Information Systems Frontiers (2023) 25:1495–1512
https://doi.org/10.1007/s10796-022-10302-4

Availability-Aware Virtual Resource Provisioning for Infrastructure Service Agreements in the Cloud

Shuai Yuan[1] · **Sanjukta Das**[2] · **Ram Ramesh**[2] · **Chunming Qiao**[3]

Accepted: 3 June 2022 / Published online: 20 June 2022

Abstract
Service availability is a key construct in Service Level Agreements (SLA) between a cloud service provider and a client. The provider typically allocates backup resources to mitigate the risk of violating the SLA-specified uptime guarantee. However, initial backups may need to be adjusted in response to real-time failure and recovery events. In this study, we first develop a recurrent intervention at fixed intervals (RIFI) strategy that allows the provider to adjust the allocation of backup resources such that the expected total cost is minimized. Next, we focus on the limit to number of interventions, starting from single intervention strategy, as frequent reallocations may be operationally disruptive. Particularly, we provide a cost minimization approach to guide the service providers in their virtual resources management, and a specific downtime minimization approach for more mission-critical applications as a more aggressive alternative. We present computational results exploring the impact of intervention on the likelihood of SLA violation for the rest of the contract period, and evaluate parameters such as time and quantum of resource level adjustment, penalty levels desired by clients, and their influences on the backup resource provisioning strategies. We also validate our models through the analysis of use cases from Amazon Elastic Compute Cloud. Finally, we summarize this study by providing key practical managerial implications for resource deployment in the availability-aware cloud.

Keywords SLA · Resource provisioning · Resource adjustment · Cloud computing

1 Introduction

Infrastructure-as-a-Service (IaaS) is an established paradigm in cloud computing for the provisioning of baseline services such as virtualized compute nodes, data storage and network connectivity (Mell & Grance, 2011). Server instances, which are also referred as virtual machines (VMs), comprise of various combinations of CPU, memory, storage and networking capabilities, and are commonly offered by cloud service providers such as Amazon Web Services (AWS), Google, Microsoft, Rackspace, and SalesForce (Randal, 2020; Smith & Nair, 2005) . The clients, ranging from individuals and small institutions to large companies, can rent and pay only for a set of VMs that are actually needed according to usage-based (pay-as-you-go) posted-pricing models.[1] The underlying business models of the cloud vendors, besides aiming to monetize their installed excess capacities, are intended to alleviate their clients' concerns about the risks of capacity under-utilization (when resources are over-provisioned) and demand non-fulfillment (when

✉ Shuai Yuan
syuan@brocku.ca

Sanjukta Das
sdsmith4@buffalo.edu

Ram Ramesh
rramesh@buffalo.edu

Chunming Qiao
qiao@computer.org

[1] Department of Finance, Operations, and Information Systems, Goodman School of Business, Brock University, St. Catharines, Ontario L2S 3A1, Canada

[2] Department of Management Science and Systems, School of Management, University at Buffalo, State University of New York, Buffalo, New York 14260, USA

[3] Department of Computer Science and Engineering, School of Engineering and Applied Sciences, University at Buffalo, State University of New York, Buffalo, New York 14260, USA

[1] https://aws.amazon.com/ec2/dedicated-hosts/pricing/

resources are under-provisioned), if the clients choose to employ in-house infrastructure. Thus, IaaS has emerged as a major service offering in the cloud computing domain.

Server virtualization plays an important role in providing infrastructure services in the cloud. A VM is an emulation of a complete computer system and provide the full functionality of a physical server (Smith & Nair, 2005). Accordingly, multiple VMs can simultaneously and independently be implemented and run in a single physical server. The virtual computing infrastructure of a cloud data center is therefore a collection of physical servers where each server would host a set of VMs. Under this framework, a client could request VM instances for a stated period of time and the provider would offer a pricing scheme for different configurations of VMs at specific levels of service availability. When the assured service availability is not fulfilled in a contract period, most service providers offer compensation in the form of service credits to the clients which can be realized in a following contract period. Such compensation is a penalty borne by the service provider, which also ensures a continuity of the service contract with the clients.

Ensuring continuity of service over long periods of time is a major challenge in most cloud datacenters (Martens & Teuteberg, 2012). These datacenters are known to be susceptible to different types of failures from frequent small-scale failures (such as disk failures) to less frequent but more catastrophic failures (such as power distribution unit or network node failures) (Dean, 2009). Failures can occur with either a VM or a physical server; since multiple VMs are hosted in a physical server, a server failure could significantly disrupt service continuity across a whole range of VMs. In this context, the notion of a penalty for the violation of the assured availability embedded in a Service Level Agreement (SLA) between a service provider and a client becomes crucial. Typically, an SLA articulates contract parameters such as contract duration, price, service availability (usually specified as an uptime guarantee), and the penalty to be paid by the provider for violation of the uptime guarantee. The latest ITIC 2020 survey data found that 87% of respondents now consider 99.99% to be the minimum acceptable levels of availability for their mission critical business servers (ITIC, 2020). In order to fulfil the uptime guarantee and reduce the chances of incurring significant penalties, a common strategy followed by providers is to allocate a set of backup VMs to substitute for failed primary VMs as needed. The service incurs *downtime* only when *all* of the backup VMs are being utilized in place of failed primary VMs, *and* there is at least one additional failed primary VM for which there is no excess capacity to mitigate its failure. The accumulated downtime for these specific disruptions are captured over the contract period, and this determines the penalty cost. While the likelihood of SLA violation can be reduced by providing more backup VMs, this would also increase the VM provisioning cost. Since the client is charged only for the VMs specified in the contract, it is important to determine the optimal number of backup VMs such that the expected total cost, which is an aggregation of the provisioning cost and the potential penalty cost, is minimized.

However, once the optimal number of backup VMs is determined, it may be unrealistic to hold it static for the duration of the contract. System failures and recoveries occur randomly, and fixed backup allocations could become suboptimal when the contracted service is fully carried out. This is especially important in data centers where system downtimes are frequent and significant. Hence, the present work develops a cost-effective resource allocation strategy that dynamically manages the backup resources by responding to the observed downtime as it evolves during the course of a contract. This is also a better reflection of reality as IaaS systems scale client resources up or down as the end user demand waxes and wanes due to extrinsic factors that may be driving that demand. By observing the cumulative incurred downtime as the service period advances, a cloud data center could dynamically intervene and adjust the backup allocations by adding or removing backup VMs at specific intervals to yield significantly better cost performance. Such interventions, while helpful, could also be disruptive to the clients due to their impact on performance and could also be costly to operationalize for the provider; thus, in order to benefit from a possibly limited number of such interventions, the provider needs to determine the optimal intervention strategy comprising of the *timing* and the *quantum* of changes in the backup allocations for a service contract.

The organization of the paper is as follows. Section 2 addresses the research context and highlights our research contributions. Section 3 presents the related work. Section 4 develops the algorithm for recurrent interventions at fixed intervals (RIFI) strategy and Section 5 develops the algorithms for the single intervention at random interval (SIRI) strategy. Section 6 presents the computational results and Section 7 presents the validation of the strategies using the Amazon EC2 service structure. The managerial implications, conclusions and directions for future research are summarized in Section 8.

2 Research Context and Contributions

The provisioning of the backup VMs ensures an appropriate level of tolerance to failures. The backup provision and the associated rollback recovery process when failures occur are collectively known as check-pointing (Marques et al., 2005), and clear industry guidelines for this are available.[2]

[2] http://technet.microsoft.com/en-us/library/bb740891.aspx

Various models of check-pointing exist, and the widely used models are the *powered-on* and *powered-off* check-pointing schemes. In the powered-on scheme, the n primary VMs are supported with $k(>1)$ backups; each backup provides a replication of all the primary VM's states and data images; and, the backups are updated periodically. Accordingly, when a primary VM fails, an available backup would serve as a substitute until the primary VM is restored. In the powered-off scheme, typically a large central server is used to periodically capture and store the states and images of all the VMs. Accordingly, when a VM fails, the central server restarts the affected processes by rolling back to the most recently updated snapshot of the failed VM available at the central server. While both the schemes provide more or less the same check-pointing functionality, their implementations could differ Du et al. (2015). Modeling the failures and recoveries of VMs as a birth-death process, Du et al. (2015) develop sample path randomization algorithms to estimate the probability distribution of downtime in a contract period under both the check-pointing schemes. Using the estimated downtime distributions under the powered-on scheme without loss of generality, Yuan et al. (2018) develop VM pricing-penalty schedules for a range of client requirements on availability. They derive the optimal number of backups to be provisioned up front in a contract for specified availability requirements, prices charged and penalties offered, by minimizing the expected total cost over the contract period to the service provider.

A major limitation of the study of Yuan et al. (2018) is the deployment of a fixed number of backups throughout the contract period. This is carried out by minimizing the expected total cost at the beginning of the contract period and holding the resulting number of backups constant throughout. Motivated by the prior limitations, we develop the following strategy for the dynamic management of backup resources in this paper. At the commencement of service, the provider would derive the optimal backup provision as in Yuan et al. (2018). Once the service starts, the downtime is continuously monitored. At specific intervention times, if the actual service level delivered is less than a threshold value, it may be advantageous for the provider to add more backups for the shortfall; similarly, if the delivered service level is more than another threshold value, it may be advantageous to remove some allocated backups. Such interventions can be carried out nearly seamlessly in most data centers and are opaque to the client since the adjustments occur at the backup VM level only. In this research, we develop two strategies for intervention: recurrent interventions at fixed intervals, and single intervention at random interval. The recurrent interventions are appropriate for large data centers where the backup adjustments can be carried out relatively seamlessly; the single intervention strategy is suitable for mission-critical client operations with limited tolerance for service disruptions. We develop algorithms for the optimal interventions under each strategy and evaluate their performance using extensive computational studies. We also validate these strategies with use-cases constructed from Amazon Elastic Compute Cloud (EC2) service structures.

