Sumer University Journal for Pure Science (ISSN: 2790-7031)
SUJPS 2022 (November), Vol. 1, No. 1, p.p.: 82-97

Strategies for managing resources well in Federated Cloud environments for Infrastructure as a Service (IaaS)

Shaimaa hadi mohammed [1]*, Sadeq thamer hlama [2], Hayder rahm dakheel [3]

[1]*Department of* Computer Science,*College of Computer Science and Information Technology, sumer of University, The-Qar, Iraq*
[2]*Department of physics, College of Science, sumer of University, The-Qar, Iraq*
[3] *sumer of University, The-Qar, Iraq*

Abstract

Increased demand for computational resources is a direct result of the maturation and expansion of the cloud computing sector. Sometimes it might be challenging for a single CSP to achieve the promised QoS and dynamically handle all sorts of resource demands. As a unified paradigm, cloud federation allows CSPs to pool their idle resources and reap financial and quality-of-service (QoS) advantages, such as increased availability and dependability. Consequently, the cloud federation is able to keep QoS consistent even when resource demands surge unexpectedly and make use of computing resources even when they are not in high demand. Therefore, an effective resource management approach for the IaaS (Infrastructure as a Service) service of CSP in cloud federation is necessary to keep QoS intact in terms of availability and dependability and to make use of unused computing resources. This dissertation describes the various considerations that must be made when designing a framework for efficient resource management that enables individual CSPs to seamlessly provision their IaaS service in order to maximise both their individual profit and the availability and reliability of their services for their customers. Consequently, the following research issues are the focus of this paper and will be examined in detail. In this research paper, a cloud federation formation issue is introduced, and it is based on the hedonic coalition game. The effort aims to increase the total profit and availability of federations created among reputable CSPs. This framework estimates the quality and trust of each CSP using the Beta Mixture Model and then invites only the most trustworthy ones to join the federation. To further assess the value of the federation's computer resources, a general cost model is presented. The suggested paradigm culminates in a stable federation split in which no CSP is incentivized to switch federations.

Keywords: quality-of-service (QoS), IaaS (Infrastructure as a Service), federation, resource management, Mixture Model

Strategies for managing resources well in Federated Cloud environments for Infrastructure as a Service (IaaS)

شيماء هادي محمد[1]*, صادق ثامر هلامه[2], حيدر رحم دخيل[3]

[1]قسم علوم الحاسوب,كلية علوم الحاسوب وتكنولوجيا المعلومات, جامعة سومر , ذي قار , العراق
[2]الفيزياء, العلوم, جامعة سومر , ذي قار , العراق
[3]جامعة سومر , ذي قار , العراق

الخلاصة

زيادة الطلب على الموارد الحسابية هو نتيجة مباشرة لنضج وتوسع قطاع الحوسبة السحابية. في بعض الأحيان قد يكون من الصعب على CSP واحد تحقيق جودة الخدمة الموعودة والتعامل الديناميكي مع جميع أنواع طلبات الموارد. كنموذج موحد ، يسمح الاتحاد السحابي لمقدمي خدمات الحوسبة السحابية بتجميع مواردهم الخاملة وجني المزايا المالية وجودة الخدمة (QoS) ، مثل زيادة التوافر والاعتمادية. وبالتالي ، فإن الاتحاد السحابي قادر على الحفاظ على اتساق جودة الخدمة حتى عندما تتزايد طلبات الموارد بشكل غير متوقع والاستفادة من موارد الحوسبة حتى عندما

---

* shsumer@yahoo.com

لا يكون الطلب عليها مرتفعًا. لذلك ، من الضروري اتباع نهج فعال لإدارة الموارد لخدمة (IaaS البنية التحتية كخدمة) لـ CSP في الاتحاد السحابي للحفاظ على جودة الخدمة سليمة من حيث التوافر والاعتمادية والاستفادة من موارد الحوسبة غير المستخدمة. تصف هذه الرسالة الاعتبارات المختلفة التي يجب إجراؤها عند تصميم إطار عمل لإدارة الموارد بكفاءة والتي تمكن مقدمي خدمات الحوسبة السحابية الأفراد من توفير خدمة IaaS الخاصة بهم بسلاسة من أجل تعظيم ربحهم الفردي وتوافر وموثوقية خدماتهم لعملائهم. وبالتالي ، فإن قضايا البحث التالية هي محور هذه الورقة وسيتم فحصها بالتفصيل. في هذه الورقة البحثية ، تم تقديم مشكلة تشكيل الاتحاد السحابي ، وهي تستند إلى لعبة تحالف المتعة. ويهدف هذا الجهد إلى زيادة إجمالي الربح وتوافر الاتحادات التي تم إنشاؤها بين مقدمي خدمات الحوسبة السحابية ذوي السمعة الطيبة. يقدّر إطار العمل هذا جودة وثقة كل CSP باستخدام نموذج Beta Mixture ثم يدعو فقط الأكثر جدارة بالثقة للانضمام إلى الاتحاد. لمزيد من التقييم لقيمة موارد الكمبيوتر الخاصة بالاتحاد ، يتم تقديم نموذج التكلفة العامة. يتوج النموذج المقترح بانقسام فيدرالي مستقر لا يتم فيه تحفيز CSP لتبديل الاتحادات.

