This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A Novel Economic Sharing Model in a Federation of Selfish Cloud Providers

Nancy Samaan, *Member, IEEE*

Abstract—This paper presents a novel economic model to regulate capacity sharing in a federation of hybrid cloud providers (CPs). The proposed work models the interactions among the CPs as a repeated game among selfish players that aim at maximizing their profit by selling their unused capacity in the spot market but are uncertain of future workload fluctuations. The proposed work first establishes that the uncertainty in future revenue can act as a participation incentive to sharing in the repeated game. We, then, demonstrate how an efficient sharing strategy can be obtained via solving a simple dynamic programming problem. The obtained strategy is a simple update rule that depends only on the current workloads and a single variable summarizing past interactions. In contrast to existing approaches, the model incorporates historical and expected future revenue as part of the VM sharing decision. Moreover, these decisions are not enforced neither by a centralized broker nor by pre-defined agreements. Rather, the proposed model employs a simple grim trigger strategy where a CP is threatened by the elimination of future VM hosting by other CPs. Simulation results demonstrate the performance of the proposed model in terms of the increased profit and the reduction in the variance in the spot market VM availability and prices.

Index Terms—Cloud federation; cloud provider; capacity outsourcing; repeated game; subgame perfect equilibrium

1 Introduction

Cloud computing is an emerging paradigm that substantiates the vision of commodifying computational power, storage and software services [1], [2]. In such a vision, software applications of different clients are executed over the shared cloud which offers its infrastructure as-a service (IaaS). Yet, all applications run in complete isolation through virtual execution environments or virtual machine (VM) instances which are seamlessly launched and terminated on the cloud data centers to host applications of cloud clients on a per-needed basis. Clients of an IaaS cloud are mostly service providers ranging from small-scale companies to world-wide enterprises and web-service providers but can also represent single users. Unfortunately, one of the major problems that face the cloud providers (CPs) is the uncertainty in their workloads; a spike in the requested VMs may result in higher service rejection rates and experienced delays by clients due to congested resources. A straightforward solution to overcome this problem is to over-provision the available resources to be able to meet their peak demands. Yet, this solution may lead to highly under-utilized capacity during other periods of low demands. A more efficient solution, followed by hybrid clouds, is to only guarantee the availability of VMs to a limited number of clients and plan the cloud capacity to meet their peak workloads [3]. The CP would then offer the spare capacity during low utilization periods to second-class clients in a spot market [4]. These clients would take advantage of a reduced prices but with no service availability guarantees. This model has been successfully adopted by a number of existing CPs such as Amazon EC2 [5]. Nonetheless, demand fluctuations of guaranteed-service clients, reflect similar variations in spot VM availability, and, in turn, results in non-predictable request delays, rejections and termination as well as price fluctuations [6].

Federated clouds [7], depicted by Fig. 1, approach the CP problem by allowing peer CPs to share their unused capacities during low demand periods and borrow spare capacity during peaks, to maximize their profits and enhance their clients' experience [8], [9], with several strategies for capacity sharing in the federation proposed in [4], [8], [10]). These efficient schemes help the federated CPs to decide on outsourcing capacity from other CPs by launching or migrating VMs on their servers or insourcing capacity by hosting VMs of clients of other CPs. A number of these sharing mechanisms employ either pre-defined or dynamic pricing rules (e.g., [7]) to regulate the VM hosting exchange among CPs, mostly, with the objective of maximizing their instantaneous revenues.

To the best of the author's knowledge, the work presented in this article is the first to address the general problem of maximizing the CP's long-term revenue where current capacity sharing decisions in the federation depend on the revenue obtained from previous sharing decisions as well as on the expected revenue in the future. To this end, the contributions of the proposed work can be summarized as follows:

- We derive the capacity sharing strategies that maximize the long-run revenue of the federation, dubbed as *socially optimal spot market allocations*, and demonstrate their enforcement limitation.
- Using a formulation based on multistage games, we introduce a set of *self-enforceable* CPs capacity sharing strategies that maximize the federation's long-term revenue yet can achieve more revenue than what the individual CP can achieve outside the federation. We prove that the profit obtained from the aforementioned repeated game can derive a higher revenue using a simple grim trigger

Digital Object Indentifier 10.1109/TPDS.2013.23

punishment strategy.

- We derive a simple update rule to find the *subgame perfect Nash equilibrium* values for the spot market allocations. This rule is only dependent on the observed current state and a single variable that summarizes the history of previous interactions. We also develop a simple dynamic program that can be employed to obtain the exact values for these allocations in practice.

The remainder of the paper is organized as follows. Section 2 discuses related approaches in the literature. The capacity sharing problem of the cloud federation is formulated in Section 3. Section 4 is dedicated to analyzing the federation's socially optimal revenue and the individual CPs revenues. The proposed repeated game model is analyzed in Section 5. A recursive formulation of the optimal strategies is then introduced in Section 6 while Section 7 discusses how to numerically obtain these strategies. Finally, the performance of the proposed mechanism is demonstrated in Section 8 and Section 9 concludes the paper.

2 Related work

Early approaches to a model of a cloud federation can be found in [7], [8], [11]. Buyya et al. [1] introduced a market-oriented VM exchange model among clouds. The proposed work complements these approaches by deriving closed form formulations for the sharing strategies among the CPs that, if followed, leads to optimal revenues. Federations of hybrid clouds were also discussed in [10], [12]. Goiri et al. [10] proposed a novel federation management architecture and focused on building an accurate revenue function of CPs when taking one of the following decisions: participate in the federation by outsourcing or in-sourcing capacity or turning off spare capacity. Similarly, Toosi et al. [12] provided a comprehensive analysis of the related costs and revenue associated with the various decisions of the CP in the federation.

A mechanism to dynamically allocate resources of distributed data centers among different spot markets with the objective of maximizing the total revenue is introduced in [13]. Similar to the proposed approach, the authors necessitate the need to model or forecast the CPs demand. It can be seen that their developed mechanism is similar to the fully-federated model discussed in Section 4. A market-clearing pricing mechanism is developed in [14] where a centralized broker dynamically adjusts a single VM price for the federation. The proposed model does not assume any specific pricing scheme for the spot markets, and can employ that of [14] following resource allocation.

Mihailescu et al. [15] relied on a centralized market broker in the federation, to which sellers publish resources and buyers send requests. Le et al. [9] addressed the cooperation problem among CPs but from the perspective of load balancing and electricity consumption. On the other hand, Lee et al. [16] focused on achieving a better quality of service (QoS) by the CPs. The problem of allocating the appropriate cloud provider when considering tasks with deadline constraints is presented in [17]. Finally, the work in [18] addresses the consumer's problem of selecting a VM reservation plan or request resources on demand with the objective of reducing the service cost.

In general, existing approaches in the literature are concerned only with the instantaneous CP gains. To the best of the author's knowledge, the presented work is the first to take into consideration the outcome of historical and future interactions among the CPs in making sharing decisions in the federation. This new generalization leads, as will be shown by the simulation results, to a higher accumulated profit by the CPs. The model can also be extended to incorporate additional constraints for each CP to account for different issues (e.g., energy consumption).

Economic theories have also been adapted to the context of resource allocation in the literature (e.g., [19], [20]). However, most of the attention of these approaches has been focused on finding efficient pricing strategies or techniques for solving the centralized optimization problem of utility maximization in a decentralized manner. For example, [21] describes a model of resource seller and buyer agents which learn from previous resource trades to decide the exchanged prices for the current resource request.

