Ph.D. Dissertation in Engineering

Economic Models for Incentivizing the Federations of IaaS Cloud Providers

August 2019

Graduate School of Seoul National University

Technology Management, Economics and Policy Program

Ram Govinda Aryal

Economic Models for Incentivizing the Federations of IaaS Cloud Providers

지도 교수 Jörn Altmann

이 논문을 공학박사학위 논문으로 제출함

2019년 8월

서울대학교 대학원

협동과정 기술경영경제정책 전공

Ram Govinda Aryal

램고빈다의 공학박사학위 논문을 인준함

2019년 8월

위 원 장 **Junseok Hwang** (인)

부위원장 **Jörn Altmann** (인)

위 원 **Bernhard Egger** (인)

위 원 **Konstantinos Tserpes** (인)

위 원 **Baseem Taher Othman Al-athwari** (인)

Abstract

Economic Models for Incentivizing the Federations of IaaS Cloud providers

Ram Govinda Aryal

Technology Management Economics and Policy Program

College of Engineering

Seoul National University

Cloud industry is susceptible to the economies of scale. Therefore, smaller providers seem to be struggling for their reasonable market shares. A recent report by Gartner shows that only five hyper-scale providers have occupied 75% of the cloud market in the Infrastructure as a Service (IaaS) segment. In this context where small cloud providers are discriminated by the economies of scale, cloud federation has been considered to have the potential of improving their competitiveness by enabling them to collaborate and gain access to increased resources, provide better service quality, offer service variety,

minimize costs, and hence benefit from the economies of scale. Cloud providers are willing to collaborate in federation only if there is a clear model defining the commercial relationships, more specifically, the rules and methods for how the payoff is collectively generated and shared. Lack of such rules and methods is one of the reasons why we do not see cloud federations operating in the commercial market.

A large body of previous research on cloud federations focuses on issues of technical nature, such as interoperability, resource discovery, resource selection, pricing, accounting & billing, Service Level Agreements, security, and monitoring. But, issues of economic nature such as the payoff generation through optimal resource sharing and its distribution with fair and lucrative allocation methods have not received adequate attention.

In this thesis, we investigate economic models for the operation of cloud federation with an aim to improve their competitiveness through the economies of scale by encouraging them to collaborate in the federation with a fair and attractive incentive mechanism. Our first aim is to provide algorithms that facilitate the composite selection of federated resources for the deployment of customer applications with optimization on cost and various QoS criteria as per individual customer stated preferences. We do so by combining the Analytic Hierarchy Process, a multi-criteria decision-making method, and evolutionary multi-objective optimization algorithm, namely A Fast

and Elitist Non-dominated Sorting Genetic Algorithm (NSGA II). Simulation programs are developed by implementing the proposed algorithm and simulations are conducted to evaluate the proposed algorithm.

The simulation results demonstrate that the proposed algorithm enables service placement at various tradeoffs points optimized on cost and various QoS parameters as per the consumer preferences allowing for cost reduction by up to 4%, processing speed increment by up to 47.8%, latency reduction by up to 36.6%, and overall availability increment by up to 5.5%. Simulation result also shows that the proposed approach outperforms benchmark approach when compared in terms of standard metrics such as Generational Distance, Spacing, and Set Coverage, which are used to compare the performance of multi-objective optimization algorithms.

Our second aim is to propose a revenue-sharing scheme that ensures fair distribution of collectively generated revenue among the federation members. We employed Shapley Value method, a solution concept in coalitional game theory to design our revenue sharing scheme, where the revenue share is allocated in proportion to the contribution made by each federation member in the value creation of the federation. Their contribution to value creation is estimated based on their infrastructure capacity and market share. The infrastructure capacity is assessed based on the resources utilized in actual service provisioning and the market

share is assessed on the basis of the service request brought in to the federation.

By developing a simulation program and performing simulations we try to answer various questions pertaining to cloud providers' decision regarding joining a federation. Simulation results demonstrate the benefits of the federation in the form of an increase in both resource utilization and return on investment by over 30%. The results demonstrate that the benefits of joining the federation depend on the capacity as well as the demand to capacity ratio. For a federation of providers with smaller capacities, the benefits of increased return on investment that could be achieved by operating in a federated model starts at a lower level of the demand-capacity ratio while that for the federation of providers with larger capacity starts at a higher level of the demand-capacity ratio. The simulation results also indicate that the proposed revenue sharing scheme provides better incentive system compared to the benchmark participatory approach as it allows for competition within the collaboration by incentivizing the member providers' efforts towards the excellence in cost reduction and service quality.

Overall, this research contributes to the industry by solving a composite service selection problem. It enables federations and cloud brokers to serve a variety of customers who seek service at different levels of price and QoS. It enables them to offer truly optimized deployment

service at the tradeoff point specified by individual customers. It provides a scheme for the operation of cloud federation along with a fair method for revenue sharing and at the same time providing benefits to providers of different characteristics. It also provides a guide to cloud providers for when it is not beneficial for them to join the federation depending on their relative position with respect to other members. The research also provides implications to the research communities working with multi-objective optimization, multi-criteria decision making, and revenue sharing within any domain.

Keywords: Cloud Economics, Economics-based Resource Allocation, Multi-objective Optimization, Consumer Preference, AHP, Evolutionary Algorithm, NSGA II, Revenue Sharing, Shapley Value

Student Number: 2016-34687

List of Abbreviations

Abbreviation	Meaning
AHP	Analytic Hierarchy Process
AWS	Amazon Web Service
CSP	Cloud Service Provider
FLA	Federation Level Agreement
GA	Genetic Algorithm
GD	Generational Distance
IaaS	Infrastructure as a Service
IEEE	Institute of Electrical and Electronics Engineers
MCDM	Multi Criteria Decision Making
MOO	Multi Objective Optimization
NSGA	Non-dominated Sorting Genetic Algorithm
PaaS	Platform as a Service
PoI	Point of Interest
QoS	Quality of Service
RTT	Round Trip Time

SaaS	Software as a Service
SLA	Service Level Agreement
SOO	Single Objective Optimization
TOSCA	Topology and Orchestration Specification for Cloud Applications
VM	Virtual Machine

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Background and motivation

Cloud computing has brought about a paradigm shift in the way how IT capabilities are managed, delivered, and consumed (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009). With an ongoing shift of computing from the traditional data center to private and public cloud (International Data Corporation, 2018), the IT industry has observed an unprecedented growth in the demand for cloud services both in the public and private sectors. According to Gartner, Worldwide Public Cloud revenue in the year 2019 is projected to grow by 17.3% from $175.8B in 2018 (Gartner, 2018a).

The increasing demand is due to the flexibility that cloud computing offers. The flexibility comes in the form of on-demand access to resources from anywhere, and a pay-as-you-go payment model (Vaquero, Rodero-Merino, Caceres, & Lindner, 2008). This allows cloud customers to deploy their applications rapidly without requiring expert technical skills and upfront costs.

Cloud service is basically delivered as one of the three models - Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) (Vaquero et al., 2008). Among these service segments, IaaS is the fastest growing segment, which is forecasted to generate $39.5B revenue in 2019, an increase of 27.6%

from the previous year (Gartner, 2018a). IaaS allows consumers of system infrastructure resources to outsource it to third-party providers. These providers provide on-demand access to a large pool of computing, network, storage and other fundamental resources which can be used by customers to deploy and execute any software including operating systems and other business applications (Mell & Grance, 2011). Typically, the IaaS resources are delivered in the form of Virtual Machine (VM) Instances with different configurations and Quality of Service (QoS) guarantees. Customers can deploy their applications making use of these VM instances.

IaaS segment, the most rapidly growing one, shows a special phenomenon in market distribution. According to Gartner (Gartner, 2018b), over three fourth of the market share is occupied by only five of the largest providers, namely, Amazon, Microsoft, Alibaba, Google, and IBM. This leaves only a fraction of the market to thousands of others. The reason for the market share structure following a power law distribution phenomenon is attributed to the discrimination provided by the economies of scale to smaller cloud providers (Kim, Kang, & Altmann, 2014). This suggests that the cloud providers in this segment, especially the smaller ones operating with limited capacity, have critical economic challenges to address to become competitive in the market (Harms & Yamartino, 2010).

This thesis is centered on addressing the economic challenges of IaaS cloud providers with the aim of improving the profitability and competitiveness of smaller ones. The competitiveness of smaller ones is affected by the discrimination provided by the economies scale which is reflected with resource limitation, inefficiencies in capacity utilization, inadequate service quality, limited service variety, and limited geographic presence.

Resource limitation is one of the most important economic challenges of IaaS cloud providers. The perishable nature of cloud services make their storage for future use impossible and hence requires them to be consumed to the extent possible in order to maximize profits (Xu & Li, 2013). This phenomenon forces data centers to operate within a resource limit, which results in their limited resource scaling capacity (Goher, Bloodsworth, Ur Rasool, & McClatchey, 2017), and making them unable to fulfill the request of large applications at the time when it is absolutely needed. This significantly hampers the provider's competitiveness in the market.

Another economic challenge is associated with inefficiencies in capacity utilization. To some extent, the effectiveness of the multi-tenancy model for efficient resource utilization is well demonstrated and proven (James Cuff, Ignacio M. Llorente, Christopher Hill, 2017); however, there still are potentials for improvement as the average utilization of cloud resources at present is reported to be only at 50%

(Householder, Arnold, & Green, 2014). Mostly, the problem is due to the inherent characteristics of the type of business that IaaS providers operate in. The nature of the customer requests being stochastic and time-varying makes the capacity planning a tough task for IaaS cloud providers making them susceptible to over-provisioning or under-provisioning problems (Goiri, Guitart, & Torres, 2012). Finding the optimum capacity level when the demand is of stochastic and the capacity is of perishable nature is a non-trivial problem (Xu & Li, 2013). The liability to fulfill the customer SLA adds to this capacity planning problem.

Potentially, the problem of inefficiency in capacity utilization is more pronounced among small cloud providers compared to the larger ones. The disparity is evident from the fact that the average utilization ratio of small providers is in the range 10 to 50 percent while that for hyper-scale providers remains in the range 40 to 70 percent (Whitney & Delforge, 2014). This is because larger ones can smooth out the spikes in the demand by averaging across a large number of user requests, commonly known as demand-side aggregation (Harms & Yamartino, 2010). In addition, they adopt different pricing policies such as subscription-based, usage-based and dynamic pricing, which enables them to absorb the demand spikes resulting in efficient utilization ratio (Toosi, 2014). The need for complex optimization procedure makes it impractical for small providers to pursue such pricing policies.

Another challenge is associated with the Quality of service. The hyper-scale cloud providers such as Amazon have data centers distributed across the globe clustered into regions and availability zones (AWS, 2018). With such facilities, they are able to reduce the network latency for a group of application users in the particular geographic region by moving application services to their data centers in the region that is in close proximity of the user group (Hornsby, 2018). This significantly increases the responsiveness of the customer application with latency minimization.

Reduced application latency is found to increase user engagement and increase sales & profitability of the application owner. It is reported that 100ms of additional page loading time reduced Amazon sales by 1% and 500ms of additional page loading time caused a reduction in Google search by 20% (Arapakis, Bai, & Cambazoglu, 2014). This way, reduced latency contributes to the competitiveness of the cloud provider. Also, hyper-scale providers can offer reliable multi-site deployment of customer applications. In the event of the failure of service at one deployment, they can still maintain the system availability by directing the user requests to another deployment where the services are still up and running (Hornsby, 2018).

The high degree of efficiency in resource utilization, deployment of an application with a high level of responsiveness and availability is feasible in the case of hyper-scale providers. Smaller providers are

unable to compete at this level. In this context, smaller providers willing to compete in the market should seek to reduce operational cost, offer better service quality and variety, and increase profitability within their limited resource capacity by adopting effective strategies and methods geared towards that direction.

Various researchers have identified Cloud federation as a way to address such existing limitations (Ferrer et al., 2012; Haile & Altmann, 2015; Petcu, 2014; Rochwerger et al., 2009). A Cloud federation can be considered as a strategic alliance among cloud providers where they have voluntary arrangements to interconnect their infrastructure to enable resource sharing with provisions for deploying application components on each other's infrastructures (Haile & Altmann, 2015). It enables small cloud providers to gain access to an increased number of cloud infrastructure resources by collaborating with others (Darzanos, Koutsopoulos, & Stamoulis, 2016; Haile & Altmann, 2018), gain economies of scale with resource aggregation (Kim et al., 2014). It also helps them ensure the users' quality of experience, for example with reduced latency, and minimize costs (Hassan, Hossain, Sarkar, & Huh, 2014). By joining a federation, a cloud provider can also provide guaranteed availability of customer applications through reliable multi-site deployments (Petcu, 2014).

1.2 Problem Description

From various researches, it has been well acknowledged that Cloud federation can be as a feasible approach to address the economic challenges of IaaS cloud providers, especially the smaller ones. By joining a federation, a cloud provider can gain access to extended resources, minimize SLA violations, provide more assured system availability, increase resource utilization efficiency, competitiveness and hence the profitability.

Realizing the potentials, substantial attraction is observed in Cloud federation research in recent years. Ample research has been carried out focusing on the cloud federation challenges, such as interoperability (de Carvalho, Trinta, & Vieira, 2018; Haile & Altmann, 2018), resource provisioning (Goher et al., 2017; Zhang, Huang, & Wang, 2016), pricing accounting & billing (Li, Wu, Li, & Lau, 2016; L. Lu, Yu, Zhu, & Li, 2018), Service Level Agreements (SLA) (Chudasama, Tilala, & Bhavsar, 2017; Dhirani, Newe, & Nizamani, 2019), security (Demchenko, Turkmen, de Laat, & Slawik, 2017; Ferdous, Margheri, Paci, Yang, & Sassone, 2017), and monitoring (Edu-yaw & Kuada, 2018; Syed, Gani, Ahmad, Khan, & Ahmed, 2017).

Despite significant promises and ample research in the field, we cannot find any cloud federation in operation and functional in the commercial market, keeping aside those that are targeted for non-commercial purposes, such as EGI Federated Cloud (Fernández-del-Castillo,

Scardaci, & García, 2015), and some cloud service brokers who provide service to their customers by making use of resources from multiple cloud providers.

Some research also has investigated the factors hindering the adoption of cloud federation (Breskovic, Maurer, Emeakaroha, Brandic, & Altmann, 2011; Haile & Altmann, 2015). A body of literature has considered revenue sharing as an important issue that incentivizes cloud providers to form and operate as coalitions and federations (Aryal & Altmann, 2017; Breskovic et al., 2011; Coronado & Altmann, 2017; Haile & Altmann, 2015; Hassan et al., 2014; Jeferry et al., 2015; Samaan, 2014). The importance and lack of models defining clearly the commercial relationships between members of the federation have also been acknowledged in a recent panel discussion comprising of speakers from cloud computing industry that include members of the IEEE Cloud Standards Committee as well (ieeeCESocTV, 2018). This shows that clear revenue sharing models are essential for us to see more cloud federations operating in the open cloud market.

Further, it is important to state that the studies relating to the issue of revenue sharing methods should also be linked to the resource sharing methods because the only way of generating revenue in a cloud federation is by making use of the shared resources to provision cloud services to customer requests. Thus, appropriate resource and revenue sharing mechanism that specify how cloud resources owned by member

cloud providers are used for provisioning services to customer requests and how the revenue generated by the collaborative efforts in service provisioning with shared resources is allocated to federation members, constitute the foundations that support the formation and the sustained operation of cloud federations.

Cloud federation research has not received adequate attention regarding the economic challenges related to these issues. Limited research has focused on the economic aspects of cloud federation. Most of these research deal cloud federation as non-cooperative coalition (Guazzone, Anglano, & Sereno, 2014; Li, Wu, Li, & Lau, 2013; Samaan, 2014), where federation members focus on individual strategies and payoffs (Hespanha, 2011), and these strategies guide how sharing of resources and revenue takes place. Very few researchers have studied the problem of cloud federation from the viewpoint of co-operative coalition focusing on socially optimal federations (Hassan, Abdullah-Al-Wadud, Almogren, Song, & Alamri, 2017; Mashayekhy, Nejad, & Grosu, 2015), which is what is required for the federation to be able to compete with hyper-scale providers by tapping the benefits of the economies of scale in the way hyper-scale providers do. This is because a cooperative setting allows not only for supply-side aggregation but also allows for demand-side aggregation (Harms & Yamartino, 2010), which enables the federation members to provide the required level of the QoS, yet, maintaining the infrastructure capacity at a lower level (Harms & Yamartino, 2010; Kim et al., 2014). This would increase the

capacity utilization ratio and thereby increase the overall profits. Complete aggregation of resource and requests are impossible with non-cooperative coalitions, as there are no binding rules to enforce such behavior in a non-cooperative coalition (Hespanha, 2011).

Very few researchers have attempted to study cloud federation by considering it as a co-operative coalition, and have tried analyzing through Cooperative Game Theory (Hassan et al., 2017; Mashayekhy et al., 2015). However, these studies focus only on the formation of the coalition, which is only one aspect of the problem in cooperative game theory (Serrano, 2007). The problem associated with the allocation of collective payoffs among the federation members, which is another important aspect of cooperative game theory (Serrano, 2007), has not been adequately addressed by existing research. Allocation of collective payoffs in a coalition of cloud federation should consider various economic issues centered on the problem of resource and revenue sharing.

The federation generates payoff or revenue by serving customer requests with resources pooled from federation members. Serving customer request is concerned with the placement of application service nodes of customer application on federated cloud resources. To serve the request, the federation platform should select the most optimal service placement plan based on different requirements and constraints. This requires for well-defined rules and methods that govern the use of

pooled resources in such a way that it provides fairness in resource exploitation, provides a fair opportunity for participation in serving customer requests, and most importantly should maximize the benefit of the overall federation. These rules also act as enforcing entities binding each of the members of the federation to work in cooperation, as is required in a coalitional game (Serrano, 2007).

Next, the rules and methods for the allocation of payoffs should be designed in such a way that it incentivizes the cooperative work of the federation members and provides a fair means of revenue distribution. Ill-defined rules and methods lead to unfair & disproportionate allocation of payoffs to federation members, possible promotion of free riders in the system, demotivation of authentic members for cooperation, and decrease the competitiveness and hence affect the sustainability of the federation.

Provided the cost and complexity involved in the formation and management, such as the management of the service level agreement at the federation level (Toosi, Calheiros, & Buyya, 2014), it is crucial to address the aforementioned issues to encourage prospective cloud providers to join the federation. Unfortunately, such crucial issues of economic nature have not received adequate attention in the literature.

Addressing these challenges requires proper scheduling algorithm for the placement of the service nodes of customer applications into one or more federated clouds based on certain rules and supporting the

heterogeneous requirements of applications and consumer preferences. Next, we require the business logic to appropriately incentivize the federation members for their contribution in the federation. The problem of service placement decision making and business logic for revenue allocation are not a trivial problem and are the most significant issues from an economic standpoint, especially challenging the cloud federations incorporated by a large number of geographically distributed providers offering heterogeneous services with varying QoS guarantees. This thesis is centered on these two problems, and the detail descriptions of the problems are presented in Chapter 3 and Chapter 4 respectively.

1.3 Research Objectives and Research Questions

In line with the arguments presented in the Problem Description section, the research work in this thesis attempts to fill the research gap and propose effective models for the governance of cloud federation with clearly defining business relationships. We frame the overall objective of our research as follows.

In a context where the competitiveness of small cloud providers is restricted due to the economies of scale, the objective of this thesis work is to design the economic model for cloud federation that can improve its competitiveness by exploiting the benefits of the economies of scale with fair and attractive incentive mechanism.

A federated cloud can compete against hyper-scale providers only by realizing its full potential and gaining the competitiveness the way that a hyper-scale provider does, for example by increasing capacity utilization (Goiri et al., 2012), providing better QoS (Petcu, 2014), and offering service variety (Toosi, 2014). This can only be achieved by aggregating both supply & demand and mobilizing the federated resource optimized in a way that maximizes the overall benefits of the federation, as true economies of scale come only with the aggregation of both supply side and demand side (Harms & Yamartino, 2010). By aggregating both resource and requests, the spikes in service requests could be absorbed as is done by hyper-scale providers. This leads to achieving higher utilization ratio and maximization of social benefits with better profitability that comes from being able to provide better service guarantee by maintaining the capacity even at a lower level. In addition, such a provision would allow the cloud federation for service provisioning with optimal selection of resources that lead to better customer satisfaction, resulting in maximization of overall social benefits with improved competitiveness.

With the aggregation of both resource and requests, we have a pool of resource and the aggregated requests should be served with 'effective' use of pooled resources. The request constitutes an application that needs to be deployed in the federated resource. This application constitutes various service components, each requiring VM

nodes with different configurations for their deployment. Federation members, on the other end, provide resources in terms of VM instances of different configurations. Serving requests with 'effective' use of pooled resources require optimal mapping of these application service nodes (VM nodes) to a large pool of VM instances, which should be performed by the specialized algorithm, which we name as a Service Placement Algorithm. Following this argument, we derive our first specific objective.

In order to achieve the overall objective, a first specific objective is to propose a Service Placement Algorithm that governs the use of federated resources in such a way that it maximizes the overall federation benefit with customer satisfaction and without discrimination to any provider

We believe that if we could set and codify the resource governance rules in such a way that resources for service placement request are chosen by making an optimal tradeoff among cost and various QoS criteria, where the optimal tradeoff points is set as per the preference specified by each individual customers, then we could have an unbiased and fair way of using the federated resource while at the same time being able to maximizes social benefits of the federation with optimized (with regards to some criteria) placement service as well as being able to reach a wide range of customer with different cost

and QoS needs. Hence, we try to achieve this specific objective by solving the following two research questions.

Research Question 1: What are the relevant and quantitatively measurable decision criteria that an application provider (consumer of federated cloud) would be interested in optimizing while making service placement decision?

Research Question 2: How to capture customer preferences, specify and make a tradeoff of multiple and, possibly, conflicting decision criteria

Research Question 3: How to select a single optimal service placement plan in a very large search space according to the tradeoff derived from individual consumer preference?

Once we have the methods and tools for the governance of the use of resources, next thing we require are the methods and tools for sharing the payoffs, which is generated by the collaborative efforts of federation members. The sustainability of the federation and the cooperation by federation members is founded upon the incentive mechanism that performs a fair allocation of the payoff and at the same time, for the long term sustainability of the federation, incentivizes the activities and features of the federation members that contribute in the value creation of the federation. In line with this argument, we derive our second specific objective as follows.

In order to achieve the overall objective, a second specific objective is to propose a Revenue Sharing Scheme for a cloud federation that ensures a fair allocation of revenue share to federation members and provides attractive incentives to federation members of all characteristics.

We believe that the mechanism that allocates the revenue share in proportion to the contribution made by each of the federation members in the value creation of the federation can provide the desired incentive system that satisfies these requirements. Revenue share in proportion to the contribution would ensure fairness and provide motivation and space for the federation members to involve in activities that would help them contribute more to the value creation of the federation. We try to achieve this specific objective by addressing the following three research questions.

Research Question 4: What features of a federation member contributes to the value creation of the federation and what indicators can we use to measure them?

Research Question 5: Based on the identified indicators, how can we fairly measure the contributions of federation members and allocate the revenue shares according to their contribution?

Research Question 6: Will the proposed scheme be universally attractive in all contexts for federation members of all characteristic types?

We attempt to address these six research questions as research work leading to two different chapters of this thesis. First, three research questions constitute the problem for the one paper, for which we propose a multi-criteria service placement algorithm considering individual consumer preferences (details in Chapter 3). And, remaining three research question forms the problem for the second chapter, where we try to address the problem by proposing a scheme for distributing the revenue shares among federation members in proportion to their contribution in generating it (details in Chapter 4).

1.4 Methodology

A combination of various methodologies was adopted to address the stated research questions.

In relation to Research Question 1, we have conducted an extensive literature review to make a comprehensive list of decision parameters used by previous research related to VM placement in clouds. From the list of all the parameters, we select only those parameters that can be objectively measured and are relevant for external scheduling of the service request, as internal scheduling is not the scope of the research. We also present the subsequent analysis to establish the appropriateness of the decision criteria with respect to the problem description.

To address Research Question 2, we capture consumers' preferences over decision criteria as a pairwise comparison between all

possible pairs and convert them into their corresponding weights resulting in the preference weight vector by applying AHP, and later apply this weight vector in the selection process.

To address Research Question 3, we employ the Fast and Elitist Non-dominated Sorting Genetic Algorithm (NSGA II) (Deb, Pratap, Agarwal, & Meyarivan, 2002) and use the preference weight vector evaluated earlier. Since the resource combination results in a very large search space of solutions, it is impossible to search for the best solution with a brute force approach. Hence we reduce the search space by employing 'natural evolution' inspired multi-criteria optimization algorithms. For this, we employ Fast and Elitist Non-dominated Sorting Genetic Algorithm (NSGA II) (Deb et al., 2002), a state of the art multi-objective optimization method, which is efficient for the simultaneous optimization of multiple criteria and reaches to a set of Pareto optimal placement plans. From the reduced search space, we select the placement plan with best overall fitness evaluated as a function of their position in the objective space and earlier evaluated preference weight vector.

We perform a literature study to find answers to Research Question 4. And, to address Research Question 5, we propose a novel revenue sharing algorithm. The revenue share for each federation member is calculated on the basis of the contribution made by them in value creation of the federation. The federation value in this context

refers to the revenue generated by the cloud providers by working as a coalition. We apply the Shapley Value Method (Shapley, 1953), an approach in Coalitional Game Theory, to generate the payoff vector that allocates the revenue share for each member on the basis of their contributions. Member providers' contributions consider both infrastructure capacity and market strength of the provider. In order to avoid free riders, the contribution in terms of infrastructure capacity is assessed from the amount of actual resource provisioning done to the customer request rather than the resources committed to the federation. And, the contributions in terms of market strength are assessed from the value of service requests brought in to the federation.

And, to address Research Question 6, we perform simulation covering various scenarios with different provider and demand characteristics.

Through extensive simulation covering a wide range of scenarios and data from sources like Amazon Web Service (AWS), Cloud Harmony, Verizon, and Dell, we demonstrate the effectiveness of the proposed service placement algorithm as well as revenue sharing algorithm and provide a comparison with a benchmark approach. We perform simulations to evaluate its performance in comparison to a benchmark approach in terms of standard metrics like Generational Distance (GD) (Veldhuizen, 1999), Spacing (Sp) (Riquelme, Von Lücken, & Baran, 2015; Schott, 1995), and Set Coverage (C)

(Hiroyasu, Miki, & Watanabe, 1999), which are commonly used in operations research field for comparing the performance of multi-objective optimization algorithms in various industry applications. We perform simulations covering various scenarios to demonstrate the effectiveness of the revenue sharing scheme, too, and compare its effectiveness with the benchmark participatory approach.

1.5 Contribution

This work contributes to the existing knowledge in cloud federation research by providing novel algorithms to be applicable to cloud federation and cloud service brokers supported by the evaluation of their performance compared to benchmark approaches in terms of standard metrics. More specifically, the contributions include the following.

- A comprehensive multi-criteria decision model for placing of application service components in federated clouds taking into account cost and as various relevant Quality of Service (QoS) criteria identified from the survey of related literature viz. *Financial Cost*, *Execution Speed*, *Network Latency*, and *Availability.* Consideration of application footprint as one of the factors in the optimization of *Network Latency* offers novelty.
- A service placement algorithm that combines knowledge from two methods, viz. - Analytic Hierarchy Process (AHP) (T. L. Saaty, 1990), a Multi-Criteria Decision Model and Elitist Non-

Dominated Sorting Genetic Algorithm (NSGA II) (Deb et al., 2002), an evolutionary approach to multi-objective optimization method. This contributes to existing knowledge base demonstrating that by augmenting with Multi-Objective Optimization (MOO) algorithms how AHP can, still, be employed to solve Multi-Criteria Decision Making (MCDM) problem that involves a large solution space which makes the search for the best solution impossible with brute force approach.

- Service placement decision making that allows making tradeoffs between cost and QoS criteria according to individual consumers' preferences enabling service differentiation
- Service placement decision making supported by true optimization (i.e. - simultaneous optimization of multiple criteria) that provides a reduction in the large search space of possible placement plans without having one criterion being affected or biased by the optimization of other criteria during the optimization process. This contributes to the knowledge base in Multi-Objective Optimization domain by suggesting that the reduction of the search space of solutions by parallel optimization of multiple objectives before the application of objective weights can yield better results in a multi-objective optimization problem that requires a single final solution.

- State-of-the-art revenue sharing algorithm that provides a novel method of allocating revenue among federation members in proportion to their contribution, where the contribution is evaluated not only from the resource contribution but also the market share that a federation member brings into the federation. It contributes to the research community working on the hot topic of revenue sharing problem in various domains that it is worthwhile to explore the possibility of the use of coalitional game theory, especially the Shapley Value as a potential solution concept
- Implementation of both algorithms and simulation to evaluate the effectiveness and performance in comparison with a benchmark approach using standard metrics.

1.6 Thesis Organization

In this section, we provide an outline for the organization of this thesis work with an overview of the contents included in each chapter. The organization with contents in each chapter and their relationship are depicted in Figure 1.

In Chapter 2, we provide theoretical background that includes the emergence of the idea of cloud federation, concept of cloud federation as to how various researchers view it, benefits of and associated challenges in realizing cloud federation, various research directions in cloud federation research, and finally the gap in existing research

works, which motivates this thesis work, with thesis scope and positioning.

In Chapter 3, we address the first three research questions by presenting user preference based evolutionary multi-objective optimization model for service placement in a cloud federation. We present it as a complete research paper along with associated literature review included within the chapter. This chapter is based on our earlier published paper (Aryal & Altmann, 2018).

In Chapter 4, we address the remaining research questions by presenting a contribution based revenue sharing scheme for cloud federation. In this case, too, we present it as a complete paper along with associated literature review included within the chapter. Like Chapter 3, this chapter is also based on our earlier published paper (Aryal & Altmann, 2017).

In Chapter 5, we conclude our thesis by providing a brief summary of the work, implications of the research work for industry and academia, limitations of the work and related possible future research works.

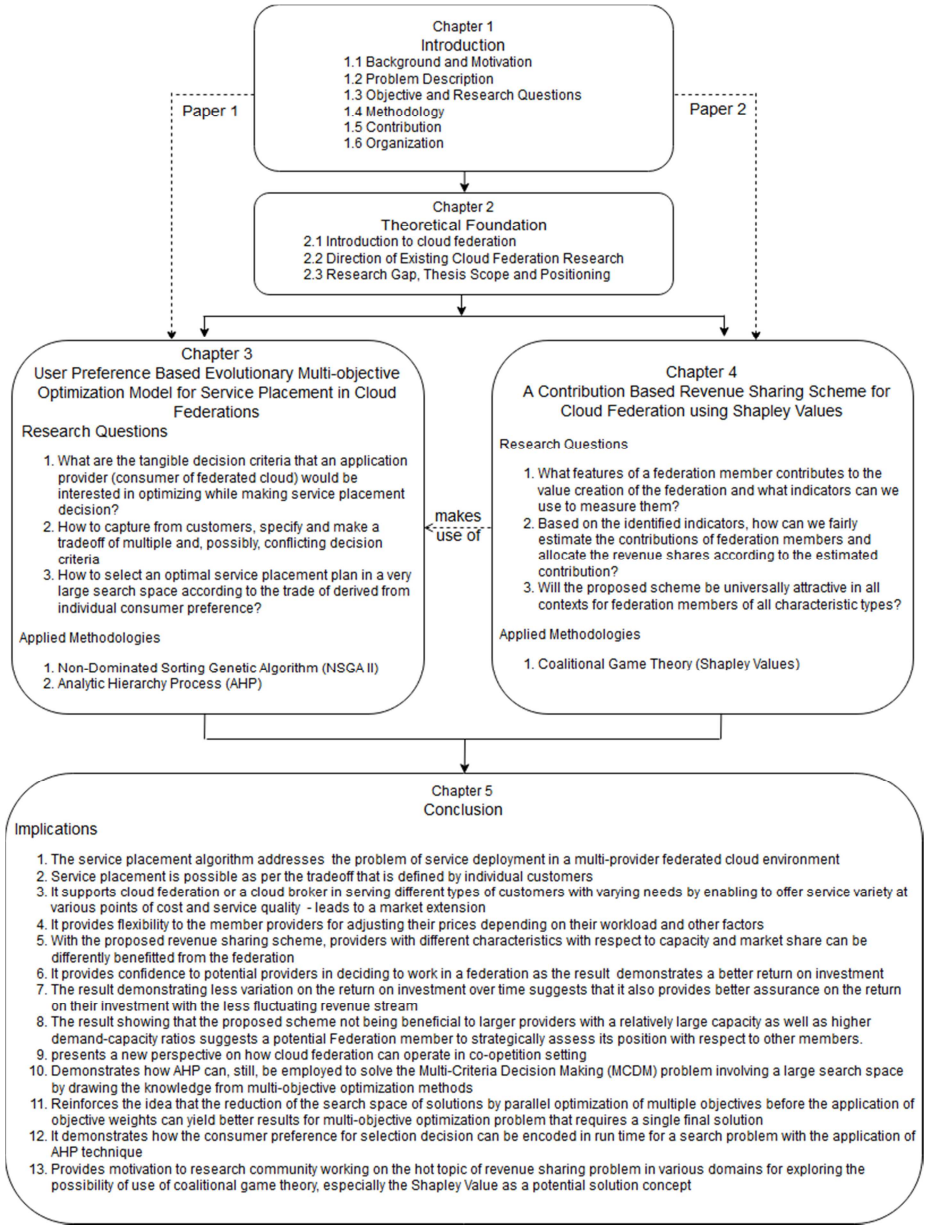


Figure 1: Thesis Organization

Chapter 2. Theoretical Background

2.1 Introduction to cloud federation

2.1.1 Background

Cloud computing allows IT capabilities to be outsourced through internet from data centers that poll large computing resources (Venters & Whitley, 2012). The resource pooling capability and metered service enables clouds to offer resources on demand and allows for pay per use (Mell & Grance, 2011). This enables cloud consumers for rapid deployment of their applications without requiring expert technical skills and infrastructure deployment costs (Harms & Yamartino, 2010). A number of benefits like economies of scale through multitenancy model (Harms & Yamartino, 2010), and flexible costings like pay-as-you-go and pay-per-use makes cloud computing widely adopted by consumers (International Data Corporation, 2018; Rimal, Choi, & Lumb, 2009).

Despite the economic benefits achieved through the economies of scale, cloud computing still suffers the problem of resource underutilization from overprovisioning and SLA violations from under-provisioning (Harms & Yamartino, 2010). Datacenter resource, being finite, limits a cloud provider's' resource scaling capacity (Goher et al., 2017). The discrimination provided by the economies of scale makes small cloud providers less competitive in the cloud service market (Kim et al.,

2014), which is reflected in the current market share structure of IaaS cloud providers. According to Gartner's report, 75% of the IaaS market is being occupied by the largest five providers (Gartner, 2018b). Amazon alone holds a 52% share, which is followed by Microsoft (13%), Alibaba (5%), Google (3%), and IBM (2%).

Strategies and methods for addressing these limitations become important to cloud service providers who are constantly seeking to reduce operational cost, increase profit, and gain competitiveness in the market. Various researchers have identified Cloud federation as a way to address such existing limitations by means of resource aggregation from multiple cloud providers (Ferrer et al., 2012; Haile & Altmann, 2015; Petcu, 2014; Rochwerger et al., 2009).

