ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2015, T. 44, Nr. 1

An Efficient Resource Allocation Scheme for Cloud Federations

Kuo-Hui Yeh

Department of Information Management, National Dong Hwa University,
Hualien 970, Taiwan R.O.C.
e-mail: khyeh@mail.ndhu.edu.tw

Nai-Wei Lo

Department of Information Management, National Taiwan University of Science and Technology,
Taipei, Taiwan R.O.C.
e-mail: nwlo@cs.ntust.edu.tw

Pei-Yun Liu

Department of Information Management, National Taiwan University of Science and Technology,
Taipei, Taiwan R.O.C.
e-mail: m10009105@mail.ntust.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.44.1.6875

Abstract. With the evolution of Internet technology, cloud computing technologies are applied to many novel applications in modern societies. In particular, most large enterprises intend to reduce management cost and improve productivity via cloud computing technologies. In order to provide uninterrupted services to client customers and reduce the maintenance cost of cloud services, how to dynamically and efficiently allocate precious resources among individual clouds has become a critical issue. In this study, the issue of dynamic allocation on computing resources across multiple cloud environments is considered. We proposed an efficient computing resource allocation mechanism based on the inter-trust relationship model, which allows one cloud to borrow extra computing resources from other clouds via cloud federation architecture when it is necessary. Simulation experiments are conducted and the results show the practicability and feasibility of our proposed mechanism in cloud federation environments.

Keywords: Cloud Computing; Resource Allocation; Cloud Federation.

1. Introduction

The rapid growth and development of cloud computing technologies brings a new level of efficiency to services offering in Internet. It is also a huge technology invention for IT industry to do all computing works within cloud environments. According to NIST [16], cloud computing is defined as a model which provides access to a shared pool of scalable resources (e.g., servers, storage, applications, and services) over the network. The cloud model is composed of five essential characteristics, three service models, and four deployment models. The five essential characteristics of cloud computing are: (a) on-demand self-service; (b) broad network access; (c) adjusted resource pooling; (d) rapid elasticity; and (e) measured service. The cloud service model [4] can be divided into three fundamental ones: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Hundreds of independent and heterogeneous cloud service providers have built their services based on these three models. Finally, there are four deployment models for a cloud environment; i.e. private cloud, public cloud, hybrid cloud and community cloud.

Academic communities and cloud environment operators have predicted that cloud infrastructure will transform toward interoperable federated intra-cloud (or inter-cloud) environments in the near future [4]. Celesti et al. [5] suggested that the future evolution of cloud computing could be divided into three stages: (1) monolithic; (2) vertical supply chain; and (3) horizontal

federation (i.e. cloud federation). In the final (third) stage, small, medium, and large cloud providers will federate horizontally in order to gain large scale capacity on resources. In a cloud federation architecture, there exists so-called Identity Providers (IdP) which act as authoritative authenticators, and service providers which provide web services [2]. In addition, Single Sign-On (SSO) technology has been introduced in recent years, which allows a user to access applications within different enterprises without being prompted to log in to each enterprise domain individually. Through federated SSO, trust relationship to a specific client customer among multiple clouds is enabled.

Even though a cloud environment with multiple data centers may have hundreds of physical machines (or thousands of virtual machines), resources are still limited in each data center. When a large amount of services are requested, a single data center probably cannot provide all required resources to clients. However, in order to provide good quality of service and fulfill Service Level Agreements (SLAs) between client customers and cloud service providers, it is very important to satisfy clients requests and avoid service interruptions [3]. As a result, resource allocation and management in cloud environments has become one of the most important issues. Several efforts [1–3, 21, 22] have been dedicated to this promising and interesting research area, where the problem of optimal resource provisioning at application level (or infrastructure level) is investigated.

Nowadays, it is important to reduce management costs and improve resource utilization among cloud vendors or IT enterprises. In this study, we propose an efficient resource allocation mechanism based on inter-trust relationships among clouds, in which each cloud can borrow computing resources from other clouds. Resource allocation cross individual clouds will occur when available resources of one cloud cannot satisfy resource requests from client customers. The resource provisioning process cross multiple clouds is based on the utilization of a common IdP among different clouds. Therefore, each IdP's performance (i.e., IdP reliability) is also required to be considered and evaluated in the proposed resource allocation mechanism.

2. Problem Formulation

It is well known that virtualized system architecture is better than non-virtualized system architecture in terms of resources sharing (e.g., CPU, RAM, disk space and network bandwidth). In recent years, more and more businesses have considered moving their existing applications and building new applications in cloud environments. The goal can be achieved by either creating their own private cloud environments or renting cloud resources from a cloud service provider. Famous cloud resources providers include Amazon AWS, Microsoft Azure and Google App Engine. These cloud resources can be acquired and used based on a pay-per-use or charge-per-use basis. In order to provide uninterrupted services to client customers and reduce the maintenance cost of cloud services, how to dynamically allocate precious resources efficiently among individual clouds has become a critical issue. In this study, how to support dynamic allocation on computing resources across multiple cloud environments is considered.

Because of business-competitive nature among enterprises, most enterprises are not willing to share their cloud resources with other enterprises. For example, Microsoft will not share their cloud resources with Google, its potential competitor. However, an enterprise which consists of business groups or companies around the world may have multiple private clouds built by different business groups. This enterprise will rent cloud resources from existing cloud resource providers for their business groups. From time to time, some business units might face the problem that there is no enough cloud resources from their local private clouds to support their business operations at some peak business-processing time period. To avoid such situations, business groups will need to borrow idle resources in other clouds. The utilization of several common IdP among different private clouds is assumed. The parameters, such as resources requested for each service work, current available resources in the cloud, trusted-IdP list for the cloud and network transmission cost between clouds, will be kept in each private cloud. In addition, a cloud collaboration list will be maintained in each IdP server and a public trusted third party will maintain the values of each IdP's reliability. Each IdP will release its successful authentication rate periodically, and an organization or company can collect those data periodically and derive the reliability values of all IdPs.

3. Related Work

In order to accurately depict the research presented in this paper, the concepts of trust relationship and identity provider, and cloud federation and resource allocation will be reviewed in this section.

3.1. Trust and Identity Provider

There is a sentence from a cartoon by Peter Steiner which reads, "On the Internet nobody knows you're a dog" [20]. This means that nobody actually knows who you are un-less you prove it to them. Therefore, there is a need for an entity, i.e. an IdP, which is a trusted provider that creates, maintains, and manages identity information for users, services, or systems, and provides identity authentication to other service providers or applications within a cloud federation (or distributed network). IdPs authenticate users and issue security tokens possessing not only the user's ID but also other identity properties of the user's claims. In the real world, some examples of IdPs are Facebook, Google Account, Salesforce.com, Windows Live ID, along with many other similarly structures enterprises.

Thus, an IdP plays an important role in the cloud federation framework.

