Journal of Network and Computer Applications 77 (2017) 87–105

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Federated cloud resource management: Review and discussion

Misbah Liaqat[a,b,*], Victor Chang[c], Abdullah Gani[a,b,*], Siti Hafizah Ab Hamid[a], Muhammad Toseef[d], Umar Shoaib[d], Rana Liaqat Ali[e]

[a] Department of Computer System and Technology, University of Malaya (UM), 50603 Kuala Lumpur, Malaysia
[b] Center for Mobile Cloud Computing Research, University of Malaya (UM), 50603 Kuala Lumpur, Malaysia
[c] Xi'an Jiaotong Liverpool University, Suzhou, China
[d] Faculty of Computing and IT University of Gujarat, Pakistan
[e] Department of Physics, COMSATS Institute of Information Technology, Islamabad, Pakistan

ARTICLE INFO

Keywords:
Resource management
Cloud computing
Federated clouds

ABSTRACT

Federation of clouds is the future of cloud computing, mobile cloud computing, Internet of things, and big data applications. The utilization of federated resources is envisioned to increase the quality of service, cost benefits, and reliability. Resource management in the federation of multiple clouds is a pressing issue owing to the lack of cross-domain knowledge, security, trust, and administrative policies. This study classifies these resource management functions in the federated cloud environment into resource pricing, resource discovery, resource selection, resource monitoring, resource allocation, and disaster management. Each federated resource management function is discussed, and insights into state-of-the-art research are then provided. These resource management functions are also compared based on the performance metrics that are suitable for an individual function. Finally, we indicate the open challenges for further research with regard to each classified resource management function.

1. Introduction

Cloud computing (Armbrust et al., 2010; Buyya et al., 1969; Nurmi et al., 2009; Vaquero et al., 2008), a user-centric computational model, is a flexible paradigm of deploying and sharing distributed services and resources with the pay-per-use model. With virtual machine (VM) technology (Smith and Nair, 2005) and data centers (DCs), computational resources, such as memory, central processing unit (CPU), and storage, are dynamically reassembled and partitioned to meet the specific requirements of end users. The cloud not only considers flexible platform-independent access to resources and information anywhere and anytime but also changes the way of designing, deploying, building, expanding, and running applications (Cao et al., 2009). The demand's growth for cloud services is presenting considerable challenges for cloud providers to meet the requirements and satisfaction of end users (Talia, 2012).

Research by Bakshi (2009) and Bernstein et al. (2009) showed that the trend in cloud computing pattern will shift from a single provider to federated clouds, which are expected to include numerous distributed public and private cloud platforms (Lopez-Rodriguez and Hernandez-Tejera, 2011). Federated clouds enable public and private clouds to share their resources with each other to scale up their resource pools at peak times. The promises of nearly infinite computing power, concomitant storage, and economies of scale can only truly be achieved by cloud federation. The reason behind cloud federation is the finite physical resources in the resource pool of a single provider.

In the federated cloud environment, a cloud provider acts as both infrastructure provider and consumer. The egocentric and rational behavior of federation members focuses on maximizing their revenue and resource utilization by serving as many consumers as possible. Here, consumers can be either federation members who rent resources from one another or regular cloud users. The role of efficient resource management is prominent in such a scenario to guarantee the service request of both the federation members and direct consumers. The resource management functions in the federated cloud environment ensure the objective of federation members and the aggregate utility of the federation, which is necessary for the continuation of the federation.

This study focuses on the resource management functions of federated cloud in which an individual cloud provider provides and consumes Infrastructure as a Service (IaaS) to and from other federation members. The terms inter-cloud, federated cloud, and multi-cloud are interchangeably used in this article for the federation of cloud providers. Three surveys (Grozev and Buyya, 2014; Petcu,

* Corresponding authors at: Department of Computer System and Technology, University of Malaya (UM), 50603 Kuala Lumpur, Malaysia.
E-mail addresses: mibahaseeb@siswa.um.edu.my (M. Liaqat), abdullah@um.edu.my (A. Gani).

http://dx.doi.org/10.1016/j.jnca.2016.10.008
Received 13 May 2016; Received in revised form 14 October 2016; Accepted 18 October 2016
Available online 20 October 2016
1084-8045/

2014; Toosi et al., 2014) were previously published on cloud federation. However, the targets of these surveys outlined the taxonomies, terminologies, definition, and challenges for inter-cloud systems. This survey differs from them in the essence that we focus only on the resource management aspects of inter-cloud computing and study the proposed solutions for inter-cloud resource management issues. Our contributions are as follows: (1) the federated resource management mechanism is classified into resource pricing, resource discovery, resource selection, resource monitoring, resource allocation, and disaster management. (2) Rigorous works on resource pricing, resource monitoring, resource discovery, resource selection, resource allocation, and disaster management are given. The operation and drawbacks of the mechanism are presented with a comparative analysis in terms of different performance metrics for each considered class. (3) Open challenges in each considered class of federated resource management is indicated. (4) Novice researchers are encouraged to work on the research problems.

The remainder of this article is divided into three sections. Section 2 presents a general background of cloud computing, federated cloud computing environments, and resource management in federated cloud environments. Section 3 provides the classification of resource management functions along with the discussion of the state-of-the-art literature. Section 4 concludes the paper.

2. Background

2.1. Cloud computing

The fundamental idea of cloud computing is to deliver computational resources as services over the Internet. Consumers are not required to invest in a large computer system to conduct their business; instead, they can acquire cloud computing services on the basis of their demands (Liaqat et al., 2016). The underlying hardware is commonly hosted in large DCs using sophisticated virtualization practices to realize high-level agility, scalability, and availability. According to Armbrust, the resource's elasticity, which does not require a premium to be paid for the pool of resources, is unprecedented in the IT history (Armbrust et al., 2010). Therefore, the establishment of enterprises systems using the cloud platform has gradually become popular in recent decades. In addition, in conventional computing system s, users are required to invest in dedicated software and hardware whereas cloud computing delivers the software and hardware resources in a pay-per-use manner (Foster et al., 2008). This feature considerably reduces the cost of maintenance and deployment. Cloud computing is offered using three service delivery models, i.e., Platform as a Service (PaaS), Software as a Service (SaaS), and IaaS (Aversa et al., 2010).

An impressive increase in the demand for and availability of cloud systems, which are best represented by Amazon's system, has been observed in the last few years. Amazon's EC2[1] is one of the most extensively used cloud services and offers a variety of VMs with different capacities where customers use the VM instances to run their application based on utility computing. Moreover, Rackspace[2] and Joyent[3] provide similar services to their consumers. The public availability of these services is often referred to as public cloud. By contrast, the infrastructure used for the particular organization is known as a private cloud (Tolba and Ghoneim, 2015). For example, EUCALYPTUS is a software environment that is used for the deployment of the private cloud and has compatibility-related concerns with Amazon's EC2. Furthermore, it represents the extensible and modularized policy for resource allocation. Currently, EUCALYPTUS supports two simple types of policies, namely, round robin and greedy (Sotomayor et al., 2009).

2.2. Federated clouds

Current cloud technologies are designed according to the needs of service providers. The main expectation of the providers is that one size fits all needs. Therefore, the service consumers are forced to adapt their applications to the available stack of software. In the standard cloud computing model, in which a customer employs a single cloud service provider (CSP) and pays for resources, service disruption can affect customers that depend exclusively on it without having access to it. Depending on a single cloud provider makes enforcing sufficient usability and responsiveness to customers throughout the globe difficult. The utilization of resources and services from multiple clouds is motivated by the requirements of their customers or their providers. The usage of services from numerous clouds is driven by the reasons stated in Table 1.

Several scenarios can be considered to accomplish cloud federations, e.g., hybrid cloud, multi-cloud, and aggregated clouds (Ferrer et al., 2012; Grozev and Buyya, 2014; Petcu, 2014) for the dynamic cooperation and balancing of workload among a set of cloud DCs and providers. Cloud federation is not yet mature mainly because of the diversity of the approaches to the concept implementation. The desired semi-automated management through the selection of offers, based on the monitoring tools for the quality of services (QoS), is not yet technically possible. The differences among the current application programming interfaces (APIs) hinder the easy composition or configuration of services to be consumed from multiple clouds. Several technical barriers should be overcome, such as interoperability and portability, data and service mobility, and middleware openness, to make the usage of services from multiple clouds a reality. Many researchers currently focus on these barriers, and the literature has presented considerable innovative solutions for particular problems. However, middleware prototypes that can support a large number of scenarios of using services from multiple clouds remain lacking.

2.3. Resource management

A cloud computing infrastructure, whether single or federated cloud, is a complex distributed system composed of a multitude of computational resources. These resources handle the unpredictable client requests and the effects of external events beyond user and system administrator control. Cloud resource management significantly affects the performance, functionality, and cost factors of system evaluation. Cloud resource management also involves complex decisions and policies for multi-objective optimization. This task is challenging because of the complexity, geographical span, and unceasing and unpredictable interactions with the system, thereby making a precise global information state impossible.

Cloud resource management strategies related to the three delivery models of cloud, namely, PaaS, IaaS and SaaS, differ from one another. In all cases, the CSPs are faced with fluctuating, large workloads that challenge the claim of cloud elasticity. In some cases, when they can predict a workload spike, they can provide resources through advance reservation, e.g., seasonal web application may be subject to spikes.

For an unplanned spike, the situation is complicated. Auto scaling can be used for unplanned spike loads, provided that a monitoring system that justifies the decision to allocate, reallocate, or release resources on demand in real time exists. Auto scaling services are given by PaaS providers, such as Google App Engine.[4] Auto scaling for IaaS is complex because of the lack of and the deficiencies in the available standards.

In cloud computing, whether single or federated, variation is

[1] http://aws.amazon.com/ec2/.
[2] http://www.rackspace.com/cloud/.
[3] http://www.joyent.com/.
[4] https://cloud.google.com/appengine/.

Table 1
Reasons of federation of clouds.

Reasons	Description
Sharing	True notion of sharing between parties have different organizational policies and different technical capabilities.
Fault tolerance	Service replication across different providers in case one provider faces an outage the service placed on other providers can be activated as a failsafe.
Improved QoS	Minimizing the latency and delays by serving the request from a more geographically nearest provider or reducing the response time by serving the request from a more capable provider.
Cost efficiency	Can shuffle between providers for a cheaper provider. For example only using spot instance from different providers.
Reducing Service Level Agreement (SLA) Violation	In the case of resource scaling out, cloud service provider (CSP) can reduce it is penalization by renting resources from other federation members.
Provider independence	The consumer would not be dependent upon a single provider.
Contract ending	If a contract with one provider is on the verge of ending, no worry of service blockage.

unpredictable and frequent, and centralized management and control may be unable to provide uninterrupted services and functional guarantees. Thus, centralized management cannot support adequate solutions to cloud resource management policies.

3. Resource management in federated cloud

Several problems should be considered while managing the resources in a federated cloud computing environment. In this section, we present a review of significant resource management techniques covering federated resource management functions, such as pricing, discovery, selection, monitoring, and allocation. The devised taxonomy is presented in Fig. 1.

3.1. Resource pricing

In federated cloud computing, consumers leverage various types of computational and storage resources and services from one or more than one resources or service providers using a fixed or variable (pay-per-use) pricing schemes. In federated clouds, consumers and suppliers of cloud resources are rational players and inclined toward maximizing their own benefits when contributing and utilizing shared resources and services (Nielson et al., 2005). In a federated cloud environment, resource supply and demand fluctuate as consumers and providers join and leave the federation. Pricing function is subsequently used to manage the individual rationality of the consumers and providers. The oscillation in the workload and resource availability in the federation is confined to the need of dynamic pricing strategies in federated cloud based on the principles of demand and supply. The concept of pricing in a standalone single cloud system and federation of multiple clouds is pictorially represented in Fig. 2. A general survey of several pricing schemes for federated cloud is presented below.

A dynamic pricing scheme was introduced in Mihailescu and Teo (2010a), which is suitable for resource's allocation on federated clouds. The study employed an auction-based validation framework for service buying, in which the proposed framework pricing function controlled the rational behavior of a user along with maintaining the supply and demand. The strategic proof of the dynamic pricing was measured by the number of successfully furnished requests. However, the cost of VM outsourcing under uncertainty and the performance measures under stochastic demand were not treated jointly. Conversely, their pricing scheme was considered to distribute a request for various resource types.

The variation in the demand and supply of globally distributed shared resources in a federation of clouds was studied in Mihailescu and Teo (2010b). This variation was due to the dynamic joining and leaving of the users in the federation over time. The authors proposed a dynamic pricing model to manage the allocation of shared federated resources with numerous types of resources. A cloud resource market was introduced, where rational users could both utilize resources (consumers) and provide (vendors). A rational user was inclined toward maximizing its own utility function. In their proposed model, the objective utility functions of the users were shaped by the buyer budget and the seller costs to achieve incentive compatibility, budget balance, and individual rationality.