2.1.2 Concept of Cloud Federation

Cloud Federation, being a relatively new concept, there is a lack of consensus on the concepts including its formation and composition. Some studies have attempted to formalize its concepts by defining the characteristics (Buyya, Ranjan, & Calheiros, 2010; Celesti, Tusa, Villari, & Puliafito, 2010; Haile & Altmann, 2015; Manno, Smari, & Spalazzi, 2012).

Buyya et al. (2010) stated three properties, which they believe are required at minimum to make the cloud federation effective, namely, dynamic expansion of resources, commercialization of resources, and

compliance of established Service Level Agreements between the customer and the cloud provider.

Haile & Altmann (2015) view cloud federation as a strategic alliance among cloud providers, where cloud providers have a cooperation agreement with regards to service component deployment and the use of resources from one another in order to meet varying customer demands.

Celesti et al. (2010) introduced the idea of governance of the cloud federation by a Federation Level Agreement (FLA) – including the technical and economic constraints - the quality of service, charging models, authentication & use restrictions, rewards on QoS satisfaction, and penalties on violations among the CSPs.

Manno et al. (2012) introduced the idea of geographic dispersion of cloud providers in the federation, and also highlight that the federation members need to have autonomy over the services they offer and the resources they possess and that they can leave the organization freely.

From the above discussions, we can view a cloud federation as a voluntary arrangement among a number of cloud providers, which are distributed over different geographic locations, for interconnecting their cloud infrastructures and enabling resource sharing and governed by Federation Level Agreements (FLA).

2.1.3 Benefits and Challenges of Cloud Federations

Cloud federation is considered as a way to address the limitations experienced by individually operating cloud providers. Its benefits to small and medium-sized cloud providers have been acknowledged in a number of literature. Challenges associated with its implementations have also been discussed.

2.1.3.1 Benefits

The first benefit comes in the form of scalability. The benefit of cloud federation enabling a cloud provider in meeting elastic needs through resource scaling, which makes use of federated resources has been acknowledged by various research (Altmann & Kashef, 2014; Aryal & Altmann, 2017; Assis & Bittencourt, 2016; Govil, Thyagarajan, Srinivasan, Chaurasiya, & Das, 2012; Haile & Altmann, 2015; Kim et al., 2014).

The benefit of availability, where a cloud provider can maintain for customer services with reliable multi-site deployments across geographically distributed federated infrastructures, is also discussed in many research work (Aoyama & Sakai, 2011; Govil et al., 2012; Kim et al., 2014; Toosi et al., 2014).

Kim et al. (2014) have studied the economic benefits of cloud federation in terms of economies of scale and network externalities. This study highlights that a cloud federation provides a cloud provider with competitive strength through economies of scale that comes with

improved resource utilization and reduction in cost that is required for maintaining scalability.

Researches have also pointed out that the cloud federation allows to meet regional demands and gain performance benefits by the dynamic distribution of workload to clouds that are closer to customers (Assis & Bittencourt, 2016; Govil et al., 2012; Toosi et al., 2014). Assis & Bittencourt (2016) mention that cloud federations can be useful in addressing legal constraints, which may be potentially imposed by administrative regulations. Some state may have strict requirements on cross border transfer of some data. In such a case, a cloud provider can make service provisioning with cloud infrastructure in the federation such that no violation of the state regulations occurs.

Other benefits include an increase in profit with improved resource utilization ratio (Assis & Bittencourt, 2016; Govil et al., 2012; Toosi et al., 2014) and performance guarantee by borrowing resources from other cloud providers (Govil et al., 2012). Table 1 presents a summary of the benefits of cloud federation with related works that highlight those benefits.

Table 1: Benefits of cloud federation

Benefits	*Description*	*Related Works*
Scalability	Increase capability to meet elastic needs by resource scaling with federated resources	(Altmann & Kashef, 2014; Aryal & Altmann, 2017; Kim et al., 2014; Toosi et

		al., 2014)
Economies of Scale	Gain competitive strength from economies of scale with improved resource utilization and reduction in cost required for maintaining scalability	(Kim et al., 2014)
Availability	Maintain availability of customer services with reliable multi-site deployments across federated infrastructures that are geographically distributed	(Aoyama & Sakai, 2011; Govil et al., 2012; Kim et al., 2014; Toosi et al., 2014)
Meet legal requirements	Address legal constraints such as restrictions on cross border data transfer by deploying applications in clouds meeting the legal compliance	(Assis & Bittencourt, 2016)
Address Regional demand	Meet regional demands and gain performance benefits by the dynamic distribution of workload to clouds that are closer to customers	(Assis & Bittencourt, 2016; Govil et al., 2012; Toosi et al., 2014)
Utilization Ratio	Increase profit with improved resource utilization	(Haile & Altmann, 2015; Kim et al., 2014)
Performance	Offer a performance guarantee by borrowing resources from other cloud providers	(Govil et al., 2012)
Energy Efficiency	Minimization of energy consumption by VM migration and shutting down	(Toosi et al., 2014)

some servers when appropriate

Thus, following the arguments made by Petcu (2014), we can say that with mutual sharing of resources, cloud providers can solve service limitations problem with resource aggregation, ensure Quality of Service guarantees with efficient deployments, improve cost-efficiency through improved resource utilization ratio, and maintain the availability of Cloud services through reliable multi-site deployment.

2.1.3.1 Challenges

There are also a number of challenges that need to be addressed for the realization of Cloud Federation. Toosi et al. (2014) have presented a comprehensive analysis of challenges that need to be addressed in the inter-cloud environment. These challenges cover wide topics including - resource provisioning, virtual machine & data portability, service level agreements, security, monitoring, economy, network, and autonomy. A description of the challenges in running a cloud federation is presented in Table 2.

Table 2: Challenges in realizing cloud federations

Challenge	Description	Related Works
Resource Provisioning	The challenge of resource provisioning includes the discovery of resources within	(Aryal & Altmann, 2017; Coronado & Altmann, 2017;

	the federation, selection of appropriate resources for service composition and allocation of resources to service request	Hassan et al., 2017; Hassan, Al-Wadud, & Fortino, 2015; Z. Lu, Wen, & Sun, 2012; Mashayekhy et al., 2015; Niyato, Vasilakos, & Kun, 2011; Samaan, 2014)
Portability	The challenge within portability class is related to live migration of virtual machine between nodes of different clouds and the ability to export data from an application in one cloud to an application to an application in another cloud.	(Di Martino, Cretella, & Esposito, 2015; Parameswaran & Chaddha, 2009; Thabet, Boufaida, & Kordon, 2014)
Service Level Agreement and Monitoring	This challenge is related to enforcing the service level agreements at the federation level where there might be a conflict between objectives of the federation and differing policies of federation members. This includes defining rules for	(Amato, Liccardo, Rak, & Venticinque, 2012; Carlini, Coppola, Dazzi, Ricci, & Righetti, 2011; Clayman et al., 2010; Rak, Venticinque, Echevarria, & Esnal,

	federation level agreement and monitoring its compliance	2011)
Network and Security	Network related challenge include network virtualization and addressing in order to support VM migration. Security in cloud federation requires the establishment of trust as well as management of identity and authorization for enabling legitimate access of resources across the federation.	(Abawajy, 2009; Celesti et al., 2010)
Economy	It includes the challenge associated with pricing policies, resource use accounting, and fair method of incentivizing federation members	(Breskovic et al., 2011; El Zant, Amigo, & Gagnaire, 2014; Haile & Altmann, 2015)

2.2 Direction of Existing Cloud Federation Research

Due to the promises and various challenges that it embodies, cloud federation and federated cloud computing environment have been the subject of research interest in the recent years (James Cuff, Ignacio M. Llorente, Christopher Hill, 2017).

A significant amount of research has been carried out towards proposing architectures and toolkits for the cloud federation. Research outcome in this direction include various architecture like Reservoir (Rochwerger et al., 2009), CompatibleOne (Yangui, Marshall, Laisne, & Tata, 2014), and BASMATI (Altmann et al., 2017), and platforms and toolkits for service provisioning, such as OPTIMIS (Ferrer et al., 2012) and Broker@Cloud ("Broker@Cloud," 2015).

Closely associated with the architecture and platform, the problem of interoperability among various clouds is also a topic of interest for a number of (Di Martino et al., 2015; Parameswaran & Chaddha, 2009; Thabet et al., 2014), where studies focus on methods of live migration of Virtual Machines across clouds (Satpathy, Addya, Turuk, Majhi, & Sahoo, 2018), data portability between various nodes across different clouds (Kaur, Sharma, & Kahlon, 2017), and ensuring security between interoperating clouds (Abawajy, 2009; Celesti et al., 2010).

Ample research has been conducted to address the challenge of resource allocation, where the researchers are interested in efficient ways of discovering resources for Virtual Machine placement across the federation (Pittaras et al., 2015), resource selection methods by optimizing multiple objectives on demand side (Aryal & Altmann, 2018), and optimized resource allocation based on various criteria on the supply side (Sim, 2016).

Various researches have been carried out focusing on the issues relating to operation and management of cloud federation. Research works in this direction include ways of enforcing Service Level Agreements between various providers , which is named as so-called Federation Level Agreements (FLA) (Toosi, Calheiros, Thulasiram, & Buyya, 2011), on top of the agreements that exist between customer and a cloud provider (Amato et al., 2012; Carlini et al., 2011), and tools & techniques for monitoring to ensure compliance of the Service Level Agreements (Clayman et al., 2010; Rak et al., 2011).

A body of research focuses on management and economic aspects such as resource use accounting and billing (Elmroth, Márquez, Henriksson, & Ferrera, 2009), and pricing policies (Goiri et al., 2012; Toosi et al., 2011; Toosi, Thulasiram, & Buyya, 2012) that take place among the members of the federation. Formation of cloud federation as a coalitional game has also been studied from an economic standpoint (Aryal & Altmann, 2017; Coronado & Altmann, 2017; Hassan et al., 2017; Z. Lu et al., 2012; Mashayekhy et al., 2015; Niyato et al., 2011). A body of research focuses on the formation of cloud federation from the viewpoint of maximizing of individual benefit (Samaan, 2014), and social benefit (Hassan et al., 2015).

On the implementations side, EGI federated cloud (Fernández-del-Castillo et al., 2015), a European Intergovernmental Research Organization's initiative, is a successful example. EGI federated cloud

federates private clouds of academic institutions that include hundreds of data centers located across the globe, mostly in Europe; and provides computing and storage resources (IaaS service model) to researchers (Fernández-del-Castillo et al., 2015).

2.3 Research Gap, Thesis Scope and Positioning

With the wide acceptance of the benefits of the cloud federation, it is natural to expect its expansion beyond the academic community and reach among commercially operating small and medium-sized cloud providers (Kim et al., 2014). Despite the aforementioned potentials, ample research in the field, and successful use case, however, cloud service market has not seen any commercial federation in operation, so far (Coronado & Altmann, 2017).

Research point out unresolved economic aspects as an important hindering factor (Breskovic et al., 2011; Haile & Altmann, 2015). It is clear that commercially operating cloud providers do not seem to be willing to cooperate without the appropriate resolution of the economic aspects. This argument is also supported by industry players participating in a panel discussion in a recent conference (ieeeCESocTV, 2018).

With respect to economic challenges of cloud federation, there exists a body of research that deals cloud federation as either non-cooperative or cooperative coalition and studies the problem by applying relevant Game Theory (Hassan et al., 2017; Mashayekhy et al., 2015). Despite a

large body of research, the problem associated with the allocation of collective payoffs among the federation members, which is an important aspect of cooperative game theory, has not been addressed by existing research.

The payoff allocation mechanism requires well-defined rules and methods that govern the use of pooled resources and the rules and methods that perform the allocation of payoffs in such a way that it incentivizes the cooperative work of the federation members and provides a fair means of revenue distribution. These rules and methods are crucial to address the aforementioned economic issues and motivate small cloud providers to join the federation. Unfortunately, such crucial issues of economic nature have not received adequate attention in the literature.

Addressing these problems requires two algorithms, namely a service placement algorithm that governs resource sharing and a revenue-sharing scheme that governs the appropriate distribution of payoff or revenue among federation members. These two problems constitute the core work of this thesis and are dealt with in chapter 3 and chapter 4 respectively.

Chapter 3. Consumer Preference Guided Multi-criteria Model for Economics-based Service Placement in Federated Clouds Using Evolutionary Multi-objective Optimization and AHP

3.1 Introduction

3.1.1 Motivation

Cloud federation is a widely researched topic during the last few years. It requires more serious attention, at present, when Gartner has reported that 75% of global Infrastructure as a Service (IaaS) cloud market is occupied by only five hyper-scale providers (Gartner, 2018b). The attention should be given for its potential in addressing the challenges, improving the competitiveness, and thus increasing the market share of smaller ones (Kim et al., 2014). Cloud federation, a strategic alliance among cloud providers with cooperation agreement for resource sharing and services deployment (Haile & Altmann, 2015), has been considered as a way to address the challenges that originate, especially, from the anticompetitive externalities due to economies of scale (Altmann & Kashef, 2014; Mohammed, Altmann, & Hwang, 2009). It is believed to possess the potential in mitigating major challenges of

IaaS providers including resource limitation (Goher et al., 2017), inefficient resource utilization (Goiri et al., 2012), limited service quality (Petcu, 2014), and limited service variety (Toosi, 2014).

A federated cloud can compete against hyper-scale providers only by realizing its full potential and gaining the competitiveness the way that a hyper-scale provider does, for example by increasing capacity utilization (Goiri et al., 2012), providing better QoS (Petcu, 2014), and offering service variety (Toosi, 2014). This can only be achieved by aggregating both supply & demand (Harms & Yamartino, 2010), and optimally mobilizing the federated resources by ensuring fairness to all the members. This requires for effective *Service Placement Algorithm*, which provides the policies & rules that govern and methods that facilitate the selection of federated resources in serving customer requests.

3.1.2 Problem Description

Resources in a federated cloud constitute a large number of Virtual Machine Instance types offered by various IaaS cloud providers. A VM Instance represents a bundle of infrastructure resources characterized by different configurations e.g.- CPU cores, memory size, and storage along with price and other service quality parameters (X.-F. Liu et al., 2018). The permutation of different VM instances from multiple clouds leads to a vast number of

possible placement plans each being unique in terms of cost and Quality of Service (QoS) guarantees.

Each service placement request requires a selection of the most appropriate placement plan specifying where each service nodes of the application are to be deployed. This should be done according to different decision criteria including cost and various QoS criteria such as execution speed, system availability, network latency, and load balancing (Bañares & Altmann, 2018). Because of the objective conflicts between the decision criteria, the selection process requires a careful tradeoff between them (Deb, 2014). Besides, for the sustainability of the federation, the tradeoff should also ensure fair treatment to each provider and the maximization of the overall benefits of the federation. Finding the best service placement plan making the tradeoff requires an exploration of a large search space.

The large search space makes the selection of a placement plan an NP-Hard problem (de Carvalho, Trinta, Vieira, & Cortes, 2018; Ziafat & Babamir, 2019). The problem being NP-Hard, exhaustive search (i.e.- brute force) for the optimal service placement plan becomes computationally impractical (Garey, 1979). Thus, due to the involvement of these sophistications, the service placement decision making becomes a non-trivial and an interesting research problem (Altmann & Kashef, 2014; Heilig, Buyya, & Voß, 2017;

Ziafat & Babamir, 2019), which requires optimization of multiple objectives. Researchers have attempted to address this problem by proposing service placement algorithms.

Many of those research focus only on internal scheduling (Feng, Wang, Zhang, & Li, 2012; Manasrah, Smadi, & ALmomani, 2017; Uzbekov & Altmann, 2016; S.-H. Wang, Huang, Wen, & Wang, 2014; Ziafat & Babamir, 2019). They aim at optimally selecting the physical machine within a cloud for VM placement. Some researches consider only a single cloud (Nawaz et al., 2018) (Coutinho, Drummond, & Frota, 2013), which ignores the possibility and benefits of involving multiple clouds in the placement plan. Large number of existing research focus on optimizing only one objective such as cost (Altmann & Kashef, 2014; Chaisiri, Lee, & Niyato, 2009; Zhang et al., 2014), energy consumption (Baker et al., 2018; Dupont, Schulze, Giuliani, Somov, & Hermenier, 2012; X. Wang & Liu, 2012), resource utilization (Calcavecchia, Biran, Hadad, & Moatti, 2012; Sayeedkhan & Balaji, 2014), traffic (Jayasinghe et al., 2011; Kanagavelu, Lee, Le, Mingjie, & Aung, 2014), load balancing (Shi & Hong, 2011; Tian, Xu, Chen, & Zhao, 2014), QoS (Bobroff, Kochut, & Beaty, 2007), or availability (Wenting Wang, Chen, & Chen, 2012). Algorithms optimizing multiple objectives either provide a set of solutions (Claro, Albers, & Hao, 2005), which requires the decision maker to select one, or provide a single solution but only performs weak

optimization because of the adopted problem-solving approach. Such problem solving approach transforms a multi-objective optimization problem (MOO) into a single-objective optimization problem (SOO) (Zitzler, Deb, & Thiele, 2000), for instance, by linear aggregation of multiple objectives (Coutinho, Drummond, Frota, & de Oliveira, 2015), which is also known as the Scalarization method (Marler & Arora, 2004).

This way, we observe that very few researches consider service placement plans that involve resource selection from more than one cloud. Previous researchers have not adequately considered individual consumer preferences for optimization to reflect the uniqueness of each application characteristics that suggest for the service placement plans optimized on different tradeoff points. There exists a gap with regards to the economic-based true optimization of multiple criteria and also with regards to the consideration of geographic footprint of the application, i.e.- the regions having a significant number of application users, in the optimization process.

3.1.3 Research Objectives and Research Questions

In this problem context, the objective of this research is to propose service placement algorithm that optimally places the application service components in the federated cloud resources, where the service placement plan is identified by true (simultaneous)

optimization of multiple objectives taking into consideration the unique tradeoff requirements of each application as stated by each individual customer and taking into account the geographic footprint of the application.

To achieve the stated objectives, three research questions have been formulated - i) what are the relevant and measurable decision criteria that an application provider (consumer of federated cloud) would be interested in optimizing while making service placement decision? ii) How to capture from customers, specify and make a tradeoff of multiple and, possibly, conflicting decision criteria? And, iii) How to select an optimal service placement plan in a very large search space of potential service placement plans according to the tradeoff?

3.1.4 Methodology

We have conducted an extensive literature review to identify objectively measurable and relevant criteria. In a subsequent analysis, we examined their appropriateness with respect to the problem description, i.e. - we performed an analysis of whether the criteria are relevant to consumers who seek an optimal deployment of their application on resources spread across a federated cloud. We, then, developed a service placement algorithm that optimally places the service components in federated resources by considering

the consumer's preferences over the decision criteria identified in the previous step.

For this, consumers' preferences are captured as a pairwise comparison between various decision criteria, converted them into their corresponding weights by applying the Analytical Hierarchy Process (AHP) (T. L. Saaty, 1990). Afterward, we applied these weights to find the most suitable single service placement plan among a set of known Pareto optimal placement plans, which were identified through the simultaneous optimization of multiple criteria. The multi-criteria optimization process in the proposed service placement algorithm is based on Elitist Non-dominated Sorting Genetic Algorithm (NSGA II) (Deb et al., 2002), a state of the art multi-objective optimization method.

We developed a simulation program for the algorithm in python and ran simulations covering wide scenarios with reference data from sources that include Amazon, Gartner, Verizon, and Dell to demonstrate its effectiveness. We also evaluated its performance in comparison to benchmark approach in terms of standard metrics like Generational Distance (GD) (Veldhuizen, 1999), Spacing (Sp) (Riquelme et al., 2015; Schott, 1995), and Set Coverage (C) (Hiroyasu et al., 1999), which are commonly used in operations research field for comparing the performance of multi-objective optimization algorithms for various applications.

3.1.5 Contribution

Our contributions include:

- A comprehensive multi-criteria decision model for service placement in the federated cloud with the identification of measurable and relevant decision criteria that include financial cost, execution speed, network latency, and system availability.
- A service placement algorithm that combines knowledge from two methods, viz. - Analytic Hierarchy Process (AHP) (T. L. Saaty, 1990), a method for Multi-Criteria Decision Making (MCDM) and Elitist Non-Dominated Sorting Genetic Algorithm (NSGA II) (Deb et al., 2002), a state-of-the-art method for true multi-objective optimization.
- Reflection of individual consumer preferences in economic-based service placement decision making
- Service placement with true optimization, i.e.- simultaneous optimization of financial cost and QoS parameters.
- Assured fairness in federated resource utilization with resource selection according to service placement algorithm that is guided by consumer preferences with no space for impartiality
- Algorithm implementation and simulation to evaluate the effectiveness and performance in comparison with a benchmark approach using standard metrics.

- Important managerial and academic implications

3.1.6 Organization

The chapter is organized as follows. In Section 3.2, we discuss the state of the art on service placement in the federated clouds. System Modelling is detailed in Section 3.3. Details on the simulation are given in Section 3.4. Presentation and analysis of the simulation results are given in Section 3.5. And finally, the conclusion is presented in section 3.6.

3.2 State of the Art

3.2.1 The Challenge of Service Placement in Federated Cloud

Cloud service market, with its tremendous growth, consists of a vast number of cloud services entailing various characteristics in terms of provider, technology, service levels, and pricing models (Do et al., 2016). In this context, one of the resource allocation challenges in federated cloud is to make an optimal service placement plan (Altmann & Kashef, 2014) (Heilig et al., 2017) (Ziafat & Babamir, 2019), which maps the application service nodes to various cloud infrastructure resources that involve such a variety (Aryal & Altmann, 2018).

Cloud providers provide infrastructure resources in the form of Virtual Machine(VM)s, which refers to a bundle of infrastructure

resources characterized by various specifications (e.g. - CPU cores, memory, storage) and their prices (X.-F. Liu et al., 2018).

Application owners who are in need of deploying their applications or a broker who provides application deployment service utilize these virtual machines from appropriate providers and data centers located at different locations. The decision regarding the selection of such resources for application deployment is referred to as a service placement decision.

In order to truly benefit from the federated clouds, service placement decisions should consider the placement of the services on multiple clouds that may be geographically distributed across the globe (Buyya et al., 2009). Application deployment done in such a way involves, for each of the services that comprise the application, a selection of VM types of certain specifications, provider, and data centers. The possibility of making service placement plans that involves multiple clouds with multiple VM types on offer leads us to a vast number of potential service placement plans (de Carvalho, Trinta, Vieira, et al., 2018), each being different in terms of cost and QoS parameters.

One service placement plan can be superior to another in some aspects while potentially being inferior in other aspects. Some placement plans may be lower in cost but may be such that the deployment is geographically too distant from the majority of the

application users, which leads to higher communication latency (Wei, Zhou, Yuan, & Yang, 2018). Such deployment plans may not be appropriate for highly interactive application where response time is critical (Arapakis et al., 2014).

Some other deployment plans may provide a slightly lower degree of availability but may offer better computing capability with more number of CPU cores and higher memory size within the same budget limit. Different applications have different levels of criticality (Jeferry et al., 2015), and hence have different requirements. Applications which have the non-significant effect of occasional system downtime may be significantly benefitted by exploiting the computing capability most of the time at the expense of occasional system downtime. This is because larger memory size can support more application users and offer better application response time by reduction of page swapping with secondary storage (Tyson, 2000), and a higher number of CPU cores provide better response time especially for multithreaded applications (Ohlhorst, 2010).

Memory intensive application requires being deployed on VMs featuring larger memory size while CPU intensive application and application with multithreaded architecture will be significantly benefitted by faster CPU and more number of CPU cores

respectively. The choice of VMs during service placement decision should also consider these issues.

Deployment of multiple instances of services at different geographic locations in close proximity of a large number of users can provide better response time to users by minimizing communication latency (Wei et al., 2018). However such deployments may not always be effective, for example, in case of database-intensive application, which may involve significant database synchronization cost (Smit, Shtern, Simmons, & Litoiu, 2012).

This means each application have its own specific requirements and is differently affected by the decision criteria such as cost, performance, and availability for service placement. And, hence, the selection of service or service composition requires optimization techniques that are driven by economic models and should optimize both user-centric parameters that include budget and response time as well as resource centric parameters that include utilization, reliability, availability, and incentives (Buyya et al., 2010).

In order to find an optimal match for the application requirements, it is necessary to explore the complete search space of possible placement plans. Permutation of resources across the federation results in a large search space making the selection an NP-Hard problem. Exhaustive search (Brute force approach) is

computationally impractical for such problems, and hence other effective approaches are required. Thus, it is important to have an effective algorithm that takes into account the uniqueness of each application requirements, optimize on each of the placement criteria according to consumer preferences, and find optimal match for service placement (Buyya et al., 2010).

3.2.2 Existing Works on Service Placement in Federated Cloud

The process of identifying worthy resources within a set of federated resources for service composition is difficult because of the variation in application requirements and heterogeneity in provider resources (Liaqat et al., 2017). The core of this process is the optimization algorithm that considers all parameters that influence the selection decision and hence multi-objective optimization is the best approach in solving such a problem.

Multi-objective optimization is a popular research topic in the area of operations research. Recently, it has also found application in the field of cloud computing. A number of researches have been carried out for optimizing the resource selection by use of multi-criteria optimization techniques. Table 3 provides a list of existing approaches for service selection or composition decision making with the multi-objective optimization process.

Table 3: Existing works in relation to service placement with multi-objective optimization

Work	Approach	Scheduling Scope (Internal (I) Vs. External(E))	Supports multiple cloud providers?	Supports service composition?	Objective Function									
					Latency	Reliability	Response Time	Execution time	Cost (Profit)	Energy	Res. Utilization	Traffic	Load Balance	Availability
(Altmann & Kashef, 2014)	Brute Force	E	Yes	No					x	x		x		
(Manasrah et al., 2017)	Heuristics	I	No	No			x		x					
(Nawaz et al., 2018)	Markov-Chain and Best-Worst Fit	E	Yes	No		x	x		x					x
(Simarro, Moreno-Vozmediano, Montero, & Llorente, 2011)	Integer Programming	E	Yes	Yes			x		x					
(C. Liu, Shen, Li, & Wang, 2014)	Genetic Algorithm	E	No	No						x		x		
(Babu & Samuel, 2014)	Bin Packing (Best Fit - Worst Fit)	I & E	No	No						x	x			
(S.-H. Wang et al., 2014)	Genetic Algorithm	I	No	No	x					x			x	
(Tordsson, Montero, Moreno-Vozmediano, & Llorente, 2012)	Integer Programming	E	Yes	Yes				x	x					
(Heilig et al., 2017)	Adaptive Large Neighborhood Search	E	Yes	Yes	x				x					
(Díaz, Entrialgo, García, García, & García, 2017)	Integer Programming And Binning	E	Yes	Yes			x		x					
(Coutinho et al., 2013)	Greedy Randomized Adaptive Search (GRASP)	E	Yes	No				x	x					

(Coutinho et al., 2015)	Extended (Coutinho et al., 2013) with weighted sum objective function	E	Yes	Yes	x			x	x					
(Ziafat & Babamir, 2019)	Linear Programming algorithm for geographically distributed DCs and GrEA for the selection of VM	I	No	No		x	x		x			x		
(Kumrai, Ota, Dong, Kishigami, & Sung, 2017)	Multi-objective Particle Swarm Optimization (MOPSO)	E	Yes	Yes			x		x	x				
(Wu, Tang, Tian, & Li, 2012)	Genetic Algorithm	E	No	No						x		x		
(Feng et al., 2012)	Multi-Objective Particle Swarm Optimization	I	No	No			x	x						

Altmann & Kashef (2014) suggest a cost model for service deployment in federated hybrid clouds with a detailed analysis of various cost factors involved. They apply the proposed cost model in a brute force algorithm for cost minimization in making a service placement decision. A significant minimization of spending in computational services is achieved by the use of the algorithm in-service placement decision making.

Manasrah et al. (2017) propose a routing policy for selecting a data center based on heuristics, which aims at optimizing (minimizing) the response time when routing the user requests. The policy considers bandwidth, delay, and request size to achieve the level of optimization in response time within an acceptable cost range. The simulation carried out with the range of light and heavy workloads

demonstrated the effectiveness of the policy. This optimization work, having been aimed at selecting a data center for task scheduling, may not be applicable for service composition that makes a selection of resources from multiple clouds.

Employing Markov-chain and Best-Worst method, the service selection method proposed by Nawaz et al. (2018) captures user preferences that are linked to the QoS parameters of available services. Then the Best-Worst method is applied to generate a ranked list of services as per the captured user preferences. Service selection is done on the basis of the ranked list. The authors evaluated the performance of the proposed model with a case study of real data from Amazon EC2 on QoS performance. It provides an effective way for the selection of cloud for service placement but does not support for service placement on cloud resources across different clouds in the federation.

An optimization method is proposed by Simarro et al. (2011) which is applicable for service composition. It employs Integer programming to find the optimal distribution of VM in data centers across multi-clouds with the aims of either minimizing cost or maximizing performance, but not both at the same time. One parameter is considered as constraints when optimizing the other parameter and vice versa. The proposed approach, though considers multiple objectives, may not truly be considered as a solution for

multi-objective optimization as only one parameter is optimized at a time.

C. Liu et al. (2014) proposed an optimization model that employs the idea of sorting procedure from Elitist Non-dominated Sorting Genetic Algorithm (NSGA II) into the Grouping Genetic Algorithms (GGA). The model optimizes energy consumption by minimizing network traffic and the number of active physical servers. This model is suitable for a provider who is willing to minimize energy consumption; but, however, is not applicable to the service composition from multiple clouds.

Other Algorithms for optimized energy consumption is proposed by Babu & Samuel (2014), and S.-H. Wang et al. (2014). Both of these works make use of Bin-Packing based algorithm. Unlike the approach by C. Liu et al. (2014), where the target is the selection of physical resources for VM placement, the works of Babu & Samuel (2014) and S.-H. Wang et al. (2014) consider both task scheduling and VM placement decisions. Job scheduling is done as per the best-fit approach and the VM placement is done as per worst-fit approach. The proposed best-fit -worst-fit strategy is said to use a minimum number of physical machines to host the VMs thereby minimizing the energy consumption and network traffic. For the same reason as for the algorithm proposed by C. Liu et al. (2014), these approaches, too, are inapplicable for service composition.

Nature-inspired algorithms have also been studied by researchers for the optimization of energy consumption (Feller, Rilling, & Morin, 2011; Wu et al., 2012). Wu et al. (2012) have proposed an algorithm based on Genetic Algorithm, which generates a VM placement plan such that the energy consumption of both network equipment and servers are minimized. Similarly, Feller et al. (2011) attempts to model the problem of tasks consolidation as an instance of the multi-dimensional bin-packing (MDBP) problem and solves the optimization problem for minimum energy consumption applying Ant Colony Optimization.

Tordsson et al. (2012) proposed an optimization algorithm that is applicable to service composition. By applying integer programming formulations, the proposed model achieves a balance of the number of VMs purchased from among different cloud locations to optimize cost and performance. The algorithm considers total VM capacity, which is assumed to reflect the performance, as an optimization function to maximize. It considers maximum cost and other parameters like load balancing and hardware requirements as constraints. The authors demonstrate that the multi-cloud deployment offers improved performance and reduced cost in comparison to single cloud deployment.

Another algorithm that could be applicable for service composition is proposed by Heilig et al. (2017). Extending the Large

Neighborhood Search (LNS) by employing multiple destroys and repair heuristics, the authors demonstrate significant cost reduction with the algorithm. They also show the impact of latency reduction in cost, which often requires paying a higher price. Also, it is shown that the latency can be improved in conjunction with cost reduction by having geographic flexibility. The optimization of multiple objective problems is done by converting it to a single objective optimization problem with the application of the weighted sum method.

Díaz et al. (2017) have presented an optimization technique for the allocation of the virtual machines required for service deployment that aims at minimizing cost by exploiting the price differences of reserved and on-demand pricing mechanisms while ensuring the required performance level. The optimization technique is based on integer programming and bin packaging and considers different availability zones and variety in virtual machine types for different providers. The algorithm is helpful in getting the benefit of the discounts offered by providers for reserved virtual machines.

The algorithm proposed by Coutinho et al. (2013), is a heuristics based multi-objective optimization algorithm that aims to minimize execution time. As in the algorithm proposed by Heilig et al. (2017), the weighted sum approach is employed for solving a multi-objective optimization problem by converting it to a single

objective optimization problem. The authors argue that the setting of these weight values gives the appropriate optimal solution. The work is further extended by the authors in Coutinho et al. (2015) by adding the communication costs, execution time and financial cost in the weighted sum objective function.

Multi-objective optimization research works such as Coutinho et al. (2013) and Nawaz et al. (2018) are effective with respect to the selection of a provider service. However, since the proposed methods are aimed at selecting a single provider service, it is unable to tap the benefits that could potentially be achieved by service composition that makes use of multiple providers that are geographically distributed, rather than a single provider in one location.

Targeted at the optimization need of a provider with multiple DCs, Ziafat & Babamir (2018) presents a Grid-Based Evolutionary Algorithm for service placement. Although it considers multiple objective functions, due to the inherent nature of the algorithm that it is based on, it puts equal importance to all the objective functions and hence unique characteristics of each application are ignored.

Multiple Objective Particle Swarm Optimization (MOPSO) algorithm has also been studied in the inter-cloud computing environment (Feng et al., 2012; Kumrai et al., 2017). While Kumrai et al. (2017) employed MOPSO for service composition (VM

placement) with the aim of minimization of energy consumption and response time and maximization of brokers profit in the IoT environment, Feng et al. (2012) proposed MOPSO based algorithm for optimization of task scheduling in a cloud computing environment.

Claro et al. (2005) have worked on the service composition problem that performs simultaneous optimization of multiple objectives. The authors approach the problem as a multi-objective optimization problem and their algorithm, which is based on NSGA II, provides a set of Pareto optimal solutions, which the broker can select according to its preference of one objective function over the other. Due to the inherent nature, it provides a set of solutions on the Pareto optimal fronts, which leads to a need for a decision maker to choose one among a set of presented solutions in the Pareto-optimal front. This requires human intervention and restricts a broker for automated orchestration that should be performed based on SLA requirements as well as application and user behavior.

Such a problem is addressed in the optimization approach as in Coutinho et al. (2013), which utilizes a weighted sum objective function in the heuristic algorithm to convert a multi-objective optimization problem to a single objective. This approach provides a single solution and hence may make technically feasible for automatic orchestration of applications with the manual intervention

being required. However, due to the conversion of the objective function from multiple to a single, there is a probability of reaching a solution that is not among the Pareto-optimal solution.

As a summary, we can find extensive researches on service placement in federated clouds taking into considerations of multiple objectives, too. However, many of these research works either focus on task scheduling (Feng et al., 2012; Manasrah et al., 2017; S.-H. Wang et al., 2014; Ziafat & Babamir, 2019), or do not consider the possibility of service placement across multiple providers (Coutinho et al., 2013; Nawaz et al., 2018).

Works on service placement with the optimization of multiple objectives either perform true optimization but provide a set of multiple solutions (Claro et al., 2005), or provide a single solution but solve the multi-objective optimization problem by converting it to a Single-objective optimization problem with linear aggregation of multiple criteria (Coutinho et al., 2015). Algorithms that suggest a set of multiple solutions (Claro et al., 2005), again, require the decision maker to choose one among the several ones. And, algorithms that solve the multi-objective optimization problem by converting it to a Single-objective optimization problem, like in Coutinho et al. (2015), does not truly optimize multiple objectives. Rather, they do a weak optimization by virtue of the problem-

solving approach as the weights of one objective can impact the optimization of other objectives (Marler & Arora, 2004).

