Performance and Revenue Analysis of Hybrid Cloud Federations with QoS Requirements

Bowen Song
Department of Computer Science
University of Southern California
941 Bloom Walk, Los Angeles, USA
bowenson@usc.edu

Marco Paolieri
Department of Computer Science
University of Southern California
941 Bloom Walk, Los Angeles, USA
paolieri@usc.edu

Leana Golubchik
Department of Computer Science
University of Southern California
941 Bloom Walk, Los Angeles, USA
leana@usc.edu

***Abstract*—Hybrid cloud architectures, where private clouds or data centers forward part of their workload to public cloud providers to satisfy quality of service (QoS) requirements, are increasingly common due to the availability of on-demand cloud resources that can be provisioned automatically through programming APIs. In this paper, we analyze performance and revenue in federations of hybrid clouds, where private clouds agree to share part of their local computing resources with other members of the federation. Through resource sharing, underprovisioned members can save on public cloud costs, while overprovisioned members can put their idle resources to work. To reward all hybrid clouds for their contributions (computing resources or workload), public cloud savings due to the federation are distributed among members according to Shapley value.**

We model this cloud architecture with a continuous-time Markov chain and prove that, if all hybrid clouds have the same QoS requirements, their profits are maximized when they join the federation and share all resources. We also show that this result does not hold when hybrid clouds have different QoS requirements, and we provide a solution to evaluate profit for different resource sharing decisions. Finally, our experimental evaluation compares the distribution of public cloud savings according to Shapley value with alternative approaches, illustrating its ability to discourage free riders of the federation.

***Index Terms*—Hybrid Clouds, Data Centers, Cloud Federations, Markov Chains, Performance, Shapley Value.**

I. Introduction

Over the past 15 years, cloud computing has radically transformed the IT industry by removing the need for upfront commitments to acquire hardware resources and expertise to operate them. In the *Infrastructure-as-a-Service* (IaaS) market, *public cloud providers* (Amazon AWS, Google Cloud, Microsoft Azure) offer a multitude of hardware resources (CPUs, GPUs, FPGAs, machine learning accelerators) remotely accessible through virtual machines (VMs). Similarly to other utilities, application developers can pay for these resources by usage time (e.g., CPU cores paid by the second) and quickly allocate them on demand to meet changes in their workloads, alleviating the risk of underprovisioning or overprovisioning to satisfy quality of service (QoS) requirements [5].

Such flexibility has had a profound impact on the design and operation of *private clouds* and data centers of large organizations: instead of acquiring enough hardware resources to satisfy QoS requirements during predicted peaks of their workloads, private clouds can *forward part of the workload to public clouds when needed.* This cloud architecture, called *hybrid cloud* [23], [25], is also common for organizations that decide to avoid investing in on-premise data centers altogether and choose instead to commit to a certain amount of cloud resources for fixed periods of time, obtaining discounts with respect to on-demand prices; for example, Amazon AWS *reserved instances* offer up to 72% discounts on VM instances in case of 1-year or 3-year commitments. Given the difficulty of workload prediction, organizations can reserve an underprovisioned pool of resources at a discounted price, allocating more resources on-demand during peak loads. Software platforms for private clouds, such as OpenStack [3], Apache CloudStack [1], and OpenNebula [2], facilitate these hybrid architectures by implementing APIs compatible with those of public clouds.

In addition to hybrid architectures, *cloud federations* [20], [17] provide another appealing strategy to operate under uncertain and variable workloads: private clouds agree to *share part of their computing resources* to serve requests from other members of the federation. In so doing, underprovisioned private clouds can satisfy their QoS requirements during peak loads, while overprovisioned private clouds can put their resources to work instead of leaving them idle. Different policies have been investigated to reward members of the federation that provide resources or workload [16], [9]: at one extreme, business operations are shared entirely (e.g., members of the federation become one business organization) and resource sharing is optimized to maximize the total profit, which is then distributed according to solution concepts for cooperative games (e.g., Shapley value); at the other extreme, members are non-cooperative players maximizing their profit by sharing part of their resources or workload, while the federation is self-enforced by a pricing mechanism used for its services. Resources can be shared with the entire federation [16] or with individual members [9], and resource owners may have priority over the use of shared resources [16].

In this paper, we focus on IaaS cloud architectures including both hybrid clouds and cloud federations. As illustrated in Fig. 1, when local queues are too large to satisfy QoS requirements (in our setting, the maximum mean waiting times allowed by service level agreements), private clouds

Work supported in part by NSF Grants CCF-1763747 and CNS-1816887.

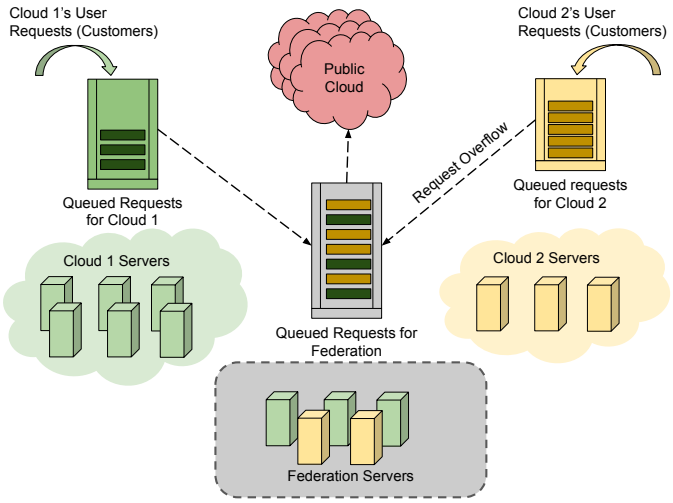


Figure 1: Federation of Hybrid Clouds

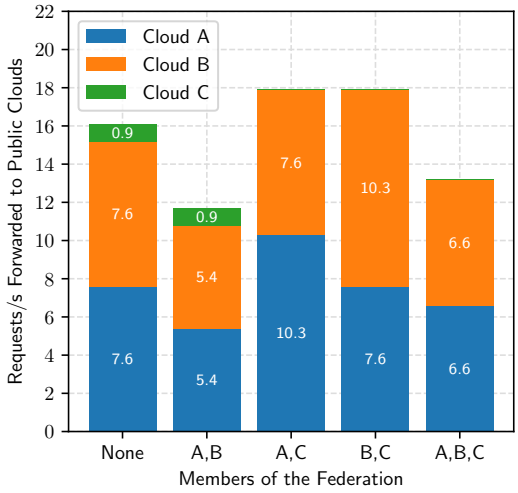


Figure 2: Forward rates to public clouds for different federations

forward user requests for VM instances (i.e., their *overflow traffic*) to a shared pool of resources contributed by members of the federation; if the queue at the shared federation pool is also too large, requests are forwarded to public clouds. Through the federation, underprovisioned hybrid clouds save on public cloud costs, while overprovisioned hybrid clouds can put their resources to work. To reward all members for their contributions (computing resources to serve overflow traffic, or workload to put idle resources to use), in our proposed mechanism *hybrid clouds pay the same public cloud costs as before joining the federation*, and then *public cloud savings due to the shared server pool are distributed among members* according to Shapley value [21].