الكلمات الرئيسية: جودة الخدمة (QoS) ، (IaaSالبنية التحتية كخدمة) ، الاتحاد ، إدارة الموارد ، نموذجMixture

1. Introduction

Over the last several years, "the cloud" has exploded in popularity as a service delivery model based on virtualization and on-demand provisioning. The long-held ideal of computing as a utility has been realised with the advent of this paradigm and the backing of several cloud service providers like Amazon, Microsoft, and IBM. Cloud customers, thanks to the pay-as-you-go (utility) model, have access to a wide range of IT resources, including processing power, data storage, apps, and more, from any internet-connected device, regardless of their physical location [1]. Users may save time and money by not installing and operating their own hardware in the cloud. What's more, users of the cloud may use computer resources as needed and only pay for what they consume.

Cloud computing is a model for delivering on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be quickly provisioned and released with minimal management effort or interaction with the service provider [2]. As was previously said, cloud computing offers a virtualized platform for providing ubiquitous services via the on-demand allocation of virtual resources. Here, a cloud service provider's primary goal is to provide cloud services in accordance with the SLA. It can be tricky for a cloud service provider to meet their customers' expectations for Quality of Service when they have to balance multiple factors, such as the volume of requests for resources, the number of resources already in use, infrastructure limitations caused by unexpected spikes in resource demand, and scaling problems. Because of their bearing on the underlying environment's efficiency, these considerations need prompt attention. Quality of service (QoS) for small IaaS CSPs may suffer as a result of the increased difficulty of dealing with such situations compared to large corporations. When compared to large IaaS CSPs (such as Amazon Web Services or Microsoft Azure), tiny IaaS CSPs have a smaller worldwide market share [3, 4]. Demand for cloud IaaS rises in step with familiarity with and appreciation for the advantages of cloud computing. Thus, increasing the number of users like Facebook, YouTube, e-commerce business, etc. Massive, irregular, and sporadic workloads are created by these users [5, 6, 7]. This makes it challenging for a single CSP (especially smaller cloud service providers) to meet all the resource demands while also satisfying cloud consumers' high standards for quality of service. However, in modern data centres, CSPs often over supply their virtual machines to prevent slowness. In contrast, when resources like CPU time, RAM, and disc space are overprovisioned for virtual machines, it results in unnecessary waste. Furthermore, resource consumption would fall and total income collected by the CSPs will decline during periods of low needs of resource requests. As a result of the aforementioned challenges, CSPs will need to organise into a federation to ensure the consistent delivery of resources and services from many clouds.

In order to provide a reliable IaaS service for CSP in a federated cloud environment, the focus of this paper is on creating an effective resource management approach. In order to enhance I individual profit and (ii) availability and dependability of service and ability to supply assured QoS, this paper creates a framework that enables individual CSPs to effortlessly provision their IaaS service. Thus, the aforementioned goal of the paper may be accomplished by achieving a number of secondary goals, including but not limited to:

The goal of this research is to design an effective federation creation process that optimises the aggregate profit and quality of service of a federation created among reputable CSPs. The goal is to simulate a robust system for handling faults in a federated cloud setting. To provide a federation creation technique that increases cloud service providers' profitability while minimising the expenses of virtual machine (VM) migration inside a given federation.

2.Literature Survey

The goal of cloud federation is to create a unified cloud platform by bringing together many service providers' cloud environments under a shared set of standards, interfaces, and regulations. [11], [12]. Additionally, there has been a rise in CSP interest in cloud federation because to the federation's benefits (improved availability, dependability, and income). As a result, scholars started paying more attention to the complications that arose as a result of federation's widespread use. Lots of study on federated cloud computing environments has been done in recent years.

Workloads created by cloud users, such as social media sites and video hosting sites, as well as online retailers, are enormous, spikey, and erratic [3, 4, 5]. As a result, it becomes challenging for a single CSP to meet all the resource demands while also meeting the QoS requirements of cloud users. Therefore, cloud federation enables CSPs to offload their burden to another member CSP of federation, thereby solving this issue.

To solve the issue of resource reservation in advance in accordance with different long-term objectives and many provisioning phases, Chaisiri et al. [9] introduced a stochastic programming-based technique called optimum cloud resource provisioning (OCRP). As a result, their suggested system accounts for consumers' hesitance to commit to future purchases and the volatility of resource costs.

In order to reduce carbon dioxide (CO2) emissions, energy use, and costs across the geographically dispersed cloud computing infrastructure that makes up a cloud federation, Kessaci et al. [10] have introduced a meta-scheduler based on a multi-objective evolutionary algorithm.

A decentralized method was proposed by Messina et al. [11], which makes it possible to identify reliable resources and distribute them throughout a federation. Avoiding a search across the whole accessible set, the suggested work also aids cloud users or providers in locating the most appropriate collaborators.