Infrastructure vendors in the cloud offer a variety of flexible contracts that include different pricing mechanisms, penalty structures and terms of usage by the clients. Based on a survey of 19 leading vendors over 27 types of services offered, Kauffman et al. (2015) classify infrastructure services into two broad categories: *reserved* services and *on-demand* services. For example, Amazon EC2 offers four such types: on-demand instances, spot instances, reserved instances and dedicated hosts. Their on-demand and spot instances can be classified as on-demand services, while the reserved and dedicated instances are reserved services. The basic difference between the two forms of services is in the client's commitment for long-term usage. The spot instances are for instantaneous usage, while the on-demand, reserved and dedicated instances can be viewed as commitments of short-, medium- and long-term usage, respectively. The study of Yuan et al. (2018) focuses on the reserved and dedicated instances of service, and models the pricing, penalties and resource provisioning trifecta for clients with medium and long term contracts. The current study is a further development of this work and focuses on the dynamic management of VM resources under flexible service contracts. Under this contextual setting, we first review the work of Yuan et al. (2018) and then summarize the central contributions of this research in the following discussion.

When a client requests n VMs (denoted as primary VMs), an additional k backup VMs are provided in a contract. Hence, the assured availability of n VMs will be disrupted only if at least $(k+1)$ VMs (out of the total $(n+k)$ VMs) simultaneously fail. Yuan et al. (2018) first model the allowable downtime (i.e., without incurring a penalty) available in a contract period as a perishable commodity. This is analogous to the supply in an inventory context. Similarly, they model the cumulative downtime incurred over time in the contract period as a random demand process for the available supply of permissible downtime. The incurred downtime is a non-increasing function of the level of backups (k) provided. Du et al. (2015) estimate the probability distribution of the incurred downtime over an interval of time for a configuration of n primary VMs and k backups under powered-on scheme of check-pointing. Therefore, we can estimate the expected downtime in a contract period T for any VM configuration (n,k). The expected penalty cost is incurred when the expected downtime in a contract period exceeds the permissible downtime. Assuming a penalty rate for each unit of downtime in excess of the allowable downtime, the expected penalty cost for the contract is determined. Note that the

Table 1 Literature Search Results

Topic	Database	Keywords	Number of Publications
Service Availability Management	IEEE	"Availability" AND "Cloud Computing"	177
	Business Source Complete		16
Resource Provisioning and Allocation	IEEE	"Resource Allocation" AND "Cloud Computing"	576
	Business Source Complete		61

expected penalty cost is non-increasing and nonlinear in k, while the backup VM provisioning cost linearly increases with k. In an analogous EOQ inventory model, the expected penalty cost corresponds to the holding cost and provisioning cost corresponds to the ordering cost. Yuan et al. (2018) show that the total cost function is convex under certain conditions. Therefore, the optimal level of backup to be allocated at the beginning of the contract period will be the level k_0 that minimizes the total cost. When the client specifies a requirement of n VMs at an availability level α, the optimal configuration (n, k_0) and its associated total cost are determined. Using this total cost as the baseline, Yuan et al. (2018) develop a price-penalty schedule that breaks even with the total cost. Adding appropriate profit margins under a determination of the client's willingness to pay and the prevailing market prices, a service provider could offer such a schedule to the client who then could choose an appropriate combination from the schedule.

In the above framework, Yuan et al. (2018) almost entirely focus on deriving the price-penalty schedule and do not consider the optimal deployment and management of backup VMs at runtime. Note that the level k^* that is determined up front may result in either an overall under-provisioning or over-provisioning of backups when evaluated at run time, since the cumulative incurred downtime is stochastic. This indicates that a continuous monitoring of the incurred downtime and appropriately adjusting the backup provision could lead to better cost performance than the expected total cost determined upfront at the beginning of the contract period as in Yuan et al. (2018). Accordingly, we focus on this research problem and develop customer-centric dynamic resource management strategies in this work. Specifically, we differentiate the clients depending on their risk preferences on the availability guarantee and seamless maintenance of services, and the nature of mission-criticality of applications running in the cloud. The RIFI strategy will be appropriate for clients requiring relatively high levels of availability, have relatively lower tolerance to downtime and higher tolerance to backup handover delays, and are less cost-sensitive; the SIRI strategy under downtime minimization will be appropriate for clients running mission-critical applications that require high levels of availability, have relatively lower tolerance for downtime and handover delays, and are less cost-sensitive; and the SIRI strategy under cost minimization will be suitable for most other clients.

3 Related Work

We review the literature in the cloud IT domain on service availability management models, resource provisioning mechanisms, and dynamic resource allocation strategies from the provider's perspective. Considering the interdisciplinary nature of research on cloud computing, we select IEEE in engineering and technology and Business Source Complete in business disciplines as the database sources. The conducted search includes papers published between January 2010 and January 2022.We summarize search results in Table 1 and discuss the most relevant research as follows.

Service Availability Management The issue of availability of cloud services and applications is a primary concern among IT professionals as pointed out in a 2012 global survey (Cisco, 2012) of more than 1300 IT decision makers in over 13 countries aimed at better understanding the top priorities and challenges during cloud migration. At present, major cloud service providers have set a high standard in this regard. Our conversations with cloud service thought-leaders in the industry inform us that it is going to be increasingly more common to see commitments upwards of 5-nines (99.999%) and even 7-nines. Since availability analysis provides a foundation to design the underlying cloud infrastructure capable of satisfying pre-determined SLA, a useful tool to evaluate the resiliency of the cloud service, and a requirement to quantify quality of service (QoS) experienced by the client, a number of models and frameworks are developed in the literature. Bruneo (2014) presents a stochastic-reward-nets model to evaluate the performance in an IaaS cloud. Availability, with other metrics such as utilization and responsiveness, are defined and investigated under different cloud-specific strategies. Jammal et al. (2018) extends CloudSim, a simulation framework on cloud infrastructure management, by incorporating high availability-aware modeling and scheduling. Multiple allocation techniques are evaluated through ACE and the

availability analysis of any placement solution are provided including the estimates of availability under various events such as failure and recovery. Jammal et al. (2016) propose an analytical model based on stochastic Petri Net to assess the availability of cloud services and their components in geographically distributed data centers. Both inter- and intra-data centers deployments, different types of failures, and redundancy approaches have been considered in this study. Ghosh et al. (2014a) quantify the availability in the IaaS cloud context through an interacting Markov Chain method. Three pools (hot, warm and cold) are modeled where physical machines may migrate from one to another pertaining to failure/repair events. Silic et al. (2014) develop a model to predict the user-perceived availability of web service by considering four-dimensional historic invocation data space: service load, user location, service class, and service location. However, the existing literature has not considered the availability management through the modeling on high level infrastructure resources allocation, and its impact on profitability of service contracts.

Resource Provisioning Due to the increasing operating and maintenance costs associated with the rapid growth of datacenter size, the number of clients and their demand for computing and storage instances, substantial amount of research has focused on the resource provisioning and capacity planning vis-à-vis either SLA requirements or monetary optimization. We address the research in this area along two dimensions: SLA-aware studies and Cost minimization/revenue maximization studies.

SLA-aware Studies: Van et al. (2009) design a two-stage resource management system which integrates both SLA fulfillment and the operating costs, by first determining the allocation of VMs to optimize a global utility as VM provisioning, following by the VM packing phase to minimize the number of active servers. Goudarzi et al. (2012) study the SLA-based resource provision problem to minimize the operational cost including power and migration cost by effective VM placement. Wu et al. (2014) proposes algorithms to reduce infrastructure VM cost and to improve customer satisfaction level by minimizing SLA violations in the cloud of Software-as-a-Service (SaaS) through resource reservation and requests rescheduling strategies. Singh et al. (2017) develop a SLA-aware autonomic technique to reduce SLA violation rate by fulfilling QoS requirements, which includes availability, latency and execution time. Yala et al. (2018) study the trade-off between deployment cost and criteria of service availability on a video content delivery service. CART model is established in (Mateo-Fornés et al., 2019) to minimize the required VM resources while ensuring the agreed availability level and response time in the SaaS cloud. This model is capable of providing guidelines for the service provider to improve client satisfaction through the trade-off between quality of service and the costs. (Yang et al., 2014) propose a regression model to predict the workload and then presents an auto-scaling approach with three techniques: self-healing, resource level, and VM level Scaling in the cloud. Both lower costs and lower SLA violation have been achieved through this approach. (Panda et al., 2019) design three task scheduling algorithms for a heterogeneous multi-cloud environment and each contains three steps on VM placement: matching, allocating, and scheduling. These algorithms outperform than the traditional Min-Min and Max-Min approach in the simulations regarding SLA metrics such as processing time, average cloud utilization and throughput are used.

Cost minimization/revenue maximization studies: Pertaining to the stream of cost-effectiveness analyses, Mansouri et al. (2019) introduce both offline and online algorithms aiming at optimizing the cost that consists of residential cost (i.e., storage, put and get costs) and potential migration cost (network cost) for cloud storage providers. Toosi et al. (2015) design a revenue-maximization framework for the optimal capacity planning by means of admission control. A joint decision on reservation, spot markets, and on-demand pricing policies are supported by this work for the IaaS cloud providers. Chase and Niyato (2015) combine both VM and bandwidth provisioning into the optimization models to mitigate the risks of demand and price uncertainty. A scenario tree reduction approach has been adopted to make its solution more scalable. Ghosh et al. (2014b) study the cost-availability trade-offs in an IaaS cloud by addressing two cost minimization problems: to minimize the total cost of ownership (TCO) of a cloud service and to minimize total infrastructure and downtime cost. Wang et al. (2008) develop an autonomic resource management model which enables allocating server capacity based on the estimated service levels. Differentiated service qualities are provided by this system whilst improving overall performance and reducing usage cost. A genetic algorithm is designed in (Gutierrez-Garcia & Sim, 2012) for Bag-of-tasks (BoT) applications constrained by budgets and deadlines in multiple cloud environments. (Hassan et al., 2014) provides cooperative game theory based VM resource allocation mechanisms for IaaS providers. It is demonstrated that a cost-effective game is achieved and can motivate providers to cooperate in a horizontal dynamic cloud federation (HDCF) platform. To the best of our knowledge, our work is the first of its kind to create policies for dynamic resource provisioning, while managing the risks and costs associated with a critical SLA-specified condition, the availability or uptime guarantee.