Trust is based on people's interactions and on how much information people are willing to reveal. Trust can also rely on someone acting as an intermediary who is a trusted, but independent, third party. According to McKnight et al. [15], a general concept of trust is defined as "the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though negative consequences are possible. "In 2009, Kylau et al. [11] highlighted that trust contains three fundamental aspects:(1) the dependence on the trusted party; (2) the reliability of the trusted party; and (3) the consequences in case the trusted party does not perform as expected. In general, trust relationships are usually established by a set of contracts defining obligations and rights and each party has its policies. A trusting relationship is divided into two main categories: (1) direct trust and (2) indirect or transitive trust. The direct trust relationship is created by the actions of the two parties without relying on any other third party, like the typical behaviour of befriending between two human beings.

In order to enable toe SSO setting on the Internet, there needs to be at least one entity playing the role of IdP. The SSO Federation authentication [15] can take place under the following three main situations: (1) large enterprise with several separate business units; (2) between an enterprise and their business partners or customers; (3) between an enterprise and outsourced providers. For example, after a business partner's employee logs on to the enterprise system, the SSO system from a business partner will provide a security assertion token through a protocol, such as OpenID, SAML (Security Assertions Markup Language), Liberty Alliance, WS Federation, Shibboleth, INames. It then allows user to access multiple applications in the enterprise system without logging on again for each application. Furthermore, with expected the future development of cloud computing, many academic researchers have investigated the field of cloud federation. In 2010, Celesti et al. [4] proposed a three-phase cloud federation process in which home clouds, foreign clouds, and single layer IdPs are involved. The authors demonstrated three main procedures, i.e. discovery, matchmaking, and authentication, to achieve resource borrowing among clouds. In addition, the authors presented a SAML profile named Cross-Cloud Authentication Agent SSO (CCAA-SSO) which defines the steps of a cloud based SSO authentication. Later, Li and Ping [12] investigated trust models for the distribution environment and presented a domain-based trust model to solve security issues of cross-cloud architecture. Their model allows cloud customers to choose different providers' services and resources in heterogeneous domains. Following this, Pearson et al. [18] introduced a privacy manager to prevent the cloud users' private data from being stolen or misused. In addition, their proposed method could assist the cloud provider in conforming to privacy laws.

In 2011, Celesti et al. [6] investigated the technique of delegated authentication of the distributed infrastructure involved with an IdP and a Service Provider (SP). The authors evaluated its possible utilization in a federated cloud scenario. To minimize the cost on the IT infrastructure, Malik et al. [14] proposed a model to utilize already-virtualized infrastructure in which cloud vendors could offer low-cost cloud services by acquiring underutilized resources from third party enterprises. Celesti et al. [7] then developed a SAML based SSO authentication profile using a third party IdP for a three-tier cloud architectures. Their proposed method could be applied in different CLEVER-based clouds for the establishment of trusted inter-domain communications. Later, the Celesti et al. [4] introduced the architecture for federation establishment by renting extra physical resources from various federated clouds. In addition to this, a technique based on the IdP/SP based model along with the SAML technology was proposed.

3.2. Cloud Federation and Resource Allocation

In a cloud federation architecture, the local cloud and external cloud refers to a cloud of clouds. The local cloud is a cloud provider which does not possess resources to provide services for the cloud client. In a resource borrowing process, a local cloud will send a request signal to an external cloud to ask for resources. The external cloud is a cloud provider which owns idle resources (e.g., CPU, RAM, storage) of its virtualization infrastructure, and the virtual resources can be lent or rented to local clouds.IdP is a trusted third party that provides an identity for authentication services. Here, the cloud federation acts as a unionization infrastructure composed of multiple clouds that can be accessed by other clouds via the Internet. Importantly, federations are not isolated structures; clouds of one federation might also be part of another one.

Resource allocation issues have been addressed in the field of computing (e.g., grid computing, operating systems, and datacenter management). The goal of resource allocation mechanisms is to ensure that the provider's infrastructure can reliably satisfy an application's requirements. In addition, to efficiently provide resources for service provider's services, and to minimize the operational costs of the cloud environment, the current status of resources in the cloud environment should be considered in the resource management mechanisms. Normally, resources are shared by multiple clients and located in a data center so that the client may see an unlimited resource. However, these clients do not know where (and how) the resource is stored. Furthermore, resources should be dynamically assigned and adjusted on demand and related parameters should be set properly during the resource allocation phases. Thus, when allocating resources for incoming service request, an important point is how the resources are modeled without wasting available resources.

Based on the above two concepts, in 2009 You et al. [23] focused on efficient resource allocation at the physical level of cloud computing. Their method, however, only considered CPU resources based on market economy theory. Celesti et al. [4, 5] later proposed a three-phase cloud federation process to consider the scenario that a home cloud might borrow resources from external clouds with one layer of IdPs. Following these, the authors of [3, 8] focused on an SLA-oriented and Quality of Service (QoS) resource allocation in the cloud computing system. Moreover, Mochizuki et al. [17] and Guazzone et al. [10] both achieved efficient resource management via reduction of energy/electric power cost. In 2012, Apostol et al. [1] presented a new provisioning mechanism for cloud systems, and addressed the key requirements for resource management at the infrastructure level. Furthermore, Wang et al. [22] proposed a threshold-based dynamic resource allocation scheme for cloud computing in the application level which could provision virtual resources dynamically among the cloud computing applications.

4. The Proposed Resource Management Framework

In context of the overall framework design, the concept of cross-IdP will first be introduced. We will then illustrate the resource allocation framework for a cloud federation. Finally, we present a flowchart of the proposed resource allocation scenario based on the concept of a trusted IdP.

4.1. Create a Federation with the Concept of Cross-IdP

An IdP is a system that manages user's identities, and provides an authentication service for client applications on the Internet. In other words, once users intend to invoke an authentication service, the IdP is a trusted third party which can be relied upon by users. The IdP sends an attribute assertion containing trusted information about the user to an SP. Note that an SP is an application that relies on the claims issued by an IdP to authorize a user, and to release appropriate access to the user. Thus, an IdP is a bridge connecting users and services provider. Instead of only one IdP with SSO authentication [4], we consider a cloud federation set through multiple IdPs. This provides more flexibility and is scalable allowing the sharing of resources among clouds and IdPs. In addition, as there exists numerous IdPs for business units to choose from, we assume that each IdP has a specific reliability for a users' preference. The trust index (i.e. R_x) of each IdP can be defined according to the IdP's previous work stability, such as frequency of server crash/shut down, and security, such as system vulnerability and risk analysis. It is essential to provide reliable QoS and a robust cloud based network environment for the cloud clients in terms of specific SLAs [9, 13], for example. response time or throughput.