In order to address resource competition in a federated cloud setting, Lee et al. [12] suggested a Distributed Resource Allocation (DRA) method. In order to reduce the burden of communication, the suggested method groups tasks according to their communication patterns and then attempts to assign these groups such that resource competition results in a stable equilibrium. In [13], Abdi et al. define the issue of resource allocation for bag-of-tasks (BoT) workflows across a federation of clouds as an integer linear programming problem. The suggested approach takes into account time and resource limits in the cloud while minimising the cost of operating virtual machines and transferring data.

Proper resource management and the smooth provisioning of computing resources within a cloud federation rely on a well-designed architecture. To elaborate, Rochwerger et al. [13] have outlined federated cloud's primary need in this setting. Now we need to accommodate the need for federated cloud computing. In order to address the issue of scalability in cloud computing, he introduced cloud federation and developed the RESERVOIR architecture. Through cloud federation, CSPs may rent out their unused or underused computing resources to other CSPs. They also highlighted several difficulties and their solutions in a multi-cloud context, including admission control, cross-cloud virtual networks, dynamic service elasticity, and policy-driven placement optimization. The creation of federations of CSPs is not addressed in their work.

Using a three-stage paradigm of discovery, matchmaking, and authentication, Celesti et al. [14] suggested a concept termed a Cross-Cloud Federation Manager to aid CSPs in forming a federation with other providers. They've given some thought to both domestic and international clouds. When domestic clouds' capacity falls short of what was planned, they turn to international clouds to fulfil their customers' demands for more processing power.

A framework called Balava was introduced by Nordal et al. [15] to handle calculations that span many clouds and include data with privacy restrictions. Real-time online interactive applications that need a lot of processing power are supported by the federated cloud computing paradigm introduced by Yang et al. [16]. (ROIA). They provide a scheme whereby many suppliers work together to supply tools and quality-of-service guarantees for return-on-investment analysis.

For the purpose of budgeting the expenses of a federation of hybrid clouds, Altmann et al. [17] established a cost model. Later, they implemented the suggested cost model into a COMBSPO cost reduction algorithm to choose where to put services across various cloud types.

Different clouds collaborate via a publish/subscribe service, as described by Esposito et al. [18]. Specifically, they offer IntCloudWare, an inter-Cloud middleware that allows for the streamlined connectivity of various resources across various Internet-based communication infrastructures.

3. Methods and Materials

This proposes a proactive fault tolerance system, in which the health of the physical machines is routinely examined, to address the problem of the defect inside the federation. The federation fault problem is modelled as a multi objective optimization problem with profit maximisation and migration cost reduction as the goals. To maximise the federation's profit while minimising the migration cost involved with moving the VM, the key objective of this research is to reallocate virtual machines (resources) from failed CSPs to performing CSPs. The Preference Based Fault Management (PBFM) method is created for handling faults inside the federation architecture. If the present members of the federation cannot manage the extra demand, the proposed PBFM algorithm leverages dominance and preference relationships to choose the best provider from a reserve pool of CSPs (those CSPs that are interested in joining the federation). An extensive experiment is performed to assess the relevance of the proposed mechanism. Experiments were conducted in both of these conditions and compared to the hypo papered process. Reducing the overall cost of migrating virtual machine instances from a poor CSP to a more suitable one is the major focus of the MCAFM technique (Migration Cost Assured Fault Management). To optimise federation profits while migrating virtual machine instances from a poor CSP to a good one, a technique known as PAFM is needed (Profit Assured Fault Management).

4.System Model

In this part, we'll go through a system that can identify unhealthy service providers and reallocate their virtual machine resources to them. Virtual machines (VMs) belonging to federated service providers are evaluated in a situation where many CSPs within the federation fail at the same time, necessitating their migration to CSPs that are not experiencing problems. Because of this, the federation's internal allocation of resources may need to shift, and the federation's structure may need to be altered by adding new CSPs if the current CSPs are unable to handle the increased demand. In order to redistribute virtual machines (VMs) in a way that I maximise the profit of the resultant federation and (ii) minimises the migration cost incurred due to the transfer of the VM, a proposal is made using an Integer Linear Program that calls PFTF (Proactive Fault-Tolerant Federation) (VM). The CSPs in the federation may increase their profits by reselling their unused or underused computing resources to other CSPs, making the monetary gain an essential metric. In addition, increasing the federation's total profit will entice CSPs to join the federation and share their unused computing resources with other member CSPs. The amount spent on the migration depends on the source and destination CSPs as well as the resource being moved. To illustrate, let Y = [F1, F2,..., Fq] where each federation Fh is composed of a collection of service providers (CSPs) [SP1, SP2,..., SPm] where signifies a set of SPi, I [1, z]. Users of the cloud may get virtual machines of sorts k (k = 1, 2,... n) from any of the participating service providers in the federation. The kth kind of virtual machine is distinguished by its number of cores crk, its memory mok, and its storage stk. In a federation, each SPi provides a certain quantity of resources such as CPU cores, RAM, and disc space. Total cores, memory, and storage supplied $y_{SP_i} = \begin{cases} 1 & f(t \mid A, \omega, t_l, t_{l+1}) > 68^\circ\text{C} \\ 0 & \text{otherwise} \end{cases}$-----(1)

A, o, tl , tl+1) is given by Eq. 5.1. A, o, tl , tl+1) > 68*C) otherwise the value of ySPi of member CSPs is 0 if CPU temperature of this CSPs are below 68*C.