Dynamic Resource Allocation Models and results are also presented that assess dynamic resource provisioning, especially for IaaS cloud services. Ran et al. (2017) focus on a dynamic instance provisioning strategy with cost optimization and QoS guarantee, as well as a reserved instance provisioning strategy for further total cost optimization. Mistry et al. (2018) propose a dynamic optimization approach for service composition from the IaaS providers' perspective, where the stochastic arrival of the requests and the long-term economic model of the provider are taken into consideration. Guo et al. (2019) develop online algorithms using dynamic programming for the optimal management of virtual infrastructures in the cloud. We complement their work by exploring customer-centric resource allocation strategies under a pre-determined SLA to fulfill the contract and optimize backups provisioning decisions in IaaS.

4 Recurrent Interventions at Fixed Intervals (RIFI) Strategy

Consider an SLA where a client requests n VMs at an uptime requirement of $0 \leq \alpha \leq 1$. Without loss of generality, we assume a powered-on check-pointing scheme with k backup VMs. Any other form of check-pointing with k backup VMs will only differ in the way the downtime distribution is estimated, and will not affect the model in our current context. Prior to the start of the service, the provider determines the optimal number of backup VMs, k_0, using the algorithm in (Yuan et al., 2018). We refer to this k_0 as the initial backup VM allocation, which is based on the predicted service level derived from the downtime distribution over the entire contract duration. Since the total allowable downtime in the whole contract is modeled as a perishable commodity, the cumulative downtime experienced till a point in time represents the consumption of this commodity. At any time during the contract, the provider could adjust the initial allocation based on the actual level of service availability realized till then, as it may deviate from the expected overall service level which covers the whole contract period. Accordingly, the number of backup VMs could be increased or decreased depending on this realization. For instance, if the actual realized service level is higher than the expected level at a certain time, it may be more cost-effective to decrease the number of backup VMs; similarly, when the realized service level is less than the expected level, an increase is indicated. While the former case indicates the opportunity to lower the VM provisioning cost, the latter case provides the chance for lowering the potential penalty cost, both with respect to their corresponding expected costs determined at the beginning of the contract. Using this principle we develop the recurrent intervention strategy in the following discussion.

A penalty is incurred when the service provider fails to meet the uptime guarantee within a finite service window. Given the finite service window and the stochastic nature of failures and recoveries, steady state presumptions on the service level cannot be relied upon to hold by the end of the contract period. More specifically, if the realized cumulative service level is more than the expected level at any point of time, although this excess could buffer against incurred downtime that may be higher than that expected in the time remaining in the contract, it may not guarantee an overall realized service level that equals the guaranteed level. Similarly, if the realized level is less than the expected level at any point, it does not indicate that the service will catch up with this shortfall in the remaining period. Hence, interventions are useful for both the provider and the client from the points of view of minimizing the overall cost and at the same time, ensuring the delivery of a guaranteed level of service. This lead to two important resource management considerations at run time: *when to* intervene, and *how much* to correct in the backup allocations. The two considerations and the ensuing decisions would occur concurrently during the course of the contract. Although a continuous consideration and corresponding resource adjustments throughout the contract period would be ideal, it will not be a practical solution. Hence, we decouple the two considerations in this strategy as follows. First, we select a fixed number of interventions, preferably but not necessarily spaced equally in time during the contract period. Next, at each intervention time, the level of service availability provided so far is evaluated against the assured level, and the optimal decision on quantum of VM allocation is made. Note that the determination of the optimal number of back VMs to be allocated at an intervention requires the estimation of the service levels in the remaining part of the contract at different allocation levels concomitantly with the available amount of allowable downtime as per the assured level of service in the contract. This modeling approach is similar to the classical newsvendor problem, albeit with some essential differences. First, the uncertainty in the consumption of the perishable commodity (the allowable downtime) arises from the birth-death process of VM failures and recoveries; second, adjustments in resource deployment are carried out at multiple intervention times; and third, the effects of the adjustment can be realized only at a future point in time, unlike the classical inventory models where inventory levels are realizable upon order arrivals.

For the sake of brevity and without loss of generality, we assume that the provider faces a single type of failure, and for a given client, uses a 1:1 mapping of physical servers to VMs in the datacenters. This mapping is essentially used in the estimation of downtime

Table 2 Notations

α	Availability guarantee specified in the SLA, $0 \le \alpha \le 1$
T	Service window over which the availability has to be fulfilled (e.g, if SLA specifies 99% availability in each week, then the service window is one week)
S	Total number of interventions chosen by the provider
$q = \{0, 1, 2, \ldots S\}$	Intervention index; $q = 0$ indicates the starting point of the contract
$\boldsymbol{\alpha_q}$	Random Variable denoting availability at qth intervention, $0 \le \boldsymbol{\alpha_q} \le 1$
$E[\boldsymbol{\alpha_q}]$	Expected value of $\boldsymbol{\alpha_q}$ at qth intervention
$\hat{\alpha}_q$	Observed availability at the time of qth intervention
B_q	Allowable downtime in the time remaining in the contract determined at the time of qth intervention
n	Number of VMs the client demands
k_q	Number of backup VMs at qth intervention
p	Price for each VM demanded
π	Penalty per unit of SLA violation time
h	Provisioning cost per VM per unit of time
$\boldsymbol{\tau_t}$	Accumulated downtime over $[0, t]$ at time t; where $t \in [0, T]$
$v(\boldsymbol{\tau_t} \mid n, k)$	Probability density function of the total downtime within a service window t for an (n, k) VM configuration, derived by the algorithm in (Du et al., 2015)

Fig. 1 RIFI Strategy for Backup VM Provisioning

Input: $n, T, S, \alpha, v(\boldsymbol{\tau_t}|n,k),\ k \in [0,1,2,\ldots], h, \pi, \delta_{threshold}$
Output: $k_q,\ q = \{0,1,2,\ldots S\}$

Begin

$q = 0$;

$\Delta t = \frac{T}{S+1}$;

$B_0 = (1-\alpha)T$;

Solve $\min_k TC = hkT + \pi \int_{B_0}^{T} v(\boldsymbol{\tau_T}|n,k)(\boldsymbol{\tau_T} - B_0) d\boldsymbol{\tau_T}$;

Let k_0 denote the optimal solution;

While $q < S$

$q = q + 1$;

* under equally spaced interventions, the qth intervention occurs at time $q\Delta t$ *\

$E[\boldsymbol{\alpha_q}] = \int_0^{(S-q+2)\Delta t} \frac{((S-q+2)\Delta t - \boldsymbol{\tau}_{(S-q+2)\Delta t})}{(S-q+2)\Delta t} v(\boldsymbol{\tau}_{(S-q+2)\Delta t}|n, k_{q-1}) d\boldsymbol{\tau}_{(S-q+2)\Delta t}$;

At time $q\Delta t$, Observe actual availability achieved $\widehat{\alpha_q}$;

Compute $\delta_q = \left|\widehat{\alpha_q} - E[\boldsymbol{\alpha_q}]\right|$;

$B_q = B_{q-1} - (1 - \widehat{\alpha_q})\Delta t$;

If $\delta_q \ge \delta_{threshold}$, then solve

$\min_k TC = hk(S-q+1)\Delta t + \pi \int_{B_q}^{(S-q+1)\Delta t} v(\boldsymbol{\tau}_{(S-q+1)\Delta t}|n,k)(\boldsymbol{\tau}_{(S-q+1)\Delta t} - B_q) d\boldsymbol{\tau}_{(S-q+1)\Delta t}$;

Let k_q denote the optimal solution;

Else $k_q = k_{q-1}$;

End While

End

probability distribution developed in Du et al. (2015), and can easily be extended to any general mapping. This is also a practical strategy used by many cloud service providers who choose to spread out the VMs for a given client across multiple server racks to reduce the risk of SLA violation by avoiding single points of failure. Table 2 lists the notations used in the following analysis along with brief definitions.

For simplicity, we initially assume that interventions are frictionless and are carried out at fixed intervals in time. We present the recurrent intervention strategy that yields the time of intervention and the quantum of VM resource adjustment at equally spaced intervals in Fig. 1. In general, the service window T is divided into $S+1$ segments each of length $\Delta t = \frac{T}{S+1}$, where S denotes the number of intervention opportunities. Note that Δt is the time interval between adjacent interventions. At each intervention q, the provider observes the availability achieved so far and determines $\delta_q = \left|\hat{\alpha}_q - E[\boldsymbol{\alpha_q}]\right|$. An acceptable bound $\delta_{threshold} \ge 0$ is cho-

sen and is used at each intervention point in the RIFI provisioning strategy. If $\delta_q > \delta_{threshold}$, then the provider resolves the underlying minimization problem on the expected total cost TC over the remaining time in the contract and determines the optimal k_q. Alongside, B_q is updated based on the observed availability $\hat{\alpha_q}$ by $B_q = B_{q-1} - \left(1 - \hat{\alpha_q}\right)\Delta t$. On the other hand, if δ_q does not exceed $\delta_{threshold}$, the provider maintains status quo on the backup provisioning until the next review. The bound $\delta_{threshhold}$ can be parameterized by the service provider based on past experience. Clients using cloud services for more mission-critical tasks with lower tolerance for non-availability of services may prefer contracts articulating high penalty rates. Thus to hedge the risk of incurring a large penalty due to SLA violation, the provider is motivated to check the actual service level and adjust the backup resources more frequently, by setting a large value for S and/or a small value for $\delta_{threshold}$. Lemma 1, which is also quite intuitive, shows that as the number of interventions increases, the total cost would decrease. In the limiting case, this would reach the ideal continuous review model which however, will not be practical. For clients who are less risk-averse, it may be more reasonable to use relatively larger values for $\delta_{threshold}$ and smaller values for S. Under RIFI, the time of intervention depends on the number of intervention opportunities while the quantum of adjustment is determined by re-solving the minimization problem on the expected total cost for the remaining time based on the information available at the times of intervention.

