5-2017

Joint optimization of resource provisioning in cloud computing

Jonathan David CHASE
Singapore Management University, jdchase@smu.edu.sg

Dusit NIYATO

IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 10, NO. 3, MAY/JUNE 2017
pp. 396-409. DOI: 10.1109/TSC.2015.2476812

Joint Optimization of Resource Provisioning in Cloud Computing

Jonathan Chase and Dusit Niyato, *Member, IEEE*

Abstract—Cloud computing exploits virtualization to provision resources efficiently. Increasingly, Virtual Machines (VMs) have high bandwidth requirements; however, previous research does not fully address the challenge of both VM and bandwidth provisioning. To efficiently provision resources, a joint approach that combines VMs and bandwidth allocation is required. Furthermore, in practice, demand is uncertain. Service providers allow the reservation of resources. However, due to the dangers of over- and under-provisioning, we employ stochastic programming to account for this risk. To improve the efficiency of the stochastic optimization, we reduce the problem space with a scenario tree reduction algorithm, that significantly increases tractability, whilst remaining a good heuristic. Further we perform a sensitivity analysis that finds the tolerance of our solution to parameter changes. Based on historical demand data, we use a deterministic equivalent formulation to find that our solution is optimal and responds well to changes in parameter values. We also show that sensitivity analysis of prices can be useful for both users and providers in maximizing cost efficiency.

Index Terms—Cloud computing, scenario tree reduction, sensitivity analysis, software defined networking, stochastic optimization

1 Introduction

In recent years, cloud computing has increased dramatically in popularity. The Infrastructure-as-a-Service (IaaS) paradigm allows a great deal of scalability, but resources must be used efficiently. Virtual Machines (VMs) can be provisioned to meet a users' demand. Services such as Amazon's EC2 [1] allow the advance reservation of VMs, which can be utilized at a lower price than the alternative on-demand instances. This introduces the risk of oversubscription or undersubscription, so uncertain demand must be considered when provisioning VMs. Stochastic programming [2] takes into account the uncertainty of future demand and chooses an optimal reservation amount.

Increasingly, VMs placed in the cloud require a large amount of bandwidth. The popular Video-on-Demand (VoD) service, Netflix [3], uses Amazon cloud services, and requires significant bandwidth between itself and users as a fundamental resource of its service. Guaranteeing bandwidth availability is very important, but has received inadequate attention in research. A combined approach to provision both VMs and bandwidth is necessary to guarantee performance and minimize cost. Software Defined Networking (SDN) [4] has emerged as a practical solution for guaranteeing bandwidth both within data centers and across the Internet. SDN decouples the routing of data from network control, allowing for virtual networks with customized bandwidth flow, that exceeds the abilities of previous techniques. There is great potential for network providers to extend their service provision to offer bandwidth guarantees in an end-to-end fashion, as cloud providers such as Google are already doing for their own WANs [5].

In this paper we take advantage of SDN and VM technology to present a joint optimization of VM and bandwidth allocation in a cloud computing environment that handles both demand and price uncertainty. Our contributions can be summarized as follows:

- We devise a stochastic optimization to jointly reserve VMs and bandwidth across multiple time stages in a multi-user, multi-provider cloud environment, with both VM and bandwidth demand and price uncertainty. We demonstrate its optimality by formulating a deterministic equivalent problem and testing it on real historical demand data. Our formulation is able to obtain the optimal solution even when the probability distribution changes.
- The stochastic optimization is made more scalable by the use of scenario tree reduction techniques. This reduces the size of the problem space significantly whilst maintaining a close semantic match to the full problem, and yields solutions that are close to the optimum.
- We perform a sensitivity analysis, analytically determining the tolerance of the optimal solution to parameter changes. We reveal that it is important to consider routing of bandwidth, as changes in cost can result in rerouted traffic. We find sensitivity analysis is useful not only for users analyzing their decisions, but also for providers in setting optimal prices.