Proactive Fault-Tolerant Federation

ILP PFTF (Proactive Fault-Tolerant Federation), redistributes VMs of federation F's faulty CSPs to other CSPs in order to maximize profit and decrease migration costs. This situation requires a compromise between maximising profit and reducing VM migration costs. Or between maximising federation profits and minimizing VM migration costs. PFTF is a multi-objective optimization problem that involves a trade-off between profit or migration cost. We define profit (MC) and migration cost (MC) as objective functions in the context of flawed CSP SPi.

$$f_{\text{profit}}\left(\left\{x_{ij}^{k}\right\}\right) = \sum_{k=1}^{n} \sum_{j=1}^{m} x_{ij}^{k} \times \text{Profit}_{j}^{k} ----(2)$$

$$f_{MC}\left(\left\{x_{ij}^{k}\right\}\right) = \sum_{k=1}^{n} \sum_{j=1}^{m} x_{ij}^{k} \times MC_{ij}^{k} ----- (3)$$

Given Eqs. (2) and (3), the following may be said for the objective functions under discussion:

$$\text{Maximize } \left(f_{\text{profit}}\left(\left\{x_{ij}^{k}\right\}\right)\right) ------(4)$$

$$\text{Minimize} \left(f_{MC}\left(\left\{x_{ij}^{k}\right\}\right)\right) ----- (5)$$

subject to $\{x^{k}_{ij}\} \in X$, where X is a collection of decision vectors that may be made as long as the following conditions are met:

$$\sum_{j=1, j\neq i}^{m} \left(1 - y_{SP_j}\right) \times x_{ij}^{k} = R_i^k \forall k ------(6)$$

$$y_{SP_j} \times x_{ij}^{k} = 0 \forall j,k$$

$$\sum_{k=1}^{n} \left[\left(cr_k \times \sum_{j=1}^{m} x_{ij}^{k}\right) + \left(cr_k \times R_j^k\right)\right] \leq Cr_j \forall j --- (7)$$

$$\sum_{k=1}^{n} \left[\left(mo_k \times \sum_{j=1}^{m} x_{ij}^{k}\right) + \left(mo_k \times R_j^k\right)\right] \leq Mo_j \forall j --(8)$$

$$\sum_{k=1}^{n} \left[\left(st_k \times \sum_{j=1}^{m} x_{ij}^{k}\right) + \left(st_k \times R_j^k\right)\right] \leq St_j \forall j --- (9)$$

$$x_{ij}^{k} \in \mathbb{N}^0 \forall j,k ----- (10)$$

VMs of type k that were transferred from the failed CSP SPi to the successful CSPs SPj are marked as xkij. For a virtual machine (VM) instance of type k, the CSP's profit is indicated by SPj as P rof itk j = P ricek j Costk j, with a normalised range of [0,1]. Here, P ricek j represents the price at which a virtual machine (VM) of type k can be purchased from a particular cloud service provider (CSP), and Costk j represents the cost to the CSP of providing an instance of VM of type k. Transferring a VM instance of type k from a malfunctioning CSP SPi to a functional CSP comes at a cost SPj is denoted by MCk ij (see Eq. 2). • crk, mok, and stk represent the number of processors, RAM, and disc space that a VM instance of type k needs. Total cores, memory, and storage for provider SPj are denoted by Crj, Moj, and Stj respectively. A service provider SPj in federation F has fulfilled Rk j if and only if there have been k requests of type j before. Here, k-type requests map to k-type virtual machines.

Constraints established by Equations 5 and 6 of PFTF need careful consideration. According to Eq. 7, the number of virtual machines redistributed to CSPs that are not in fault is equal to the number of virtual machines that were originally assigned to the CSP that is in fault (SPi). As a result, if SPj is in good health, then 1 ySPj must also be 1. Thus, in Equation 2, the LHS is the sum of all virtual machines (VMs) of type k that can be migrated from SPi to all available, healthy service providers, and the right-hand side (RHS) is the number of VMs of type k that will be migrated k that have already been transferred out of SPi. According to Eq. 7, SPi must never hand out virtual machines to CSPs with flaws. Cores, memory, and storage are all examples of resources that are subject to the limits stated in Equations 8, 9, and 10. All variables are required to be positive integers according to the constraint provided in Equation 10.