Lemma 1 *The expected total cost is decreasing when the number of equally-spaced interventions is increasing.*

Proof We prove that the expected total cost TC is decreasing when the number of interventions in RIFI is increasing by three sequential steps: we first analyze the cases $S = 2 * 2^{i-1} - 1, \ \ i = 1, 2, 3\ldots$ where S represents the number of interventions in the period of $[0, T]$ in step 1, then in step 2 we justify the cases when $S = 3 * 2^{i-1} - 1, \ \ i = 1, 2, 3\ldots$ and generalize the results in the final step.

$S = \left\{0, 1, 3, \ldots 2 * 2^{i-1} - 1\right\}, \ \ i = 1, 2, 3, \ldots$
Step 1:

Case 1: $S = 0$ vs. $S = 1$ The expected total cost over $\left(\frac{T}{2}, T\right]$ for non-intervention $(S = 0)$ is: $TC_{S=0} = hk_0\frac{T}{2} + \pi\int_{B_0-(1-\hat{\alpha})\frac{T}{2}}^{\frac{T}{2}} v\left(\tau_{\frac{T}{2}}|n, k_0\right)\left(\tau_{\frac{T}{2}} - B_0 + (1-\hat{\alpha})\frac{T}{2}\right)d\tau_{\frac{T}{2}}$ where $B_0 = (1-\alpha)T$, $\hat{\alpha}$ is the observed level of service at time $\frac{T}{2}$, and k_0 is obtained by solving the cost minimization problem at the beginning of the service window. The expected total cost for $S = 1$ over $\left(\frac{T}{2}, T\right]$ is: $TC_{S=1} = \min_k\left(hk\frac{T}{2} + \pi\int_{B_0-(1-\hat{\alpha})\frac{T}{2}}^{\frac{T}{2}} v\left(\tau_{\frac{T}{2}}|n, k\right)\left(\tau_{\frac{T}{2}} - B_0 + (1-\hat{\alpha})\frac{T}{2}\right)d\tau_{\frac{T}{2}}\right)$. $TC_{S=1} \le TC_{S=0}$ over $\left(\frac{T}{2}, T\right]$ since $hk_0\frac{T}{2} + \pi\int_{B_0-(1-\hat{\alpha})\frac{T}{2}}^{\frac{T}{2}} v\left(\tau_{\frac{T}{2}}|n, k_0\right)\left(\tau_{\frac{T}{2}} - B_0 + (1-\hat{\alpha})\frac{T}{2}\right)d\tau_{\frac{T}{2}}$ is no better (less) than $\min_k\left(hk\frac{T}{2} + \pi\int_{B_0-(1-\hat{\alpha})\frac{T}{2}}^{\frac{T}{2}} v\left(\tau_{\frac{T}{2}}|n, k\right)\left(\tau_{\frac{T}{2}} - B_0 + (1-\hat{\alpha})\frac{T}{2}d\tau_{\frac{T}{2}}\right)\right)$.

Case 2: $S = 1$ vs. $S = 3$ If three interventions occur at $\frac{T}{4}$, $\frac{2T}{4}$, and $\frac{3T}{4}$ in $S = 3$, comparing to $S = 1$, the former one is better (less or equal expected total cost), since $TC_{S=1} \le TC_{S=0}$ over $\left[0, \frac{T}{2}\right]$ and $\left(\frac{T}{2}, T\right]$ as proved in case 1. Therefore, $TC_{S=3} \le TC_{S=1}$. Similarly, it is provable that $TC_{S=2*2^i-1} \le TC_{S=2*2^{i-1}-1}, i = 1, 2, 3, \ldots$

$S = 0, 2, 5, \ldots 3 * 2^{i-1} - 1, \ \ i = 1, 2, 3, \ldots$
Step 2:

Case 1: $S = 0$ vs. $S = 2$ As proved from Case 1 in *Step 1*, $TC_{S=1} \le TC_{S=0}$ over $\left[0, \frac{2T}{3}\right]$, therefore, $TC_{S=2} \le TC_{S=0}$.

Case 2: $S = 2$ vs. $S = 5$ If five interventions occur at $\frac{T}{6}$, $\frac{2T}{6}\ldots$ and $\frac{5T}{6}$in $S = 5$, comparing to $S = 2$ at $\frac{T}{3}$ and $\frac{2T}{3}$, the former one is better (less or equal expected total cost), since $TC_{S=1} \le TC_{S=0}$ over $\left[0, \frac{2T}{6}\right]$, $\left(\frac{2T}{6}, \frac{4T}{6}\right]$ and $]$, $\left(\frac{4T}{6}, T\right]$as proved in Case 1 of *Step 1*. Therefore, $TC_{S=5} \le TC_{S=2}$. Similarly, it is provable that $TC_{S=3*2^i-1} \le TC_{S=3*2^{i-1}-1}, i = 1, 2, 3, \ldots$

Step 3: Following the similar procedure, it is provable that$TC_{S=j*2^i-1} \le TC_{S=j*2^{i-1}-1}, i = 1, 2, 3, \ldots$for $j = 2,\ 3,\ 4, \ldots$

Therefore, in general, more number of interventions leads to less expected total cost as long as intervention is frictionless and the interventions are spaced equidistant in time. **QED.**

The above lemma also leads to the asymptotic behavior of total cost with increasing number of interventions. As the number of interventions increases, the total cost progressively decreases and asymptotically converges to the cost of the continuous review model. This result is summarized in the following lemma.

Lemma 2 *The continuous review model minimizes the total cost of the contract.*

The continuous review model yields a lower cost than any periodic review model. However, continuous review is not practical in data center operations. Along the same lines, although increasing the number of interventions in a periodic review model would lower the total cost, this could incur greater interruptions in the service. This could adversely affect both the data center resource management operations and the continuity of service required by the client's applications running on these platforms.

Therefore, data centers would tend to keep the number of interventions at a minimum and evaluate the cost-benefits of increasing the number of interventions if necessary. Ideally, if the service proceeds more or less the same as planned and the downtime follows the estimated distribution fairly closely, then either no intervention or at most one intervention may be necessary. VM infrastructures with these attributes can be considered more reliable and fault-tolerant than those that exhibit significant deviations from the projected behaviors. Therefore, for reliable VM infrastructures, when a single intervention is contemplated, the when and how much decisions can be concomitantly evaluated and optimized. This approach is developed in the following section. Furthermore, when using the RIFI strategy, using interventions that are equally spaced in time over the contract interval may not yield a cost-minimizing solution for the same number of interventions. This implies that when a fixed number of interventions S in a time window T are considered, the times of these interventions need not be equally spaced in the optimal solution. This is shown in the following lemma.

Lemma 3 *For a given number of interventions, equally-spaced intervention times may not guarantee a minimum cost solution.*

Proof We consider two cases when $S=1$ as follows where t denotes the time of intervention.

Case 1 Let k_0 be the number of backup VMs obtained at the beginning of the service window $[0,T]$. Let choice (a) represents $t=\frac{T}{2}$, whereas choice (b) represents $t=t^*<\frac{T}{2}$.

(1) $\delta_1>\delta_{threshold}$ during $[0,t^*]$: For choice (b), the deviation of availability level is observed at t^*, therefore, $k_1>k_0$ on $(t^*,T]$. For choice (a), this deviation can be observed at $\frac{T}{2}$, thus, $k_1>k_0$ on $\left(\frac{T}{2},T\right]$. The possibility of incurring further downtime of (a) is higher than choice (b) over the interval of $\left(t^*,\frac{T}{2}\right)$, since k_0 has been updated to k_1 and $k_1>k_0$ in (b). Therefore, (b) is a better choice than (a) from the perspective of cost saving.

(2) $\delta_1>\delta_{threshold}$ during $\left(t^*,\frac{T}{2}\right)$: For choice (b), the deviation of availability level cannot be observed at t^*, therefore, $k_1=k_0$ on $(t^*,T]$. For choice (a), this deviation can be observed at $\frac{T}{2}$, thus, $k_1>k_0$ on $\left(\frac{T}{2},T\right]$. Therefore, (a) is a better choice.

Case 2 Let choice (a) represents $t=\frac{T}{2}$, whereas choice (b) represents $t=t^*>\frac{T}{2}$.

(1) $\delta_1>\delta_{threshold}$ during $\left[0,\frac{T}{2}\right)$: For choice (a), the deviation of availability level can be observed at $\frac{T}{2}$, thus, $k_1>k_0$ on $\left(\frac{T}{2},T\right]$. For choice (b), this deviation is observed at t^*, therefore, $k_1>k_0$ on $(t^*,T]$. The possibility of incurring further downtime of (b) is higher than choice (a) over $\left(\frac{T}{2},t^*\right)$, since k_0 has been updated to $k_1$and $k_1>k_0$ in (a). Therefore, (a) is a better choice than (b).

(2) $\delta_1>\delta_{threshold}$ during $\left(\frac{T}{2},t^*\right)$: For choice (a), the deviation of availability level cannot be observed at $\frac{T}{2}$, therefore, $k_1=k_0$ in $\left(\frac{T}{2},T\right]$. For choice (b), this deviation can be observed at t^*, thus, $k_1>k_0$ in $(t^*,T]$. Therefore, (b) is a better choice than (a). **QED.**

Intuitively, since the real failure and repair events may result in some deviation from the predicted level of service at runtime, the influence of intervention on the downtime distribution depends not only on when and by how much this deviation occurs, but also on whether or not such deviation is observed at the point of intervention. Therefore, as demonstrated in Lemma 3, when considering only a single intervention, an equally-spaced interval intervention strategy may not always guarantee an optimal solution from the purpose of cost minimization. This result, along with the considerations of practical intervention strategy in more reliable VM infrastructures discussed above, together lead to the development of optimal single intervention strategy in the following section.