In this type of cloud environment, clients use a broker to provision virtual machines from cloud providers. Cloud providers deploy VMs using a hypervisor to manage instances. For example, Amazon's EC2 service uses a hypervisor based on Xen's [6] paravirtualization approach. SDN can be used to provide a virtual data center network, or Virtual

- *The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore.*
E-mail: jonathandchase@gmail.com, dniyato@ntu.edu.sg.

Private Cloud (VPC), to allow communication between instances. The OpenFlow protocol [7] is a popular 'Southbound' API for passing routing instructions to OpenFlow-enabled switches and routers. OpenFlow allows an SDN controller to provide software-based virtual networks that can guarantee Service Level Agreements (SLAs). Cloud provider-based VPCs can be joined across data centers to form an SDN-based WAN. Google's Andromeda [5] service already exploits OpenFlow for this purpose. Clients can then connect to their VPCs via VPN and make them an extension of their own networks. However, network providers currently lag behind Over-The-Top (OTT) service providers in supporting SDN. We therefore envision an SDN-based VPN that offers bandwidth provisioning through a similar model to VM provisioning, extending the SDN-based WAN concept to include the client. In this way, network providers can offer customisable bandwidth provisioning, by allowing clients to provision bandwidth along a specified network path. This provisioning of both VMs and bandwidth allows the optimization of cost, whilst guaranteeing performance. We assume that there is no inter-VM bandwidth requirement, as placing VMs efficiently to work together is beyond the scope of this paper. However, this would be a suitable direction for future work, where a web service may be comprised of multiple component VMs.

The rest of the paper is organised as follows. Section 2 describes the related work. Section 3 introduces the cloud environment and system model. Section 4 outlines the detailed stochastic problem formulation. Section 5 lays out the scenario tree reduction algorithms. Section 6 presents the sensitivity analysis formulation and method. Section 7 contains an example scenario with corresponding numerical results to compare with the previously introduced techniques.

2 Related Work

Resource provisioning in cloud computing can be broadly divided into two categories: virtual machine allocation, and bandwidth allocation. There has been extensive work in this area, each of which approaches the problem from a different angle.

[8] and [9] aim to automate VM provisioning to handle applications with QoS requirements so that the provisioning of resources is decoupled from the placement of VMs by exploiting hypervisor live migration. [10] also takes a dynamic approach to VM allocation, performing migration and consolidation. Interestingly, it uses historical demand data to predict demand so that VM provisioning can be adapted accordingly. Centralized optimization-based solutions are often NP-hard and [11], which focuses on energy efficiency, adopts a heuristic algorithm to approximate the inefficient optimal solution.

However, whilst these solutions can dynamically manage VM allocation, they do not address the bandwidth requirements that are increasingly part of VM applications. [12] addresses the problem of placing VMs with high bandwidth requirements. It simplifies the complexity of bandwidth allocation by analyzing bandwidth usage in relation to a single terminus. The optimization problem considers bandwidth as a constraint, but does not address routing of bandwidth, and is therefore too simplistic. Live migration is an important ability for balancing load on a network, and [13], [14], and [15] account for bandwidth costs during the migration. A cost function is formulated for the bandwidth of migration whilst maintaining QoS. [16] places VMs while giving bandwidth high priority, by placing VMs with a high degree of inter-communication in close proximity, limiting congestion and improving scalability. This consolidation can also be employed at the inter-data center level. [17] aims to optimize VM placement at both levels. Bandwidth is accounted for by placing VMs first at data centers to limit the communication cost and latency between data centers, whilst intra-data center placement is similar to [16]. Data center networks are more regularly structured than the Internet, which allows for greater efficiencies when allocating bandwidth. However, the technology used to guarantee data center bandwidth can also be employed in WANs. [18] uses a mixed integer optimization problem to allocate VMs on an inter-data center basis, factoring in the bandwidth requirements of VMs and aiming to maximize bandwidth availability and minimize latency.