Linear Scalarization

$$\text{Maximize } \left(w_{\text{profit}} \times f_{\text{profit}}\left(\{x_{ij}^{k}\}\right) - w_{MC} \times f_{MC}\left(\{x_{ij}^{k}\}\right)\right) -----(11)$$

$$\text{Maximize } \sum_{k=1}^{n}\left[\sum_{j=1}^{m}\left(w_{\text{profit}} \times \text{ Profit } t_{j}^{k} - w_{MC} \times MC_{ij}^{k}\right)x_{ij}^{k}\right] -----(12)$$

where w is the parameter of a weight vector and is written as w = (wprof it, wMC) with the condition that wprof it + wMC = 1.

Moreover, SPi SPi |ySPi = 1 is the anticipated execution of the aforementioned ILP. Each non-faulty CSP's virtual machine allocation is revised after each step as:

$$R_{j}^{k} = R_{j}^{k} + x_{ij}^{k} \forall j --- (13)$$

Errors are represented here as a string of fresh demands on the error-free system. If that infrastructure is capable of handling the influx of requests on its own, there will be no need to establish a separate federation to handle them.

Process-Based Function-Matrix (PBFM) Algorithm

Let's look at an example before diving into the details of the proposed Preference Based Fault Management (PBFM) method. The suggested framework's step-by-step procedure is shown in Fig. 1. Let's assume that SPA, SPB, SPC, SPD, and SPE are the CSPs that make up the starting federation F. In order to determine which CSP in the original federation F is malfunctioning, the cloud broker should keep a close eye on all of them. The following procedures are used in order to foretell the malfunctioning CSPs:

First, let's assume that poor SPE results in subpar CSP.

Step 1: If the current R number of requests being handled by the defective CSP SPE (federation F) may be distributed among the remaining functional members of the current federation F. The ILP PFTF then distributes the VM requests throughout the other functional nodes in the federation, based on the request size, R.

Step 2: If the available number of requests handled by faulty CSP cannot be shared among all other current non-faulty members of federation F, then look for qualified CSPs which may be selected to join in federation F. The CSPs who are currently free to join the federation are gathered in the reserve pool (RP), from which the relevant CSPs are picked. Here, we take the list of potential CSPs from the reserve pool and narrow it down to only the ones that aren't already dominating the market with SPL, SPM, and SPP.

Third, the best provider SPM is chosen from the providers that are not in the dominant position. As a result, the optimal service provider is the one that maximises federation profits while incurring the least amount of expense to migrate virtual machines away from CSPs with poor SPE.

A new federation F* is formed when the service provider SPM is integrated into the federation F in Step 4. The newly constituted federation then divides up requests for deficient CSP SPE among its members. In addition, if the newly established federation F* is unable to meet the combined demands of the faulty CSPs, the procedures from the process of steps 2–4 is continued until a federation is created and all requests from broken CSPs may be sent to the federation's functional members. All of the remaining CSPs are re-evaluated to see which ones will be most valuable in the long run.

Because of the frequent occurrence of faults in the present cloud federation system, we propose a formal mechanism by which the cloud broker may construct a new federation. The cloud broker's major role in this approach is to transfer workloads (resources) from troublesome federation CSPs to those that are performing normally. The term for this method is "preference-based fault management" (PBFM). I the initial federation F, which comprises the problematic CSPs; (ii) a binary fault vector y, where ySPj = 1 if SPj F (service provider SPj is a member of federation F) is faulty, and 0 otherwise; and (iii) a reserve pool (RP) of interested CSPs are needed to run the PBFM. Importantly, the PBFM technique first makes an effort to move all virtual machines (VMs) from failing CSPs to non-failing CSPs inside the same federation F. PBFM will not initiate the formation of a new federation if all supplemental VM requests can be satisfied by the non-defective CSPs in the existing federation F. In addition, the federation will choose a CSP from the reserve pool RP following the procedures described in paragraph 5.2.2 if the present federation's member CSPs are unable to satisfy all VM requests. To do this, we must first isolate the suppliers who are not in the dominant position, Then, the CSP with the smallest value of j in the set is added to the current federation F. The method first extracts all of the broken CSPs in F and places them into their own set, then iterating through all of the broken CSPs in set. One by one, each broken CSP in set gives over its virtual machines (resources) to F0 in accordance with the PFTF. The present federation F0 has exhausted its available resources, or VMs, and a new CSP must be introduced to the federation if PFTF is to be achieved for a non-faulty CSP (CSP of existing federation F0). The RP that was previously set aside is being activated again. The CSP with the lowest value of min1 j is chosen to join the current federation F0 after the set of non-dominated CSPs has been received from the RP. The next step is to determine whether PFTF can be achieved by adding the selected SPj to F0. If PFTF is not achievable, repeat until it is by selecting the SPj+1 RP whose min2 j+1 value is the second minimum in set. If RP becomes null at any point throughout this operation, an error will be generated. If this can't be done, a new federation, F0, will be founded to meet the increased needs. There is a comparable change in the set when a lower proportion of CSPs are removed from the reserve pool at the end of each cycle. Therefore, a provider that was not a member of in a particular iteration may be added in the next iteration if all of its dominant providers are removed from RP.