The concept of cross-IdP can be seen as a relationship between clouds, i.e. direct, indirect or transitive trust relationship. For example, take the relationships among three clouds and two IdPs as an example: cloud A (C_A) trusts IdP_x, cloud B (C_B) trusts IdP_x and IdP_y, and cloud C (C_C) trusts IdP_y. From this, the concept of cross-IdP can be implemented in the situation where C_A can get resources from C_C via the help of IdP_x and IdPy with indirect trust relationship, i.e. C_A trusts C_B via IdP_x, and C_B trusts C_C via IdP_y. Three trust levels, low, moderate and high, are defined. In order to keep good quality for cloud data centers, IdP with low level reliability will not be considered when constructing cloud federation. Once one identity provider has a higher reliability value, it means that the identity provider is a better candidate for resource borrowing. In this situation, an IdP with a higher reliability may be given higher priority to be used in the federation over other IdPs with lower reliability.

4.2. Cloud Federation Resource Allocation Framework

In this section, we consider the general architecture of each cloud with three-layered stack presented in [19] as our basic communication model (Fig. 1). Starting from the top down in Fig. 1, we can identify the three-layers: the Cloud-crossing Federation Manager (at cloud provider side) / Federation- Checking Manager (at IdP side), Virtual Infrastructure (VI) Manager, and Virtual Machine Manager. The middle layer VI manager is a basic component of hybrid/private clouds. It acts as a dynamic adjustment for Virtual Environments (VEs), which automates VEs setup, deployment and management, regardless of the underlying Virtual Machine Manager layer (i.e. Xen, KVM, or VMware). The top layer, i.e. Cloud-crossing Federation Manager / Federation- Checking Manager, is able to merge the existing infrastructure into a cloud which handles the creation of new VM, resources provisioning management, identity management, policy management, and monitoring management. The functionality of each module is explained below.

- Cloud-crossing Federation Manager (The middle section of Fig. 1)
  a) Current Resource Status Module (CRSM): Checks required/idle computing resources status.
  b) Message Exchange Module (MEM): Makes request or sends acknowledgement between cloud and trusted IdP.
  c) Resource Matching Module (RMM): Selects appropriate external clouds to allocate resources.
  d) Identity Verification Module (IVM): Makes an authentication request to IdP's identity verification module for federation establishment.
- Resource allocation manager for cloud federation

a) Message Exchange Module (MEM): Sends reply or makes s request between cloud and IdP. (The right section of Fig. 1)

b) Identity Verification Module (IVM): Makes an authentication request to a cloud's identity verification module for federation establishment. (The left section of Fig. 1)

4.3. Target Scenarios in this Study

This study is based on the following scenarios:

a) Available computing resources are dynamically changeable in the cloud environment. Hence, our mechanism operates in real-time.

b) We assume different network transmission overhead among clouds.

c) According to some information security techniques, it is secure to get resources status message and allocate resources among clouds.

d) Local cloud can only get other clouds' idle/required computing resources information from IdP.

e) The layer can be defined by the local cloud. For instance, if external clouds have a direct trust relationship with a local cloud, the trust relationship between them is defined as such in the first layer.

The external clouds may have indirect trust relationship with local cloud. In this case, the trust relationship between them is in second layer.

f) A Local cloud is allowed to rent resources from external clouds. An external cloud might be located in different layers based on the trust relationship.

In our method, once a Local Cloud's (LC's) resources are insufficient, LC first retrieves resources from the external clouds whose layer is 1. If all of the

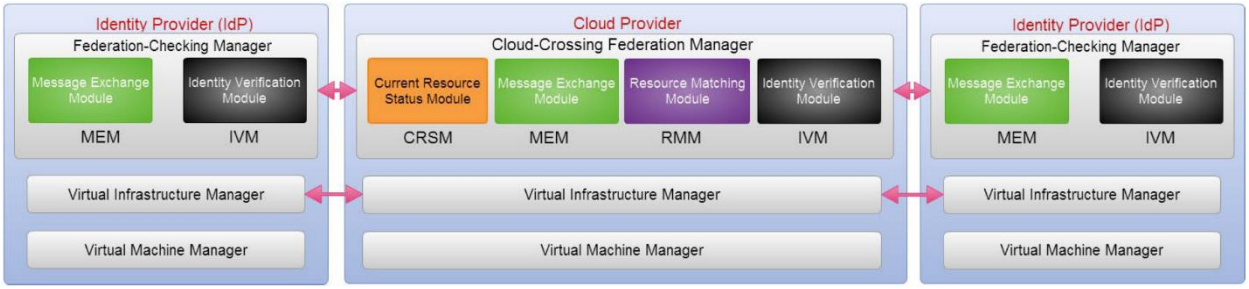

Figure 1. A general resource allocation framework with the three-layer architecture for cloud federation

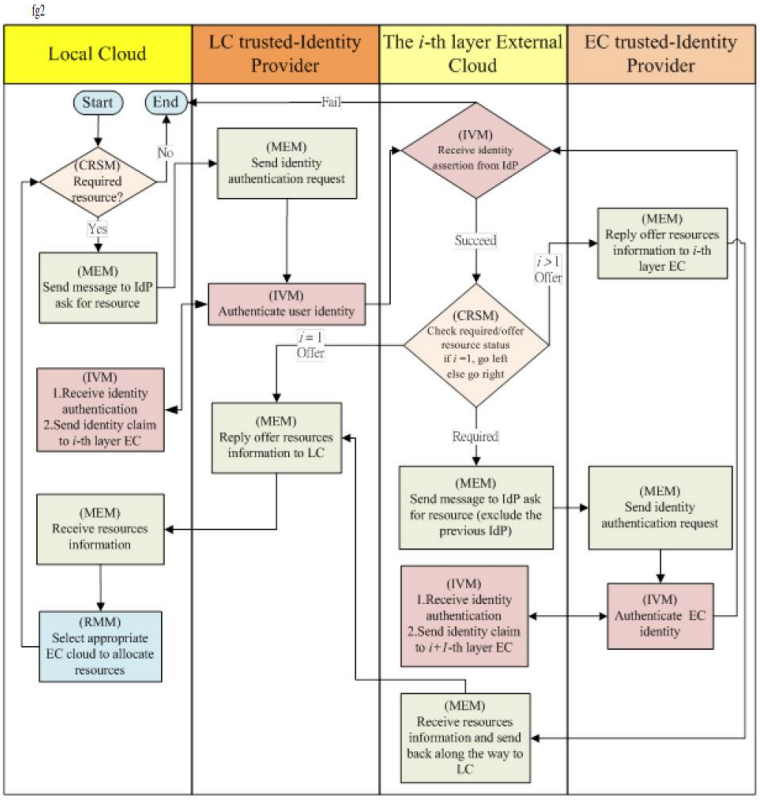

Figure 2. The Normal Operation Process of the *i*–th Layer External Cloud to Establish Cloud Federation

idle resources in layer 1 are still not enough to fulfill a local clouds borrowing request, then the clouds of layer 1 should request resources from the external clouds belonging to layer 2, and so on. It will stop when the local cloud obtains adequate computing resources. The normal operation process of our methods is shown in Fig. 2.