Bandwidth guarantees throughout the network are increasingly important in cloud applications. [19] satisfies bandwidth requirements between multiple users on a data center network within a certain probability. The authors propose a virtualized cluster, such as SecondNet [20], that can provide bandwidth guarantees combined with VM allocation. [21] extends this to hetereogeneous bandwidth demands. [22] formulates an MILP to jointly place VMs and route bandwidth, with power efficiency as the main priority. Demand uncertainty is not considered, but the approach to joint allocation can be considered for the inter-data center level as well, as it is in [23].

The works outlined above consider problems with predictable demand, but in reality, demand is uncertain. [24] uses fuzzy logic to allocate VMs together with an iterative bandwidth allocation algorithm. This handles uncertainty by remaining dynamically flexible to changing demand. In contrast, [25] takes a two-phase planning approach to anticipate demand uncertainty through resource reservation. The first phase uses past usage data to estimate future reservation. The second phase aims to predict exact demand to optimally configure the usage of VMs, accounting for a potential delay in provisioning on-demand resources. [26] extends this to optimal VM reservation between cloud providers across multiple time periods, and also considers price uncertainty. The paper does not consider bandwidth allocation, however. [27] prioritises bandwidth-aware consolidation of VMs with a probability of meeting their SLAs. Bandwidth demand is a random variable per VM, which is a similar model of bandwidth demand to the one we employ here. [28] allows advance bandwidth reservation for users connecting to cloud data centers. Allocation is in two phases, first responding minimally to dynamically arriving requests, and second allocating additional bandwidth as required. Either bandwidth or transmission time can be prioritised. [29] proposes a bandwidth pricing scheme for VoD services that chooses pricing based on demand that optimizes bandwidth allocation despite selfish users. [30] combines VM allocation across multiple cloud providers and bandwidth allocation over the Internet. This paper allows for

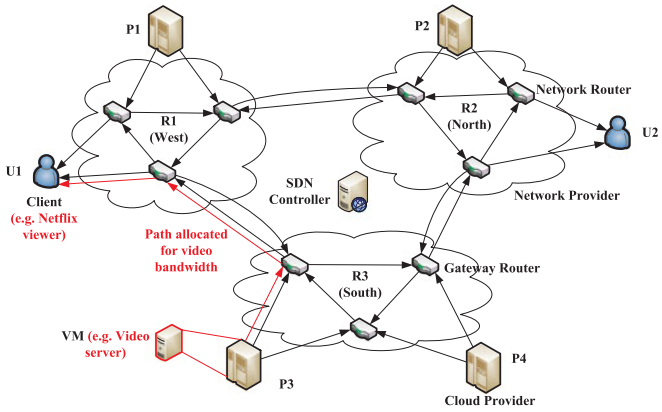


Fig. 1. Layout of network with regions, cloud providers, and clients (users) labeled.

uncertainty of VM demand by formulating a stochastic optimization, similar to the VM allocation method used in [26]. The combined method is shown to be superior to separate solutions. Flexible, custom virtualized networks afforded by SDN [4] make this possible. SDN decouples network control from data routing, and allows network providers to guarantee custom bandwidth provisions.

3 System Model

The cloud computing environment under consideration, with an example allocation scenario of a user streaming video, is illustrated in Fig. 1. Cloud providers host data centers, providing cloud services under the IaaS paradigm. Clients may provision VMs from providers, choosing an appropriate VM specification according to their needs. Providers may locate their data centers in different geographic locations, requiring communications between client and provider to go via Internet links. This traffic is transferred through routers provided by network service providers. By implementing SDN-enabled networking, bandwidth can be provisioned according to each client's demand. Clients provide VM demand requirements to an SDN controller, which places VMs with providers and provisions router bandwidth accordingly. Network routers are connected in clusters according to geographic location, with clusters connected by higher capacity gateway routers. The SDN controller must provision VMs and bandwidth in combination to optimize costs, by trading off VM savings against bandwidth costs incurred by geographic distance.

