The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 1, 2018
Pages 1-9
Regular Paper

Comparison of Cloud Federation Models with Selfish Cloud Service Providers

Moslem Habibi Ali Movaghar Hamid Haghshenas

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Abstract

With the advent of Cloud Computing and its ever growing role in today's IT marketplace, Cloud Federation has recently emerged as a new frontier to be tackled by researchers so as to enable opaque resource sharing and cooperation between separate (public or private) clouds. Current federation models, which are based on either centralized or P2P methods, can be violated by the natural selfish behavior of cloud service providers, and no clear consensus exists as to which models perform more efficiently under different situations. In this paper, alongside introducing a new weighted P2P method, we use mathematical models to capture the behavior of different federation mechanisms based on each method, taking into consideration how the selfish behavior of cloud service providers might violate the federation mechanism. Our goal is to show how each model behaves under these violations and how they compare based on obtained individual profits and overall federation efficiency. Our results show that these three cloud federation methods, whilst providing the same amount of social welfare, counter the aforementioned violations differently and that individual cloud profits, whilst comparable, are in correlation with different federation scenarios.
Keywords: Cloud Computing, Cloud Federation, Cloud Collaboration, P2P Federation, Centralized Federation, Interclouds.

1. Introduction

In recent years Cloud Computing has become an important paradigm that has created a huge shift in the way resources (be they computation, storage, business services etc.) are consumed. To this end many cloud service providers, or CSPs, have sprung up in the global marketplace that provide services to cloud clients via public clouds in many of the different layers of cloud computing (Infrastructure as a Service, Software as a Service, Platform as a Service, ...) such as Amazon EC2, Google App Engine and Microsoft Azure. Private Clouds, which are hosted on-site for larger corporations and government agencies, have also begun taking larger and larger strides using commercial (e.g. VMware) and open source (e.g. OpenStack) technologies and software. This area, although containing many advantages, has many pitfalls that must be addressed, like that of disparate solution providers each using internal standards and interfaces that lock in the client to a predefined set of features and resources available on their chosen cloud platform (a situation parallel to that of the early days of commercial computing) and also SLA violations due to peaks in client requests that CSPs cannot individually handle at some given time. At the moment cloud service providers themselves cannot benefit from the main advantages of the cloud computing paradigm, mainly opaque and on demand use of virtualized and as a service resources, be it because of the aforementioned lack of standards or lack of market incentives that can make cooperation between CSPs commercially feasible. The existence of large CSPs has also made entry in this field difficult for smaller players and new paradigms are needed to balance the scale in this regard [1].

This problem area is the focus of a (somewhat) new area in cloud computing called Cloud Federation. Cloud Federation (or Federated Cloud Computing) attempts to provide a solution for better, more reliable and cost incentive delivery of cloud services to clients by enabling federation between CSPs, e.g. the outsourcing of client requests by a CSP to other CSPs when the need arises. Although such federation of requests between different cloud service companies has precedence in many areas such as the use of sub-contractors in manufacturing sectors [2], limitations exist in the IT sector prohibiting such cooperation, which can be summarized into two main areas [3]:

- Insufficient market forces to motivate sharing at the service layer
- Lack of standard interfaces

Both the areas described above have seen a rise in interest in recent years with many valuable results presented (as will be described in Section 2). Federation itself can be done in two overall methods, i.e. P2P and centralized federation [4]:

- **Centralized Federation**: Here a common entity exists that bears the main federation responsibility. Usually this takes the form of a repository in which cloud service providers register to receive federated requests, but other tasks such as the efficient delegation of client requests to each cloud service provider and profit distribution can also be done here. An example of such an entity commonly used in centralized federation models is the broker [5].
- **Peer to Peer (P2P) Federation**: In such models the cloud service providers are directly in contact with each other and they themselves bear the responsibility of delegating client requests to cooperating CSPs and other related tasks.

So far there is no clear indication as to which of the two above models has clear advantages over the other, and recent work in the field of cloud federation has been presented for both, as will be discussed in Section 2. Whilst from existing literature we know the general advantages that centralized models have over P2P ones (e.g. simplification of nodes other than the shared entity) and those that P2P models have over centralized ones (e.g. no single point of failure), in this paper we examine more closely the discrepancy (if it exists) between P2P and centralized federation based on the efficiency of the federation, i.e. total profit obtained in the federation mechanism. Such efficiency is the basis of another important question that arises in this context, mainly how the benefit of participating in a cloud federation environment differs based on the size of a CSP, i.e. do large cloud service providers with access to many customers have the same amount of incentives to participate in cloud federations compared to smaller sized CSPs?

To find answers to the above questions and reach an efficient cloud federation model that is applicable to various scenarios, we take as our starting point the well cited federation model presented by Mashayekhi *et al.* [6]. There a centralized cloud federation method is presented in which participants (the CSPs) have a common goal of optimizing the total profit obtained by the federation mechanism, i.e. to maximize social welfare. Although containing novel results, this model lacks in some key areas, including:

- It does not take into account the fact that cloud service providers (being business entities) usually behave selfishly and if maximizing social welfare conflicts with their individual profits, the CSPs may not participate in the federation effort (or worse yet cheat the mechanism)
- The effects of federation on CSPs with different sizes and how this may influence their participation in a federation mechanism is not taken into consideration
- The model can only be used in a centralized federation method, which whilst containing some advantages over the P2P method (e.g. simplification of nodes other than the shared entity), limits its use in many real world scenarios. These limitation can be business related (e.g. using a centralized model entails CSPs to redirect their customers to an external broker, something large providers would not appreciate) or technical (e.g. changes needed in different CSPs to be able to interact with a common broker)
- Keeping in mind the differences that arise in P2P and centralized methods, an important issue that is not analyzed is whether one of these models is more efficient, first for the individual participating CSPs' profits, and second for the social welfare or overall profit that can be obtained by the model

Another important matter that must be addressed is the distribution mechanism of tasks in P2P federation, as unlike the centralized one where the broker federates the tasks with the goal of obtaining the maximum total profit for the whole model, in the P2P method the selfish CSPs are individually federating tasks to each other and the way they do this may by itself make or break the federation mechanism.