Algorithm (Preference Based Fault Management) PBFM The parameters here are the initial federation F, the fault vector y, where ySPj = 1 if SPj F is faulty and 0 otherwise, and the reserve pool of CSPs RP. The final result of fault handling is the federation F0 (if any) that is best suited to handle the incoming requests. It's possible but unlikely that F0 and F are the same.

```
Let $\alpha = \{\forall SPj \in F|ySPj = 1\}$
$F0 = F\backslash\alpha$ / / adding all non-faulty CSPs from F to F0
for all SPfault $\in \alpha$ do
    WHILE OPT-FDR is not feasible for F0 and SPfault REPEAT
    {
        if $RP = \phi$ then
            Exit with failure
        end if
        Find $\xi \subseteq RP$, $\xi = \{SPj \in RP \mid @(SPl \in RP \text{ and } SPl\ d\ SPj\ )\}$
```

```
Choose the next CSP $SP_j \in \xi$ such that $\delta_{min\,j}$ has minimum value in set $\sigma$
$F0 = F0 \cup SP_j$
$RP = RP \setminus SP_j$
}
Allocate VM (resources) in F0 according to OPT-FDR and SPfault
end for
return F0
```

Results

Take a look at Figures 1 and 2 to observe how the functionality of PBFM, MCAFM, and PAFM changes when the number of malfunctioning CSPs in the federation increases. Examining the impact on earnings and migration costs of adding a single problematic CSP to a federation is the focus of this study (federation formed by different size request vectors) Show Fig. 1 and Fig. 2 to see how the proportion of fault changes for federations established by various request vectors when the number of problematic CSPs grows by one. Average profits from PBFM, MCAFM, and PAFM all drop as the proportion of fault rises (Fig. 1a–d).

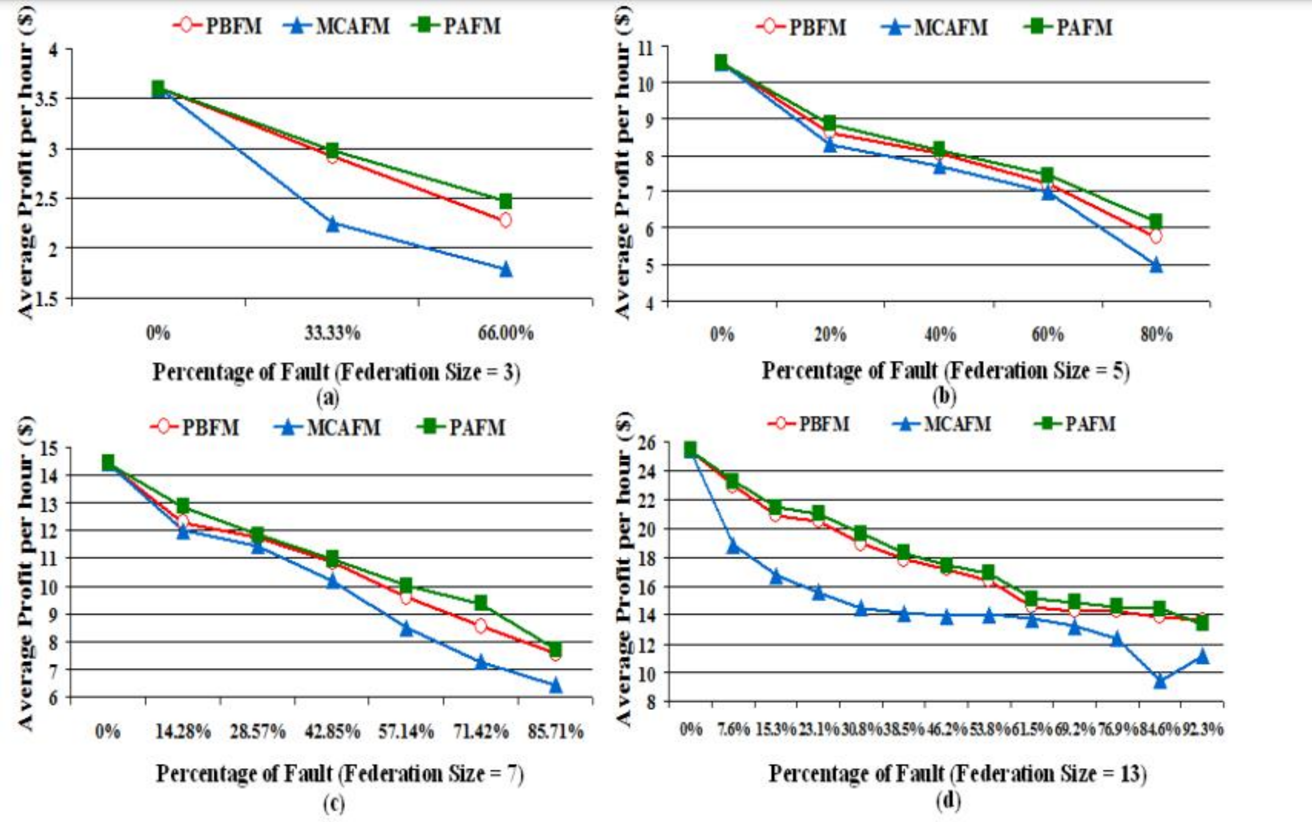

Figure 1: When the number of faults in a federation rises, the average profit of federations created by PBFM, MCAFM, and PAFM decreases when (a) the size of the request is small, (b) the size of the request is medium, (c) the size of the request is big, and (d) the size of the request is extra-large.