In Gomes et al. (2012), general equilibrium theory was employed in a marketplace, where multiple cloud providers coordinated with one another using a trading market, to calculate the resource price. The

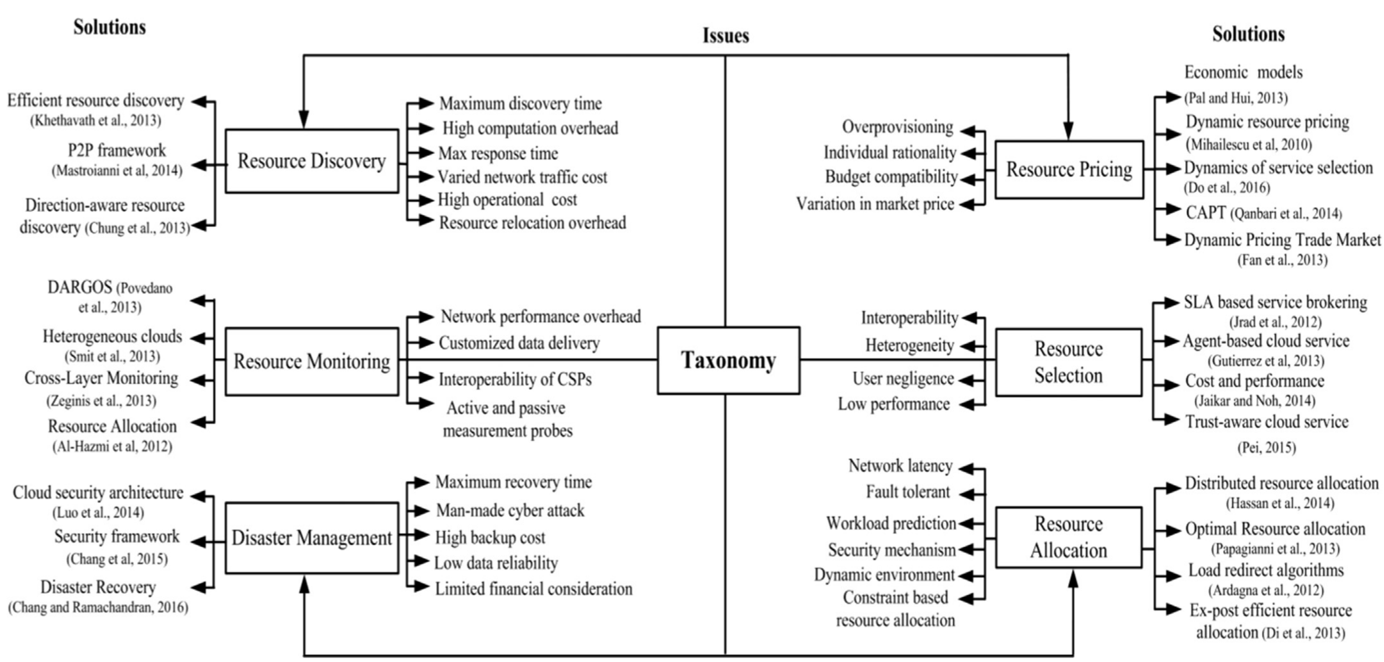

Fig. 1. Taxonomy of resource management techniques Issues and solutions.

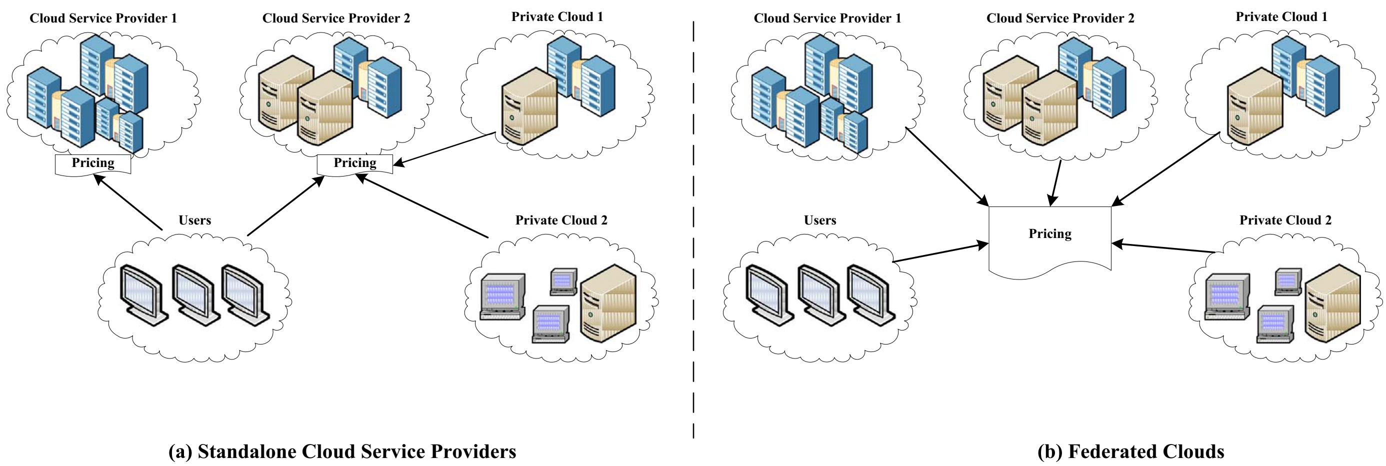

Fig. 2. Pricing in (a) single cloud and (b) federated cloud.

trading market consisted of cloud coordinators, cloud exchangers, and user agents. The authors proposed a dynamic pricing strategy in which the cloud exchangers sent the resource prices to the cloud's coordinators and obtained transaction demand from the cloud coordinators continuously until equilibrium between demand and supply was reached. The price discovery in the pricing method was performed many times, but the equilibrium price could not be achieved.

(Toosi et al. (2012) and Qanbari et al. (2014) presented a financial option-based pricing mechanism for a federation cloud environment to mitigate the risks of paying extra money in case of service-level agreement (SLA) violation. A financial option (Hull, 1997) in this case was a contract undertaken for an upcoming future transaction between two CSPs. In the proposed model, a cloud provider purchased multiple backup on-demand options from other providers to maintain the SLA with its customer because of unforeseen scenarios caused in its resource pool. The seller was compelled to fulfill the demand, and the risk of resource unavailability was mitigated. Moreover, the buying option contract could protect the provider against a high deviation in the market price.

Pricing and capacity planning were both considered by Roh et al. (2013) in a consumer-driven federation of geographically distributed clouds. The resource pricing problem between SaaS and CSPs providers was devised as a competitive price/QoS Rosen's concave game (Rosen, 1965) with mild assumptions. In addition, the equilibrium study investigated two solution concepts for understanding the game, namely, Rosen's normalized equilibrium point generalized from Nash equilibrium (NE) (Damme, 2000) and a two-stage Stackelberg equilibrium (Stackelberg, 1934). The payoff function was designed using log-utility fairness function and M/M/1 queuing model (Gross et al., 2008), which ensured fairness in geographic diversity and theoretical tractability. However, this function did not account for the optimal behavior of the SaaS providers.

Similar to the authors in Roh et al. (2013), the authors in Pal and Hui (2013) examined the pricing models and capacity planning in a multi-provider marketplace, where several cloud providers offered the same application type. They analyzed three game-theoretic prices—QoS models. In the first one, the QoS guarantees were previously specified, and the CSPs contended with one another for the best prices. In the second one, the CSPs contended for QoS and price simultaneously. The third game was analyzed at the same time as the second one, and the price ranges were previously specified while the providers competed for QoS range. The authors provided proof of the existence of a unique pure NE strategy in the first and third games. For the second game, the convergence of the NE implied that no unique NE existed, and the largest equilibrium was then found. The authors followed their analysis with an examination of the optimal provisioning of resources in cloud networks once the QoS levels and equilibrium price for each provider had been determined. They provided a framework through which resources could be provisioned in the cloud in such a way that overprovisioning was minimized dynamically. Their analysis was limited to single-application type.

The sustainable development of open sharing and better resource utilization was promoted in Fan et al. (2013) using a market mechanism for a federation of clouds. A dynamic pricing mechanism was discussed based on the utilization pattern of the shared federated resources. The revenue rate of resources was adjusted dynamically related to their statistical data of the demand and supply association to attain a dynamic variation in resource prices along with the demand and supply relationship. The dynamic pricing model proposed in the current study does not take into account the consumption level and variance in the cost of the same type of resources in different regions and neglects the effect of non-working time on the demand and supply relationship.

The effect of pricing policies on the federation service rejections was evaluated by El Zant and Gagnaire (2014). In their evaluation, they considered the on-demand leases, spot, and advance reservation requests. The pricing policies defined by the authors controlled the federation and its characteristics. They defined three policies as follows: a) Full Federation (FFed): Each CSP aims to handle the requirements locally; b) No Federation (NoFed): Each CSP manages its demands locally. In case of unavailability, federation members serve new requests. In this policy, all types (reserved, on- demand, and spot) can be outsourced; c) Full Federation without Spot [FFed(-S)]: Each CSP tries to handle the requests locally. In case of unavailability, federation members serve new on-demand and reservation-based requests and not spot requests. Requests are discarded when available resources do not exist in the federation or NoFed is present. The authors obtained the following conclusions based on their experimental evaluation: First, federation is a multi-cloud environment because it generates maximum revenues and a smaller number of rejected sessions than the NoFed policy. Second, FFed(-S) is better than FFed because it provides the CSPs better revenues. Third, the reduction in the proportion of spot instances in such a system leads to high CSP profits.

In this study, the authors evaluate the dynamic behavior of users by integrating the heterogeneity of the CSPs, which comprise a cloud broker and public providers (Do et al., 2016). The proposed algorithm considers the fundamental questions on the basis of the selection of cloud services while taking into account the better performance of CSPs. Moreover, the price competition among heterogeneous CSPs is studied using the contract mapping theorem based on two stages. In stage I, to sell the opportunities of services as non-cooperative games, a competition is formulated among the CSPs, in which the service price can be set based on the maximum revenue generation on the CSP side.

Table 2
Summary of resource pricing schemes in federated cloud environment.

Scheme	Objective	Strength	Weakness
(Mihailescu and Teo, 2010a)	Dynamic Pricing to balance the market based on demand and supply		VM pricing under uncertainty was ignored.
(Mihailescu and Teo, 2010b)	Dynamic pricing model with strategic proof to handle the allocation of shared federated resources of multiple types.	Individual rationality, budget balance, and incentive compatibility.	Single point of failure in case the third party auctioneer fails.
(Gomes et al., 2012)	Calculate the equilibrium price in order to enhance cooperation between cloud providers in a federated environment.	Balancing the resource supply and demand.	Sometimes cannot get the equilibrium price.
(Toosi et al., 2012)	Increasing CSPs profit and mitigating the risks of paying extra money in case of SLA violation.	Protection against variation of the market price.	Overprovisioning
(Qanbari et al., 2014)	Influence the pricing elasticity of upcoming supply and demands	Provides the quality price of the Cloud federation options.	Overprovisioning
(Roh et al., 2013)	Considering both pricing and capacity planning together.	Fairness in geographic diversity and theoretical tractability	Does not account for the optimal behaviors of the SaaS providers
(Pal and Hui, 2013)	Pricing and capacity planning when cloud providers provide the same application type.	Over-provisioning is minimized in a dynamic manner	Limited toward single application type.
(Fan et al., 2013)	Improving resource utilization	Dynamic change of resource prices along with the supply-demand relationship	Incentive compatibility and truthfulness are not ensured.
(El Zant and Gagnaire, 2014)	Investigate the impact of pricing on the no. of service rejections.	Proving benefits of federation.	Future contracts are not compared in the investigation
(Do et al., 2016)	Management of price competition of heterogeneous CSPs market	User's requests arrival rate is maximized	Does not consider the SLA issues

At the side of cloud users, a service is selected from the CSP with the best payoff and the minimum delay and price in stage II. Results show that with the proposed model, the arrival rate of resources has a strong effect on CSP and cloud user in terms of utilities, prices, and equilibrium at the CSP side and the arrival rates and user cost at the cloud user side. The limitation of this study is that it does not focus on the SLA issues.

3.1.1. Remarks on resource pricing in federated cloud computing environment

From our analysis of the different federated resource pricing schemes presented above, the association between SLA and pricing model is not clear, and incomplete information exists about the resources of the federated environment. We present the most recent available pricing models to qualitatively measure their applicability and relevance. However, most of them have a bias and do not work in improving the overall utility of the federation. In addition, an individual model does not fit all potential scenarios because of the varying nature of the business objectives and enterprise policies of the federation members. Our analysis also indicates that the functions and features of auditing and accounting are insufficiently included in these models because of the distributed administration of the federation to follow up legal requirements. Several open issues listed below still need to be resolved.

- The effect and overhead of pricing model on the multi-tier hierarchy of the federation of CSPs should be evaluated.
- QoS-differentiated pricing schemes are not yet considered for federated setup.
- In case of ad hoc federation with no prior agreement among the federation members, a malicious member that reveals untruthful resource prices to the marketplace can comprise the efficiency of the pricing model.
- Pricing model should consider the workflow characteristics of composite web services.
- The effect of utilizing business intelligence and integration services should be investigated to analyze the federated cloud marketplace and price predictions for handling misreporting resource bidding functions (Chang, 2014; Wills et al., 2012).

A summary of several resource pricing schemes is given in Table 2. Table 3 lists a few performance metrics identified from the federated resource pricing schemes.

3.2. Resource discovery

The resource discovery function describes how a CSP exposes its resources and service to enable other CSPs in the federation to find these resources and services for automating the resource selection process and ensuring easy use of services, thereby complying with requests. The responsibility of resource discovery in the federated environment is extended to handle physical and geographical proximities and the costly inter-domain traffic of resources. Fig. 3 shows generic resource and service discovery mechanisms used in federated cloud environments.

Decentralized resource discovery in inter-cloud systems by employing a meta-brokering component was proposed in Sotiriadis et al. (2012). Meta-broker implied the generation of unique transient meta-brokers for every request and discovery service submission. Their proposed solution was based on clustering resources by matchmaking service specification instead of such characteristics as weights or distances of interconnection links. In their proposed procedure, the cloud set a local broker that accessed the DC information and generated a meta-broker. The meta-brokers then accessed the directory of meta-brokers and forwarded requests to interconnected CSPs for resource discovery. The approach was fault resilient and effective in reducing the discovery time. However, the transparency of data exchange among the meta-brokers could affect the privacy of the internal characteristics of a CSP.

A constraint-based resource discovery model was proposed for multi-provider cloud environments in Wright et al. (2012) . The proposed model selects the cloud resources from the pool of resources by using a two-stage resource discovery approach. In the first stage, feasible resources that met all complicated requirements (such as location) were identified. In the second stage, appropriate heuristics in terms of performance or cost, were used to sort out the primary sets of resources. Once the finest resource's set was identified, the resources could be managed and instantiated via a cloud management API ("Apache jclouds® :: Home," n.d., "Deltacloud API," n.d.; Querna, 2010).

A scalable gossip-based hybrid multi-attribute overlay (GHMO) was proposed for the discovery of resources in federated clouds in Le et al. (2012). In the proposed approach, the authors mapped the VMs of federated cloud environments into resource nodes in peer-to-peer (P2P) overlay. With the use of the gossiping protocol, a structured

Table 3
Performance Metrics for resource pricing in federated cloud.