Keeping in mind the above facts, in this paper we analyze the initial centralized model to identify various ways in which federation violations can occur, i.e. instances in which selfish decisions by CSPs can negatively affect their individual profits or the federation model's social welfare. We then examine each violation and either show that it has no serious effect on the federation mechanism or expand the existing model to counter such violations. Our main goal is to show that through all this, federation is always preferable for the participating CSPs.

The rest of this paper is organized as follows; in Section 2 we discuss some of the key related work done in the field of cloud federation, be it P2P or centralized. Next in Section 3 we describe our framework alongside relevant notations and also the workings of the presented federation models. Finally in Section 4 we explain the simulation environment used whilst analyzing the obtained results.

2. Related Work

In the last couple of years more and more literature has been published dealing with cloud federation and related challenges. This literature can broadly be categorized into those papers that deal with modelling cloud federation formation alongside relevant market forces and the subsequent interactions that they entail, and those that present implementation based and practical results such as designing cloud federation reference architectures or providing APIs that enable cooperation between various cloud service providers. Whilst such results showcase the importance of the collaboration and cooperation needed to bring cloud computing to its next level of maturity as a computing paradigm, many important topics and areas are as yet untouched and therefore much more work needs to done in this regard.

Mashayekhy *et al.* [6] present a centralized cloud federation model in which CSPs make use of a broker that acts on their behalf by accepting client requests, comprising instances of various VM types. A cloud federation game, which is a coalitional game, is then used to decide which subset of the available CSPs will accept the incoming client request and how this request is divided between them. The characteristic function used here is based on the total profit that is obtained by all CSPs involved in the federation, which is calculated using an integer program called IP-CFPM. This is further used to define the cloud federation formation mechanism, which uses an algorithm based on merge and split functions to provide an approximate solution to the problem of deciding which subset of CSPs should receive the current client request. This paper does not deal with direct federation between CSPs, i.e. P2P federation, and therefore the results are only applicable to scenarios in which a common broker is agreed upon by the various cloud service providers.

In [7], Breitgand *et al.* focus on ways to distribute client requests between various CSPs in a federated model so as to maximize CSP profit and also comply with service level agreements. For this an integer linear programming model alongside a greedy algorithm is presented which results in a decrease of power consumption in data centers whilst at the same time load balancing requests between different CSPs. Hassan *et al.* [8] also provide a method for distributed resource allocation in a federated environment based on cooperative game theory. Here centralized and distributed algorithms are presented where each CSP's goal is to increase its individual profit. Although the results here incentivize collaboration between CSPs, the presented model cannot be easily adapted to general cloud computing environments.

Das *et al.* [9] present a quality of service and profit aware cloud confederation model based on different VM allocation policies when both local client requests as well as federated ones must be processed. Taking into account the importance of maintaining SLAs, thresholds are defined to decide on which instances of VMs from different allocation policies are accepted by a CSP so as to not reduce QoS for local client requests. This is important as when CSPs only consider short term profit motives in a federation, they risk losing current customers or violating SLAs. Ye *et al.* [10] provide a novel risk aware solution to this problem via a reinsurance-emulated collaboration mechanism where each CSP determines its resource retention by predicting future resource demands. Here a centralized broker is used to assign requests to CSPs once resource demands exceed retention in such a way that maximizes the resource utilization of the whole federation environment.

Ray *et al.* [11] go further by considering the impact of QoS alongside other parameters, mainly trust & price. This is done using a multi-criteria federation selection mechanism which defines domination relations among federations based on the mentioned parameters and finds their euclidean distance with a no-fluctuation line, i.e. the deviation between the parameter values. Similar trust & QoS parameters are also considered by Thomas *et al.* [12] to propose an efficient partner selection mechanism in cloud federations. Here the Analytical Hierarchy Process method is used to calculate the weights of various QoS parameters which for the basis of a Technique for Order Preference by Similarity to Ideal Solutions method to rank various CSPS according to user requirements.

In [13] El Zant *et al.* focus on different revenue sharing methods in a federated cloud environment, where the CSPs are interacting in a P2P setting. Here it is shown that none of the different methods, revenue sharing based on a constant factor, revenue sharing proportional to the amount of requests accepted by each CSP from the total client request and revenue sharing based on the Shapley Value [14], provide best results in general. At the same time it is shown that constant prices should be offered by the CSPs accepting federated requests so as to justify federation as a means to increase the involved CSP's bottom line.

Li *et al.* [15] also present a model for pricing in federated clouds, but here a cloud exchange model is used based on service level agreements where CSPs register in a central directory that clients can then look up. Chang *et al.* [16] use a mathematical model to compare federated and non-federated cloud computing models, where client requests can be deterministic or random. Here it is shown that federation optimizes the number of servers that are needed to service all client requests successfully in comparison to CSPs acting individually in this regard.

Another important topic of research here is the difference between smaller private clouds and larger public ones when participating in a federation mechanism. To this end Chen *et al.* [17] describe two federation models; horizontal federation which is formed between individual private clouds so that they can scale up their services by pooling resources, and vertical federation in which private clouds outsource part of their requests to a larger private cloud. Stable federations and their economical aspects using this joint vertical-horizontal federation framework are studied with the efficiency of formed federations evaluated based on proposed workload factoring strategies.