This is due to the fact that the process outlined in [9] is used to create the first federation. The authors of outline a technique whose primary goal is to identify the set of CSPs that maximises the federation's net benefit. Since this is the case, no subsequent federation will be more successful. In addition, the results of all three methods are compared in Fig. 1 for a federation of size 3. When the proportion of faults goes from 0% to 33.33 percent, both PBFM and PAFM earn almost the same average profit, but MCAFM obtains a much lower average profit for the federation. This trend continues as the percentage of faults climbs to 66.6 percent. Because there is no assurance that profits will be maximised by using the MCAFM process. When the proportion of faults increases from 33.33 percent (the already deployed single defective CSP) to 66.666 percent, however, PAFM is shown to function slightly better than PBFM (present two faulty SSPs). When the federation size has been set to 5, the results of all three methods can be compared. (Federation formed through a mediumized request. Fig. 1 (b) shows that when the percentage of fault increases from 0% up to 60%, average profits are produced by PBFM or PAFM. PAFM performed better than PBFM when there was 60% of fault, but it dipped to 80% when there was 80%. This is not surprising considering that PAFM's primary objective is to maximize federation profit. Figure 5.6 (b), which shows that the MCAFM method produces a consistent lower profit than the PBFM or PAFM methods when faults increase from 0% up to 80%. This is because MCAFM's primary goal is to lower the total migration cost of the Federation. Therefore, the overall profit from the MCAFM technique is modest and decreases as the fault percentage within the Federation increases. It is also shown in Fig. 1 (c), and (d), show that the three methods of PBFM (MCAFM), PAFM, and PAFM have almost identical patterns to those in Fig. 1 (a and (b) - i.e., even though PAFM has a higher average profit per federation than PBFM, this difference is very small. MCAFM, on the other hand, has the lowest average profit of federation.

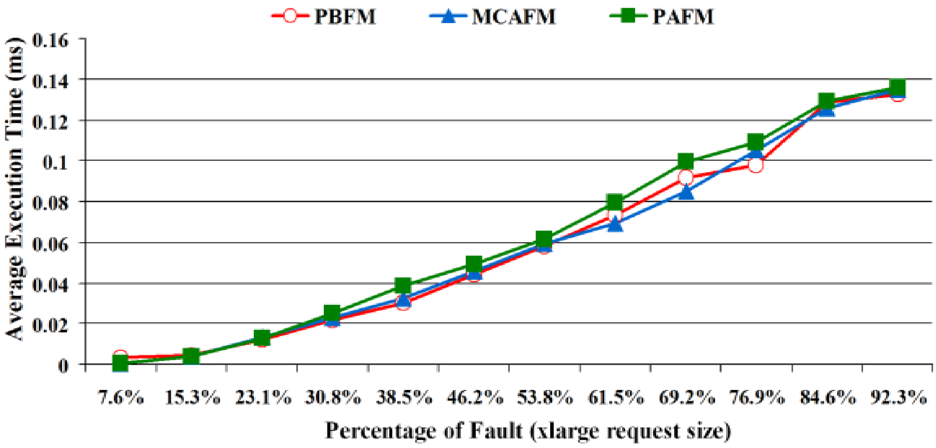

Figure 2: Examining the relative efficiency of PBFM, MCAFM, and PAFM in practise.

As can be seen in Fig. 2, a higher proportion of faults results in a longer execution time. Theorem 5 states that the time required to finish these procedures grows as both the number of flaws and the size of the reserve pool increase (the number CSPs who are eager to participate in federation construction).

In this section, we describe an ILP-based paradigm for fault redistribution across federated clouds, allowing for the rapid relocation of virtual machines in the event of a cloud outage. Redistributing virtual machines in a cloud federation setting with a view toward increasing profit while decreasing migration costs is a multi-objective optimization issue that this ILP seeks to solve. CSPs inside the federation may utilise a proactive fault tolerance mechanism based on CPU temperature to foresee when a physical machine will fail. To deal with the issue of an unpredictable number of broken CSPs, we offer an algorithm based on dominance and preference relationships. This algorithm's efficacy has been measured by analysing and testing it on an actual cloud dataset. Two alternative algorithms are compared with the suggested method to see which is best for a certain scenario. The suggested approach indicates that VM migration expenses have a major impact on the federation's bottom line. For this reason, it is critical that federation formation solutions be created with the purpose of minimizing the cost of VM migration in order to maximize the overall profit for the federation. Therefore, this continues with a discussion of the new federation creation architecture that aims to minimise the cost of VM migrations in order to increase the federation's profit as a whole.