Scheme	Scalability	Accuracy	Fairness	Deployment ease	Management overhead	Experimental platform
(Mihailescu and Teo, 2010a) (Mihailescu and Teo, 2010b)	Yes	No	Yes	No	No	FreePastry (Rowstron and Druschel, 2001)
(Gomes et al., 2012)	No	No	No	No	Yes	CloudSim (Calheiros et al., 2011)
(Toosi et al., 2012)	Yes	No	No	No	Yes	CloudSim (Calheiros et al., 2011)
(Qanbari et al., 2014)	No	Yes	No	No	No	CloudSim (Calheiros et al., 2011)
(Roh et al., 2013)	Yes	Yes	Yes	No	Yes	Formal Analysis
(Pal and Hui, 2013)	No	Yes	No	No	Yes	Formal Analysis
(Fan et al., 2013)	Yes	Yes	No	No	Yes	Custom
(El Zant and Gagnaire, 2014)	No	Yes	Yes	Yes	No	Custom
(Do et al., 2016)	Yes	Yes	No	Yes	No	Formal Analysis

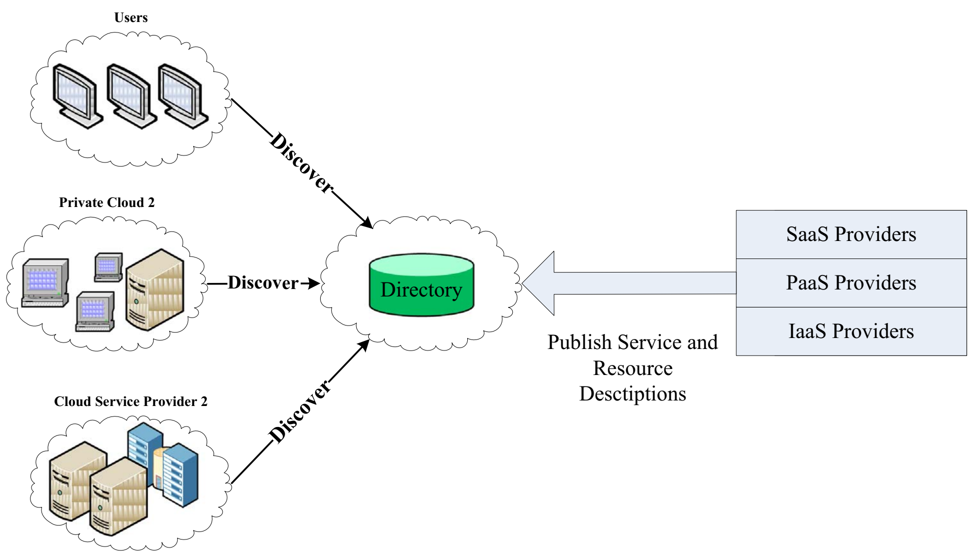

Fig. 3. A generic resource discovery mechanisms in federated cloud.

cluster overlay was constructed with a particular resource vector of nodes whereas the nodes have managed the list of unstructured neighbors. Moreover, with the use of underlying hybrid overlay the multi-attribute routing algorithm was considered in this study. In addition, to enhance the efficiency, GHMO presents the possible routing ranges and improves the selection strategy of neighbor nodes that decrease the routing cost in the search process of multi-attributes.

Resource discovery in a disseminated cloud environment was proposed with the concept of local multi-valued hash table in Khethavath et al. (2013). The key idea was to use two different hash tables, namely, one for finding the location of node and one for resource discovery. The Kademlia protocol (Maymounkov and Mazieres, 2002) was used to find the node location. K-buckets in the routing table were used to store the routing information. Resource discovery in the proposed model was conducted by routing to the adjacent nodes and discovering all the possible nodes that satisfied the consumer demand of the required resources. Peers with similar resources were stored under the same key in the hash table. Thus, a resource request could be mapped to many different node locations.

On the basis of an unstructured overlay (Chung et al., 2013) proposed a direction-aware fully distributed resource discovery strategy to prevent a single point of failure or communication bottleneck in the resource discovery process of large-scale cloud networks. In the proposed system, a resource expressed its characteristic attributes, such as memory capacity and system load. Resource discovery in such a distributed resource sharing system needed to determine the discovery queries with the desired set of pairs of desired values and attributes. Consequently, the characteristic-based and direction-aware routing strategy discovered the requested resource by looking in the local directory first. If no local resource that fulfills the request requirement was found, then the query was forwarded to neighbors for inquiring from their local directory for the requested resource. Upon the reception of a request, the neighbor looked up its local resource indexing information and responded back to the requesting node in case a similar resource was found; otherwise, the request was broadcasted to other neighbors until the required resource was found or a stop condition was reached.

The components for resource mapping and discovery over a federated multi-domain of the EU Project Networking innovations Over Virtualized Infrastructures (NOVI) (Papagianni, n.d.) were presented in Pittaras et al. (2014). The resource discovery framework was powered by the NOVI Information Model (van der Ham et al., 2015) based on Web Ontology Language (McGuinnes and Van Harmelen, 2004) to provide abstraction and semantics for interoperability in the federated virtualized platforms. Semantic expertise facilitated the reasoning while selecting services and resources improved the accuracy of resource discovery.

The two discussed components for distributed resource allocation and discovery, namely, Intelligent Resource Mapping (IRM) and Resource Information Service (RIS), in the NOVI Service Layer enabled each platform to determine resources in RIS's domains, assisted them in deciding which platform would fulfill a VN request or part of it in a cost-efficient way (IRM), and embedded the VN (partial) request in the underlying platform (RIS/IRM).

A set of nature-inspired algorithms and a self-organizing P2P framework and to improve the efficiency of collective discovery of the basic cloud services required for the composition of a workflow were studied in Mastroianni and Papuzzo (2014). The main focus was to

Table 4
Summary of resource discovery schemes in federated cloud environment.

Scheme	Objective	Strength	Weakness
(Sotiriadis et al., 2012)	A fault tolerant and real-time resource discovery method for the federated cloud environment.	The approach is fault resilient and effective in reducing the discovery time.	Transparency of data exchange among the meta-brokers can leak the privacy of internal characteristics of a CSP.
(Wright et al., 2012)	To design a cost and performance aware resource discovery mechanisms for the multi-cloud environment.	Constraint-based optimization limits the search space based on hard and soft requirements.	Scalability based on large number of CSPs.
(Le et al., 2012)	To reduce the routing costs by an improved neighbor selection strategy for discovering resources among federated clouds.	Expanding feasible ranges of routing to enhance the efficiency of multi-attribute search.	Minimization of search performance of GHMO when the resource attributes are maximized.
(Khethavath et al., 2013)	To design a multi-valued distributed hash tables for efficient resource discovery.	Decentralization and decoupling the location and discovery	Experiments are not conducted in order to explain effectiveness as compared to existing schemes.
(Chung et al., 2013)	To resolve the resource discovery queries with less overhead and higher efficiency based on network traffic.	Handling the communication bottleneck by reducing the no. of communicating nodes.	The increase in discovery time as the search is a linear fashion.
(Pittaras et al., 2014)	To design a semantic-based distributed virtual network and resource discovery mapping framework over the NOVI-federated test beds.	Controlling the information regarding infrastructure while enhancing the scalability of the embedding process.	The mechanism is designed for large scientific testbeds and is less practical for a federation of commercial CSPs.
(Mastroianni and Papuzzo, 2014)	To improve the efficiency of collective discovery of the basic services required to compile a workflow.	Exploiting historical data of service co-occurrence to reduce the response time, processing load and bandwidth.	The overhead due to service relocation and its impact upon the cost is not studied.
(Pham et al., 2016)	To improve the datacenters efficiency while the placement of VM	Reduce the operation cost and network traffic cost	Scalability is not considered

spatially cluster service descriptors on the basis of the co-use history of corresponding services. In other words, the services that were frequently executed in the same workflow were located in the controlled network's region. The service's relocation was achieved by using the mobile agents where the service descriptors are moved over the network, and was supported with regard to the service's co-occurrence in past workflow instances by using the statistical information. This relocation allowed search messages to find most or all the required services on hosts to minimize the response time experienced by the consumers and save the bandwidth and processing load.

A novel framework was designed in Pham et al. (2016) for allocating resources in the cloud. The heterogeneous nature of tasks was considered in this framework. The authors proposed the network traffic cost and dubbed a joint operation cost framework, which efficiently managed the set of active serves while allocating VMs. They implemented Gibbs sampling and matching theory to minimize the objective function. The main contribution of that study was that it reduced the operational and network traffic costs.

3.2.1. Remarks on resource discovery in federated cloud computing environments

The literature has indicated that the resource discovery solutions for federated cloud are mostly inspired by P2P systems given the autonomic nature of the federation members. Moreover, several of the discovery solutions employ a central brokerage for the discovery process. In addition to the issues of P2P resource discovery mechanisms discussed in Meshkova et al. (2008), a few open issues pertaining to the efficiency of federated cloud resource discovery are listed below.

- Controllable resource advertising protocols taking the peculiarities of the federated clouds should be designed.
- The semantic description of resources should be standardized to enhance cross-domain discovery and interoperability.
- Runtime SLA negotiation in ad hoc federation should be empowered by first discovering and then negotiating, but this could be turned into an attack tool. Thus, a trusted third party can be involved, thereby leading to the use of resource discovery as a service.
- QoS-differentiated resource discovery is an interesting aspect in a large federation with numerous users.

Table 4 presents a summary of the reviewed federated resource discovery schemes. Table 5 outlines several performance metrics derived from the reviewed federated resource discovery schemes.

3.3. Resource selection

The resource selection process finds a configuration that fulfills all user requirements and optimization of the infrastructure. Selecting worthy resources from a federated resource set is difficult because of the different requirements relevant to the provider, the high algorithm complexity, and dynamicity. The cornerstone of the resource or service selection is an optimization algorithm that considers all variables influencing the allocation. The selection process should also consider behavioral aspects to maintain a user satisfaction level. A general survey on selection solutions for federated infrastructures is presented below.

An index structure was designed by Sundareswaran et al. (2012) based on B+-tree (Bayer and Unterauer, 1977) to simplify the process of information insertion and retrieval for CSPs. In the proposed structure, different properties, such as security, service type, pricing units and measurement, and QoS had precise positions to be considered and stored. Service vendors with the same characteristics should be stored together in adjacent rows to increase the speed at which the information management operators are executed and appropriate vendor queries could be found. The researchers also proposed a query algorithm based on a designed structure to search the provider database for the best vendors. The proposed architecture was compared with a brute-force search algorithm and showed almost 100 times better execution speed for solving the cloud computing service composition problem with 10,000 service providers.

A high-level generic brokerage architecture to find the most worthy CSP fulfilling the service requirements of the user in terms of non-functional and functional SLA parameters was proposed in Jrad et al. (2012). The proposed architecture integrated a brokerage-based technology for assisting the user in SLA negotiation and finding the best provider for his service needs with respect to specified SLA.

Table 5
Performance Metrics for resource discovery in federated cloud.

Scheme	Directory	Push mode	Pull mode	Caching	Service description	Architecture
(Sotiriadis et al., 2012)	No	Yes	No	No	–	P2P
(Wright et al., 2012)	No	Yes	No	Yes	RDF (Klyne and Carroll, 2004)	Client-Server
(Le et al., 2012)	No	No	Yes	No	–	P2P
(Khethavath et al., 2013)	Yes	No	Yes	No	RDF (Klyne and Carroll, 2004)	P2P
(Chung et al., 2013)	No	Yes	No	No	–	P2P
(Pittaras et al., 2014)	Yes	No	Yes	No	RDF (Klyne and Carroll, 2004)	P2P
(Mastroianni and Papuzzo, 2014)	No	Yes	No	No	–	P2P
(Pham et al., 2016)	Yes	No	No	No	–	Client-Server

The way software agents are employed for cloud resource/service selection in a multi-cloud marketplace, where the consumer selects the best cloud provider based on their utility function, was explored in Sim (2012, 2008, 2006). In their work, they proposed a negotiation protocol based on Rubinstein's alternating offer protocol (Rubinstein, 1982) and a negotiation strategy based on the functions of time, opportunity, and competitiveness for multiple consumer–broker agents negotiating simultaneously. Furthermore, they proposed service capability tables (SCTs) to store their services and the cloud agent's list.

The coordination of self-organizing participants in a multi-cloud environment for automating a service selection in the presence of incomplete information about CSPs and their resources was investigated in Gutierrez-Garcia and Sim (2013). To handle this problem, a collection of two agent-based distributed problem-solving techniques, namely, SCTs and a semi-recursive contract net protocol, was integrated and devised into agent behavior to cope with (i) service selection based on dynamic service fees and (ii) incomplete knowledge about the existence and location of service providers and the cloud resources they offer. An agent-based cloud service composition testbed was implemented to support persistent, one-time, vertical, and horizontal cloud service compositions. Mechanisms to update and create service compositions based on constantly changing consumer needs were designed using self-organizing agents as building blocks.

In Son (2013), a resource selection decision maker (RSDM) was presented. The proposed decision maker listed the suggested resource providers and their resources by analyzing user demands. Users initially provided their requirements for the cloud service. According to these requirements, the RSDM retrieved all resources that match the requirements from a cloud information database. Once all candidate resource providers and their service types were retrieved, the estimated price was calculated. After the calculation of prices for each provider and service, the provider list was recorded by price and given to the user. Each item of the recommended list comprised information, including the resources to be allocated, name of the cloud provider, contract period, service type, and expected price.

The selection of a suitable CSP in the inter-cloud of computing services for fulfilling user tasks when no relevant resources are available in the public and private clouds was analyzed in Vilutis et al., (2013). The Quality of Grid Services QoGS (Plestys et al., 2007) method was selected to determine the appropriate CSP. However, the QoGS method works appropriately only if the correct set of weighted coefficients (SoWC) is selected. Therefore, an algorithm was designed for selecting the best SoWC. Experimental results showed that the proposed methodology minimized the workload of inter-cloud by test tasks significantly.

The selection of cloud resource in a federated cloud environment was divided into two subproblems by Jaikar and Noh (2014), namely, DC selection and physical machine selection. The DC collection played a vital role in improving the performance and reducing the cost. An algorithm for selecting a DC in a federated cloud computing environment was presented. The approach was validated using a Cloud Analyst toolkit (Wickremasinghe et al., 2010). Results indicated that the DC selection algorithm offered considerable performance gains with respect to throughput, cost, and response time.