As mentioned before, some research in this area focuses on more practical aspects of cloud federation, such as reference architectures and interoperability APIs [18]. Rochwerger *et al.* [19] describe the RESERVOIR architecture, a modular, extensible cloud architecture created for the P2P cloud federation model. Villegas *et al.* [3] also describe a P2P cloud architecture, but here federation can be done in cloud computing service layers other than Infrastructure as a Service, mainly Platform as a Service and Software as a Service. Lastly as an example of a centralized federation architecture, one can point out the Cloud Exchange architecture proposed by Buyya *et al.* [20]. Here the clients and also the CSPs participate in the exchange, with CSPs registering their provided services in the exchange and clients announcing their service needs both via dedicated exchanges.

3. Federation Models & Framework

Here the framework used in this paper is described and the defined models are presented alongside relevant notations. In each step we consider the possible violations which may break the federation mechanism, presenting a modified version of each model in order to prevent such violations. Here two overall instantiations of the federation concept are presented, one based on the centralized (broker based) federation method and another based on the P2P one.

We assume that there exists a set of cloud service providers $\boldsymbol{CSP}$ where $CSP = \{CSP_1, CSP_2, CSP_3, \ldots, CSP_n\}$ and $\boldsymbol{n}$ is the total number of cloud service providers in the model. Whilst cloud service providers can themselves control and use multiple data centers, without any loss of generality we abstract from this detail and deal with each CSP as a single atomic entity that processes client requests. Each CSP has a total quantity of resources that are represented by the vector

$RCSP_i = <Cor_i, Mem_i, Sto_i>$ where

- Cor_i: The total number of processing cores available at the cloud service provider,
- Mem_i: The total amount of main memory available at the cloud service provider in gigabytes,
- Sto_i: The total amount of storage (or secondary memory) available at the cloud service provider in gigabytes.

The CPs offer virtual machine instances to clients from the set ***VM*** where $VM = \{VM_1, VM_2, VM_3, ..., VM_m\}$ and ***m*** is the total number of available VM types. The needed resources to run an instance of each type of VM are represented by the vector $RVM_i = <C_i, M_i, S_i>$ for each VM type *i* where

- C_i: The number of cores needed to run an instance of the VM,
- M_i: The amount of main memory needed to run an instance of the VM in gigabytes,
- S_i: The amount of storage needed to run an instance of the VM in gigabytes.

Therefore as can be seen, due to resource constraints, a limited number of instances of each VM type can be processed by a CSP at any given time. There is also a cost associated to running an instance of a VM on a CSP (power consumption costs, hardware costs, etc.). This cost differs for each VM type and on each CSP, and is denoted by $cost_{ij}$ where $1 \leq i \leq n$ and $1 \leq j \leq m$. Also, the asking price for each VM type is constant across the board and is denoted by p_i where $1 \leq i \leq m$.

The client therefore requests any number of instances of each VM type. In the centralized model this request is given to the broker, who then federates it to the other CSPs as will be explained below, whilst in the P2P model the request is directly given to one of the CSPs. The client request is represented by the vector $Req = <r_1, r_2, r_3, ..., r_m>$ where $\forall_i\, r_i \geq 0$. Therefore for each VM type, the client specifies the number of instances that it requires. Keeping in mind the above information, federation can happen using different methods, mainly based on Centralized or P2P interactions, which will be described next.

3.1. Centralized Federation Model

The centralized federation model presented here (depicted in Figure 1) is taken from Mashayekhi *et al.* [6]. Here the CSPs have agreed on a common broker that acts on their behalf by accepting each client request and then federating it between the different CSPs. When the results are ready and the VMs have been provisioned, it is the broker that acts on the CSP's behalf and interacts with the client. The broker also bills the client based on the total price of the request vector ***Req***, and then pays each CSP after the job is complete. The way to compensate the CSPs can vary and whilst in [6] the Banzhaf value is used to determine each CSP's payoff, other methods have also been proposed [21, 13, 22].

The most important question here is how the broker should partition the client request so as to delegate it to the CSPs. To this end the overall goal of the broker is to maximize total profit (social welfare), therefore it solves an integer program (IP) called IP-CFPM that maximizes efficiency and then delegates the client request based on its results:

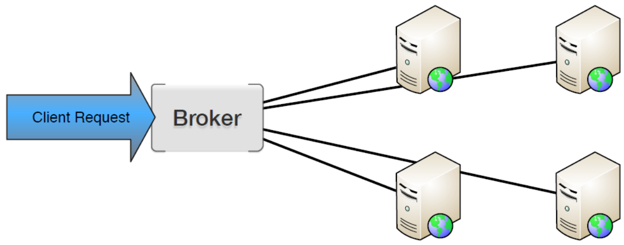

Figure 1: Centralized Federation Model. Here the broker accepts client requests on behalf of the cloud service providers and then federates them so as to increase federation efficiency.

$$Maximize \quad \sum_{CP_i \in CP} \sum_{j=1}^{m} x_{ij}(p_j - cost_{ij}),$$

$$Subject\ to:$$

$$\sum_{j=1}^{m} C_j x_{ij} \leq Cor_i \ \ (\forall\, 1 \leq i \leq n),$$

$$\sum_{j=1}^{m} M_j x_{ij} \leq Mem_i \ \ (\forall\, 1 \leq i \leq n),$$

$$\sum_{j=1}^{m} S_j x_{ij} \leq Sto_i \ \ (\forall\, 1 \leq i \leq n),$$

$$\sum_{j=1}^{n} x_{ij} = r_j \ (\forall\, 1 \leq j \leq m),$$

$$\sum_{j=1}^{m} x_{ij} \geq 0 \ \ and\ x_{ij}\ is\ an\ integer\ (\forall\, 1 \leq i \leq n)$$

Here the value of $\sum_{CP_i \in CP} \sum_{j=1}^{m} x_{ij}(p_j - cost_{ij})$, where x_{ij} shows the number of instances of VM_j delegated to CP_i, is the efficiency of the federation (*federation efficiency* or *FE*), a value that the broker, and therefore consequently the IP tries to maximize.