First, LC (CRSM) checks its current resource status and LC (MEM) makes resource request to the IdP (MEM) which is trusted by LC. Next, LC (MEM) sends a resource request message to the cloud which it trusts. When EC (MEM) receives the request, MEM transmits an inner dialog to RMM and checks how much resources it currently has, and shares its idle resources according to the local cloud's needs. Otherwise, the layer-i EC (MEM) will forward the request message to the next layer EC (MEM). It will ask for re-sources via IdPs which it trusts, similar to the above mentioned process. The reply of resource information is sent back to LC along the same route in the request stage, in order to minimize network transmission costs. Once the computing resources collected from ECs are enough for LC, LC (ME) sends the resource information to its RMM. Here, RMM is responsible for matching external clouds' resources, and deciding which cloud could be used for resource allocation. During the resource allocation phases, each cloud and IdP (IVM) utilize SSO authentication with SMAL technology [4] to create trust context, and to establish the cloud federation within the concept of cross-IdP.

5. The Proposed Resource Allocation Algorithm

We define our algorithm as a Trusted-based Resource Al-location algorithm using IdPs (TRA algorithm). A current resource status table, trusted-IdP list and network transmission cost table will be stored in each cloud and a collaborated-cloud list will be maintained in each IdP server. Clouds in a federation environment will exchange IdP reliability values periodically, based on historical data for user authentication, and access authorization in an individual cloud.

In this section, we first introduce the notations used throughout this study. Next, we illustrate the detailed procedures of our proposed resource allocation algorithm and provide an example.

5.1. Notation Description

All notations involved are listed in Table 1. We next formally present the assumptions of our algorithm and constraints.

Table 1. Notations

Notation	Description
LC_i	The i-th local cloud
EC_j	The j-th external cloud
IdP_x	The x-th identity provider
R_x	Trust index of IdP_x's reliability
λ_l, λ_u	The lower/upper thresholds to divide IdP's reliability into three levels
ρ	The value to increase/decrease IdP's reliability
$\alpha_{x,i,j}$	The IdP_x is used to connect with cloud i and cloud j
$c_{i,j}$	The network transmission cost from cloud i to cloud j
$\tau_{i,n}$	The sum of network transmission cost from cloud i to cloud n
rr_i	Current required resources of cloud i
ar_j	Current available resource of cloud j
$r_{i,n}$	Resource allocated from cloud n to cloud i
$CI_{i,j}$	Cost-effective index which is the ratio of r_i, n to π_i
E_i	The total effectiveness of resource allocation for resource-requesting cloud i
TE	Total effectiveness of every resource-requesting cloud

- We assume that a mutual trust relationship exists in our algorithm. An example of a mutual trust relationship between IdP and a cloud is shown in Fig. 3. Suppose that cloud I is federated with other clouds, where cloud I is a local cloud. Local cloud I trusts external clouds J and K with an IdP_x. In addition, external clouds K, L, and M trust with IdP_y, and external clouds K, M and N trust with IdP_z. In this case, we can infer that local cloud I has an indirect trust relationship with external clouds L, M, and N.
- We assume that the cloud federation possesses enough idle resources to fulfill the service requests from all of the clouds:

$$\sum_{j=1}^{n} ar_j \geq \sum_{i=1}^{n} rr_i, \begin{cases} ar_j \geq 0 \\ rr_j \geq 0 \end{cases}. \quad (1)$$

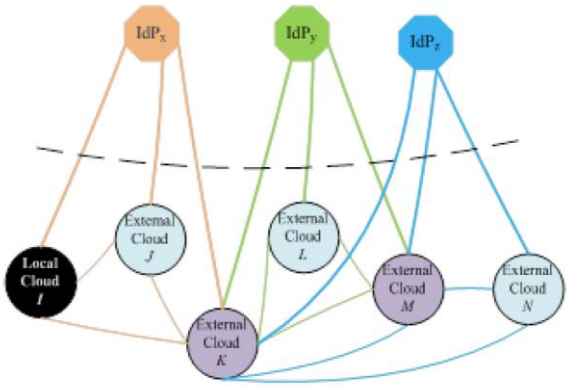

Figure 3. The mutual trust relationship between IdPs and clouds

The reliability (R_x) of IdPs can be set with the same value, for example 0.5, at the initialization phase, and this value will be changed based on IdP's performance. The reliability might be updated periodically by a third party which is an organization that collects the users' previous usage experiences, and then derives the reliability of each IdP.

The actual value $\alpha_{x,i,j}$ is based on IdP_x which is used to being connected with cloud i and cloud j. The value will be dynamically updated depending on the previous performance of the IdP_x. That is, $\alpha_{x,i,j}$ may be increased when successfully allocating resources, or may be decreased when a server crashes, unstable work performance is encountered, there is resource allocation failure, or there are information security threats and attacks. For example, a cloud federation is composed of several clouds (e.g., LC_A, EC_D, EC_F, EC_G). There are several IdPs in the cloud environment but only two IdPs (e.g., IdP_k, IdP_y) that are used in this federation. The $R_x(a_{k,A,B})$ will be increased as long as IdP_k successfully helps LC_A obtain resources:

$$R_x = \alpha_{x,i,j}, 0 < \alpha_{x,i,j} < 1. \quad (2)$$

- Table 2 presents IdP's reliability level which is divided into three levels: (a) Low; (b) Moderate; (c) High.

Table 2. IdP's reliability level $F(\lambda_l, \lambda_u)$

Reliability Value	Level
$0 \le R_x < \lambda_l$	Low
$\lambda_l \le R_x < \lambda_h$	Moderate
$\lambda_u \le R_x < 1$	High

- We assume that the network transmission cost ($c_{i,j}$) is a real number from 1 to t:

$$\tau_{i,n} = c_{i,j} + c_{j,k} + \cdots + c_{l,m} + c_{m,n},$$
$$c_{i,j} = \begin{cases} 1 < c_{i,j} < t \\ \forall t \in R \end{cases}. \quad (3)$$

- The amount of resources ($r_{i,n}$) which is allocated from cloud n to cloud i should be less or equal to the total amount of available resources (ar_n) of Cloud n:

$$ar_n \ge r_{i,n}, \text{ where } ar_n > 0 \text{ and } r_{i,n} > 0. \quad (4)$$

- During the process of finding an EC_n to allocate resources, considering cost-effectiveness Index ($CI_{i,n}$) which means a ratio of $r_{i,n}$ to $\tau_{i,n}$ that is calculated from EC_n to LC_i. The $r_{i,n}$ depends on both the resource requirement of LC_i and the idle resources offered from EC_n:

$$CI_{i,n} = \frac{r_{i,n}}{\tau_{i,n}}. \quad (5)$$

- We assume that the requests of a resource will stop when the available resources, collected from n external clouds, are enough to fulfill the requests from k local clouds:

$$\sum_{i=1}^{k} r_{i,n} \ge \sum_{j=1}^{k} rr_j, 1 \le k \le n. \quad (6)$$

- According to the above constrains, we assume that i is the number of local clouds which require idle resources, and x is the IdP used between two clouds. In addition, the effectiveness will be calculated depending on the following formulation for constructing the cloud federation:

$$E_i = \sum \left[\frac{r_{i,n}}{\tau_{i,n}} \times \left(a_{x,i,j} \times a_{x,j,k} \times \ldots \times a_{x,l,m} \times a_{x,m,n}\right)\right]. \quad (7)$$

$$TH = \sum_{i=1}^{n} E_i, \forall n \in N. \quad (8)$$

5.2. The Proposed Algorithm

5.2.1. General Rules of Heuristic Algorithm for Cloud-Federation (Fig. 4)

Step 1. Sort local cloud's trusted IdP reliability, then select one of the local cloud's trusted IdP's with the highest reliability within the high and moderate levels.