5 Single Intervention at Random Interval (SIRI) Strategy

The RIFI strategy allows the provider to intervene and adjust the number of backup VMs depending on the difference between the actual realized service level and the expected service level at a time of intervention. The time of intervention is governed by Δt, the decision on whether to change the backup level or not by $\delta_{threshold}$, and if a change is required, then the quantum of intervention is determined by re-solving the underlying resource optimization problem. Clients using cloud services for more mission-critical tasks may seek non-interrupted service and emphasize service continuity and stability. In such cases, intervening too frequently as in RIFI (when $\delta_{threshold}$ is small) is not advisable due to potential service disruptions as well as the added cost of operationalizing frequent interventions. The greater control over resources under frequent interventions comes at a cost because all processes in a

Fig. 2 Cost Minimization Algorithm under SIRI

Input: $n, T, \Delta, \alpha, v(\boldsymbol{\tau_t}|n,k), k \in [0,1,2,\ldots], h, \pi, \beta$
Output: $k_q, q = \{0,1\}$

Begin
$t = 0$;
Solve $\min_k TC = hkT + \pi \int_{(1-\alpha)T}^{T} v(\boldsymbol{\tau_T}|n,k)(\boldsymbol{\tau_T} - (1-\alpha)T)d\boldsymbol{\tau_T}$;
Let k_0 denote the optimal solution;
$E[\boldsymbol{\tau_T}] = \int_0^T \boldsymbol{\tau_T} v(\boldsymbol{\tau_T}|n,k_0)d\boldsymbol{\tau_T}$;
Repeat
$t = t + \Delta$;
At time t, Observe actual incurred downtime $\hat{\tau_t}$;
If $\hat{\tau_t} \geq \beta E[\boldsymbol{\tau_T}]$, then solve
$\min_k TC = hk(T-t) + \pi \int_{(1-\alpha)T-\hat{\tau_t}}^{T-t} v(\boldsymbol{\tau_{T-t}}|n,k)(\boldsymbol{\tau_{T-t}} - (1-\alpha)T + \hat{\tau_t})d\boldsymbol{\tau_{T-t}}$;
Let k_1 denote the optimal solution;
End
Until $t = T$
End

running application may need to be temporarily paused during the intervention process in order to maintain synchronicity across primary and backup images.

As a less resource-intensive and less disruptive alternative to the RIFI strategy, and also motivated by Lemma 3 above, we now focus on the planned limited intervention strategy, starting with a single intervention case where the provider chooses to adjust the backup provisioning in a contract period at most once. The central question here lies in determining the time to intervene. If a maximum of only one intervention is practically feasible, it is worth noting that if the intervention is scheduled too early, then a larger time frame is left open in the contract period with no recourse to interventions. Consequently, the risk of significant service level degradation in the remaining contract period could increase, resulting in a potential increase in the penalty for violating the assured service level. On the other hand, if the intervention is scheduled too late in the service window, the time left may be insufficient to catch up with the assured service level. Using these ideas, we develop two approaches under the SIRI strategy as follows.

5.1 Cost Minimization Policy

This policy principally focuses on the expected downtime denoted by $E[\boldsymbol{\tau_T}] = \int_0^T \boldsymbol{\tau_T} v(\boldsymbol{\tau_T}|n,k_0)d\boldsymbol{\tau_T}$. Note that in certain contracts, since the total cost is an aggregate of provisioning cost and penalty cost, it may be optimal for the service provider to not fulfil the uptime guarantee, and as a result, pay a penalty to the client in order to minimize the total cost. In such scenarios, we observe that $E[\boldsymbol{\tau_T}] \geq (1-\alpha)T$. Specifically, this policy is well-suited for clients with less critical usage patterns, or who are less risk-averse, or those who are more price-sensitive. Such clients would primarily seek lower prices for the services from the provider rather than expecting penalty compensations for service level violations. Therefore, we present a cost minimization approach to manage the re-provisioning of backup resources, where the quantum of intervention is determined by deriving the optimal number of backup VMs such that the expected total cost which aggregates both provisioning cost and expected penalty cost is minimized for the remaining contract period. In this policy, starting from the beginning of the service window, the provider monitors the service levels attained thus far at regular intervals. Let Δrepresent a fixed interval of time between any two successive monitoring events. The cost minimization algorithm is presented in Fig. 2.

5.2 Downtime Minimization Policy

Typical cloud clients use the IaaS cloud to deliver a variety of end-user functionalities, from data collection and analysis to running user authentication services to managing configurations on a multitude of end user devices. These functionalities may vary in their mission-criticality. The clients may also vary in their risk tolerance, particularly pertaining to the risk of service non-availability. The downtime minimization policy will be appropriate when mission-critical applications are involved and the clients have a low tolerance for the risks arising out of extreme cost minimization, and are less price-sensitive than the cost-minimizing clients.

The downtime minimization policy also follows the same principle as the above cost minimization algorithm in determining the time of intervention. Using the regular monitoring strategy, the time of intervention is determined when the accumulated downtime reaches the threshold value: $\beta E[\boldsymbol{\tau_T}]$. At the time of intervention, this policy aims to minimize the expected downtime in the remaining portion of the contract period. Consequently, this approach also minimizes the likelihood of violating the availability assured in the SLA. This approach can be considered as an aggressive strategy for contracts with high availability requirements. The algorithm is presented in Fig. 3. We computationally explore the relationship between downtime and cost minimization policies in Section 6.

Fig. 3 Downtime Minimization Algorithm under SIRI

Input: $n, T, \Delta, \alpha, v(\boldsymbol{\tau_t}|n,k), k \in [0,1,2,\dots], h, \pi, \beta$
Output: k_q, $q = \{0,1\}$

Begin
$t = 0$;
Solve $\min_k TC = hkT + \pi \int_{(1-\alpha)T}^{T} v(\boldsymbol{\tau_T}|n,k)(\boldsymbol{\tau_T} - (1-\alpha)T)d\boldsymbol{\tau_T}$;
Let k_0 denote the optimal solution;
$E[\boldsymbol{\tau_T}] = \int_0^T \boldsymbol{\tau_T} v(\boldsymbol{\tau_T}|n,k_0)d\boldsymbol{\tau_T}$;
Repeat
$t = t + \Delta$;
If $\widehat{\tau_t} \geq \beta E[\boldsymbol{\tau_T}]$, then solve
$min\ k$
s.t. $\int_0^{T-t} v(\boldsymbol{\tau_{T-t}}|n,k)\boldsymbol{\tau_{T-t}}d\boldsymbol{\tau_{T-t}} = 0$
Let k_1 denote the optimal solution;
End
Until $t = T$
End

5.3 Generalized Multiple Interventions at Random Intervals (MIRI) Framework

Both the cost-minimization and downtime minimization algorithms in the SIRI policy can be generalized to incorporate multiple interventions carried out at random intervals in a recursive manner during the contract period. The strategy underlying this generalization is as follows. First, as in the SIRI algorithms, the provider follows a regular monitoring process. When a time of intervention is determined using the observed downtime and the risk-adjusted expected threshold downtime, the appropriate modification to the backup allocation is carried out as per the cost-minimization or downtime-minimization criteria used by the provider. Next, the monitoring process is continued throughout the contract period, and using the same criterion for intervention, the next intervention time is determined. Following this, solving the underlying optimization problem, a revised optimal backup allocation is determined. This process is repeated recursively until the end of the service contract period. The multiple interventions in this framework are denoted as $q = \{0, 1, 2, \dots\}$ in the contract period. Note that the intervention times are chosen as per the intervention criterion, and hence are random and are not predetermined.

Fig. 4 Generalized MIRI Framework

Input: $n, T, \Delta, \alpha, v(\boldsymbol{\tau_t}, n, k), k \in [0,1,2,\dots], h, \pi, \beta$
Output: k_q, $q = \{0, 1, 2, \dots\}$

Begin
* Let D_q denote the remaining downtime from the initial expected downtime over $[0, T]$ at the time of the qth intervention. *\
$t = 0$; $q = 0$;
Solve $\min_k TC = hkT + \pi \int_{(1-\alpha)T}^{T} v(\boldsymbol{\tau_T}|n,k)(\boldsymbol{\tau_T} - (1-\alpha)T)d\boldsymbol{\tau_T}$;
Let k_0 denote the optimal solution;
$E[\boldsymbol{\tau_T}] = \int_0^T \boldsymbol{\tau_T} v(\boldsymbol{\tau_T}|n,k_0)d\boldsymbol{\tau_T}$;
$D_0 = E[\boldsymbol{\tau_T}]$;
Repeat
$t = t + \Delta$;
If $\widehat{\tau_t} \geq \beta D_q$, then
$q = q + 1$;
If the Total Cost TC is to be minimized, then solve
$\min_k\ TC = hk(T-t) + \pi \int_{(1-\alpha)T-\widehat{\tau_t}}^{T-t} v(\boldsymbol{\tau_{T-t}}|n,k)(\boldsymbol{\tau_{T-t}} - (1-\alpha)T + \widehat{\tau_t})d\boldsymbol{\tau_{T-t}}$;
Let k_q denote the optimal solution;
Elseif Downtime is to be minimized, then solve
$min\ k$
s.t. $\int_0^{T-t} v(\boldsymbol{\tau_{T-t}}|n,k)\boldsymbol{\tau_{T-t}}d\boldsymbol{\tau_{T-t}} = 0$
Let k_q denote the optimal solution;
$D_q = D_{q-1} - \widehat{\tau_t}$;
Until $t = T$
End

Hence, the number of interventions is also not pre-determined in this generalized framework. In this context, note that the MIRI framework is also a generalization of the RIFI strategy which uses a pre-determined number of interventions at fixed intervals of time. The generalized MIRI framework is presented in Fig. 4.

6 Experimental Results

In this section, we explore the impacts of both RIFI and SIRI strategies on the performance of a contract through comparisons with no intervention solution. We also evaluate some parameters in our models, such as the intervention time and the penalty levels desired by clients, and their influences on our backup resource reprovision policies.