A challenging problem for hybrid clouds is to decide whether to join a federation, and to determine the amount of resources to share in order to maximize profit; this problem is particularly difficult when each hybrid cloud has different QoS requirements for its users. Fig. 2 illustrates forwarding rates to public clouds for different federations among hybrid clouds A, B, C where: (1) each hybrid cloud has 100 servers processing requests with service times exponentially distributed with rate 1 and interarrival times exponentially distributed with rate 100; (2) all resources of a member are shared with the federation; (3) requests of C can tolerate mean waiting time equal to 1, while requests of A and B must begin service immediately, without queueing. When A and B form a federation, their cumulative forwarding rate to public clouds is reduced from 15.2 to 10.8, and Shapley value splits these savings equally; in contrast, when A and C (or, similarly, B and C) form a federation, *their cumulative forwarding rate to public clouds increases* from 8.5 to 10.3 (the forwarding rate of C becomes 0, while the forwarding rate of A becomes 10.3). This negative effect is due to the different QoS requirements of A and C: requests from C can accumulate at the shared federation pool, forcing A to use the public cloud during peak loads. A similar, but less evident effect is also present when all hybrid clouds A, B, C join the federation: in this case, the cumulative forwarding rate to public clouds (13.2) is higher than that of a federation including only A and B, and leaving C on its own (11.7).

Contributions. Our work provides a solution to evaluate the effects of different sharing strategies in federations of hybrid clouds, which we model as a network of queues [13], each with a finite number of servers and capacity determined by QoS requirements (maximum mean waiting time, as in [8], [16]). By analyzing the underlying continuous-time Markov chain (CTMC), we leverage existing results for resource sharing in queueing networks [22] to prove that, if all hybrid clouds have the same QoS requirements, *profit is maximized when each hybrid cloud shares all of its resources* (without priority over their use). Next, we show that this result does not hold when hybrid clouds have different QoS requirements, and we provide a solution to evaluate profit for different resource sharing decisions. Finally, we compare our profit sharing mechanism (assigning public cloud savings according to Shapley value) with alternative approaches, illustrating its ability to discourage "free riders" of the federation.

II. Performance Model

In this section, we present a performance model to evaluate the stationary rate of requests that cannot be served by a federation of private clouds without violating their individual QoS requirements, thus requiring to be forwarded to a public cloud provider. We prove that, if each private cloud has the same QoS requirements, this rate of "rejected requests" is minimized when all resources are shared with the federation; in contrast, for private clouds with heterogeneous QoS requirements, we provide counterexamples showing that sharing all resources may increase rejected traffic for some clouds, and in some cases for the entire federation.

A. System Description and Notation

We consider the federation of N private clouds illustrated in Fig. 3, where each private cloud $i = 1, \ldots, N$ receives requests according to a Poisson process with rate λ_i and owns n_i servers:

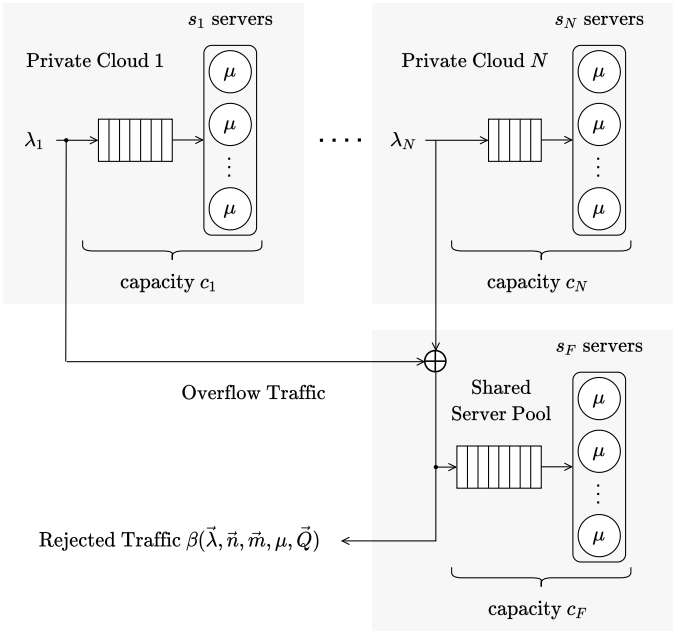


Figure 3: Queueing Model of a Hybrid Cloud Federation

m_i servers are contributed to the shared federation pool (which has a total of $s_F = \sum_i^N m_i$ servers), while $s_i = n_i - m_i$ servers are used exclusively by i.

When all s_i local servers are busy, requests received by i are queued locally (and processed according to a FCFS policy) as long as the queue size q_i is such that $q_i/(s_i\mu) \leqslant Q_i$, i.e., the expected waiting time of the last request in the queue (the sum of q_i i.i.d. exponential random variables with rate $s_i\mu$) is lower than Q_i, the QoS requirement for users of private cloud i. Similarly, a request rejected locally at private cloud i can be served by the pool of s_F servers contributed to the federation as long as its queue size q_F is such that $q_F/(s_F\mu) \leqslant Q_i$.

Note that this type of QoS requirements corresponds to a maximum *capacity* $c_i := C(s_i, \mu, Q_i)$ for each private cloud i, where

$$C(s_i, \mu, Q_i) = \lfloor s_i\mu Q_i \rfloor + s_i = \lfloor s_i(\mu Q_i + 1) \rfloor$$

is the maximum number of requests queued ($\lfloor s_i\mu Q_i \rfloor$) or in service ($s_i$) at private cloud i. For example, no queueing is allowed when $Q_i = 0$, since c_i is equal to the number of servers s_i. The maximum capacity of the shared federation pool is equal to $c_F := \max_{i=1,\ldots,N} C(s_F, \mu, Q_i)$, i.e., to the maximum capacity allowed by the QoS requirements of any private cloud. A large class of QoS requirements (such as bounds on percentiles of waiting time or service time of a request) can be defined similarly from the distribution of waiting times (Erlang with shape q_i and rate $s_i\mu$) and service time (exponential with rate μ); these QoS requirements also result in restrictions on the capacity of each private cloud, and they are supported by our model.

Our goal is to evaluate the stationary (i.e., steady-state) rate $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q})$ of requests cumulatively rejected by all the private clouds of the federation for the given arrival rates λ_i, shared servers $0 \leqslant m_i \leqslant n_i$, service rate μ, and maximum mean waiting times Q_i, for $i = 1, \ldots, N$. Since the overflow traffic in Fig. 3 is not a Poisson process, we need a joint CTMC model of the private clouds and shared server pool to evaluate $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q})$.

B. CTMC Model

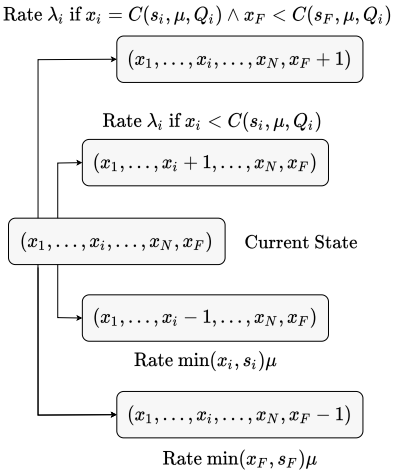


Figure 4: Transition rates for the CTMC model

Given our assumptions, the evolution of a hybrid cloud federation is described by a CTMC with state vector $(x_1, \ldots, x_N, x_F)$ where $0 \leqslant x_i \leqslant C(s_i, \mu, Q_i)$ represents the number of requests (queued or in service) at each private cloud $i = 1, \ldots, N$, while $0 \leqslant x_F \leqslant \max_{i=1,\ldots,N} C(s_F, \mu, Q_i)$ is the number of requests (queued or in service) at the shared server pool.

From state $(x_1, \ldots, x_i, \ldots, x_N, x_F)$, the CTMC can transition to the following states, for all $i = 1, \ldots, N$ (Fig. 4):

- If $x_i < C(s_i, \mu, Q_i)$, to $(x_1, \ldots, x_i+1, \ldots, x_N, x_F)$ with rate λ_i (an arrival to private cloud i can be served locally while satisfying QoS requirements).
- If $x_i = C(s_i, \mu, Q_i)$ and $x_F < C(s_F, \mu, Q_i)$, to state $(x_1, \ldots, x_i, \ldots, x_N, x_F + 1)$ with rate λ_i (an arrival to private cloud i cannot be served locally, but it can be served by the shared federation pool while satisfying QoS requirements of private cloud i).
- To $(x_1, \ldots, x_i - 1, \ldots, x_N, x_F)$ with rate $\min(x_i, s_i)\mu$ (service of a request completes at private cloud i).
- To $(x_1, \ldots, x_i, \ldots, x_N, x_F - 1)$ with rate $\min(x_F, s_F)\mu$ (service of a request completes at the pool of servers shared by members of the federation).