Step 2. Consider external clouds which have available resources and have a trusted relationship with a local cloud. Choose the route from local cloud i to borrow resources from external cloud j with maximum performance ($CI_{i,j}$); the highest amount of resources with the least amount of cost. In the case of the same $CI_{i,j}$, it would be better to choose the one who has the most amount of actual idle resources to provide for local cloud.

Step 3. The reliability (R_x) of the IdP_x should be increased with ρ once the path was selected in step 2.

Step 4. Check step by step as one route is constructed and stop extending until satisfying formula (6) and then go to step 5. Otherwise, go to step 1 through step 4 to find more external clouds' resources until satisfying formula (6).

Step 5. Calculate the total effectiveness of resource allocation for resource-requesting cloud i according to formula (7).

Step 6. Check whether any required-resources cloud exists. If it exists, then go to step 1 through step 6; if it does not exist, go to step 7.

Step 7. Summarize Total Effectiveness (TE) of all required resource clouds.

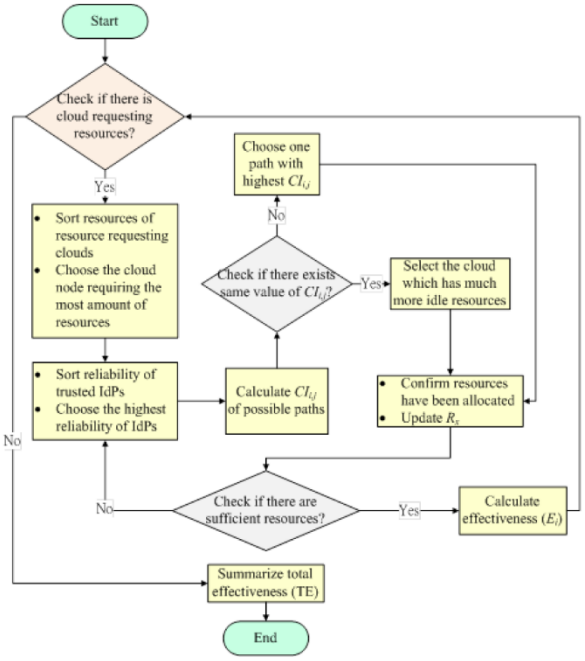

Figure 4. The flowchart of the proposed algorithm

5.2.2. An Example to Illustrate the Procedure of our Algorithm

Assume that a company consists of 5 clouds. There are two clouds that have insufficient resources at a peak time, and the remaining clouds have a trust relationship with those two clouds based on a common IdP which they trust. This would help the two clouds obtain resources for providing continuous services. For some parameters, we set $F(\lambda_l, \lambda_u)$ as $F(0.4, 0.7)$ and $\rho = 0.05$ in this case. Thus, the reliability of IdP in the following example with IdP_4 not being used in the federation (refer to the Table 3) defines the three levels by $F(0.4,0.7)$.

Table 3. IDP's reliability level $F(0.4,0.7)$

Reliability Value	Level
$0 \le \lambda < 0.4$	Low
$0.4 \le \lambda < 0.7$	Moderate
$0.7 \le \lambda \le 1$	High

The scenario is shown in Fig. 5 which contains the amount of required (R_{req}) and idle (R_{idle}) resources and the trusted-IdP at each cloud. The local cloud (LC_B) is able to connect with three external clouds (EC_A, EC_D, EC_E) through the same identity providers (i.e. IdP_1, IdP_2, IdP_3) which they trust. On the other hand, the local cloud (LC_E) is able to connect with two external clouds (EC_B, EC_D) through the one identity providers (IdP_3).

- Round 1 in our example (Fig. 6):

Step 1. Select one of the most reliable IdP from the LCB's list, which includes IdP_1, IdP_2, and IdP_3. Here, IdP_2 with reliability value 0.7 will be first considered in the federation.

Step 2. Either EC_A or EC_D is a possible cloud able to be chosen in this step. However, $CI_{B,A}$ is better than $CI_{B,D}$; thus EC_A should be selected:

$CI_{i,n}$	$r_{i,n}/\tau_{i,n}$
$CI_{B,A}$	300 / 5 = 60
$CI_{B,D}$	100 / 4 = 25

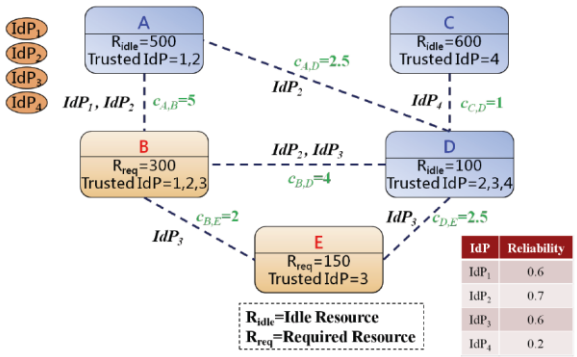

Figure 5. An example to illustrate the proposed algorithm

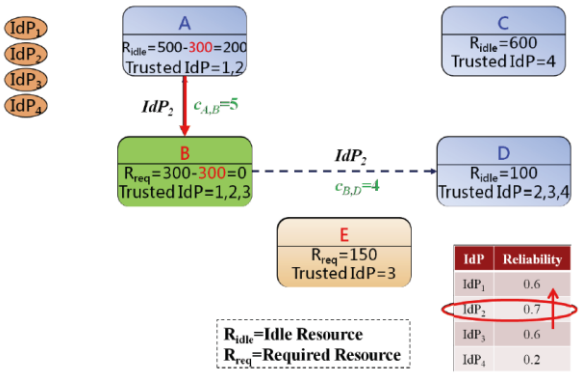

Figure 6. Round 1 in our example

Step 3. Because IdP_2 is used to help the resource allocation from LC_B to EC_A, R_2 should be increased by 0.05 at this time.

Step 4. Satisfy formula (6)

$$\sum_{i=1}^{k} r_{i,n} \ge \sum_{j=1}^{k} rr_j,\ 1 \le k \le n,$$

$300 \ge 300$, stop extending and go to step 5.

