. 13, even though CSP-19 is having the highest rank, this CSP is not selected because the trust value of this CSP (0.534) is less than the trust threshold (0.65) by the CSP-1. Hence the CSP with the next highest rank (CSP-4) is selected from the Rank Table. Now, CSP-1 verifies that the QoS requirements of the user and the QoS features offered by the CSP-4 match, and also the trust value of CSP-4 (0.684) is above the trust threshold maintained by the CSP-1 (0.65). Hence, CSP-4 is selected for meeting the resource requirements of the user, and the request for 44 VMs is transferred to CSP-4. Now, the CSP-4 checks the SLA agreed between CSP-4 and CSP-1 in the federation.

As shown in the Fig. 13, as per the SLA, the number of VMs agreed between them is 30; but the current requirement is for 44 VMs. Also, the instance type of the VMs agreed between them is small; but the current requirement is for 'large'. Upon receiving the resource request from the CSP-1, as per the proposed approach, in order to deal with this resource request of 44 VMs from CSP-1, CSP-4 calculates the trust value of CSP-1 in the federation so that SLA renegotiation is avoided at that time, and the resource request may be accepted. Firstly, the CSP-4 calculates the local trust value of CSP-1.

The Fig. 14 shows the calculation of the local trust of the CSP-1 by CSP-4. As explained before, for calculating the local trust of the CSP-1, the trust parameters such as Probability of Success, Degree of Association, History of Interaction, Existing Trust and the QoS values are considered. As shown in the figure, the values of these parameters for the local trust calculation of CSP-1 are 0.916, 1.0, 0.832, 0.855, and 0.859 respectively. To calculate the local trust of the CSP-1, the average value of all the above parameters is taken and it is found to be 0.892 as shown in the figure. The Trust Decay Factor for CSP-1 is calculated as 0.625, and hence the final local trust of CSP-1 is 0.558. But, this local trust is less than the trust threshold maintained by the CSP-4 (0.6) for granting the resource request.

Hence, the CSP-4 calculates the recommended trust of the CSP-1. The recommended trust is calculated by taking feedback from the set of trusted CSPs of CSP-4.

```
########################################
Calculation of Recommended Trust of the CSPs
########################################
------------------------------------------
Trust Table
------------------------------------------
CSP_ID          Trust Value
  2               0.804
  3               0.894
  5               0.86
  6               0.802
  7               0.87
  8               0.763
  9               0.799
  10              0.815
  11              0.859
  12              0.872
  13              0.86
  14              0.805
  15              0.778
  16              0.862
  17              0.799
  18              0.81
  19              0.79
  20              0.884
  21              0.809
  22              0.881
  23              0.87
  24              0.776
  25              0.875
```

Figure 15: Trust Table of CSP-4

Hence, the CSP-4 calculates the trust values of all the relevant CSPs in the federation considering the parameters such as Probability of Success, Degree of Association, History of Interaction, Existing Trust and the QoS values. The trust table generated by the CSP-4 upon receiving the access request from the CSP-1 is shown in the Fig. 15. From this trust table, the set of CSPs having trust value greater than a predefined trust threshold (0.85) is identified. This table is shown in the Fig. 16 as Trusted CSPs. In our case, the number of CSPs having trust value greater than the threshold is 11. These CSPs are asked for recommendation of the CSP-1.

The Fig. 17 shows the Recommendation Table of the CSP-4, and as shown in the figure, the number of CSPs responded with the trust values of CSP-1 is 8. The other three CSPs may not have the history of interaction with the CSP-1 to calculate its trust vale as required by the CSP-4. The table shows the trust values of various CSPs who have responded with the required recommendation, and their corresponding returned trust value of the CSP-1. Now, the trust value of the responded CSPs and their returned trust values are multiplied to get the recommended trust values of CSP-1. In our work, in order to filter the recommendation values given by the CSPs, we have implemented the outlier filtering algorithm proposed by Azzedin et al. in [34]. Hence the resulting filtered recommendation table is shown in the Fig. 18. In the filtered recommendation table, the number of recommendations considered is 5, and the 3 recommendations are filtered out.