The rate $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q})$ of requests rejected by the federation of hybrid clouds can be evaluated from the steady-state probabilities $p(\vec{x})$ for each $\vec{x}$ in the state space $\mathcal{X}$ of the CMTC model: $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q})$ is equal to the sum of the arrival rate of each private cloud, multiplied by the probability of being

in a state where arrivals are rejected by the private cloud and by the federation (and thus forwarded to the public cloud):

$$\beta(\vec{\lambda},\vec{n},\vec{m},\mu,\vec{Q}) = \sum_{i=1}^{N} \lambda_i \left(\sum_{\substack{\vec{x}\in\mathcal{X}: x_i = C(s_i,\mu,Q_i) \\ \wedge x_F \geqslant C(s_F,\mu,Q_i)}} p(\vec{x}) \right). \quad (1)$$

C. Optimal Sharing Strategy with Homogeneous QoS

Each private cloud i can choose to share $0 \leqslant m_i \leqslant n_i$ servers with the federation pool, which receives overflow traffic from all the private clouds, each with different arrival rate, number of servers, and QoS requirements. We are interested in the strategy $\vec{m} = (m_1, \ldots, m_N)$ that is the most efficient for the entire federation, i.e., the strategy that minimizes the stationary rate of rejected requests in Eq. (1). We prove that sharing all servers is the most efficient strategy when private clouds have the same QoS requirements $Q_1, \ldots, Q_N$.

Theorem 1. *Let $i = 1, \ldots, N$ be a set of $M/M/s_i/c_i$ queues, each with Poisson arrival rate λ_i, service rate μ, s_i servers and total capacity (requests queued or in service) $c_i := C(s_i)$, where $C(x+y) \geqslant C(x) + C(y)\,\forall x, y$. When these queues send their overflow traffic to a shared $G/M/s_F/c_F$ queue (without external arrivals) with $s_F := \sum_{i=1}^{N}(n_i - s_i)$ servers and capacity $c_F := C(s_F)$, the stationary rate of rejected requests is minimized with respect to s_i if $s_i = 0$ for all $i = 1, \ldots, N$ (i.e., when $s_F = \sum_{i=1}^{N} n_i$).*

Proof. First, we leverage Theorem 7 of [22], which states that the cumulative rejection rate of a set of $M/M/s_i/c_i$ queues for $i = 1, \ldots, N$ is greater or equal to the rejection rate of a single $M/M/s/c$ queue with combined arrival rate $\lambda = \sum_{i=1}^{N} \lambda_i$, servers $s = \sum_{i=1}^{N} s_i$ and capacity $c = \sum_{i=1}^{N} c_i$. This means that the system in Fig. 5 has lower overflow traffic than the one in Fig. 3.

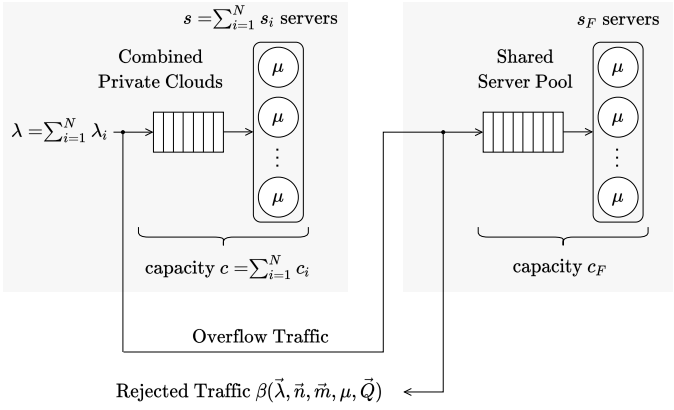


Figure 5: Queueing model after combining the private clouds

Next, we show that the rejection rate of an $M/M/s/c$ queue forwarding its overflow traffic to a $G/M/s_F/c_F$ queue without other external arrivals (Fig. 5) is greater or equal to the rejection rate of a single $M/M/(s+s_F)/(c+c_F)$ queue with the same arrival rate. Our argument hinges on the comparison of transition rates between states of the two systems. The underlying stochastic process of the combined $M/M/(s+s_F)/(c+c_F)$ queue (Fig. 6) is a birth-death CTMC where state $j+1$ is reached from $j = 0, \ldots, c+c_F-1$ with rate λ, while state $j-1$ is reached from $j = 1, \ldots, c+c_F$ with rate $\min(j, s+s_F)\mu$. In the system with separate queues (Fig. 7), state $j+1$ is still reached from $j = 0, \ldots, c+c_F-1$ with the same rate λ; in contrast, state $j-1$ is reached from $j = 1, \ldots, c+c_F$ with rate $[\min(j_1, s) + \min(j_2, s_F)]\mu$, which depends not only on the total number of requests j, but also on the number of requests j_1 and j_2 in each subsystem (and thus on the past history of transitions that led to $j = j_1 + j_2$).

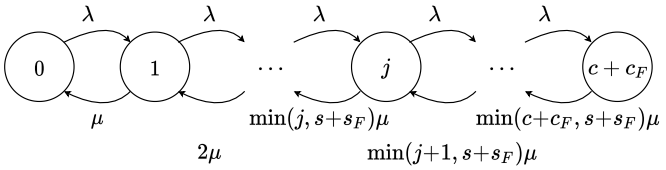


Figure 6: Transition rates of an $M/M/(s+s_F)/(c+c_F)$ queue

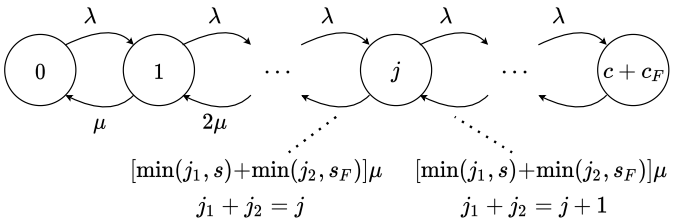


Figure 7: Transition rates after combining the private clouds

Since $\min(j_1, s) + \min(j_2, s_F) \leqslant \min(j, s+s_F)$ for all j_1, j_2 such that $j_1 + j_2 = j$, the transition rate from j to $j-1$ is always lower or equal for the system with separate queues, while the transition rate from j to $j+1$ is the same. As a consequence, the steady-state probability of state $j = s + s_F$ (the only state where requests are rejected, with rate λ) is greater or equal in the system with separate queues.