With regard to this centralized model, one has to point out that the assumption that the CSPs are content with maximizing the total profit of the environment irrespective of their own individual profit is not a general one and can violate the federation mechanism. This matter becomes more important if we consider the fact that without federation the incoming client requests could have gone directly to one of the participating CSPs (usually the larger ones) where it could have selfishly obtained the maximum individual profit possible. To this end an important question that arises is whether CSPs have any incentives to participate in the centralized cloud federation model against going at it alone? Another important question is whether changing the federation method can help in resolving such possible violations? Therefore next we modify the existing model so that it uses a P2P method of cloud federation.

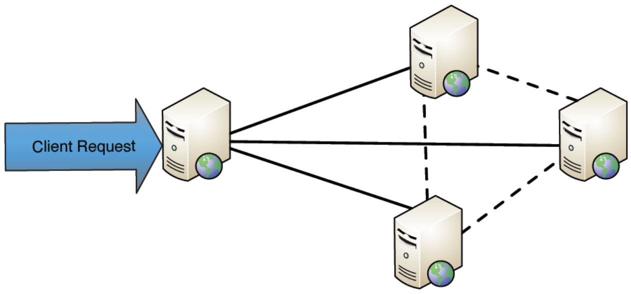

Figure 2: P2P Federation Model. Here each cloud service provider receives client requests directly, and federates a subset that it cannot process itself, due to resource limitations, to other CSPs.

3.2. P2P Federation Model

In the P2P federation model there is no agreed upon broker, and all cloud service providers receive requests directly from the client (see Figure 2), but if the request size cannot be accepted by the single CSP, then federation is needed. The cloud service providers in our model are rational, and therefore selfishly choose the best subset of the request vector based on total obtained individual profit (*Individual Efficiency* or *IE*), keeping in mind resource limitations. This selfish behavior is not only intuitive, but can in itself be an enabler in incentivizing federation between cloud service providers [23]. This can be modeled as a multidimensional knapsack problem [24], where if the CSP receiving the initial client request has index ***g***, then the dimensions are the three resources described above and we have:

$$Maximize \quad \sum_{j=1}^{m} x_{gj}\left(p_j - cost_{gj}\right),$$

$$Subject\ to:$$

$$\sum_{j=1}^{m} C_j x_{gj} \le Cor_g,$$

$$\sum_{j=1}^{m} M_j x_{gj} \le Mem_g,$$

$$\sum_{j=1}^{m} S_j x_{gj} \le Sto_g,$$

$$\sum_{j=1}^{m} x_{gj} \ge 0 \;\; and\ x_{gj}\ is\ an\ integer$$

The remaining client requests that CSP_g could not handle due to resource constraints are then federated to the remaining CSPs just as in the centralized model's IP-CFPM, but with request vector $\acute{Req}$ where:

$$\acute{Req} = Req - < x_{g1}, x_{g2}, x_{g3}, \dots, x_{gm} >$$

Here another possible violation arises, as there is no incentive for the CSPs to federate the remaining tasks efficiently as the above model dictates. Can this negatively affect the social welfare of the federation model or the CSPs' individual profits? Next we present a modified version of the P2P model where the delegation of tasks by the initial CSP receiving the client request takes into account the provider's own incentives.

3.3. Weighted P2P Federation Model

In the weighted P2P federation model (WP2P for short) after the initial phase where the CSP receiving the client request selfishly selects a subset for itself, the remaining VMs in the request vector are federated based on the history of past interactions between the CSPs, as opposed to maximizing the total efficiency of such federations. To this end we define the matrix $FedWeight = \left[FedWeight_{ij}\right]_{i,j\in\{1,\dots,n\}}$ where $FedWeight_{ij}$ represents the total amount of profit obtained by CSP_i from tasks that were federated to it by CSP_j in various federation rounds. Now when CSP_i wants to federate the remaining tasks between the various CSPs, it does this with a probability that is correlated with the amount of profit it has gained due to the actions of those CSPs. The pure P2P model described in the previous subsection can be seen as a special case of the WP2P model where the *FedWeight* matrix contains the same value in all entries.

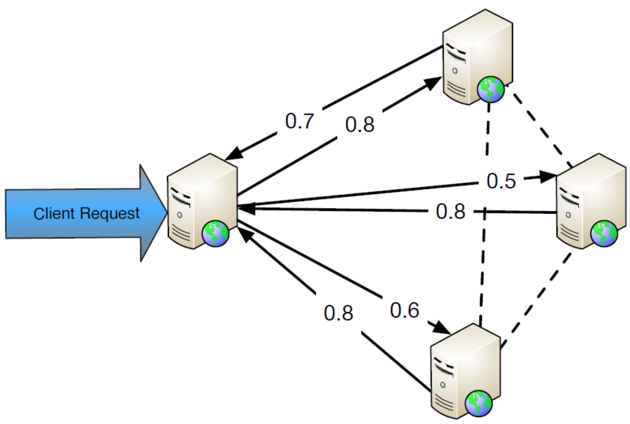

Figure 3: The Weighted P2P Federation Model. Here similar to the pure P2P model each cloud service provider receives client requests directly, but the federation phase is done based on past interaction captured in the *FedWeight* matrix.