6.1 Impact of $\delta_{threshold}$ on Contract Performance in RIFI Strategy

We first demonstrate the influence of the threshold value $\delta_{threshold}$ under RIFI, since the time of intervention in RIFI depends on the run-time deviation from the expected level of service, which is captured by δ_q. The provider re-solves the cost minimization problem for the remaining contract window if an only if the difference exceeds $\delta_{threshold}$. For clients who use cloud services for more mission-critical tasks with lower tolerance for non-availability risk, the provider will be motivated to set up a lower value for $\delta_{threshold}$, thus reducing the probability of incurring huge penalty. This experiment is run for $n=50$, $\Delta=5$ minutes, and $T=30$ days, with the initial optimal backup allocation k_0 solved from (Yuan et al., 2018). We consider this static, no intervention optimal solution as a benchmark. We then explored the impacts of intervention in RIFI by varying $\delta_{threshold}=0.1\,\%$, $0.5\,\%$, $1.0\,\%$, $1.5\,\%$, 2% respectively, such that the provider updates and adjusts the backup resources at time $T/2$ but with various incurred downtime ($\delta_1=\delta_{threshold}$). We also change the ratio between provisioning cost and penalty rate, $h:\pi=1:100/1000/5000$ in each setting of $\delta_{threshold}$, since the intervention policy also depends on the penalty rate requested by the client. We define *expected penalizable downtime* as the amount of downtime accumulated within the contract in excess of the downtime allowable under the SLA-specified uptime guarantee.

Table 3 shows the performance of RIFI under different configurations. First, RIFI models in all settings have significantly reduced both the expected penalizable downtime and the expected total cost due to the ability to adjust the backup provision over the contract duration, in contrast to no intervention benchmark. Second, all else being equal, as $\delta_{threshold}$ increases, the expected total cost increases even if intervention is scheduled. This is not

Table 3 Impact of $\delta_{threshold}$ on Contract Performance in RIFI

$h:\pi=1:100$, $k_0=10$				$h:\pi=1:1000$, $k_0=12$				$h:\pi=1:5000$, $k_0=14$			
$\delta_{threshold}$		*Expected Penalizable Downtime*	*Expected TC*	$\delta_{threshold}$		*Expected Penalizable Downtime*	*Expected TC*	$\delta_{threshold}$		*Expected Penalizable Downtime*	*Expected TC*
0.1%	*w/o RIFI*	170.49	103449	*0.1%*	*w/o RIFI*	49.54	153223	*0.1%*	*w/o RIFI*	22.64	234179
	RIFI	75.24	98244		*RIFI*	1.48	118116		*RIFI*	0.45	136157
0.5%	*w/o RIFI*	187.77	105177	*0.5%*	*w/o RIFI*	60.86	164537	*0.5%*	*w/o RIFI*	30.8	274979
	RIFI	92.52	99972		*RIFI*	2.14	118779		*RIFI*	0.7	137416
1.0%	*w/o RIFI*	209.37	107337	*1.0%*	*w/o RIFI*	80.02	183702	*1.0%*	*w/o RIFI*	47.08	356345
	RIFI	114.12	102132		*RIFI*	8.07	124711		*RIFI*	0.93	142887
1.5%	*w/o RIFI*	230.97	109497	*1.5%*	*w/o RIFI*	101.62	205302	*1.5%*	*w/o RIFI*	68.68	464345
	RIFI	135.72	104292		*RIFI*	29.67	146311		*RIFI*	22.53	250887
2.0%	*w/o RIFI*	252.57	111657	*2.0%*	*w/o RIFI*	123.22	226902	*2.0%*	*w/o RIFI*	90.28	572345
	RIFI	*157.32*	106452		*RIFI*	51.27	167911		*RIFI*	44.13	358887

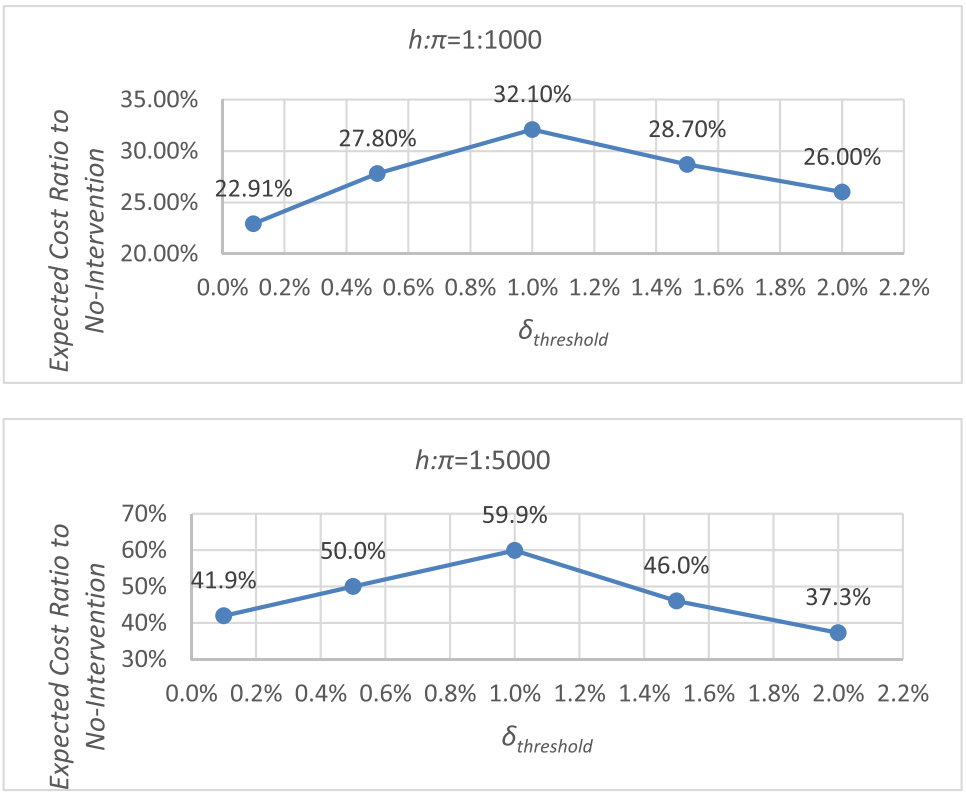

Fig. 5 Expected Cost Performance Ratios of RIFI Models

Table 4 Impact of Intervention Time on SIRI

		Downtime Minimization Policy			Cost Minimization Policy		
	Time of Intervention	k_1	*Expected Penalizable Downtime*	*Expected TC*	k_1	*Expected Penalizable Downtime*	*Expected TC*
$n = 50\ k_0 = 10$	$t = 0.25\ T$	22	90.60	173220	11	177.72	110652
	$t = 0.50\ T$	22	90.60	147300	11	159.56	106676
	$t = 0.75\ T$	21	90.60	119220	12	110.76	101796
$n = 100\ k_0 = 17$	$t = 0.25\ T$	34	90.86	266126	18	196.26	172986
	$t = 0.50\ T$	33	90.86	225086	18	174.78	168678
	$t = 0.75\ T$	33	90.86	190526	19	120.91	163291

surprising because higher $\delta_{threshold}$ implies more incurred downtime before the intervention, which directly leads to higher expected penalty cost. This highlights the necessity of backup resource reprovision policy for high-availability clients associated with potentially high penalty rates.

We also define the expected total cost ratio to benchmark as $\frac{TC[No\ Intervention]-TC[RIFI]}{TC[No\ Intervention]}$, to measure the relative cost reduction of RIFI models from the non-intervention benchmark. An interesting observation is that we see inverse U-shapes curves and they peak at $\delta_{threshold}=1\%$ in the settings of $h:\pi=1:1000$ and $h:\pi=1:5000$ in Fig. 5. Intuitively, this is because beyond 1%, the deviation from the expected service level is large enough, i.e., $(\boldsymbol{\tau_t} \geq E[\boldsymbol{\tau_T}])$, such that any downtime incurred after the intervention would lead to a larger than anticipated penalty payment. Thus this finding also empirically validates the rationale for the threshold value in SIRI: $E[\boldsymbol{\tau_T}]$, as the contract manager has limited chances to reprovision.

6.2 Impact of Intervention Time on the Backups Reprovisioning in SIRI

As the downtime distribution after intervention is also a function of intervention time, we now evaluate the impact of intervention time on both cost minimization and downtime minimization policies under SIRI strategies where the provider has only one opportunity to adjust his/her backup provisioning in a contract. Because the major challenge lies

Table 5 Impact of Penalty on Cost Minimization Policy

Penalty Level	*Time of Intervention*	k_1	*Expected Penalizable Downtime*	*Expected TC*
$h{:}\pi = 1{:}100,\ k_0 = 10$	$t = 0.25\ T$	11	177.72	110652
	$t = 0.50\ T$	11	159.56	106676
	$t = 0.75\ T$	12	110.76	101796
$h{:}\pi = 1{:}1000,\ k_0 = 12$	$t = 0.25\ T$	15	13.07	136187
	$t = 0.50\ T$	15	12.72	129359
	$t = 0.75\ T$	15	12.47	122626
$h{:}\pi = 1{:}5000,\ k_0 = 14$	$t = 0.25\ T$	17	2.35	152138
	$t = 0.50\ T$	18	1.29	144708
	$t = 0.75\ T$	18	1.32	136211

in determining both the quantum and time of the adjustment, we set the time of intervention at $0.25T$, $0.50T$ and $0.75T$, with $T = 30$ days and $n = 50$ and 100. We find in our experiments on the downtime minimization approach in Table 4 that the updated k (k_1) is non-increasing as the remaining contract duration shrinks. This is because the less time remains, less downtime will possibly incur, and therefore fewer additional backups will be required for the remainder of the contract to reach the point where the expected penalizable downtime drops to zero.

For the cost minimization policy, however, k_1 keeps increasing as the provider chooses to intervene later as Table 4 illustrates. Typically providing more backup VMs reduces the likelihood of SLA violation but incurs higher provisioning cost. This provisioning cost is limited to a function of the remaining time of the service window, especially when $t = 0.75T$, thus incentivizing the provider to add more backup resources. Meanwhile, the k_1 for the cost minimization approach is much lower than the downtime minimization alternative since given a failure and repair time distribution, a larger k is required such that the expected penalizable downtime reduces until it becomes close to zero as the provider is more conservative regarding the risk of SLA violation. Note that this is also because the cost minimization policy not only depends on the time of intervention, but is also contingent on the penalty rate driven by the client. We conduct further computational experiments specifically regarding the ratio between the penalty rate and provisioning cost in the next section.