Thus, the combined system with $s + s_F$ servers (i.e., where all servers are shared by the private clouds) has a lower rate of rejected requests than the one with separate queues with s and s_F servers, which, in turn, has a lower rate of rejected requests than the original system $\forall s_1, \ldots, s_N$ such that $\sum_{i=1}^{N} s_i = s$. Since $C(x+y) \geqslant C(x) + C(y)\,\forall x, y$, the capacity of the combined system is in fact greater or equal to the sum of individual capacities, i.e., $C\big(\sum_{i=1}^{N} s_i\big) \geqslant \sum_{i=1}^{N} C(s_i)$; this increases the number of queueing slots, further reducing the rejection rate when all servers are shared. □

Corollary 1.1. *If $Q_i = Q$ for all i, the rate $\beta(\vec{\lambda},\vec{n},\vec{m},\mu,\vec{Q})$ of requests rejected by the private clouds and federation server pool in Eq. (1) is minimized when $m_i = n_i$ for all $i = 1, \ldots, N$.*

Proof. Each private cloud is an $M/M/s_i/c_i$ queue with $s_i = n_i - m_i$ and $c_i = C(s_i, \mu, Q)$ sending its overflow traffic to the server pool of the federation, a $G/M/s_F/c_F$ queue

with $s_F = \sum_{i=1}^{N} m_i = \sum_{i=1}^{N}(n_i - s_i)$ servers and capacity $c_F = C(s_F, \mu, Q)$. Since $C(s, \mu, Q) := \lfloor s(1 + \mu Q) \rfloor$ satisfies

$$C(x + y, \mu, Q) \geqslant C(x, \mu, Q) + C(y, \mu, Q)$$

for all x, y, μ, Q, by Theorem 1 the rate $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q})$ of rejected requests is minimized when all servers are shared. □

Note that, when all servers are shared ($m_i = n_i$ for all $i = 1, \ldots, N$), the system reduces to an $M/M/s_F/c_F$ queue with $s_F = \sum_{i=1}^{N} n_i$ servers and arrival rate $\lambda = \sum_{i=1}^{N} \lambda_i$. In this case, the stationary rate of rejected requests can be evaluated as $\beta(\vec{\lambda}, \vec{n}, \vec{m}, \mu, \vec{Q}) = \lambda B(s_F, c_F, \lambda/\mu)$, where $B(s_F, c_F, \lambda/\mu)$ is the blocking probability of an $M/M/s_F/c_F$ queue [13]:

$$B(s_F, c_F, \lambda/\mu) = \frac{(\lambda/\mu)^{c_F}/(s_F!\, {s_F}^{c_F - s_F})}{\sum_{j=0}^{s_F}(\lambda/\mu)^j/j! + \sum_{j=s_F+1}^{c_F}(\lambda/\mu)^j/(s_F!\, {s_F}^{j-s_F})}. \quad (2)$$

D. Optimal Sharing Strategy with Heterogeneous QoS

When the private clouds $i = 1, \ldots, N$ have different QoS requirements Q_i (i.e., maximum mean waiting times), the CTMC model presented in Section II-B uses different capacity bounds $C(s_i, \mu, Q_i)$ at their local queues and different criteria to reject incoming requests at the shared server pool: requests from private cloud i are rejected at the shared server pool if $x_F \geqslant C(s_F, \mu, Q_i)$; i.e., depending on the number of requests x_F at the shared server pool, private clouds with less stringent QoS requirements Q_i can add their requests to its queue, while other private clouds with more stringent QoS may be forced to forward their requests to the public cloud. In this case, when private clouds share all of their servers, the federation does not reduce to an $M/M/s_F/c_F$ queue, and it can incur a higher rejection rate (i.e., forwarding rate to the public cloud) with respect to other sharing strategies.

As an example, we consider a federation where $\vec{\lambda} = (50, 50)$, $\vec{n} = (50, 50)$, $\mu = 1$, and $\vec{Q} = (0, 1)$; due to the difference in QoS requirements, for a given sharing strategy $\vec{m} = (m_1, m_2)$, only cloud 2 can queue up to $s_F \mu Q_2 = m_1 + m_2$ requests at the shared server pool of $s_F = m_1 + m_2$ servers. Using our CTMC model and Eq. (1), we evaluate the rate of requests rejected by the federation and forwarded to public clouds for different sharing strategies, as depicted in Fig. 8. Here, we observe that, for any sharing strategy $m_2 > 0$ of cloud 2, as we increase the number of servers m_1 shared by cloud 1, the rate of requests forwarded to public clouds first decreases (as cloud 1 gains access to shared servers of cloud 2) and then noticeably increases (as cloud 2 is able to queue its requests at the shared server pool and to force cloud 1 to forward requests to public clouds). In contrast with our theoretical result for hybrid clouds with homogeneous QoS (where sharing all servers, i.e., $(m_1, m_2) = (50, 50)$ in this example, is always optimal), the optimal sharing strategy is $(m_1, m_2) = (10, 2)$, a partial sharing of resources. While a theoretical result is not available for hybrid clouds with heterogeneous QoS, their policies can be evaluated using our CTMC model.

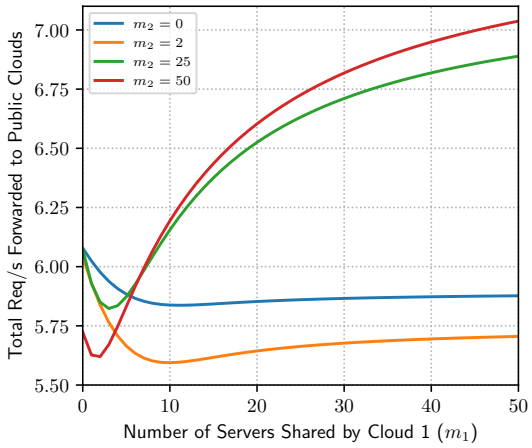


Figure 8: Sharing Strategies with Heterogeneous QoS

III. Revenue Model

Our proposed federation mechanism does not require setting a price for resources shared by members: *servers of the shared pool are available to the federation members at no cost*. To avoid situations where some members of the federation take advantage of shared servers to the detriment of their owners, we establish that, as part of the federation agreement, *each member is charged the same public cloud costs as previously paid, i.e., before joining the federation.* When the shared server pool results in savings of public cloud costs (as demonstrated in Section II-C, this is always the case with homogeneous QoS requirements when sharing all resources), *we distribute savings among federation members according to Shapley value.*

In so doing, we favor resource sharing without more complex policies like granting priority over shared resources to the members who own and operate them [16]. At the same time, through the use of Shapley value, we are able to reward overprovisioned members who share idle resources with the federation, as well as underprovisioned members who share part of their workload; as discussed in Section IV, both types of members are necessary to produce public cloud savings, and thus both should be rewarded in the federation.

A. Sharing Public Cloud Savings using Shapley Value

Given a subset $S = \{i_1, \ldots, i_K\}$ of K private clouds, i.e., $S \subseteq \{1, \ldots, N\}$, we indicate by

$$\vec{\lambda}_S := (\lambda_{i_1}, \ldots, \lambda_{i_K})$$
$$\vec{n}_S := (n_{i_1}, \ldots, n_{i_K})$$
$$\vec{m}_S := (m_{i_1}, \ldots, m_{i_K})$$
$$\vec{Q}_S := (Q_{i_1}, \ldots, Q_{i_K})$$

the vectors of arrival rates $\vec{\lambda}_S$, number of servers $\vec{n}_S$, shared servers $\vec{m}_S$, and maximum mean waiting times $\vec{Q}_S$, respectively, for the private clouds in S, and by $\beta(\vec{\lambda}_S, \vec{n}_S, \vec{m}_S, \vec{Q}_S)$ the rate of requests sent to the public cloud when they form a federation and share servers $\vec{m}_S$ (β can be evaluated using the CTMC model presented in Section II-B).

Then, the savings of public cloud costs obtained by the federation S are given by

$$R(S) = P\Big(\eta_0(S) - \eta_F(S)\Big) \tag{3}$$

where

$$\eta_0(S) := \sum_{i\in S} \beta(\vec{\lambda}_{\{i\}}, \vec{n}_{\{i\}}, \vec{0}_{\{i\}}, \vec{Q}_{\{i\}}) \tag{4}$$

$$\eta_F(S) := \beta(\vec{\lambda}_S, \vec{n}_S, \vec{m}_S, \vec{Q}_S) \tag{5}$$

i.e., by the difference between the rate of requests forwarded before joining the federation (by individual private clouds) and after (when the queue at the federation pool is too large), multiplied by P, the public cloud cost per request. Note that, by Corollary 1.1, Eq. (5) is minimized under homogeneous QoS when all servers are shared with the federation, maximizing $R(S)$ and thus the profit of the federation.