Optimization techniques have been employed in solving resource allocation problems other than cloud computing, too. Hwang (2001) worked on bandwidth management model for differentiated service network with interconnection to integrated service network by employing network economic approach for solving a prominent resource allocation problem of backbone Internet Service Provider (ISP).

In addition, although some research exists in the closely related domain, such as energy consumption optimization for mobile phones (Al-athwari & Altmann, 2015), and the selection of cloud software service (Rohitratana & Altmann, 2012), which consider consumer preference in the optimization process, research focusing on the optimization of the federation of IaaS cloud providers are yet to observe it.

This shows that existing works on service placement do not collectively address these problems and fail to provide a single service placement plan derived from true/simultaneous optimization of multiple decision criteria (including the application footprint, a novel concept) considering individual consumer preferences and composition of resources from geographically distributed multiple clouds.

3.3 System Model

3.3.1 Use Case and the Architecture of Service Placement Framework

The use case and architecture for service placement is shown in Figure 2. In the use case, a cloud consumer, who is in need of a cloud service for the deployment of its application, sends a request to a cloud provider who is a member of the cloud federation.

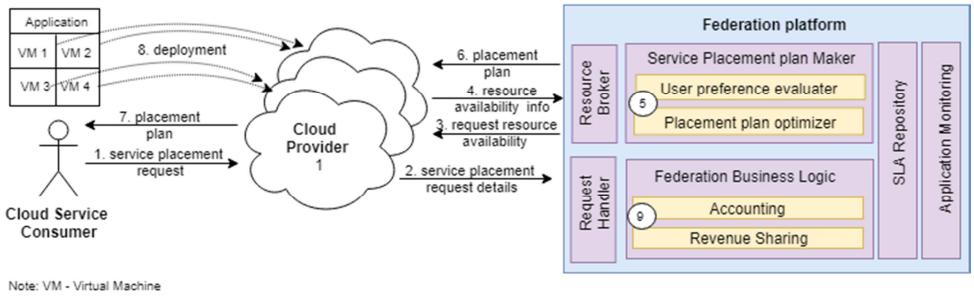


Figure 2: Use case and architecture of Service Placement Framework

The request is forwarded to the cloud federation platform where the *Request Handler* component is responsible to capture the application requirements. The customer is required to provide details of application requirements such as application topology, which contains node requirements (configuration of application service nodes) and their data communication relationships. In addition, the customers state their preference over various decision criteria required for application service placement decision making in the form of pairwise comparisons which is later converted to

preference weight vector by *User Preference Evaluator* sub-component (details in section 3.3.3). The application topology, which is specified following industry standards such as the Topology and Orchestration Specification for Cloud Applications (TOSCA) (OASIS, 2017), and the preference weight vector will form the part of Service Level Agreement.

Then, the *Resource Broker* component of the federation platform checks the available resources (VM Instance types) at each of the member cloud providers. Based on the details captured from *Request Handler* and *Resource Broker*, it is now the job of the *Service Placement Plan Maker* component to decide on the most appropriate cloud resources for the placement of services of the requested application. For this, it evaluates the preference weight vector from the consumer stated preference for service placement through its *User Preference Evaluator* sub-component, as stated earlier. Next, it finds a set of Pareto optimal Placement Plan through *Placement Plan Optimizer* sub-component (details in section 3.3.4). And finally, it finds a single most appropriate placement plan out of the set of Pareto optimal plan as per the overall fitness, which is determined as a function of the objective functions and the preference weights. *Request Handler* component communicates about the identified plan to the customer and initiates requests for application deployment to selected providers. The SLA, which is a set of service level requirements mutually agreed upon by service

consumer and service provider (Breskovic, Altmann, & Brandic, 2013), is stored in the *SLA Repository* for future reference, and the service provisioning continues. The *Federation Business Logic* component performs *Accounting* of service provisioning related to all the requests and performs and provides business logic for *Revenue Sharing* among federation members. The *Application Monitoring* component monitors the application behavior including their footprints throughout the application lifecycle and initiates application service replacement decision if needed. For readers' convenience, a brief description of the function of each component of the federation platform is given in Table 4.

Table 4: Function description of each component in the federation platform for service placement

Component	**Role**
Request Handler	Captures the requirements of the application including the preferences of the consumer on the criteria for service placement and reaches the agreements on the service levels.
Accounting	Maintains the record of the service provisioning details with respect to each request to be used for financial settlements
Resource Broker	Identifies potential provider resources that can fulfill application requirements and triggers the Service

	Placement Plan Maker. It initiates the application deployment by sending a placement request to selected cloud providers.
SLA Repository	Maintains SLAs that have been agreed between the cloud federation and cloud provider. It also maintains the SLAs with the customers.
Monitoring	Collects information about application performance and footprint from across the federated clouds and feeds the results into accounting, SLA repository, and resource broker
User Preference Evaluator	Converts the consumer preference over different decision criteria that are stated as pairwise comparison into a preference weight vector by applying the AHP method (T. L. Saaty, 1990).
Placement Plan Optimizer	Identifies a set of Pareto optimal placement plans with multi-objective optimization process using NSGA II (Deb et al., 2002).
Service Placement Plan maker	With support from *User Preference Evaluator* and *Placement Plan Optimizer* subcomponents, decides on the most appropriate service placement plan. Details in section 3.3.2

3.3.2 Multi-criteria Model for Service Placement Decision Making

In this section, we present a description of various decision-making criteria selected for the model and outline a generic decision-making model with the workflow (flowchart) representing the decision process.

3.3.2.1 Decision Criteria for Service Placement

From the previous related works that have been discussed in section 3.2.3, we observe that there are some criteria that are considered by most of the research works related to VM or task placement in a cloud computing environment. Those criteria include financial cost, execution speed, network latency, availability, reliability, security, load balancing, and energy consumption (Table 3). Among them, optimization on factors like energy consumption and load balancing requires the decision maker to be able to choose the physical server within a cloud data center. For this reason, these factors make sense in a service placement decision for a single cloud or a single provider scenario, but are not applicable to a federation platform or a broker whose interest lies in finding the appropriate clouds, but do not have an interest in or control over the internal scheduling that involves the selection of a particular physical machine within a cloud data center. Hence we do not include these parameters for service placement decision criteria. Similarly, the security factor is a matter of subjective judgments, and the reliability factor is use-case dependent. Therefore, they are also excluded from the

proposed model. Following these arguments, our service placement decision model is based on the following factors (Table 5).

1. Financial Cost

Financial cost refers to the cost of service provisioning. There are various factors that determine the cost of service provisioning: hardware infrastructure cost, energy cost, and the administrative cost incurred by the cloud provider and the administrative cost of the federation (Altmann & Kashef, 2014). IaaS cloud providers offer services in the form of VM Instances, and they charge for the consumption at a specified price per hour (AWS, 2019). Cloud Providers set the VM Instance price by considering the entire costs (infrastructure, energy and other administrative costs) and the market situation. The administrative cost of the federation remains same irrespective of the chosen member cloud provider and, therefore, does not contribute to service placement decision. Thus, for the purpose of service placement decision making, the financial cost of a service placement plan is evaluated as the sum of the cost of all the VM instances involved in the service deployment plan. Minimization of financial cost is one of the objectives in the proposed model. It is expressed as:

$$Minimize\ f_{cst}(x) = \sum_{j=1}^{m} Cost(V_j);\ \forall\ V_j \in\ x$$

Where $f_{cst}(x)$ is the financial cost of placement plan x and V_j refers to one of the m VM instances included in the placement plan x for j^{th} VM node of the application.

2. CPU Speed

The CPU speed is an important factor determining the execution speed. An application provides a better response time when deployed on a VM Instance with faster CPU speed. A poorly responsive application has found to decrease user engagement and hence negatively affects the profitability of the application owner (cloud service customer). Various data centers host servers with different CPU speeds (Dell, 2019), which is aimed at maintaining a tradeoff between infrastructure cost and performance. Maximization of the CPU speeds is one of the objectives in service placement decision making.

$$Maximize\ f_{cspd}(x) = \sum_{j=1}^{m} \frac{CPU\_speed(V_j)}{m};\ \forall\ V_j \in\ x$$

Where $f_{cspd}(x)$ is the average of the CPU speed of the VM instances comprising placement plan x and V_j refers to one of the m VM instances included in the placement plan x corresponding to the j^{th} service node of the application.

Table 5: Criteria and optimization objectives

Criteria	Objective	Expression

Financial Cost	Minimization of financial cost	Minimize $f_{cst}(x) = \sum_{j=1}^{m} Cost(V_j)$
CPU Speed	Minimization of execution time with maximizing average processor speed	Maximize $f_{cspd}(x) = \sum_{j=1}^{m} \frac{CPU\_speed(V_j)}{m}$
Memory	Minimization of execution time by maximizing memory size	Maximize $f_{mem}(x) = \sum_{j=1}^{m} Mem\_Size(V_j)$
Network Latency	Minimize the average network latency experienced by application users	Minimize $f_{lat}(x) = (\sum_{p} RTT(V_{source}, V_{target}) + \frac{\sum_{i=1}^{l}\left(\sum_{j=1}^{m} RTT(POI_i, v_j)\right) * users_i}{\sum_{i=1}^{l} users_i})$
Availability	Maximize application availability by minimizing application downtime	Minimize $f_{avl}(x) = \prod_{i=1}^{k} Availability_i$

3. Memory

Another important factor determining the computing capability is the memory size. Operating systems use a technique called swapping for memory management. This technique allows operating systems to work with a large number of data files that require more memory than the physically available main memory. This is achieved by moving data between main memory and secondary

storage. The process of copying data from the secondary storage to main memory requires significant time (Tyson, 2000). Therefore, with large memory size, the number of memory swaps is reduced leading to a reduction in program execution time. Hence, maximization of the memory size is a desired objective function.

$$Maximize\ f_{mem}(x) = \sum_{j=1}^{m} Mem\_Size(V_j)$$

Where, $f_{mem}(x)\ is$ the total memory sizes as per placement plan x.

4. Network Latency

A study suggests that the acceptable waiting time for retrieving information is 2 seconds (Nah, 2004). Therefore, it is desired to have the application response time at a lower level. The response time of an application refers to the time it takes to react to a user request. Network latency, a delay in data communication over a network, is one of the major factors that contribute to application response time (Tse-Au & Morreale, 2000). In our model we consider two types of network latencies, namely - i) network latency between application nodes, and ii) network latency between user and application server. The latency is measured in Round-Trip-Time (RTT) (Obraczka & Silva, 2000), referring to the time taken by a data packet or a signal to travel from a source to a destination and back to the source.

Minimization of the average latency between a pair of nodes that hold data communication relationships will improve the application response time. Data dependency relationships are specified as node relationships within the application topology which forms a part of the SLA. The average latency is expressed as $\sum_p RTT(V_{source}, V_{target})$, where p is the number of node relationships.

Similarly, users experience a better application response time if the network latency between them and the application server is minimized. The average of the network latencies experienced by users in all specified points of interests (POI) is expressed as $\frac{\sum_{i=1}^{l}\left(\sum_{j=1}^{m} RTT(POI_i, v_j)\right) * users_i}{\sum_{i=1}^{l} users_i}$, where l is the number of POIs and m is the number of VM nodes. A Point of Interest (POI) represents a geographic location from where significant number of user requests for application service is originated, and hence the application owner is interested in minimizing the network latency experienced by users in these locations.

Following this discussion, the third decision criteria for service placement is to improve the application response time by minimizing network latencies (node to node latency and user to node latency). It is expressed as follows.

$$Minimize\ f_{lat}(x)$$

$$= \left(\sum_{p} RTT(V_{source}, V_{target})\right.$$

$$\left. + \frac{\sum_{i=1}^{l}\left(\sum_{j=1}^{m} RTT(POI_i, v_j)\right) * users_i}{\sum_{i=1}^{l} users_i}\right)$$

Where, $f_{lat}(x)\ is$ the average network latency as per placement plan x.

While considering average Round Trip Time (RTT) to measure and optimize on Network Latency, it is important to note that service providers, at present, are interested in preventing long tail latencies rather than reducing the average latency. The motivation comes from the fact that in a production environment, latency is a probability distribution. For example, at 75% percentile, the latency may be two times the average value, while at 99% percentile, it may be 100 times (Accela, 2016). In that case, 1% of the users of cloud service at the tail end experience intolerable delays. Hence Service providers are willing to minimize the delays experienced by these (for example, 1%) users at the tail end experiencing the worst delays. Optimization of service placement plan on network latency using tail latency becomes too complex due to the involvement of various factors contributing to tail latency. Hence, for simplification without much compromise in the desired objective, we consider average latency measured in terms of Round Trip Time (RTT) for the optimization of the service placement plan.

5. Availability

Availability is a non-functional requirement of a system that specifies the percentage of the time a system is accessible. In other words, it specifies the maximum permitted system downtime during a certain time period. In cloud computing, it is considered one of the SLA requirements and is a commitment made by the cloud service provider. Availability requirements depend on the nature of the application. However, other factors remaining constant, higher values are preferred at all times. Downtime of one data center deploying a service of the application may affect the availability of the whole application. Thus, it is desired to deploy application service nodes in such a way that it leads to the minimum system downtimes and maximum availability. For any non-replicated application, the application goes down if the application service component running on any of the clouds is unavailable. The event of one cloud data center being unavailable is independent of the other cloud data centers. Hence, following the probability theory for k independent events, the availability of the application is evaluated as the product of the availability probabilities of all cloud datacenters that are involved in the service placement plan. Thus, the objective function for maximization of system availability is expressed as -

$$Maximize\ f_{avl}(x) = \prod_{i=1}^{k} Availability_i\ ;\ \forall\ i \in\ x$$

where, $f_{avl}(x)$ is the overall availability of placement plan x, $Availability_i$ is the availability of cloud data center i and i varies for k data centers involved in the placement plan x.

3.3.2.2 The Decision Model

Service placement decisions are made based on how each customer puts preference over five decision criteria identified in section 3.3.2.1. The generic service placement decision model is given in **Figure 3**.

In the model, details on application topology and preferences over service placement decision criteria are taken from the customer when they initiate a service request. The details on resource availability are provided by cloud providers. It is important to note that the application footprint, which is the predicted number of users at various Points of Interests (POIs) during the initial deployment is taken from customer (as distinguished by dash lines), while their actual values are collected by monitoring component during the application lifecycle to be used to make any (re)placement plan according to the changing application footprint, when required.

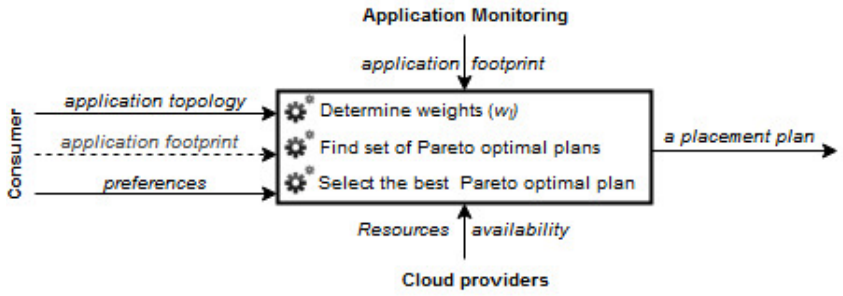


Figure 3: Generic decision model for service placement

Considering the service placement decision problem for an application that requires *m* different application nodes in a federated cloud possessing *n* different VM types from all providers leads to n^m number of possible placement plans. The search for the most optimal plan within this large number of potential plans is computationally infeasible for large *m* and *n*. Thus we propose a three-step procedure for the decision model as depicted in **Figure 3**. The details on each of these three steps are outlined in the swim lane chart in **Figure 4**.

The first step involves getting customer preferences along with other inputs, and evaluating weight vector for the preferences, and storing them into the SLA Repository. The method of determining weights by capturing consumer preferences is explained in section 3.3.3.

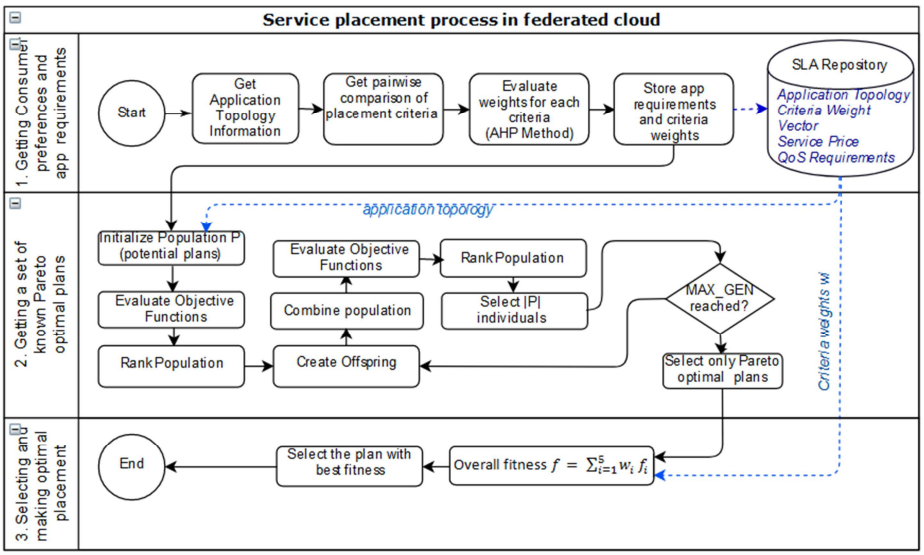


Figure 4: Swim lane diagram for the service placement decision model

The second step involves the reduction of the large search space of potential solutions into a set of few Pareto optimal solutions. Unlike single objective optimization problem, where a decision-maker could look for a single best solution that is a global maximum or minimum, identification of a single global best solution is impossible in the case of a multi-objective optimization problem. This is due to the presence of objective conflicts (Hans, 1988). Instead, there exist a number of non-dominated or Pareto-optimal solutions. A solution is called non-dominated if no other solution in the solution space is superior to it in any of the objectives without being inferior in the remaining objectives (Deb, 2014).

A classical approach such as Scalarization (Marler & Arora, 2004) avoids the objective conflict by combining many objectives into single objectives and allows reaching to a single solution. For this, with expert domain knowledge, weights are assigned to each objective and the overall objective is evaluated as the weighted sum of all the objectives. In such an approach, emphasis due to the weight of one the objectives has a significant effect on the optimization of other objectives. The obtained solution is usually Pareto-optimum (Deb et al., 2002); however, it is highly sensitive to the weights and limits the simultaneous optimization of multiple parameters.

Hence, to avoid the situation where the weight of one objective affects the optimization on other objectives, in the second step, we employ evolutionary algorithms to select a set of non-dominated (known Pareto optimal plans) by optimizing each of the decision criteria simultaneously. We present the method for reducing the search space and finding a set of non-dominated plans in section 3.3.4. And, the third step we perform the selection of the most appropriate Pareto optimal solution with the best fitness value, which is determined as the sum of the product of objective functions and their corresponding weights. Details on this step are described in section 3.3.6.

The proposed three-step decision model, thus, allows for i) making service placement decision as per the unique preference of individual consumers, ii) simultaneous optimization of all five criteria without having the influence of the optimization on one objective to the optimization on the other objectives, and iii) identify a single final solution, which is essential for application deployment without human intervention.

3.3.3 Capturing User Preferences over Decision Criteria and Determining their Weights

We employ the Analytic Hierarchy Process (AHP) (T. L. Saaty, 1990) method to select the most appropriate service placement plan from a set of Pareto optimal plans based on consumer preferences.

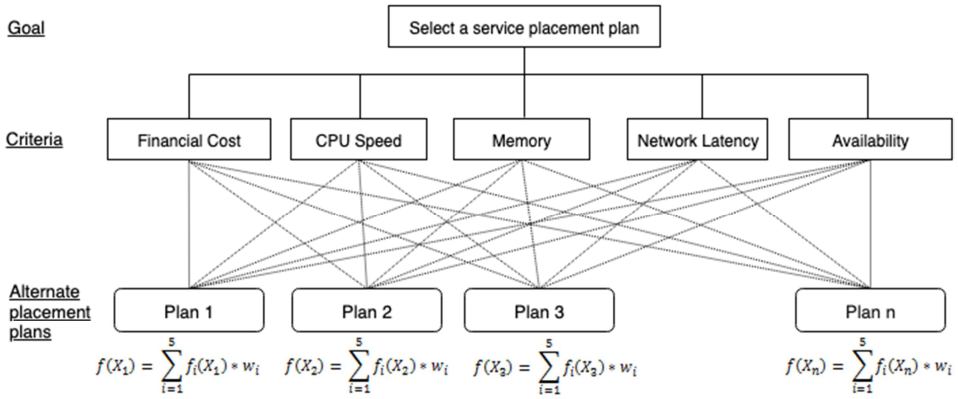


Figure 5: Decomposition of service placement decision problem into a hierarchy

AHP is a solution approach to multi-criteria decision-making problem by arranging the decision factors into the hierarchic

structure. **Figure 5** depicts the decomposition of service placement decision-making problem into a hierarchic structure according to the AHP method. The hierarchy contains three levels.

Below mentioned steps are undertaken for the selection of service placement plan following the AHP method (R. W. Saaty, 1987).

- First, the service placement decision problem is decomposed into a three-layer hierarchical structure that includes Goal at the top, Criteria in the middle and Alternatives at the bottom as shown in **Figure 5**. As shown on the top of the hierarchy, the goal is to select an optimal service placement plan. The decision criteria in the middle include *Financial Cost,* Average *CPU Speed, Memory Size,* Average *Network Latency,* and *Availability.* This criteria layer in the middle can be divided into various sub-layers to decompose any criteria into sub-criteria if the problem demands so, however, no criteria in this problem require decomposition. The alternatives at the bottom of the hierarchy contain a set of Pareto optimal service placement plans, which are identified by the optimization process. The optimization process is explained in detail in section 3.3.4.

- The customer is asked to make a pairwise comparison of decision criteria by assigning importance values based on the scale as shown in Table 6 and a corresponding comparison

matrix is built. And, since different criteria involve different units of measurement, we normalize the comparison matrix so that the value of each criterion lies in the range 0 to 1. Please refer to (Teknomo, 2006) for detail procedure.

Table 6: Scale for the importance intensity used for pairwise comparison of decision criteria

Importance (Intensity)	**Meaning**
9	Absolutely more important
8	An intermediate value between very strongly and absolutely more important
7	Very Strongly more important
6	An intermediate value between strongly and very strongly more important
5	Strongly more important
4	An intermediate value between weakly and strongly more important
3	Weakly more important
2	An intermediate value between equally and weakly more important
1	Equally important

- From the comparison matrix, the weight vector that represents the customer assigned importance of each criterion is evaluated by using the approximation method. The decision problem in our case requires only weights to be evaluated and does not require ranking. Hence, the approximation method is employed as the approximation

method offers simplicity in calculations without loss of accuracy in problems that require only weights but not a ranking of criteria (R. W. Saaty, 1987). This process gives the weight vector $w = (w_{cst}, w_{cspd}, w_{mem}, w_{lat}, w_{avl})$ representing customer assigned weights for *Financial cost, CPU speed, Memory Size, Network Latency,* and *Availability* respectively. Please refer to (Teknomo, 2006) for detail procedure.

3.3.4 Finding a Set of Known Pareto Optimal Placement Plans

In order to select the service placement plan based on consumer preference using AHP procedure, it is essential to reduce the search space of potential service placement plans (hereafter referred to as a solution to match the general term in evolutionary optimization problem). We do so based on five decision criteria identified in section 3.2.1, namely - Financial Cost, CPU Speed, Memory, Network Latency, and Availability. To simultaneously optimize on each of the criteria for our service placement decision problem, we formulate and solve the problem as a multi-objective optimization problem. It is expressed as:

$$Minimize\ f(x) = \left(f_{cst}(x), f_{cspd}(x), f_{mem}(x), f_{lat}(x), f_{avl}(x)\right)^T$$

$$subject\ to\ x \in X,$$

Where,

$x = (x_1, x_2, \dots, x_n)$ *is a decision vector consisting of* n *decision variables. It represents a service placement plan, where a decision variable* x_i *represents selected VM Instance for the* i^{th} *node of the application.*

$f(x) : R^n \rightarrow R^4$, *objective vector*

$f_{cst}(x) : R^n \rightarrow R$, *objective function (Financial Cost)*

$f_{cspd}(x) : R^n \rightarrow R$, *objective function (CPU Speed)*

$f_{mem}(x) : R^n \rightarrow R$, *objective function (Memory)*

$f_{lat}(x) : R^n \rightarrow R$, *objective function (Network Latency)*

$f_{avl}(x) : R^n \rightarrow R$, *objective function (Availability)*

X, *a feasible set of decision vectors (service placement plans)*

Feasible set of placement plan is determined by constraints such as technical specification (e.g. a number of CPU cores, memory size, and storage size) of the selected VM instances for each application nodes, data center location. The constraints are expressed as follows.

$Region\ (x_i) \in \{region_1, region_2, \dots, region_n\}, \quad \forall\ x_i$

$CPUCount(x_i) >= MINIMUM\_CPU(x_i), \ \forall\ x_i$

$MemorySize(x_i) >= MINIMUM\_MEMORY(x_i), \ \forall\ x_i$

Where,

$Region\ (x_i)$, *is the location of the datacenter chosen for VM node* i

$\{region_1, region_2, \dots, region_n\}$, *is a vector of customer preferred cloud locations for application deployment*

$CPUCount(x_i)$, *is the number of CPU cores offered by the VM instance selected for node i*

$MINIMUM_{CPU}(x_i)$, *is the minimum number of CPU cores required for node i, which is specified in application requirement as part of node topology*

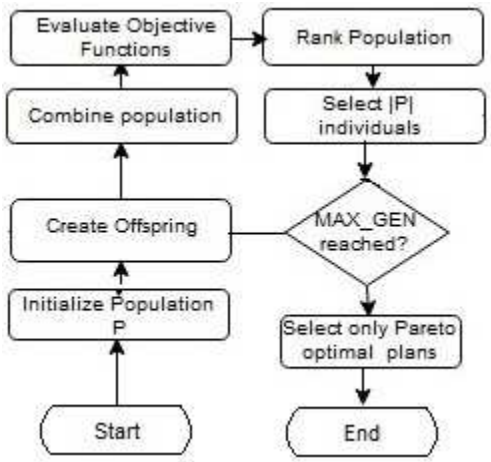
$MemorySize(x_i)$, *is the memory size offered by the VM instance selected for node i*

$MINIMUM_{MEMORY}(x_i)$, *is the minimum memory size required for node i, which is specified in application requirement as part of node topology*

The objectives considered in the model conflict with each other, for instance, a higher degree of availability incurs a higher cost, too. Due to the conflict, a single best multi-objective solution that is optimized on each objective functions simultaneously is near to impossible. A practical approach to solving such multi-objective problem would be to start by finding solutions that are Pareto-optimal (Fonseca & Fleming, 1995). A Pareto-optimal solution is one for which none of the objectives can be improved without degrading at least one of the others (Zitzler et al., 2000). Depending on the number of design variables, there could be an enormous number of Pareto-optimal solutions, and their identification is, thus, not computationally feasible or practical. It is also computationally infeasible to prove the optimality of the solution sets (Konak, Coit, & Smith, 2006). Hence, a reasonable approach is to find a set of best known Pareto-optimal solutions within the feasible region.

```
1  Main ()
2     i = 0
3     P_i = generatePopulation()
4     Q_i = performGeneticOperation(P_i)
5     R_i = P_i ∪ Q_i
6     WHILE (i < MAX_GENERATION)
7        F_i = performNonDominatedSorting (R_i)
8        j = 1
9        WHILE (F_i,j in F_i)
10             assignCrowdingDistance(F_i,j)
11             j = j + 1
12       END WHILE
13       P_i+1 = ∅
14       j = 1
15       WHILE (|P_i+1| < |P_i|)
16             F_i,j = sort(F_i,j)
17             P_i+1 = P_i+1 ∪ F_i,j[1: |P_i| − |P_i+1|]
18             j = j + 1
19       END WHILE
20       Q_i+1 = performGeneticOperation(P_i+1)
21       R_i+1 = P_i+1 ∪ Q_i+1
22       i = i + 1
23    END WHILE
24    Pareto_optimal_placement_plans = F_i,1
```

(a)Algorithm (Pseudo-code)



(b)Flowchart

Figure 6: Process to find a set of Pareto optimal Service Placement Plans

Various evolutionary algorithms have been suggested for such problems. The best known Pareto-optimal solution set should ideally be a subset of the Pareto-optimal set or should be as close to the Pareto-optimal set as possible; and, be evenly distributed over the whole spectrum of the Pareto-optimal front (Zitzler et al., 2000).

We employ Elitist Non-dominated Sorting Genetic Algorithm II to find a set of Pareto optimal solutions as it allows for true optimization of multiple objectives by maintaining the diversity in the solution while at the same time implicitly preserving the elitism properties (Deb et al., 2002). Following (Deb et al., 2002), the process of finding a set of Pareto optimal service placement plans through the optimization process is given in **Figure 6**.

The optimization process starts with the generation of the population comprising of different individuals (Figure 6(a): Line 3). The individuals are also known as solutions, representing potential service placement plans and hence, in this section, individual, solution and placement plan are used interchangeably. Section 3.5 provides more details on solution design, which is the process of creating individual chromosome that represents a solution or a service placement plan. Next, the population of this first generation undergoes genetic operations that include selection, crossover, and mutation to generate the next set of the population known as

offspring (Figure 6(a): Line 4). Details on the genetic operation are provided in Appendix 1.

Both the populations (initial population and offspring) are merged to make a combined pool of solutions (Figure 6(a): Line 5). The solutions in the combined pool are, then, ranked by use of non-dominated sorting method and Crowding Distance Operator. To do so, the solutions are, first, sorted and grouped into various Pareto Fronts. Sorting and grouping into different pareto fronts depend on their fitness with respect to five objective functions, namely - Financial Cost, CPU Speed, Memory, Network Latency, and Availability. Detail description of sorting is provided in Appendix 1. Solutions in the first Pareto Front are non-dominated solutions, and hence are better than those in the second Pareto Front and so on and hence ranked higher. Since a Pareto Front may have more than one solutions, the ranking of the solutions within a particular Pareto Front, however, is determined by use of Crowding Distance. Crowding Distance is a measure of the density of the solutions at the neighborhood of the solution (Deb et al., 2002). Solutions with higher values of crowding distances are selected to maintain the diversity of the solutions. And hence, within a particular Pareto Front, solutions with higher Crowding Distance values are ranked higher.

Next, crowding distance is evaluated for all solutions in each of the Pareto fronts and assigned to them (Figure 6(a): Line 10). Details on the assignment of the crowding distance to solutions of a front are given in Appendix 1. Once all the solutions in the combined population are ranked using the non-dominated sorting and crowding distance metrics, only top N solutions (|P|) are selected as a new population based on their rank determined by their Pareto Front and assigned Crowding distance (Figure 6(a): Lines 15 through 19). This new population undergoes a genetic operation (Figure 6(a): Line 20). The two populations are combined (Figure 6(a): Line 21). The overall process is repeated until the termination condition is reached (Figure 6(a): Line 6). On the termination of the evolution process, we remain with N number of solutions from which only the Pareto optimal ones are selected for the further decision-making process (Figure 6(a): Line 24).

3.3.5 Solution Design (Population Generation)

Figure 7 shows an example of the formulation of the solution variable with the mapping of the application service nodes to VM instances (provider resources). There are altogether 10 VM types (resources or VM instances) from different providers. The application that is to be deployed in federated cloud resources requires seven VM nodes each with different configurations.

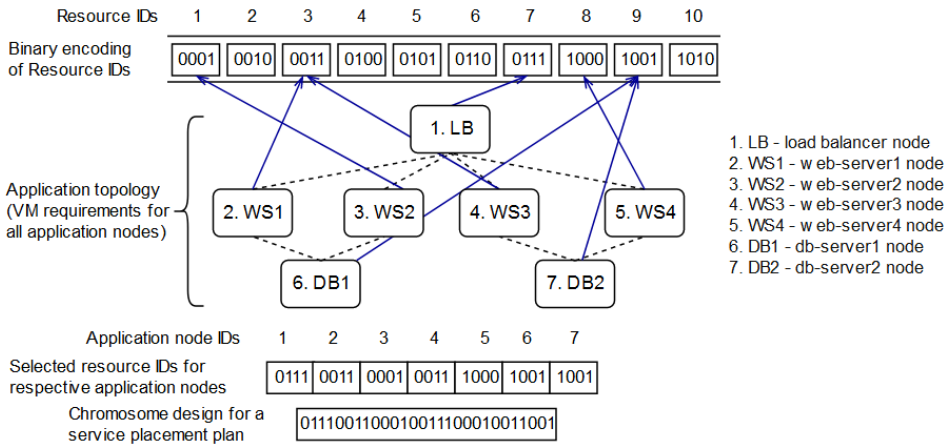


Figure 7: Solution design (generation of a population of solutions)

Since we are employing a Genetic Algorithm, solutions are represented in the form of a chromosome. Here, our objective is to specify the structure and formation of the chromosomes. To design the chromosomes, we make an index of all the available VM instance types from across the federation and assign them their index number as unique IDs representing them. We use the binary conversions of these IDs to represent them during the optimization process. Binary representations are padded with the required number of '0's, to make the binary string of uniform length. To generate a potential placement plan, for each node of the customer application, a VM instance type from among the ones satisfying the minimum configuration requirements for that node is chosen at random. The binary strings of the IDs representing the selected VM instances for all the nodes are concatenated in a particular sequence to form a chromosome that represents a potential placement plan,

known as a solution or an individual of the population. This process is repeated to generate a n number of chromosomes or individuals for a population of size n.

3.3.6 Selection of a Single Optimal Plan from the Identified Known Pareto Optimal Set

After the optimization that is performed as described in Section 3.3.4, we have a set of known pareto optimal placement plans or solutions. Presence of multiple criteria makes a comparison of the solution in the objective space impossible; hence, we convert multiple objective functions into a single objective function, hereafter known as fitness function, by means of the Scalarization (Marler & Arora, 2004) using the weight vector w, which is identified through the process as described in Section 3.3. This gives a single fitness value representing the overall fitness of the solution by taking into account each individual fitness values for Financial Cost (f_{cst}), CPU Speed (f_{cspd}), Memory (f_{mem}), Network Latency (f_{lat}), and Availability (f_{avl}).

Here, objective functions f_{cst} and f_{lat} are of minimization while f_{cspd}, f_{mem}, and f_{avl} are of maximization type. We convert the objective functions of maximization types (i.e.- f_{cspd}, f_{mem}, and f_{avl}) to minimization type by multiplying the function with a negative one to make the overall objective vector $f(x)$ of objective minimization type. Since the units of the objective functions differ,

we use the normalized functions to make each of their values fall within the range of 0 to 1.