A SLA-based hierarchical service selection was presented for multi-cloud environments in Farokhi et al. (2014). In their efforts, the authors adopted the idea of the algorithm presented in Yau and Yin (2011) and developed it to support service selection for a cloud composite service and to cover all the functional and non-functional parameters of inter-cloud SLAs. The architecture and phases involved in the selection process were based on prospect theory to evaluate the infrastructure services on the basis of the given SLAs and the degree of user satisfaction. The evaluation and a comparison of the utility-based matching algorithm showed that the approach effectively selected a set of services for the composition that satisfied SLA parameters.

On the basis of multi-attribute trust value evaluation cloud service selection was studied by Pei (2015). Their trust value estimation was based on two trust's characteristics, namely, reputation-based trust and perception-based trust, in which the trust facts were recorded on the trust reputation base and value base. Users could obtain the trust facts from the two bases and then apply the evidential reasoning approach to achieve the final trust results. After the service users used the service, they would give their feedback evaluation to the cloud system, and this evaluation would be stored in the trust value base and reputation value base for other users to generate the indirect trust evidence. In the proposed framework, the trust value was produced from both the personalized indirect trust evidence and direct trust evidence, which was reliable with the service's requirement of users.

3.3.1. Remarks on resource selection in federated cloud computing environment

The dynamic changes in the resource utilization of the resources in a federated environment and the changes in the workload characteristics turn the resource selection into an iterative repetitive task considering user-specified functional and non-functional constraints. However, the task is difficult and has the following open issues, which should be addressed to make the resource selection in a multi-cloud environment feasible:

- Monitoring data should be integrated as historical feedback to judge the credibility and effectiveness of the resources to be considered for selection.
- Workflow modeling, breakdown, and mapping should be enhanced to utilize the full potential of the federated environment by selecting resources from the federated resource pool for individual components of the workflow according to some constraints.
- Resource selection mechanism should consider the combination of networking factors and failure and energy indexes.
- The effect of the selected resources on the utility of the CSP in the federation should be evaluated to verify whether this selection really improves the utility in the long run.

A summary of the reviewed resource selection schemes in a federated cloud environment is given in Table 6. Table 7 lists the extracted performance metrics from the federated resource selection schemes.

Table 6
Summary of resource selection schemes in a federated cloud environment.

Scheme	Objective	Strength	Weakness
(Sundareswaran et al., 2012)	To simplify and increase the speed of searching the CSP database for the best vendor selection.	100 times faster than brute-force search algorithm.	No opportunity for users to negotiate some terms of the SLAs.
(Jrad et al., 2012)	To find the most worthy CSPs in order to fulfill the user's service requirements non-functional and functional SLA parameters.	Handling the interoperability and heterogeneity.	No Experimental evaluation to show its efficacy compared to existing schemes.
(Sim, 2012, 2008, 2006)	How software agents are employed for cloud resource selection in a multi-cloud marketplace where consumer selects the best cloud provider based on their utility function.	Concurrent negotiating agents for the best SP selection.	Changing user requirements are not captured.
(Gutierrez-Garcia and Sim, 2013)	Automating the service selection in the presence of incomplete information about cloud providers and their services.	Constantly changing consumer's needs are captured.	In the case of service migrations maintaining the agents and can be a difficult problem.
(Son, 2013)	To select the best provider in term of cost and requirements of the user.	The system components are pluggable with other programs and not depended on each other	The systems need human interaction to populate the database of the candidate CSPs and their characteristics.
(Vilutis et al., 2013)	Selection of suitable cloud in inter-cloud of computing services when there are no relevant resources available in the public and private cloud.	Minimizing the number of test tasks for quantifying the CSPs.	Networking factors are not used in the selection of a CSP.
(Jaikar and Noh, 2014)	To support the dynamic load while selecting the best position for allocating the request to attain the better performance.	Minimize the cost with acceptable performance	The failure index and energy consumption index of data centers are not included in decision making.
(Farokhi et al., 2014)	To automatically select infrastructure services for SaaS provider such that the SLA claims of the SaaS provider for their customers are captured.	Cover functional and non-functional parameters of Inter-Cloud SLAs.	No SLA violation detection and penalty in case of violations.
(Pei, 2015)	To select trustworthy cloud services for cloud users.	Feedback driven trust basis.	The granularity of the historical data for decision making is not considered so that outdated history does not impact in decision making.

(continued on next page)

Table 6 *(continued)*

3.4. Resource monitoring

The function of cloud resource monitoring is to provide wide monitoring information data about service management and infrastructure, such as access control, service elasticity, service billing, and SLA management (Bernsmed et al., 2011; Carlini et al., 2011). Customers are provided with monitoring data about their running services deployed in federated clouds. In the federated environment, resource monitoring is important for CSPs to maintain the federation and fairness in the distribution of revenue generated by the cloud clients.

Massonet et al. (2011) showed how an existing solution of federated cloud monitoring can be employed for monitoring data locality without cooperating cloud isolation to allow users to monitor the data location. In their proposed approach, collaboration among the cloud infrastructure provider, service provider, and the cloud user was exploited for monitoring the data location. The infrastructure provider monitored the VMs on behalf of the service provider and made the infrastructure-level monitoring data available to the service provider. From these monitoring data, the service provider produced the auditing logs needed for compliance auditing.

An interface for inter-site monitoring of VM's resources, both on application- and infrastructure-specific levels in a federated cloud environment, was proposed in Elmroth and Larsson (2009). The proposed scheme was compatible with grid-monitoring solution (Tierney et al., 2002). The basic idea laid in the augmentation of VM descriptors with additional configuration parameters with regard to monitoring, such as the monitoring interval length, the delivery interval length, the format of delivery, i.e., processed or raw, and the public key of the primary site used to facilitate the encryption of monitoring data.

Monitoring and obtaining information about the resources of federated future Internet research and experimentation testbeds were studied by Al-Hazmi and Magedanz (2012). The authors introduced a flexible monitoring system called MOST4FIRE, which provided a flexible and an easy way for experimenters to analyze low-level resources to high-level services across heterogeneous domains. This model provided monitoring at various levels, namely, platforms, virtualized environments (VMs), physical infrastructures, and services. The cross-level monitoring system provided customers with many resources, in terms of spectrum analyzers, active and passive measurement probes, data converters and transporters, data view resources (graphical user interfaces, APIs, and visualization tools), and data collection resources (aggregators, repositories, and collectors). Users could then identify the resources to be monitored. Unified representation of the monitoring data gathered from various testbeds was not discussed, and the effect on the resource consumption was not analyzed.

Cloud Kick Cloud (2011) is a management platform for a multi-cloud environment with a wide collection of both low- and high-level monitoring metrics and features along with the capability of custom plugin addition. Monitoring information gathered from the multi-cloud environment can be visualized in real time, thereby allowing the user to be informed on time through the alert systems (e.g., via e-mail or SMS) (Caron et al., 2012). The platform addresses mainly the adaptability and scalability of the monitoring solution. Similarly, Nimsoft Monitoring Solution (NMS) (CA Technologies, 2013) is able to monitor the DCs of both private and public clouds. This platform renders a

Table 7
Performance Metrics for resource selection in federated cloud.

Scheme	Cost	QoS	Load-aware	Energy-aware	Feedback-driven	Selection method	Experimental platform
(Sundareswaran et al., 2012)	No	Yes	No	No	No	Search based	Custom C Based
(Jrad et al., 2012)	No	Yes	No	No	Yes	Simple Match Making	CloudSim (Calheiros et al., 2011)+OCCI4Java (Mohamed et al., n.d.)
(Sim, 2012, 2008, 2006)	Yes	No	No	No	No	Negotiation	JADE (Bellifemine et al., 2005)
(Gutierrez-Garcia and Sim, 2013)	Yes	No	Yes	No	No	Negotiation	JADE (Bellifemine et al., 2005)
(Son, 2013)	Yes	No	No	No	No	Search based	CloudSim (Calheiros et al., 2011)
(Vilutis et al., 2013)	No	Yes	Yes	No	No	Weighted Rank	Not mentioned
(Jaikar and Noh, 2014)	Yes	Yes	Yes	No	No	Matrix Based	Cloud Analyst (Wickremasinghe et al., 2010)
(Farokhi et al., 2014)	No	Yes	No	No	Yes	Prospect Based	Custom Java Based
(Pei, 2015)	No	Yes	No	No	Yes	Evidential Reasoning	Formal analysis

unified view of the monitoring information captured from both infrastructure and service providers, such as Google Apps, Amazon, and Rackspace Cloud. NMS is a suitable tool for monitoring SLAs (Dastjerdi et al., 2012), and it addresses the comprehensiveness and scalability features of a federated cloud monitoring solution.

Kertesz et al. (2013, 2012) proposed an architecture that utilized a sophisticated service monitoring approach and offered federated cloud management to estimate the status of basic cloud health. The proposed SLA Monitor and the minimal metric monitoring service approach enabled autonomous service provisioning, enhanced the inter-cloud service executions and provider selection, and could measure the reliability and availability of the allocated services in different providers. The cost-effective and transparent operations of the commercial cloud along with the ability to simultaneously analyze public and private clouds were the major concern of this approach. However, the expansiveness of the monitoring process itself and the monitoring of the broker was not explored.

A cross-layer monitoring framework was presented for multi-cloud service-based requests in Zeginis et al. (2013). The framework integrated monitoring systems for each cloud layer and across cloud providers, and used the component model and an event to explain observed events and their cloud source components. Moreover, the proposed evaluation framework within the different deployment situations showed that a multi-cloud setup is minimally affected by the time-series database's performance scales. The limitation of the study is that the framework model was not formally validated.

Authors in Smit et al. (2013) aimed to collect monitoring information in an integrated environment of private and public clouds at PaaS, SaaS, and IaaS levels. The proposed monitoring architecture was based on a publish–subscribe mechanism, and data collection was performed using a set of custom-defined plugins that communicated with a monitoring manager through the standard Java Remote Invocation. The presented architecture was based on stream processing to provide near-real-time in-memory access to the monitoring platform based on the publish–subscribe mechanism. Amazon Cloud Watch metrics were integrated with private cloud execution based on Open Stack.

A complete distributed monitoring architecture for large-scale distributed systems, such as federated clouds, was presented in Povedano-Molina et al. (2013) to propagate the resource monitoring information. The proposed Distributed Architecture for Resource manaGement and mOnitoring in cloudS (DARGOS) is shown in Fig. 4. DARGOS took into account the customization of both the

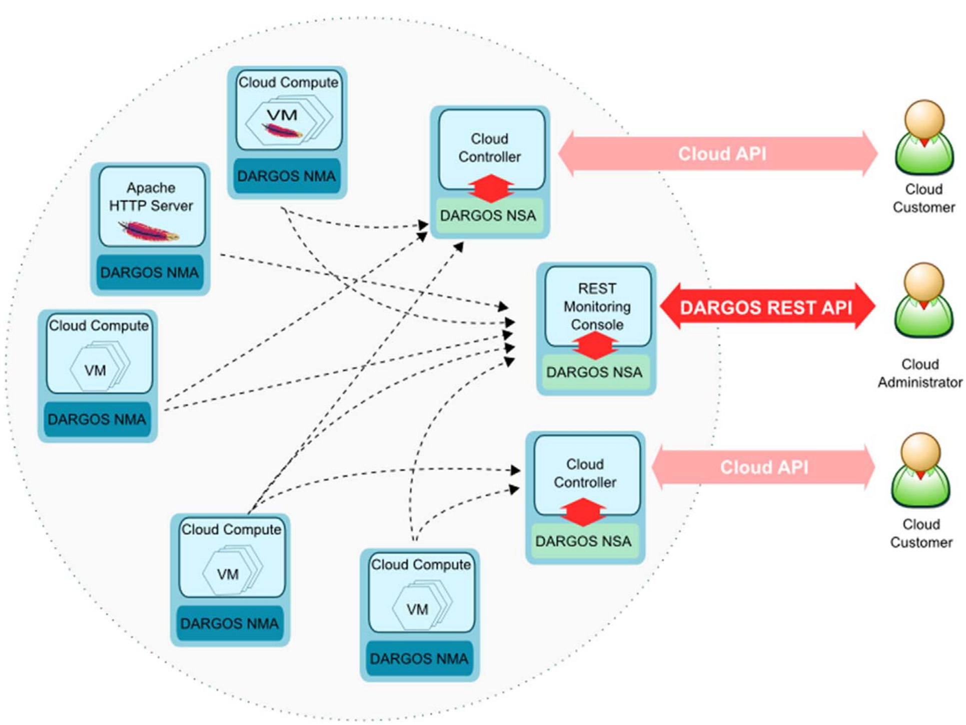

Fig. 4. DARGOS architecture (source (Povedano-Molina et al., 2013)).

Table 8
Summary of resource monitoring schemes in federated cloud environment.

Scheme	Objective	Strength	Weakness
(Massonet et al., 2011)	How an existing monitoring solution for federated cloud can be employed for monitoring data location without compromising cloud isolation.	The monitoring process is performed by individual infrastructure providers. This decentralization improves the fault resilience and resource utilization of the monitoring process.	The monitoring process can be compromised as the infrastructure providers monitor VMs for service providers.
(Elmroth and Larsson, 2009)	Intra- and inter-site monitoring of virtual machine resource, both on an application and on an infrastructure level.	Backward compatibility with grid monitoring architectures.	The augmentation of VM descriptors.
(Al-Hazmi and Magedanz, 2012)	Monitoring of resources from low-level to the high-level services over the heterogeneous domains	Passive and active measurement probes.	Unified representation of monitoring data gathered from multiple infrastructures.
(Cloud Kick Cloud, 2011)	A scalable and adaptive commercial management platform based on alerting and virtual monitoring mechanisms within the multi-cloud.	Customizable and real-time delivery of monitoring data to the users.	Interoperability with another CSP is not addressed.
(CA Technologies, 2013)	To provide a unified monitoring dashboard for monitoring SLAs of different CSPs.	A single and comprehensive view of the IT infrastructures and services.	Customization over the monitoring events is not addressed.
(Kertesz et al., 2013, 2012)	Transparent and cost-effective simultaneous monitoring of public and private cloud infrastructures	Service level agreement monitoring and minimal metric between different providers delivered as service.	The expensiveness of the monitoring process itself and monitoring of the broker is not explored.
(Zeginis et al., 2013)	A cross-layer monitoring framework for service based applications deployed in the environment of multi-cloud.	Exploiting the dependencies among layers for gathering monitoring information	The framework model is not formally validated and neither the adaptation procedure due to infrastructure variations is defined.
(Smit et al., 2013)	Collect monitoring information in an integrated environment of private and public clouds at PaaS, IaaS, and SaaS levels.	Provide near-real-time, in-memory access to the monitoring platform based in publish-subscribe mechanisms.	Technology specific, in the case of SaaS and PaaS only java specific platforms, can be monitored.
(Povedano-Molina et al., 2013)	A complete distributed and highly customizable, interoperable and efficient cloud monitoring architecture for large scale distributed systems.	Customizing both granularity and frequency of monitoring events and avoiding vendor “lock-in”.	The multicasting of monitoring data can result in network performance bottlenecks

(continued on next page)

frequency and granularity of the monitoring probes according to tenant and service requirements. Moreover, they introduced a data distribution service for easy interoperability with legacy requests and avoiding vendor lock-in. The data distribution service embedded flexible QoS strategies within the monitoring task to provide timely, reliable, and efficient delivery of cloud monitoring data. In addition, these policies hide the communication-related issues of the application.