6.3 Impact of Penalty on Cost Minimization Strategy

The penalty rate for non-availability in cloud SLAs would largely be driven by the mission-criticality of the tasks that a client assigns to the datacenter. A client running highly mission-critical jobs may insist on high penalty rates to hedge against loss of revenue and reputation from non-availability of services to its end-users. The cloud provider in turn reacts to the high penalty rate by adjusting backup resources accordingly during the intervention, especially for the cost-minimization strategy, which raises the following question: how does the penalty level requested by a client affect the backup reprovision decision? We therefore explore how updated k, the number of backup VMs for the remaining contract, changes with changing ratio between the penalty and the provisioning cost, h. We set h to 1 and derived the updated k,for increasing penalty rates, from 1:100, 1:1000 to 1:5000, for $n = 50$ and $T = 30$ days.

As Table 5 illustrates, given the time of intervention, when the ratio is increased from 1:100 to 1:1000 and 1:5000, k_1 increases dramatically in all scenarios. At that point, the penalty rate is large enough to induce the provider to reallocate significantly more backup VMs, such that the SLA violation probability is as close to zero as possible, thus driving the solution for a cost minimization problem closer to a downtime minimization problem, with regard to a very small amount of expected downtime in the remaining contract. In addition, given a penalty level, k_1 becomes non-decreasing and both expected penalizable downtime and expected *TC* decrease, as the time of intervention is closer to the end of such contract. This is because as the intervention is triggered later during a contract window, higher performance is achieved on the underlying infrastructure, which results in lower operating costs and penalty payment due to SLA violation. Whereas if the single intervention is scheduled early, the provider faces potential higher costs associated with more potential downtime. This experiment highlights how the penalty rate, which is largely client-driven, affects the provider's decisions regarding backup resources reprovisioning, given the provisioning cost.

7 Model Validation with Amazon EC2 Service Structure

In this section, we validate our models based on actual pricing and service credit data on dedicated hosts obtained from the Amazon EC2 website. Consider the case of a client requesting to contract with Amazon for 100 instances (VMs) for simplicity. A dedicated host is configured to support one VM at a time. The contract can have different configurations based on Amazon instance types and its pricing/

Table 6 Policies Comparisons under AWS Structure

			No Intervention		Cost Minimization Policy		Downtime Minimization Policy	
p	$h/\overline{p}$	π/h	k_1	*Expected TC*	k_1	*Expected TC*	k_1	*Expected TC*
206.59 *(Fault-Prone)*	0.1	152.64	11	289.87	12	276.64	23	375.34
	0.3	50.88	9	769.09	10	747.10	21	1045.47
	0.5	30.53	9	1142.34	10	1141.08	21	1667.55
13415.94 *(Fault-Tolerant)*	0.1	244.78	4	10794.34	5	7962.28	10	10606.53
	0.3	81.59	4	21525.22	5	20034.52	10	29385.57
	0.5	48.96	3	34256.63	4	31167.53	10	49699.17

penalty structures. For illustration purposes, we choose the 1-month contract for the cheapest instance type $a1$ and a similar 1-month contract for the most expensive instance type $p3$. These instance type designations are from the Amazon EC2 website. The monthly price p for one $a1$ VM is \$206.59 and for one $p3$ VM is \$13,415.94. Since service credits for violations of uptime guarantees are offered as fractions of the prices charged, it is realistic to consider the low-cost $a1$ hosts to be less fault-tolerant (or equivalently, more fault-prone) than the high-cost $p3$ hosts. Accordingly, we term the two instance types $a1$ and $p3$ considered in this study as fault-prone and fault-tolerant instances, respectively. As we do not have access to the mean time between failures (MTBF) and mean time to repair (MTTR) data from Amazon, we obtained these parameters from the server logs provided by the Center for Computational Research (CCR) at the University at Buffalo, which is a high-performance computing node. Using these parameters as surrogates for the Amazon data center operations, we conducted a detailed computational study of the proposed algorithms using their price and penalty structures for the $a1$ and $p3$ instance types. These results can be easily replicated if the server logs data from Amazon are available.

We set $\Delta = 5$ minutes such that there are 8640 number of discrete time intervals in the one month evaluation period, thus the selling price per VM per unit of time becomes $\overline{p} = p/8640$. We also assume the resource provisioning cost per VM per unit time is a percentage of the selling price per VM per unit time: $h = \{10\%, 30\%, 50\%\} * \overline{p}$, which are three levels of the provisioning cost. Second we estimate the penalty payment per unit time π based on the AWS price-penalty structure: $\pi = 0.3650$ for $a1$ and $\pi = 38.01$ for $p3$ respectively. This is also consistent with our investigation that a client expecting a higher penalty payment in the event of SLA violation may be charged a relatively higher price, as the provider uses more backup resources to mitigate the penalty risk. Note that to make a fair comparison across the various intervention models, we also assume that the provider adjusts backup resources at $t = 0.5\ T$ under all policies. Therefore, RIFI strategy with $S = 1$ becomes equivalent to the Cost Minimization policy in SIRI regarding the reprovision decision. We compare our two models under SIRI strategy with the static benchmark. Table 6 presents amount of backup resource adjustment and expected total cost under different treatment conditions.

Similar to the insights gained from the prior experiments, we find that implementing the cost minimization policy yields lower expected total cost than the no intervention model; in addition, fault-tolerant systems achieve higher cost saving than fault-prone systems. This is not surprising because the provider is capable of offsetting the risk of higher expected penalty costs in fault-tolerant VMs through additional backup provisioning. Meanwhile, it justifies our discussion in Section 5.2 that, the downtime minimization approach requires more additional backups, in contrast to cost minimization policy, and should be considered as an aggressive strategy for contracts with high availability requirements where cost saving is not the sole purpose. This is because it may not necessarily be cost-effective to achieve higher service availability level given a price-penalty schedule and underlying infrastructure.

We also see in general, fault-prone systems require more backup resources than fault-tolerant systems. Intuitively, this is because AWS defines a common SLA with guaranteed 99.99% service availability for all the consumers purchased EC2 services. Other things being equal, with smaller ratio of MTBF and MTTR, additional VMs are inevitably needed to achieve this uptime guarantee for fault-prone systems. In other words, it is more beneficial to deploy high availability requirement services on fault-tolerant infrastructure. Furthermore, interactions between various SLA constructs are visualized through these treatments. Clients using cloud services for more mission-critical tasks or possessing a low tolerance for risk may favor fault-tolerant systems with higher penalty levels as a hedge against the risk of non-availability. In turn, they may need to be charged more since the provider may have to provision less fault-prone infrastructure to increase resiliency, leading to higher

provisioning and operating cost and potential higher penalty cost as failures occur.

8 Discussion

Since the real failure and repair events during a service window may result in some deviation from the predicted level of service at runtime, we first provide a periodic intervention strategy RIFI to check and adjust the backup resources dynamically in order to minimize the impact of random runtime failure and repair events. We also propose two single intervention policies under SIRI strategy with different purposes to determine the time and quantum of resource reallocation when frequent adjustments are costly, from the perspective of an IaaS provider. We also conduct extensive computational studies to supplement the analytical work. We first show the impact of the threshold value $\delta_{threshold}$ under RIFI on the contract performance. Next, we explore the influence of intervention time, on the backup VMs reprovisioning, for both non-downtime and cost-minimization policies in SIRI. Furthermore, we highlight how the penalty rate, which is largely client-driven and a crucial component specified in SLA, affects the provider's decisions regarding backup resource reallocation, especially for the cost-minimization policy. Finally, we conduct computational experiments and validate our model performance based on use cases constructed from Amazon EC2 price-penalty schemas.

8.1 Implications for Practice

The following key managerial implications emerge from this study. First, the provider needs to be able to differentiate availability of the cloud services, given the heterogeneity amongst the clients, based on the end-use of their offerings and ensuing risk implications since backup resource reallocation strategies also depend on the client type. For instance, for clients who are either price-sensitive or who run less critical services on the cloud, the provider would be inclined to adjust backup VMs less frequently by setting a smaller value for S and/or a larger threshold value $\delta_{threshold}$ under RIFI, and adopting cost-minimization approach for better cost-effective purpose. On the other hand, the provider would be encouraged to reprovisioning backup resources more frequently and adapt non-downtime strategy for those clients who emphasize high availability, and thus favor higher penalty levels as a hedge against the risk of non-availability. Second, for those services defined by "plain vanilla" posted SLA framework (e.g., 99.99% in AWS), differentiated contacts should also be explored since both initial backup provision and run-time intervention polices are influenced by key decision making criteria when comparing data center infrastructure systems, such as MTBF and MTTR.

Finally, it is crucial to obtain a better understanding of the provisioning cost for effective resource provisioning, such as electricity, network bandwidth, cooling, labor, operations, software, and hardware. As we demonstrated in our experiments, the ratio between provisioning cost and penalty rate has a direct impact on the adjustment of backup resources. E.g., when the penalty rates are significantly higher than the provisioning cost, the provider will update considerably more backup resources such that the SLA violation probability is reduced as close to zero as possible. At this point, the quantum of adjustment during a cost-minimization intervention converges to the result from a non-downtime strategy, in order to decrease the expected downtime for the rest of the service window as much as possible.

8.2 Implications for Research

Our results show that the expected total cost decreases as the number of interventions increase. Therefore, future studies should consider how to derive a set of policies that make it easier for cloud datacenters to apply different intervention frequencies for different customer types. AWS, for instance, caters to a wide range of customers from online travel agencies to credit card companies to cryptocurrency trading platforms. The business operations of each of these three types are radically different, with resulting implications on the demand and the supply side of resources; optimal policies for one may prove to be detrimental for another. It is thus important to study the derivation of policies to tailor them to these starkly different needs.

Our research also finds that equally spaced interventions may not be as effective as the MIRI framework. Therefore, a key takeaway for researchers would be that future resource allocation studies ought to incorporate, even in small ways, dynamic responsiveness to real-world unfolding of events, despite the mathematical tractability of more regularized policies.