The overall fitness of the solution $f_o(x)$ is evaluated as -

$$f_o(x) = w_{cst} * f_{cst}^{norm}(x) - w_{cspd} * f_{cspd}^{norm}(x) - w_{mem} * f_{mem}^{norm}(x) + w_{lat} * f_{lat}^{norm}(x) - w_{avl} * f_{avl}^{norm}(x)$$

Where,

x , $a\ placement\ plan$

$f_o(x)$, $overall\ fitness\ value\ of\ placement\ plan\ x$

w_{cst} , $preference\ weight\ (Financial\ Cost)$

w_{cspd}, $preference\ weight\ (CPU\ Speed)$

w_{mem}, $preference\ weight\ (Memory)$

w_{lat} , $preference\ weight\ (Network\ Latency)$

w_{avl}, $preference\ weight\ (Availability)$

$f_{cst}^{norm}(x)$, $normalized\ objective\ function(Financial\ Cost)$

$f_{cspd}^{norm}(x)$, $normalized\ objective\ function(CPU\ Speed)$

$f_{mem}^{norm}(x)$, $normalized\ objective\ function(Memory)$

$f_{lat}^{norm}(x)$, $normalized\ objective\ function(Network\ Latency)$

$f_{avl}^{norm}(x)$, $normalized\ objective\ function(Availability)$

The expressions for the objective functions with their normal forms are shown in Table 7.

Once the overall fitness of each of the solutions is calculated; the one with the minimum fitness value representing the most preferred

Pareto optimal placement plan is selected as the chosen placement plan.

Table 7: Expressions for objective functions and their normal forms

Criteria	**Objective Functions (Expressions)**	**Normalized Objective Functions**
Financial Cost	$f_{cst}(x) = \sum_{j=1}^{m} Cost(V_j)$	$f_{cst}^{norm}(x) = \frac{f_{cst}(x) - \min(f_{cst})}{\max(f_{cst}) - \min(f_{cst})}$
CPU Speed	$f_{cspd}(x) = \sum_{j=1}^{m} \frac{CPUSpeed(V_j)}{m}$	$f_{cspd}^{norm}(x) = \frac{f_{cspd}(x) - \min(f_{cspd})}{\max(f_{cspd}) - \min(f_{cspd})}$
Memory	$f_{mem}(x) = \sum_{j=1}^{m} MemorySize(V_j)$	$f_{mem}^{norm}(x) = \frac{f_{mem}(x) - \min(f_{mem})}{\max(f_{mem}) - \min(f_{mem})}$
Network Latency	$f_{lat}(x) = (\sum_{p} RTT(V_{source}, V_{target}) + \frac{\sum_{i=1}^{l}(\sum_{j=1}^{m} RTT(POI_i, V_j)) * users}{\sum_{i=1}^{l} users_i}$	$f_{lat}^{norm}(x) = \frac{f_{lat}(x) - \min(f_{lat})}{\max(f_{lat}) - \min(f_{lat})}$
Availability	$f_{avl}(x) = \prod_{i=1}^{k} Availability_i$	$f_{avl}^{norm}(x) = \frac{f_{avl}(x) - \min(f_{avl})}{\max(f_{avl}) - \min(f_{avl})}$

3.4 Simulation

We evaluate the effectiveness of the proposed model and the algorithm by performing an extensive simulation covering a wide range of scenarios. For this, we implemented the proposed model

and the algorithm in a computer program written using Python programming language. In this section, we provide a description of the scenario and settings of the parameters employed for the simulation.

3.4.1 Simulation Scenario

For the simulation purpose, we consider a request from an application provider (customer) for deploying its application in the federated cloud. The service placement request is for a typical web application with multiple services, the topology of which is shown in **Figure 8**.

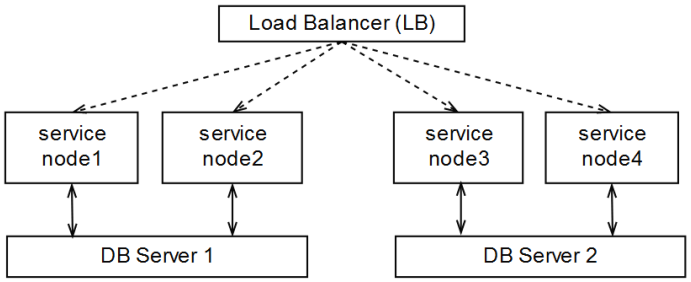


Figure 8: Application topology considered for simulation

As shown, the application comprises of seven nodes. The configuration of the application nodes is described in section 3.4.2. The bi-directional arrows represent a significant data communication requirement between the nodes involved.

We also consider the cloud federation consisting of six cloud providers each having their cloud data centers in two of the five regions, which makes up for 12 clouds in total (**Figure 9**).

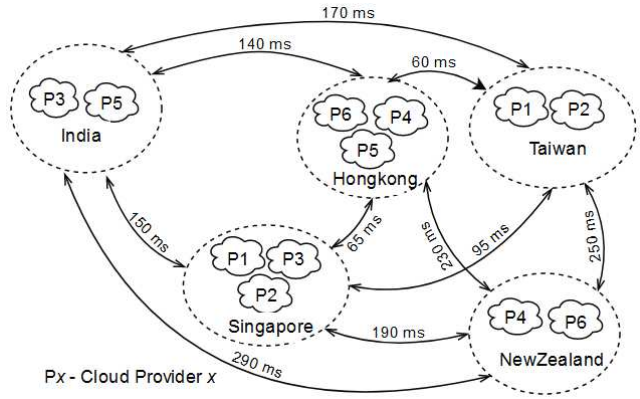


Figure 9: A federation of clouds with network latencies as Round Trip Time (RTT)

These data centers offr a commitment for its availability at various levels. The values chosen for the simulation purpose are given in section 3.4.2. We assume that the availabilities of all the VM types offered from the same data center are identical.

The communication latencies as Round Trip Time (RTT) between clouds in different regions are depicted as labels of the bidirectional arrows in Figure 9, and the source for the latency figures are explained in section 3.4.2. We consider the application footprints spread in five regions, referred to as Points of Interests (POIs), which are the locations from where the majority of user requests to the application are supposed to be originated. For simplicity, we consider the POIs to be the same regions that are considered for cloud locations. Now, the task is to find the optimal plan for the placement of services by selecting optimal resources (VM types) from one or more of these clouds (data centers). As the selection should be guided by consumer preferences, we assume that the

application provider states it's preferences over the decision criteria, for which the customer is provided with pairwise comparison tool in the web form integrated into the website of each provider.

Based on the consumer preference, the Service Placement Algorithm makes service placement plan by simultaneously optimizing on multiple criteria as listed in **Table 5**. We assume that the customers state their application requirement as per Topology and Orchestration Specification for Cloud Applications (TOSCA) standard (OASIS, 2017). As per the standard, the application requirement is stated in the form of *node-topology* and *relationship-topology*. The *node-topology*, for each application service node, specifies the detail configurations such as the number of CPU cores, memory size, storage size, operating system and so on. And, the *relationship-topology* lists the pair of application service nodes that require data communication. This provision allows customers or application owners to explicitly state the memory size required for each node. In that case, the memory size is fixed, and hence optimization on memory size becomes less significant. Here, it is important to state that there will, still, be room for optimization on memory if we considered customer stated requirement as a minimum rather than absolute. For simplifying the simulation, however, we exclude this criterion from the optimization without significantly compromising on the main objective.

To analyze how the proposed Service Placement Algorithm performs with service placement according to different consumer preferences, we run the simulations for a wide set of preference vectors. For this, different preference vectors representing different user preferences are derived from pairwise comparisons of decision criteria. Then, the simulation is performed by employing the parameter settings as described in section 3.4.2.

3.4.2 Parameter Setting

To show the federation, we consider 12 clouds distributed over 5 regions. This number can represent a cloud federation of moderate size. In each of the POIs, the number of users is selected at random within the range of values between 250 and 7 million with an assumption that such a wide range can represent the user requests originating from a region for an application such as an e-commerce application.

The topology of consumer application is considered to have 7 nodes with data communication relationships between 4 node-pairs. The minimum requirement for the number of CPU cores for the nodes set with carefully selected value in the range between 1 and 4. The minimum memory requirements for the nodes are set with a careful selection of values in the range 2 to 32 GB. Such an application topology and configuration is based on the assumption (made after discussion among colleagues with relevant experience) for the

requirement of a typical web application with few services and a moderate number of users.

Table 8: Parameter settings for the simulation

Parameters	Values	Basis
Providers and end users		
Number of clouds	12	Federation of moderate size
Number of users in each POIs	[250, 7 Million]	Wide enough range
Application Topology		
Number of application nodes	7	Assumption (based on discussion among colleagues) for a typical web application with few services and a moderate number of users
Number of Node-pairs with data communication requirements	4	
CPU cores	[2, 8]	
Memory Size	[2GB, 32GB]	
Network Latency - Round Trip Time		
Intra-cloud	0 ms	Same data center
Inter-cloud (same region)	[30, 45] ms	(Verizon, 2018)
Inter-cloud (different regions)	[60, 290] ms	(Verizon, 2018)
Specifications of provider resources (VM Instances)		

Number of CPU cores	[1, 128]	Amazon (AWS, 2019)
Memory Size	[0.5, 976] GB	Amazon (AWS, 2019)
CPU Speed	[1.67, 4.73] GHz	DELL (Dell, 2019)
Per unit price	[0.00518, 1999.76] $/hr	Amazon (AWS, 2019)
Availability of hosting Data Center	[97, 100] %	Gartner (Gartner Inc. CloudHarmony, 2018)

Parameter settings for the inter-cloud network latency are based on the average value of the yearly statistics (Nov 2017 - Oct 2018) published by Verizon (Verizon, 2018). Inter-cloud network latency within a region is set with randomly selected values in the range 30 to 45 millisecond, based on the statistics from Verizon (Verizon, 2018), as a guideline. And, the inter-cloud network latency between different regions is set with exact values from the source(Verizon, 2018), which lies in the range of 60 to 290 milliseconds for the regions selected.

Similarly, for the provider resources, the specification of VM types is set on the basis of the Amazon EC2 Pricing (AWS, 2019). Pricing is based on the Amazon EC2 Pricing for Seoul-Korea for on-demand instances. To make the pricing variation among providers and regions, we set the per unit price as a selected value in the range of 92.5% to 107.5% of the base price. Here, the base price refers to

the price for the instance type in the source mentioned (AWS, 2019). A value in this range is set as determined by the availability and CPU speed of the data centers to reflect the availability level and CPU speed in the price of the VM Instances.

The availability values for the provider resources, i.e.- cloud data centers are set with randomly selected values within the range of 97% to 100% which are the minimum and maximum availability offered by various cloud data centers as published by Gartner (Gartner Inc. CloudHarmony, 2018). The CPU speed for the VM Instance is based on Dell PowerEdge Server specifications (Dell, 2019).

3.5 Result Analysis

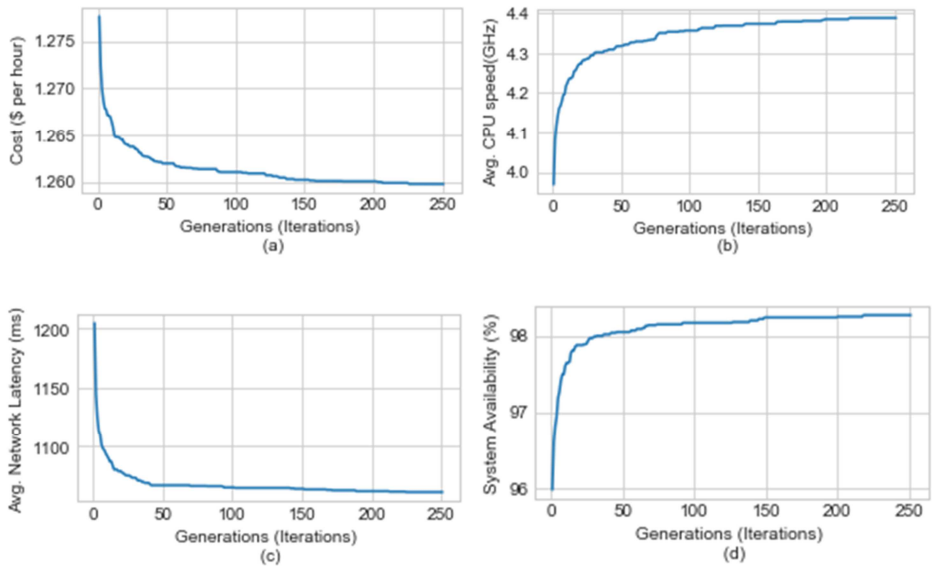
The results of the simulation are recorded by running the simulation program a number of times covering a wide range of scenario with respect to consumer preferences. Unless otherwise stated, all the results represent a mean value drawn from the results of 100 simulation runs. In this section, we present an analysis of the simulation results.

3.5.1 Does it Solve the Problem?

3.5.1.1 Convergence of Solutions with respect to the Objective Functions

To demonstrate the convergence process for various objective functions as the *set of placement plans* (hereafter referred to as a *population of solutions*), evolve through multiple generations, we performed the simulation with various consumer preference vectors such that one objective function (at a time) considered being '*Absolutely more important*' compared to all other objectives in pairwise comparison, while keeping the preference over objective functions in remaining pairs as '*equally important'*. And, we repeat the process for all the other objectives. The results are shown in Figure 10.

From the result, we observe a drastic convergence within the first few generations and then gradual convergence towards the final solution in case of all the objective functions. For instance, the criterion 'COST' is an objective minimization type. The best solution with respect to '*cost*' function in the initial *set of placement plans* (hereafter referred to as a *population of solutions*) incurs $1.278/hour (see Figure 10 (a)). Within the first 12 generations, the population of solutions evolves significantly, it takes a gradual pace thereafter and saturates after 225 generations at a value of $1.26/hour.



(Note: Graphs are not zero-based)

Figure 10: Convergence of objective functions as the population of placement plans evolve through multiple generations

Unlike 'COST' function, which is of objective minimization type, the 'Average CPU Speed' function is of objective maximization type. During the evolution process, the solutions with higher values are selected. The evolution starts with the population with the best solution providing average CPU speed of 3.97 GHz (Figure 10(b)). It evolves rapidly until the first 20 generations and then shows a gradual increasing process until it saturates in 217 generations providing the best solution with 4.39 GHz of average CPU speed.

The Average Network Latency (Figure 10(c)) evolves through by starting at 1204ms in the initial population and makes a rapid decrease until 14th generation decreasing it to 1095ms. Thereafter, it

follows a gradual decreasing process until coming to saturation after 190th generation at a value of 1061ms.

If we observe the curve for System Availability (Figure 10(d)), we see that it starts with an initial value of 95.98% in the first generation, increases rapid until 17th generation, and then gradually increases until 220 generations and saturates with the best solution in the population providing 98.27% of overall system availability, resulting in a gain of 2.4% in overall system availability.

In summary, to the extent of the performed simulation, the result shows that for any objective functions, the evolution process seems to saturate after passing through a certain number of generations. This number depends on the number of choices available for the criterion in the decision space. The result shows that each of the considered objective functions seems to be saturating within 225 iterations at maximum, which is an acceptable number of iteration. The simulation results show that, at least within the limits of the simulation scenarios considered, the algorithm can perform well in terms of stability and its' convergence to the final solution through the evolution process.

3.5.1.2 Meeting Different Customer Needs with Placement Service Optimized at Different Tradeoff Points

The objective of the service placement algorithm is to make service placement according to unique customer requirements as specified

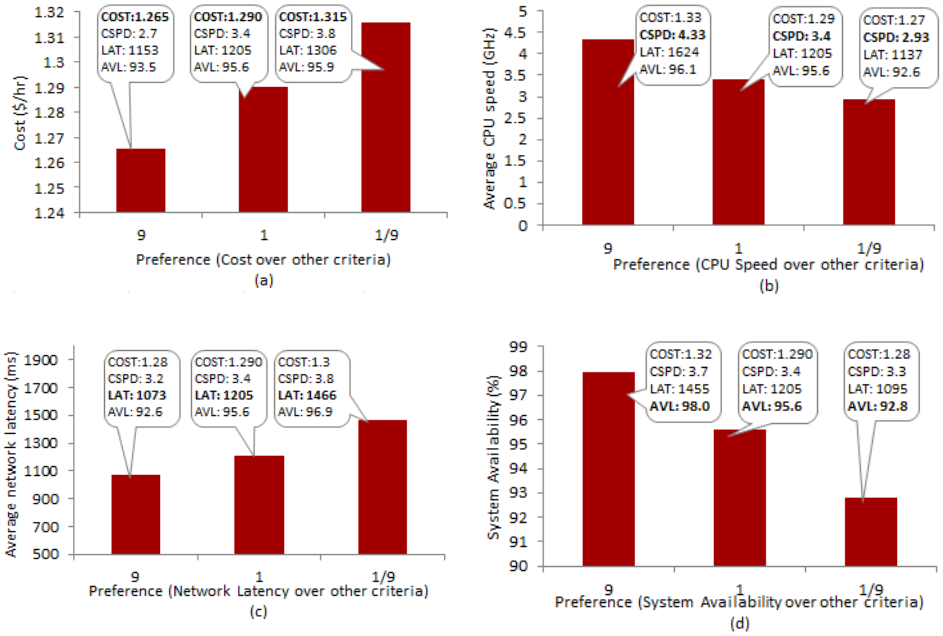
by their stated preferences. To see how different consumer preference is satisfied by the service placement plans, we demonstrate how it is possible to have different placement plans, which perform differently in the objective space, with different tradeoff points determined by the consumer preferences. Having four decision criteria yields six pair-wise comparisons of the criteria, and the fact that each pairwise comparison can take any of the 17 values makes possible for millions of tradeoff points.

Table 9: Weight Vectors with different preferences for 'COST', as an example, over other decision variables

Preference of *cost* over other variables		**Weight Vector (Evaluated through AHP method)**			
Value	***Meaning***	W_{COST}	W_{CPU_SPEED}	$W_{LATENCY}$	$W_{AVAILABILITY}$
9	Absolutely more important	0.7500	0.0833	0.0833	0.0833
1	Equally important	0.2500	0.2500	0.2500	0.2500
1/9	Absolutely less important	0.0357	0.3214	0.3214	0.3214

For simplicity, we demonstrate the results only for selected tradeoff points that represent the two extreme points and a mid-point in the preference comparison bar. For each of the objectives, these three tradeoff points are determined by the pairwise comparisons with following conditions i) the chosen objective is '9 - Absolutely more important' than the other ones, ii) the chosen objective is '1 -

Equally important' as the other ones, and iii) the chosen objective is '1/9 - Absolutely less important' than the other ones.



(Note: Not all graphs are zero-based)

Note: descriptions of criteria

COST: Cost of the service placement plan ($/hr)

CSPD: Average of the CPU Speeds of VM Instances chosen (GHz)

LAT: Average of network latency experienced by users (ms)

AVL: Availability of overall system/consumer application (%)

Note: descriptions of preferences

9: Absolutely more important

1: Equally important

1/9: Absolutely less important

Figure 11: Change in the values of objective functions with a change in the preferences

The preference weight vectors that define the corresponding tradeoff points representing the stated consumer preference as described above are, then, determined by the application of the Analytic Hierarchy Process (AHP) method. As a reference, the tradeoff points or the preference weight vectors, thus, evaluated are given in Table 9, in case of 'Cost' as the chosen objective. The preference weight vectors, in case of other objectives, as the chosen ones follow the same pattern and, hence, are omitted in the table.

The values in the objective space of the placement plan selected by the algorithm at the tradeoff points mentioned above are shown in **Figure 11**. It shows how consumers with different preferences can be served with differently optimized service placement plan.

In **Figure 11** (a), we observe that the optimally selected placement plan costs $1.29/hr. if *Cost* is considered equally important as all the other criteria. The *CPU speed, network latency* and *availability* for this placement plan are 3.4GHz, 1205ms, and 95.6% respectively. If the consumer states that *Cost* is considered absolutely important compared to all other criteria, then the cost of optimally selected placement plan is lowered to $1.265/hour. However, it is at the cost of CPU Speed which reduces to 2.7GHz, and Availability which reduces to 93.5%. If the consumer further states that *Cost* is absolutely less important compared to all the other criteria, the selected placement plan provides better CPU

Speed (3.8GHz) and Availability (95.9%) while the cost increases to $1.315/hour. We also observe that the network latency does not necessarily increase or decrease with a decrease or increase in the cost of the selected placement plan. This is because, unlike CPU Speed and Availability, the network Latency is not dependent on cost; rather, it is dependent on the application topology, location of selected VM Instances and location of application users. So, we see that it is possible to have different service placement plans with different cost values ranging between $1.265 per hour to $1.315 per hour, with each one being one of the known pareto optimal plan.

From **Figure 11** (b), it is seen that the optimally selected placement plan offers an average of 3.54GHz of CPU Speed if it is considered equally important as all the other criteria. The *Financial Cost, CPU Speed,* and *availability* for this placement plan are $1.29/hour, 3.3GHz, and 95.3% respectively. If the consumer states its preference such that *Network Latency* is considered absolutely more important compared to all other criteria, then the CPU Speed of optimally selected placement plan is increased to 4.33GHz. However, they should pay the price for Financial Cost of $1.33 per hour and Network Latency of 1624ms, while gains in availability (96.1%). If the consumer considers *CPU Speed* to be absolutely less important compared to all the other criteria, the selected placement plan costs low ($1.27/hour) while the average *CPU Speed* is reduced to 2.93GHz. Although it is observed that both Network

Latency and Availability seem to increase or decrease with an increase or decrease in CPU Speed, it is not due to their dependency relationships. Rather, Network Latency is decided by the characteristics of application topology, selected VM instance location and location of the majority of users, as said earlier; and, *Availability* is the characteristics of clouds hosting the selected VM Instances, which has an impact on *Financial Cost* but not in *CPU Speed*. The CPU Speed and Availability are not directly correlated; however, faster CPU Speed involves more Financial Cost, and clouds setting a higher price of VM instances generally offer better Availability. In this case, too, we see that it is possible to have different service placement plans with different CPU Speeds ranging between 2.93GHz to 4.33GHz, with each one being one of the known pareto optimal plan.

From **Figure 11** (c), we see that the optimally selected placement plan offers 1197ms of average *Network Latency* to users if it is considered equally important as all the other criteria. The *Financial Cost, network latency* and *availability* for this placement plan are $1.29/hour, 1209ms, and 95.4% respectively. If the consumer states its preference such that *Network Latency* is considered absolutely more important compared to all other criteria, then the selected placement plan provides a reduction in network latency (1073ms) and reduction in cost ($1.28/hour); however should pay the price with reduced *CPU Speed* (3.2GHz) and reduced *Availability*

(92.6GHz). If the consumer considers *Network Latency* to be absolutely less important compared to all the other criteria, the selected placement plan increases network latency (1466ms), still costs higher ($1.3/hour); but, results on faster average CPU Speed (3.8GHz) and availability (96.9%). Here too, we see that it is possible to have different service placement plans offering different Network Latencies ranging between 1073ms to 1466ms, with each one being one of the known pareto optimal plan.

From **Figure 11** (d), we observe that the optimally selected placement plan offers 94.8% of System Availability if Availability is considered equally important to all the other criteria. The *Financial Cost, CPU Speed, and network latency* for this placement plan are $1.29/hour, 3.4GHz, and 1147ms respectively. If the consumer states its preference such that *Availability* is absolutely more important compared to all other criteria, then the selected placement plan increases availability (98%), and at the same time provides faster average CPU speed (3.7GHz); however, it requires paying for increase in *Financial Cost* ($1.32/hour) and *Network Latency* (1455ms). If the consumer considers *Availability* to be absolutely less important compared to all the other criteria, the selected placement plan decreases Availability, and also reduces average CPU Speed; however, it offers an advantage with decreased cost (1.28/hour) and reduced Network Latency (1095ms). This way, we see that it is possible to have different service placement

plans offering different Network Latencies ranging between 1073ms to 1466ms, with each one being one of the known pareto optimal plan.

In conclusion, the proposed algorithm provides the flexibility to incorporate preferences of individual consumers for the multi-objective optimization and provide a guide in deciding the optimal service placement plan addressing the need of individual consumer (applications).

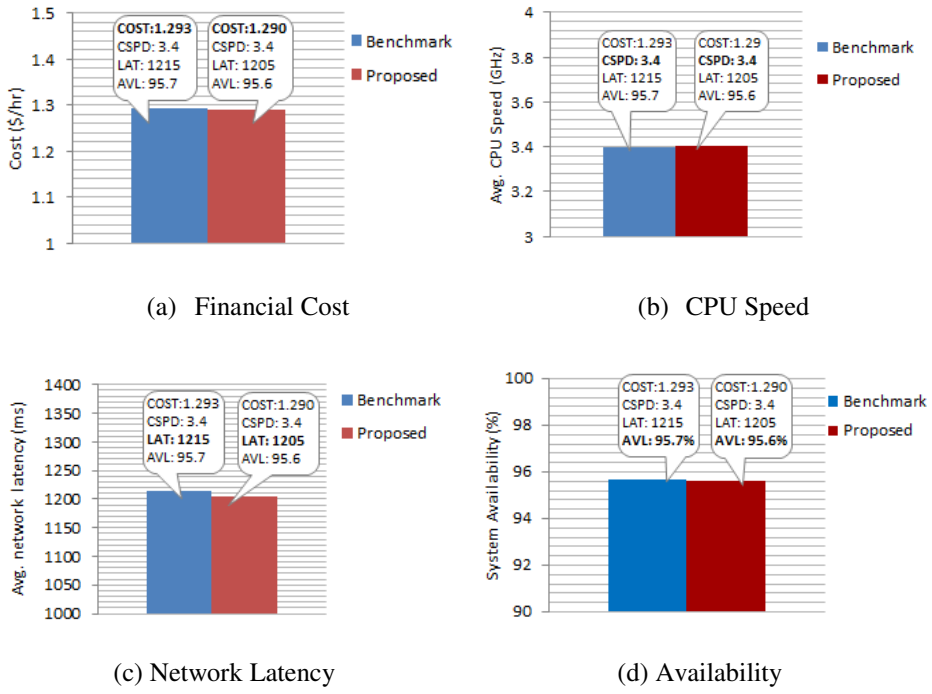
3.5.3 How Does it Perform Compared to the Benchmark Approach?

Now, in this section, we perform an analysis of how our proposed algorithm performs in comparison to the benchmark approach, a weighted sum approach as proposed in (Coutinho et al., 2015), where the Multi-objective optimization problem is solved by converting it to single objective optimization with weighted sum of multiple criteria. First, we show how the placement plans selected by two algorithms perform in the objective space and, second, we show how the comparison of the algorithms in terms of standard comparison metrics.

3.5.3.1 Comparison of Solutions in the Objective Space

We evaluated on how our algorithm performed in comparison to the benchmark approach, i.e.-evolutionary multi-objective optimization approach following weighted average method, as proposed in

(Coutinho et al., 2015), where multiple objectives are converted into single objective with a linear summation of multiple objectives. We performed the comparison with a tradeoff representing a neutral point. For this, we considered the same weight vector (0.25, 0.25, 0.25, 0.25) for both the algorithms to make the comparison on equal footage. The results of the comparison of the objective space are shown in Figure 12.



(a) Financial Cost

(b) CPU Speed

(c) Network Latency

(d) Availability

(Note: Graphs are not zero-based)

Figure 12: Comparison of solutions from proposed and benchmark approach in the objective space

The results show that the proposed algorithm provides a placement plan that costs slightly lower, $1.29/hr as compared to $1.293/hr (**Figure 12** (a)). It offers slightly lower network latency, 1205ms as

compared to 1215ms (**Figure 12** (c)). The average CPU speed of the placement plans from the two algorithms remains the same (**Figure 12** (b)). In the case of System Availability, the proposed algorithm generated placement plan performs slightly poor than that generated by the benchmark, 95.6% compared to 95.7% (**Figure 12** (d)). These results show that the proposed algorithm performs better in terms of 2 criteria, poorer in terms of one criterion, and performs equally in terms of one criterion. Only with these results of the comparison in the objective space, however, we cannot conclude that the proposed algorithm outperforms the benchmark approach. This is due to the presence of multiple criteria and their different units of measurements. Due to the presence of the objective conflict, reaching a conclusion with results of the comparison in the objective space would not be the right thing to do. Hence we perform the comparison based on other reliable comparison metrics that are widely used to compare two multi-objective optimization algorithms in operation research domain (details in the next section).

3.5.3.2 Comparison of Solutions in terms of Standard Comparison Metrics

Unlike evolutionary algorithms for single objective optimization problems, where the comparison between the two is simple as the performance can be directly linked to objective functions, comparison of two multi-objective optimization algorithms is not

that simple for it involves a set of decision variables and their corresponding objective vectors (Deb & Jain, 2004). Due to such dimensionality feature, instead of only comparing through an objective vector, we employ other reliable methods that are suggested for comparison of two algorithms.

Such reliable performance comparison methods should consider two aspects of the algorithm – convergence, and diversity (Deb & Jain, 2004) (Veldhuizen, 1999). Convergence refers to the ability of the algorithm to approach the Pareto optimal front as close as possible, and diversity refers to the ability to maintain a diverse set of solutions. Various performance metrics have been suggested to compare either diversity or convergence or both (Riquelme et al., 2015).

We employ three of such metrics to compare the performance of our proposed algorithm with the benchmark approach as is employed in (Coutinho et al., 2015), which performs the optimization of multiple objectives with a linear transformation of the multi-objective optimization problem to a single objective optimization problem. The first metric measures convergence, second measures diversity and the third metric give a measure of both convergence and diversity.

A. Comparison with respect to Generational Distance (GD Metrics) - A Convergence Measure

To evaluate the algorithm for convergence, we employ Generational Distance (GD) (Veldhuizen, 1999). It shows how far are the solutions that are generated by the algorithm from the actual Pareto Optimal front (Veldhuizen, 1999), or from a reference set when actual Pareto optimal Front is unknown (Riquelme et al., 2015). It is evaluated by calculating the average Euclidean distance between the solutions in the final set of solutions and the nearest member of the actual Pareto Optimal Front (or reference sets). Since the actual Pareto Front is unknown in our case; we take the respective final evaluated solutions, as a reference, to calculate the average Euclidean distances of the solutions in each case. This provides a measure of convergence or a measure of how well the algorithm is approaching the final solutions. Obviously, the lower values are performed.

Figure 13 (a) provides a comparison of the evaluated GD for benchmark approach and the proposed approach. The superiority of the proposed approach is not apparent from the graph. However, if we observe the average values of GD from 100 simulation runs, we see that the benchmark provides 0.98 and the proposed approach provides 0.95. It suggests that the proposed approach is slightly superior to the benchmark approach.

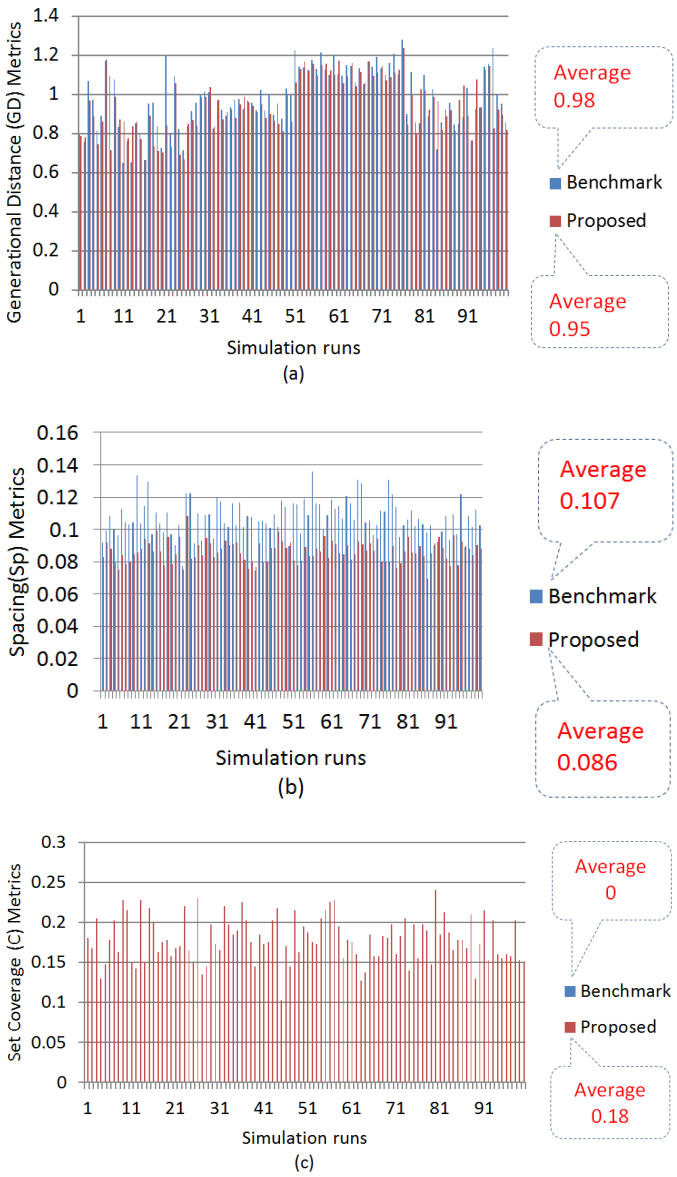


Figure 13: Plots of the performance metrics

B. Comparison with respect to Spacing (Sp- Metrics) - A Diversity Measure

To evaluate the algorithm for diversity we employ Sp metric (Riquelme et al., 2015; Schott, 1995). Sp metrics gives a measure of

how evenly the members of an approximation set are distributed (Riquelme et al., 2015). An Sp-metric value of 0 means the solutions are equidistantly spaced in the objective space. Lower values are preferred over higher ones. The graph for 100 simulation runs is shown in **Figure 13** (b).

From the graph (**Figure 13** (b)), it is evident that the Sp-metric from proposed approach in most of the simulation runs has lower values with the average of 0.086 compared to the benchmark approach which yields an average of 0.107 This indicates that the solutions (placement plans) from the proposed approach in the objective space are better in terms of equidistant distribution compared to that from the benchmark approach. It shows that the proposed algorithm outperforms the benchmark approach by providing solutions that better represents and carries all the features of potential placement plans.

C. Comparison with respect to Set Coverage (C-Metrics) - A Measure of both Diversity and Convergence

Two Set Coverage or Coverage, or simply C metrics (Hiroyasu et al., 1999) is used for comparing the performance of two multi-objective optimization algorithms. For two approximation sets A and B, the C-metric C(A, B) yields the fraction of solutions in B that are dominated by at least one solution in A (Riquelme et al., 2015). In 100 simulation runs, we recorded the average value of

C(P, B) to be 0.18, while that of C(B, P) being 0, where P represents the solution sets generated by proposed approach while B refers to the solution sets generated by the benchmark approach. It means that none of the solutions generated by the proposed algorithm is dominated by any of the solutions generated by the benchmark approach. While on the other hand, 18% of the solutions generated by the benchmark approach is dominated by at least one of the solutions generated by the proposed approach (**Figure 13** (c)). C-metric being a measure of both convergence and diversity (Hiroyasu et al., 1999), we can say that the proposed approach outperforms the benchmark approach in terms of both convergence and diversity.

3.6 Conclusion

3.6.1 Summary

In this chapter, we presented a comprehensive multi-criteria decision model for service placement in the federated cloud. Through an extensive literature review, we identified *financial cost, processing speed, network latency*, and *system availability* as relevant and measurable criteria that are important for the service placement decision making. As a measure of network latency, we considered the effect of application footprint in addition to that of the application topology. Based on the identified criteria, we proposed a service placement algorithm, which makes an optimal

selection of a set of provider resources distributed across the federation for the placement of service nodes of the customer applications. The selection is based on the application requirements and consumer preference stated as a pairwise comparison of the aforementioned decision criteria. In order to select the service placement plan, we first employed NSGA II to perform multi-objective optimization and selected a set of known pareto optimal. Next, we employed AHP to convert consumer stated preferences into respective weights and applied these weights to the identified set of known pareto optimal placement plans to evaluate their overall fitness and select the one with the best fitness value.