We distribute the public cloud savings $R(S)$ of a federation using *Shapley value* (SV) [21]; each member $i \in S$ receives $R_i^{SV}(S)$, which is the increase in savings due to i, averaged over all possible subsets of other members (i.e., all alternative federations including i):

$$R_i^{SV}(S) = \sum_{S'\subseteq S\setminus\{i\}} \frac{|S'|!(|S| - |S'| - 1)!}{|S|!}\big(R(S'\cup\{i\}) - R(S')\big)\,.$$

Shapley value has many desirable properties for our application: the sum of Shapley values is equal to the total savings, i.e., $\sum_{i\in S} R_i^{SV}(S) = R(S)$; if $i \in S$ and $j \in S$ produce the same increase in savings, i.e., $R(S' \cup \{i\}) = R(S' \cup \{j\})$ for all $S' \subseteq S \setminus \{i, j\}$, then $R_i^{SV}(S) = R_j^{SV}(S)$, i.e., they have the same Shapley value; $R_i^{SV}(S)$ scales linearly with respect to R (and thus with respect to the public cloud price P); if $i \in S$ does not increase savings in any federation, $R_i^{SV}(S) = 0$.

In particular, the last property implies that "free riders," the members that do not contribute to public cloud *savings*, are not rewarded by Shapley value. Note that a member can contribute by providing either resources (servers) or workload (overflow traffic of requests); for example, in a federation with two private clouds where one is overprovisioned and the other is underprovisioned, both members are necessary to generate public cloud savings, and their Shapley values are the same (specifically, $R(S)/2$, half of the generated savings).

Our analysis assumes that idle and busy servers have the same operational costs. If busy servers incur additional costs $P_{busy} - P_{idle}$ per request, then P can be replaced in Eq. (3) with $P - (P_{busy} - P_{idle})$: in this case, additional per-request costs are paid to the owners of the shared servers before calculating the public cloud savings to distribute among members of the federation (P_{idle} is already accounted for in Eq. (6) below). Note that when $P \leqslant (P_{busy} - P_{idle})$, i.e., the public cloud price is lower than the additional cost sustained by private clouds to put their resources to work, private clouds have no incentive to use their local resources (public cloud providers are cheaper); in the following, we assume that $P > (P_{busy} - P_{idle})$ and also $P > P_{busy}$, so that savings can be obtained by using local resources owned by private clouds.

B. Overall Reduction in Operating Costs

Public cloud costs are only a fraction of the operational costs of a private cloud. To assess the significance of public cloud savings, we consider the *relative cost reduction* of each private cloud i:

$$\delta_i = \frac{R_i^{SV}(S)}{P\beta(\vec{\lambda}_{\{i\}}, \vec{n}_{\{i\}}, \vec{0}_{\{i\}}, \vec{Q}_{\{i\}}) + n_i\xi} \tag{6}$$

where $R_i^{SV}(S)$ is the Shapley value distributed to i after joining the federation (due to public cloud savings), while $P\beta(\vec{\lambda}_{\{i\}}, \vec{n}_{\{i\}}, \vec{0}_{\{i\}}, \vec{Q}_{\{i\}})$ and $n_i\xi$ are the public cloud cost and private operation cost, respectively (these costs are sustained both before and after joining the federation). Based on [10], we assume that $\xi \approx 0.7P\mu$, i.e., serving a request with local servers is 30% less expensive than using cloud resources (after accounting for all operational costs such as energy, cooling, management).

C. Alternative Reward Policies

In this section, we consider alternative policies to share public cloud savings of the federation among private clouds $S = \{i_1, \ldots, i_K\}$, $S \subseteq \{1, \ldots, N\}$, with vectors of arrival rates $\vec{\lambda}_S$, number of servers $\vec{n}_S$, shared servers $\vec{m}_S$, and maximum mean waiting times $\vec{Q}_S$.

Shared Resources (SR). This policy distributes public cloud savings proportionally to the amount of resources shared by each private cloud:

$$R_i^{SR}(S) := \frac{m_i}{\sum_{j=1}^K m_j} R(S)$$

Note that this policy does not take into account the utilization of shared resources before or after joining the federation.

Shared Idle Resources (SIR). This policy distributes public cloud savings proportionally to the amount of shared resources *and to their utilization before joining the federation*:

$$R_i^{SIR}(S) := \frac{m_i(1-\rho_i)}{\sum_{j=1}^K m_j(1-\rho_j)} R(S)$$

where

$$\rho_i = \frac{\lambda_i - \beta(\vec{\lambda}_{\{i\}}, \vec{n}_{\{i\}}, \vec{0}_{\{i\}}, \vec{Q}_{\{i\}})}{n_i\mu}$$

is the utilization of the servers of private cloud i before joining the federation. With this policy, a higher fraction of cloud savings is received if shared servers had lower utilization ρ_i before joining the federation.

Shared Idle Resources and Workload (SIRW). This policy distributes public cloud savings proportionally to the amount of *shared idle resources and overflow traffic*:

$$R_i^{SIRW}(S) := \frac{1}{2}\left(R_i^{SIR}(S) + \frac{\lambda_i - \rho_i n_i\mu}{\sum_{j=1}^K \lambda_j - \rho_j n_j\mu} R(S)\right)$$

where $\lambda_i - \rho_i n_i\mu = \beta(\vec{\lambda}_{\{i\}}, \vec{n}_{\{i\}}, \vec{0}_{\{i\}}, \vec{Q}_{\{i\}})$ is the overflow traffic of private cloud i. With this policy, a higher fraction

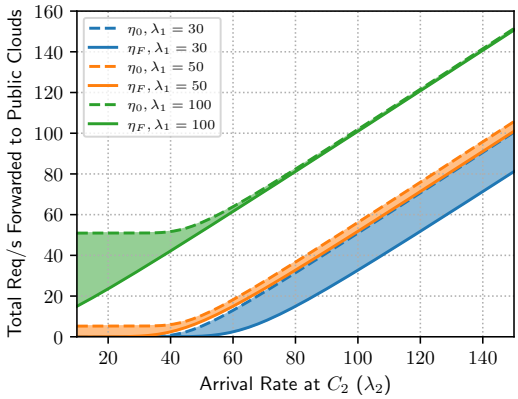


Figure 9: Rate of requests sent to public clouds by a federation of two clouds with $n_1 = n_2 = m_1 = m_2 = 50$ servers and $Q_1 = Q_2 = 0$, for different arrival rates λ_1 and λ_2.

of cloud savings is received if shared servers had lower utilization ρ_i before joining the federation, or if the private cloud was forwarding many requests to the public cloud.

Note that, while sharing policies SV and SIRW reward members of the federation contributing servers or workload, policies SR and SIR distribute public cloud savings only to members sharing servers. With SR or SIR, members are not rewarded for sharing workload: in particular, when $m_i = 0$ (no shared servers), private cloud i has no incentive to stay in the federation (since cloud costs are the same and no cloud savings are distributed to it, its profit would be the same without the federation). Nonetheless, we use policies SR and SIR in cases where $m_i > 0$ as additional baselines to illustrate the drawback of not considering workload contributions.

IV. Experimental Evaluation

In this section, we evaluate the effects of our federation mechanism on public cloud savings and analyze whether these savings are shared fairly among members of the federation.

All results presented in this section were obtained using PRISM [14] to solve our CTMC model through an iterative method (specifically, the *power method* with up to $200,000$ iterations and absolute threshold $\epsilon = 10^{-6}$ to detect convergence); results are thus exact (up to a numerical error). In our federations, we use $\mu = 1$ and vary $\vec{\lambda}$, $\vec{n}$, and $\vec{Q}$.