In order to save costs in the long term, both VM and network providers offer reservation options. Resources can be reserved in advance, and then utilized at a reduced rate. VM demand, and costs, may be uncertain in advance, as well as each VM's bandwidth requirements, and bandwidth costs of each router, during use. Well chosen reservation quantities and placement can provide significant cost savings over time. However, as well as meeting demand, the provisioning amounts of each resource must remain within the capacity constraints of each resource provider.

We now provide elaboration on key features of the system model. Major mathematical notations are listed in Table 1.

3.1 Provisioning Scheme

Resources are provisioned from both cloud providers and network providers under one of two provisioning plans. The first is reservation, allowing clients to guarantee resource availability in advance for an extended period (e.g. 3 months, 1 year). At the time of use, resources may be utilized under the reservation plan, or provisioned under the alternative, on-demand, plan at a costlier rate. Since the duration of a reservation plan is relatively long, we divide time into a set of provisioning stages of shorter duration (e.g.

TABLE 1
List of Key Notations

Symbol	Definition
$\mathcal{U}$	Set of all users, while $u \in \mathcal{U}$ denotes the user index
$\mathcal{P}$	Set of all cloud providers, while $p \in \mathcal{P}$ denotes the provider index
$\mathcal{V}$	Set of all VM classes, while $V_i \in \{V_1, V_2, \ldots, V_{last}\}$ denotes the VM class index
$\mathcal{R}$	Set of all network routers, while $\mathcal{R} = \{1, \ldots, R\}$ denotes the router index
$\mathcal{T}$	Set of all provisioning stages while $t \in \mathcal{T}$ denotes the time stage index
$\mathcal{L}$	Set of reservation contracts for bandwidth, while $L_1 \in \{L_1, L_2, \ldots, L_{last}\}$ denotes the contract index
$\mathcal{K}$	Set of reservation contracts for VMs, while $K_1 \in \{K_1, K_2, \ldots, K_{last}\}$ denotes the contract index
$t_p^{(h)}, t_p^{(s)}, t_p^{(n)}$	Capacity of cloud providers for processing, storage, and internal network bandwidth
t_r	Bandwidth capacity of routers
$d_i^{(h)}, d_i^{(s)}, d_i^{(n)}$	Resource demands of VM classes for processing, storage, internal network bandwidth
$d_i^{(b)}$	Resource demands of VM classes for Internet bandwidth
$v_{iut}(\omega)$	Number of VMs required under scenario ω
$c_{ipk}^{(R)}, c_{rl}^{(R)}$	Fixed first stage costs for VMs and network bandwidth
$c_{ipkt}^{(re)}(\omega), c_{ipkt}^{(u)}(\omega), c_{ipkt}^{(o)}(\omega)$	VM costs for reservation, utilization, and on-demand under all stages and scenarios
$c_{rlt}^{(re)}(\omega), c_{rlt}^{(u)}(\omega), c_{rlt}^{(o)}(\omega)$	Bandwidth costs for reservation, utilization, and on-demand under all stages and scenarios
$c_{ipkt}^{(s)}(\omega), c_{ipkt}^{(b)}(\omega)$	Storage and outbound bandwidth additional costs for VM provisioning
$X_{rul}^{(R)}, Y_{ipk}^{(R)}$	Deterministic first stage decision variables for bandwidth and VM reservation
$X_{rult}^{(re)}(\omega), X_{rult}^{(u)}(\omega), X_{rult}^{(o)}(\omega)$	Decision variables for bandwidth reservation, utilization and on-demand allocation
$Y_{uipkt}^{(re)}(\omega), Y_{uipkt}^{(u)}(\omega), Y_{uipkt}^{(o)}(\omega)$	Decision variables for VM reservation, utilization and on-demand allocation
Ω	Set of all scenarios, while $\omega \in \Omega$ denotes the scenario index

1 hour, 1 day). User demand and on-demand costs are uncertain ahead of time, meaning that each provisioning stage involves a combination of utilizing reserved resources, and provisioning on-demand resources. A provisioning stage is therefore divided into three phases. The first phase allows the initiation of additional reservation plans if necessary—for example, if a previous plan needs renewing. The second phase takes the realized demand and utilizes as much of the available reserved resources as required. In the event that this is insufficient, the third, on-demand, phase provisions resources under the on-demand plan to fulfill the shortfall.