The simulation results demonstrated the effectiveness of the algorithm in making service placement decisions by making optimal tradeoffs between cost and various QoS parameters as per the consumer preferences expressed as a pairwise comparison between those criteria. The results show that the algorithm allows a selection of the placement plans at various tradeoff points allowing for the optimization of multiple objective functions, for instance cost reduction by up to 4% (ranging between $1.265/hr. to $1.315hr), increase in CPU speed by up to 47.8% (ranging between 2.93 to 4.33), decrease in network latency by up to 36.6% (ranging between 1073ms to 1466ms), an increase in system availability by up to 5.5% (ranging between 92.8% to 98%), while each of them being

among the known pareto optimal placement plans identified through the parallel optimization of all the four criteria.

Results also demonstrated that the algorithm outperforms the benchmark approach (weighted sum) (Coutinho et al., 2015). The proposed-algorithm-generated placement plan performed better in two, equally in one, and poorer in one out of four objectives in comparison to the benchmark-approach-generated placement plan when compared in the objective space. Results also demonstrated that the proposed algorithm outperformed the benchmark approach in terms of standard comparison metrics. The result shows that the proposed approach is better in terms of convergence, as suggested by the Generational Distance (GD) metric (Veldhuizen, 1999), with 0.95 from proposed compared to 0.98 from the benchmark. The proposed approach is better in terms of diversity, as suggested by Spacing (Sp) metrics (Schott, 1995) (Riquelme et al., 2015), with 0.086 from proposed compared to 0.107 from the benchmark. And, the proposed approach is better in terms of both convergence and diversity, as suggested by Set Coverage(C) metrics (Hiroyasu et al., 1999), with 0.18 from proposed compared to 0 from the benchmark. The comparison results for the measurement of convergence and diversity is in line with and supports the arguments made by existing research work (Deb & Jain, 2004).

3.6.2 Implications

The proposed service placement decision model and the algorithm attempts to address the problem of service deployment in a multi-provider federated cloud environment. It does not only selects the cloud for service deployment, rather specifies at a more granular level, for each application service node, the selected VM type hosted at a particular data center of a cloud provider, where it should be deployed.

The algorithm is beneficial to various stakeholders of the cloud service market, viz. cloud consumer, cloud providers, a federation of cloud providers, and cloud brokers. It allows a cloud federation or cloud broker to deploy its customer application in an optimal way possible, where individual consumers define what the '*optimal way'* is for the deployment of their applications.

The algorithm can also be employed by cloud federation or a cloud broker to present a number of alternative deployment plans near the boundary region of the identified placement plan. With this, a cloud consumer could be presented with a number of what-if scenarios. For example, the degree of system availability that could be achieved if the budget limit is increased by a certain amount, or how cost can be lowered if the consumer is still satisfied with a reduction in system availability by a certain value. Similarly, consumers can be presented with an idea of how much they can

benefit with regards to network latency if they can compromise on some degree of system availability or vice versa. There could be a number of other what if cases, too. This enables the cloud federation or a cloud broker to offer service variety, which has been found to be helpful in extending the market (Wei Wang, Li, & Liang, 2012). It not only benefits a cloud federation or a broker with the expanded market, but also an application provider allowing them to have their application deployed with optimal QoS level that is within their budget limit. It also allows for better QoE by the end users of the application with reduced application response time, which have been found to increase the engagement time, providing an additional advantage to the application provider.

The algorithm enables the cloud federation operator or a cloud-broker to offer customized placement services with additional what-if analysis, such as how much gain in system availability is possible if the cost can be increased by a certain amount. Such services can distinguish them in the competitive cloud service market and help them retain existing and attract new customers (Wei Wang et al., 2012).

In the proposed system model, the cloud providers are inquired by the resource broker component of the federation platform, for resource availability, configuration, and price of VMs, for each incoming service request. This approach provides flexibility to the

member providers for adjusting their prices depending on their workload and other factors and hence enables them to maximize their individual benefits, too.

The usefulness and application of the algorithm become more pronounced in the coming days as more and more applications are being developed or converted into micro-service architectures (Balalaie, Heydarnoori, & Jamshidi, 2015), and the cloud industry becoming more fungible with standards and protocols (Altmann, Bañares, & Petri, 2018). An application built on micro-service architecture, is made up of a number of independent and loosely coupled micro-services that involve minimal data communication, can be better benefitted by their distribution on cloud resources across the federation (Buyya et al., 2009). In this context, the algorithm allows for the selection of resources considering the specifications at a more granular level and optimize for a specific component of the application.

For the academic community, it provides important implications demonstrating how AHP can be applied together with the evolutionary multi-objective optimization to solve multi-criteria decision-making problems involving a large search space. It also shows the demonstrated benefits of the reduction of the search space by the application of multi-objective optimization techniques

before the application of weight vector for searching a final solution over conventional *scalarization* approach.

3.6.3 Future Works

Our future work will focus on the integration of the user prediction model based on machine learning in the service placement algorithms, thereby considering a varying number of users for different service components of a single application originating from different Points of Interests for finer optimization of service placement plan.

Chapter 4. A Contribution Based Revenue Sharing Scheme for Cloud Federation using Shapley Value

4.1 Introduction

4.1.1 Motivation

Cloud industry is susceptible to the economies of scale. Due to the discrimination by the economies of scale to small cloud providers (Kim et al., 2014), the majority of the market share has been occupied by a handful of providers. According to a recent report by Gartner (Gartner, 2018b), 75% of the market in Infrastructure as a Service (IaaS) segment belongs to five hyper-scale providers. In this context, cloud federation has been considered as a solution to address the existing challenges of smaller IaaS providers and increase their competitiveness (Ferrer et al., 2012; Haile & Altmann, 2015; Petcu, 2014; Rochwerger et al., 2009). It has the potential to enable cloud providers, especially smaller ones, to collaborate and gain access to an increased number of cloud infrastructure resources, and benefit from the economies of scale (Kim et al., 2014). It also helps them ensure the users' quality of service (e.g. with reduced latency) (Toosi, 2014), minimize costs (Hassan et al., 2014), and provide guaranteed availability of customer applications through reliable multi-site deployments (Petcu, 2014).

Therefore, cloud federation has been an active research area in recent years (James Cuff, Ignacio M. Llorente, Christopher Hill, 2017). Ample research has been carried out focusing on various challenges (Haile & Altmann, 2018; Heilig et al., 2017; Risch & Altmann, 2009). It has already shown promising results within the academic and research community, EGI federated cloud being a notable example (Fernández-del-Castillo et al., 2015). However, despite significant potential benefits, extensive research, and successful use case in the research and academic domain, no commercial cloud federation seems to exist in the market.

Some researchers have investigated the factors hindering the adoption of cloud federation (Breskovic et al., 2011; Haile & Altmann, 2015). Factors that incentivize cloud providers to collaborate as federation members have also been investigated (Breskovic et al., 2011; Haile & Altmann, 2015; Hassan et al., 2014; Jeferry et al., 2015; Roth, 1988; Samaan, 2014). Revenue sharing issue has been recognized as one of the important factors influencing cloud providers' decision to join and continue working in the Federation (Coronado & Altmann, 2017). Revenue sharing mechanism specifies how federation members share the infrastructure resources and distribute the revenue generated from the collaborative efforts. This calls for tools and methods for managing the business relationships such that it incentivizes the federation members to cooperate for the federation to be able to compete with the hyper-scale providers for the market share.

4.1.2 Problem Description

For a cloud federation to be able to gain the market share from hyper-scale providers, it should increase their competitiveness as that of a hyper-scale provider, for example by increasing capacity utilization (Goiri et al., 2012), providing better QoS (Petcu, 2014), and offering service variety (Toosi, 2014). It can do so by exploiting the benefits of the economies of scale, which is possible only through the aggregation of both supply & demand (Harms & Yamartino, 2010).

This requires the federation to operate in a co-operative setting. In order for the cloud providers to be willing to work in such settings, there should be a clear business model that should incentivize the federation members by ensuring i) more profits than they would earn by working individually, ii) fair allocation of revenue shares, iii) stable revenue stream, and iv) incentives for individual excellence. At present, the cloud industry is not clear on the models that define the business relationships for cloud federations (ieeeCESocTV, 2018). Majority of the research works focus on technical aspects such as energy efficiency, virtualization technologies, performance requirements, resource management, and latency (Bañares & Altmann, 2018) (Ataie et al., 2018). Some research has been carried out focusing on the economic aspects of cloud federations, too (Guazzone et al., 2014; Hassan et al., 2017; Hespanha, 2011; Li et al., 2013; Mashayekhy et al., 2015; Rohitratana & Altmann, 2012; Samaan, 2014; Uzbekov & Altmann, 2016).

Majority of those research deal cloud federation as a non-cooperative coalition (Guazzone et al., 2014; Li et al., 2013; Samaan, 2014), where federation members focus on individual strategies and payoffs (Hespanha, 2011), and these strategies guide how sharing of resources and revenue takes place. Few researchers have attempted to study cloud federation by considering it as a co-operative coalition, and have tried analyzing through Cooperative Game Theory (Hassan et al., 2017; Mashayekhy et al., 2015). However, these studies focus only on coalition formation, which addresses only one aspect of the problem in cooperative game theory (Serrano, 2007). The problem associated with the allocation of collective payoffs among the federation members, which is another important aspect of cooperative game theory (Serrano, 2007), has not been adequately addressed by existing research. Features of an allocation mechanism are crucial as any ill-defined methods may lead to unfair allocation, possible promotion of free riders, demotivation for cooperation, and decreased competitiveness, and hence affects the sustainability of the federation. Hence it is crucial to address the aforementioned issues, which have not received adequate attention by previous research, to encourage prospective cloud providers for joining the federation.

4.1.3 Research Objective and Research Questions

The objective of this research is to fill the research gap and present a Revenue Sharing Scheme for a cloud federation, which ensures higher profits than individual operation, fair allocation of revenue, stable

revenue stream, and incentives for individual excellence by allocating revenue shares in proportion to the contribution made in the value creation of the federation. Thus we try to address these research questions - i) what features of a federation member contribute to the value creation of the federation and what indicators can we use to measure them? ii) Based on the identified indicators, how can we fairly estimate the contributions of federation members and allocate the revenue shares according to the estimated contribution? and iii) Will the proposed scheme be universally attractive in all contexts for federation members of all characteristic types?.

4.1.4 Methodology

We model the cloud federation as a cooperative organization that competes with other federations or hyper-scale providers, but at the same time provides space for individual excellence and profitability. We apply Shapley Value Method (Shapley, 1953), an approach in Coalitional Game Theory for allocating the revenue share among the federation members; where, the revenue is generated from serving consumer requests by making use of resources selected through Service Placement Algorithms, such as (Aryal & Altmann, 2018). We chose Shapley Value Method because it enables to generate a single payoff vector that allocates the revenue share on the basis of individual contribution. Use of Shapley Value for revenue settlement has been discussed in other domains like the coalition of Internet Service Providers (Ma, Chiu, Lui, Misra, & Rubenstein, 2010). The revenue

share is evaluated on the basis of the contribution made by a federation member in the value creation of the federation. The federation value in this context refers to the revenue generated by the cloud providers by working as a coalition. We consider both infrastructure capacity and market strengths in evaluating one's contribution to the value creation of the federation. The infrastructure capacity is evaluated only in terms of the resource contribution made by the members. The resource contribution is evaluated only in terms of the actual resources utilized in service provisioning rather than the resource reserved for the federation. Resource utilization is determined by the service placement algorithm such as in (Aryal & Altmann, 2018), the discussion of which is beyond the scope of the research work in this chapter. The market strength of members is evaluated in terms of the value of the service requests brought in to the federation. The effectiveness of the proposed model is evaluated through extensive simulations covering different scenarios.

4.1.5 Contribution

Our contributions include the following.

- An innovative approach to the operation of cloud federation based on the economic model that fosters cooperation and competition at the same time, allowing for maximizing social benefits as well as incentivizing individual contributions of the federation members.

- A novel revenue sharing algorithm based on Shapley Value, a solution concept in Coalitional Game Theory, that provides a fair mechanism for revenue sharing that is based on the contributions that the federation members make.
- A novel approach to assess the contribution of the federation member in the value creation of the federation, which takes into account both the infrastructure capacity and the market strength of the provider.
- Evaluation of the proposed model and the algorithm through simulation covering a wide range of scenarios.
- Analysis, through simulation, to find the boundary line for when it is and it is not beneficial for IaaS cloud providers to work as members of the federation
- Demonstrate through simulation for how and why the proposed model is better than the benchmark Participatory approach.

4.1.6 Organization

The chapter is organized as follows. Related works are presented in section 2. Contribution based revenue sharing scheme – system model - is presented in section 3. In section 4, details on the simulation setup are given. Result and analysis are presented in Section 5. And, finally, the conclusion is presented in section 6.

4.2 State of the Art

4.2.1 The Issue of Revenue Sharing in Cloud Federation

Amongst the various challenges of realizing cloud federations, the issue of revenue sharing is one of the important one. Revenue sharing mechanisms determine how federation members share their resources to provision services to customer applications and do the allocation of the revenue generated from the collaborative efforts. An efficient mechanism for resource and revenue sharing is desired since it is a driving force to motivate cloud providers to work in a federation (El Zant et al., 2014).

Such revenue sharing mechanism has to deal with the management and utilization of a common pool of cloud resources in such a way that working in the federation is beneficial to everyone and there is no possibility for anyone to benefit as a free rider. This requires for pricing policies, resource use accounting, and a fair way of incentivizing federation members. We provide details on pricing policies used for a cloud computing environment in section 4.2.2 and on the existing method for revenue sharing for cloud federation in section 4.2.3

4.2.2 Pricing Policies Being Adopted by Current Cloud Industry

Cloud providers need pricing of their services for their business operations. Pricing is a process that determines the fee or the amount that the provider gets as an exchange for providing or selling its services to customers. For this, cloud providers are required to estimate

the value of their services and gain the estimated value through pricing. Cloud providers consider various pricing factors and pricing models to adapt their pricing strategies.

4.2.2.1 Pricing Factors

Basically, three factors have been suggested to be considered by cloud providers when setting their pricing strategies (Toosi, 2014). Cost of Service provisioning is one such factor, which requires that the final price be fixed by adding a certain percentage (a margin) to the actual cost of service provisioning. According to this, the price for the service varies with different providers due to the variations of the costs involved in provisioning service of different characteristics (quality). Various researchers have studied the cost models for IaaS cloud providers (Altmann & Kashef, 2014; Greenberg, Hamilton, Maltz, & Patel, 2008), which help in determining the actual cost of service provisioning for such cases.

The second factor that influences pricing strategy is the competition within the market for similar services. Pricing strategies based on market competition should set the price of services by being aware of the prices set by competing providers for service of similar configuration and quality. Researchers such as (Pal & Hui, 2013; Roh, Jung, Lee, & Du, 2013) have worked on market competition based pricing schemes that include Auction.

The third factor is the value of the service to the customers. The value of the service as perceived by customers may not always be correlated to the cost incurred in the service provisioning. The perception of the value of service may be subjective; however, at times, the providers may set the final price of the service by considering the service value to the customers rather than just the cost involved in the service provisioning. Few studies have considered this fact and have tried to assess the value of the cloud service from a consumer point of view (Padilla, Milton, & Johnson, 2013).

4.2.2.2 Pricing Models

At present, the cloud industry has adopted three basic pricing models, viz. Subscription based, Usage-based, and Demand-based. Subscription based pricing model allows customers to consume cloud service uninterrupted for a certain period of time by charging them a subscription fee for that period of time. Software as a Service segment has been widely adopting this model. Providers such as Amazon have adopted this pricing model for the IaaS segment, too. Customers make an annual or monthly reservation contract with the provider and consume the service for the contract period. The customer has to pay the subscription charge despite the utilization level. Such pricing model allows the customers to get the service at a discounted price while allows the providers to have a predictable and more assured revenue stream.

The most common pricing model in the cloud industry for IaaS segment is the Usage-based pricing model. In this model, customers pay as per their service consumption. Services are quantified and bundled with the name of Virtual Machine types, which represents different configurations with respect to CPU, memory, bandwidth, and storage. Service charge is specified in terms of VM Instance used per unit hour. Customers pay for the amount of VM instances utilized in per hour basis. Providers set the price point for this model at a higher level compared to that for the subscription-based model. The benefit to customers is that they need to pay only when the service is required and consumed. Research investigating this model include (Sharma, Shenoy, Sahu, & Shaikh, 2011; H. Wang, Jing, He, Qian, & Zhou, 2010).

Another model used for pricing in the cloud industry is the demand-oriented dynamic model. The price is set on the basis of demand for service; price is higher when the demand goes up and vice versa. Although this is a less common in the industry and non-existent among small scale providers as it requires complex optimization techniques and processes, it has been adopted by hyper-scale providers like Amazon referred to as Spot Pricing model (Amazon, 2019). Various researchers have investigated on this pricing model.

4.2.3 Existing Works Related to Revenue Sharing in Cloud Federation

Cloud federation being a coalitional game, the way how the revenue sharing takes place among the members of the federation is very important for incentivizing the federation members and for the sustainability of the federation. In order to achieve this, federations require an effective method that allows for revenue sharing among the federation members in a fair manner.

From existing works on cloud federations, we can observe three basic methods (namely, participation, auction, and contribution) and hybrid methods that combine the basic ones for sharing revenue among the federation members. In many cases, the revenue sharing is linked to pricing strategies such as an auction.

Table 10 provides some research works that guide revenue sharing in case of cloud federations.

A participation-based method is proposed by Niyato et al. (2011). It uses a stochastic linear programming approach to a coalitional game for the formation of an optimal and stable coalition. The coalition is formed taking into account internal users demand and coalitional cost. A similar model is proposed by Z. Lu et al. (2012). Both of these models assume CSPs to commit a certain level of resources to the federation, which involves some cost and compromise on the individual freedom of CSPs.

Table 10 Existing approaches to revenue sharing

Method	Approach & Objective	Remarks	Reference
Participation	Maximize federation benefits.	Compromised individual freedom	(Z. Lu et al., 2012; Niyato et al., 2011)
Auction (Spot Pricing)	Maximize individual profits	Discrimination due to economies of scale	(Samaan, 2014)
Auction (Modified)	Social benefit is maximized	Compromised individual freedom and discrimination due to economies of scale	(Hassan et al., 2017)
Hybrid (Participation + Auction)	Maximize federation benefits.	Compromised individual freedom and discrimination due to economies of scale	(Hassan et al., 2015)
Contribution (Market Share)	Maximize federation benefits	Evaluation of contribution may be unfair.	(Mashayekhy et al., 2015)
Contribution(resource)	Social benefit is maximized	Evaluation of contribution is limited to resources. It may be suitable for resource expansion by a provider. But may not be suitable for sustained operation of the federation	(Aryal & Altmann, 2017; Coronado & Altmann, 2017; Kaewpuang, Niyato, Wang, & Hossain, 2013)

Spot pricing, which is an Auction based method is proposed by Samaan (2014). This method models cloud providers' interactions as a repeated game played among a set of selfish providers who aim at maximizing individual benefits. These providers interact with each other to sell their unused resources in the spot market with individual profit maximization objectives. This method is applicable in non-cooperative settings and the drawback with this method is that smaller providers are still at a disadvantaged position due to the discrimination by the economies of scale.

Hassan et al. (2017) propose a varied form of the auction method where the auction is carried out with the aim of social welfare maximization rather than the maximization of individual benefit. For the maximization of social welfare, a game model is proposed that looks for a set of cloud providers with low energy cost. As with other auction models, this too has an effect on the fairness in revenue sharing putting smaller ones in a disadvantaged position.

Another method proposed by Hassan et al. (2015) includes a coalitional formation game that aims to maximize social benefits. It employs a hybrid method that combines participation based and auction methods for revenue sharing. Provider resources are selected in such a way that the total cost is minimized. Broker fixes the revenue rate. It then receives a number of VMs offers from the CSPs on that rate. Revenue rate is adjusted (increased or decreased) according to the participation

of the CSPs and an optimal value is reached in a number of iterations. Individual freedom is compromised in this approach and is unfair as economies of scale benefit larger providers who can decrease operation cost. How the surplus revenue (profit) is distributed is not explained.

A revenue sharing scheme in a cooperative setting is proposed by Mashayekhy et al. (2015), and also by Kaewpuang et al. (2013). In the case of Mashayekhy et al. (2015)'s approach, the resource selection is done using integer programming in a way that maximizes federation profit through minimizing the cost of service provisioning. And, in the case of Kaewpuang et al. (2013)'s approach, three different methods are proposed, namely linear programming, stochastic programming, and robust optimization for optimized resource allocation in different scenarios. For revenue sharing, both approaches employ Shapley Value method. Mashayekhy et al. (2015)'s approach relies on the market share of each provider for the estimation of their marginal contribution in the federation. Kaewpuang et al. (2013) consider resources for calculating their marginal contribution. The issue of fairness arises in both of these approaches. As the contribution is calculated only according to the market share in the first case, new entrants with substantial resource contribution may receive lower revenue due to the fact that they are yet to occupy appropriate market share. Similarly, in the second approach, the providers who have significant market presence may not be incentivized for bringing in the business to the federation

4.2.4 Existing Works in Revenue Sharing with Shapley Value Method in Various Fields

Shapley value method provides a way of evaluating the marginal contribution of a member in a coalition of a number of members. Such marginal contribution may be taken as a basis for allocating the payoffs of the coalition among the members. The use of Shapley value method for revenue sharing has been studied in various fields including among the coalitions of network service providers (Amigo, Belzarena, Larroca, & Vaton, 2011), internet service providers (Lee, Jang, Cho, & Yi, 2012), supply chain (Kemahlioglu Ziya, 2004; Yi, 2009), and cloud computing (Mashayekhy et al., 2015) (Kaewpuang et al., 2013).

Application of Shapley value in the coalition of Internet Service Providers is investigated by Lee et al. (2012). The authors investigated the stability of a grand coalition with Shapley Value method for revenue sharing under over-demanded and under-demanded conditions of the network. They conclude that the grand coalition is stable at under demand condition but not in the over-demand condition.

Amigo et al. (2011) proposed a mechanism for revenue sharing based on Shapley value among the federation of network service providers and demonstrated that it provided fair sharing. It is also proved that the model provided an incentive to the providers for adding more resources.

Revenue sharing model based on Shapley Value in the supply chain network that includes a coalition of suppliers and retailers is proposed by Kemahlioglu Ziya (2004). Another work on revenue sharing for supply chain industry is also proposed by Yi (2009), where, in order to improve the effectiveness of the revenue share, risk and investment factors are considered in the evaluation of Shapely Value.

As discussed in section 2.3, Shapley Value method has also found application in the field of cloud computing to solve the problem of revenue sharing among member providers (Kaewpuang et al., 2013; Mashayekhy et al., 2015). Mashayekhy et al. (2015) propose the allocation of the revenue share based on the contribution assessed in terms of market share. Kaewpuang et al. (2013) allocate the revenue share based on the contribution assessed in term of the resources.

4.3 System Model

In this section, we outline the use case and architecture for the federation, Definitions, and notations of the parameters involved, and the various aspects of the Revenue Sharing Algorithm

4.3.1 Use Case and Federation Architecture

We model the system as a number of cloud consumers seeking application deployment service, and a set of cloud providers who are willing to cooperate with each other in serving the consumer requests. The interactions between the providers are coordinated by the federation platform engine as described in Chapter 3 (**Figure 2**). We

consider the interaction between member cloud providers as a coalitional game with *n* number of player. The objective of the coalitional game is to evaluate the payoff vector for distributing the revenue among the members of the coalition in a fair manner.

The federation platform engine in the system model, as shown in Chapter 3 (**Figure 2**), is inspired by the federation platform proposed in BASMATI (Altmann et al., 2017). It consists of six major components – i) Request Handler, ii) Resource Broker iii) Service Placement Maker which constitutes User Preference Evaluator and Placement Plan Optimizer, iv) Federation Business Logic that constitutes Accounting and Revenue Sharing modules, v) SLA Repository, and vi) Application Monitoring.

Application provider or a cloud service consumer requests for the service placement to the cloud federation with requirement details through a Cloud Provider. The request is handled by the Request Handler Component. The request Handler gets all the required details needed for the application deployment in predefined structure and format. The Resource Broker, of the federation platform, then, requests the information with each of the providers regarding the resource availability. Based on the availability information received from each of the providers and the application requirements, the Placement Plan maker finds the optimal placement plan for the service placement. The Placement Plan Maker does so by making use of two of its sub-

components – Placement Plan Optimizer component and User Preference Evaluator subcomponent. The Placement Plan Optimizer component identifies a set of pareto optimal placement plans by employing the evolutionary multi-objective optimization algorithm. Similarly, the User Preference Evaluator sub-component evaluates the preference weight vector representing the consumer preference over various decision criteria. The Service Placement Plan Maker makes selects one of the plans from the identified set of pareto optimal ones based on the fitness evaluated as the weighted sum of the fitness values by applying the consumer preference weight vector. We make use of the service placement plan maker as in (Aryal & Altmann, 2018), and hence the detail discussion of this is out of the scope of the research work in this chapter.

After the service provisioning is started with application deployed as per the selected placement plan, the Federation Business Logic Component now gets activated for this request. The accounting sub-component keeps accounts for all resource requests that come to the federation with the details of the providers serving that request along with the characteristics of the request that includes unique request ID, type and number of the VM, service start time and service end time. The Revenue Sharing sub-component, at pre-specified time intervals, by use of the revenue sharing algorithm, decides the payoff vector and allocates the revenue share to each of the federation members.

4.3.2 Parameter Definition and Notations

In this section, we present the descriptions and notations of the parameters involved in the model. For the readers' convenience, we provide a summary of it in **Table 11**.

A. Cloud Federation

A *cloud federation* is a strategic alliance of a set of cloud providers who have voluntary agreement to interconnect their cloud infrastructure and enable resource sharing among them (Haile & Altmann, 2015). It is denoted as $N = \{P_1, P_2, \dots, P_n\}$, where P_i denotes Cloud Provider i.

B. Cloud Providers

In the given problem context, A *cloud provider* is an entity which provides infrastructural IT capabilities in the commonly known Infrastructure as a Service (IaaS) cloud service delivery model. With respect to any service request, a Cloud provider in the federation can play the role of either Insourcing Provider or Provisioning Provider or both or none.

C. Insourcing Provider

An Insourcing provider is one that brings the service request into the federation. Insourcing Providers are represented as a vector $B = \{b_1, b_2, \dots, b_k\}$, where b_i represents a provider bringing the request i into the federation.

D. Provisioning Provider

A Provisioning provider is one that hosts one or more of the application services comprising the service request in their data center by provisioning one or more of the virtual machine instances. The Provisioning Providers for k requests are denoted by a vector $F = \{F_1, F_2, \dots, F_k\}$, where F_i denotes the vector of Provisioning Providers for a service request i. Application topology with respect to each of the service request involves m nodes, each of which requires one virtual machine instance, provisioned by one or more Provisioning providers. The Provisioning Providers for a service request i is denoted by $F_i = \{f_{1,i}, f_{2,i}, \dots, f_{m,i}\}$, where $f_{j,i}$ denotes the provider provisioning virtual machine instance for node j in service request i.

E. Service Request

A service request refers to a request made by an application provider (cloud consumer) to a cloud provider for the deployment of its application in the cloud. The service requests for k requests are denoted as $R = \{R_1, R_2, \dots, R_k\}$. where, R_i denotes service request i. A service request is specified by the application topology specifying application service nodes. Application service node is specified by the CPU and memory configuration, which is mapped to certain VM types for provisioning. A service request i is represented as $R_i = \{v_{1,i}, v_{2,i}, \dots, v_{m,i}\}$. Where, $v_{j,i}$ refers to the VM type for the j^{th} node in the i^{th} request. The relationship defines the pair of nodes that

hold strong data communication requirements, which is later used by the service placement algorithm for the optimization purpose.

F. Service Duration

The requested service runs for a certain duration of time until it is terminated or a certain condition is met. The service durations of k requests are denoted by a vector $D = \{d_1, d_2, \dots, d_k\}$. Where, d_i refers to the duration for which request i is served.

G. Virtual Machine (VM) Instance

Virtual Machine Instance is the unit used by cloud providers to bundle a set of cloud infrastructure resources. A Virtual Machine instance is specified by the number of CPU cores and Memory. A cloud provider can provide different types of Virtual Machines with different configurations. For simplicity, we assume all of the member providers of the federation offer the same types of virtual machines. We assume that there are m types of virtual machines defined as $V = \{ v_1, v_2, \dots, v_m \}$, where v_j refers to virtual a machine of type j. The number of Virtual machines of different types that a provider can make available for the customer request at a point of time inquired by the resource broker (federation platform) depends on the amount of resources in use and the total capacity of the cloud owned by the provider.

H. VM Instance Cost

VM Instance cost refers to the amount in USD that a provider charges for its use for the duration of one hour. Hour is the smallest time unit for charging purposes. Mostly, the configurations of the VM Instance determine its cost of the VMs. However, the cost for the same VM type differs among different members depending on various factors such as Cloud availability, CPU speed, location, time, etc. The cost that a provider pays for the VM instance corresponding to an application node depends on the VM type required for the node and the provider provisioning the VM instance for this node. The cost for the node i^{th} node of an application with respect to service request k is represented as $c_{v_{i,k},f_{i,k}}$, where $v_{i,k}$ denotes the VM type of the i^{th} node in service request k, and $f_{i,k}$ denotes the provisioning provider provisioning the VM instance for the i^{th} node in service request k.

I. Cloud Capacity and Availability

A provider can own one or more clouds. A cloud refers to a data center situated at a certain geographic location that makes computing resources available through the internet. For the problem context, we consider IaaS cloud, and hence their capacities are expressed as the number of computing resources that it hosts. For simplicity, we consider a total number of CPU cores and the total memory capacity. For reference purpose, we simplify and express the capacity as a single resource unit where one unit of resource means one CPU Core and 4 GB of memory. The 1:4 ratio between CPU core and memory is chosen

based on the ratio derived by taking the average from the configurations among different VM Instances offered by Amazon (AWS, 2019). The availability of cloud is expressed as the fraction of the time the system is up and running.

J. Sub-coalition and Sub-coalition Value

A *sub-coalition* S is any non-empty subset of the federation N. i.e. - $S \subseteq N$. In simple terms, a sub-coalition is a coalition composed of any number of providers within the federation N. The value of the *sub-coalition* S, denoted as $v(S)$is defined as the revenue generated by the collaborative efforts of the members in the sub-coalition S. This means the revenue that the member providers in the sub-coalition generate without the support of any other members federation which is out of sub-coalition S. This requires that, for any job request the provider who brings in the job request to the federation and the providers who serves the request should be within the sub-coalition S. The detail on the evaluation method for the sub-coalition value is presented in section 4.3.3.3.

Table 11: Parameter Definition and Notation

Parameter	**Notation**	**Meaning**
Cloud Federation	$N = \{P_1, P_2, \dots, P_n\}$	A cloud federation N comprising of n number of providers
Cloud Provider	P_i	Cloud Provider i
Sub-coalition	$S \subseteq N$	A sub-coalition S formed out

of federation		of the grand coalition N
Service Requests	$R = \{R_1, R_2, \dots, R_k\}$	Service request vector comprising of a total k number of requests during a specified period
A service request	$R_i = \{v_{1,i}, v_{2,i}, \dots, v_{m,i}\}$	A vector representing the required VM types of each application service node in service request i
Virtual Machine node	$v_{j,i}$	Required Virtual Machine type for the j^{th} node in the i^{th} request
Service durations	$D = \{d_1, d_2, \dots, d_k\}$	Service duration vector comprising of the service durations for each of k requests
Service duration of a request	d_i	Service duration in hours for request i
Insourcing Providers	$B = \{b_1, b_2, \dots, b_k\}$	Vectors representing the providers bringing in each of the k requests to the federation
Insourcing provider for a request	b_i	Provider bringing the service request i in to the federation
Provisioning Providers	$F = \{F_1, F_2, \dots, F_k\}$	Vector of Provisioning Providers for k service requests
Provisioning Providers of a request	$F_i = \{f_{1,i}, f_{2,i}, \dots, f_{m,i}\}$	Vector of providers serving each of the application service node (VM) in request i
Provisioning provider for a node of a request	$f_{j,i}$	Provider provisioning the VM instance for j^{th} application node(VM) of the service request i
Value of a sub-coalition	$v(S)$	Revenue generated by the sub-coalition S

4.3.3 Resource Scheduling and Revenue Sharing for Cloud Federation

The business relationship between the federation members is defined by how the resources are shared for serving the customer requests and how the revenue generated by collectively serving the customer requests is shared among the federation members. This requires three models to be defined properly, namely pricing, resource scheduling, and revenue sharing. Pricing determines the fee or the amount received as an exchange for providing or selling the services to customers. The detail on the pricing strategy for the proposed model is presented in section 4.3.4. Resource scheduling determines the resources across the federation that should be combined for serving the customer requests in an optimal way. We employ the Service Placement Algorithm as described in chapter 3 for scheduling the resources for serving the customer requests and hence any further detail is not required in this chapter. The revenue sharing specifies the business rules and methods that determine how the collectively generated revenue is shared between the federation members. Details on the revenue sharing are presented in section 4.3.5.

4.3.4 Pricing

For the proposed federation, we adopt usage-based pricing model. The federation members individually set price for their services in terms of Virtual Machine Instance per hour. Federation members reveal their

price for each virtual machine types along with their specifications while reporting resource availability information on request, with respect to each service request, to Resource Broker of the federation platform. With this mechanism, they are allowed to set the price by taking into account various factors such as the cost of service provisioning, competition in the market for service with similar quality, and their perception of customer value of their services. Such pricing strategy has been chosen based on the reasoning that service characteristics of VM instances from different member clouds are different, and so are the characteristics of demand for those services. In this context, the pricing model that allows taking into account the demand, cost of service provisioning, and service quality and the value of service from customer viewpoint would allow for better federation benefits with the flexibility. This will also encourage the formation of cloud federation consisting of providers with different service characteristics by allowing some form of competition within the collaboration. This also provides a better incentive mechanism compared to the federation that adopts flat pricing strategies, where the price for each VM instance types are fixed and the same throughout the federation.

A question of efficiency and fairness that could arise in the proposed differentiated pricing strategy is that some aggressive federation member could direct the majority of requests to itself by setting the price at a much lower point and thereby limiting other providers'

participation in the service provisioning. However, such behavior and phenomenon are restricted in our proposed model with the adoption of the multi-criteria algorithm, which determines how the provider resources are selected for the service placement. With the adopted service placement algorithm, the resource scheduling for the service request is done based not only on a single factor 'cost'; rather, it is dependent on the various other factors such as processing speed, network latency (location of data center), and availability of the cloud data center, as well. Further, the tradeoff among these factors is determined not by the cloud provider or the federation; rather, it is determined as per individual consumer preference. This will minimize if not eliminate the significance of the question with respect to fairness and efficiency.