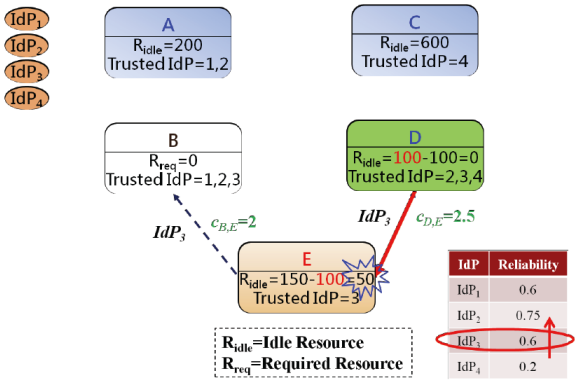

Figure 7. Round 2 in our example

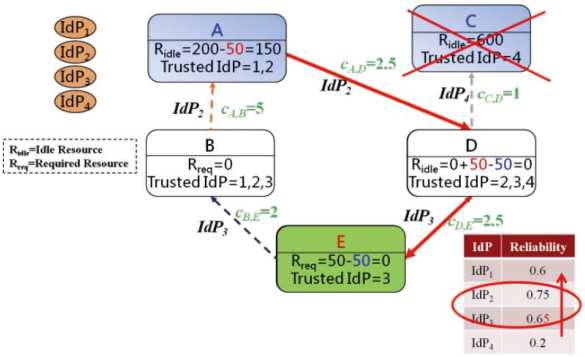

Figure 8. Round 3 in our example

Step 5. $E_B = \left\lceil \frac{300}{5} \times 0.7 \right\rceil = 42$.

Step 6. So far, cloud B has gotten enough resources, but cloud E is still in need of resources.

- Round 2 in our example (Fig. 7):

Step 1. Select one of the most reliable IdP from the LC_E's list, which only includes IdP_3 and it is therefore to be considered in the federation.

Step 2. Either EC_B or EC_D is a possible cloud to be chosen in this step. Nevertheless, $CI_{E,D}$ is better than $CI_{E,B}$. EC_D should thus be selected:

Step 3. Because IdP_3 is used to help the resource allocation from LC_E to EC_D, R_3 should be increased by 0.05 at this time.

$CI_{i,n}$	$r_{i,n}/\tau_{i,n}$
$CI_{E,B}$	0 / 3 = 0
$CI_{E,D}$	100 / 2.5 = 40

Step 4. It does not satisfy formula (6)

$$\sum_{i=1}^{k} r_{i,n} \geq \sum_{j=1}^{k} rr_j,\ 1 \leq k \leq n,$$

i.e. 100 ≤ 150, do the next round from step 1 to step 4 to find more available resources.

- Round 3 in our example (Fig. 8):

Step 1. Since there is no cloud which can directly allocate resources to LC_E, consider EC_D and EC_B as a temporary Local Cloud at this moment. Select one of the most reliable IdP from the LC_D's and LC_B's list, which includes IdP_1, IdP_2, IdP_3, and IdP_4. Similarly, IdP_2 is to be considered in the federation. At the same time, exclude the path which connects to the IdP's with low reliability, such as cloud D to cloud C.

Step 2. EC_A is the only one possible cloud to be chosen in this step, but there exists two ways back to cloud E. As $CI_{E,A-D}$ is better than $CI_{E,A-B}$, EC_A and EC_D are selected:

$CI_{i,n}$	$r_{i,n}/\tau_{i,n}$
$CI_{E,A-D}$	50 / 2.5 + 2.5 = 10
$CI_{E,A-B}$	50 / 2 + 5 = 7.14

Step 3. Because IdP_2 and IdP_3 are used to help the resource allocation from EC_A and EC_D to LC_E, R_2 and R_3 should both be increased by 0.05.

Step 4. In satisfying formula (6)

$$\sum_{i=1}^{k} r_{i,n} \geq \sum_{j=1}^{k} rr_j,\ 1 \leq k \leq n,$$

i.e. 100+50 ≥ 150, stop extending and go to step 5.

Step 5. Calculate

$$E_E = \left\lceil \frac{100}{2.5} \times 0.6 + \frac{50}{2.5+2.5} \times (0.65 \times 0.75) \right\rceil = 28.875.$$

Step 6. Cloud E has already received adequate resources and there is no cloud requesting resources. Stop and go to Step 7.

Step 7. Fig. 9 shows that the final cross cloud federation consists of C_B, C_A, C_D, and C_E. Calculate total effectiveness of all the roads chosen in the federation, i.e. TE = E_B + E_E = 70.875.

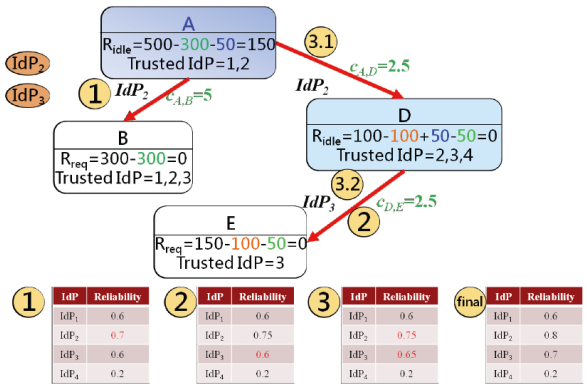

Figure 9. Final results in our example

6. Simulation Experiments

6.1. Simulation Environment

In this section, we build a simulation environment with Java (TM) SE Runtime Environment version 1.7.0. Experiments were performed on Windows 7 and hardware CPU Intel(R) Core(TM) i5-2400M 3.10, RAM 4 GB. We present two simulation experiments to analyze the effectiveness of resource allocation in cloud federations. These two simulations can be described as occurring at two peak time points, so that there are two results in this section. Cloud providers (nodes) are divided into two parts, one part refers to re-source-required cloud and the other is the cloud with avail-able (or idle) resources which can be borrowed. We set the number of cloud providers (nodes) be 5, 10 and 20, including 2 resource-required cloud nodes and others with available resources in each topology. There are 6 IdPs that will be exploited to construct trust relationship. Note that each cloud has at most 3 trusted IdPs.

In order to satisfy the constraint that the sum of the idle resources should be more than the total of the required re-sources, all the initial statuses of resources are randomly created by the program with the constraint that the idle re-sources are 1.5 times the amount of total required resources. For instance, when the total required resources are 500, the sum of the idle resources are 750. The amount of required/idle re-sources of each cloud and the reliability of each IdP are different in these two experiments (or at different time points). The reliability of each IdP and transmission cost between two cloud providers (nodes) was randomly generated by the program and the values are all real numbers. The performance was evaluated with different factors, including an algorithm, the utility of each cloud provider node, and effectiveness. Table 4 lists the parameters used in the simulation scenarios along with their default values.

6.2. Performance Evaluation

Here we investigate the effectiveness of the proposed re-source allocation framework along with the performance of the proposed solution. To illustrate the superior performance of our proposed approach, we evaluated 360 test cases for our resource allocation algorithm under two scenarios. In addition, we compared our proposed algorithm with existing and most relevant work [5].