From the filtered recommendation table, the total recommended trust is calculated as the average of the recommended trust of the filtered CSPs, and in our case, the total recommended trust is calculated as 0.754 as shown in the Fig. 18. Now, the total trust value is calculated and it is found to be 0.656. Since this trust value is greater than the trust threshold (0.6), the VM request from CSP-1 is accepted by the CSP-4, even though there is a QoS violation between them (as shown in the Fig. 13).

5.4 Results and Analysis

In order to test and validate the proposed approach in the Cloud Federation environment, we have implemented the Cloud Federation of 25 CSPs using the CloudSim toolkit [35]. Sample database is created and used as the database for testing our algorithm. We have considered the resource request in such a way that there is chance for QoS violation between the CSPs so that the proposed approach can be used to deal with the dynamic QoS violations. The Fig. 19 shows the number of resource requests of CSP-1 accepted/rejected in the cloud federation

```
CSP_Trust_Threshold = 0.85
------------------------------------------------------------
Trusted CSPs
------------------------------------------------------------
CSP_ID          Trust Value
  3               0.894
  5               0.86
  7               0.87
  11              0.859
  12              0.872
  13              0.86
  16              0.862
  20              0.884
  22              0.881
  23              0.87
  25              0.875

Total number of CSPs above the CSP_Trust_Threshold : 11
------------------------------------------------------------
```

Figure 16: Trusted CSPs

```
---------------------------------------------------------------------
Recommendation Table
---------------------------------------------------------------------
CSP_ID  Trust Value     Returned Trust Value    Recommmended Trust
3       0.894                   0.836           0.747
5       0.86                    0.773           0.665
7       0.87                    0.881           0.766
11      0.859                   0.85            0.73
12      0.872                   0.788           0.687
16      0.862                   0.763           0.658
22      0.881                   0.885           0.78
25      0.875                   0.856           0.749

---------------------------------------------------------------------
Total number of CSPs responded : 8
```

Figure 17: Recommendation Table

```
------------------------------------------------
Filtered Recommendation Table
------------------------------------------------
CSP_ID          Recommmended Trust
3               0.747
7               0.766
11              0.73
22              0.78
25              0.749

The number of Filtered Recommendations is 5

Total Recommended Trust = 0.754
Total Trust Value       = 0.656

Total Trust value > Trust Threshold (0.6)
VM Request from CSP-1 is accepted by CSP-4
```

Figure 18: Filtered Recommendation Table

environment.

The figure shows three cases: the first one indicates the total number of times the resource requests of CSP-1 is accepted considering the local trust of CSP-1 alone. Second case shows the total number of times the local trust value of CSP-1 alone was not sufficient, and hence the CSPs had to calculate the recommended trust of CSP-1, and the total trust was sufficient to accept the resource requests of CSP-1. The third case shows the total number of times the resource requests of CSP-1 were rejected as the total trust value was less than the trust threshold maintained in the system. As shown in the figure, out of 100 requests from CSP-1, 28 times the service requests were accepted using local trust, and 42 times using recommended trust of the CSP-1. 'Insufficient Trust' means the case when the service request is rejected even with the recommended trust. Hence, in our simulation 30 times the requests got rejected. From the figure, it is seen that reputation of the CSP plays an important role in the Cloud Federation. As compared to local trust, recommended trust also plays an important role in solving the dynamic QoS violations, and thereby accepting the resource requests from CSP-1.