3.4.1. Remarks on resource monitoring in federated cloud computing environment

From our analysis of the literature on monitoring solutions for federated cloud environments, the capability of monitoring cross-domain services has been regarded as a privacy and security risk and monitoring as an attack tool (Ristenpart and Tromer, 2009). This fact leads to a no-production-level monitoring solution with a federation of cloud members having different business objectives and enterprise policies. The expensiveness of the monitoring solutions and their effects on the application QoS are not explored for most of the solutions. Open issues pertaining to the efficacy of resource monitoring in federated environments are listed below.

- Standardization is lacking when logical or physical domain boundaries are crossed, and monitoring activities is a challenge because of vendor lock-ins and heterogeneous infrastructures and architecture.
- Architectural standardization efforts should be made to standardize APIs for gathering monitoring data from CSPs.
- Energy monitoring of federated CSPs should be conducted to encourage efficient, green cloud computing by scheduling and rescheduling an application according to the energy consumption index monitored to reduce the energy consumption.
- Cross-domain data leakage of applications and the internal configurations of a cloud member need to be addressed to enable third-party monitoring and unified monitoring of federated environments.
- An autonomous monitoring tool for validation and performance measuring of heterogeneous application sets deployed in a federated cloud environment is required.
- No monitoring data of single or federated cloud environment are publicly available, and no workload traces of the monitoring solutions themselves exist to analyze the data by statistical tools to acquire more insight into the monitoring process.

A summary of federated resource monitoring schemes is presented in Table 8. Table 9 lists the performance metrics for the federated resource monitoring schemes.

3.5. Resource allocation

Resource allocation is integral for obliging unpredictable resource requirements and capital return in cloud federation. In the context of federated clouds, application developers can lease resources in a pay-per-use manner from multiple geographically distributed CSPs to minimize the cost and SLA violation, and to enhance the application availability and fault tolerance. A general review of several resource allocation and scheduling strategies for federated cloud environments is presented below.

In Malet and Pietzuch (2010), middleware for cloud management was presented to migrate part of user’s services (represented by

Table 9
Performance Metrics for resource monitoring in federated cloud.

Scheme	Scalability	Elasticity	Adaptability	Timeliness	Automaticity	Comprehensiveness	Extensibility	Intrusiveness
(Massonet et al., 2011)	Yes	Yes	No	No	Yes	No	No	No
(Elmroth and Larsson, 2009)	No	No	No	Yes	Yes	No	No	No
(Al-Hazmi and Magedanz, 2012)	No	Yes	No	No	No	Yes		No
(Cloud Kick Cloud, 2011)	Yes	No	Yes	No	No	No	No	No
(CA Technologies, 2013)	No	No	No	Yes	No	Yes	No	No
(Kertesz et al., 2013, 2012)	Yes	No	No	No	Yes	No	Yes	No
(Zeginis et al., 2013)	Yes	No	Yes	Yes	No	No	Yes	No
(Smit et al., 2013)	Yes	No	Yes	Yes	No	No	Yes	No
(Povedano-Molina et al., 2013)	No	No	Yes	No	No	No	Yes	Yes

number of VMs) among DC to manage the workload at the DC and to minimize total response times. On the basis of the workload monitoring in DC, the middleware initiated VM migration to shift the components of application closer to the customer. The proposed approach was mainly designed for multiple DCs under a single cloud provider, but the formulation was still worthwhile for multiple cloud providers.

Authors in (Ai et al., 2011) considered the deadline-constrained scheduling and resource allocation in a hybrid cloud environment for various composite web services. The authors also took into account the running cost because in a hybrid cloud, the in-house private cloud resources were cheaper than their counterparts from public clouds. They proposed a cooperative coevolutionary genetic algorithm (CCGA). In the proposed CCGA, the cooperation among populations occurred while the individual's fitness (resource component) in a subpopulation was being evaluated. In the population, the fitness value of a specific entity was an estimate of how well it cooperated with different classes to generate good results. The populations worked cooperatively to explain the crises that are guided by the fitness value. This communication among the populations involved a selection of greedy collaborator and the credit assignment based on the fitness value. The performance of their algorithm depended on the credit assignment and the collaborator selection strategy method. However, only one process could be assigned to a machine at a time, thereby preventing the system from using many configurations.

The game-theoretic resource allocation scenario was studied in a federated cloud environment in Hassan et al. (2014, 2011). The authors considered a horizontal dynamic cloud federation (HDCF), in which various CSPs cooperated dynamically for expanding their infrastructure capacity to meet end user QoS to gain economies of scale and to handle heterogeneous cloud resource demands without increasing and enhancing a number of physical resources. In this prospect, the authors proposed game-theoretic cooperative/non-cooperative price-based centralized and distributed resource allocation strategies to ensure that this horizontal cooperation was beneficial for each CSP. The authors formulated two resource provisioning games (i.e., cooperative and non-cooperative) in HDCF platforms to enhance the utility of the federation, specified as the total of the CSP utilities of the buyers. The authors applied a direct search method with multiple startup guesses to determine the best price (Kolda et al., 2003). In both games, a buyer CSP publicized and defined the total amount of virtual resources provided. On the basis of this information, each seller CSP then updated its own strategy to maximize its utility. They concluded the existence of non-unique equilibrium states that produce an undesirable outcome under a non-cooperative game through experimental and theoretical analyses. Under the cooperative setting for resource allocation, the game was scalable and cost effective. The problem with this approach was that it modeled the resources as a single type.

The unpredictable workload fluctuation in a cloud computing environment to reduce the allocation costs in terms of VMs to meet the SLA constraints was studied in Ardagna et al. (2012). They applied the open queuing model for modeling the coordination of multiple cloud providers operating in geographically distributed sites and presented a solution based on capacity allocation and load redirection distributed algorithms acting upon two angles, namely, time scale and workload prediction employing a closed-loop queuing technique with nonlinear optimization models. Furthermore, they investigated and demonstrated their solutions to be close to the solutions found by an oracle with perfect information about future workload. The workload prediction improved content locality. Aside from the comprehensiveness and rigorous evaluation of the work, the proposed method considered only the response time of the server and did not take network latency into account. They also did not study the capacity of the running instances while making the decision.

Di et al. (2013) considered the allocation of resources in a fully distributed self-organizing cloud (SoC), such that both resource consumers and contributors were satisfied with their prior results based on the declaration of their resource. In SoC, each host was deployed with an autonomous VM monitor and a resource state collector to act both as resource contributor and task scheduler. They constructed a VM with the resource distribution concept from the execution nodes to optimize the efficiency of task execution and proposed a novel next-price bidding double-sided strategy based on the traditional second-price bidding to achieve ex-post incentive compatibility. For the resource query protocol, a random index diffusion strategy was adopted to reduce network traffic on query-message propagation. In this study the resource provisioning problem is devised as a convex optimization problem considering task characteristic, user budget demand, and resource availability. A polynomial time algorithm, local optimal VM resource allocation, was also designed to locate the optimal solution. Moreover, the basic idea was to temporarily remove the resource availability constraint and then recursively tune the solution with the constraint of resource availability until a distribution that satisfies this constraint was found. However, the system might be compromised because of the fully distributed and self-organizing nature, given that the fault tolerance and security mechanism were not discussed but were suggested as prominent future directions.

Networking and computing resources were jointly optimized and treated for dynamic allocation of virtual resources to the physical resources inside networked clouds in Papagianni et al. (2013). To manifest the joint resource allocation solution for a federation of cloud network, the federated cloud resource mapping problem was formulated as a mixed integer programming (IP), taking into account the cost efficiency objective, such that QoS for user requests was met. Link mapping was formulated as the corresponding multi-commodity flow problem. Subsequently, a heuristic methodology for efficient mapping of networked cloud resources on shared substrate was proposed. The proposed framework did not consider dynamic heterogeneous infrastructures and environment beyond the conventional Internet (e.g., wireless), which presented further issues owing to wireless environment (e.g., uniqueness of nodes, isolation, and coherence), the stochastic environment of the corresponding resources, and the challenges related with the existence of mobile nodes.

Journal of Network and Computer Applications 77 (2017) 87–105

A fully distributed scheduling scheme was proposed in Palmieri et al. (2013) for the uncoordinated federated cloud environment. The scheduling framework was based on self-organized and independent agents, which did not depend on any centralized control that covered the NE solution, with the potential contradiction between the interests of service provider and client in the cloud environment taken into account. A high performance was gained based on the completion time. This study had reliable results in large organizations where a wide variety of complex tasks can be observed by using the efficient partitioning strategy.

The effect of multi-cloud over a single provider for allocating components of distributed application for a variety of realistic scenarios was studied in Woo and Mirkovic (2014). The distributed cloud application workflow was modeled as a sequence of transactions composed of micro-tasks. The resource allocation for a given application workflow was formulated as the problem of determining a set of resources from the multiple clouds that met the SLA, cost, and performance constraints. Subsequently, an algorithm for resource allocation was proposed to find the best allocation for components of the distributed application over the multi-cloud environment. The algorithm worked by considering all possible allocations for each transaction and selected the one that met the respective SLA, cost, and performance constraints. The algorithm exhaustively searched from the cheapest to the most expensive solution until a viable allocation was found. This exhaustive search could lead to a scalability problem in case of a very large environment.

The issue of scaling out cloud resources while executing CPU-intensive applications with non-proportional cost to performance ratios was resolved for the cost-effective deployment of an application into multiple cloud environments by HoseinyFarahabady et al. (2014). The authors indicated that the degree of performance gain had no strong correlation with the usage cost of cloud resources. They presented fully polynomial-time randomized approximation algorithms to enable the execution of bag-of-tasks with known and unknown running time spanning beyond the private system/cloud (i.e., hybrid cloud) by explicitly taking into account the cost efficiency, that is, the cost-to-performance ratio.

By meeting the peak demand while preserving QoS proactive machine purchasing, cloud federation resolves the problem of achieving economies of scale for IaaS. However, the former is not economic, and the latter is difficult in practice. In Zuo et al., (n.d.), a resource allocation framework was proposed where the providers of IaaS could outsource their tasks to ECs when their own resources were not adequate to fulfill the user's demand. This framework did not require any inter-cloud formal agreement for the federation of cloud. The key challenge was how to assign user tasks to enhance the revenue of the IaaS provider while satisfying the QoS. In this study the problem was devised as an IP model and solved by a self-adaptive learning particle swarm optimization (SLPSO)-based scheduling mechanism for scheduling the inter-cloud resources. In SLPSO, each aspect of a particle represented a particle, and task as a whole represented the priorities of all tasks. This scheme could acquire a high-quality scheduling by adaptively selecting velocity updating strategies to update each particle. The scheduling approach could find the suboptimal or optimal allocation scheme of external and internal resources to greatly improve the profit of IaaS providers and maximize the quality of the scheduling solution.

3.5.1. Remarks on resource allocation in the federated cloud computing environment

Considerable work on resource allocation mechanisms for federated clouds is being conducted, but these mechanisms still need improvement. From our analysis, we find that controlling the effect of reconfiguration cost over the federation and cloud utility is not studied. Another shortcoming is the lack of generality to make the allocation scheme visible for all types of service provisioning. Several open issues

Table 10
Summary of resource allocation schemes in a federated cloud environment.

Scheme	Objective	Strength	Weakness
(Ai et al., 2011)	To handle the deadline-constrained scheduling and resource allocation problem for multiple web services.	The algorithm depends on the credit assignment and the collaborator selection strategy method making it highly adaptable.	Only one process can be assigned to a machine at a time.
(Hassan et al., 2014, 2011)	To meet end-user QoS and economies of scale without increasing and enhancing a number of physical resources.	Maximizing the total utility of the federation.	The problem with this approach is that they model the resources as a single type.
(Ardagna et al., 2012)	To handle the unpredictable workload fluctuation while guaranteeing SLA constraints by the coordination of multiple geographically distributed clouds.	Capacity allocation and load redirection distributed algorithms acting upon two angles, time scale, and workload prediction.	They did not study the capacity of the running instances while decision making and also overlooked the network latency.
(Papagianni et al., 2013)	Networking and computing resources are jointly optimized and treated for dynamically allocating virtual resources to physical resources within cloud network.	Handling both resource mapping and link mapping.	Overlooks the dynamic heterogeneous infrastructures and environments.
(Palmieri et al., 2013)	To take into account the possible contradiction between the interests of service provide r and client in the cloud.	This scheme had great benefits in a huge cloud organization which have maximum number of nodes with a wide varaity of tasks to be served.	Unpredicted situations and deviations in the environment.
(Di et al., 2013)	To manage the social competition relations among resource contributors and consumers where everyone satisfied with its payoff.	Polynomial time complexity to find the best solution and ex-post incentive compatibility.	The fault tolerance and security mechanism is not discussed.
(Woo and Mirkovic, 2014)	To study the benefits of allocating components of a distributed application on multiple public clouds.	Search based technique to meets the set of SLA constraints and achieve the performance and cost constraints.	The search time in case of a very large federation of cloud and large workflows.
(HoseinyFarahabady et al., 2014)	To handle the scaling out of cloud resources while executing CPU-intensive	Fully polynomial-time randomized approximation algorithms for	The scheduling decision does not take into consideration the fault

(continued on next page)

Table 10 *(continued)*

Scheme	Objective	Strength	Weakness
	applications with the non-proportional cost to performance ratios multi-cloud environment.	the task with known and unknown running time.	tolerance.
(Zuo et al., n.d.)	To provision the user's tasks in order to enhance the revenue of IaaS provider while satisfying QoS.	A high-quality scheduling solution by adaptively updating strategies.	The computational complexity of the proposed technique.

should be addressed to achieve efficient resource allocation mechanisms.