To the best of the author's knowledge, the presented work is the first to address the problem of the federated CPs long-term revenue maximization given future workloads uncertainty.

3 Problem Formulation

As shown in Fig. 1, we consider a federation $\mathcal{N}$ of $|\mathcal{N}| = N$ hybrid cloud providers (CPs), $\text{CP}_1, \cdots, \text{CP}_N$. Each CP, CP_i, owns a fixed amount of computational resources (e.g., CPU and storage) which are sold to clients as instances of virtual machines (VMs) that are hosted on the CP's data centers. For simplicity, we will focus on a single VM type, with C_i denoting the number of VM instances that can be hosted by CP_i. This assumption can be easily relaxed by either assuming a different market for each VM configuration or by adopting a more elementary resource measure such as the Amazon EC2 compute units [3], [5] which are used as the building units for different types of VMs. Each CP services two classes of clients [22], namely, guaranteed-service and spot market clients. Those in the first class pay up-front fees and have pre-defined contracts with the CP so that they are always guaranteed the allocation of their requested VMs. On the other hand, spot market clients are public-users who are allowed access to the unused computing resources over the Internet through remote interfaces (e.g., web-services) to create and manage VM instances. This model is similar to Amazon's guaranteed and spot market service classes [5]. These public users can utilize the spare capacity after assigning sufficient VM instances to satisfy the guaranteed-service requests. We assume a discrete time horizon, $t = 1, 2, \cdots,$

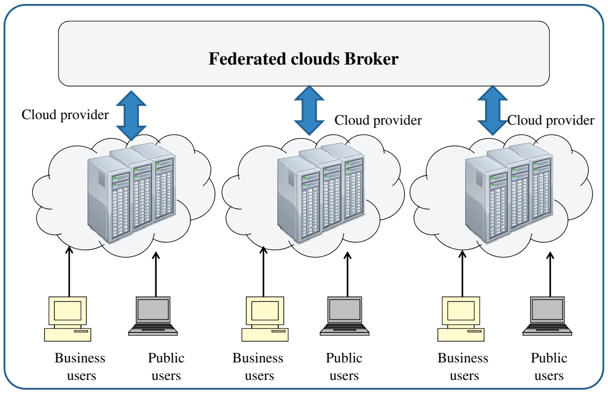


Fig. 1. The adopted model of the federated clouds

where at the beginning of each time period t (e.g., t can represent a one hour period [5]), CP$_i$ observes the total demand of the guaranteed-service clients $d_i(t)$, and, in turn, determines the remaining capacity, $e_i(t) = C_i - d_i(t)$. CP$_i$ is then faced with the problem of determining the number of VMs to offer on the spot market, $w_i(t)$, and that to offer for sharing in the federation, $e_i(t)-w_i(t)$, with the difference $e_i(t)-w_i(t)$ being negative whenever CP$_i$ is outsourcing VMs from the federation.

As shown by Fig. 1, the VM exchange is coordinated through the federation broker which performs the actual VM allocation after the CPs have agreed on sharing them. The broker relies on the existence of a virtual infrastructure management architecture (e.g., the OpenNebula manager and its external-resource lease manager Haizea [23]) that manages the VMs life cycles and encompasses the necessary means for the configuration of the resources.

Offering VMs to host the federation tasks is free of charge, hence, no profit is obtained from $e_i(t) - w_i(t)$. On the other hand, the revenue obtained from offering $w_i(t)$ on the spot market is $r_i(w_i(t))$=$w_i(t)P(w_i(t))$ where $P_i(w_i(t))$ is the inverse of the spot market demand function, i.e., the expected price of a VM when there are $w_i(t)$ VMs offered in the spot market of CP$_i$. No specific pricing model is imposed, however, similar to other approaches (e.g., [22]), it is assumed that $r_i(w_i(t))$ is increasing, concave and twice continuously differentiable with $r_i(0) = 0$. Next, we use $r_i(w_i(t))$ to formally define the CP's marginal revenue.

Definition 1. *The marginal revenue of CP$_i$, $r'_i(w_i(t))$, is the additional revenue generated by an additional VM instance in its spot market, i.e.,* $r'_i(w_i(t))$=$\frac{\partial r_i(w_i(t))}{\partial w_i(t)}$.

We note that the spot market client is allowed to retain the requested VM as long as the consumed resources are not needed by the CP. Hence, at the beginning of each epoch t, CP$_i$ may terminate a spot VM instance and possibly direct its resources to serve its guaranteed-service clients or the spot market clients of other CPs in the federation, without incurring any additional cost. The CP makes this decision only if it is more profitable to contribute in the federation. The advantage to this approach is that it alleviates the CPs' need to migrate these shared spot market VMs.

Nonetheless, this simple model can be easily modified to allow a CP to maintain spot market VM instances whenever it has enough resources borrowed from other CPs by migrating them to the other CPs' clouds. In this case, the revenue function must be modified to account for the incurred migration cost, denoted by $c_i(w_i(t) - e_i(t))$, of the borrowed $w_i(t) - e_i(t)$ VMs. The cost function is defined such that $c_i(w_i(t) - e_i(t)) = 0$ for $w_i(t) \leq e_i(t)$ and is convex otherwise. In this case, we have $r_i(w_i(t))$= $w_i(t)P(w_i(t))-c(w_i(t)-e_i(t))$, which remains concave due to the convexity of $c_i(.)$.

3.1 A Markovian model of spot market resources

Since the majority of guaranteed-service clients operate business applications that exhibit strong temporal and spatial correlations, the CPs can easily characterize their expected VM demands by monitoring their behavior over a period of time [18], [24]–[27]. By predicting the guaranteed-service clients demands, the CP can also estimate the unused capacity facilitating the decision of selecting $w_i(t)$ and the shared portion $e_i(t) - w_i(t)$. Motivated by recent advances in Markovian modeling of the expected workloads (for example, see [24], [25] and [28]), we model the transition of the CPs spare capacities as follows.

At each period $t = 1, 2, \cdots$, the observed unused capacity of the N CPs is described by a state vector $s_t = (e_1(s_t), \cdots, e_N(s_t))$. s_t is drawn from a, possibly large, finite set $\mathcal{S} = \{s_1, \cdots, s_{|\mathcal{S}|}\}$ and its evolution follows a Markov process with a probability of transition from state $s \in \mathcal{S}$ at $t-1$ to a state $s' \in \mathcal{S}$ at t given by $\pi(s'|s)$=$Pr(s_t = s'|s_{t-1} = s)$.

The cloud federation undergoes a change in the workloads and, in turn, the spare capacity in discrete time epochs, according to a first order Markov chain described by the state transition matrix $\Pi =$ $[\pi(s_i|s_j)]$, $i, j = 1, \cdots, |\mathcal{S}|$, with the state transition coefficients having the properties: $\pi(s_i|s_i) \geq 0$ and $\sum_{j=1}^{|\mathcal{S}|} \pi(s_j|s_i) = 1$. With a slight abuse of notation, s_t will be used to denote the actual state observed at t and we will use $h_t = (s_1, s_2, \cdots, s_t)$ to denote a history of state observation up to time t. Moreover, if we know the state s_t or history h_t at t, we will refer to the spot market allocation as $w_i(s_t)$ and $w_i(h_t)$, respectively. We also note that, unused capacities at s_t are history independent, i.e., $e_i(h_t)$= $e_i(s_t)$, $\forall i$. Moreover, because of the Markov property,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

we can see that the conditional probability $\pi(h_\tau|h_t)$ for $\tau > t$ is only dependent on the last observed state s_t in h_t and does not depend on the history before t, i.e., $\pi(h_\tau|h_t) = \pi(s_\tau|s_{\tau-1})\pi(s_{\tau-1}|s_{\tau-2})\cdots\pi(s_{t+1}|s_t)$. Finally, we note that the presented work can easily be adapted to include other Markovian workload models (e.g., Markovian Arrival Processes [24]).