The Fig. 20 shows the average time taken for the service decision by a CSP when it gets the resource requests from CSP-1 in the federation, and also when there is a QoS violation between the CSPs. The figure shows the average time taken in two cases of service decision, considering 100 service requests. The first one shows the average time taken considering the local trust of the requesting CSP-1 alone. The second case shows the average time taken for the service decision, considering the local and the recommended trust of CSP-1. As shown in the figure, the average time taken for the service decision using local trust is 8989 ms and using the recommended trust is 9169 ms. Even though, calculation of the recommended trust takes longer compared to the calculation of the local trust alone, the performance of the cloud federation is improved in such a way that more user requests are satisfied. Hence, the economic benefits and the reputation of the partner CSPs in the cloud federation are improved.

In the real time cloud federation environments, SLA renegotiation between two parties (CSPs) involves the following steps. The user (CSP-A in the cloud federation) submits the resource request specifying the QoS parameters required or to be changed, to other CSP (CSP-B) in the federation. The CSP-B then proposes the initial offer based on its current availability and service features to fulfill the service request submitted by CSP-A. On receiving the initial offer, the requesting CSP-A can prepare the counter offer (if needed) which is sent to the CSP-B. The CSP-B then evaluates the counter offer (proposal). If the counter offer cannot be accepted, then that CSP proposes another counter offer. Finally, the negotiation or renegotiation process is terminated by the CSPs upon reaching mutual agreements regarding the services and QoS, or when there are no mutual agreements reached between the parties. If mutual agreements are reached, then the SLA is created using the templates, and it is signed by the parties. Thus, it becomes the modified SLA after the renegotiation process. Since the renegotiation involves several steps or processes as explained, it is expected to take longer than 10 seconds. Hence the time taken in our prototype simulation is considered to be better.

5.5 Pros and Cons of the Approach

The major advantage of the proposed approach of partner selection in the cloud federation environment is that it optimizes the search for partners in the cloud fedration. It helps a CSP in the federation to identify the suitable partner when it is running out of resources, to offload the resource requests of the clients. The SSO approach implemeneted in the cloud federation is both secure and efficient as we have used AES-256 algorithm and the FHMQV protocol. The proposed trust based approach helps to solve the dynamic QoS violations in the cloud federation environment without requiring the SLA renegotiation at run time. Thus, the approach improves the performance, responsiveness, efficiency of the CSPs, and thereby the reputaion and the profits of the CSPs in the federation.

In the proposed trust-based approach, we consider the trusted CSPs of any CSP to get the recommendetaion of any other CSP in the federation. Here, we have assumed that the trusted CSPs of any CSP in the federation have a good transaction history with the specific CSP. Also, we have used the specified filtering algorithm to remove any outlier(s) among the recommended values. Here, unlike the stand-alone CSPs, the cloud federation is existing in a cooperative and mutually-benefitting manner, and hence, it is assumed that majority of the trusted CSPs of any CSP won't behave maliciously. Thus, our approach helps to meet the clients' requirements of a CSP during emergency situations, without requiring the SLA renegotiation, ensuring timely and efficient service to the clients.

As far as we know, this is the first work that employs the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. Since there are no similar works available that deals with the management of dynamic QoS violations in the cloud federation domain, we were not able to compare our approach with other approaches.