Therefore, here the CSPs make their federation decisions based on past interactions and reward other CSPs who have provided a benefit to them. Keeping the above points in mind the cloud service provider which has received the client request (e.g. CSP_g) uses a modified IP for federation in which the objective function is changed to:

$$Maximize \sum_{CP_i \in CP'} \sum_{j=1}^{m} x_{ij}\left(p_j - cost_{ij}\right) \overline{FedWeight}_{gi}$$

where $\overline{FedWeight}$ is the normalized matrix of $FedWeight$ and $CP' = CP/\{CP_g\}$.

In the next section we describe the simulation environment used to compare the above mentioned federation methods with each other and also with the scenario in which no federation occurs. We then present and analyze the obtained results.

4. Simulation & Results

To evaluate and compare the above proposed models and how they affect overall efficiency and the CSPs' individual efficiency, a set of simulation experiments have been performed. We use the setup of Mashayekhi et al. [6], which consider VM types offered by Amazon EC2 [25] as can be seen in Table 1. Table 2 shows the CSPs used in the simulation with their costs for running instances of each VM type shown in Table 3. For different simulation scenarios, different number of cloud service providers were needed, therefore as the results below will show, not all 12 CSPs where involved in all simulation runs.

Table 1: VM types offered by Amazon EC2 that are used in the simulation.

VM Types / Resources	Small (VM_1)	Medium (VM_2)	Large (VM_3)	Extra Large (VM_4)
Cores (C_i)	1	2	4	8
Memory (M_i)	1.7 GB	3.75 GB	7.5 GB	15 GB
Storage (S_i)	220 GB	480 GB	980 GB	1990 GB
Price	0.12	0.24	0.48	0.96

Table 2: Cloud Service Providers used in the simulation.

Resources / CSP	Cores ($Core_i$)	Memory (Mem_i) GB	Storage (Sto_i) GB
CSP_1	512	87	1120
CSP_2	658	111.8	1440
CSP_3	804	136.6	1760
CSP_4	950	161.4	2080
CSP_5	1096	186.2	2400
CSP_6	1242	211	2720
CSP_7	1388	235.8	3040
CSP_8	1534	260.6	3360
CSP_9	1096	186.2	2400
CSP_{10}	512	87	1120
CSP_{11}	950	161.4	2080
CSP_{12}	1242	211	2720

Table 3: The cost of running an instance of each VM type on each CSP.

	VM_1	VM_2	VM_3	VM_4
CSP_1	0.03	0.06	0.12	0.24
CSP_2	0.045	0.09	0.182	0.364
CSP_3	0.048	0.096	0.0192	0.384
CSP_4	0.033	0.065	0.130	0.260
CSP_5	0.055	0.111	0.222	0.444
CSP_6	0.04	0.08	0.16	0.32
CSP_7	0.058	0.115	0.230	0.460
CSP_8	0.044	0.088	0.175	0.350
CSP_9	0.04	0.08	0.16	0.32
CSP_{10}	0.045	0.091	0.182	0.364
CSP_{11}	0.048	0.096	0.192	0.384
CSP_{12}	0.058	0.115	0.230	0.460

Now with the basic setup explained, we can delve into the workings of the simulation environment. For the centralized model, we set up a broker that accepts client requests. In each round of the simulation the client request vector (generated randomly) is received by the broker, who then federates the request using IP-CFPM explained in Section 3. When the simulation ends, the sum of the total profit obtained by the cloud service providers is recorded as the federation efficiency (FE). To solve the IP, a hill climbing local search algorithm is used.

For the P2P model the general setup is the same, but at the start of each round one CSP is chosen and the client request vector is given to that CSP. This is done randomly but in such a way that larger CSPs have a higher chance of being chosen, reflecting the fact that large CSPs have a bigger customer base and therefore more client jobs will enter the federation mechanism through them. A greedy algorithm based on local search is used here to solve the multidimensional knapsack problem and for the chosen CSP to maximize its individual efficiency. The remaining instances of the request vector are then federated similar to before. Therefore, the FE for the P2P model equals the individual efficiency obtained by the client facing CSP and the FE obtained by federating the remaining VMs.

To simulate the WP2P model, we use the same setup as for the P2P model, but the remaining instances of the request vector are federated based on the updated IP mentioned in Subsection 3.3, where the FedWeight matrix is used in the objective function so as to incentivize federation to CSPs which have previously provided benefit to the federating CSP. Finally to compare individual profits with the case in which federation is not done, the simulation is run in a scenario where CSPs work in a traditional cloud setting, trying to fulfill all client requests by themselves.

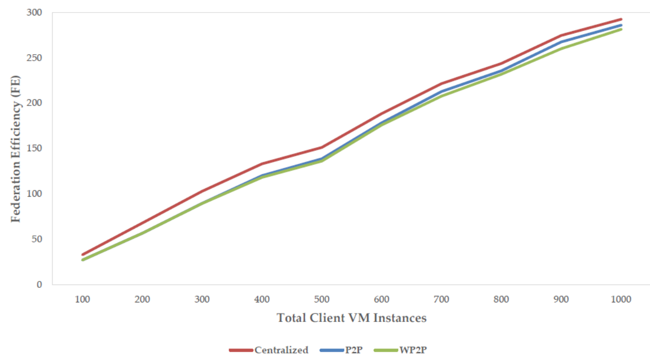

Figure 4: Comparison of Federation Efficiency in the Centralized, P2P & WP2P federation models.