- TRA algorithm: the proposed algorithm which considers both security and cost-effectiveness.
- Revised TRA algorithm: the modified algorithm first considers minimal network transmission cost instead of reliability.
- Revised Celesti algorithm [3]: this algorithm selects a cloud with most idle resources to support the need of a resource-requesting cloud which trusts the same IdP as the supporting cloud. Note that here we implement the Celesti algorithm under our two scenarios, where all designs of the Celesti algorithm are fully adopted in the experiments. However, as the simulation setting may not be completely the same with that in [5], we simply call our implementation of the Celesti algorithm as Revised Celesti algorithm.

Table 4. Parameter settings in the simulation experiments

Parameter	Value
Number of cloud providers (nodes)	$[55, 10, 20]$
Number of resource-requesting cloud nodes in each network	2
Number of identity providers	6
Maximum of trusted IdP for each cloud provider	3
Minimum of trusted IdP for each cloud provider	1
Total required resources for a federation	$[500, 1000, 2000]$
Total available resources in a network	$[750, 1500, 3000]$
Transmission cost between two cloud providers (nodes)	$[1 - 5] \in R$
Reliability of each IdP	$[0 - 1] \in R$

6.3. Numerical Results and Comparison

We assume that the experiments are happening at two different time-points, therefore the results are shown in two different sets, i.e. all the resources are randomly created by the program with a constraint that total amount of required resources units are 500 in 5, 10 and 20 cloud nodes, 1000 in 5, 10 and 20 cloud nodes, and 2000 in 5, 10, and 20 cloud nodes. Figures 10 and 11 express the two experiments which are categorized by the number of cloud nodes and required 500, 1000 and 2000 resource units. For example, in experiment 2, the effectiveness rises significantly as the required resource units increase when the number of cloud nodes is not very high. On the other hand, the effectiveness growth is not significant when the required resources units increase with even more number of cloud nodes. Figures 12 and 13 show the comparison of the two experiments with an average of 500, 1000 and 2000 resource units. When requiring 1000 resource units, the total effectiveness is twice than required 500 resource

units. When requiring 2000 re-source units, the total effectiveness is much more twice than required 1000 resource units. As a result, we infer that the effectiveness will be substantially higher as the requirement load increases.

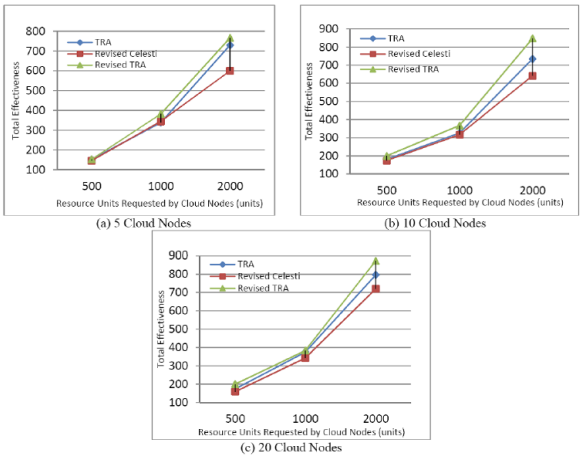

Figure 10. Experiment 1 - categorized by the number of cloud nodes

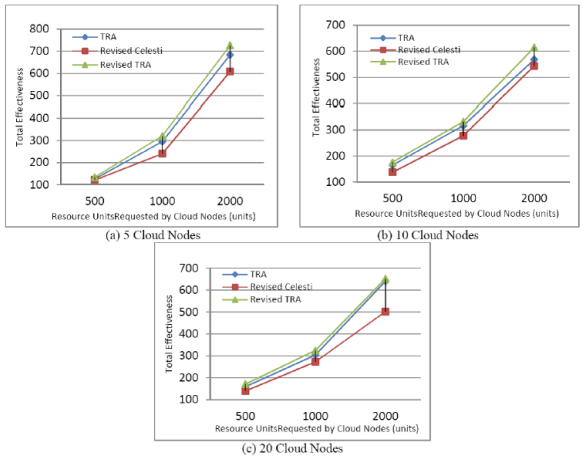

Figure 11. Experiment 2 - categorized by the number of cloud nodes

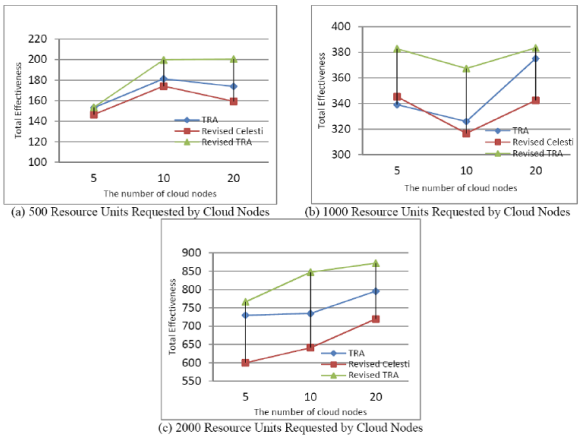

Figure 12. Experiment 1 - categorized by the resource units requested by cloud nodes

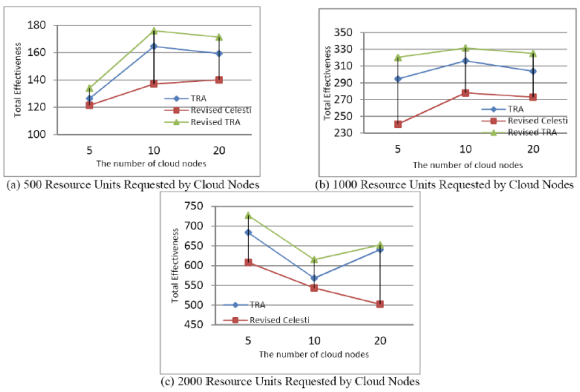

Figure 13. Experiment 2 - categorized by the resource units requested by cloud nodes

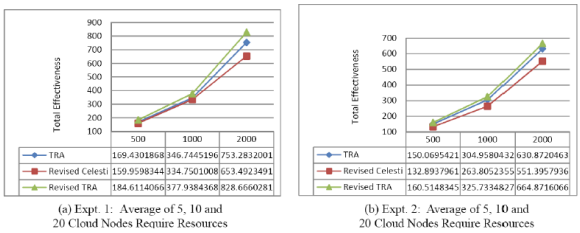

Figure 14. Comparison of two experiments of average of 5, 10 and 20 cloud nodes require resources

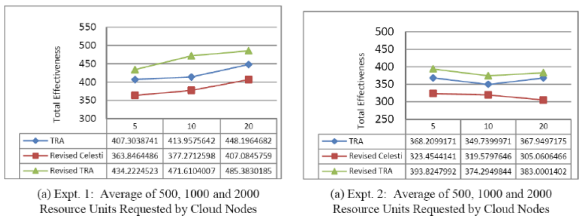

Figure 15. Comparison of two experiments of average of 500, 1000 and 2000 resource units requested by 5, 10 and 20 cloud nodes

Fig. 14 shows the comparison of two experiments with an average of 5, 10 and 20 cloud nodes requiring resources, and the results of each type (e.g., 500, 1000, and 2000 required resources units) below correspond to an average of 60 simulations. The effectiveness will increase as the amount of required resource units increase. Fig. 15 shows the comparison of two experiments with an average of 500, 1000 and 2000 resource units requested by cloud nodes. We find out that the total effectiveness of the TRA algorithm and revised TRA algorithm become more significant than the revised Celesti algorithm as the number of required resources increases. Hence, we infer that no matter how many cloud nodes there are, the proposed algorithm will continue to surpass competing algorithms. Although the revised TRA algorithm has a better effectiveness than the TRA algorithm, the TRA algorithm is preferred better owing to its consideration on both security and cost-effectiveness.