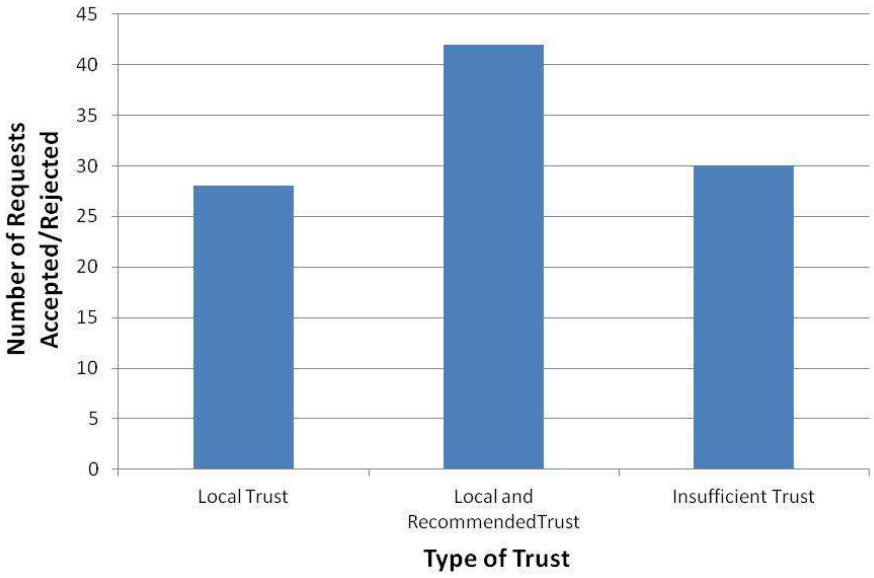

Figure 19: Analysis of the Accepted Requests in the Federation

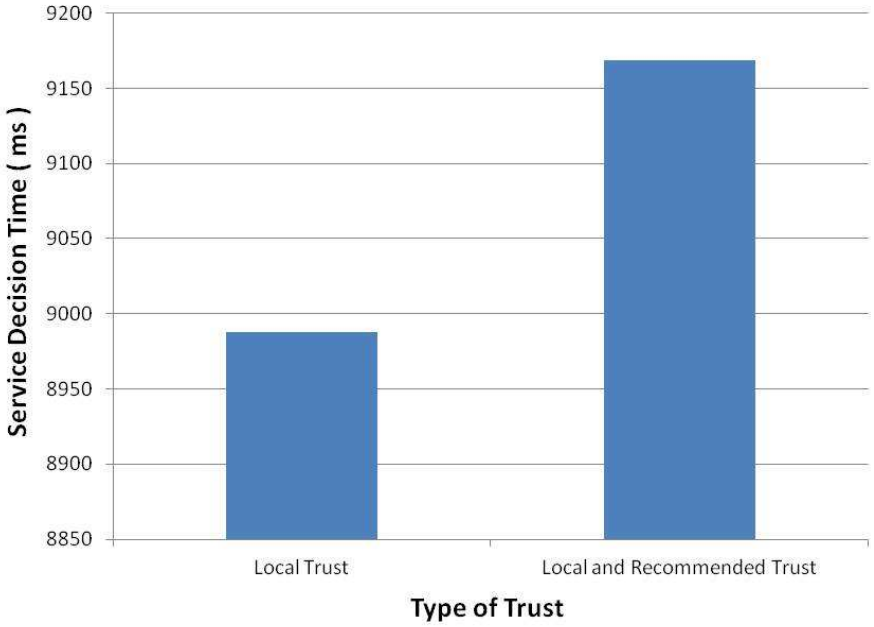

Figure 20: Analysis of the Service Decision Time

6 SUMMARY AND CONCLUSIONS

In this paper, we have implemented the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. We have also implemented the partner selection approach for a CSP when it does not have enough resources to meet the resource requirements of its users using the AHP and the TOPSIS methods. Also, this paper talks about the implementation of the SSO approach in the cloud federation environment using the AES-256 algorithm and the FHMQV protocol. The proposed trust based approach shows that by calculating the local trust and the recommended trust of the CSPs, the dynamic QoS violations can be effectively solved. The proposed approach was validated using the CloudSim toolkit. The analysis of the results obtained shows the effectiveness of the proposed approach. In our implementation, we have used the sample data base created for testing the approach. As a future work, we plan to implement the proposed approach in an Opennebula cloud environment using real time test data.

REFERENCES

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services", *Algorithms and architectures for parallel processing*, Springer Berlin Heidelberg, pp.13-31, 2010.

[2] F. Azzedin and M. Maheswaran, "Towards trust-aware resource management in grid computing systems", in *Proc. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, Washington, USA, 2002, pp. 452-457.