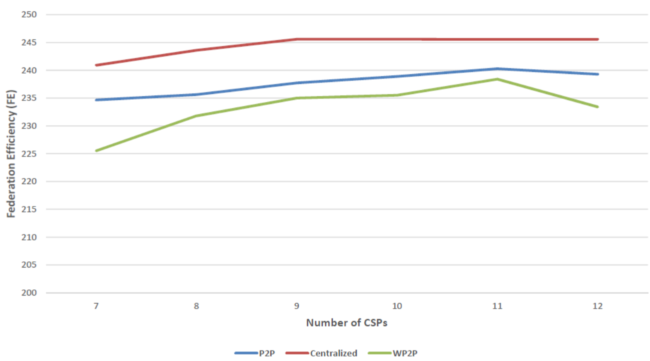

Figure 5: Comparison of Federation Efficiency in the three proposed federation models when the number of CSPs changes. Here 800 total VM instances were requested by the client.

In Figure 4, the federation efficiency of the centralized, P2P and WP2P models are compared. Eight cloud service providers were used here with the FE demonstrating the total

profit obtained by all of them, with the total number of client requests in the range [100; 1000]. As can be seen, although the centralized FE is higher than the P2P and WP2P FE, their difference is very small to the point of being negligible. Therefore one can conclude that overall all three models have very similar efficiency and therefore the decision to use either one can fall on other factors like individual profit, implementation considerations and cooperation limitations.
To further verify this theory, in Figure 5 the request count was kept constant but CSPs were added or removed from the active cloud service providers participating in the federation. As can be seen, just like the previous situation, although centralized federation generally has more efficiency, the difference with both P2P models is small and all methods perform similarly.

Next we try to analyze the effects of these federation models on CSPs' individual profits. To this end we consider the individual profit of a small CSP, medium CSP and large CSP (indexes 1, 4 and 8 from Table 2 respectively) in the various federation scenarios and also when there is no federation. Figures 6, 7 and 8 show these results. As can be seen, all three federation models incentivize the CSPs to participate in the federation mechanism, as their individual profits increase by doing so. The only exception to this rule is in the large CSP, but as can be seen this happens only on light workloads and pretty soon the federation models catch up with and surpass the profits provided to the large CSP when working in a non-federated environment. Therefore no provider has reason to violate the federation mechanism by not cooperating with it.
On the amount of profit obtained by the various federation mechanisms, we can see that centralized federation initially provides better incentives to the smaller CSPs, whilst for the large one this only happens as the number of requests grows. In many places the gap between individual profits gradually decreases and they perform comparably, especially when dealing with larger request counts. Therefore for scenarios in which social welfare cannot incentivize CSPs to collaborate and keeping in mind the previously mentioned violation in which CSPs rather not redirect their customers to a centralized broker, we can see that P2P federation can be an effective alternative solution, providing larger returns for all CSPs than the scenario in which they do not collaborate with each other. This is also the case for the WP2P method, which whilst sometimes providing smaller benefit than other models (for the smaller CSPs) and sometimes larger (for the larger CSPs), always performs better than not federating, even converging to the other models in many places. So we can see that even if the participating clouds take into account selfish methods of federating tasks, this does not violate the federation mechanism.
To sum up, we initially studied what violations can occur in different federation mechanisms and how they can be overcome, presenting the P2P and WP2P federation models based on the original centralized model. As there is still no clear consensus as to which models perform more efficiently and are a better match for federation, describing them in a similar setting and in such a way that one can lend its paradigms and notations to the other can be useful for further research into this area. Taking into account the obtained results, we can see that:

- All models perform comparably in regard to the total amount of profit generated, i.e. the social welfare or federation efficiency of the federation mechanism.
- The centralized model performs better when a federation of smaller (or comparably sized) CSPs would like to collaborate. One reason for this is that in this model the client request is delegated by the broker in such a way so as to maximize overall efficiency, and as smaller providers usually charge lesser rates, they will be chosen more frequently by the broker.
- The P2P & WP2P models can be used when trying to incentivize large cloud service providers to collaborate with smaller ones. This is especially true for the WP2P model, where larger CSPs, because of their higher capacity, obtain better weights on the FedWeight matrix and therefore subsequently receive more federated requests from other cloud service providers.

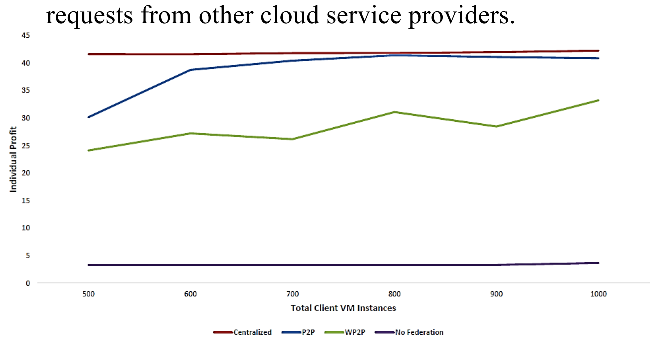

Figure 6: Total amount of profit obtained by a small sized CSP in the Centralized, P2P & WP2P federation models alongside the scenario in which no federation takes place.

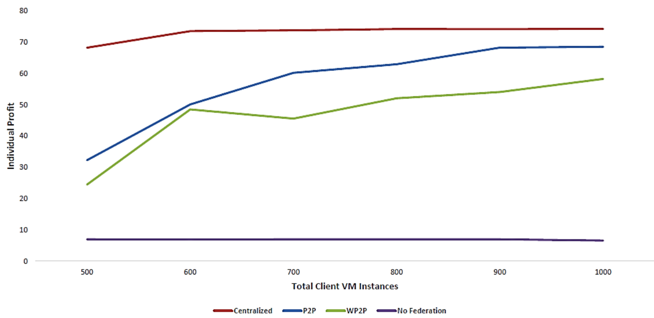

Figure 7: Total amount of profit obtained by a medium sized CSP in the Centralized, P2P & WP2P federation models alongside the scenario in which no federation takes place.