- Interoperability among virtualization engines, such as VMware ESX (Esxi, 2016), KVM, and Xen (Xen, 2014), should be investigated to realize seamless flow of data between their local applications and across clouds.
- VM migration across subnets and realization of maintaining network flows by decoupling should be analyzed (Kalim and Gardner, 2013).
- VM behavior modeling and workload of a federated cloud environment should be evaluated to realize the peculiarities of the workload and VM.
- Precise forecasting in a distributed and heterogeneous federated environment, where the common information among the entities is lacking because of the different administrative policies, is needed.
- Security: Inter-VM attacks when a malicious VM is being migrated from a malicious provider.

A summary of several resource allocation schemes is presented in Table 10. Table 11 lists the performance metrics extracted from the federated resource allocation schemes.

3.6. Disaster management

Disaster management and fault tolerance play an important role in restoring organizational data in case of natural hazards or man-made disasters. Disaster management functions enable a system or component to continue normal operation despite hardware/software failures or compromises. In scenarios of federated cloud environment disaster, management functions should be distributed and coordinated among each node of the federated setup to enable a micro-level disaster-aware federated cloud infrastructure, which ensures the QoS level required by members of the federations. A general review of disaster management, fault tolerance, and intrusion prevention for federated cloud environments is presented below.

Wood and Lagar-Cavilla (2011) argued that pipelined synchronous replication can recover an application and its crash point provide generous application overhead restricted by economic considerations, and ensue minimal geographical division between the recovery and primary sites. By efficiently overlapping replication with application processing for multi-tier servers, pipelined synchrony addressed the effect of wide-area network replication latency on performance. Applications realized forward growth while guaranteeing of full consistency for the client-visible state in the disaster event by tracking the cost of the disk modifications that continued to a recovery site all the way to client-directed messages. The proposed solution could maintain these agreements for multi-node servers composed of black-box VMs without the requirement of application modification, thereby resulting in a perfect fit for the arbitrary nature of VM-based cloud hosting.

Once a system and an environment become secure, the second issue is how system operation can be enabled in case of hardware malfunctions or network disruption. In this regard, (Liao and Shen, 2011; Liao et al., 2015; Tran et al., 2015) considered min-cost fault-tolerant resource provisioning in a cloud computing environment by employing a facility location formulation. They believed that the existing models, such as uncapacitated facility location and fault-tolerant facility location (FTFL), were insufficient because both models restricted at most one resource/facility to open at each federated DC/site, and connection leases/requests of a client must be served by different sites. They studied both the constrained fault-tolerant and the unconstrained fault-tolerant approaches generalized from employed greedy, FTFL, primal-dual, and linear optimization techniques. Both approaches allowed multiple client connection requests to be served, and for facilities to open for each site by facilities inside the same site. These approaches were rational because a client could access multiple machines in parallel on the same site. They also presented a star-greedy algorithm with a consistent rate of requests. This algorithm terminated when all client connection requirements were satisfied.

Togawa and Kanenishi (2013) built a disaster recovery (DR) framework against tsunami and earthquake disasters for cloud-based e-learning environments (Chang, 2015a). The proposed system's prototype was designed using a number of private cloud fabrics. To operate a large private fabric these private cloud fabrics were constructed under a virtual private network connection. Moreover, to manage large file systems, a distributed storage system was built for each fabric of private cloud that was managed almost like the same block device. Moreover, when some private cloud fabric does not work because of any problems then the distributed storage system could keep running as one large file system.

A federated cloud architecture is proposed in Luo et al. (2014). This architecture included the technologies of proactive cloud defense for agile and secure development of a cloud. Their proposed federated cloud model consisted of a set of systematic security mechanisms that are seamlessly integrated at the system, network, and application layers in federated cloud environments. The features of the framework included the following: (1) it facilitated early detection of cyberattacks against one layer and deployed early warning sign of an attack to

Table 11
Performance Metrics for resource allocation in the federated cloud.

Scheme	Cost	Scalability	Load balancing	Energy-aware	Makespan	Availability	Fault tolerance
(Ai et al., 2011)	Yes	No	No	No	Yes	No	No
(Hassan et al., 2014, 2011)	Yes	Yes	No	No	No	No	No
(Ardagna et al., 2012)	Yes	No	Yes	No	Yes	No	No
(Papagianni et al., 2013)	Yes	No	No	No	No	No	No
(Palmieri et al., 2013)	Yes	Yes	Yes	No	No	Yes	No
(Di et al., 2013)	Yes	No	No	No	Yes	Yes	No
(HoseinyFarahabady et al., 2014)	Yes	Yes	No	No	Yes	No	No
(Woo and Mirkovic, 2014)	Yes	No	Yes	No	Yes	No	Yes
(Zuo et al., n.d.)	Yes	No	No	No	No	No	No

Table 12
Summary of disaster management schemes in a federated cloud environment.

Scheme	Objective	Strength	Weakness
(Wood and Lagar-Cavilla, 2011)	To recover an application to the point of crash, limited by economic considerations.	Pipelined synchrony addresses the impacts on the duplication latency of WAN by resourcefully overlapping the replication with processing of application for multi-tier servers.	Reconfiguration of the network parameters/ connections and application configurations is not an easy job in case of disaster.
(Liao and Shen, 2011; Liao et al., 2015)	Consider a min-cost fault tolerant resource provisioning in a cloud environment by a employing a facility location formulation.	Formalizing the constrained and unconstraint fault tolerant resource allocation problem along with approximation ratio.	Factors like makespan, system performance, and load balancing are not taken into account.
(Togawa and Kanenishi, 2013)	Built an inter-cloud disaster recovery framework for the cloud-based e-Learning environment.	The distributed storage system built on each private cloud fabric, which handled the one large file system almost like same block device.	The centralized nature of the framework can result in single-point of failure thus stopping the backup process.
(Luo et al., 2014)	proactive cloud defense technologies for secure and agile cloud development	The proposed framework based on a set of systematic security seamlessly integrated mechanisms at the system layer, the application layer, and the network layer in federated cloud environments	Unpredicted variations and scenarios in the presence or environment of failures.
(Khoshkholghi, 2014)	After a disaster (natural/man-made) happens to ensure the continuous delivery of application services.	Considers a decentralized backup approach where, every cloud in the federation system needs a backup cloud to make a system state replication, data backup and also disaster recovery	The recovery time and network traffic redirection without connection losses are not explored.
(Gu et al., 2014)	How to utilize resources of multiple CSPs transparently and cooperatively to guarantee disaster recovery service.	Utilizes multiple optimization scheduling schemes to stable the disaster recovery objectives, in terms of short recovery time, low backup cost, and high data reliability.	How to handle the velocity of data in such environment would be an interesting option to investigate.
(Sengupta and Annervaz, 2014)	Planning back-up of critical data related to business within DCs across numerous geographic locations	The data of client is distributed and replicated in a best way based on consideration of major business criteria such as cost of storage, the level of protection against site failures, and other business and operational parameters like recovery time objective (RTO), and recovery point objective (RPO).	The security mechanism is very crucial in such distribution to a large number of sites.
(Lenk and Tai, 2014)	A warm standby approaches for updating and setting up a standby system in the cloud	Differentiated period updates reduce the network traffic and improve the bandwidth consumption.	Some computation/ data can be lost in such periodic updates. But it is a tradeoff between traffic congestion and data loss.
(Chang and Ramachandran, 2016; Chang et al., 2016)	To secure cloud data from a cyber-security breach	Highly scalable, In penetration testing, CCAF multilayered security could detect and block 99.95% Trojans and viruses, and could attain ≥85% of blocking for 100 h of continuous attack.	The major research limitation is the use of Trojans and viruses for penetration testing, which is the 2013 known vulnerabilities.
(Chang, 2015b)	Disaster recovery (DR) for a Big Data system.	a “multi-purpose” approach, which permits data to be restored with multiple methods to various sites to guarantee that the organization recovers a very high percentage of data close to 100%	The mechanism with multiple sources to replicates is too expensive so an intelligent mechanism should be incorporated to reduce cost and automate certain tasks.

(continued on next page)

different layers for countermeasures; (2) it was centered on proactive cyber defense (3) it used control and command to manage both cross-cloud and in-cloud defense activities via the security centers of the federated cloud.

Khoshkholghi (2014) proposed a DR in federated cloud systems, in which every cloud in the federation system needed a backup cloud to achieve system state replication, data backup, and DR. To this end, each cloud (primary site) selected another cloud as a backup site. However, after a disaster (natural/man-made) happened and the primary site became inaccessible, the failed cloud traffic was redirected to the backup cloud, and the DR cluster delivered the application services. After a failed primary cloud came back online, new data, which were created during the disaster, had to be resynchronized from the secondary site to the original site. The security module encrypted, scrambled, fragmented, and duplicated the data based on the expected level of recovery for security and confidentiality.

Gu et al. (2014) presented DR-Cloud, which was a service model for multi-cloud-based DR. DR-Cloud utilized the resources of multiple CSPs cooperatively through a unified interface provisioned to the customers to adapt according to the heterogeneity of CSPs implicated in the DR service, and the internal processes among clouds were undetectable to the consumers. DR-Cloud proposed multiple optimization scheduling techniques to stable the DR objectives, such as short recovery, low backup cost, and time high data reliability, which were visible to the customers. In addition, on the basis of DR-Cloud different data scheduling schemes could be employed for different data DR

Table 13
Performance Metrics for disaster management in the federated cloud.

Scheme	Real-time	Replication	Multi-Layer	DR Level	Cost effectiveness	Sync/Async	Recovery type
(Wood and Lagar-Cavilla, 2011)	Yes	Yes	Yes	Data level/App level	Yes	Sync	Post disaster
(Liao and Shen, 2011; Liao et al., 2015)	No	Yes	No	Application level	Yes	n/a	Fault tolerance
(Togawa and Kanenishi, 2013)	No	Yes	No	Data level	No	Sync	Prevention
(Luo et al., 2014)	n/a	No	Yes	System level	Yes	n/a	Prevention
(Khoshkholghi, 2014)	No	Yes	No	Application level	Yes	Sync	Post disaster
(Gu et al., 2014)	No	Yes	No	Data/App level	Yes	n/a	Post disaster
(Sengupta and Annervaz, 2014)	No	Yes	No	Data level/App level	No	Async	Post disaster
(Lenk and Tai, 2014)	No	Yes	No	Data level	Yes	Async	Post disaster
(Chang and Ramachandran, 2016; Chang et al., 2016)	Yes	No	Yes	Data level	Yes	n/a	Prevention
(Chang, 2015b)	Yes	Yes	Yes	Data level	No	Sync	Post disaster

scenarios and to achieve their optimization objectives.

Sengupta and Annervaz (2014) presented a data distribution planner for DR (DDP-DR). DDP-DR provided an suitable solution of backing up critical data related to business into DCs across numerous geographic locations. DDP-DR also provided a plan across a potentially large number of DCs for replicating backup data. Therefore, (i) in the event of catastrophic failure the client data were recoverable at one or more DCs (DR), and (ii) the client data were distributed and replicated in an optimal way to consider the important business criteria, such as protection level against site failures, storage cost, and other operational and business parameters, including recovery time objective and recovery point objective. The planner used erasure coding to codify and divide data chunks into fragments and distribute the fragments across storage zones or DR sites so that failure of one or more zone/site could be tolerated and data could be restored.

Lenk and Tai (2014) presented Cloud Standby, which was a warm standby approach in the cloud for updating and setting up a standby system. The proposed method was based on a DR method for updating the standby scheme, analyzing the standby site, and initiating the emergency operation. The standby methods were updated periodically within the data backup store with the data backup available as part of the data backup system. Therefore, in the next run, only the modified data had to be restructured on the image, thereby minimizing the time until the instance was entirely available.

To protect a federated cloud environment and its individual CSP from a man-made cyber attack, (Chang and Ramachandran, 2016; Chang et al., 2016) developed a cloud computing adoption framework (CCAF), which was customized to secure cloud data from a cyber security breach. The authors further explained the components and rationale in the CCAF to protect the security of the cloud data. On the basis of the requirements CCAF was demonstrated by the system design and the implementation illustrated by the CCAF multi-layered security. The authors also presented that CCAF multi-layered security could protect data in real time, and it had three layers of security, namely, 1) convergent encryption, 2) intrusion prevention and identity management, and 3) access control and firewall. The authors conducted two sets of ethical hacking experiments that involved penetration testing with 10,000 viruses and Trojans to validate CCAF and claimed that CCAF quarantined 97.43% of them.

Chang (2015b) proposed a multi-site, multi-technique approach toward DR in unforeseen situations, guaranteeing that if one process did not succeed, then more processes could restore and retrieve data on time. Their proposed methods included a comprehensive approach that could fulfill the needs for value, velocity, volume, veracity, and variety with all data updated and restored. Their solution also offered the traditional transmission control protocol/Internet protocol (TCP/IP) method, snapshot recovery (for VMs), and a hybrid replication system based on the TCP/IP method and snapshot. Their proposed framework utilized tapes and mirror sites to guarantee a high percentage to full recovery. In this way, a CSP can safeguard valuable data of the adopting organizations in the federated cloud environment and protect such data against all forms of hazards.

3.6.1. Remarks on disaster management in the federated cloud computing environment

Considerable literature is available on handling the DR issue in federated cloud environments. The primary site becomes unavailable when a disaster happens, and the secondary site has to be activated. In this case, in a backup site no sync or async replication ability exists, but system and data states can only be locally stored. This phenomenon is a serious threat to the system, yet it is temporary and will be removed after recovery of the primary site. However, all risky situations should be considered to attain the best DR solutions, especially in high-availability services (such as business data storage). Several open research issues are briefly stated below.

- To provide true business continuity for a DR service, it must assist seamless reconfiguration of the network for an application once it is brought online in the backup site.
- Synchronizing the in-memory intermediate states is part of the DR process to save computation rather than data.
- The cost of DR mechanism should be analyzed to identify which DR mechanism is more suitable and does not reduce the net system utility.
- The time required to detect a failure strongly affects the service downtime and initiation of a DR process. However, while replicating across multiple mirror sites, the problem is how to differentiate between network failure and component failure.
- As mentioned before, DR can be human made or it can form by nature. A cyberterrorism attack is a human-made disaster that is accomplished for many reasons. In this case, recovery and protection of essential data will be the major objective in DR plans aside from system restoration.