[3] V. Vijayakumar, R. S. D. Wahida Banu, and Jemal H. Abawajy, "An efficient approach based on trust and reputation for secured selection of grid resources", *International journal of parallel, emergent and distributed systems*, vol. 27, no. 1, pp. 1-17, 2012.

[4] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision support systems*, vol. 43, no. 2, pp. 618-644, 2007.

[5] S. M. Habib, S. Ries, and M. Muhlhauser, "Cloud computing landscape and research challenges regarding trust and reputation", in *Proc. 7th IEEE International Conference on Ubiquitous Intelligence & Computing and Autonomic & Trusted Computing (UIC/ATC)*, 2010, pp. 410-415.

[6] J. Abawajy, "Establishing trust in hybrid cloud computing environments", in *Proc. 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 118-125.

[7] V. Vijayakumar and R. S. D. W. Banu, "Security for resource selection in grid computing based on trust and reputation responsiveness", *International Journal of Computer Science and Network Security*, vol. 8, no. 11, pp. 107-115, 2008.

[8] B. B. Govil, K. Thyagarajan, K. Srinivasan, V. K. Chaurasiya, and S. Das, "An approach to identify the optimal cloud in cloud federation", *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol. 1, no. 1, pp. 35-44, 2012.

[9] R. Sanchez, F. Almenares, P. Arias, D. Daz-Snchez, and A. Marn, "Enhancing privacy and dynamic federation in IdM for consumer cloud computing", *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 95-103, 2012.

[10] W. Li and L. Ping, "Trust model to enhance security and interoperability of cloud environment", *Cloud Computing*, Springer, pp. 69-79, 2009.

[11] M. Ahmed and Y. Xiang, "Trust ticket deployment: a notion of a data owner's trust in cloud computing", in *Proc. 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 111-117.

[12] I. Goiri, J. Guitart, and J. Torres, "Economic model of a Cloud provider operating in a federated Cloud", *Information Systems Frontiers*, vol. 14, no. 4, pp. 827-843, 2012.

[13] C. S. Wu and I. Khoury, "QoS-aware dynamic research component composition for collaborative research projects in the clouds", in *Proc. 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2012, pp. 883-888.

[14] M. M. Hassan and E. N. Huh, "Resource Management for Data Intensive Clouds Through Dynamic Federation: A Game Theoretic Approach", *Handbook of Data Intensive Computing*, Springer, New York, pp. 169-188, 2011.

[15] M. M. Hassan, B. Song, S. M. Han, E. N. Huh, C. Yoon, and W. Ryu, "Multi-objective optimization model for partner selection in a market-oriented dynamic collaborative cloud service platform", in

Proc. 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI'09), 2009, pp. 637-644.

[16] A. Kertsz, G. Kecskemti, A. Marosi, M. Oriol, X. Franch, and J. Marco, "Integrated monitoring approach for seamless service provisioning in federated clouds", in *Proc. 20th IEEE Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2012, pp. 567-574.

[17] Y. Chen, B. Khoussainov, and X. Ye, "A Game Theoretic Approach to Service Discovery and Selection", in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 4072-4079.

[18] M. Stihler, A. O. Santin, A. L. Marcon Jr, and J. D. S. Fraga, "Integral federated identity management for cloud computing", in *Proc. 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1-5.

[19] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation", in *Proc. 3rd International Conference on Cloud Computing (CLOUD)*, 2010, pp. 337-345.

[20] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Security and cloud computing: Intercloud identity management infrastructure", in *Proc. 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*, 2010, pp. 263-265.

[21] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Threephase cross-cloud federation model: The cloud SSO authentication", in *Proc. Second International Conference on Advances in Future Internet (AFIN)*, 2010, pp. 94-101.

[22] F. Tusa, A. Celesti, M. Paone, M. Villari, and A. Puliafito, "How clever-based clouds conceive horizontal and vertical federations", in *Proc. IEEE Symposium on Computers and Communications (ISCC)*, 2011, pp. 167-172.