3.2 Reservation Contracts

Reservation plans are offered to clients in the form of reservation contracts. For example, a cloud provider may offer a 3-month or 6-month contract, where resources are reserved for either 3 or 6 months, respectively. A contract may be started at any point where sufficient time stages remain to complete the duration of the contract. The set of stages at which a contract can be provisioned is therefore denoted by $\mathcal{T}_k$, defined as follows:

$$\mathcal{T}_k = \{1, \ldots, |\mathcal{T}| - |k| + 1\}, \tag{1}$$

for cloud providers, and $\mathcal{T}_l$ for network providers. Resources reserved under a contract may be utilized at any of the subsequent time stages within the duration of the contract. The resources available for utilization in time stage t, are therefore all resources reserved in previous time stages whose contracts have not yet expired. $\mathcal{M}_{kt}$, defined as follows:

$$\mathcal{M}_{kt} = \{\max(1, t - |k| + 1), \ldots, \min(t, |\mathcal{T}| - |k| + 1)\}, \tag{2}$$

is given as the set of time stages prior to t in which contract k could have been reserved, such that it is available for utilization in time stage t. $\mathcal{M}_{lt}$ denotes the equivalent term for bandwidth contract. Similarly, we give $\mathcal{N}_{tk}$, defined as follows:

$$\mathcal{N}_{tk} = \{t, \ldots, \min(|\mathcal{T}|, t + |k| - 1)\}. \tag{3}$$

as the set of time stages covered by contract k, if it was reserved in time stage t. We define $\mathcal{N}_{tl}$ equivalently for bandwidth contracts.

3.3 Uncertainty of Parameters

If all parameters are known exactly in advance, the solution is a simple deterministic optimization, with no need for an on-demand phase. In this system model, we consider the uncertainty of three parameters. The number of VMs of each class required by each client is unknown in advance. Each VM class has a combination of VM type and bandwidth requirement, and to provide per-VM bandwidth uncertainty, we allow for VM classes with identical resource requirements, but with differing external bandwidth requirements. Thus the bandwidth requirements from each provider is dependent on the number and bandwidth requirements of VMs assigned to a cloud provider, making the placement of bandwidth demands unknown in advance. Additionally, and independently, the costs of both VMs and bandwidth are also uncertain in advance, with an exception given for reservation costs in the first time stage, which are assumed to be known. It is assumed that the possible values and probability distributions of the uncertain parameters are known. To solve the problem with these uncertain factors, we formulate a stochastic integer programming problem, which considers a set of scenarios, which encompass the parameters listed. The set of all scenarios in every provisioning stage is denoted by Ω; with Ω_t denoting the set of all scenarios in provisioning stage t. Thus Ω is given by

$$\Omega = \prod_{t \in \mathcal{T}} \Omega_i = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_{|\mathcal{T}|}. \tag{4}$$

Once a scenario is known, it is called a realization. Each realization is a composite of the possible time stage demands and is defined as $\omega = (\omega_1, \ldots, \omega_{|\mathcal{T}|}) \in \Omega$. Each realization can be considered as a tuple, containing the set of VM demands for each client and the cost realization.

3.4 Provisioning Costs

Reserving VMs and bandwidth incurs three different cost types—one for each of the three provisioning phases. The reservation, utilization and on-demand costs may all be varied by the cloud providers, with only reservation in the first provisioning stage remaining fixed. The purpose of the optimization is to minimize the client's costs despite the uncertain demand and prices. First stage reservation costs are fixed, as they are assumed to be known. All subsequent pricing may vary with uncertainty. In addition to VM and bandwidth provisioning costs, cloud providers, such as Amazon [1], also charge for outbound bandwidth and storage, for certain VM types. We model this with additional cost weighting for VMs provisioned in the utilization and on-demand phases.