A. Public Cloud Savings and Operating Cost Reduction

Sharing All Servers. First, we evaluate the savings in public cloud costs achieved by a federation where two private clouds, C_1 and C_2, have no queueing of requests (i.e., $Q_1 = Q_2 = 0$) and share all of their $n_1 = n_2 = 50$ servers ($m_1 = m_2 = 50$). Fig. 9 shows the rate of requests forwarded to the public cloud before (dashed lines) and after (solid lines) forming the federation, i.e., η_0 and η_F as defined in Eqs. (4) and (5), respectively, for different arrival rates at C_1: $\lambda_1 = 30$ (blue lines), $\lambda_1 = 50$ (orange lines), $\lambda_1 = 100$ (green lines); the arrival rate at C_2 varies from $\lambda_2 = 10$ to $\lambda_2 = 150$.

- For $\lambda_1 = 30$, C_1 is overprovisioned and, when $\lambda_2 > 50$, it is able to serve part of the workload of C_2, which is underprovisioned. The federation achieves a reduction of the rate of requests forwarded to public clouds from $\eta_0 = 100.5$ to $\eta_F = 81.2$ (19.2%) when $\lambda_2 = 150$. Note that, as λ_2 increases, η_F grows linearly after $\lambda_2 = 75$ because the federation is overloaded, while the reduction in public cloud usage (the difference between η_0 and η_F) converges to a constant value.
- For $\lambda_1 = 50$, C_1 is close to underprovisioning: when $\lambda_2 < 50$, C_2 is overprovisioned, and it can serve some of C_1's overflow traffic; for $\lambda_2 > 50$, both C_1 and C_2 are underprovisioned, but some savings are still achieved by the federation (at $\lambda_2 = 150$, the rate of requests forwarded to public clouds is reduced by 4.5%, from $\eta_0 = 105.7$ to $\eta_F = 101.0$).
- For $\lambda_1 = 100$, C_1 is underprovisioned: when $\lambda_2 < 50$, C_2 is overprovisioned and can serve a large fraction of C_1's overflow traffic; for example, at $\lambda_2 = 10$, the rate of requests forwarded to public clouds is reduced by 70.6%, from $\eta_0 = 50.9$ to $\eta_F = 15.0$. When $\lambda_2 > 50$, the federation of C_1 and C_2 is heavily underprovisioned; the reduction in forwarding rate to public cloud is negligible (0.5% at $\lambda_2 = 150$, from $\eta_0 = 151.4$ to $\eta_F = 150.7$).

(λ_1, λ_2)	η_0	η_F	δ_1	δ_2
$(30, 150)$	100.5	81.2	27.6%	7.1%
$(50, 150)$	105.7	101.0	5.9%	1.8%
$(100, 150)$	151.4	150.7	0.4%	0.3%

Table I: Forwarding rate to public clouds before (η_0) and after (η_F) joining the federation and relative cost reduction (δ_1, δ_2) for different arrival rates when $\vec{n} = \vec{m} = (50, 50)$, $\vec{Q} = (0, 0)$.

Table I reports the relative cost reduction from Eq. (6) for these scenarios where all servers are shared with the federation.

Sharing a Fraction of the Servers. Next, we evaluate the increase in public cloud savings with respect to the amount of resources shared by the two private clouds, C_1 and C_2, with the same number of servers $n_1 = n_2 = 50$ and QoS $Q_1 = Q_2 = 0$.

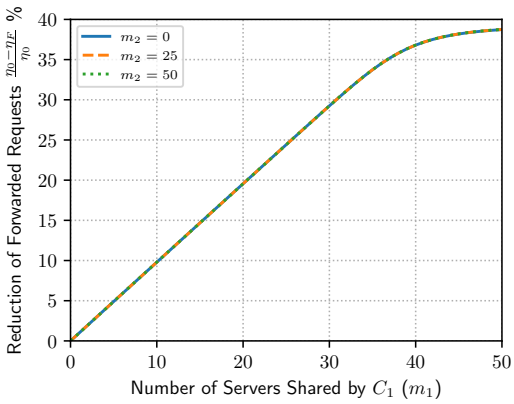


Figure 10: Reduction of requests sent to public clouds by two federated clouds with $n_1 = n_2 = 50$ servers, $Q_1 = Q_2 = 0$, $\lambda_1 = 10$, $\lambda_2 = 150$ for different sharing strategies m_1 and m_2.

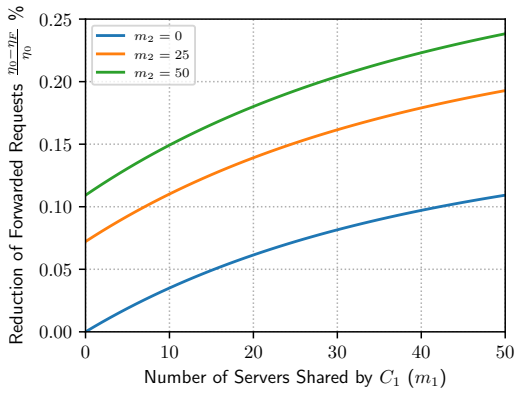


Figure 11: Reduction of requests sent to public clouds by two federated clouds with $n_1 = n_2 = 50$ servers, $Q_1 = Q_2 = 0$, $\lambda_1 = \lambda_2 = 150$ for different sharing strategies m_1 and m_2.

Fig. 10 illustrates that, since C_1 is overprovisioned ($\lambda_1 = 10$) and C_2 is heavily underprovisioned ($\lambda_2 = 150$), the amount of servers shared by C_2, m_2, has no effect on the savings in public cloud costs (these resources have very high utilization); instead, the percentage reduction of the rate of requests sent to public clouds increases linearly with m_1, the servers shared by C_1 (which is overprovisioned), until converging to 38.7% savings when the shared server pool becomes highly utilized. Fig. 11 shows that, when both clouds are underprovisioned ($\lambda_1 = \lambda_2 = 150$), public cloud savings are still monotonic with respect to the amount of resources shared, but almost negligible (at most 0.24%, when all resources are shared).

Heterogeneous QoS Requirements. Fig. 12a presents the forwarding rates to the public cloud before (solid line) and after (dashed line) two private clouds C_1 and C_2 with heterogeneous QoS $\vec{Q} = (0, 1)$ share all of their resources; in this scenario, $\mu = 1$ and $\lambda_1 = n_1 = 50$, while λ_2 varies from 30 to 100 and $n_2 = \lambda_2$ (so that the load $\lambda_2/(n_2\mu) = \lambda_1/(n_1\mu) = 1$ is always the same for both private clouds). We observe that while C_2 no longer forwards requests to the public cloud after joining the federation (orange line), the forwarding rate increases for C_1 (blue line). Notably, *the cumulative forwarding rate is greater after sharing resources in the federation* and, as the workload of C_2 increases, a larger fraction of requests of C_1 (which cannot be queued at the shared server pool) is forwarded to the public cloud. Fig. 12b shows that, for $\vec{\lambda} = \vec{n} = (50, 50)$, the disadvantage of C_1 (i.e., the increase in requests forwarded to the public cloud after joining the federation) is greater as Q_2 increases, allowing more requests of C_2 to queue at the shared server pool; when C_2's forwarding rate reaches 0, C_2 does not need additional resources and C_1's forwarding rate stops increasing. Finally, Fig. 12c considers a scenario with an increasing number N of hybrid clouds, where $\lambda_1 = n_1 = m_1 = 1000$ and $Q_1 = 1$ for C_1, while $\lambda_i = n_i = m_i = 10$ and $Q_i = 0$ for $i = 2, \ldots, N$. In this case, while the federation is initially inefficient (the average rate of requests sent to the public cloud is greater than it would be without the federation) because of C_1 (which is able to queue requests at the shared pool), it becomes advantageous when $N > 10$ hybrid clouds with $Q_i = 0$ are present.