4.3.5 Revenue Sharing

We model the cloud federation as a coalitional game. As a coalitional game, we are not interested in the way how individual providers make choices within the coalition; rather, we are interested in how the group of providers can achieve payoffs for itself. An important question to solve is – how to fairly divide the revenue generated by the collective work of the federation members among themselves. Our objective, here, is to develop a *revenue allocation mechanism.*

Definition 1. A *revenue allocation mechanism* is an operator φ on the federation of a set of cloud providers (N, v) that assigns a unique revenue vector

$$\varphi(P, v) = (\varphi_1, \varphi_2, \ldots, \varphi_n)$$

Where each $\varphi_i(P, v)$ refers to the revenue allocated for provider p_i as a result of its contribution in the federation. And, v is the value (total revenue) generated by the federation.

To address the issue of fairness it is important to define what fairness implies. Shapley (1953) argues that a coalitional game is considered to be fair if each member of a coalition receives a payoff share in proportion to their marginal contribution to the coalition. Thus, following Shapley (1953) and Young (1985), we want our *revenue sharing mechanism* for a cloud federation to exhibit the following properties (section 4.3.5.1) to demonstrate its effectiveness.

4.3.5.1 Properties of a Revenue Sharing Scheme

We design an appropriate mechanism for $\varphi(N, v)$ such that the following properties are satisfied.

1. *Efficiency Property*

The Efficiency property requires that all the revenues generated as a result of collective efforts by the federation members should be subject to distribution among member providers and no revenue amount should be left undistributed.

$$v(N) = \sum_{i \in N} \varphi_i(N, v)$$

2. *Dummy Property*

The Dummy property requires that a dummy provider should receive no revenue from the federation. A dummy provider is one which has zero *marginal contribution* to the federation.

The explanation of this property and the other that comes in the text below requires definitions of dummy player, sub-coalition, grand coalition, marginal contribution, and value. The definitions follow.

Definition 2. Any provider i is said to be a *dummy provider* if its' marginal contribution to all the sub-coalitions $S \subseteq N \setminus \{i\}$ is zero.

Definition 3. A *sub-coalition* S is any non-empty subset of the grand coalition, $i.e. - S \subseteq N$. This means, a sub-coalition is a coalition of any number of providers within the grand coalition that can potentially be formed in order to compose service for the resource request.

Definition 4. A *grand coalition* is defined as a coalition that includes all the providers in the federation. It is denoted by *N* and is defined as

$$N = \{p_1, p_2, \ldots, p_n\}$$

Obviously, it is the largest coalition that can be formed by a set of *n* providers.

Definition 5 The *marginal contribution* Δ_i of a member provider i in a sub-coalition $S \subseteq N \setminus \{p_i\}$ is defined as

$$\Delta_i = v(S \cup \{p_i\}) - v(S)$$

Where $v(S \cup \{p_i\})$ is the value of the coalition S including provider p_i and $v(S)$ is the value of the coalition S without provider p_i.

Definition 6. The *value* $v(S)$ of a sub-coalition S is defined as the revenue that the member providers in the sub-coalition generate without the support of any other members of the grand coalition who are not in the sub-coalition S. This means for any job request the provider who brings in the job request to the federation and the providers who serve the request should be within the sub-coalition$S$. The details on the evaluation of value are provided in section 4.3.5.3.

3. Symmetry Property

The Symmetry property requires that any two interchangeable member cloud providers of the federation should receive an equal share of the revenue. For any value v, if *i* and *j* are interchangeable then

$$\varphi_i(P, v) = \varphi_j(P, v).$$

Definition 5. Providers p_i and p_j are said to be *interchangeable* if

$$v(S \cup \{p_i\}) = v(S \cup \{p_j\})$$

This is to say that both providers p_i and p_j contribute equally in revenue generation to every sub-coalition. In order to be *interchangeable*, it is not required that the resources made available by providers p_i and p_j should be equal for every job requests. What is required is the resources made available by providers p_i and p_j for every job requests are such that every coalition of other providers beside p_i and $p_j$can fulfill the job requests by including either none of them, or both of them or any one of them, but not that they can fulfill the request with p_i and not with p_j, or vice versa.

4. *Fairness Property*

The fairness property requires that for all pairs of providers p_i and $p_j$in the federation, the contribution of p_i to p_j is equal to the contribution of p_j to p_i.

$$\varphi_i(P, v) - \varphi_i\big(P \setminus \{p_j\}, v\big) = \varphi_j(P, v) - \varphi_j(P \setminus \{p_i\}, v)$$

The idea here is to distribute the surplus that is generated by any collaborative efforts equally among the coalition members. To illustrate it further, let us consider a coalition of two providers, say p_1 and p_2. Now for these two providers in the federation, we can say that the revenue sharing mechanism is fair if their revenue share is evaluated as follows.

$$\varphi_1(\{p_1, p_2\}, v) = v(\{p_1\}) + \frac{1}{2}[v(\{p_1, p_2\}) - v(p_1) - v(p_2)]$$

$$\varphi_2(\{p_1, p_2\}, v) \;=\; v(\{p_2\}) + \frac{1}{2}[v(\{p_1, p_2\}) - v(p_1) - v(p_2)]$$

Here, the term $v(\{p_1, p_2\}) - v(p_1) - v(p_2)$ refers to the surplus revenue generated by the federation, i.e.- only with the involvement of both providers p_1 and p_2, where the involvement may be in the form of bringing in the service request or providing the VM resources to serve the request. Both providers p_1 and p_2 get half of the surplus revenue on top of what they could get without collaboration. Since the federation includes *n* number of providers, this same concept should be extended to all the sub-coalitions that can be formed out of these *n* providers in the federation.

4.3.5.2 The Revenue Sharing Mechanism

We are interested in designing the *revenue allocation mechanism* that satisfies the properties mentioned in section 4.3.5.1. More importantly, we need to generalize the idea mentioned in fairness property in section 4.3.5.1 for more than two players. In order to evaluate a unique vector of revenue share for the cloud providers working in the federation, we apply the Shapley Value Method, proposed by Lloyd Shapley (Roth, 1988), which generalizes the idea for more than two players (Mas-Colell, Whinston, & Green, 1995).

As a Shapley Value (Roth, 1988), the revenue share for provider p_i in a federation of providers N, is given by

$$\varphi_i(N) = \sum_{S \subseteq N \setminus \{p_i\}} \frac{|S|!\ (|N| - |S| - 1)!}{(|N|)!} [v(S \cup \{p_i\}) - v(S)]$$

Where,

N, *is the vector representation of the cloud federation with n number of providers*

S, *is the vector representation of a subcoalition formed from N*

p_i, *is the* i^{th} *federation member (cloud provider i)*

$v(S \cup \{p_i\})$, *is the value of a subcoalition S*

$v(S \cup \{p_i\})$, *is the value of a subcoalition S including provider* p_i

$[v(S \cup \{p_i\}) - v(S)]$, *is the marginal contribution of pi to subcoalition S*

$\varphi_i(N)$, *is the revenue share (value) of provider* p_i *in federation N*

$\frac{|S|!\ (|N| - |S| - 1)!}{|N|!}$, *is the probability that provider* p_i *joins exactly after coalition S is formed*

To calculate the revenue share of each provider p_i, the idea here is to calculate its marginal contribution in each of the sub-coalition S, which can be formed from N less provider Pi, i.e. - from $(N - \{pi\})$ and sum it across all such possible sub-coalitions by multiplying with respective probabilities of occurrences (probability of provider p_i joining the

coalition exactly after S). While we are using the probabilities here, it is important to note that allocation of value based on marginal contribution is influenced by the order in which a member enters the coalition, and hence, the Shapley value method provides a way to allocate just the average value to the federation members if they are entered in complete random order. It is achieved so by the expression $\frac{|S|!\,(|N|-|S|-1)!}{|N|!}$ in the equation above, which is the probability that the provider p_i enters the coalition exactly after sub-coalition S is formed

All possible sub-coalitions S are derived from the permutations. For example, in the federation of three providers denoted by $N = \{p_1, p_2, p_3\}$, all possible sub-coalitions are - $\{\emptyset\}$, $\{p_1\}, \{p_2\}, \{p_3\}, \{p_1, p_2\}, \{p_1, p_3\}, \{p_2, p_3\}$, and $\{p_1, p_2, p_3\}$. So, to calculate the value (revenue share) of the provider p_i in this federation, we evaluate the marginal contribution of provider p_1 in each of the sub-coalitions$\{\emptyset\}$, $\{p_2\}, \{p_3\}$, and $\{p_2, p_3\}$. Marginal contribution of provider p_1 in a sub-coalition $\{p_2, p_3\}$, for example, is calculated as value of the sub-coalition $\{p_2, p_3\}$ including p_1, i.e- $\{p_1, p_2, p_3\}$ minus the value of the sub-coalition $\{p_2, p_3\}$, which is expressed as $\Delta_{p_1}(\{p_2, p_3\}) = v(\{p_1, p_2, p_3\}) - v(\{p_2, p_3\})$. The procedure to evaluate the value of any sub-coalition $v(S)$ is given in section 4.3.5.3.

4.3.5.3 Calculating the Value of a Sub-coalition

The value of a sub-coalition is the revenue generated by that sub-coalition. In order to distribute the revenue share, at the end of every reconciliation period, it is necessary to evaluate the value of each of the possible sub-coalitions that can be formed from the members of the federation N. The value of a sub-coalition S with respect to a certain time period (such as a month) is the revenue generated from all the service requests during that time periods served by the members in the sub-coalition S, without the involvement of any other members which are not in S. Assuming there are k requests served by the federation during that time period, the value of a sub-coalition S is determined as a sum of the value of the sub-coalition S with respect to each of those k service requests. It is evaluated as follows.

$$v(S) = \sum_{i=1}^{k} v_i\,(S)$$

Where,

k, $is\ number\ of\ requests\ served\ by\ federation\ during\ a\ specified\ time\ period$

$v(S)$, $is\ the\ value\ of\ the\ subcoalition\ S\ for\ the\ specified\ time\ period$

$v_i(S)$, $is\ the\ value\ of\ the\ subcoalition\ S\ with\ respect\ to\ request\ i$

The value of a sub-coalition S with respect to a particular service request i is determined based on the cost of the service placement plan as determined by the service placement algorithm for the service request i. It is expressed as follows

$$v_i(S) = \begin{cases} \sum_{j=1}^{m} Cost(v_{j,i}) * d_i, & \text{if all } f_{j,i} \in S \text{ and } b_i \in S \\ 0, & otherwise \end{cases}$$

Where,

m, $is\ the\ nuber\ of\ the\ service\ nodes\ (VM\ nodes)\ in\ the\ application\ in\ request\ i$

$v_i(S)$, $is\ the\ value\ of\ the\ subcoalition\ S\ with\ respect\ to\ request\ i$

$Cost(v_{j,i})$, $is\ the\ Price\ of\ VM\ instance\ selected\ by\ the\ service\ placement\ algorithm\ for\ j^{th}\ application\ node\ in\ request\ i$

d_i, $is\ the\ service\ duration\ of\ request\ i$

$f_{j,i}$, $is\ the\ provider\ chosen\ by\ the\ service\ placement\ algorithm\ for\ serving\ jth\ application\ service\ nodeof\ request\ i$

$Cost(v_{j,i})$ is the per hour cost of the VM instance for j^{th} node of the application with respect to request i. It is determined as the cost of the Virtual Machine instance as quoted by the provider $f_{j,i}$, where $f_{j,i}$ is the provider which is selected by the service placement algorithm to provision VM instance to j^{th} node of the application in request i. As seen from the above equation, the value of the sub-coalition for a job

request will be equal to the sum of the product of the service duration d_i and cost of the virtual machine $Cost(v_{j,i})$ for m service nodes of the application, if it satisfies two conditions:

i) If all $f_{j,i}$, i.e. - all the providers selected by the service placement algorithm for provisioning VM instances for all m application nodes in request i are in the sub-coalition S, and

ii) Provider b_i, which is the provider that brought the request i to the federation, is in the sub-coalition S.

However, if the provider b_i or any of the providers among $f_{j,i}$ are not included in S, then in this case, the sub-coalition S is unable to generate the revenue from this request without the involvement of any other provider out of the sub-coalition S and hence the value of the sub-coalition S is evaluated to be zero.

4.4 Simulation

This section provides a description of the experimental setup for the simulation and settings for the simulation parameters.

4.4.1 Experimental Setup

For the evaluation of the proposed revenue sharing scheme, we performed a number of simulation runs covering different scenarios. We developed a computer program for simulating the service requests with different application requirements and performed the scheduling

of those requests to the appropriate providers (clouds) by employing the service placement algorithm as detailed in chapter 3 of this thesis following Aryal & Altmann (2018).

To generate service requests, we did the following.

1. We created a database of application topologies representing four different types of applications the federation is expected to receive service placement requests for. Each application topology specifies the number of nodes required, their configuration in terms of CPU and memory, and the data communication requirement between the pair of nodes if any. The settings of the parameters for these application topologies are given in section 4.4.2.
2. The infrastructure capacities of the providers are set in terms of the number of CPU cores and the memory that they possess. The market strengths of the providers are set in terms of the number of requests they receive during the study period. Specific settings of the parameters are given in section 4.4.2
3. For each provider, the simulator program generates service requests and assigns to the provider. For this, the application requirement corresponds to the application topology chosen at random from the pool as mentioned in step 1. And, the number of requests to them is made corresponding to their market strength as mentioned in step 2. The service duration (start and

end time for the service) is chosen randomly with exact values between the study period of 4 months. The settings of the parameters for this is made as explained in section 4.4.2.

The service placement algorithm then selects the provider resources for the deployment, as defined in section 4.3.1. We accounted for the details of each of the service requests, including the number and configuration of each of the application nodes, the provider bringing in the request to the federation, the providers serving the request, and the duration in hours for which the request was served. This accounting information is consumed by the proposed revenue sharing algorithm to allocate the revenue share to each of the federation members. The service requests are simulated for over a period of 4 months, and the details on the handling of the request are accounted for every hour.

For each provider, the resource utilization ratio is accounted for each hour during the study period deriving from the resource provisioned to requests that are active during that hour. Similarly, the revenue stream for each of the providers is calculated for each hour derived from the requests that are active during that hour. The revenue stream is calculated following the detailed procedure described in section 4.3.5.2 and 4.3.5.3.

The detail on how the setting is done for simulation parameters is presented in section 4.4.2, and the analysis of the simulation results in section 4.4.5. The accounting details during the first 200 hours and last

200 hours from time period considered for the simulation are chopped off from the result analysis to remove the outliers and present the results only for the period when the system is stabilized.

4.4.2 Parameter Setting

A summary of parameter settings for the simulation is presented in **Table 12**. A description of the settings of the parameters follows next.

4.4.2.1 Providers and End Users Related Parameters

This section provides details on the parameters related to Characteristics of Providers and End Users, and Provider Resources (see **Table 12** for a summary).

a. *Number of Providers and Number of Clouds*. We consider a federation of six IaaS cloud providers, each of them having two clouds in different geographic locations. We assume that this size is sufficient enough to represent a federation of moderate size and to evaluate the effectiveness of the model. This number remains fixed throughout the simulation.

b. *Number of users in each Point of Interest (POI).* The algorithm employed for the scheduling service requests to appropriate cloud requires this information to optimally minimize the network latency between user and application nodes. We set a randomly chosen values in the range 250 to seven million for each POIs with an assumption that it provides a wide enough

range to accommodate the requirements of any application of moderate size

c. *Utilization ratios of clouds.* It is the ratio of resources (memory and CPU cores) in use to the total resource capacity of a cloud on average. The basic value of average utilization ratio for each cloud in case of individual operation is set to a carefully chosen value in the range 10% to 70% referring to the Issue Paper by Natural Resources Defense Council (NRDC) (Whitney & Delforge, 2014).

d. *Provider Capacity.* The Provider capacity is determined by the physical server capacity of the provider. It is expressed in terms of the physical server capacity of the clouds that they own. Details on the cloud capacity are given in section 4.4.2.2.

Table 12: Parameter settings

A. Providers and end users related parameters	***Min Value***	***Max Value***	***Basis (Reference)***
Number of Providers	6	6	An assumption for a moderate size of federation
Number of clouds	12	12	Assuming each provider owning two clouds
Number of users in each POIs	250	7 Million	Wide enough range to cover the heterogeneity required for the simulation study
B. Cloud related parameters	***Min Value***	***Max Value***	***Basis (Reference)***
Cloud capacity –	150	500	The assumption for small

CPU cores (#)			cloud providers. wide enough range to cover the heterogeneity required for the simulation study
Cloud capacity – Memory (GB)	600	2000	
Utilization Ratio of clouds (%)	10	70	NRDC Issue Paper (Whitney & Delforge, 2014).
CPU Speed (GHz)	1.67	4.73	Dell Server Specifications (Dell, 2019)
Availability	97%	100%	Gartner (Cloud Harmony) (Gartner Inc. CloudHarmony, 2018)
C. Service request related parameters	***Min Value***	***Max Value***	***Basis (Reference)***
Application nodes (#)	3	9	An assumption made based on the discussion among colleagues regarding the requirements of typical web applications comprising few services and a moderate number of users
Node pairs with data comm. Requirements (#)	2	6	
CPU cores for each node (#)	1	8	
Mem. Size for each node (GB)	2	32	
Service duration (hr)	1	240	Wide enough range to study the effect of heterogeneity in service duration of applications
D. Network latency related parameters	***Min Value***	***Max Value***	***Basis (Reference)***
Intra-cloud (ms)	0	0	Same data center Verizon (Verizon, 2018) Verizon (Verizon, 2018)
Inter-cloud - same region (ms)	30	45	
Inter-cloud - diff. regions (ms)	60	290	
E. Provider	**Min**	**Max**	**Basis (Reference)**

resources related parameters	Value	Value	
VM Instance types in each cloud	14	14	General purpose Amazon EC2 - On demand instance specifications (AWS, 2019).
CPU cores (#)	1	96	
Memory Size (GB)	0.5	976	
VM Instance Price ($/hr)	0.0065	6.672	Carefully calculated values between ±15% of the price of Amazon (AWS) EC2 for the instance type (AWS, 2019).
Availability (%)	97	100	Gartner (Cloud-Harmony) (Gartner Inc. CloudHarmony, 2018)

4.4.2.2 Cloud Specific Parameters

This section provides details of the parameters related to clouds, which includes Cloud Capacity, defined in terms of the number of CPU cores and the Memory Size, CPU speed, and the availability of the data center (see **Table 12** for a summary).

a. *Cloud Capacity, CPU Cores (#), and Memory Size*. At a granular level in the simplest terms, the capacity of a cloud is defined by the number of CPU cores and the memory size it hosts. We set the capacities of each cloud in the range of 150 and 500 for CPU cores and within the range of 600 and 2400 for memory, assuming that it provides sufficient level of heterogeneity for the simulation study.

b. *CPU Speed*. This parameter is used by the service placement algorithm to optimally maximize the processing speed for the customer application. We set the speed in the range of 1.67GHz to 4.73GHz referring to speed of various server processors by Dell (Dell, 2019).

c. *Availability.* This parameter is used by the service placement algorithm to optimally maximize the availability of the customer application. Referring to the statistics from Cloud Harmony (Gartner Inc. CloudHarmony, 2018), a Gartner company, we set the values of various clouds within the range of 97% to 100%.

4.4.2.3 Service Request Related Parameters

This section provides details on the parameters related to service requests initiated by customers (see **Table 13** for a summary). Aside from *Service Duration*, all the below mentioned parameters are specified by the customers in the form of application topology. For this, we maintain a database of four different application topologies. For each of the service requests during the simulation, we choose, at random, one of these application topologies and set the parameter values related to the application topologies. Parameter settings for all the parameters related to a service request are given below. The values for each of the parameters below are set within a wide enough range,

with the assumption made based on the discussion among colleagues for internet applications of moderate scales.

a. *Application nodes (#).* This refers to the number of nodes that constitute the customer application. Application topologies considered for the simulation consists of 3 to 9 nodes.

b. *Node pairs with data comm. Requirements (#).* This parameter refers to the number of node pairs that hold a strong relationship for data communication requirements. It is used by the employed service placement algorithm for optimally minimizing the inter-node network latency. Application topologies considered for the simulation consists of 2 to 6 node pairs.

c. *CPU cores for each node (#).* The number of CPU cores required for each node in the selected application topology varies in the range 1 to 8.

d. *Memory Size for each node (GB).* The size of memory required for each node in the selected application topology varies in the range 2GB to 32 GB.

e. *Service duration (hr).* The service duration for each service request is assumed to be randomly selected values in the range of 1 to 240 hours.

4.4.2.4 Network Latency Related Parameters

Network Latency related parameters are used by the service placement algorithm to optimally minimize the Network Latency measured as Round Trip Time (RTT).

a. *Intra-cloud Network Latency.* This refers to the Network Latency experienced in the communication between two application nodes that are hosted in the same cloud (data center). Assumed to be zero.

b. *Inter-cloud Network Latency (Same Region).* This refers to the Network Latency experienced in the communication between two application nodes that are hosted in different clouds located in the same region. Exact values are considered referring to the statistics provided by Verizon (Verizon, 2018), which lies in the range 30 to 45ms.

c. *Inter-cloud Network Latency (Same Region).* This refers to the Network Latency experienced in the communication between two application nodes that are hosted in different clouds located in different regions. Exact values are considered referring to the statistics provided by Verizon (Verizon, 2018), which takes values in the range 60 to 290ms.

4.4.2.5 Provider Resources (VM Instances) Related Parameters

This section provides details on the parameters related to Virtual Machine (VM) Instances offered by the providers.

a. *VM Instance types.* We consider 14 number of VM instance types with reference to general purpose VM Instance types (t2.xxx and t3.xxx) from Amazon (AWS, 2019).

b. CPU Cores (#). The Number of CPU cores varies between 1 to 96 referring to the instance specification for Amazon EC2 general purpose VM instances (t2.xxx and t3.xxx) (AWS, 2019).

c. Memory Size. Memory Size varies between 0.5 GB and 976 GB referring to the instance specification for Amazon EC2 general purpose VM instances (t2.xxx and t3.xxx) (AWS, 2019).

d. *VM Instance Price.* Prices for various instances are set by referring to the instance specification for Amazon EC2 pricing for general purpose VM instances (t2.xxx and t3.xxx) (AWS, 2019). We vary the price of the VM instance for the simulation within the range of ±7.5% of the base price depending on the CPU speed and the availability of the clouds.

4.5. Result Analysis

We captured the simulation results with a view to getting answers to three questions, namely - i) How can the proposed revenue sharing model encourage cloud providers to join and work within a federation, ii) Does the proposed model always provide better incentives to all the federation members?, and, iii) How does it provide better incentives to federation members compared to benchmark revenue sharing approach.

Analysis of the simulation results regarding each of these questions follows next.

4.5.1 How Can the Proposed Revenue Sharing Scheme Encourage Cloud Providers to Join and Work in a Federation?

In order to answer the first question, we performed an analysis of the results from both social benefits as well as individual benefits perspectives. Our assumption is that a revenue sharing model for a cloud federation that provides individual incentives, in addition to social benefits, will be capable to encourage individual cloud providers to join and cooperate in the federation.

4.5.1.1 How Does the Proposed Revenue Sharing Scheme Perform from the Social Benefits Point of View?

In order to see how the proposed revenue sharing model can encourage cloud providers to join and work within a federation, we present the result analysis for two different kinds of the federation. Firstly, a Symmetric Federation, which is a federation of providers having similarity in capacity and market share, and secondly, Asymmetric Federation, which is a federation of providers having dissimilarity in capacity and market share. The resource utilization ratio for the federation compared to the cumulative of all members in the individual operation for symmetric federation is shown in **Figure 14**.

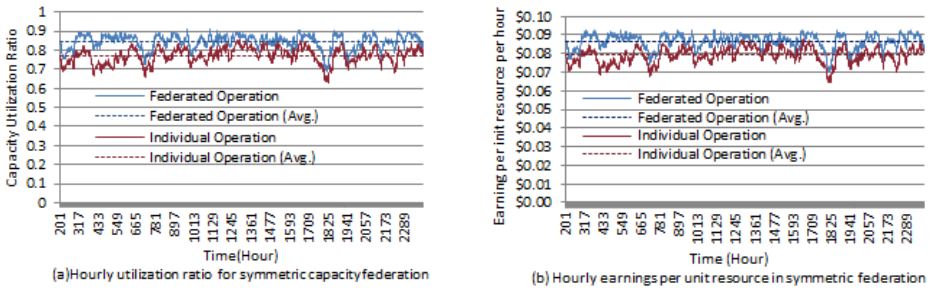


Figure 14: Federation level resource utilization ratio and hourly earnings per unit resource in case of a federation of providers with symmetry in capacity and market share

For this, we set the parameters for all the providers with similar values. Capacities of each provider are set to be 300 CPU cores and 1200GB of memory, the market share was set to be equal creating 500 service requests for over a period of 3 months.

The graph shows the hourly average utilization ratio of the provider capacity as a result of serving the service requests. It can be observed from **Figure 14** that the cloud federation operated as per the proposed model increases overall resource utilization ratio from 0.76 to 0.85, an increase of almost 12%. Similarly, the average hourly earnings per unit resource increased from $0.079 to $0.0865, an increase of 9.5%. The increase does not seem to be significant in this particular scenario. This is because the parameter settings for capacity and request rate are in such a proportion that the average utilization ratio of provider capacities before joining the federation is at 0.76, which is already very high value.

Similarly, we observed the utilization and earnings of the asymmetric federation, too. For this, we considered six providers each one different from each other in terms of their capacities and market share (**Table 13**).

Table 13: Parameter settings for the provider characteristics for asymmetric federation

Provider	Provider Characteristics		Capacity		Number of Service Requests
	Capacity	Market Share	CPU	Memory	
Pr1	low	low	150	600	100
Pr2	high	low	300	1200	100
Pr3	low	moderate	150	600	300
Pr4	high	moderate	300	1200	300
Pr5	low	high	150	600	500
Pr6	high	high	300	1200	500

The resource utilization ratio for the federation compared to the cumulative of all members in the individual operation for the asymmetric federation is shown in **Figure 15**. The characteristics of providers in the asymmetric federation are considered as per **Table 13**. As seen from the **Figure 15** (a), the proposed revenue sharing model can improve the federation level resource utilization ratio from 0.56 to 0.74 gaining 32% increase, yielding a gain of 30% increase in average hourly earnings (from $0.058 in individual operation to $0.0755 in federated operation (see **Figure 15** (b))). This increase in the utilization ratio and hence the earning is due to the ability of the federation to serve additional requests by some of the federation members that would

otherwise be dropped due to the provider receiving the request lacking sufficient resources available at that point of time in case of the individual operation.

This shows that whatever be the characteristics of the potential cloud providers, the federation powered by the proposed contribution based revenue sharing model will provide better earnings per unit resource considering the overall federation allowing for the maximization of overall social benefits. This means it will provide a better return in investment for whatever is the structure of the federation.

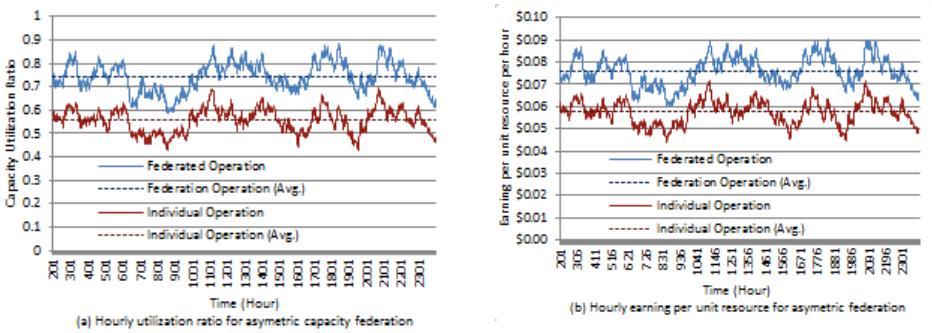


Figure 15: Federation level resource utilization ratio and hourly earnings per unit resource in case of federation having providers with asymmetry in capacity and market share

4.5.1.2 How Does the Proposed Revenue Sharing Scheme Perform from the Individual Benefits Point of View?

We performed the analysis of individual benefits for the case of Asymmetric Federation, which is a federation of providers having dissimilarity in capacity and market share. In order to see how the proposed revenue sharing model encourages cloud providers of

different characteristics, in term of capacity and market shares, to join the federation, we studied the characteristics of revenue inflows for each of these providers from individual benefits perspectives.

For this, we considered six providers each one different from each other in terms of their capacities and market share (See **Table 13**). We compared the revenue inflows generated by the proposed revenue sharing model in federated operation with that that would earn for the same capacity and market share if they worked individually. **Figure 16** shows the comparison of earnings per unit resource per hour in federated operation to that in individual operation (only for four providers located in the extreme positions with respect to capacity and market). Related statistics are presented in **Table 14**.

Table 14: Statistics for capacity utilization and hourly earnings for asymmetric federation compared to their respective individual operation

Provider	Utilization Ratio				Earnings (return on Investment)			
	Federated Op.		Individual Op.		Federated Op.		Individual Op.	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Pr1	0.6955	0.1084	0.3445	0.1312	0.0818	0.0125	0.0355	0.0135
Pr2	0.795	0.0782	0.1767	0.088	0.0642	0.0067	0.0176	0.0087
Pr3	0.8003	0.0711	0.7446	0.1031	0.0984	0.0118	0.0777	0.0105
Pr4	0.6582	0.1156	0.6024	0.1572	0.0655	0.0082	0.0641	0.0167
Pr5	0.8274	0.0545	0.8054	0.0795	0.1035	0.0093	0.0789	0.0079
Pr6	0.7297	0.0942	0.7803	0.0864	0.0706	0.0063	0.081	0.009
Average	**0.751017**	**0.087**	**0.57565**	**0.107567**	**0.080667**	**0.009133**	**0.059133**	**0.01105**

The benefit of joining the federation for Provider Pr1 is apparent from **Figure 16**(a). In case of provider Pr1, which possess the low capacity and low market share, increases its average hourly capacity utilization ratio from 0.3445 to 0.6955 (see **Table 14**), an increase of more than 100%. Similarly, it increases the average hourly earnings from $0.0355 to $0.0818 (see **Table 14**), an increase of more than 130%. This gain of 130% in hourly earnings is, firstly, due to its increase in resource utilization ratio by 100%, and secondly, by being able to earn certain fraction of the revenue in federated operation from the requests incoming to this provider but would be dropped due to inadequate resources at the time of receiving the requests, in case of individual operation. It is important to note that, due to having capacity at the lower side, the probability of received requests being dropped for lack of sufficient resources at the time of receiving the request, is higher in case of this provider when worked individually. Hence these providers seem to benefit significantly by joining the federation with other providers having higher capacity and market share.

The benefit is more apparent in the case of Provider Pr2 (see **Figure 16** (b)), which possess high capacity but low market share. The resource utilization ratio, which is at 0.1767 in case of individual operation due to low market share making resources idle most of the time, is increased to 0.795 in federated operation, which is an increase of almost 350%. Similarly, the average hourly earnings per unit resource increase to $0.0642 in federated operation yielding an increase of

almost 265% compared to individual operation providing $0.0176. Due to the low market share of its own, hourly earnings for this provider comes, mostly, by serving the requests that came into the federation through other providers. Hence, the rate of hourly revenue increase is lower than the rate of increase in hourly utilization ratio.

Provider Pr5, having low capacity and high market share, already has over 80% of capacity utilization (**Figure 16** (e.1)). Hence, by joining a federation it can only marginally increase the resource utilization by 2% (an increase from 0.8054 in individual operation to 0.8274 in federated operation). However, since it has high market share, it can get the incoming service requests from customers to be served by other federation members, and hence, can gain a significant increase in earnings per unit resource, i.e. by over 31% from $0.0789 in individual operation to $0.1035 in federated operation (**Figure 16** (e.2)).

However, unlike all other providers in the federation, provider Pr6, do not gain direct individual benefit by joining in the federation. As seen from **Figure 16** (f) and **Table 14**, the utilization decreases from 0.7803 to 0.7297, a decrease of almost 7%. And, the earnings per unit resource decreases from $0.081 to $0.0706, a decrease of almost 15%. Having both capacity and market share high compared to other members in the federation, it does not possess the potential to earn additional revenue that would otherwise be lost for lack of sufficient resources to fulfill the request.

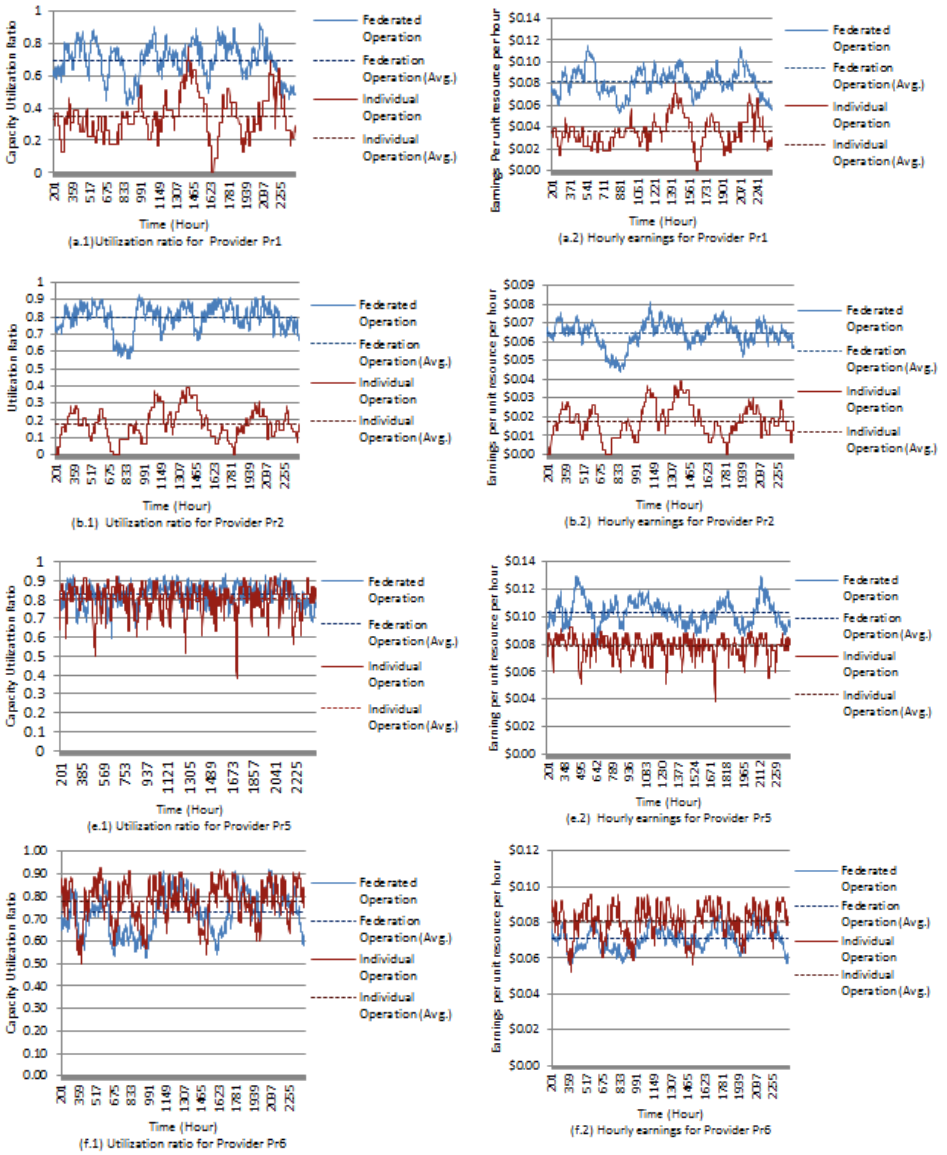


Figure 16: Comparison of capacity utilization and hourly earnings per unit resource per hour for providers in asymmetric federation compared to their respective individual operation

Rather, by joining the federation, its earnings decreased because, by virtue of the consumer preferences and service placement algorithms, some of the requests that it could serve with its own resources in

individual operation, now will be served by other members of the federation, providing this it only a fraction of the revenue earned from this request. Hence, other things aside, a provider with a sufficiently large capacity and market share compared to the other members of the federation, will not directly benefit from joining the federation. However, the indirect benefits that it can get cannot be understated. Such indirect benefits include – geographic presence, varied service (cost, and QoS), and need of resources for spontaneous spikes in demand and utilization of idle resources to some extent, and also the potential of operating with lower capacity level.