4 Problem Formulation

4.1 Stochastic Optimization Formulation

To solve the problem, we present a stochastic programming optimization with multi-stage recourse. The recourse action provisions utilization and on-demand resources in the multiple provisioning stages. The problem formulation is as follows:

$$\min_{X^{(R)}_{rl}, Y^{(R)}_{ipk}} \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} \sum_{u \in \mathcal{U}} \sum_{V_i \in \mathcal{V}} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}} \Big(c^{(R)}_{rl} X^{(R)}_{rul} + c^{(R)}_{ipk} Y^{(R)}_{uipk} + E_\Omega[\mathcal{Q}(X^{(R)}_{rul}, Y^{(R)}_{uipk}, \omega)] \Big), \tag{5}$$

$$Y^{(R)}_{uipk} \in N_0, \forall u \in \mathcal{U}, \forall V_i \in \mathcal{V}, \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \tag{6}$$

$$X^{(R)}_{rul} \geq 0, \forall r \in \mathcal{R}, \forall u \in \mathcal{U}, \forall l \in \mathcal{L}. \tag{7}$$

The objective function given in (5) minimizes the total cost of resource provisioning by minimizing the cost of reservation in the first stage, and the expected cost from provisioning in the remaining stages. (6) ensures the VM reservation is an integer, whilst (7) ensures the bandwidth reservation is non-negative. The expected cost from provisioning under the uncertainty set Ω is given by $E_\Omega[\mathcal{Q}(X^{(R)}_{rul}, Y^{(R)}_{uipk}, \omega)]$, where $\mathcal{Q}(X^{(R)}_{rul}, Y^{(R)}_{uipk}, \omega)$ minimizes the provisioning cost given scenario ω.

$$
\begin{aligned}
&\mathcal{Q}\big(X^{(R)}_{rul}, Y^{(R)}_{uipk}, \omega\big) \\
&= \min \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}}\sum_{t\in\mathcal{T}_l} c^{(re)}_{rlt}(\omega) X^{(re)}_{rult}(\omega) \\
&\quad + \sum_{u\in\mathcal{U}}\sum_{V_i\in\mathcal{V}}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}_k} c^{(re)}_{ipkt}(\omega) Y^{(re)}_{uipkt}(\omega) \\
&\quad + \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}}\sum_{t\in\mathcal{T}} \Big(c^{(u)}_{rlt}(\omega) X^{(u)}_{rult}(\omega) \\
&\quad + c^{(o)}_{rlt}(\omega) X^{(o)}_{rult}(\omega) \Big) \\
&\quad + \sum_{u\in\mathcal{U}}\sum_{V_i\in\mathcal{V}}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \Big((c^{(u)}_{ipkt}(\omega) \\
&\quad + c^{(s)}_{ipkt}(\omega) d^{(s)}_i + c^{(b)}_{ipkt}(\omega) d^{(b)}_i) Y^{(u)}_{uipkt}(\omega) \\
&\quad + (c^{(o)}_{ipkt}(\omega) + c^{(s)}_{ipkt}(\omega) d^{(s)}_i \\
&\quad + c^{(b)}_{ipkt}(\omega) d^{(b)}_i) Y^{(o)}_{uipkt}(\omega) \Big).
\end{aligned} \tag{8}
$$

The cost function (8) minimizes the cost under uncertainty scenario ω using six decision variables. These variables represent the bandwidth and VM allocations for each provisioning phase. The total provisioning cost is summed across all routers, providers, contracts, clients, and time stages—weighted by the corresponding provisioning costs. This cost function, together with (5), form the objective for our stochastic formulation. This formulation is subject to a set of constraints.