A summary of a few disaster management schemes is presented in Table 12. Table 13 lists the performance metrics extracted from the federated disaster management schemes.

4. Conclusions

This study reviews the resource management schemes in federated cloud computing environments based on a classification of functional characteristics, such as resource pricing, resource discovery, resource selection, resource monitoring, resource allocation, and disaster management. Each resource management function is briefly defined, and further insights are presented from the state-of-the-art research. The state-of-the-art research is compared based on the performance metrics suitable for individual classified resource management function. Prominent research directions in single statements are also presented for every resource management function.

Many issues need to be addressed in federated cloud resource management with respect to adaptability, flexibility, and standardization. Performance metrics, such as bandwidth overhead, security, and quality of experience, have to be considered while designing a resource management scheme for federated cloud environments. Intelligent computational and cognitive software agents may provide flexible, adaptable, and customized services. Human reasoning can be embedded in agents by using cognitive models and may provide better performance metric values than traditional classical approaches.

Conflict of interest

The author declares that there is no conflict of interests among the authors regarding the publication of this manuscript. Further, the manuscript has not been published, accepted for publication elsewhere or under editorial review for publication elsewhere; and that my Institute's (University of Malaya, Kuala Lumpur, Malaysia) representative and all the co-authors are fully aware of this submission.

Acknowledgement

We would like to thank the University of Malaya in Kuala Lumpur, Malaysia. This work is fully funded by the University of Malay, Malaysian Ministry of Education under the High Impact Research Grant UM.C/HIR/625/1/MOE/FSIT/03 . We also extend our sincere appreciation to grant Newton Ungku Omar (72730770)

References

Ai, L., Tang, M., Fidge, C., 2011. Resource allocation and scheduling of multiple composite web services in cloud computing using cooperative coevolution genetic algorithm. In: Lu, B.-L., Zhang, L., Kwok, J. (Eds.), Neural Information Processing, Lecture Notes in Computer Science.. Springer, Berlin, Heidelberg, 258–267. http://dx.doi.org/10.1007/978-3-642-24958-7_30.

Al-Hazmi, Y., Magedanz, T., 2012. A flexible monitoring system for federated future internet testbeds, In: Proceedings of the 2012 3rd International Conference on the Network of the Future, NOF 2012. IEEE, pp. 14–19. doi: ⟨http://doi.org/10.1109/NOF.2012.6463985⟩.

Apache jclouds® :: Home, n.d. Java Multi-Cloud Toolkit. ⟨https://jclouds.apache.org/⟩ (accessed 31.10.14).

Ardagna, D., Casolari, S., Colajanni, M., Panicucci, B., 2012. Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. J. Parallel Distrib. Comput. 72, 796–808. http://dx.doi.org/10.1016/j.jpdc.2012.02.014.

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., 2010. A view of cloud computing. Commun. ACM 53, 50. http://dx.doi.org/10.1145/1721654.1721672.

Aversa, R., Di Martino, B., Rak, M., Venticinque, S., 2010. Cloud agency: a mobile agent based cloud system. In: Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems. pp. 132–137. doi: ⟨http://doi.org/10.1109/CISIS.2010.143⟩.

Bakshi, K., 2009. Cisco cloud computing – data center strategy, architecture, and solutions point of view white paper. Solutions, 1–16.

Bayer, R., Unterauer, K., 1977. Prefix B-trees.. ACM Trans. Database Syst. 2, 11–26. http://dx.doi.org/10.1145/320521.320530.

Bellifemine, F., Bergenti, F., Caire, G., Poggi, A., 2005. JADE—a java agent development framework. Multi-Agent Program.

Bernsmed, K., Jaatun, M.G., Meland, P.H., Undheim, A., 2011. Security SLAs for Federated Cloud Services. In: Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security. IEEE, pp. 202–209. doi: ⟨http://doi.org/10.1109/ARES.2011.34⟩.

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., Morrow, M., 2009. Blueprint for the intercloud – protocols and formats for cloud computing interoperability. In: Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services, ICIW 2009. IEEE, pp. 328–336. doi: ⟨http://doi.org/10.1109/ICIW.2009.55⟩.

Buyya, R., Yeo, C.S., Venugopal, S., Engineering, S., 1969. Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities. In: Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008. pp. 5–13. doi: ⟨http://doi.org/10.1109/HPCC.2008.172⟩.

CA Technologies, 2013. CA Nimsoft Monitor. ⟨http://www.ca.com/us/products/detail/ca-nimsoft-monitor/details.aspx⟩ (accessed 30.09.14).

Calheiros, R.N., Ranjan, R., Beloglazov, A., Cesar, A.F.D.R., Buyya, R., 2011. loudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource. Softw. – Pract. Exp. 41, 23–50.

Cao, B.-Q., Li, B., Xia, Q.-M., 2009. A service-oriented Qos-assured and multi-agent cloud computing. Architecture, 644–649. http://dx.doi.org/10.1007/978-3-642-10665-1_66.

Carlini, E. et al, 2011. Cloud federations in contrail. In: Proceedings of the European Conference on Parallel Processing. Springer, pp. 159–168.

Caron, E., Rodero-Merino, L., Desprez, F., Muresan, A., 2012. Auto-scaling, load balancing and monitoring in commercial and open-source clouds. Inria Inform. Mat., 24. http://dx.doi.org/10.1201/b11149-17.

Chang, V., 2014. The business intelligence as a service in the cloud. Futur. Gener. Comput. Syst. 37, 512–534. http://dx.doi.org/10.1016/j.future.2013.12.028.

Chang, V., 2015a. Review and discussion: e-learning for academia and industry.. Int. J. Inf. Manag. 36, 476–485. http://dx.doi.org/10.1016/j.ijinfomgt.2015.12.007.

Chang, V., 2015b. Towards a big data system disaster recovery in a private cloud. Ad Hoc Netw. 35, 65–82. http://dx.doi.org/10.1016/j.adhoc.2015.07.012.

Chang, V., Ramachandran, M., 2016. Towards achieving data security with the cloud computing adoption framework. IEEE Trans. Serv. Comput. 9, 138–151. http://dx.doi.org/10.1109/TSC.2015.2491281.

Chang, V., Kuo, Y.H., Ramachandran, M., 2016. Cloud computing adoption framework: A security framework for business clouds. Futur. Gener. Comput. Syst. 57, 24–41. http://dx.doi.org/10.1016/j.future.2015.09.031.

Chung, W.-C., Hsu, C.-J., Lai, K.-C., Li, K.-C., Chung, Y.-C., 2013. Direction-aware resource discovery in large-scale distributed computing environments. J. Supercomput. 66, 229–248. http://dx.doi.org/10.1007/s11227-013-0899-6.

Cloud Kick Cloud, M.I., 2011. Cloud Kick Cloud. ⟨www.cloudkick.com⟩ (accessed 30.09.14).

Damme, E. van, 2000. Non-cooperative games.. Ann. Math., 286–295.

Dastjerdi, A.V., Tabatabaei, S.G.H., Buyya, R., 2012. A dependency-aware ontology-based approach for deploying service level agreement monitoring services in Cloud. Softw. – Pract. Exp. 42, 501–518. http://dx.doi.org/10.1002/spe.1104.

Deltacloud API, n.d. ⟨https://deltacloud.apache.org/⟩ (accessed 31.10.14).

Di, S., Wang, C.-L., Chen, L., 2013. Ex-post efficient resource allocation for Self-organizing Cloud. Comput. Electr. Eng. 39, 2342–2356. http://dx.doi.org/10.1016/j.compeleceng.2012.12.018.

Do, C.T., Tran, N.H., Huh, E.-N., Hong, C.S., Niyato, D., Han, Z., 2016. Dynamics of service selection and provider pricing game in heterogeneous cloud market. J. Netw. Comput. Appl. 69. http://dx.doi.org/10.1016/j.jnca.2016.04.012.

El Zant, B., Gagnaire, M., 2014. New pricing policies for federated cloud. In: Proceedings of the 2014 6th International Conference on New Technologies, Mobility and Security – NTMS 2014 Conference and Workshops. IEEE, pp. 1–6. doi: ⟨http://doi.org10.1109/NTMS.2014.6814036⟩.

Elmroth, E., Larsson, L., 2009. Interfaces for placement, migration, and monitoring of virtual machines in federated clouds. In: Proceedings of the 8th International Conference Grid Coop. Comput. GCC 2009. pp. 253–260. doi: ⟨http://doi.org/10.1109/GCC.2009.36⟩.

Esxi, V., 2016. VmWare Esxi. ⟨https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html⟩ (accessed 16.04.14).

Fan, T., Liu, J., Gao, F., 2013. Dynamic Pricing Trade Market for Shared Resources in IIU Federated Cloud. vol. 22, pp. 170–175.

Farokhi, S., Jrad, F., Brandic, I., Streit, A., 2014. HS4MC – hierarchical sla-based service selection for multi-cloud environments. In: Proceedings of the 4th International Conference Cloud Computing and Service Science. pp. 722–734. doi: ⟨http://doi.org/10.5220/0004979707220734⟩.

Ferrer, A.J., Hern??ndez, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S.K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forg??, N., Sharif, T., Sheridan, C., 2012. OPTIMIS: a holistic approach to cloud service provisioning. Futur. Gener. Comput. Syst. 28, 66–77. http://dx.doi.org/10.1016/j.future.2011.05.022.

Foster, I., Zhao, Y., Raicu, I., Lu, S., 2008. Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, GCE 2008. IEEE, pp. 1–10. doi: ⟨http://doi.org/10.1109/GCE.2008.4738445⟩.

Gomes, E.R., Vo, Q.B., Kowalczyk, R., 2012. Pure exchange markets for resource sharing in federated clouds. Concurr. Comput. Pract. Exp. 24, 977–991. http://dx.doi.org/10.1002/cpe.1659.

Gross, D., Shortle, J., Thompson, F., Harris, C., 2008. Fundamentals of Queueing Theory. John Wiley & Sons, Reading. http://dx.doi.org/10.1017/CBO9781107415324.004.

Grozev, N., Buyya, R., 2014. Inter-Cloud architectures and application brokering: taxonomy and survey. Softw. – Pract. Exp. 44, 369–390. http://dx.doi.org/10.1002/spe.2168.

Gu, Y., Wang, D., Liu, C., 2014. DR-Cloud: multi-cloud based disaster recovery service. Tsinghua Sci. Technol. 19, 13–23. http://dx.doi.org/10.1109/TST.2014.6733204.

Gutierrez-Garcia, J.O., Sim, K.M., 2013. Agent-based cloud service composition. Appl. Intell. 38, 436–464. http://dx.doi.org/10.1007/s10489-012-0380-x.

van der Ham, J., Stéger, J., Laki, S., Kryftis, Y., Maglaris, V., de Laat, C., 2015. The NOVI information models. Futur. Gener. Comput. Syst. 42, 64–73. http://dx.doi.org/10.1016/j.future.2013.12.017.

Hassan, M.M., Hossain, M.S., Sarkar, A.M.J., Huh, E.N., 2014. Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform. Inf. Syst. Front. 16, 523–542. http://dx.doi.org/10.1007/s10796-012-9357-x.

Hassan, M.M., Song, B., Huh, E.-N., 2011. Distributed resource allocation games in horizontal dynamic cloud federation platform. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (Eds.), 2011 IEEE International Conference on High Performance Computing and Communications, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 822–827. doi: ⟨http://doi.org/10.1109/HPCC.2011.116⟩.

HoseinyFarahabady, M.R., Lee, Y.C., Zomaya, A.Y., 2014. Randomized approximation scheme for resource allocation in hybrid-cloud environment. J. Supercomput. 69, 576–592. http://dx.doi.org/10.1007/s11227-014-1094-0.

Hull, J.C., 1997. Options, Futures and Other Derivates. Prentice Hall, Pearson. http://

dx.doi.org/10.1017/CBO9781107415324.004.

Jaikar, A., Noh, S.-Y., 2014. Cost and performance effective data center selection system for scientific federated cloud. Peer-to-Peer Netw. Appl, 1–7. http://dx.doi.org/10.1007/s12083-014-0261-7.

Jrad, F., Tao, J., Streit, A., 2012. SLA based service brokering in intercloud environments. In: CLOSER 2012 – Proceedings of the 2nd International Conference Cloud Computing and Services Science, pp. 76–81.

Kalim, U., Gardner, M.K., Brown, E.J., Feng, W., 2013. Seamless Migration of Virtual Machines across Networks, in: 2013 22nd International Conference on Computer Communication and Networks (ICCCN). IEEE, pp. 1–7, doi:10.1109/ICCCN.2013.6614190

Kertesz, A., Kecskemeti, G., Oriol, M., Kotcauer, P., Acs, S., Rodríguez, M., Mercè, O., Marosi, A.C., Marco, J., Franch, X., 2013. Enhancing federated cloud management with an integrated service monitoring approach. J. Grid Comput 11, 699–720. http://dx.doi.org/10.1007/s10723-013-9269-0.

Kertesz, A., Kecskemeti, G., Marosi, A., Oriol, M., Franch, X., Marco, J., 2012. Integrated monitoring approach for seamless service provisioning in federated clouds. In: - Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2012. IEEE, pp. 567–574. doi: ⟨http://doi.org/10.1109/PDP.2012.25⟩.

Khethavath, P., Thomas, J., Chan-Tin, E., Liu, H., 2013. Introducing a distributed cloud architecture with efficient resource discovery and optimal resource allocation. In: Proceedings of the 2013 IEEE 9th World Congress on Services, SERVICES 2013. IEEE, pp. 386–392. doi: ⟨http://doi.org/10.1109/SERVICES.2013.68⟩.

Khoshkholghi, M.A., 2014. Cluster as a service for disaster recovery in intercloud systems: design and modeling. Int. J. Cloud Comput. Serv. Sci.. http://dx.doi.org/10.11591/closer.v3i3.6375.

Klyne, G., Carroll, J.J., 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recomm.

Kolda, T., Lewis, R., Torczon, V., 2003. Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. 45, 385–482. http://dx.doi.org/10.1137/S003614450242889.