4 Fully- and Non-Federated Models

4.1 Limitation of the fully-federated model

Here, a centralized broker (Fig. 1) is tasked with redistributing the unused capacities among CPs' spot markets with the objective of maximizing the federation's total revenues. Termed as *socially optimal spot market allocation*, this problem is formulated as follows

Definition 2. *The socially optimal spot market allocation, $w_1(h_t), \cdots, w_N(h_t)$, after each history h_t, is the one that maximizes the net federation revenue without regard to the individual CPs gains, i.e., it solves:*

$$\textbf{P1:}\ \max_{w_i(h_t)} \sum_{i=1}^{N} \lambda_i \sum_{t=1}^{\infty} \delta^t \sum_{h_t} \pi(h_t) r_i(w_i(h_t)) \quad (1)$$

$$\text{s.t. ``capacity constraint:''} \sum_{i=1}^{N} w_i(h_t) \leq \sum_{i=1}^{N} e_i(s_t), \quad (2)$$

$\forall h_t, \forall t$, where $\delta \in (0, 1)$[1] is the future discount factor and λ_i is the normalized exogenous constant weight of CP$_i$ in the federation such that $\sum_{i=1}^{N} \lambda_i = 1$.

We note that, the weights λ_i can be used to express some service characteristics such as reliability or offered service quality. The following proposition fully characterizes the solution of **P1**.

Proposition 1. [2] *A VM allocation, $w_1(h_t), \cdots, w_N(h_t)$, in the CPs spot markets is the socially optimal allocation for the federation if all VMs are allocated in the spot markets, i.e., $\sum_{i=1}^{N} w_i(s_t)=\sum_{i=1}^{N} e_i(s_t)$, and for any state $s_t \in \mathcal{S}$, the ratios of the marginal revenues, remain constant for any CP$_i$, CP$_j$ $i, j = 1\cdots, N$, $i \neq j$, after any history of workload states $h_{t-1} = (s_1, \cdots, s_{t-1})$, such that:*

$$\frac{r'_i(w_i(h_t))}{r'_j(w_j(h_t))} = \frac{r'_i(w_i(s_t))}{r'_j(w_j(s_t))} = \frac{\lambda_j}{\lambda_i}, \forall h_t, \forall t \quad (3)$$

The relation in (3) is typically used to construct a system of $N - 1$ equations, that along with (2) uniquely determines the solution of **P1**. Note that the share of VMs for CP$_i$, $w_i(s_t)$, depends only on its relative weight λ_i w.r.t. to the other CPs' weights and is independent of the history of interactions between the CPs before t. The above proposition indicates that the ratios of the CPs' marginal revenues at each t is always fixed regardless of the current state which determines the number of unused VMs for each CP. For example, in the case where all CPs share the same function $r(.)$ and weights $\lambda_i = \lambda$, the amount of unused VMs will be distributed equally on their spot markets, $w_i = w_j$, $i, j = 1, \cdots, N$, at each state.

Unfortunately, while this allocation guarantees the maximum revenue for the federation, there is no incentive for individual CPs to cooperate. This situation is most stringent particularly during off-peak periods of the CP where it can achieve a much higher revenue by offering a large number of its VMs in its spot market. Hence, rational CPs aiming at maximizing their profit cannot be enforced to cooperate. More precisely, this sharing rule cannot be enforced if there exists any state s_t with at least one CP, say CP$_i$, having an unused capacity $e_i(s_t)$ such that:

$$r_i(e_i(s_t)) - r_i(w_i(s_t)) > \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \sum_{h_\tau} \pi(h_\tau|s_t) \left(r_i(w_i(h_\tau)) - r_i(e_i(h_\tau))\right) \quad (4)$$

In Section 5, a new dynamic sharing rule will be derived to overcome the limited enforcement problem of the above model.

4.2 Limitation of the non-federated model

In this section, we regard the sharing problem from the perspective of the individual CPs and characterize the maximum revenue they can achieve outside the federation. Each CP, CP$_i$, aims at maximizing its long-term revenue by solving the following problem:

$$\textbf{P2:}\ \max_{w_i(h_t)} \sum_{t=1}^{\infty} \delta^t \sum_{h_t} \pi(h_t) r_i(w_i(h_t)) \quad (5)$$

subject to the same capacity constraint of (2).

Given the uncertainty in the sharing behavior of the other CPs, and since $r_i(.)$ is monotonically increasing, it follows that the solution to **P2** is simply given by setting $w_i(h_t) = e_i(s_t)$. The long-run revenue, $\underline{R}_i(s_t)$, obtained from selling $e_i(s_t)$ VMs in each state s_t, after observing s_t at time t, can be formulated as:

$$\begin{aligned}\underline{R}_i(s_t) &= r_i(e_i(s_t)) + \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \sum_{h_\tau} \pi(h_\tau|s_t) r_i(e_i(h_\tau)) \\ &= r_i(e_i(s_t)) + \delta \sum_{s_{t+1}\in\mathcal{S}} \pi(s_{t+1}|s_t) \underline{R}_i(s_{t+1}) \quad (6)\end{aligned}$$

The model described by **P2** guarantees the maximum long-term revenue for a CP outside a federated model since it will offer all available VMs in the spot market. However, the revenue gain is not necessarily higher than what could have been obtained had it opted to work within the federation. In addition, the above solution does not achieve the best performance in terms of a smooth spot VM availability and prices since the VM availability will depend only on the usage pattern of the guaranteed-service users.

1. Typically, δ reflects the CPs' interest in future profits, where a value approaching zero reflects no interest in future profits.
2. All proofs can be found in Appendix B

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

4.3 A Game-theoretic Framework

The proposed approach is motivated by the observation that the behavior of the CPs in the above two models represents two extreme forms of a strategy adopted by players in a game of sharing unused VMs. Recognizing this fact enables us to reformulate the problem in a more general setting that alleviates the limitations of the above two models.

In game theory [29], a stage game is typically defined by a triplet consisting of a set of players, strategies and payoffs, where the players are assumed to be rational agents aiming at maximizing their payoffs. In our setting, the set of N CPs represents the players and the strategies are represented by the allocations of VMs to the spot markets, $(w_i(s_t))_{i=1}^N$, while the revenue functions $(r_i(w_i(s_t)))_{i=1}^N$ represent their payoffs. Hence, at each time epoch, t, and after observing the current state, $s_t = (e_1(s_t), \cdots, e_N(s_t))$, the CPs engage in the game $\langle \mathcal{N}, (w_i(s_t))_{i=1}^N, (r_i(w_i(s_t)))_{i=1}^N \rangle$ by deciding the best strategies to maximize their revenues. The game solution represented by the player's strategies where no player has an incentive to deviate from its chosen strategy after considering all other players' strategies is called a *Nash Equilibrium* (NE) [29].