$$
\begin{aligned}
&Y^{(u)}_{uipkt}(\omega) \le \sum_{\hat{t}\in\mathcal{M}_{kt}} Y^{(re)}_{uipk\hat{t}}(\omega), \\
&\forall u \in \mathcal{U}, \forall V_i \in \mathcal{V}, \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T},
\end{aligned} \tag{9}
$$

$$
X^{(u)}_{rult}(\omega) \le \sum_{\hat{t}\in\mathcal{M}_{lt}} X^{(re)}_{rul\hat{t}}(\omega), \forall u \in \mathcal{U}, \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}. \tag{10}
$$

(9) and (10) ensure that the VMs and bandwidth provisioned in the utilization phase do not exceed the available reserved resources, as only pre-reserved resources are entitled to the cheaper utilization rate. If a reserved contract's duration does not cover a given time stage, no resources from that contract can be utilized—a new reservation is required,

$$
X^{(R)}_{rul} = X^{(re)}_{rul\bar{t}}(\omega), \bar{t} = 1, \forall u \in \mathcal{U}, \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \tag{11}
$$

$$
Y^{(R)}_{uipk} = Y^{(re)}_{uipk\bar{t}}(\omega), \bar{t} = 1, \forall u \in \mathcal{U}, \forall V_i \in \mathcal{V}, \forall p \in \mathcal{P}, \forall k \in \mathcal{K}. \tag{12}
$$

(11) and (12) ensures that any reservations made in the first stage are exempt from price uncertainty, as the reservation decision is identical across all scenarios in the first time stage,

$$
\sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}} \Big(X^{(u)}_{rult}(\omega) + X^{(o)}_{rult}(\omega) \Big) \le t_r, \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \tag{13}
$$

$$
\begin{aligned}
&\sum_{V_i\in\mathcal{V}} d^{(h)}_i \Big(\sum_{u\in\mathcal{U}}\sum_{k\in\mathcal{K}} (Y^{(u)}_{uipkt}(\omega) + Y^{(o)}_{uipkt}(\omega)) \Big) \le t^{(h)}_p, \\
&\forall p \in \mathcal{P}, \forall t \in \mathcal{T},
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
&\sum_{V_i\in\mathcal{V}} d^{(s)}_i \Big(\sum_{u\in\mathcal{U}}\sum_{k\in\mathcal{K}} (Y^{(u)}_{uipkt}(\omega) + Y^{(o)}_{uipkt}(\omega)) \Big) \le t^{(s)}_p, \\
&\forall p \in \mathcal{P}, \forall t \in \mathcal{T},
\end{aligned} \tag{15}
$$

$$
\begin{aligned}
&\sum_{V_i\in\mathcal{V}} d^{(n)}_i \Big(\sum_{u\in\mathcal{U}}\sum_{k\in\mathcal{K}} (Y^{(u)}_{uipkt}(\omega) + Y^{(o)}_{uipkt}(\omega)) \Big) \le t^{(n)}_p, \\
&\forall p \in \mathcal{P}, \forall t \in \mathcal{T}.
\end{aligned} \tag{16}
$$

(13)-(16) ensure that the VMs and bandwidth that are used by the client do not exceed the resource capacities of the cloud providers and routers from which they are provisioned. The total quantities of each resource used should not exceed the capacity of each provider for that resource,

$$
\begin{aligned}
&\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}} \Big(Y^{(u)}_{uipkt}(\omega) + Y^{(o)}_{uipkt}(\omega) \Big) \ge v_{iut}(\omega), \\
&\forall V_i \in \mathcal{V}, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}.
\end{aligned} \tag{17}
$$

(17) requires that the utilization and on-demand provisioning of VMs from each class across all cloud providers is sufficient to meet the realized demand for that class of VM, for each client, in each time stage. This demand is summed across all providers, allowing the placing of VMs with cloud providers such that both VM and bandwidth costs are optimized jointly,

$$
Y^{(re)}_{uipkt}(\omega) \in N_0, \forall u \in \mathcal{U}, \forall V_i \in \mathcal{V}, \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}_k, \tag{18}
$$