To evaluate the impact of heterogeneous QoS requirements on operating costs of private clouds sharing their resources, we show in Table II that the relative cost reductions δ_1 and δ_2 can be negative (i.e., public cloud costs can *increase*) when two private clouds C_1 and C_2 have significantly different QoS requirements. In the example, C_2 has greater workload (λ_2), number of servers (n_2), and allowed mean waiting time for its requests (Q_2); when C_1 and C_2 share all of their resources, C_1 is not able use the shared server pool, where requests of C_2 can wait in a queue but requests of C_1 must be served immediately. As a result, most requests of C_1 are forwarded to the public cloud, increasing public cloud costs. While Shapley value distributes additional costs equally, the relative increase in operating costs is greater for C_1 (a smaller private cloud).

(λ_1, λ_2)	(n_1, n_2)	(Q_1, Q_2)	δ_1	δ_2
$(50, 100)$	$(50, 100)$	$(0, 10)$	-3.7%	-2.1%
$(50, 100)$	$(50, 100)$	$(0, 50)$	-3.8%	-2.2%
$(50, 1000)$	$(50, 1000)$	$(0, 50)$	-14.8%	-0.8%
$(7, 10000)$	$(10, 10000)$	$(0, 1000)$	-37.4%	-0.05%

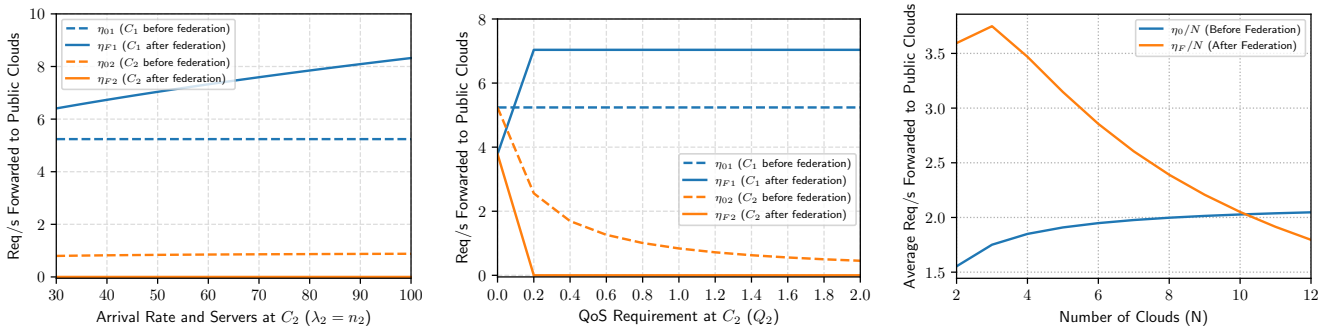
Table II: Relative cost reduction with heterogeneous QoS

B. Reward Policies

To motivate the use of Shapley value (SV) in our system, we evaluate the effects of alternative policies defined in Section III-C: shared resources (SR), shared idle resources (SIR), and shared idle resources and workload (SIRW). First, in Fig. 13, we compare the rewards assigned in three federation scenarios, each with three private clouds C_1, C_2 and C_3.

Federation 1. In this federation, $\vec{\lambda} = (10, 70, 140)$, $\vec{n} = (30, 50, 100)$, $\vec{m} = \vec{n}$ (all servers are shared), $\vec{Q} = (0, 0, 0)$ (no queueing is allowed): C_1 is overprovisioned (contributing most idle servers), while C_2 and C_3 are similarly underprovisioned; notably, C_3 contributes more servers and workload. SV assigns most cloud savings to C_1, the only provider of idle servers; C_2 and C_3 are rewarded similarly, with higher reward assigned to C_3 due to its higher shared workload. In contrast, SR unfairly rewards C_3 for sharing most servers, although its servers have very high utilization and cannot serve workload from other members of the federation. SIR addresses this issue by taking resource utilization into account: most of the reward goes to C_1 for sharing idle servers. This assignment is also unfair, since the workload shared by C_2 and C_3 is necessary to obtain public cloud savings; SIRW accounts for shared workload, obtaining a more balanced reward distribution.

Federation 2. In this federation, $\vec{\lambda} = (30, 30, 150)$, $\vec{n} = (50, 100, 50)$, $\vec{m} = \vec{n}$ (all servers are shared), $\vec{Q} = (0, 0, 0)$ (no queueing is allowed): C_1 and C_2 are overprovisioned, while C_3 is heavily underprovisioned. SV assigns most cloud savings to C_3 (for sharing its workload) and to C_2 (for sharing more servers than C_1, and with lower utilization). SR rewards C_2



(a) Increasing C_2's arrivals and servers (b) Increasing C_2's QoS (c) Increasing number of clouds

Figure 12: Forwarding to public cloud before (dashed lines) and after (solid lines) clouds with different QoS share all resources

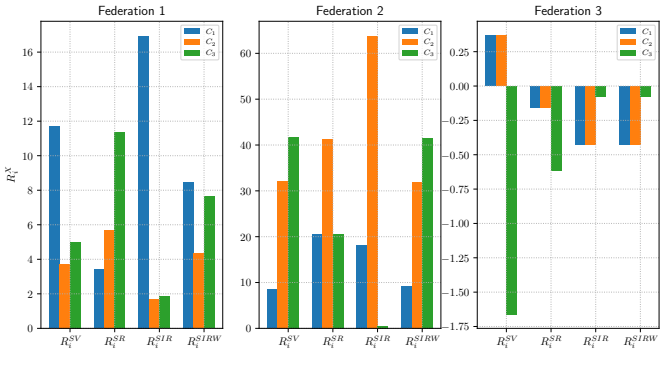


Figure 13: Comparison of different reward policies

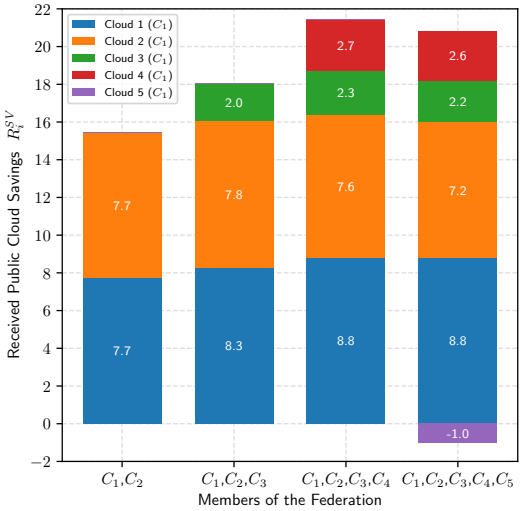


Figure 14: Reward distributed for different federations

for sharing most servers, and C_2's reward is even greater with SIR, since its shared servers have low utilization; instead, the reward of C_3 (providing workload) is much lower, especially with SIR. SIRW achieves a reward distribution very similar to SV, illustrating that Shapley value is rewarding members of the federation for both shared workload and servers.

Federation 3. In this federation, $\vec{\lambda} = (50, 50, 200)$, $\vec{n} = (50, 50, 200)$, $\vec{m} = \vec{n}$ (all servers are shared), $\vec{Q} = (0, 0, 1)$: all private clouds have similar loads (close to underprovisioning), but C_3 can queue requests, with a buffer of up to $s_F \mu Q = 300$ requests. The shared federation pool will not be available to serve traffic of private clouds C_1 and C_2, which will forward most of their requests to the public cloud, resulting in an *increase* of public cloud costs. In contrast, a federation including only C_1 and C_2 results in public cloud savings. For this reason, SV assigns negative reward to C_3 (its presence reduces the profit of the federation) and positive reward to C_1 and C_2 (their presence increases the profit of the federation). Other policies are not able to distinguish between the negative contribution of C_3 and the positive contributions of C_1 and C_2.