Le, G., Xu, K., Song, J., 2012. Gossip-based hybrid multi-attribute overlay for resource discovery in federated clouds. In: - Proceedings of the 9th IEEE International Conference on E-Business Engineering, ICEBE 2012. IEEE, pp. 279–284. doi: ⟨http://doi.org/10.1109/ICEBE.2012.52⟩.

Lenk, A., Tai, S., 2014. Cloud standby: Disaster recovery of distributed systems in the cloud. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 8745 LNCS, 32–46. http://dx.doi.org/10.1007/978-3-662-44879-3_3.

Liao, K., Shen, H., 2011. Unconstrained and constrained fault-tolerant resource allocation. In: Fu, B., Du, D.-Z. (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Lecture Notes in Computer Science. Springer Berlin Heidelberg, 555–566. http://dx.doi.org/10.1007/978-3-642-22685-4_48.

Liao, K., Shen, H., Guo, L., 2015. Improved approximation algorithms for constrained fault-tolerant resource allocation. In: Gąsieniec, L., Wolter, F. (Eds.), Theoretical Computer Science, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 118–128. http://dx.doi.org/10.1016/j.tcs.2015.02.029.

Liaqat, M., Ninoriya, S., Shuja, J., Ahmad, R.W., Gani, A., 2016. Virtual machine migration enabled cloud resource management: a challenging task. Distrib. Parallel Clust. Comput. Cornell Univ., 1–7, (arXiv Prepr. arXiv 1601.03854).

Lopez-Rodriguez, I., Hernandez-Tejera, M., 2011. Software Agents as Cloud Computing Services. pp. 271–276. doi: ⟨http://doi.org/10.1007/978-3-642-19875-5_35⟩.

Luo, W., Xu, L., Zhan, Z., Zheng, Q., Xu, S., 2014. Federated cloud security architecture for secure and agile clouds. High perform. Cloud Audit. Appl 9781461432, 169–188. http://dx.doi.org/10.1007/978-1-4614-3296-8_7.

Malet, B., Pietzuch, P., 2010. Resource allocation across multiple cloud data centres. In: Proceedings of the 8th Int. Work. Middlew. Grids, Clouds e-Science – MGC '10. pp. 1–6. doi: ⟨http://doi.org/10.1145/1890799.1890804⟩.

Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., Villari, M., 2011. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: Proceedings of the IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum. IEEE, pp. 1510–1517. doi: ⟨http://doi.org/10.1109/IPDPS.2011.304⟩.

Mastroianni, C., Papuzzo, G., 2014. A self-organizing P2P framework for collective service discovery. J. Netw. Comput. Appl. 39, 214–222. http://dx.doi.org/10.1016/j.jnca.2013.07.002.

Maymounkov, P., Mazieres, D., 2002. Kademlia: a peer-to-peer information system based on the xor metric. First Int. Work. Peer-to-Peer Syst., 53–65. http://dx.doi.org/10.1007/3-540-45748-8.

McGuinnes, D.L., Van Harmelen, F., 2004. OWL Web Ontology Language.

Meshkova, E., Riihijärvi, J., Petrova, M., Mähänen, P., 2008. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. Comput. Netw. 52, 2097–2128. http://dx.doi.org/10.1016/j.comnet.2008.03.006.

Mihailescu, M., Teo, Y.M., 2010a. Dynamic resource pricing on federated clouds. In: Proceedings of the CCGrid 2010 – 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. pp. 513–517. doi: ⟨http://doi.org/10.1109/CCGRID.2010.123⟩.

Mihailescu, M., Teo, Y.M., 2010b. Strategy-proof dynamic resource pricing of multiple resource types on federated clouds. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 337–350. doi: ⟨http://doi.org/10.1007/978-3-642-13119-6_30⟩.

Mohamed, M., Belaïd, D., Tata, S., Amziani, M., n.d. OCCI4Java and Autonomic Computing. ⟨http://www-inf.it-sudparis.eu/SIMBAD/tools/OCCI/autonomic/⟩ (accessed 29.10.14).

Nielson, S.J., Crosby, S.A., Wallach, D.S., 2005. A taxonomy of rational attacks, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 36–46. doi: ⟨http://doi.org/10.1007/11558989_4⟩.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D., 2009. The Eucalyptus Open-Source Cloud-Computing System. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009. IEEE, pp. 124–131. doi: ⟨http://doi.org/10.1109/CCGRID.2009.93⟩.

Pal, R., Hui, P., 2013. Economic models for cloud service markets: pricing and Capacity planning. Theor. Comput. Sci. 496, 113–124. http://dx.doi.org/10.1016/j.tcs.2012.11.001.

Palmieri, F., Buonanno, L., Venticinque, S., Aversa, R., Di Martino, B., 2013. A distributed scheduling framework based on selfish autonomous agents for federated cloud environments. Futur. Gener. Comput. Syst. 29, 1461–1472. http://dx.doi.org/10.1016/j.future.2013.01.012.

Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervell??-Pastor, C., Monje, ??lvaro, 2013. On the optimal allocation of virtual resources in cloud computing networks. IEEE Trans. Comput. 62, 1060–1071. http://dx.doi.org/10.1109/TC.2013.31.

Papagianni, C., n.d. NOVI : Networking innovations Over Virtualized Infrastructures Policy Information Model Information Model of Policy.

Pei, W.-J.F.-L.Y.P., 2015. A multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach. Int. J. Autom. Comput., 12.

Petcu, D., 2014. Consuming resources and services from multiple clouds: from terminology to cloudware support. J. Grid Comput 12, 321–345. http://dx.doi.org/10.1007/s10723-013-9290-3.

Pham, C., Tran, N.H., Nguyen, M.N.H., Ren, S., Saad, W., Hong, C.S., 2016. Hosting virtual machines on a cloud datacenter: a matching theoretic approach. In: Proceedings of the NOMS 2016 – 2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 659–664. doi: ⟨http://doi.org/10.1109/NOMS.2016.7502873⟩.

Pittaras, C., Papagianni, C., Leivadeas, A., Grosso, P., Van Der Ham, J., Papavassiliou, S., 2014. Resource discovery and allocation for federated virtualized infrastructures. Futur. Gener. Comput. Syst. 42, 55–63. http://dx.doi.org/10.1016/j.future.2014.01.003.

Plestys, R., Vilutis, G., Sandonavicius, D., Vaskeviciute, R., Kavaliunas, R., 2007. The measurement of Grid QoS parameters. In: Proceedings International Conference on Information Technology and Interfaces, ITI. pp. 703–707. doi: ⟨http://doi.org/10.1109/ITI.2007.4283857⟩.

Povedano-Molina, J., Lopez-Vega, J.M., Lopez-Soler, J.M., Corradi, A., Foschini, L., 2013. DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds. Futur. Gener. Comput. Syst. 29, 2041–2056. http://dx.doi.org/10.1016/j.future.2013.04.022.

Qanbari, S., Li, F., Dustdar, S., Dai, T., 2014. Cloud asset pricing tree (CAPT): elastic economic model for cloud service providers. Chinacloud, 221–229. http://dx.doi.org/10.5220/0004849702210229, (Cn).

Querna, P., 2010. Apache Libcloud.

Ristenpart, T., Tromer, E., 2009.Hey, you, get off of my cloud: exploring information leakage in Proceedings of the third-party compute clouds, in: … Conference on Computer …. ACM Press, New York, New York, USA, pp. 199–212. doi:10.1145/1653662.1653687

Roh, H., Jung, C., Lee, W., Du, D.Z., 2013. Resource pricing game in geo-distributed clouds. In: Proceedings of the – IEEE INFOCOM. pp. 1519–1527. doi: ⟨http://doi.org/10.1109/INFCOM.2013.6566947⟩.

Rosen, J.B., 1965. Existence and uniqueness of equilibrium points for concave N/nobreakdash-person games. Econometrica 33, (520/nobreakdash–534).

Rowstron, A., Druschel, P., 2001. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Middlew 2001 (2218), 329–350. http://dx.doi.org/10.1007/3-540-45518-3.

Rubinstein, A., 1982. Perfect equilibrium in a bargaining model. Econometrica 50, 97–109. http://dx.doi.org/10.2307/1912531.

Sengupta, S., Annervaz, K.M., 2014. Multi-site data distribution for disaster recovery – a planning framework. Futur. Gener. Comput. Syst. 41, 53–64. http://dx.doi.org/10.1016/j.future.2014.07.007.

Sim, K.M., 2006. Grid commerce, market-driven G-negotiation, and grid resource management. IEEE Trans. Syst. Man Cybern. Part B Cybern. 36, 1381–1394. http://dx.doi.org/10.1109/TSMCB.2006.877791.

Sim, K.M., 2008. Evolving fuzzy rules for relaxed-criteria negotiation. IEEE Trans. Syst. Man Cybern. Part B Cybern. 38, 1486–1500. http://dx.doi.org/10.1109/TSMCB.2008.928210.

Sim, K.M., 2012. Agent-based cloud computing. IEEE Trans. Serv. Comput. 5, 564–577. http://dx.doi.org/10.1109/TSC.2011.52.

Smit, M., Simmons, B., Litoiu, M., 2013. Distributed, application-level monitoring for heterogeneous clouds using stream processing. Futur. Gener. Comput. Syst. 29, 2103–2114. http://dx.doi.org/10.1016/j.future.2013.01.009.

Smith, J., Nair, R., 2005. Virtual machines [electronic resource] : versatile platforms for systems and processes. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier.

Son, J., 2013. Automated Decision System for Efficient Resource Selection and Allocation in Inter-Clouds By.

Sotiriadis, S., Bessis, N., Kuonen, P., 2012. Advancing inter-cloud resource discovery based on past service experiences of transient resource clustering. In: - Proceedings of the 3rd International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2012. IEEE, pp. 38–45. doi: ⟨http://doi.org/10.1109/EIDWT.2012.16⟩.

Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I., 2009. Virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput 13, 14–22. http://dx.doi.org/10.1109/MIC.2009.119.

Stackelberg, 1934. Market Structure and Equilibrium (Marktform und Gleichgewicht). doi: ⟨http://doi.org/10.1007/978-3-642-12586-7⟩.

Sundareswaran, S., Squicciarini, A., Lin, D., 2012. A brokerage-based approach for cloud service selection. In: Proceedings of the – 2012 IEEE 5th International Conference on Cloud Computing. CLOUD 2012. pp. 558–565. doi: ⟨http://doi.org/10.1109/CLOUD.2012.119⟩.

Talia, D., 2012. Clouds meet agents: toward intelligent cloud services. IEEE Internet Comput. 16, 78–81. http://dx.doi.org/10.1109/MIC.2012.28.

Tierney, B., Aydt, R., Gunter, D., Smith, W., Swany, M., 2002. A Grid Monitoring Architecture.

Togawa, S., Kanenishi, K., 2013. Private cloud cooperation framework of e-learning environment for disaster recovery. In: Proceedings of the – 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013. IEEE, pp. 4104–4109. doi: ⟨http://doi.org/10.1109/SMC.2013.700⟩.

Tolba, A., Ghoneim, A., 2015. A stepwise self-adaptive model for improving cloud efficiency based on multi-agent features. J. Softw. 10, 1037–1044. http://dx.doi.org/10.17706/jsw.10.8.1037-1044.

Toosi, A.N., Calheiros, R.N., Buyya, R., 2014. Interconnected cloud computing environments. ACM Comput. Surv 47, 1–47. http://dx.doi.org/10.1145/2593512.

Toosi, A.N., Thulasiram, R.K., Buyya, R., 2012. Financial option market model for federated cloud environments. In: Proceedings of the – 2012 IEEE/ACM 5th International Conference on Utility and Cloud Computing UCC 2012, pp. 3–12. doi: ⟨http://doi.org/10.1109/UCC.2012.42⟩.

Tran, N.H., Do, C.T., Ren, S., Han, Z., Hong, C.S., 2015. Incentive mechanisms for economic and emergency demand responses of colocation datacenters.. IEEE J. Sel. Areas Commun. 33, 2892–2905. http://dx.doi.org/10.1109/JSAC.2015.2483420.

Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M., 2008. A break in the clouds. ACM SIGCOMM Comput. Commun. Rev. 39, 50. http://dx.doi.org/10.1145/1496091.1496100.

Vilutis, G., Butkiene, R., Lagzdinyte-Budnike, I., Sandonavicius, D., Paulikas, K., 2013. The QoGS method application for selection of computing resources in intercloud. Electron. Electr. Eng.. http://dx.doi.org/10.5755/j01.eee.19.7.2080.

Wickremasinghe, B., Calheiros, R.N., Buyya, R., 2010. CloudAnalyst: a cloudsim-based visual modeller for analysing cloud computing environments and applications. In: Proceedings of the – International Conference Advanced Information Networking and Applications, AINA, pp. 446–452. doi: ⟨http://doi.org/10.1109/AINA.2010.32⟩.

Wills, G., Chang, V., Walters, R.J., 2012. Business integration as a service. Int. J. Cloud Appl. Comput.. http://dx.doi.org/10.4018/ijcac.2012010102.

Woo, S.S., Mirkovic, J., 2014. Optimal application allocation on multiple public clouds Comput.. Networks 68, 138–148. http://dx.doi.org/10.1016/j.comnet.2013.12.001.

Wood, T., Lagar-Cavilla, H.A., 2011. PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery. Symp. Cloud.

Wright, P., Sun, Y., Harmer, T., Keenan, A., Stewart, A., Perrott, R., 2012. A constraints-based resource discovery model for multi-provider cloud environments. J. Cloud Comput. Adv. Syst. Appl. 1, 6. http://dx.doi.org/10.1186/2192-113X-1-6.

Xen, 2014. The Xen Project. ⟨http://www.xenproject.org⟩ (accessed 16.04.14).

Yau, S.S., Yin, Y., 2011. QoS-based service ranking and selection for service-based systems. In: Proceedings of the – 2011 IEEE International Conference on Services Computing, SCC 2011. IEEE, pp. 56–63. doi: ⟨http://doi.org/10.1109/SCC.2011.114⟩.

Zeginis, C., Kritikos, K., Garefalakis, P., 2013. Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications. pp. 188–195.

Zuo, X., Zhang, G., Tan, W., n.d. Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud. ⟨Ieeexplore.Ieee.Org⟩.