Definition 3. *A VM allocation decision, $(w_i(s_t))_{i=1}^N$ at a state s_t is a NE if every CP cannot increase its revenue by unilaterally changing its spot market allocation, i.e., there is no other feasible allocation $(w_1(s_t), \cdots, \grave{w}_i(s_t) \cdots, w_N(s_t))$, such that $r_i(\grave{w}_i(s_t)) > r_i(w_i(s_t))$.*

Clearly, the solution to **P2** represents a strategy in which every CP shares nothing with the federation. Since no CP gains from lending its resources, this strategy, i.e., $w_i(s_t)$=$e_i(s_t)$ $\forall i$, is the unique NE solution to this game. Next, a more general formulation which can lead to higher revenues while avoiding the enforcement limitation of **P1** and the long-term revenue loss in that of **P2** is presented.

5 SELF-ENFORCING COOPERATION

The proposed repeated game mechanism for VMs sharing borrows from the economic model of risk-sharing introduced by Ligon et al. [30], [31]. Using this model, we demonstrate that if the CPs engage in a repeated sequence of the previously described stage game, where their strategies become history-dependent, they can achieve a long-term revenue that is always better than that obtained by (6). Hence, overcoming the limitation of the non-federated model. The obtained results can be intuitively understood as follows; CPs who have excess VMs during periods of low-demand of guaranteed-services class can voluntarily share $e_i(t) - w_i(t)$ to build a model of informal insurance against the risk of future peaks in the workload. Such an informal insurance, commits other CPs, already borrowing some VMs to host their jobs from that CP, to play the reverse role when they have spare resources. While such an informal contract is not explicitly enforced, the CPs with higher capacity at each state are bound to commit in these interactions due to the risk of future peak workloads and with the threat of a simple grim trigger punishment strategy [31]. This strategy threatens a CP which refuses to share spare VMs, when it is supposed to, by allowing all other CPs to revert to the non-cooperative NE strategy, outlined in the previous section, for the rest of the game. Hence, the result of non-cooperation at one period is that the CP's future revenue will be reduced to (6) thereafter. In this manner, the problem of lacking the sharing enforcement encountered by the fully-federated approach, introduced in Section 4, is also solved in the proposed mechanism.

Moreover, as will be demonstrated, this sharing mechanism can be carried out in a distributed manner, where each CP voluntarily determines the amount of VMs to contribute to or lend from the federation. The broker needs only to perform the actual allocation of VMs. The already sketched game will be now described in more details.

5.1 Subgame perfect spot allocations

In the repeated game setting, the CPs are interested in finding a *subgame-perfect Nash Equilibrium (SPNE)*.

Definition 4. *A subgame-perfect Nash Equilibrium (SPNE) is a strategy that is a NE for the CPs in every subgame (i.e, up to any history h_t) of the original game.*

Clearly, the solution to the non-federated model described by (**P2**), where each CP offers all its VMs in the spot market, i.e., $w_i(h_t) = e_i(s_t)$, is also an SPNE. In what follows, we aim at finding a better SPNE that guarantees that each CP's revenue is at least equal to, if not better than, its non-federated payoff.

In other words, we search for an SPNE strategy that should determine the amount of VMs to offer in the spot market $w_i(h_t)$ at time t, and to share in the federation $e_i(s_t) - w_i(h_t)$ after observing a history of states h_t. The goal of this strategy is to maximize the Federation's expected revenue while guaranteeing the participation of the CPs, or, in other words, while ensuring the self-enforcement of these strategies. This is achieved by ensuring that each CP's long-term revenue is at least equal to that obtained if it opted-out of the federation. Mathematically, this problem can be formulated as follows.

$$\textbf{P3}: \max_{w_i(h_t)} \sum_{i=1}^{N} \lambda_i \sum_{t=1}^{\infty} \sum_{h_t} \delta^t \pi(h_t) r_i(w_i(h_t)) \tag{7}$$

subject to

$$r_i(w_i(s_t)) + \sum_{\tau=t+1}^{\infty} \sum_{h_\tau} \delta^{\tau-t} \pi(h_\tau|h_t) r_i(w_i(h_\tau)) \geq \underline{R}_i(s_t), \tag{8}$$

$\forall i, \forall s_t, \forall h_t$ and the capacity constraints (2).

We refer to the constraints defined in (8) as the federation *commitment constraints*. We also say that a commitment constraint for CP_i at state s_t *binds* if it is satisfied with strict equality. The following proposition characterizes the solution to **P3**.

Proposition 2. *A VMs sharing strategy,* $(w_i(h_t))_{i=1}^N$*, in the infinitely repeated game* $\langle \mathcal{N}, (w_i(s_t))_{i=1}^N, (r_i(w_i(s_t)))_{i=1}^N \rangle$ *that solves* **P3** *is an SPNE.*

Solving **P3** can be carried out by first considering its Lagrangian which is given by,

$$\sum_{t=1}^{\infty}\sum_{h_t}\Big[\delta^t \pi(h_t)\sum_{i=1}^{N}\lambda_i r_i(w_i(h_t)) + \zeta(h_t)\sum_{i=1}^{N}(e_i(h_t) - w_i(s_t)) + \eta_i(h_t)\Big(\sum_{\tau=t}^{\infty}\sum_{h_\tau}\delta^{\tau-t}\pi(h_\tau|s_t)r_i(w_i(h_\tau)) - \underline{R}_i(s_t)\Big)\Big].$$

Here, $\zeta(h_t) \geq 0$ denotes the multiplier on the capacity constraints (2) at state s_t after history h_t and $\eta_i(h_t) \geq 0$ denotes the multiplier on the commitment constraint of CP_i at h_t. To proceed with the solution, noting that $\pi(h_\tau) = \pi(h_\tau|s_t)\pi(s_t)$, the first-order condition with respect to $w_i(h_t)$ can be written as,

$$\delta^t \pi(h_t) r_i'(w_i(h_t))\Big(\lambda_i + \sum_{\tau=1}^{t}\frac{\eta_i(h_\tau)}{\delta^\tau \pi(s_\tau)}\Big) = \zeta(h_t) \quad (9)$$

which yields the following ratio between the CPs' marginal revenues:

$$\frac{r_i'(w_i(h_t))}{r_j'(w_j(h_t))} = \frac{\lambda_j + \sum_{\tau=1}^{t}\frac{\eta_j(h_\tau)}{\delta^\tau \pi(s_\tau)}}{\lambda_i + \sum_{\tau=1}^{t}\frac{\eta_i(h_\tau)}{\delta^\tau \pi(s_\tau)}} \equiv \frac{\lambda_j(h_t)}{\lambda_i(h_t)}, \quad (10)$$

$i, j = 1, \cdots, N$, such that $\sum_{i=1}^N \lambda_i(h_t) = 1$. The obtained $\lambda_i(h_t)$ represents the new normalized relative weight of CP_i in the federation after a history h_t. If $\lambda_i(h_t)$ is known for $i = 1, \cdots, N$, then a solution to **P3** using (10) along with the capacity constraint (2) can be obtained. According to Proposition 2, this solution represents the SPNE strategies for the CPs.