9 Limitations and Future Research

This study leads to some important and practical directions for future research. First, we assume frictionless interventions on RIFI strategy, we do not explicitly model the cost of intervention in MIRI framework. Future research may focus on more complex intervention questions in IaaS cloud infrastructure, e.g., taking the monitoring and intervention overheads into account on reprovisioing and adjusting backup resources in the runtime environment, given risk preferences of the clients. How should the provider schedule the number

of intervention opportunities, based upon the client type? Second, we also assume independent VM failure events in the downtime distribution estimation of our models. Recently, traditional dedicated network hardware appliances, such as routers, firewalls and load balancers, are replaced by virtualized software implementation in Network Function Virtualization (NFV) architecture. These modular software components of a network function are called virtualized network functions (VNFs) and deployed over VMs. Although each failure on a VM is independent, the VNF failures may be correlated because of the hierarchical network structures. How to extend our virtual resource provisioning strategies to an NFV context is another avenue for future investigation. The cloud service providers and practitioners would benefit from this research line to effectively control and manage risks on availability commitment in an SLA by dynamically allocating backup resources in the cloud.

Declarations

Conflict of Interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Bruneo, D. (2014). A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems. *IEEE Transactions on Parallel and Distributed Systems, 25*(3), 560–569.

Chase, J., & Niyato, D. (2015). Joint optimization of resource provisioning in cloud computing. *IEEE Transactions on Services Computing, 10*(3), 396–409.

Cisco. (2012). *Cisco Global Cloud Networking Survey*.

Dean, J. (2009) *Designs, lessons and advice from building large distributed systems*. https://www.cs.cornell.edu/projects/ladis2009/talks/deankeynote-ladis2009.pdf

Du, A. Y., Das, S., Yang, Z., Qiao, C., & Ramesh, R. (2015). Predicting transient downtime in virtual server systems: An efficient sample path randomization approach. *IEEE Transactions on Computers, 64*(12), 3541–3554.

Ghosh, R., Longo, F., Frattini, F., Russo, S., & Trivedi, K. S. (2014a). Scalable analytics for IaaS cloud availability. *IEEE Transactions on Cloud Computing, 2*(1), 57–70.

Ghosh, R., Longo, F., Xia, R., Naik, V. K., & Trivedi, K. S. (2014b). Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. *IEEE Transactions on Services Computing, 7*(4), 667–680.

Goudarzi, H., Ghasemazar, M., & Pedram, M. (2012). *SLA-based optimization of power and migration cost in cloud computing* 2012 12th IEEE/ACM international symposium on cluster, Cloud and Grid Computing, Ottawa, ON, Canada.

Guo, Z., Li, J., & Ramesh, R. (2019). Optimal Management of Virtual Infrastructures under flexible cloud service agreements. *Information Systems Research, 30*(4), 1424–1446.

Gutierrez-Garcia, J. O., & Sim, K. M. (2012). GA-based cloud resource estimation for agent-based execution of bag-of-tasks applications. *Information Systems Frontiers, 14*(4), 925–951.

Hassan, M. M., Hossain, M. S., Sarkar, A., & Huh, E.-N. (2014). Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform. *Information Systems Frontiers, 16*(4), 523–542.

ITIC (2017) ITIC 2017–2018 global server hardware, server OS reliability report. https://www.ibm.com/downloads/cas/DV0XZV6R#:~:text=ITIC's%202020%20Reliability%20poll%20finds,mission%20critical%20servers%20and%20applications

Jammal, M., Kanso, A., Heidari, P., & Shami, A. (2016). *A formal model for the availability analysis of cloud deployed multi-tiered applications* 2016 IEEE international conference on cloud engineering workshop Berlin, Germany.

Jammal, M., Hawilo, H., Kanso, A., & Shami, A. (2018). ACE: Availability-aware CloudSim extension. *IEEE Transactions on Network and Service Management, 15*(4), 1586–1599.

Kauffman, R. J., Ma, D., Shang, R., Huang, J., & Yang, Y. (2015). On the Financification of cloud computing: An agenda for pricing and service delivery mechanism design research. *International Journal of Cloud Computing.*

Mansouri, Y., Toosi, A. N., & Buyya, R. (2019). Cost optimization for dynamic replication and migration of data in cloud data centers. *IEEE Transactions on Cloud Computing, 7*(3), 705–718.

Marques, D., Bronevetsky, G., Fernandes, R., Pingali, K., & Stodghill, P. (2005). *Optimizing checkpoint sizes in the C3 system* 19th IEEE international parallel and distributed processing symposium, Denver, CO, USA.

Martens, B., & Teuteberg, F. (2012). Decision-making in cloud computing environments: A cost and risk based approach. *Information Systems Frontiers, 14*(4), 871–893.

Mateo-Fornés, J., Solsona-Tehàs, F., Vilaplana-Mayoral, J., Teixidó-Torrelles, I., & Rius-Torrentó, J. (2019). CART, a decision SLA model for SaaS providers to keep QoS regarding availability and performance. *IEEE Access, 7*, 38195–38204.

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

Mistry, S., Bouguettaya, A., Dong, H., & Qin, A. K. (2018). Metaheuristic optimization for long-term IaaS service composition. *IEEE Transactions on Services Computing, 11*(1), 131–143.

Panda, S. K., Gupta, I., & Jana, P. K. (2019). Task scheduling algorithms for multi-cloud systems: Allocation-aware approach. *Information Systems Frontiers, 21*(2), 241–259.

Ran, Y., Yang, J., Zhang, S., & Xi, H. (2017). Dynamic IaaS computing resource provisioning strategy with QoS constraint. *IEEE Transactions on Services Computing, 10*(2), 190–202.

Randal, A. (2020). The ideal versus the real: Revisiting the history of virtual machines and containers. *ACM Computing Surveys (CSUR), 53*(1), 1–31.

Silic, M., Delac, G., Krka, I., Srbljic, S., & J. I. T. o. S. C. (2014). Scalable and accurate prediction of availability of atomic web services. *IEEE Transactions on Services Computing, 7*(2), 252–264.

Singh, S., Chana, I., & Buyya, R. (2017). STAR: SLA-aware autonomic Management of Cloud Resources. *IEEE Transactions on Cloud Computing*.

Smith, J. E., & Nair, R. (2005). The architecture of virtual machines. *Computer, 38*(5), 32–38.

Toosi, A. N., Vanmechelen, K., Ramamohanarao, K., & Buyya, R. (2015). Revenue maximization with optimal capacity control in infrastructure as a service cloud markets. *IEEE Transactions on Cloud Computing, 3*(3), 261–274.

Van, H. N., Tran, F. D., & Menaud, J.-M. (2009). *SLA-aware virtual resource Management for Cloud Infrastructures* 2009 ninth IEEE international conference on computer and information technology, Xiamen, China.

Wang, X., Du, Z., Chen, Y., & Li, S. (2008). Virtualization-based autonomic resource Management for Multi-tier web Applications in shared data center. *Journal of Systems and Software, 81*(9), 1591–1608.

Wu, L., Garg, S. K., Versteeg, S., & Buyya, R. (2014). SLA-based resource provisioning for hosted software-as-a-service applications in cloud computing environments. *IEEE Transactions on Services Computing, 7*(3), 465–485.

Yala, L., Frangoudis, P. A., Lucarelli, G., & Ksentini, A. (2018). Cost and availability aware resource allocation and virtual function placement for CDNaaS provision. *IEEE Transactions on Network and Service Management, 15*(4), 1334–1348.

Yang, J., Liu, C., Shang, Y., Cheng, B., Mao, Z., Liu, C., Niu, L., & Chen, J. (2014). A cost-aware auto-scaling approach using the workload prediction in service clouds. *Information Systems Frontiers, 16*(1), 7–18.

Yuan, S., Das, S., Ramesh, R., & Qiao, C. (2018). Service agreement trifecta: Backup resources, Price and penalty in the availability-aware cloud. Information Systems Research.

Shuai Yuan is an Assistant Professor of Department of Finance, Operations, and Information Systems at Brock University. He completed his PhD in Management Science and Systems from State University of New York at Buffalo. His current research interests include resource management, service availability analysis, and economics of Service Level Agreement management in the cloud and high-performance computing.

Sanjukta Das is the Department Chair of and an Associate Professor in the Department of Management Science and Systems in the School of Management at the University at Buffalo the State University of New York. Her research interests include cloud computing (specifically, service availability modeling, resource allocation, and contract design) and health information systems. She has published in top-tier journals such as IEEE Transactions on Computers, Information Systems Research, INFORMS Journal on Computing, and Journal of Management Information Systems. Her research has been funded by Google Faculty Awards and the National Science Foundation, Division of Computer and Network Systems.

Ram Ramesh is a Professor at the Department of Management Science & Systems in the School of Management, State University of New York at Buffalo. His research spans the areas of predictive and prescriptive analytics of availability-aware cloud computing, energy-efficient high performance computing systems, optimal design of service-level contracts in cloud computing markets, and predictive analytics of health information exchanges. His research has been funded by the National Science Foundation, Google Research, Samsung, Raytheon and Westinghouse, besides various U.S. military research programs including U.S. Army Research Institute, U.S. Air Force Office of Scientific Research, U.S. Air Force Research Laboratory and the U.S. Naval Training Systems Center. He serves as an area editor for the machine learning & knowledge management area of INFORMS Journal on Computing and is a founding co-editor-in-chief of Information Systems Frontiers (published by Springer).

Chunming Qiao is a SUNY Distinguished Professor at the CSE Department at University at Buffalo. He has lead the Lab for Advanced Network Design, Evaluation and Research (LANDR) at UB since 1993. His current research interests cover not only the safety and reliability of various cyber physical systems (such as transportation systems with connected and autonomous vehicles, critical infrastructures involving power grid and communications networks, and cloud services), but also algorithms and protocols for the Internet of Things, including smartphone based systems and applications. He has published extensively with an h-index of over 75 (according to Google Scholar). Several of his papers have received the best paper awards from IEEE and Joint ACM/IEEE venues. He also has 7 US patents and served as a consultant for several IT and Telecommunications companies since 2000. His research has been featured in BusinessWeek, Wireless Europe, CBC and New Scientists. He was elected to IEEE Fellow for his contributions to optical and wireless network architectures and protocols.