[23] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the intercloud - protocols and formats for cloud computing interoperability", in *Proc. Fourth International Conference on Internet and Web Applications and Services (ICIW '09)*, 2009, pp. 328-336.

[24] D. Bernstein, D. Vij, and S. Diamond, "An intercloud cloud computing economy-technology, governance, and market blueprints", in *Proc. SRII Global Conference (SRII)*, 2011, pp. 293-299.

[25] D. Bernstein and D. Vij, "Intercloud Exchanges and Roots Topology and Trust Blueprint", in *Proc. 11th International Conference on Internet Computing*, 2011, pp. 135-141.

[26] I. Goiri, J. Guitart, and J. Torres, "Characterizing cloud federation for enhancing providers' profit", in *Proc. 3rd IEEE International Conference on Cloud Computing (CLOUD)*, 2010, pp. 123-130.

[27] P. Armstrong, A. Agarwal, A. Bishop, A. Charbonneau, R. Desmarais, K. Fransham, N. Hill, I. Gable, S. Gaudet, S. Goliath, R. Impey, C. Leavett-Brown, J. Ouellete, M. Paterson, C. Pritchet, D. Penfold-Brown, W. Podaima, D. Schade, and J. Sobie, "Cloud scheduler: a resource manager for distributed compute clouds", *CoRR*, abs/1007.0050, 2010.

[28] E. Triantaphyllou and S. H. Mann, "Using the analytic hierarchy process for decision making in engineering applications: some challenges", *International Journal of Industrial Engineering: Applications and Practice*, vol. 2, no. 1, pp. 35-44, 1995.

[29] K. P. Yoon and C. L. Hwang, *Multiple attribute decision making: an introduction*, Sage Publications, Vol. 104, 1995.

[30] C. L. Hwang, Y. J. Lai, and T. Y. Liu, "A new approach for multiple objective decision making", *Computers & operations research*, vol. 20, no. 8, pp. 889-899, 1993.

[31] C. C. Chang and C. Y. Lee, "A secure single sign-on mechanism for distributed computer networks", *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 629-637, 2012.

[32] A. P. Sarr, P. Elbaz-Vincent, and J. C. Bajard, "A secure and efficient authenticated diffie-hellman protocol", *Public Key Infrastructures, Services and Applications*, Springer Berlin Heidelberg, pp. 83-98, 2010.

[33] P. J. Linstrom and W. G. Mallard, "National Institute of Standards and Technology", http://www.nist.gov/, accessed 27th May 2015.

[34] F. Azzedin and A. Ridha, "Feedback behavior and its role in trust assessment for peer-to-peer systems", *Telecommunication Systems*, vol. 44, no. 3-4, pp. 253-266, 2010.

[35] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.

AUTHOR BIOGRAPHIES

Manoj V. Thomas received his B.Tech degree in Computer Science and Engineering from Govt. Engg. College, Kottayam, Kerala in 2003, and M.Tech degree from NITK, Surathkal as a gold medallist, in 2008. He has more than 10 years of teaching experience, and his research interests include computer networks, cloud computing, and cloud security. He is a life member of Computer Society of India (CSI), and Indian Society for Technical Education (ISTE). He is currently a research student in the department of CSE at NITK, Surathkal, India.

Dr. K. Chandra Sekaran is currently Professor in the Department of Computer Science and Engineering, National Institute of Technology Karnataka, having 26 years of experience. He has more than 120 research papers published by various reputed peer-reviewed International Journals and Conferences. He has received best paper awards and best teacher awards. He serves as a member of various reputed societies including IEEE (Senior member), ACM (Senior Member), CSI, ISTE and Association of British Scholars (ABS). He is also a member in IEEE Computer Society's Cloud Computing STC (Special Technical Community). His areas of interest include Computer Networks, Distributed Computing (includes Cloud Computing and Security) and Business Computing and Information Systems Management.