In contrast to the fixed ratio between the marginal revenues (3) obtained for **P1**, the solution in (10) shows that, in the proposed mechanism, the ratio is dynamically updated based on the history of interactions among the CPs during previous states $s_1, s_2, \cdots, s_{t-1}$ up to s_t. More precisely, the ratio of the marginal revenues between any two CPs, CP_i and CP_j at each time, $\frac{\lambda_j(h_t)}{\lambda_i(h_t)}$, is determined by the initial exogenous weights, λ_i and λ_j, along with a trail of multipliers $\eta_k(h_1), \eta_k(h_2), \cdots, \eta_k(h_t)$, $k = 1, \cdots, N$, for all the observed states up to t. There are two possible cases for each $\eta_i(h_t)$, either $\eta_i(h_t) = 0$ which means that the corresponding constraint is relaxed or $\eta_i(h_t) > 0$, reflecting a satisfied constraint with strict equality. The first case (i.e., $\eta_i(h_t) = 0$) means that at s_t the obtained $w_i(h_t)$ makes the left hand side of (8) strictly larger than $\underline{R}_i(s_t)$. This can happen when there is an allocation $w_i(h_t)$ that is larger than the actual unused capacity $e_i(s_t)$. This must correspond to CP_i borrowing VMs from some CP_j. In turn, this situation must be reducing CP_j's revenue up to the lowest permissible value that allows CP_j to participate in the federation. CP_j's spot share $w_j(h_t)$ in this case must be such that its expected revenue does not decrease below the revenue obtained outside the federation, $\underline{R}_j(s_t)$. Hence, in this case $w_j(h_t)$ will satisfy (8) with strict equality with the corresponding multiplier $\eta_j(h_t) > 0$. The second case of $\eta_i(h_t) > 0$ with some other CP_j having $\eta_j(h_t) = 0$ reflects the same scenario with the borrowing and lending roles of CP_i and CP_j reversed. A third possible scenario is where all CPs, CP_i, $i = 1 \cdots, N$, are in a state s_t were their current shares ensure a revenue larger than (6), due to cooperation, by being given a promise of better future revenues, and we have $\eta_i(h_t) = 0$, $i = 1, \cdots, N$. Finally, the case where both $\eta_i(h_t), \eta_j(h_t)$ are positive must involve at least another CP, CP_k with $\eta_k(h_t) = 0$ borrowing resources from CP_i and CP_j.

As these multipliers accumulate along the history of interactions h_t, they clearly guide the sharing among the CPs to maintain the satisfaction of their commitment constraints and hence, the self-enforceability of the SPNE strategies. Appendix A presents an illustrative example of the interactions among two CPs.

The next section demonstrates how a simple update rule can be employed to summarize CP_i's efficient SPNE strategy through a recursive formulation.

6 Recursive Formulation

To obtain a solution for **P3**, the ratios of the marginal revenues in (10), $\frac{\lambda_j(h_t)}{\lambda_i(h_t)}$, must be computed and employed to generate a system of $N-1$ equations for the unknowns $w_i(h_t)$. Combining those equations with the capacity constraint at equality (2) provides a system whose unique solution would serve as the strategy adopted by the CPs. The difficulty with this approach, however, is that the problem becomes quickly intractable due to the exponential growth of the dimension of the histories h_t.

Fortunately, using the Markov properties of the workload, this problem can be cast as a recursive one, which yields a simple update rule to compute ratios of the CPs' marginal revenues at t, $\frac{\lambda_j(h_t)}{\lambda_i(h_t)}$. We first begin with the derivation of the update rule.

By the definition of the normalized dynamic weights, we have

$$\lambda_i(h_1) = \frac{\lambda_i + \frac{\eta_i(h_1)}{\delta\pi(s_1)}}{\sum_{j=1}^N(\lambda_j + \frac{\eta_j(h_1)}{\delta\pi(s_1)})} = \frac{\lambda_i + \frac{\eta_i(h_1)}{\delta\pi(s_1)}}{1 + \sum_{j=1}^N \frac{\eta_j(h_1)}{\delta\pi(s_1)}} \quad (11)$$

then $\lambda_i(h_t)$ can also be defined recursively by

$$\lambda_i(h_t) = \frac{\lambda_i(h_{t-1}) + \frac{\eta_i(h_t)}{\delta\pi(s_t)}}{1 + \sum_{j=1}^N \frac{\eta_j(h_t)}{\delta\pi(s_t)}}, t > 1 \quad (12)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

i.e., it is not necessary to carry the history of all past values for $\eta_i(.)$ to find $\lambda_i(h_t)$. Instead, only the previous value $\lambda_i(h_{t-1})$ and the current multiplier values $\eta_i(h_t)$, $i = 1, \cdots, t$ are sufficient.

In the following analysis, we show that it is not even necessary to explicitly compute $\eta_i(h_t)$ at s_{t+1} to find $\lambda_i(h_{t+1})$; only the value $\lambda_i(h_t)$ can be used in a simple update rule to determine $\lambda_i(h_{t+1})$. To derive this rule, we first show that at any given state, s_t, $\lambda_i(h_t)$, or the normalized weight of CP$_i$ in the federation is bounded.

Proposition 3. *In the infinitely repeated game $\langle \mathcal{N}, (w_i(s_t))_{i=1}^N, (r_i(w_i(s_t)))_{i=1}^N \rangle$, at any state $s_t = s$, the normalized relative weight of the marginal revenue of any CP, CP$_i$, in the SPNE strategies is always bounded such that $\underline{\lambda}_i^s \le \lambda_i(h_t) \le \bar{\lambda}_i^s$, where the bounds $\underline{\lambda}_i^s$ and $\bar{\lambda}_i^s$ are only state-dependent constants.*

A useful corollary follows immediately from the above proposition.

Corollary 1. *In the infinitely repeated game $\langle \mathcal{N}, (w_i(s_t))_{i=1}^N, (r_i(w_i(s_t)))_{i=1}^N \rangle$, at any state $s_t = s$, the normalized relative weight of the marginal revenues for any CP, CP$_i$, in the SPNE, $\lambda_i(h_t) = \underline{\lambda}_i^s$ if, and only if, the commitment constraint (8) for CP$_i$ binds. Also, $\lambda_i(h_t) = \bar{\lambda}_i^s$ if, and only if, the commitment constraints (8) for all CP$_j$, $j \neq i$ bind.*

The above results furnish the basis for the following update rule.

Proposition 4. *In the infinitely repeated game $\langle \mathcal{N}, (w_i(s_t))_{i=1}^N, (r_i(w_i(s_t)))_{i=1}^N \rangle$, the normalized weight of the marginal revenue of any CP, CP$_i$, $i = 1, \cdots, N$, in the SPNE strategies, follows the following update rule,*

$$\lambda_i(h_t) \begin{cases} = \underline{\lambda}_i^s & \lambda_i(h_{t-1}) \le \underline{\lambda}_i^s \\ = \lambda_i(h_{t-1}) & \underline{\lambda}_j^s < \lambda_j(h_{t-1}) < \bar{\lambda}_j^s \forall j \\ \in [\underline{\lambda}_i^s, \lambda_i(h_{t-1})] & \underline{\lambda}_i^s < \lambda_i(h_{t-1}) < \bar{\lambda}_i^s \\ & \wedge \exists j \text{ s.t. } \lambda_j(h_{t-1}) < \underline{\lambda}_j^s \\ = \bar{\lambda}_i^s & \lambda_i(h_{t-1}) \ge \bar{\lambda}_i^s \end{cases} \quad (13)$$

given an initial federation weight $\lambda_i(h_0) = \lambda_i$.