With the average effectiveness shown by our simulations, we infer that the effectiveness increases as

the number of required resources increases. In addition, our proposed algorithm has better effectiveness than the revised Celesti algorithm under our proposed scenarios. Compared to the revised Celesti algorithm, as shown in experiment-1, the TRA algorithm is better with an 8.8% improvement, and the revised TRA algorithm is better with a 17.5% improvement. On the other hand, in experiment-2, the TRA algorithm is better with 5.7% improvement and revised TRA algorithm is better with 17.6% improvement.

7. Conclusions

Cloud computing is a popular technology which delivers significant benefits for cloud customers. However, it raises resource management problems while maintaining quality of services. In order to provide strong availability and better efficiency of cloud services, an efficient resource allocation algorithm for cloud federation environment is proposed along with the consideration of network transmission cost and the reliability of identity provider. Trust relationship among individual clouds through common identity providers is the main factor for constructing the resource provisioning framework. In addition, we conduct simulation experiments and compare our results with the most relevant study, i.e. the Celesti algorithm. From the simulation results, we show that our scheme is feasible and practical for resource allocation management in cloud federation environments.

Acknowledgment

The authors gratefully acknowledge the support from Taiwan Information Security Center (TWISC) and Ministry of Science and Technology, Taiwan, under the Grants Numbers MOST 103-2221-E-011-091-MY2, MOST 103-2221-E-259-016-MY2 and MOST 103-2221-E-011-090-MY2.

References

[1] **E. Apostol, I. Baluta, A. Gorgoi, V. Cristea.** Efficient Manager for Virtualized Resource Provisioning in Cloud Systems. *2011 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2011, pp. 511–517.

[2] **A. Buecker, P. Ashley, N. Readshaw.** Federated Identity and Trust Management. *IBM Redbooks*, 2008.

[3] **R. Buyya, S. K. Garg, R. N. Calheiros.** SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. *2011 IEEE International Conference on Cloud and Service Computing*, 2011, pp. 1–10.

[4] **A. Celesti, F. Tusa, M. Villari, A. Puliafito.** Three-phase Cross-Cloud Federation Model: the Cloud SSO Authentication. *2010 Second International Conference on Advances in Future Internet*, 2010, pp. 94–101.

[5] **A. Celesti, F. Tusa, M. Villari, A. Puliafito.** How to Enhance Cloud Architectures to Enable Cross-Federation. *2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 337–345.

[6] **A. Celesti, F. Tusa, M. Villari, A. Puliafito.** Evaluating a Distributed Identity Provider Trusted Network with Delegated Authentications for Cloud Federation. *The 2nd International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2011)*, 2011, pp. 79–85.

[7] **A. Celesti, F. Tusa, M. Villari, A. Puliafito.** Federation Establishment between CLEVER Clouds through a SAML SSO Authentication Profile. *International Journal on Advances in Internet Technology*, 2011, Vol. 4, No. 1 & 2, 14–27.

[8] **H. Goudarzi, M. Pedram.** Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems. *IEEE 4th International Conference on Cloud Computing*, 2011, pp. 324–331.

[9] **N. Gonzalez, C. Miers, F. Redígolo, M. Simplício, T. Carvalho, M. Näslund, M. Pourzandi.** A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 2012, Vol. 1, Article No. 11.

[10] **M. Guazzone, A. Cosimo, C. Massimo.** Energy-efficient resource management for cloud computing infrastructures. In: *Proc. of the Cloud Computing Technology and Science (CloudCom)*, 2011, pp. 424–431.

[11] **U. Kylau, I. Thomas, M. Menzel, C. Meinel.** Trust Requirements in Identity Federation Topologies. *2009 International Conference on Advanced Information Networking and Applications*, 2009, pp. 137–145.

[12] **W. Li, L. Ping.** Trust Model to Enhance Security and Interoperability of Cloud Environment. *The Cloud-Com 2009, LNCS 5931*, 2009, pp. 69–79.

[13] **F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, D. Leaf.** NIST Cloud Computing Reference Architecture. *National Institute of Standards and Technology Special Publication*, 2011, SP 500–292.

[14] **S. Malik, F. Huet.** Virtual Cloud: Rent Out the Rented Resources. *The 6th IEEE International Conference for Internet Technology and Secured Transactions (ICITST-2011)*, 2011, pp. 536–541.

[15] **D. H. McKnight, N. L. Chervany.** The Meanings of Trust. *Technical Report, Carlson School of Management, University of Minnesota*, 1996, pp. 94–104.

[16] **P. Mell, T. Grance.** The NIST Definition of Cloud Computing. *National Institute of Standards and Technology Special Publication 800–145*, 2011.

[17] **K. Mochizuki, S. Kutibayashi.** Evaluation of optimal resource allocation method for cloud computing environments with limited electric power capacity. *IEEE International Conference on Network– Based Information Systems*, 2011, pp. 1–5.

[18] **S. Pearson, Y. Shen, M. Mowbray.** A Privacy Manager for Cloud Computing. The *CloudCom 2009, LNCS 5931*, 2009, pp. 90-106.

[19] **B. Sotomayor, R. Montero, I. Llorente, I. Foster.** Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 2009, Vol. 13, No. 5, pp. 14–22.

[20] **P. Steiner.** On the Internet nobody knows you're a dog. *The New Yorker*, 1993, Vol. 69, No. 20.

[21] **M. Stihler, A. O. Santin, A. L. Marcon, J. da Silva Fraga.** Integral federated identity management for cloud computing. *5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–5.

[22] **X. Wang, J. Sun, M. Huang, C. Wu, X. Wang.** A resource auction based allocation mechanism in the cloud computing environment. *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, 2012, pp. 2111–2115.

[23] **X. You, X. Xu, J. Wan, D. Yu.** RAS-M: Resource allocation strategy based on market mechanism in cloud computing. *Fourth China Grid Annual Conference*, 2009, pp. 256-263.

Received April 2014.