5.Conclusion

An increasingly common method of service delivery, cloud federation involves several, dissimilar cloud providers working together to produce a single, unified experience for end users. Participating CSPs gain a lot from the federation in terms of effectively managing their IaaS (such computing resources) service via collaboration with other CSPs to establish a federation. The IaaS service of a CSP in a federated cloud environment, however, requires a solid structure for effective administration. The fundamental goal of this paper is to provide a framework for CSPs that allows them to make better use of their idle and underused computing resources while still providing their customers with dependable, always-on cloud IaaS service. At the end of the paper, I briefly outline the research's original contributions. Finally, we address some of the paper's weaknesses and provide some suggestions for further research.

6. References

[1] M. Schwarzkopf, D. Murray, and S. Hand, “The seven deadly sins of cloud computing research,” Proc. the 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud ’12), Jun. 2012.

[2] CloudHarmony Service Status of different cloud service providers. [Online]. Available: https://cloudharmony.com/status-group-by-regions

[3] Amazon EC2 Pricing. (2016). [Online]. Available: https://aws.amazon.com/ec2/pricing/

[4] E. Khosrowshahi-Asl, J. Bentahar, H. Otrok, and R. Mizouni, “Efficient Community Formation for Web Services,” IEEE Transactions on Services Computing, vol. in press, 2014.

[5] M. Bellaiche, A. Adel, H. Talal, “A Cooperative Game for Online Cloud Federation Formation Based on Security Risk Assessment.” 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). IEEE, 2018.

[6] R. Van den Bossche, K. Vanmechelen, J. Broeckhove, “Costoptimal scheduling in hybrid IaaS clouds for deadline constrained workloads”, in Proc. 3rd IEEE Int. Conf. Cloud Comput., pp. 228–235, (2010).

[7] I. Goiri, J. Guitart, and J. Torres, “Characterizing cloud federation for enhancing providers’ profit.” Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010.

[8] M. Hassan, B. Song, and E. Huh, “Distributed resource allocation games in horizontal dynamic cloud federation platform,” in Proc. IEEE Intl.Conf. on High Perf. Comp. and Comm., 2011, pp. 822–827.

[9] A. Toosi, R. Calheiros, R. Thulasiram, R. Buyya, “Resource provisioning policies to increase IaaS provider’s profit in a federated cloud environment”, in Proc. 13th IEEE Intl. Conf. High Perform.Comput.Commun., pp. 279–287, (2011).

[10] S. Chaisiri, B. S. Lee, D. Niyato, “Optimization of Resource Provisioning Cost in Cloud Computing”, in IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 164-177, (2012).

[11] Y. Kessaci, N. Melab, E. G. Talbi, “A Pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation”. Cluster Computing, 16(3), pp. 451-468, (2013).

[12] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, G. M. L. Sarne, “A trustaware, self-organizing system for large-scale federations of utility computing infrastructures”, Future Gen. Comput. Syst. 56, pp. 77–94, (2015).

[13] Y. H. Lee, K. C. Huang, M. R. Shieh et al. “Distributed resource allocation in federated clouds.” The Journal of Supercomputing 73.7 (2017): 3196-3211.

[14] S. Abdi, L. PourKarimi, M. Ahmadi et al. “Cost minimization for bag-oftasks workflows in a federation of clouds.” The Journal of Supercomputing 74.6 (2018): 2801-2822.

[15] B. E. Zant, I. Amigo, M. Gagnaire, “Federation and Revenue Sharing in Cloud Computing Environment”, IEEE International Conference on Cloud Engineering, Boston, MA, 2014, pp. 446-451, (2014).

[16] S. Rebai, M. Hadji, D. Zeghlache, “Improving profit through cloud federation”, 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2015, pp. 732-739, (2015).

[17] H. Li, C. Wu, Z. Li, F. C. M. Lau, Profit-maximizing virtual machine trading in a federation of selfish clouds, Proceedings IEEE INFOCOM, Turin, 2013, pp. 25-29 (2013).

[18] B. Rochwerger, D. Breitgand, E. Levy, A. Galis et al., “The reservoir model and architecture for open federated cloud computing,” IBM J. Of Res. and Dev., vol. 53, no. 4, pp. 4–1, 2009.

[19] B. Rochwerger, D. Breitgand, A. Epstein et al. “Reservoir—when one cloud is not enough”, IEEE computer vol. 44, no. 3, pp. 44-51, (2011).

[20] A. Celesti, F. Tusa, M. Villari, A. Puliafito, “How to Enhance Cloud Architectures to Enable Cross-Federation, 2010 IEEE 3rd International Conference on Cloud Computing”, Miami, FL, pp. 337-345, (2010).

[21] A. Nordal, A. Kvalnes, J. Hurley, D. Johansen, “Balava: Federating private and public clouds”, in Proc. IEEE World Congr. Serv., pp. 569–577, (2011).

[22] X. Yang, B. Nasser, M. Surridge, S. Middleton, “A business-oriented Cloud federation model for real-time applications”, Future Generation Computer Systems vol. 28, no. 8, pp. 1158-1167, (2012).