Thus, given a current state s_t, the variables $\lambda_i(h_t)$, $i = 1, \cdots, N$, can be used in (10) to obtain $N-1$ equations in the unknowns $w_1(h_t), \cdots, w_N(h_t)$ which along with (2), at equality, are sufficient to obtain the CPs spot allocations strategies, $w_1(h_t), \cdots, w_N(h_t)$. The next section presents a computational procedure aimed at numerically approximating $\lambda_i(h_t)$.

7 Numerical Computation

The goal of this section is to find the numerical values for the bounds $\bar{\lambda}_i^s$ and $\underline{\lambda}_i^s$ needed for numerical computations, through formulating **P3** from the perspective of a single CP aiming at maximizing its profit, using the derived recursive relation (12). More precisely, we show that CP$_i$ can cast its strategy update problem in the form of a Bellman equation [30] that is solved iteratively. We first proceed by noting that, from the system in (10) and the capacity constraint (2), $w_i(h_t)$ can be regarded as an implicit function of $\lambda_i(h_{t-1})$ and the state s_t, in the sense that knowledge of the latter ones can be used to calculate the former. This observation can be stated by rewriting the revenue function $r_i(w_i(h_t))$ as $r_i(w_i({\lambda_i}^{t-1}), s_t)$, where ${\lambda_i}^{t-1} \equiv \lambda_i(h_{t-1})$. The basic idea of using the Bellman equation is the ability to transform our optimization problem that has an infinite time dimension into a state-dependent smaller optimization problem where we only focus on finding the solution $w_i(s_t)$ for the current state s_t. Since, this decision also affects the selected strategies in the future states, we also need to find the optimal value for the new variable ${\lambda_i}^t$ to maintain optimal decisions in the next state s_{t+1} and so on. To this end, the following steps follow the standard derivation procedure for Bellman equation for stochastic problems [30].

Given initial state s_1, and initial weights ${\lambda_i}^0 = \lambda_i$, then from CP$_i$'s perspective, **P3** is equivalent to

$$R_i({\lambda_i}^0, s_1) \equiv \max_{w_i} r_i(w_i({\lambda_i}^0), s_1) + \sum_{t=2}^{\infty} \sum_{h_t} \delta^{t-1} \pi_{s_t}^{h_t} r_i(w_i({\lambda_i}^{t-1}), s_t) \quad (14)$$

subject to (8) and (2), where $R_i({\lambda_i}^{t-1}, s_t)$ is CP$_i$'s expected long-term revenue at t starting at s_1, and $\pi_a^b = \pi(b|a)$. It can be shown that, in general, $R_i({\lambda_i}^{t-1}, s_t)$, can be cast recursively,

$$R_i({\lambda_i}^{t-1}, s_t) = \max_{w_i} r_i(w_i({\lambda_i}^{t-1}), s_t) + \sum_{s_{t+1}} \delta \pi_{s_t}^{s_{t+1}} R_i({\lambda_i}^t, s_{t+1})) \quad (15)$$

Similarly, (8) can be recast recursively as,

$$r_i(w_i({\lambda_i}^{t-1}), s_t) + \delta \Sigma_{s_{t+1}} \pi_{s_t}^{s_{t+1}} R_i({\lambda_i}^t, s_{t+1}) \ge \underline{R}_i(s_t) \quad (16)$$

Solving (15) s.t. (16) can be done using iterative methods [30], [32] to obtain the optimal values of $w_i(.)$ and ${\lambda_i}^t$. Appendix C describes details pertaining to the dynamic program that solves this problem.

8 Performance Evaluation

The employed simulation environment models a federation of three CPs, all offering a single type of VMs with the following configurations: 1 CPU core, 1.7 GB RAM, 1 EC2 Compute Unit, and 160 GB of local storage, which is similar to that of Amazon EC2 small instances [5]. Following Toosi et al. [12], each CP is equipped with 128 servers, each supporting 8 VMs. So, each provider's capacity is $C_i = 1024$. All CPs follow a dynamic pricing mechanism for the spot market according to the market demand. Since there is no available information for Amazon EC2

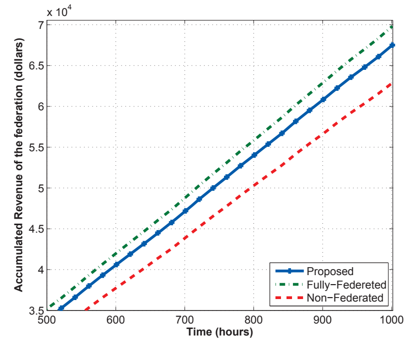
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

pricing mechanism, rather only an expected range of $0.02 - 0.085$ dollars per hour per VM, we followed the approach of Zhang et al. [13] in constructing $r_i(w_i(t)) = w_i(t).P(w_i(t))$ from a common piecewise-linear pricing function $P(w_i(t))$ in the number of VMs. $P(w_i(t))$ was constructed such that the VM price range falls within the interval $0.02 - 0.085$. The experiments simulated 6 weeks of operation. The workload of the guaranteed-service users was constructed such that each CP may experience 3 possible states of workloads termed as *low (l)*, *medium (m)* and *high (h)* with an average spare capacity to host 100, 300 and 800 spot market and/or federation VMs in these states, respectively, for CP_1, 100, 400 and 900 for CP_2 and 150, 400 and 950 for CP_3, respectively. Hence, at each time t, the federation workload state may be described by one of the possible 27 states. The state transition matrix between states was generated randomly from a normal distribution and normalized to satisfy the transition matrix properties. The dynamic programming problem in (15) and (16) was solved using Matlab once to obtain the intervals and the weights of the CPs in Proposition 4. These values were then used to calculate the VMs shares using (10) and (2) at each time step.

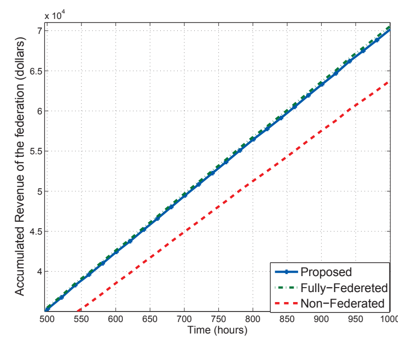
Federation and CPs revenues: Fig. 2 depicts a comparison of the accumulated federation revenue after 6 weeks of operation using the proposed scheme against that obtained by the fully-federated and non-federated models for $\delta = 0.9$ and 0.98. When the CPs are *sufficiently patient*, with a corresponding $\delta = 0.98$, and hence associating a higher weight to future revenue, the accumulated revenue in the proposed scheme coincides with the maximum possible gain of the federation which is obtained by the fully-federated model. As δ decreases, the CPs preference for more instantaneous gains at each state starts to increase, leading to states where the CPs are less interested in sharing their resources and the number of hosted VMs of other providers decreases. This indicates that the federation's total revenue decreases gradually as δ decreases. Table 1 gives a comparison between the achieved mean and variances of the hourly federation revenue for the three mechanisms for $\delta = 0.9$. The values show a significant increase in the mean of the hourly revenue and a reduction of approximately 40% for its variance compared to the non-federated approach.

Fig. 3 also demonstrates the efficiency of the proposed scheme from the perspective of individual CPs. As shown by the graphs, the increase of the 6 week revenue between the proposed and non-federated mechanisms reaches approximately 4000 and 2000 dollars for CP_1 and CP_3, respectively, for $\delta = 0.9$.