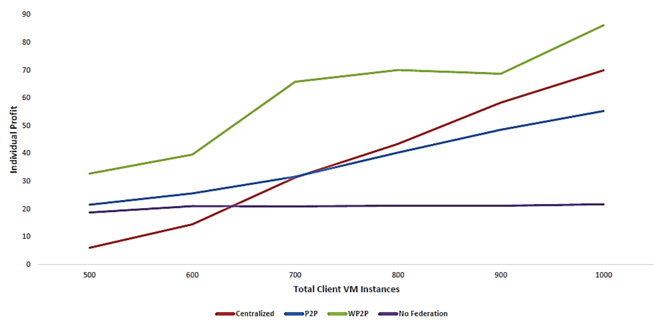

Figure 8: Total amount of profit obtained by a large sized CSP in the Centralized, P2P & WP2P federation models alongside the scenario in which no federation takes place.

- All models provide better returns to the CSPs than a scenario in which no federation occurs.
- In many scenarios, especially larger request counts, individual profits for each CSP are comparable in all presented models.

The proposed P2P models (P2P and weighted P2P) also have advantages that can enable federation even further, such as:

- In a centralized model CSPs must adapt to a new paradigm in which they delegate a portion of their responsibilities to an external entity. There will be resistance to such a change and as CSPs are usually private organizations or companies, this could hamper the use of federation as a practical tool. In the above P2P models not only is such an external interaction unnecessary, but because the CSP receiving the original client request initially acts just as it would in a normal non-collaborative situation, then no extra incentivization would be needed. The federation is done on requests that the CSP cannot handle by itself, and therefore constitute extra profit to be obtained with no risk of change in the CSP's core business model.
- Apart from the CSP receiving client requests, in the normal P2P model the CSPs that are delegated federated requests can be agnostic to the federation taking place, because here they do not need to register to an external broker. Therefore the requests that they receive can be viewed as any other request. For example if a private cloud wants to federate client requests to better provide services to its clients, it can use the resources offered by public clouds without the need for any predetermined agreement or contract. This is similar to the hybrid cloud paradigm [26]. In the WP2P model things are a bit more complicated, as all CSPs must be aware of the federation taking place so as to update the FedWeight matrix.
- The management of a broker-like entity to regulate federation is a challenging task. Not only do we have a single point of failure, but coordinating all federation activities from a central point that must work with varying CSPs (which themselves consist of multiple data centers) is a complex technical and administrative issue. In P2P federation these risks are mitigated, although some of that complex behavior has to be appended to the CSPs themselves. This becomes more challenging in the WP2P model where past interactions must be kept and therefore the CSPs cannot be stateless in regard to the federation mechanism.
- In P2P federation collaboration between CSPs is more fluid and customizable. If for some reason two cloud service providers do not want to work together this does not mean that federation itself will be hampered. As the relations here are peer to peer, any CSP has a choice as to which other CSP it wants to work with. This choice can also be more fine-tuned; for example to federate to a set of CSPs given some circumstance and with other CSPs otherwise. Such decisions cannot be made locally in a centralized federation model. As we saw in the WP2P model such customizations do not necessarily disrupt the federation mechanism, but more work is needed in this regard to obtain optimal local behavior for each CSP based on various internal and external factors.

5. Conclusion and Future Work

In this paper we expanded existing work on a centralized cloud federation model by examining possible violations that could occur in the federation mechanism. To this end we proposed a P2P cloud federation model and a weighted P2P cloud federation model so as to counter such violations. Formally describing these models alongside the centralized one, we used simulation to evaluate their efficiency compared to each other and to the scenario in which no federation is done. The results show that whilst each model is sometimes more suitable for specific scenarios, they nearly provide the same amount of social welfare, and also that in cases like larger request counts, the gap between the CSPs' individual profits in each model lessen.

To further continue this work we plan on expanding our model with real time client workloads where the time dimension of request arrival is also considered. This is in line with how most workloads are processed in the real world. We also plan on injecting quality attributes into our model, as preventing SLA violations and ways to adhere to service level agreements is a must for practical cloud federation. More work is also needed in determining the best method for delegating client requests to other CSPs in the P2P mode and how historical interactions can be used in such a way as to better incentivize cloud service providers to cooperate whilst at the same time providing a fair approach that does not conflict with each CSP's natural selfish behavior.

References

[1] K. Kim, S. Kang, J. Altmann, "Cloud Goliath versus a Federation of Cloud Davids," *International Conference on Grid Economics and Business Models*, Springer, pp. 55-66, 2014.

[2] M. Abdur Razzaque, C. Chen Sheng, "Outsourcing of logistics functions: a literature survey," *International Journal of Physical Distribution & Logistics Management,* vol. 28, no. 2, pp. 89-107, 1998.

[3] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. M. Sadjadi, M. Parashar, "Cloud federation in a layered service model," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330-1344, 2012.

[4] N. Grozev, R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369-390, 2014.

[5] A. N. Toosi, R. N. Calheiros, R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," *ACM Computing Surveys (CSUR)*, vol. 47, no.1, 2014.

[6] L. Mashayekhy, M. M. Nejad, D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14-27, 2015.

[7] D. Breitgand, A. Marashini, J. Tordsson, "Policy-driven service placement optimization in federated clouds," *IBM Research Division, Tech. Rep 9*, pp. 11-15, 2011.

[8] M. M. Hassan, M. S. Hossain, A. J. Sarkar, E.-N. Huh, "Cooperative game based distributed resource allocation in horizontal dynamic cloud federation platform," *Information Systems Frontiers*, vol. 16 no. 4, pp. 523-542, 2014.