Results in Table 14 also show that the average utilization over a period of time is higher (0.75 compared to 0.57) while the standard deviation of the utilization is lower (0.087 compared to 0.10) in case of federated operation in comparison to the individual operation. This suggests that the providers can operate at lower capacity with more assurance that the SLA violations do not occur for the demand at the same level. From the same table, we also see that the average earnings over a period of time are higher ($0.08/hr compared to $0.06/hr per unit resource) and standard deviation for the earnings is lower (0.009 compared to 0.011) with the federated operation. This suggests that the providers can get higher and consistent revenue stream over a period of time by working in the federation.

4.5.2 Does the Proposed Model Always Enable the Federation to Outperform Individual Operation? If Not, What is the Departure Point?

In order to study if the proposed model enables the federation to outperform individual operation in any circumstances, we compared the earning per unit resource with respect to demand-capacity ratio generated from the federated operation compared to that generated from the individual operation.

For this, we considered two federations each comprising of six providers. Members in the first federation are of relatively smaller capacities, each with 150 CPU cores and 600 GB of memory. While, the members in the second federation are of double the capacities of the first one, each with 300 CPU cores and 1200 GB memory.

For each of the federations, 500 service requests of random durations are generated. The requests are served in two different ways - i) individual operation where the requests are served as if the members of the federation worked individually without federation, and ii) federated operation, where the requests are served as if the members worked in the federation powered by the proposed service placement algorithm and revenue sharing algorithm. In both of the cases, for every hour, the demand capacity ratio, and earning per unit resource is calculated and accounted. The accounted demand-capacity ratio and earning per unit resource for every hour are grouped into different demand-capacity

ratio classes (interval of 0.05), and the average earning per unit resource per hour is calculated for each of these classes of demand capacity ratio.

Results are given in **Figure 17**. **Figure 17** (a) shows the comparison of how the rate of earnings increase with respect to demand capacity for a federation with smaller capacity (CPU: 150. memory: 600) compared to the same federation members working individually. And, **Figure 17** (b) shows the same for a federation with larger capacity (CPU: 300. memory: 1200). In either of the cases, it is clear that the hourly earnings per unit resource increase with the increase in demand-capacity ratio until a certain point and remains constant both in case of federated as well as individual operation. The rates of earnings are the same for federated and individual operations until a point of departure (Demand capacity ratio).

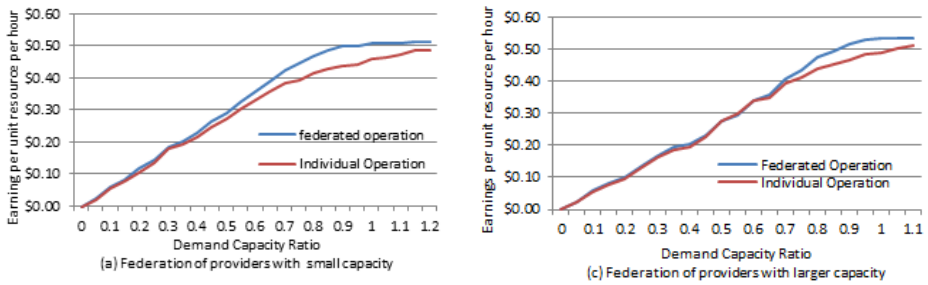


Figure 17: Departure point for benefits in small capacity and large capacity federation

From **Figure 17** (a), it is apparent that this point of departure in case of the smaller federation is at a demand-capacity ratio at around 0.3. While on the other hand, for the federation of larger capacity, this

departure point comes when the capacity demand ratio reaches around 65%.

This clearly shows that it is beneficial to work in the federation when either the capacity is lower or when the demand capacity ratio is higher. More specifically, small providers with demand-capacity ratio beyond 30% should consider joining a federation. But providers with high capacity and lower demand-capacity ratio should only consider joining the federation only if the other potential members of the federation can potentially contribute to increasing the overall demand-capacity ratio of the federation. Otherwise, for these large providers, the administrative overhead (such as financial settlement, Federation level SLA management) involved with the federated operation may outweigh the marginal benefits.

4.5.3 How Does it Perform in terms of Providing Incentives to Federation Members in Comparison to the Benchmark Revenue Sharing Approach?

We also performed the comparative study on how the proposed contribution based revenue sharing approach performs compared to the benchmark, i.e. - participatory approach to revenue sharing for cooperative federation (Z. Lu et al., 2012). For this, we consider a federation of six providers with the parameter settings as per the asymmetric federation considered earlier (see **Table 13**). The comparison of the hourly allocation of revenue in both approaches is

depicted in **Figure 18** (only for four providers located in the extreme positions with respect to capacity and market).

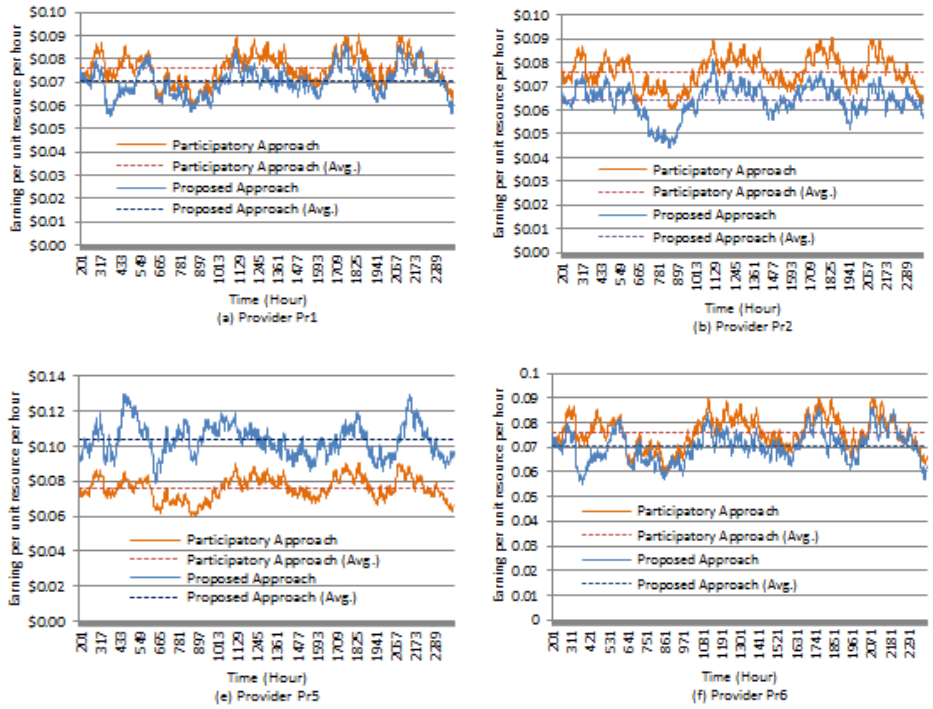


Figure 18: Comparison of revenue shares from the proposed approach to that from the benchmark approach to members of an asymmetric federation

From **Figure 18**, it is evident that, in a participatory approach, the average hourly earnings per unit resource are the same for all the providers with an average just below $0.08. However, in the case of a proposed contribution-based approach, it varies for different providers according to their characteristics, capacity, market share and actual utilization of their capacities. In this case, the average hourly earnings per unit resource for different providers vary between the ranges $0.0642 to $0.1035. The resource utilization ratio in the proposed

approach is governed by the capacity of the provider and consumer preferences for service placement, as stated earlier. As apparent from the figure, three among the four providers, namely Pr1, Pr2, and Pr6 get better revenue share by the benchmark participatory approach while the remaining one, namely Pr5 receive better revenue share by the proposed contribution based revenue sharing approach compared to the other. The fact that a larger number of providers receive more revenue shares from the benchmark participatory approach does not mean that it outperforms the proposed contribution-based approach. If we observe the curves for the participatory approach, the average hourly earnings per unit resource remain same for all the providers despite their differences in capacity and market share. In this case, providers receive the revenue share in proportion to the capacity without considering the actual work done in service provisioning. This phenomenon does not incentivize the utilization; thereby do not encourage the providers to excel in terms of cost and QoS parameters. On the other hand, as seen in **Figure 18**, the curve for the proposed approach does not remain the same for all the providers. Instead, it shows variations indicating that the proposed approach incentivizes the actual resource utilization in service provisioning and incoming service requests (i.e.- market share) of the providers. Thus, it provides the opportunity of competition within the cooperation. Never the less, both of the approaches provide other indirect benefits of joining the federation (increased geo-

presence, service variety, resource scalability, etc.) to the federation members.

4.6 Conclusion

Revenue sharing is a prominent economic issue challenging cloud federations for the formation and sustaining its operation. In this chapter, we proposed a contribution based revenue sharing scheme to address this challenge. Our revenue sharing scheme makes use of the multi-criteria optimized service placement algorithm to schedule the service requests to federation members. And, based on how the requests are served, the proposed scheme allocates the payoff generated to each of the federation members according to the contributions made by each of the federation members in generating the revenue.

Unlike existing approach to cooperative cloud federation (Aryal & Altmann, 2017; Coronado & Altmann, 2017; Kaewpuang et al., 2013; Mashayekhy et al., 2015), our scheme considered not only the resources provisioned for the request but also the requests brought in to the federation by a provider for evaluating their contributions. We made use of a solution concept in coalitional game theory, namely, *Shapley Value* to allocate the payoffs among the members based on their contribution.

By performing the comparative study, we demonstrated that, despite the structure of the federation, be it symmetric or asymmetric, the federation can improve the overall resource utilization ratio, and

thereby increase the earnings per unit resource by over 30%. This means the providers can achieve a better return on the investment that they make in cloud infrastructure if they work in a federation. Results also demonstrated that the variation of the earnings per unit resource over a period of time is less (standard deviation of 0.009 compared to 0.011) for all the providers with different capacities and market shares. This means that, with the proposed revenue sharing scheme, providers can get the better assurance of the return on their investment. Similarly, we showed that the variation in resource utilization over a period of time is lower (standard deviation of 0.087 compared to 0.010). This means that providers can achieve higher scalability capacity compared to individual operation even by maintaining their infrastructure capacity at a much lower level. This phenomenon also reinforces the concept of a better return on investment and better assurance of a higher and more consistent revenue stream.

With an analysis of the comparison of providers' earnings per unit resource in the asymmetric federation, we highlighted how the providers with different characteristics (capacity and market share) can be benefitted from the federation. The results showed that providers with high capacity and low demand tend to benefit the most from the federation (up to 265% as shown from the simulation results). Next, the provider with low capacity and low demand comes in second place with a 130% increase in revenue per unit resource.

Providers with low capacity and high demand also benefit from the federation with an increase in the earnings of up to 31%. In this way, we see that the proposed scheme provides benefits for providers with the disparity between their capacity and demand or for providers with low capacity irrespective of the demand. The reason for the benefit in the earnings for providers with disparity in capacity and demand is because of the additional earnings that come from serving the requests coming from other providers in case of providers with higher capacity but lower demand, and additional earnings from the requests they received and served from other providers which would have to be dropped for lack of resources if they worked individually in case of providers with lower capacity but higher demand. Moreover, among the providers with the disparity in capacity and demand, too, the providers with lower capacity seem to be benefitted more as their probability of dropping the requests for lack of resources would be higher if they worked individually.

The results showed that the providers with sufficiently high capacity and high demand with similarity in both demand and capacity did not seem to benefit from joining the federation. This is because the probability of increasing the utilization ratio for such providers is low. Rather they tend to lose some revenue (a decrease of up to 7% as shown by the simulation results) because of the distribution of the requests to the federation members as a result of the service placement algorithm. This result supports the arguments made by existing research

(Varghese & Buyya, 2018), which states that providers with higher capacity and resources spread over various geographic locations are less likely to join the federation. However, for providers with larger capacity and substantive market share too, the importance of other benefits like being capable of offering services with varied features like price, availability, speedy processors, specific geographic location, etc. cannot be understated.

This result will provide guidance to cloud providers in deciding whether or not to join a certain federation or choose a federation that helps maximize their benefits depending on where they stand in relation to that of other potential federation members in terms of capacity and market share.

With the comparative analysis with various demand capacity ratios and provider capacities, we demonstrated that it is not the case that providers can always improve resource utilization by joining a federation; especially, when the demand capacity of the overall federation is low and the provider capacity is high. Results showed that

Federation of providers with smaller capacities shows benefits with an increase in the earnings starting from a lower threshold level of demand to capacity ratio, while the federation of larger capacity starts to show the benefits starting at a higher threshold level of demand to capacity ratio. This result also provides support for the cloud providers in

making strategic decisions regarding joining the federation based on their relative position in the federation.

By comparing with the benchmark approach (participatory approach) the earnings per unit resource of members of the asymmetric federation, we showed that the proposed scheme incentivizes for the actual work performed in serving the request, where the opportunity of performing the work is increased based on cost and QoS parameters. This encourages providers to excel in terms of cost and QoS parameters. Thus, it provides the opportunity of competition within the cooperation.

Thus, the proposed revenue sharing scheme addresses the problem of revenue sharing for a co-operative cloud federation. It provides better incentives in most of the cases. Relatively smaller providers who may lack enough resources but well at marketing strengths can bring customers to the federation and get benefitted. Providers with high resource capacity but lacking marketing power and market share can maximize the utilization of their otherwise idle resources and increase their earning per unit resource. Providers who can maintain good data center availability or servers with high processing speeds can get benefitted with the extra charge that they can get from premium customers. The study provides guidance for the cloud providers in strategically deciding whether to join the cloud federation based on their relative position in the federation. This study also contributes to

the research community working on the topic of revenue sharing to explore the application of Shapley Value as a solution concept.

Chapter 5. Conclusion

In this chapter, we provide a summary of the thesis work, its implications for the research community and industry, the limitations of the work, and suggestions for future research in this topic.

5.1 Summary

A large body of literature has considered Cloud federation as a way to address the existing limitations of small cloud providers and gain competitiveness in the market. However, no cloud federations seem to be operating in the commercial market. Not having clearly defined business relationships that govern the sharing of resource and revenue among the participants has been identified as factors hindering the formation of cloud federations despite acknowledged promises.

In this context, aiming to fill this gap, we presented two different algorithms that provide rules and methods governing the act for sharing of resources and revenue among the federation members. We presented the first algorithm, namely the Service Placement Algorithm in Chapter 3. The algorithm governs the act of resource sharing in cloud federation with an aim to maximize the benefits of the federation. For this, through extensive literature survey, we identified four criteria, namely - financial cost, processing speed, network latency, and availability, as reasonable and measurable criteria that are important for

service placement decision making in a federated cloud computing environment. Employing those identified criteria, we developed a multi-criteria service placement algorithm by drawing knowledge from two different approaches, namely Analytic Hierarchy Process (AHP) (T. L. Saaty, 2008) and Fast and Elitist Non-dominated Sorting Genetic Algorithm (NSGAII) (Deb et al., 2002). The algorithm takes consumer preference as a pairwise comparison of decision variables, converts them into respective weights using the AHP method. It performs simultaneous optimization of multiple criteria employing NSGAII method and finds a set of Pareto optimal solutions with the optimization process aiming at minimizing cost and latency while maximizing the computing capacity and system availability. From the set, it then selects one that is most appropriate according to their overall fitness. The overall fitness of the plans is evaluated as a function of normalized values of the objective functions and their respective weights evaluated earlier from the AHP method.

The results showed that the algorithm can effectively find the appropriate service placement plan by making optimal tradeoffs as per the consumer preferences within 225 iterations, which is a reasonable number. The results of the comparison of solutions from a proposed algorithm with that for the benchmark in the objective space demonstrated that the solutions generated by proposed approach provided better values with respect to two criteria, same values for one criterion and poorer values for one criterion. The simulation result also

showed that the proposed algorithm outperforms benchmark algorithm (weighted sum) with respect to standard metrics. Result w.r.t. Generational Distance (GD) metric (Veldhuizen, 1999) showed that the proposed algorithm provides better convergence compared to the benchmark (with 0.95 from the proposed algorithm compared to 0.98 from the benchmark). Results with respect to Spacing (Sp) metric (Riquelme et al., 2015; Schott, 1995) showed that the proposed algorithm is better in terms of diversity (with 0.086 from the proposed compared to 0.107 from the benchmark). Results w.r.t. the Set Coverage (C) metric (Hiroyasu et al., 1999) shows that the proposed approach is better than the benchmark in both convergence and diversity (with 0.18 from the proposed algorithm compared to 0 from the benchmark). These results are in line with the arguments made in existing research work (Deb et al., 2002).

Now, the next problem was associated with revenue sharing among the federation members. For this, we proposed a contribution based revenue sharing scheme in chapter 4. The revenue sharing scheme made use of the earlier proposed multi-criteria optimized service placement algorithm, which is presented in Chapter 3, to schedule the service requests to federation members. And, based on the requests served, the generated payoff share is allocated to each of the federation members based on the contribution that they make in generating the revenue. Infrastructure capacity and market share of the providers have been considered for the evaluation of members' contribution.

Implicitly, it also considers the demand for the service characteristics the provider offers. The market strength is assessed from the revenue value of the requests brought in to the federation and capacity is assessed from the actual amount of resource provisioned in serving the customer requests. And, the allocation of collective payoffs in proportion to their contribution is done by employing *Shapley Value* (Shapley, 1953), a solution concept in coalitional game theory.

Comparative study through simulation shows that despite the structure of the federation, be it symmetric or asymmetric, the federation, which is enabled and operated as per the proposed Service Placement Algorithm and Contribution Based Revenue Sharing Scheme, can improve/maximize social benefits by increasing the overall utilization of resources and return on investment by over 30%. This means the providers can achieve a better return on their investment that they make in cloud infrastructure if they work in a federation. Results also demonstrated that the variation of the earnings per unit resource over a period of time is less (standard deviation of 0.009 compared to 0.011) for all the providers with different capacities and market shares. This suggests that the proposed revenue sharing scheme can ensure a better return on their investment. Similarly, we showed that the variation in resource utilization over a period of time is lower (standard deviation of 0.0087 compared to 0.010), which suggests for a more consistent revenue stream over a period of time.

Evaluation of the proposed scheme through simulation reveals that the scheme is beneficial for both symmetric as well as an asymmetric federation from the social benefits point of view. There is a cost associated with joining the federation, such as for the management of federation level agreement and financial settlements recurring at the end of a specified time period (Toosi et al., 2014). Due to this cost, it may not always be beneficial to join the federation. Hence it is desired to have suggestions for when it is and it is not beneficial to join the federation. To analyze this, we conducted resource utilization and revenue distribution in the case of the symmetric and asymmetric federation and did important observations. The observation shows that whether or not it is beneficial to join a federation depends on the demand capacity ratio and the capacity of the federation members. The same demand-capacity ratio has a different return on investments with different capacities. It shows that marginal benefits of joining the federation depend on these factors and hence providers should consider these factors. Our simulation result showed that for relatively larger providers, the marginal benefits of the federation starts to show only when the Demand to capacity ratio is over 65% when relatively smaller providers could see the benefits when the demand capacity ratio crosses just 30% mark. However, other benefits of the federation such as expanded geographic footprint and the ability to offer service variety and quality still prevail irrespective of these points, and hence cannot be understated.

Next, we also compared the incentive model with the benchmark participatory approach, which clearly indicated that the proposed model provides a better incentive system and enable competition within the cooperation. This will empower the federation to be competitive with respect to other federation and other hyper-scale providers but at the same time provide space for the federation members for competition in price and service quality.

5.2 Implications

This thesis work provides important academic and managerial implications.

5.2.1 Managerial Implications

The proposed service placement decision model and the algorithm attempts to address the problem of service deployment in a multi-provider federated cloud environment. It does not only selects the cloud for service deployment, rather specifies at a more granular level, for each application service node, the selected VM type hosted at a particular data center of a cloud provider, where it should be deployed. This is a real problem to be solved in the industry which has not been addressed appropriately.

The algorithm is beneficial to various stakeholders of the cloud service market, viz. cloud consumer, cloud providers, a federation of cloud providers, and cloud brokers. It allows a cloud federation or cloud broker to deploy its customer application in an optimal way, where

individual consumers define what the ‘optimal way’ is for the deployment of their applications. In addition to the initial deployment, the algorithm is applicable throughout the application lifecycle as it supports run-time adaptation of applications by providing better placement plans, such as reduction in average network latency experienced by consumers by exploiting the information about the change in application footprint and migrating service to locations that are in close proximity to majority of the users during the lifetime once the consumer preferences are captured.

Thus, the algorithm enables the cloud federation operator or a cloud-broker to offer customized placement services with additional what-if analysis, which distinguish them in the competitive cloud service market and help them retain existing and attract new customers (Wei Wang et al., 2012). For example, the degree of system availability that could be achieved if the budget limit is increased by a certain amount, or how cost can be lowered if the consumer is still satisfied with a reduction in system availability by a certain value. Similarly, consumers can be presented with an idea of how much they can benefit with regards to network latency if they can compromise on some degree of system availability or vice versa. There could be a number of other what if cases, too. This enables the cloud federation or a cloud broker to offer service variety, which has been found to be helpful in extending the market (Wei Wang et al., 2012).

It not only benefits a cloud federation or a broker with the expanded market, but also an application provider allowing them to have their application deployed with optimal QoS level that is within their budget limit. It also allows for better QoE by the end users of the application with reduced application response time, which have been found to increase the engagement time, providing an additional advantage to the application provider (Arapakis et al., 2014).

In the proposed system model, the cloud providers are inquired by the resource broker component of the federation platform, for resource availability, configuration, and price of VMs, for each incoming service request. This model provides flexibility to the member providers for adjusting their prices depending on their workload and other factors.

The usefulness and application of the proposed Service Placement Algorithm become more pronounced in the coming days as more and more applications are being developed or converted into microservice architectures (Balalaie et al., 2015). An application built on micro-service architecture, is made up of a number of independent and loosely coupled micro-services that involve minimal data communication, can be better benefitted by their distribution on cloud resources across the federation (Buyya et al., 2009). In this context, the algorithm allows for the selection of resources considering the specifications at a more granular level and optimize for a specific component of the application.

The simulation results with respect to the Revenue Sharing Scheme provides confidence to potential providers in deciding to work in a federation with demonstrated better return on their investment while working in the federation.

The result also shows that the variation of the earnings per unit resource over a period of time is less for all the providers with various capacities and market shares. This means that, with the proposed revenue sharing scheme, providers can get the better assurance of the return on their investment.

The result also shows that the variation in resource utilization over a period of time is lower. This means that providers can achieve higher scalability capacity compared to individual operation even by maintaining their infrastructure capacity at a much lower level. This phenomenon also reinforces the concept of a better return on investment and hence potential cloud providers can be well assured of the better return on the investment by joining the federation.

With the comparative analysis with various demand capacity ratios and provider capacities, we demonstrated that it is not always the case that providers can always improve resource utilization by joining a federation; especially, when the demand capacity ratio of the overall federation is low and the provider capacity is high. This result also provides support for the cloud providers in deciding the right federation to join based on their relative position to other providers.

By comparing with the benchmark approach (participatory approach) the earnings per unit resource of members of the asymmetric federation, we showed that the proposed scheme incentivizes for the actual work performed in serving the request, where the opportunity of performing the work is increased based on cost and QoS parameters. This encourages providers to excel in terms of cost and QoS parameters. Thus, prospective cloud providers who are constantly seeking to excel by improving service quality and reducing cost are suggested not to be discouraged to join the federation as they can continue doing so and get appropriately incentivized for such efforts.

With an analysis of the comparison of providers' earnings per unit resource in the asymmetric federation, we highlighted how the providers with different characteristics with respect to capacity and market share can be differently benefitted from the federation. Prospective cloud providers are suggested that by working in the federation enabled with the proposed Service Placement Algorithm and Revenue Sharing Scheme, they can earn more than they would be working individually despite their characteristics such as capacity, market strength, and offered service quality. Relatively smaller providers who may lack enough resources but well at marketings can bring customers to the federation and get benefitted. Providers with high resource capacity but lacking marketing power and market share can maximize the utilization of their otherwise idle resources and increase their earning per unit resource. Providers who can maintain

good data center availability or servers with high processing speeds can get benefitted with the extra charge that they can get from premium customers.

From the simulation results involving asymmetric members of the federation, it is observed that the proposed scheme may not be beneficial for larger providers with a relatively large capacity as well as higher demand-capacity ratios. For providers with such characteristics, it is suggested that if their only intention of joining a particular federation is an increase in earnings per unit resource, then they may not get expected a gain in earnings per unit resource. For such providers, too, other benefits of the federation such as the being capable of offering services with varied features like price, availability, application response time, and the need to meet specific regional requirements by joining a federation cannot be understated. As suggested by Varghese & Buyya (2018), if such providers also have multiple geographic presences, then they are suggested that the benefits that they receive by joining the federation are only marginal.

5.2.1 Academic Implications

This thesis work presents a new perspective on how cloud federation can operate in a competitive co-operative setting that requires the cooperation of the federation members while at the same time provides them the opportunity to excel on their own in terms of service quality and cost and get incentivized for these efforts. The discussion and

arguments in this line included in this thesis work will potentially trigger a fresh discussion within the cloud research community.

This thesis work demonstrates that by augmenting with Multi-Objective Optimization (MOO) algorithms how AHP can, still, be employed to solve Multi-Criteria Decision Making (MCDM) problem with solution space so large that make the search for the best solution impossible using the brute force approach.

It also contributes to the knowledge base in Multi-Objective Optimization domain by suggesting that the reduction of the search space of solutions by parallel optimization of multiple objectives before the application of objective weights can yield better results in a multi-objective optimization problem that requires a single final solution.

Similarly, it demonstrates how the consumer preference for selection decision can be encoded in run time for a search problem with the application of AHP technique.

It contributes to the research community working on the hot topic of revenue sharing problem in various domains that it is worthwhile to explore the possibility of the use of coalitional game theory, especially the Shapley Value as a potential solution concept

5.3 Limitations

Traces of the service requests of IaaS cloud data centers are considered strictly confidential and hence are not available in the public domain

(Toosi, 2014). Therefore, the evaluation of the proposed algorithm is based on the emulation of service requests with reference to suggestion in the existing literature and findings from discussion among the colleagues possessing related experiences.

5.4 Suggestions for Further Research

As future research, two directions are foreseen, namely, implementation and advancement. Taking the first direction, an investigation into different aspects related to the implementation of the proposed algorithms in the production environment may be carried out. For implementation, it is necessary to investigate three different aspects - legal & administrative, technical, and financial. From a legal and administrative standpoint, it is necessary to have agreements for resource sharing, revenue sharing scheme. In addition, there should be agreement about the rules for the separation of liabilities and responsibilities as the service quality are determined collectively. Proper conflict resolution mechanism should be in place to deal with such potential issues.

From a technical standpoint, for a federation to operate as proposed by this thesis, it should be facilitated by a federation platform, such as BASMATI (Altmann et al., 2017). Such a platform may be operated by the federation or may be outsourced to be operated by a third party. The proposed algorithms need to be integrated into the federated platform and linked to its accounting and billing components. In addition,

coordinating agents should be installed in every member providers' clouds to monitor the resource availability and report to the federation platform.

From a financial standpoint, it is necessary to investigate the cost of designing, operating, and maintaining system based on the proposed model. Also, the proposed algorithm provides the rules and methods for revenue sharing, however, the administrative issue of revenue settlement as to when and how this takes place should be investigated.

Detailed investigations into these legal & administrative, technical, and financial aspects for the real implementation in the production environment make good research in this direction.

The second direction is related to the advancement of the models. This thesis work can be extended by investigating strategies and methods for run-time adaptation of application. This may be achieved with dynamic optimization of the service placement plan where the application topology and hence the number of VMs are changed in runtime according to the workload due to, for instance, the change in the number of users. This, however, is a complex process and moving VMs from one cloud to the other involves significant cost. An investigation into the technical and financial aspects for run-time adaptation of application, thus, could be a good research topic. Further, the model may be extended to include user prediction models for predicting users in major Points of Interests (POIs) based on machine learning

techniques once the application has been used for a while and enough dataset regarding the application use scenario is generated. With respect to the revenue sharing scheme, further investigation into the accommodation of different pricing policies and their optimization could be another good research topic. Investigations into a universal pricing model that can help federation members to strategically decide on the pricing model would make another topic of interest for further research.

References

Abawajy, J. (2009). Determining service trustworthiness in intercloud computing environments. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on* (pp. 784–788). IEEE.

Accela. (2016). Tail Latency Study. Retrieved May 29, 2019, from http://accelazh.github.io/storage/Tail-Latency-Study

Al-athwari, B., & Altmann, J. (2015). Utility-Based Smartphone Energy Consumption Optimization for Cloud-Based and On-Device Application Uses. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 164–175). Springer.

Altmann, J., Al-Athwari, B., Carlini, E., Coppola, M., Dazzi, P., Ferrer, A. J., … Pages, E. (2017). BASMATI: An Architecture for Managing Cloud and Edge Resources for Mobile Users. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 56–66). Springer.

Altmann, J., Bañares, J. Á., & Petri, I. (2018). Economics of Computing Services: A literature survey about technologies for an economy of fungible cloud services. Elsevier.

Altmann, J., & Kashef, M. M. (2014). Cost model based service placement in federated hybrid clouds. *Future Generation Computer Systems*, *41*, 79–90. https://doi.org/https://doi.org/10.1016/j.future.2014.08.014

Amato, A., Liccardo, L., Rak, M., & Venticinque, S. (2012). Sla negotiation and brokering for sky computing.

Amazon. (2019). Amazon EC2 Spot Instances. Retrieved February 3, 2019, from https://aws.amazon.com/ec2/spot/

Amigo, I., Belzarena, P., Larroca, F., & Vaton, S. (2011). Network bandwidth allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations. In *International Workshop on*

Internet Charging and QoS Technologies (pp. 50–62). Springer.

Aoyama, T., & Sakai, H. (2011). Inter-cloud-computing. *Wirtschaftsinformatik*, *53*(3), 171–175.

Arapakis, I., Bai, X., & Cambazoglu, B. B. (2014). Impact of response latency on user behavior in web search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 103–112). ACM.

Aryal, R. G., & Altmann, J. (2017). Fairness in Revenue Sharing for Stable Cloud Federations. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 219–232). Springer.

Aryal, R. G., & Altmann, J. (2018). Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization AI algorithm. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on* (pp. 147–154). IEEE.

Assis, M. R. M., & Bittencourt, L. F. (2016). A survey on cloud federation architectures: Identifying functional and non-functional properties. *Journal of Network and Computer Applications*. https://doi.org/10.1016/j.jnca.2016.06.014

Ataie, E., Entezari-Maleki, R., Etesami, S. E., Egger, B., Ardagna, D., & Movaghar, A. (2018). Power-aware performance analysis of self-adaptive resource management in IaaS clouds. *Future Generation Computer Systems*, *86*, 134–144.

AWS. (2019). Amazon EC2 Instance Pricing. Retrieved January 4, 2019, from http://aws.amazon.com/ec2/pricing/

AWS, A. (2018). Regions and Availability Zones. Retrieved November 12, 2018, from http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Babu, K. R. R., & Samuel, P. (2014). Virtual machine placement for improved quality in IaaS cloud. In *Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on* (pp. 190–194). IEEE.

Baker, T., Aldawsari, B., Asim, M., Tawfik, H., Maamar, Z., & Buyya, R. (2018). Cloud-SEnergy: A Bin-Packing Based Multi-Cloud Service Broker for Energy Efficient Composition and Execution of Data-intensive Applications. *Sustainable Computing: Informatics and Systems*.

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2015). Migrating to cloud-native architectures using microservices: an experience report. In *European Conference on Service-Oriented and Cloud Computing* (pp. 201–215). Springer.

Bañares, J. Á., & Altmann, J. (2018). Economics behind ICT infrastructure management. *Electronic Markets*, *28*(1), 7–9.

Bobroff, N., Kochut, A., & Beaty, K. (2007). Dynamic placement of virtual machines for managing sla violations. In *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on* (pp. 119–128). IEEE.

Breskovic, I., Altmann, J., & Brandic, I. (2013). Creating standardized products for electronic markets. *Future Generation Computer Systems*, *29*(4), 1000–1011.

Breskovic, I., Maurer, M., Emeakaroha, V. C., Brandic, I., & Altmann, J. (2011). Towards Autonomic Market Management in Cloud Computing Infrastructures. In *CLOSER* (pp. 24–34). Citeseer.

Broker@Cloud. (2015). Retrieved April 12, 2018, from https://www.sintef.no/en/projects/broker-cloud/

Buyya, R., Ranjan, R., & Calheiros, R. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *Algorithms and Architectures for Parallel Processing*, 13–31.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599–616. Retrieved from http://ac.els-cdn.com/S0167739X08001957/1-s2.0-S0167739X08001957-main.pdf?_tid=13899726-1160-11e7-81b2-00000aacb35e&acdnat=1490449180_dac1fc4dfcb4bd105ea1975932cfd3b2

Calcavecchia, N. M., Biran, O., Hadad, E., & Moatti, Y. (2012). VM placement strategies for cloud scenarios. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* (pp. 852–859). IEEE.

Carlini, E., Coppola, M., Dazzi, P., Ricci, L., & Righetti, G. (2011). Cloud Federations in Contrail. In *Euro-Par Workshops (1)* (pp. 159–168).

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010). How to enhance cloud architectures to enable cross-federation. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* (pp. 337–345). IEEE.

Chaisiri, S., Lee, B.-S., & Niyato, D. (2009). Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific* (pp. 103–110). IEEE.

Chudasama, V., Tilala, D., & Bhavsar, M. (2017). SLA Management in Cloud Federation. In *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 397–404). Springer.

Claro, D. B., Albers, P., & Hao, J.-K. (2005). Selecting web services for optimal composition. In *ICWS international workshop on semantic and dynamic web processes, Orlando-USA*.

Clayman, S., Galis, A., Chapman, C., Toffetti, G., Rodero-Merino, L., Vaquero, L. M., … Rochwerger, B. (2010). Monitoring service clouds in the future internet. In *Future Internet Assembly* (pp. 115–126). Valencia, Spain.

Coronado, J. P. R., & Altmann, J. (2017). Model for Incentivizing Cloud Service Federation. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 233–246). Springer.

Coutinho, R. de C., Drummond, L. M. A., & Frota, Y. (2013). Optimization of a cloud resource management problem from a consumer perspective. In *European Conference on Parallel Processing* (pp. 218–227). Springer.

Coutinho, R. de C., Drummond, L. M. A., Frota, Y., & de Oliveira, D. (2015). Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. *Future Generation Computer Systems*, *46*, 51–68.