Spot market pricing and workloads: Two major performance issues for CPs' spot markets are the unpredictability of VM availability and the high vari-



(a) $\delta = 0.9$



(b) $\delta = 0.98$

Fig. 2. Federation accumulated revenue

TABLE 1
The federation hourly revenue

	Fully-federated	Non-federated	Proposed
mean	70.4	61.6	**68.2**
variance	13.84	21.324	**14.813**

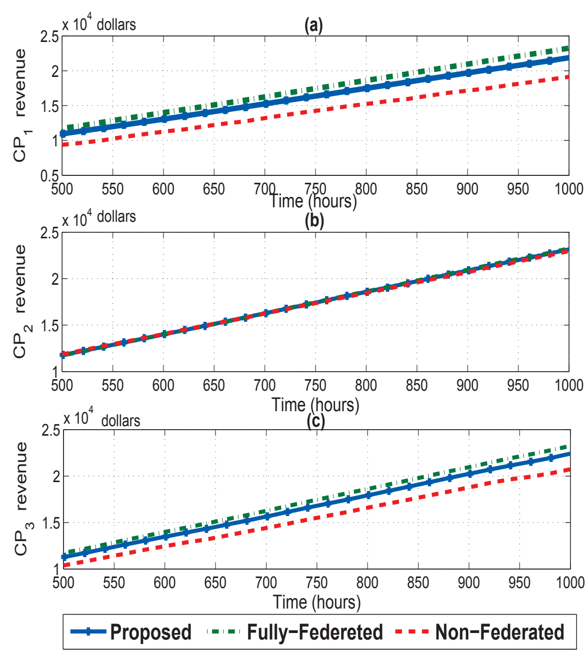


Fig. 3. Individual CPs' accumulated revenue

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

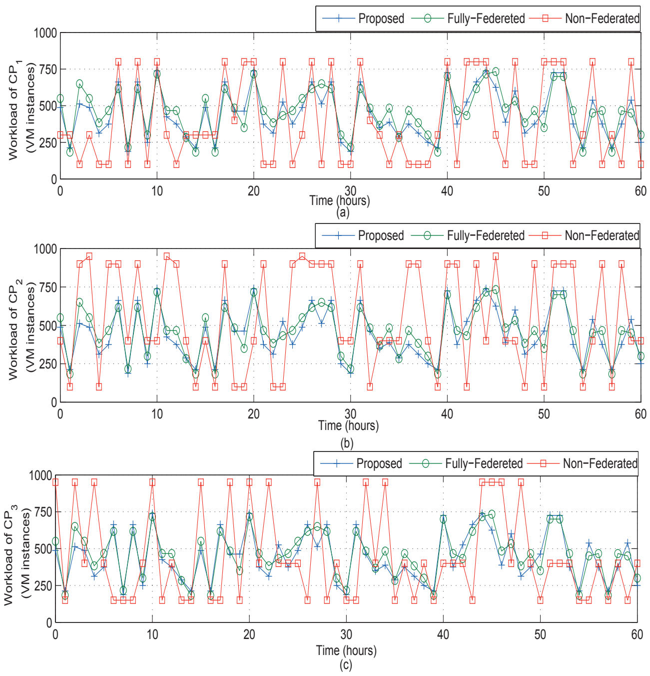


Fig. 4. The hourly spot markets workloads

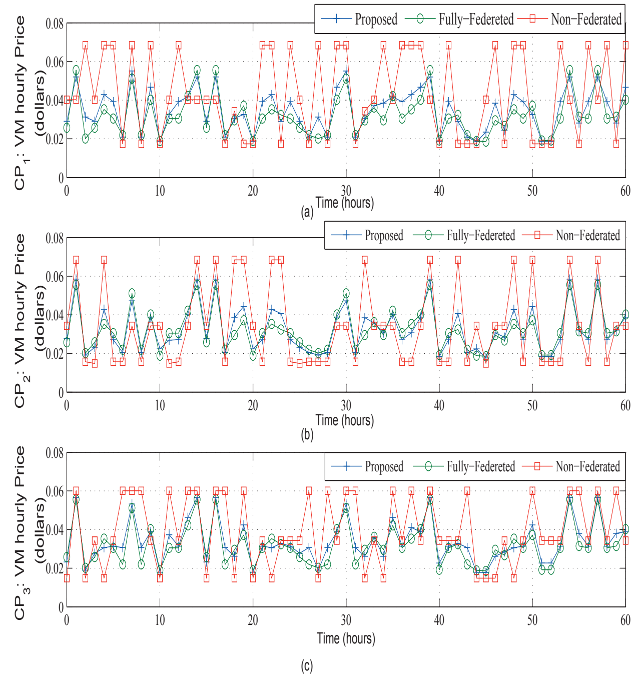


Fig. 5. Hourly spot markets VM prices (dollars)

ance in the VM hourly prices. These two problems are easily detected from the performance of the non-federated mechanism in Figs. 4 and 5 which provide a 60 hours snapshot of the operation of the CPs' spot markets. The effects of price fluctuation in the non-federated strategy are also reflected by similar fluctuations in the hourly revenue as shown in Fig. 6. For example, consider the spot market of CP_3, where we can easily see that a sudden decrease of workloads due to a decrease in the VM availability from 950 VMs at $t = 18$ to 150 spot VMs at $t = 19$ (Fig. 4(c)) results in a VM price spike from less than $\$0.02$ to almost $\$0.06$ dollars (Fig.5(c)). In contrast, the proposed mechanism significantly smoothes the fluctuations in the workloads due to the large variance in the VM availability. Taking for example, the same time points, we can see that at hours t=18 and 19, the number of VMs in the spot market for CP_3 is around 350 and 450 with corresponding prices set at 0.035 and 0.04 dollars, respectively. Here, the CPs almost equally divide their VM shares at each step. As $\delta = 0.9$ is large enough, both CPs have a value, high enough, for future revenue to self-enforce the best outcome of the game which is that imposed originally by the fully-federated model. This smoothing effect is clearly demonstrated through the achieved low variance in the CPs resource availability in their spot markets, and in turn CP workloads, pricing and hourly revenue, as indicated by the three sections in Table 2, respectively. The obtained values demonstrate how the proposed scheme effectively reduced the variance in the prices and workloads by more than 50%.

It is important to notice that although the spike in the pricing in the case of the non-federated scheme may achieve a higher instantaneous revenue, as shown in Fig. 6, the long-term achieved revenue for the proposed scheme is always guaranteed to be at least equal to that obtained by the non-federated mechanism due to the commitment constraints (8). Furthermore, as indicated by Fig. 3, a much higher revenue can always be achieved if the CPs maintain a higher value (δ) for future revenues.