[9] A. K. Das, T. Adhikary, M. A. Razzaque, E. J. Cho, C. S. Hong, "A QoS and profit aware cloud confederation model for IaaS service providers," *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, ACM, pp. 42, 2014.

[10] S. Ye, H. Liu, Y.W. Leung, X. Chu, "Reinsurance-emulated collaboration mechanism in cloud federation," *10th International Conference on Cloud Computing (CLOUD)*, IEEE, pp. 727-732, 2017.
[11] B. K. Ray, A. I. Middya, S. Roy, S. Khatua, "Multi-criteria based federation selection in cloud," *9th International Conference on Communication Systems and Networks (COMSNETS)*, IEEE, pp. 182-189, 2017.
[12] M. V. Thomas, K. Chandrasekaran, "Dynamic partner selection in cloud federation for ensuring the quality of service for cloud consumers," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 8, no. 03, p. 1750036, 2017.
[13] B. El Zant, I. Amigo, M. Gagnaire, "Federation and revenue sharing in cloud computing environment," *IEEE International Conference on Cloud Engineering (IC2E)*, IEEE, pp. 446-451, 2014.
[14] L. S. Shapley, "A value for n-person games," *The Shapley value*, pp. 31-40, 1988.
[15] H. Li, H. Lu, Y. Kang, J. Lu, "A Research of Pricing for the Cloud Resource based on the Cloud Bank SLA," *Journal of Network & Information Security*, vol. 4, no. 1, pp. 69-78, 2013.
[16] B.-Y. Chang, B. Kim, S. Yoon, D.-W. Seo, "An evaluation of federated cloud computing effect with service level," *International Conference on Computational Problem-Solving (ICCP),* IEEE, pp. 105-108, 2011.
[17] H. Chen, B. An, D. Niyato, Y. C. Soh, C. Miao, "Workload factoring and resource sharing via joint vertical and horizontal cloud federation networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 557-570, 2017.
[18] M. R. Assis, L. F. Bittencourt, "A survey on cloud federation architectures: identifying functional and non-functional properties," *Journal of Network and Computer Applications*, vol. 72, pp. 51-71, 2016.
[19] B. Rochwerger, D. Breitgand, E. Levy, A. Maraschini, P. Massonet, H. Munoz, G. Toffetti, A. Epstein, D. Hadas, I. Loy, et al., "Reservoir-when one cloud is not enough," *Computer*, vol. 44, no. 3, pp. 44-51, 2011.
[20] R. Buyya, R. Ranjan, R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," *Algorithms and architectures for parallel processing*, Springer, pp. 13-31, 2010.
[21] B. El Zant, M. Gagnaire, "New pricing policies for federated cloud," *6th International Conference on New Technologies, Mobility and Security (NTMS),* IEEE, pp. 1-6, 2014.
[22] S. Flake, J. Tacken, C. Zoth, "Real-time rating and charging in federated cloud environments," *17th Conference on Emerging Technologies & Factory Automation (ETFA)*, IEEE, pp. 1-6, 2012
[23] N. Samaan, "A novel economic sharing model in a federation of selfish cloud providers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 12-21, 2014.
[24] T. Lust, J. Teghem, "The multiobjective multidimensional knapsack problem: a survey and a new approach," *International Transactions in Operational Research*, vol. 19, no. 4, pp. 495-520, 2012.
[25] Amazon.com-Inc., AWS | Amazon Elastic Compute Cloud (EC2) – Scalable Cloud Hosting, http://aws.amazon.com/ec2/, July 2016
[26] A. Srinivasan, M. A. Quadir, V. Vijayakumar, "Era of cloud computing: A new insight to hybrid cloud," *Procedia Computer Science*, vol. 50, pp. 42-51, 2015.

Moslem Habibi received his BSc and MSc degree in software engineering from Sharif University of Technology, in 2010 and 2012, respectively. He is currently a PhD candidate in software engineering at Sharif University of Technology, where his research centers on modeling economic incentives for cloud federation. His research interests include cloud computing, cloud interoperability, cloud computing economic models and complex networks.
Email: moslem_habibi@ce.sharif.edu

Hamid Haghshenas received his BSc and MSc degrees in software engineering from Sharif University of Technology, in 2010 and 2012, respectively. Currently, he is a PhD candidate in software engineering at Sharif University of Technology, focusing on cloud federation architectures. His research interests include cloud computing, cloud federation, software architecture and discrete mathematics.
Email: haghshenas@ce.sharif.edu

Ali Movaghar is a Professor at the Department of Computer Engineering at Sharif University of Technology in Tehran, Iran, and has been in the Sharif faculty since 1993. He received his B.S. degree in Electrical Engineering from the University of Tehran in 1977, and M.S. and Ph.D. degrees in Computer, Information, and Control Engineering from the University of Michigan, Ann Arbor, in 1979 and 1985, respectively. He visited the Institut National de Recherche en Informatique et en Automatique in Paris, France and the Department of Electrical Engineering and Computer Science at the University of California, Irvine in 1984 and 2011, respectively, worked at AT\&T Information Systems in Naperville, IL in 1985-1986, and taught at the University of Michigan, Ann Arbor in 1987-1989. His research interests include performance/dependability modeling and formal verification of wireless networks and distributed real-time systems. He is a senior member of the IEEE and the ACM.
Email: movaghar@sharif.edu

Paper Handling Data:

Submitted: 27.05.2018
Received in revised form: 21.08.2018
Accepted: 27.08.2018
Corresponding author: Moslem Habibi
Computer Engineering Department, Sharif University of Technology, Tehran, Iran