Darzanos, G., Koutsopoulos, I., & Stamoulis, G. D. (2016). Economics models and policies for cloud federations. In *IFIP Networking Conference (IFIP Networking) and Workshops, 2016* (pp. 485–493). IEEE.

de Carvalho, J. O., Trinta, F., & Vieira, D. (2018). PacificClouds: A Flexible MicroServices based Architecture for Interoperability in Multi-Cloud Environments. In *CLOSER* (pp. 448–455).

de Carvalho, J. O., Trinta, F., Vieira, D., & Cortes, O. A. C. (2018). Evolutionary solutions for resources management in multiple clouds: State-of-the-art and future directions. *Future Generation Computer Systems*, *88*, 284–296.

Deb, K. (2014). Multi-objective optimization. In *Search methodologies* (pp. 403–449). Springer.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature* (pp. 849–858). Springer.

Deb, K., & Jain, S. (2004). EVALUATING EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS USING RUNNING PERFORMANCE METRICS. In *Recent Advances in Simulated Evolution and Learning* (Vol. Volume 2, pp. 307–326). WORLD SCIENTIFIC. https://doi.org/doi:10.1142/9789812561794_0017

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Dell. (2019). Dell PowerEdge Rack Servers.

Demchenko, Y., Turkmen, F., de Laat, C., & Slawik, M. (2017). Defining Intercloud Security Framework and Architecture Components for Multi-Cloud Data Intensive Applications. In

Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 945–952). IEEE Press.

Dhirani, L. L., Newe, T., & Nizamani, S. (2019). Federated Hybrid Clouds Service Level Agreements and Legal Issues. In *Third International Congress on Information and Communication Technology* (pp. 471–486). Springer.

Di Martino, B., Cretella, G., & Esposito, A. (2015). Cloud portability and interoperability. In *Cloud Portability and Interoperability* (pp. 1–14). Springer.

Díaz, J. L., Entrialgo, J., García, M., García, J., & García, D. F. (2017). Optimal allocation of virtual machines in multi-cloud environments with reserved and on-demand pricing. *Future Generation Computer Systems*, *71*, 129–144.

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. *Journal of Network and Computer Applications*, *69*, 152–165.

Dupont, C., Schulze, T., Giuliani, G., Somov, A., & Hermenier, F. (2012). An energy aware framework for virtual machine placement in cloud federated data centres. In *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on* (pp. 1–10). IEEE.

Edu-yaw, T., & Kuada, E. (2018). Service Level Agreement Negotiation and Monitoring System in Cloud Computing. In *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)* (pp. 1–8). IEEE.

El Zant, B., Amigo, I., & Gagnaire, M. (2014). Federation and revenue sharing in cloud computing environment. In *Cloud Engineering (IC2E), 2014 IEEE International Conference on* (pp. 446–451). IEEE.

Elmroth, E., Márquez, F. G., Henriksson, D., & Ferrera, D. P. (2009). Accounting and billing for federated cloud infrastructures. In *8th International Conference on Grid and Cooperative Computing, GCC 2009* (pp. 268–275). https://doi.org/10.1109/GCC.2009.37

Feller, E., Rilling, L., & Morin, C. (2011). Energy-aware ant colony based workload placement in clouds. In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing* (pp. 26–33). IEEE Computer Society.

Feng, M., Wang, X., Zhang, Y., & Li, J. (2012). Multi-objective particle swarm optimization for resource allocation in cloud computing. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on* (Vol. 3, pp. 1161–1165). IEEE.

Ferdous, M. S., Margheri, A., Paci, F., Yang, M., & Sassone, V. (2017). Decentralised runtime monitoring for access control systems in cloud federations. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (pp. 2632–2633). IEEE.

Fernández-del-Castillo, E., Scardaci, D., & García, Á. L. (2015). The EGI federated cloud e-infrastructure. *Procedia Computer Science*, *68*, 196–205.

Ferrer, A. J., HernáNdez, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., … Djemame, K. (2012). OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, *28*(1), 66–77.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, *3*(1), 1–16.

Garey, M. R. (1979). Computers and intractability: A guide to the theory of np-completeness. *Revista Da Escola De Enfermagem Da USP*, *44*(2), 340.

Gartner. (2018a). Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.3 Percent in 2019. Retrieved March 13, 2019, from https://www.gartner.com/en/newsroom/press-releases/2018-09-12-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2019

Gartner. (2018b). Gartner Says Worldwide IaaS Public Cloud Services Market Grew 29.5 Percent in 2017. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2018-08-01-gartner-says-worldwide-iaas-public-cloud-services-market-grew-

30-percent-in-2017

Gartner Inc. CloudHarmony. (2018). Research and compare cloud providers and services - Service Status. Retrieved October 18, 2018, from https://cloudharmony.com/status

Goher, S. Z., Bloodsworth, P., Ur Rasool, R., & McClatchey, R. (2017). Cloud provider capacity augmentation through automated resource bartering. *Future Generation Computer Systems*. https://doi.org/https://doi.org/10.1016/j.future.2017.09.080

Goiri, Í., Guitart, J., & Torres, J. (2012). Economic model of a cloud provider operating in a federated cloud. *Information Systems Frontiers*, *14*(4), 827–843.

Govil, S. B., Thyagarajan, K., Srinivasan, K., Chaurasiya, V. K., & Das, S. (2012). An approach to identify the optimal cloud in cloud federation. *International Journal of Cloud Computing and Services Science*, *1*(1), 35.

Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008). The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, *39*(1), 68–73.

Guazzone, M., Anglano, C., & Sereno, M. (2014). A game-theoretic approach to coalition formation in green cloud federations. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on* (pp. 618–625). IEEE.

Haile, N., & Altmann, J. (2015). Risk-Benefit-Mediated Impact of Determinants on the Adoption of Cloud Federation. In *PACIS* (p. 17).

Haile, N., & Altmann, J. (2018). Evaluating investments in portability and interoperability between software service platforms. *Future Generation Computer Systems*, *78*, 224–241.

Hans, A. E. (1988). Multicriteria optimization for highly accurate systems. *Multicriteria Optimization in Engineering and Sciences*, *19*, 309–352.

Harms, R., & Yamartino, M. (2010). The economics of the cloud. *Microsoft Whitepaper, Microsoft Corporation*.

Hassan, M. M., Abdullah-Al-Wadud, M., Almogren, A., Song, B., & Alamri, A. (2017). Energy-aware resource and revenue management in federated cloud: a game-theoretic approach. *IEEE Systems Journal*, *11*(2), 951–961.

Hassan, M. M., Al-Wadud, M. A., & Fortino, G. (2015). A socially optimal resource and revenue sharing mechanism in cloud federations. In *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on* (pp. 620–625). IEEE.

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. *Information Systems Frontiers*, *16*(4), 523–542.

Heilig, L., Buyya, R., & Voß, S. (2017). Location-aware brokering for consumers in multi-cloud computing environments. *Journal of Network and Computer Applications*, *95*, 79–93. https://doi.org/https://doi.org/10.1016/j.jnca.2017.07.010

Hespanha, J. P. (2011). An introductory course in noncooperative game theory.

Hiroyasu, T., Miki, M., & Watanabe, S. (1999). Divided range genetic algorithms in multiobjective optimization problems. *Proc. of IWES*, *99*, 57–65.

Hornsby, A. (2018). How to build a multi-region active-active architecture on AWS. Retrieved from https://read.acloud.guru/why-and-how-do-we-build-a-multi-region-active-active-architecture-6d81acb7d208

Householder, R., Arnold, S., & Green, R. (2014). On cloud-based oversubscription. *ArXiv Preprint ArXiv:1402.4758*.

Hwang, J. (2001). A market-based model for the bandwidth management of IntServ-DiffServ QoS interconnection: A network economic approach.

ieeeCESocTV. (2018). Cloud Federation Standards Panel: Discussion. Retrieved January 10, 2019, from https://www.youtube.com/watch?v=LS9hQ9kllvI&index=6&list=

PLcS-HUwSHpFpbQrrDhsVFAFfLHWbMvyJe

International Data Corporation. (2018). *Worldwide Public Cloud Services Spending Forecast*. Retrieved from https://www.idc.com/getdoc.jsp?containerId=prUS43511618

James Cuff, Ignacio M. Llorente, Christopher Hill, A. M. (2017). Future Challenges in federated Cloud Computing. Retrieved April 3, 2018, from https://rcc.harvard.edu/future-challenges-federated-cloud-computing

Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., & Snible, E. (2011). Improving performance and availability of services hosted on iaas clouds with structural constraint-aware virtual machine placement. In *Services Computing (SCC), 2011 IEEE International Conference on* (pp. 72–79). IEEE.

Jeferry, K., Kousiouris, G., Kyriazis, D., Altmann, J., Ciuffoletti, A., Maglogiannis, I., … Zhao, Z. (2015). Challenges emerging from future cloud application scenarios. *Procedia Computer Science*, *68*, 227–237.

Kaewpuang, R., Niyato, D., Wang, P., & Hossain, E. (2013). A framework for cooperative resource management in mobile cloud computing. *IEEE Journal on Selected Areas in Communications*, *31*(12), 2685–2700.

Kanagavelu, R., Lee, B.-S., Le, N. T. D., Mingjie, L. N., & Aung, K. M. M. (2014). Virtual machine placement with two-path traffic routing for reduced congestion in data center networks. *Computer Communications*, *53*, 1–12.

Kaur, K., Sharma, D. R., & Kahlon, D. R. (2017). Interoperability and portability approaches in inter-connected clouds: A review. *ACM Computing Surveys (CSUR)*, *50*(4), 49.

Kemahlioglu Ziya, E. (2004). Formal methods of value sharing in supply chains. Georgia Institute of Technology.

Kim, K., Kang, S., & Altmann, J. (2014). Cloud Goliath versus a federation of cloud Davids. In *International Conference on Grid Economics and Business Models* (pp. 55–66). Springer.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective

optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*(9), 992–1007.

Kumrai, T., Ota, K., Dong, M., Kishigami, J., & Sung, D. K. (2017). Multiobjective Optimization in Cloud Brokering Systems for Connected Internet of Things. *IEEE Internet of Things Journal*, *4*(2), 404–413. https://doi.org/10.1109/JIOT.2016.2565562

Lee, H., Jang, H., Cho, J., & Yi, Y. (2012). On the stability of ISPs' coalition structure: Shapley Value based revenue sharing. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on* (pp. 1–6). IEEE.

Li, H., Wu, C., Li, Z., & Lau, F. C. M. (2013). Profit-maximizing virtual machine trading in a federation of selfish clouds. In *INFOCOM, 2013 Proceedings IEEE* (pp. 25–29). IEEE.

Li, H., Wu, C., Li, Z., & Lau, F. C. M. (2016). Virtual machine trading in a federation of clouds: Individual profit and social welfare maximization. *IEEE/ACM Transactions on Networking*, *24*(3), 1827–1840.

Liaqat, M., Chang, V., Gani, A., Hamid, S. H. A., Toseef, M., Shoaib, U., & Ali, R. L. (2017). Federated cloud resource management: Review and discussion. *Journal of Network and Computer Applications*, *77*, 87–105. https://doi.org/https://doi.org/10.1016/j.jnca.2016.10.008

Liu, C., Shen, C., Li, S., & Wang, S. (2014). A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center. In *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on* (pp. 272–275). IEEE.

Liu, X.-F., Zhan, Z.-H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018). An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Transactions on Evolutionary Computation*, *22*(1), 113–128.

Lu, L., Yu, J., Zhu, Y., & Li, M. (2018). A double auction mechanism to bridge users' task requirements and providers' resources in two-sided cloud markets. *IEEE Transactions on Parallel and Distributed Systems*, *29*(4), 720–733.

Lu, Z., Wen, X., & Sun, Y. (2012). A game theory based resource sharing scheme in cloud computing environment. In *Information and Communication Technologies (WICT), 2012 World Congress on* (pp. 1097–1102). IEEE. https://doi.org/10.1109/WICT.2012.6409239

Ma, R. T. B., Chiu, D. M., Lui, J., Misra, V., & Rubenstein, D. (2010). Internet Economics: The use of Shapley value for ISP settlement. *IEEE/ACM Transactions on Networking (TON)*, *18*(3), 775–787.

Manasrah, A. M., Smadi, T., & ALmomani, A. (2017). A Variable Service Broker Routing Policy for data center selection in cloud analyst. *Journal of King Saud University-Computer and Information Sciences*, *29*(3), 365–377.

Manno, G., Smari, W. W., & Spalazzi, L. (2012). Fcfa: A semantic-based federated cloud framework architecture. In *High Performance Computing and Simulation (HPCS), 2012 International Conference on* (pp. 42–52). IEEE.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395.

Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic Theory*. *Oxford University Press* (Vol. 1). Oxford university press New York. https://doi.org/10.2307/135312

Mashayekhy, L., Nejad, M. M., & Grosu, D. (2015). Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing*, *3*(1), 14–27.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Mohammed, A. B., Altmann, J., & Hwang, J. (2009). Cloud computing value chains: Understanding businesses and value creation in the cloud. In *Economic models and algorithms for distributed systems* (pp. 187–208). Springer.

Nah, F. F.-H. (2004). A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, *23*(3), 153–163.

Nawaz, F., Asadabadi, M. R., Janjua, N. K., Hussain, O. K., Chang, E., & Saberi, M. (2018). An MCDM method for cloud service selection using a Markov chain and the best-worst method. *Knowledge-Based Systems*.

Niyato, D., Vasilakos, A. V, & Kun, Z. (2011). Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. In *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on* (pp. 215–224). IEEE.

OASIS. (2017). OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC. Retrieved from https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

Obraczka, K., & Silva, F. (2000). Network latency metrics for server proximity. In *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE* (Vol. 1, pp. 421–427). IEEE.

Ohlhorst, F. (2010, June). Improve application performance with multithreaded applications. *EZine (TechTarget)*. Retrieved from https://searchitoperations.techtarget.com/tip/Improve-application-performance-with-multithreaded-applications

Padilla, R. S., Milton, S. K., & Johnson, L. (2013). Service value in IT outsourcing. *International Journal of Engineering and Management Sciences*, *4*(3), 285–302.

Pal, R., & Hui, P. (2013). Economic models for cloud service markets: Pricing and capacity planning. *Theoretical Computer Science*, *496*, 113–124.

Parameswaran, A. V, & Chaddha, A. (2009). Cloud interoperability and standardization. *SETlabs Briefings*, *7*(7), 19–26.

Petcu, D. (2014). Consuming resources and services from multiple clouds. *Journal of Grid Computing*, *12*(2), 321–345.

Pittaras, C., Papagianni, C., Leivadeas, A., Grosso, P., van der Ham, J., & Papavassiliou, S. (2015). Resource discovery and allocation for federated virtualized infrastructures. *Future Generation Computer Systems*, *42*, 55–63.

Rak, M., Venticinque, S., Echevarria, G., & Esnal, G. (2011). Cloud application monitoring: The mosaic approach. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on* (pp. 758–763). IEEE.

Rimal, B. P., Choi, E., & Lumb, I. (2009). A taxonomy and survey of cloud computing systems. *INC, IMS and IDC*, 44–51.

Riquelme, N., Von Lücken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. In *Computing Conference (CLEI), 2015 Latin American* (pp. 1–11). IEEE.

Risch, M., & Altmann, J. (2009). Capacity planning in economic grid markets. In *International Conference on Grid and Pervasive Computing* (pp. 1–12). Springer.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., … Caceres, J. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, *53*(4), 1–4.

Roh, H., Jung, C., Lee, W., & Du, D.-Z. (2013). Resource pricing game in geo-distributed clouds. In *2013 Proceedings IEEE INFOCOM* (pp. 1519–1527). IEEE.

Rohitratana, J., & Altmann, J. (2012). Impact of pricing schemes on a market for Software-as-a-Service and perpetual software. *Future Generation Computer Systems*, *28*(8), 1328–1339.

Roth, A. E. (1988). *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.

Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling*, *9*(3–5), 161–176.

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, *48*(1), 9–26.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83–98.

Samaan, N. (2014). A novel economic sharing model in a federation of selfish cloud providers. *IEEE Transactions on Parallel and Distributed Systems*, *25*(1), 12–21.

Satpathy, A., Addya, S. K., Turuk, A. K., Majhi, B., & Sahoo, G. (2018). Crow search based virtual machine placement strategy in cloud data centers with live migration. *Computers & Electrical Engineering*, *69*, 334–350.

Sayeedkhan, P. N., & Balaji, S. (2014). Virtual machine placement based on disk I/O load in cloud. *Vol*, *5*, 5477–5479.

Schott, J. R. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.* AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.

Serrano, R. (2007). *Cooperative games: Core and Shapley value*. Working Paper.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, *2*(28), 307–317.

Sharma, U., Shenoy, P., Sahu, S., & Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In *2011 31st International Conference on Distributed Computing Systems* (pp. 559–570). IEEE.

Shi, W., & Hong, B. (2011). Towards profitable virtual machine placement in the data center. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on* (pp. 138–145). IEEE.

Sim, K. M. (2016). Agent-based approaches for intelligent intercloud resource allocation. *IEEE Transactions on Cloud Computing*, (1), 1.

Simarro, J. L. L., Moreno-Vozmediano, R., Montero, R. S., & Llorente, I. M. (2011). Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on* (pp. 1–7). IEEE.

Smit, M., Shtern, M., Simmons, B., & Litoiu, M. (2012). Partitioning applications for hybrid and federated clouds. In *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research* (pp. 27–41). IBM Corp.

Syed, H. J., Gani, A., Ahmad, R. W., Khan, M. K., & Ahmed, A. I. A.

(2017). Cloud monitoring: A review, taxonomy, and open research issues. *Journal of Network and Computer Applications*, *98*, 11–26.

Teknomo, K. (2006). Analytic Hierarchy Process (AHP) Tutorial. Retrieved from http://people.revoledu.com/kardi/tutorial/AHP

Thabet, M., Boufaida, M., & Kordon, F. (2014). An approach for developing an interoperability mechanism between cloud providers. *International Journal of Space-Based and Situated Computing 27*, *4*(2), 88–99.

Tian, W., Xu, M., Chen, Y., & Zhao, Y. (2014). Prepartition: A new paradigm for the load balance of virtual machine reservations in data centers. In *Communications (ICC), 2014 IEEE International Conference on* (pp. 4017–4022). IEEE.

Toosi, A. N. (2014). On the Economics of Infrastructure as a Service Cloud Providers: Pricing, Markets and Profit Maximization. University of Melbourne, Department of Computing and Information Systems.

Toosi, A. N., Calheiros, R. N., & Buyya, R. (2014). Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Computing Surveys (CSUR)*, *47*(1), 7.

Toosi, A. N., Calheiros, R. N., Thulasiram, R. K., & Buyya, R. (2011). Resource provisioning policies to increase iaas provider's profit in a federated cloud environment. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on* (pp. 279–287). IEEE.

Toosi, A. N., Thulasiram, R. K., & Buyya, R. (2012). Financial option market model for federated cloud environments. In *2012 IEEE Fifth International Conference on Utility and Cloud Computing* (pp. 3–12). IEEE.

Tordsson, J., Montero, R. S., Moreno-Vozmediano, R., & Llorente, I. M. (2012). Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Computer Systems*, *28*(2), 358–367.

Tse-Au, E. S. H., & Morreale, P. A. (2000). End-to-end QoS measurement: Analytic methodology of application response time

vs. tunable latency in IP networks. In *Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP* (pp. 129–142). IEEE.

Tyson, J. (2000). How Virtual Memory Works.

Uzbekov, A., & Altmann, J. (2016). Enabling business-preference-based scheduling of cloud computing resources. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 225–236). Springer.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, *39*(1), 50–55.

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, *79*, 849–861.

Veldhuizen, D. A. Van. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.*

Venters, W., & Whitley, E. A. (2012). A critical review of cloud computing: researching desires and realities. *Journal of Information Technology*, *27*(3), 179–197. https://doi.org/10.1057/jit.2012.17

Verizon. (2018). IP Latency Statistics. Retrieved September 18, 2018, from https://enterprise.verizon.com/terms/latency/

Wang, H., Jing, Q., He, B., Qian, Z., & Zhou, L. (2010). Distributed systems meet economics: pricing in the cloud.

Wang, S.-H., Huang, P. P.-W., Wen, C. H.-P., & Wang, L.-C. (2014). EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks. In *Information Networking (ICOIN), 2014 International Conference on* (pp. 220–225). IEEE.

Wang, Wei, Li, B., & Liang, B. (2012). Towards optimal capacity segmentation with hybrid cloud pricing. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on* (pp. 425–434). IEEE.

Wang, Wenting, Chen, H., & Chen, X. (2012). An availability-aware virtual machine placement approach for dynamic scaling of cloud applications. In *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on* (pp. 509–516). IEEE.

Wang, X., & Liu, Z. (2012). An energy-aware VMs placement algorithm in cloud computing environment. In *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on* (pp. 627–630). IEEE.

Wei, J., Zhou, A., Yuan, J., & Yang, F. (2018). AIMING: Resource Allocation with Latency Awareness for Federated-Cloud Applications. *Wireless Communications and Mobile Computing*, *2018*.

Whitney, J., & Delforge, P. (2014). Data center efficiency assessment. *Issue Paper on NRDC (The Natural Resource Defense Council)*.

Wu, G., Tang, M., Tian, Y. C., & Li, W. (2012). Energy-efficient virtual machine placement in data centers by genetic algorithm. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 7665 LNCS, pp. 315–323). https://doi.org/10.1007/978-3-642-34487-9_39

Xu, H., & Li, B. (2013). Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing*, *1*(2), 158–171.

Yangui, S., Marshall, I.-J., Laisne, J.-P., & Tata, S. (2014). CompatibleOne: The open source cloud broker. *Journal of Grid Computing*, *12*(1), 93–109.

Yi, C. (2009). Using Modified Shapley Value to Determine Revenue Allocation within Supply Chain. In *Information Management, Innovation Management and Industrial Engineering, 2009 International Conference on* (Vol. 1, pp. 78–80). IEEE.

Young, H. P. (1985). Monotonic solutions of cooperative games. *International Journal of Game Theory*, *14*(2), 65–72.

Zhang, J., He, Z., Huang, H., Wang, X., Gu, C., & Zhang, L. (2014). SLA aware cost efficient virtual machines placement in cloud computing. In *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International* (pp. 1–8). IEEE.

Zhang, J., Huang, H., & Wang, X. (2016). Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, *64*, 23–42. https://doi.org/https://doi.org/10.1016/j.jnca.2015.12.018

Ziafat, H., & Babamir, S. M. (2019). A hierarchical structure for optimal resource allocation in geographically distributed clouds. *Future Generation Computer Systems*, *90*, 539–568.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195.

Glossary of Terms

Analytic Hierarchy Process (AHP)	Devised by Thomas L. Saaty, Analytic Hierarchy Process is a structured method based on psychology and mathematics for making complex decisions that involve multiple criteria
Application Footprint	The geographic locations with a significant number of users of the customer application that is hosted in the federated cloud
Application Topology	A description of the application service components with their configurations and relationships
Availability	Expressed as a percentage, it refers to the amount of time that the system or application services are running.
Cloud Federation	A voluntary arrangement between a number of cloud providers for interconnecting their infrastructure resources to enable resource sharing among each other
Coalitional Game Theory	A framework for analyzing cooperative games, which focuses on the formation, joint strategies, and

	collective payoffs of coalitions.
Decision Space	The space containing the solutions or the potential service deployment plans
Generational Distance Metric	An average distance of the solutions contained in the identified set of Pareto optimal service placement plans from a reference set (final chosen service placement plan)
Multi-Criteria Decision Making (MCDM)	A discipline within operations research that is concerned with the decision making by evaluating a number of conflicting criteria
Multi-objective Optimization(MOO)	A method within multi-criteria decision making which is concerned with the optimization of more than one objective functions simultaneously
Network Latency	Often expressed as Round Trip Time (RTT), it is the delay experienced by a data packet in a data communication network
Normalization	The act of adjusting values of different objective functions measured in terms of different units and scales to bring them into a common scale
Objective Space	The space containing the evaluations of the solutions or of the potential service placement plans

Pareto-optimal (Non-Dominated) Solution	A solution for which improvement in one objective function is not possible without compromising on at least one of the other objectives
Preference Weight Vector	A vector specifying the weights corresponding to the preference of a consumer (application owner) over various service placement decision criteria
Quality of Service (QoS)	Measurement or description of the performance of the cloud service such as a latency, availability, processing speed
Service Placement	Deployment of application service nodes in federated cloud
Service Placement Decision	The decision regarding the selection of Service Placement Plan, in other words, the decision regarding where each application service nodes should be deployed in the federated cloud resource
Service Placement Plan	Mapping of application service nodes to the federated cloud resources
Set Coverage Metric	A measure of the comparison of two non-dominated fronts (approximation sets) expressed as the fraction of the solutions in an approximation set that are dominated by at least one solution in another

	approximation set
Shapley Value	Named in the honor of scientist Lloyd Shapley, it is a solution concept in coalitional game theory, which uniquely allocates the total surplus generated from the collaboration of a set of players
Spacing Metrics	A measure of the distribution of the service placement plans, which is measured as a relative distance between consecutive solutions in the identified set of non-dominated service placement plans
Virtual Machine (VM)	Software abstraction of a physical computing system
Virtual Machine Instance	A Virtual Machine hosted on a physical computing system
Analytic Hierarchy Process (AHP)	Devised by Thomas L. Saaty, Analytic Hierarchy Process is a structured method based on psychology and mathematics for making complex decisions that involve multiple criteria

Appendix 1

In section 3.4, we proposed our service placement algorithm employing A Fast and Elitist Non-dominated Sorting Genetic Algorithm. Here, we present further details on some operations within the algorithm.

A. Genetic Operation

For the evolution of the solution, offspring solutions are generated by the process of the genetic operation, which involves three steps, namely - selection, crossover, and mutation. Parent solutions for mating are selected with binary tournament selection process. For this, two solutions are selected at random from the population. Better one between these two is selected as first parent for undergoing crossover operation. Which one is better is assessed by their rank, which is determined by the Fronts they belong to and the corresponding crowding distance assigned to them. Second parent is selected the same way. The selected pair of parents undergoes two-point crossover with a given probability of CX, and mutation process with a given probability of MUT. As the process of crossover and mutation is the same with every Genetic Algorithm, details are omitted.

$performGeneticOperation(P)$ Source: (Deb, Agrawal, Pratap, & Meyarivan, 2000)

$Q = \emptyset$

$WHILE\ (|Q| < |P|)$

$Parent1 = binaryTournamentSelection(\mathrm{P})$

$Parent2 = binaryTournamentSelection(\mathrm{P})$

$Offspring1, Offspring2$

$= performCrossoverAndMutation(Parent1, Parent2, CX, MUT)$

$Q = Q\ \cup \{Offspring1, Offspring2\}$

$RETURN\ Q$

B. Non-dominated Sorting of Population

We perform the non-dominated sorting of population as suggested by (Deb et al., 2000), In this process, we group the solutions in the populations into different fronts. From the population, all the non-dominated solutions are identified. A solution is said to be non-dominated if it is not dominated by any of the solutions in the populations. Solution 'a' is said to be dominated by solution 'b' if solution 'b' is better than solution 'a' in terms all of the objectives functions (Deb et al., 2002). For this, each of the objective functions for all solutions are evaluated and compared one to one to check the dominance. These non-dominated solutions form the first Front F1, and they are removed from the population set. From the remaining solution in the populations, again, second set of all the non-dominated solutions are identified and they form the second front F2 and are removed from the population. This process continues for other fronts until all the solutions in

the population are assigned to some fronts and the population set becomes empty.

$performNondominatedSorting(P)$ Source: (Deb et al., 2000)

$P^{'} = \{\emptyset\}$

$for\ each\ p\ in\ P\ and\ p\ not\ in\ P'$

$P^{'} = P^{'} \cup \{p\}$

$for\ each\ q\ in\ P' and\ q\ \neq p$

$if\ p\ dominates\ q, then\ P' = P'\backslash\{q\}$

$else\ if\ q\ dominates\ p, then\ P' = P'\backslash\{p\}$

C. Assignment of crowding distance to the population of solutions

All the solutions on a given front are non-dominated with respect to each other. Crowding Distance metric is used to rank different solutions of the same front. For this, the solutions in a given front are sorted on the basis of each of the objective functions. The extreme solutions with highest and lowest values of the objective functions are assigned a very large ($\propto$) distance value. For remaining solutions in the front, it is evaluated as the ratio of the difference of objective function value of solutions just above and just below in the sorted list to the difference of the maximum and minimum objective function values in the list. These distance values of a solution are summed up for all the objective functions to find the final crowding distance of each of the solutions in the front (Deb et al., 2002).

$assignCrowdingDistances(P)$ Source: (Deb et al., 2000)

$POPULATIONSIZE = |P|$

$for\ i\ in\ 1\ to\ POPULATIONSIZE$

$P[i]_{crodinddistance} = 0$

$for\ each\ objectiveFunction\ f$

$P = sort(P, f)$

$P[1]_{crowdingdistance} = \propto$

$P[POPULATIONSIZE]_{crowdingdistance} = \propto$

$for\ i = 2\ to\ (POPULATIONSIZE - 1)$

$$P[i]_{crowdingdistance} = P[i]_{crowdingdistance} + \big(f(p[i+1]) - f(p[i-1])/(f_{max}(P) - f_{min}(P))$$

Abstract in Korean (국문초록)

클라우드 산업은 규모의 경제에 영향을 받기 쉽다. 따라서 소규모 공급 업체는 합리적인 시장 점유율로 인해 어려움을 겪고 있다. 가트너(Gartner)의 최근 보고서에 따르면 Infrastructure as a Service (IaaS) 부문에서 클라우드 시장의 75 % 만 차지한 하이퍼 스케일 공급 업체는 5 곳뿐이다. 소규모 클라우드 제공자가 규모의 경제로 인해 차별화되는 이러한 맥락에서 클라우드 연합은 협력하고 향상된 자원에 대한 액세스를 얻고, 더 나은 서비스 품질을 제공하고, 다양한 서비스를 제공하고, 서비스 품질을 향상 시키며, 비용, 그리고 규모의 경제로부터 이익을 얻는다. 클라우드 제공 업체는 상업 관계를 정의하는 명확한 모델, 보다 구체적으로는 이윤을 내고 나누어 갖는 방법에 대한 규칙 있는 경우에만 페더레이션에서 공동으로 작업하려 한다. 이러한 규칙과 방법이 없는 것이 상용 시장에서 클라우드 연합이 운영되지 않는 이유 중 하나이다.

클라우드 연합에 대한 이전 연구의 많은 부분은 상호 운용성, 자원 발견, 자원 선택, 가격 책정, 회계 및 청구, 서비스 수준 계약, 보안 및 모니터링과 같은 기술적 성격의 문제에 중점을 두고 있다. 그러나 최적의 자원 공유와 공정하지 못한 분배 방법으로 분배하는 것과 같은 경제적 성향의 문제는 적절한 관심을 받지 못했다.

이 논문에서는 규모의 경제를 통해 경쟁력을 높이기 위해 클라우드 연맹 운영에 대한 경제 모델을 조사하고, 연계에서 공정하고 매력적인 인센티브 메커니즘을 통해 협력하는 방법을 제시한다. 우리의 첫 번째 목표는 개별 고객의 선호도에 따라 비용 및 다양한 QoS 기준에 대한 최적화를 통해 고객 응용 프로그램 배포를 위한 연합 리소스의 복합 선택을 용이하게 하는 알고리즘을 제공하는 것이다.우리는 분석적 계층 구조 프로세스, 다중 기준 의사 결정 방법 및 진화 적 다중 목적 최적화 알고리즘, 즉 A Fast and Elitist Non-dominated Sorting Genetic Algorithm (NSGA II)을 결합하였다. 제안된 알고리즘을 구현하여 시뮬레이션 프로그램을 개발하고 제안된 알고리즘을 평가하기 위한 시뮬레이션을 수행한다.

시뮬레이션 결과는 제안된 알고리즘이 소비자 선호에 따라 비용 및 다양한 QoS 파라미터에 최적화된 다양한 절충 점에서 서비스 배치를 가능하게 하여 최대 4 % 의 비용 절감, 47.8 % 의 처리 속도 증가, 최대 지연 시간 감소 ~ 36.6 %, 전체 가용성 증가율 5.5% 까지 향상되었다. 시뮬레이션 결과는 다중 목표 최적화 알고리즘의 성능을 비교하는 데 사용되는 세대 간 거리, 간격 및 세트 적용 범위와 같은 표준 메트릭의 측면에서 비교할 때 제안된 접근법이 벤치 마크 접근법을 능가한다는 것을 보여준다.

우리의 두 번째 목적은 연맹 회원들 사이에서 공동으로 창출된 수익의 공정한 분배를 보장하는 수입 분배 계획을 제안하는 것이다. 우리는

연계 게임 이론의 솔루션 개념 인 Shapley Value 방법을 사용하여 수익 배분 방식을 설계했다. 수익 분배 비율은 각 연맹 회원이 연맹 가치 창출에 기여한 비율에 비례하여 배분된다. 가치 창출에 대한 그들의 기여는 인프라 용량과 시장 점유율을 기반으로 추정된다. 인프라 용량은 실제 서비스 프로비저닝에 사용된 리소스를 기반으로 평가되며 시장 점유율은 해당 서비스 요청에 따라 평가된다.

시뮬레이션 프로그램을 개발하고 시뮬레이션을 수행함으로써 우리는 연맹 참여와 관련된 클라우드 제공 업체의 결정과 관련된 다양한 질문에 답하려고 한다. 시뮬레이션 결과는 자원 활용도와 투자 수익률이 30 % 이상 증가하는 형태로 연합의 이점을 입증했다. 결과는 연맹 가입의 이점이 수용량과 수용량 비율에 달려 있음을 보여준다. 용량이 작은 공급 업체 연합의 경우 페더레이션 모델에서 작동함으로써 얻을 수 있는 투자 수익 (ROI)의 이점은 수요 - 용량 비율의 낮은 수준에서 시작한다는 것이고, 용량이 큰 공급 업체의 연맹에서는 수요 - 용량 비율의 높은 수준에서 시작한다는 것이다. 시뮬레이션 결과에 따르면 비용 절감 및 서비스 품질의 탁월성에 대한 회원사의 노력을 장려함으로써 협력 내 경쟁을 허용함으로써 제안된 수익 분배 제도가 벤치 마크 참여 방식에 비해 더 우수한 인센티브 제도를 제공한다는 것을 보였다.

전반적으로 이 연구는 복합 서비스 선택 문제를 해결함으로써 관련 업계에 기여한다. 이를 통해 연방 및 클라우드 중개인은 다양한 가격

및 QoS 수준에서 서비스를 찾는 다양한 고객에게 서비스를 제공할 수 있다. 개별 고객이 지정한 트레이드 오프 지점에서 진정으로 최적화된 배포 서비스를 제공할 수 있다. 이는 수익 공유를 위한 공정한 방법과 클라우드 연합 운영을 위한 계획을 제공함과 동시에 다양한 특성을 가진 제공자에게 이익을 제공합니다. 또한 클라우드 제공 업체가 다른 회원에 대한 상대적인 지위에 따라 연맹에 가입하는 것이 유익하지 않은 경우에 대한 안내를 제공한다. 또한 이 연구는 다목적 최적화, 다중 기준 의사 결정 및 모든 도메인 내에서의 수익 공유와 관련된 연구 커뮤니티에 의미를 제공한다.

주요어 : 클라우드 경제, 경제 기반 자원 배분, 다목적 최적화, 소비자 선호도, AHP, 진화 알고리즘, 수익 공유, Shapley Value

학 번 : 2016-34687