TABLE 2
Spot markets behavior

		CP_i	Fully-federated	Non-federated	Proposed
Workload (VM instances/hour)	mean	CP_1	462	413	**455**
		CP_2	462	460	**459**
		CP_3	462	460	**462**
	var.	CP_1	146	300	**155**
		CP_2	146	334	**165**
		CP_3	146	312	**168**
VM price (¢/hour)	mean	CP_1	3.187	3.94	**3.4**
		CP_2	3.187	3.57	**3.28**
		CP_3	3.187	3.87	**3.35**
	var.	CP_1	0.97	2.3	**1.07**
		CP_2	0.97	2.09	**1.07**
		CP_3	0.97	1.7	**1.01**
revenue ($/hour)	mean	CP_1	23.47	19.3	**22.2**
		CP_2	23.47	22.87	**23.5**
		CP_3	23.47	21.06	**22.67**
	var.	CP_1	4.7	9.95	**5.3**
		CP_2	4.7	10.06	**5.7**
		CP_3	4.7	9.027	**5.29**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

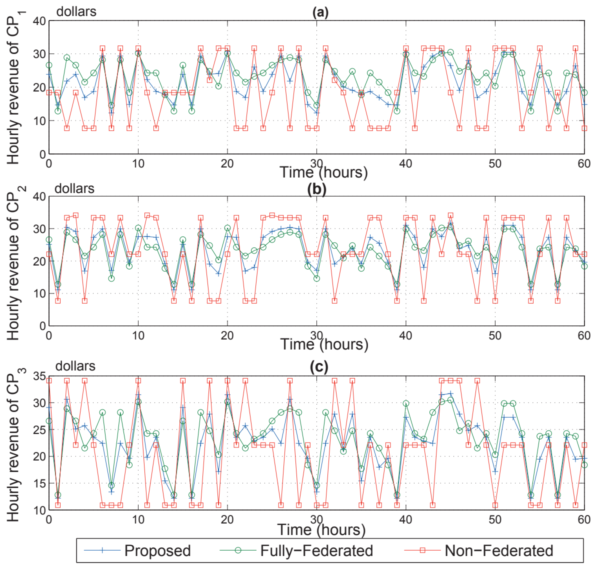


Fig. 6. The CPs hourly revenue (dollars)

9 Conclusion and Future Work

This article presented a novel model for capacity sharing in a federation of IaaS CPs, via modeling the interactions among CPs as a repeated game of VM outsourcing with the option of offering all unused capacity in the spot market. Performance evaluation results quantified the profit gained by the federation as well as by individual CPs and demonstrated significant smoothing effects on the spot market prices. In future, we plan to extend the proposed model to achieve full decentralization.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Comp. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[3] Í. Goiri, F. Julià, J. Oriol Fitó, M. Macías, and J. Guitart, "Supporting CPU-based guarantees in cloud SLAs via resource-level QoS metrics," *Future Gener. Comput. Syst.*, vol. 28, no. 8, pp. 1295–1302, Oct. 2012.
[4] M. Mattess, C. Vecchiola, and R. Buyya, "Managing peak loads by leasing cloud infrastructure services from a spot market," in *IEEE HPCC, 2010*, pp. 180–188.
[5] "Amazon EC2," http://aws.amazon.com/ec2/, Dec 2011.
[6] J. Chen, C. Wang, B. Zhou, L. Sun, Y. Lee, and A. Y. Zomaya, "Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud," in *HPDC 2011*. ACM, pp. 229–238.
[7] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muñoz, and G. Toffetti, "Reservoir - when one cloud is not enough," *IEEE Computer*, vol. 44, no. 3, pp. 44–51, 2011.
[8] B. Rochwerger and D. e. a. Breitgand, " The Reservoir model and architecture for open federated cloud computing," *IBM J. Res. Dev.*, vol. 53, no. 4, pp. 1–11, July 2009.
[9] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. Nguyen, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in *SC 2011*, pp. 1–12.
[10] Í. Goiri, J. Guitart, and J. Torres, "Economic model of a cloud provider operating in a federated cloud," *Information Systems Frontiers*, vol. 12, pp. 827–843, 2012.
[11] A. Avetisyan, R. Campbell, I. Gupta, M. Heath, S. Ko, G. Ganger, M. Kozuch, D. O'Hallaron, M. Kunze, T. Kwan, *et al.*, "Open cirrus: A global cloud computing testbed," *Computer*, vol. 43, no. 4, pp. 35–43, 2010.
[12] A. Toosi, R. Calheiros, R. Thulasiram, and R. Buyya, "Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud Environment," in *IEEE HPCC 2011*, pp. 279–287.
[13] Q. Zhang, E. Gürses, R. Boutaba, and J. Xiao, "Dynamic resource allocation for spot markets in clouds," in *USENIX Hot-ICE'11*.
[14] E. Gomes, Q. Vo, and R. Kowalczyk, "Pure exchange markets for resource sharing in federated clouds," *Concurrency Computat.: Pract. Exper*, vol. 24, no. 9, pp. 977–991, 2012.
[15] M. Mihailescu and Y. Teo, "Dynamic resource pricing on federated clouds," in *IEEE CCGrid 2010*, pp. 513–517.
[16] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profit-driven service request scheduling in clouds," in *IEEE CCGrid 2010*, pp. 15–24.
[17] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads," in *IEEE CLOUD 2010*, pp. 228–235.
[18] S. Chaisiri, B. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Service Comput.*, vol. 5, no. 2, pp. 164 – 177, 2012.
[19] R. Wolski, J. Plank, J. Brevik, and T. Bryan, "Analyzing market-based resource allocation strategies for the computational grid," *INT J HIGH PERFORM C*, vol. 15, no. 3, pp. 258–281, 2001.
[20] X. Bai, D. Marinescu, L. Bölöni, H. Jay Siegel, R. Daley, and I. Wang, "A macroeconomic model for resource allocation in large-scale distributed systems," *Jrnl. Parallel Distr. Comput.*, vol. 68, no. 2, pp. 182–199, 2008.
[21] W. Streitberger and T. Eymann, "A simulation of an economic, self-organising resource allocation approach for application layer networks," *Comput. Netw.*, vol. 53, no. 10, pp. 1760 – 1770, 2009.
[22] M. Mazzucco and M. Dumas, "Reserved or on-demand instances? a revenue maximization model for cloud providers," in *IEEE CLOUD 2011*, pp. 428–435.
[23] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Comput.*, vol. 13, no. 5, pp. 14–22, 2009.
[24] S. Pacheco-Sanchez, G. Casale, B. W. Scotney, S. I. McClean, G. P. Parr, and S. Dawson, "Markovian workload characterization for qos prediction in the cloud," in *IEEE CLOUD 2011*, pp. 147–154.
[25] Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," in *IEEE CNSM 2010*, pp. 9–16.
[26] S. Maguluri, R. Srikant and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *IEEE INFOCOM*, 2012.
[27] W. Wong and J. Davis, "Boostpred: An automatic demand predictor for the cloud," in *DASC 2011*, pp. 411 –418.
[28] H. Khazaei, J. Misic, and V. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queueing Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 5, pp. 936 – 943, 2012.
[29] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.
[30] L. Ljungqvist and T. Sargent, *Recursive macroeconomic theory, Second Edition*. MIT press, 2004.
[31] E. Ligon, J. Thomas, and T. Worrall, "Mutual insurance, individual savings, and limited commitment," *Review of Economic Dynamics*, vol. 3, no. 2, pp. 216–246, 2000.
[32] D. Bertsekas, *Dynamic programming and stochastic control*. Academic Pr, 1976, vol. 125.

Nancy Samaan Received her Ph.D. degree in computer science from the University of Ottawa, Canada, in 2007 and her BSc and MSc

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

degrees from the department of computer science, Alexandria University, Egypt. She is currently an assistant professor with the School of Electrical Engineering and Computer Science at the University of Ottawa, Canada In 2008, she was awarded the Natural Sciences and Engineering Research Council of Canada University Faculty Award. Her current research interests include network resource management, wireless communications, Quality of Service (QoS) issues and autonomic communications.