Next, we analyze profit when public cloud savings are distributed according to Shapley value and up to 5 private clouds $C_1, C_2, \ldots, C_5$ join the federation. In this setting, we have $\vec{\lambda} = (10, 50, 30, 50, 100)$, $\vec{n} = (30, 30, 30, 50, 100)$, and $\vec{Q} = (0, 0, 0, 0, 1)$; we assume that all servers are shared after joining the federation ($\vec{n} = \vec{m}$). The results, illustrated in Fig. 14, show that, as we add more clouds to the federation, C_1 and C_2 are still responsible for most of the public cloud savings, since C_1 is overprovisioned while C_2 is underprovisioned. Instead, clouds C_3 and C_4 have similar loads before joining the federation, and they are responsible for a lower fraction of public cloud savings, thus receiving lower rewards from Shapley value. In contrast, cloud C_5 has negative reward (i.e., it is charged to be part of the federation): due to its higher allowed mean waiting time $Q_5 = 1$, requests of C_5 can use servers from the shared pool more frequently; this reduces the ability of C_1 and C_2 to share their resources and workload, respectively, and hence C_5 is penalized by Shapley value.

V. Related Work

A fairly large literature on hybrid clouds and their characteristics exists; as representative examples, [23] describes the operation of hybrid cloud architectures, while [25] analyzes their advantages in reducing cost under high workload variability. Similarly, a number of works consider cloud federations, with representative examples including [20], [17].

Different models of private cloud federations are presented in [9], where members of the federation are modeled as $M/M/1$ queues sharing either requests or capacity; "reward-driven" or "joint-business" mechanisms of cooperation are considered to evaluate the effects on the profit of federation members. A model of federated small clouds similar to federations of hybrid clouds is presented in [16], where a CTMC model is adopted to evaluate performance metrics used in a repeated game among federation members. Notably, our work proposes a more detailed model with a simpler cooperation mechanism: we model private clouds as $M/M/n/m$ queues (in contrast with $M/M/1$ queues of [9]) where members do not need priority over their shared resources (in contrast with [16]); our cooperation mechanism is based on *sharing of cloud savings*, while [9], [16] assume either a joint business or a pricing model for services offered to members of the federation, or to external customers. Shapley value [21] is adopted as in [9] and in many works on coalitions.

Another notable difference is that a number of papers (for example [15]) do not consider QoS, while others (for example [4]) do not consider hybrid clouds, thus leading to different models of cooperation within the federation (e.g., borrowing resources at a price from individual members instead of optimizing the amount of shared resources).

While several works propose CTMC models to evaluate the performance of hybrid or federated cloud systems [16], [8], [18], [7], [11], our work, by leveraging theoretical results [22], provides proofs on the advantage of hybrid cloud federations (with homogeneous QoS), together with their optimal policies.

Finally, while many works [6], [12], [24], [19] present *analytical* models of performance and revenue in cloud federations to maximize profit of their members, we focus on stochastic models accounting for workload variability.

VI. Conclusions

In this paper, we proposed a CTMC model to predict the rate of requests forwarded to public cloud providers by a federation of hybrid clouds, where each member can share workload or resources to generate cost savings while satisfying QoS requirements. When all members have the same QoS requirements, we provided theoretical results showing that sharing all resources is the best strategy. For heterogeneous QoS requirements, we provided a solution to evaluate different sharing strategies, illustrating how sharing all resource can be, in fact, counterproductive for the members of the federation.

As a cooperation mechanism, we proposed sharing of public cloud savings according to Shapley value. Through our experimental evaluation, we compared Shapley value with alternative sharing policies rewarding members of the federation for the amount of shared resources and workload. The results illustrate the ability of Shapley value to reward both, and to discourage (through negative payoffs) members of the federation reducing its cloud savings.

As future work, we plan to extend our theoretical results to characterize optimal policies with heterogeneous workloads (e.g., where requests can require different amounts of work) and resources (e.g., different classes of VM instances, each with different tradeoff between performance, cost, and availability in the federation).

References

[1] CloudStack: Open Source Cloud Computing. cloudstack.apache.org.

[2] OpenNebula: Open Source Cloud and Edge Computing. opennebula.io.

[3] OpenStack: Open Source Cloud Computing Infrastructure. openstack.org.

[4] R. Abozariba, A. Amjad, and M. Patwary. Optimized resource sharing for federated cloud services with desired performance and limited opex. In *IEEE Global Communications Conference (GLOBECOM)*, 2017.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, 2010.

[6] M. R. M. Assis and L. F. Bittencourt. Multicloud tournament: A cloud federation approach to prevent free-riders by encouraging resource sharing. *J. Netw. Comput. Appl.*, 166:102694, 2020.

[7] J. Bi, Z. Zhu, R. Tian, and Q. Wang. Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center. In *IEEE CLOUD 2010*, pages 370–377. IEEE Computer Society, 2010.

[8] D. Bruneo. A stochastic model to investigate data center performance and qos in iaas cloud computing systems. *IEEE Trans. Parallel Distributed Syst.*, 25(3):560–569, 2014.

[9] G. Darzanos, I. Koutsopoulos, and G. D. Stamoulis. Cloud federations: Economics, games and benefits. *IEEE/ACM Trans. Netw.*, 27(5):2111–2124, 2019.

[10] C. Fisher et al. Cloud versus on-premise computing. *American Journal of Industrial and Business Management*, 8(09):1991, 2018.

[11] R. Ghosh, F. Longo, R. Xia, V. K. Naik, and K. S. Trivedi. Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. *IEEE Trans. Serv. Comput.*, 7(4):667–680, 2014.

[12] T. Halabi, M. Bellaiche, and A. Abusitta. A cooperative game for online cloud federation formation based on security risk assessment. In *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 83–88. IEEE, 2018.

[13] L. Kleinrock. *Queueing Systems, Vol. 1: Theory*. Wiley and Sons, 1975.

[14] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: verification of probabilistic real-time systems. In *Computer Aided Verification*, volume 6806, pages 585–591, 2011.

[15] K. Li. Profit maximization in a federated cloud by optimal workload management and server speed setting. *IEEE Trans. Sustain. Comput.*, (01):1–1, 2021.

[16] S. Lin, R. Pal, M. Paolieri, and L. Golubchik. Performance driven resource sharing markets for the small cloud. In *ICDCS 2017*, pages 241–251. IEEE Computer Society, 2017.

[17] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures. *Computer*, 45(12):65–72, 2012.

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

[19] B. K. Ray, A. Saha, S. Khatua, and S. Roy. Toward maximization of profit and quality of cloud federation: solution to cloud federation formation problem. *Journal of Supercomputing*, 75(2):885–929, 2019.

[20] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. S. Montero, Y. Wolfsthal, E. Elmroth, J. A. Cáceres, M. Ben-Yehuda, W. Emmerich, and F. Galán. The reservoir model and architecture for open federated cloud computing. *IBM J. Res. Dev.*, 53(4):4, 2009.

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

[22] D. R. Smith and W. Whitt. Resource sharing for efficiency in traffic systems. *Bell System Technical Journal*, 60(1):39–55, 1981.

[23] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. T. Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Comput.*, 13(5):14–22, 2009.

[24] Y. Wang and H. Chen. Dynamic resource arrangement in cloud federation. In *2012 IEEE Asia-Pacific Services Computing Conference*, pages 50–57. IEEE, 2012.

[25] J. Weinman. Hybrid cloud economics. *IEEE Cloud Comput.*, 3(1):18–22, 2016.