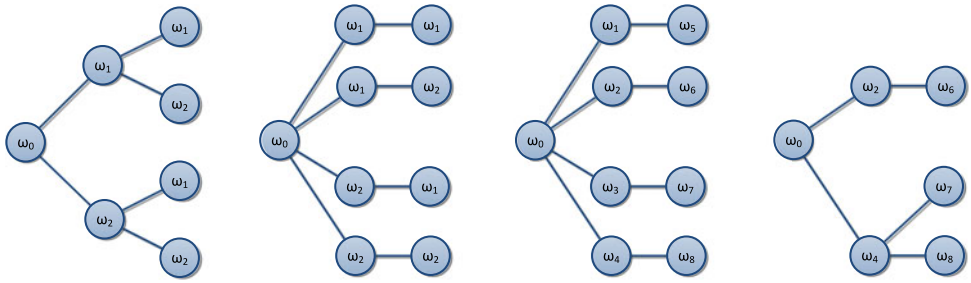
$$
\begin{aligned}
&Y^{(u)}_{uipkt}(\omega), Y^{(o)}_{uipkt}(\omega) \in N_0, \\
&\forall u \in \mathcal{U}, \forall V_i \in \mathcal{V}, \forall p \in \mathcal{P}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T},
\end{aligned} \tag{19}
$$

$$
X^{(re)}_{rult}(\omega) \ge 0, \forall r \in \mathcal{R}, \forall u \in \mathcal{U}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}_l, \tag{20}
$$

$$
X^{(u)}_{rult}(\omega), X^{(o)}_{rult}(\omega) \ge 0, \forall r \in \mathcal{R}, \forall u \in \mathcal{U}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}. \tag{21}
$$

(18) and (19) ensure that the number of VMs provisioned in each phase is a non-negative integer. Similarly, (20) and (21) require that all router bandwidth provisioning is non-negative. Treating VMs as atomic means that our problem is mixed integer, and therefore more complex to solve. It is possible to consider VM provisioning as a percentage of time used rather than as an atomic unit, thus we could reasonably convert a mixed integer problem into a linear program. As long as VM demand is an integer, we find that the result in this case remains the same. This relaxation becomes important when performing sensitivity analysis, covered in Section 6.

As well as meeting VM demand, it is essential that bandwidth demand is also met for a joint solution to work correctly. This means that as well as sufficient bandwidth being allocated to the routers adjacent to the cloud providers' data centers, the allocation must be preserved consistently across the network back to the client. We consider the networks as a graph, $G = (\mathcal{R}, \mathcal{E})$, where each vertex is a router in $\mathcal{R}$, and each edge in $\mathcal{E}$ is a link. We assume that provisioned upload and download bandwidth is symmetric. In general, traffic is considered to flow 'out' from cloud providers and 'in' to clients. All routers have both 'in' and 'out' links. This network labelling is illustrated in Fig. 1, where example nodes and edges are labelled accordingly. The flow constraints are defined as follows:



(a) Original Tree (b) Fan (c) Relabeled Tree (d) Reduced Tree

Fig. 2. Possible scenario tree layouts.

$$\sum_{e\in\mathcal{E}_r^{out}}\sum_{l\in\mathcal{L}} f_{eult}(\omega) - \sum_{e\in\mathcal{E}_r^{in}}\sum_{l\in\mathcal{L}} f_{eult}(\omega) = 0, \quad \forall r\in\mathcal{R}, \forall u\in\mathcal{U}, \forall t\in\mathcal{T}, \tag{22}$$

$$\sum_{e\in\mathcal{E}_r^{out}}\sum_{l\in\mathcal{L}} f_{eult}(\omega) - \sum_{l\in\mathcal{L}}\Big(X_{rult}^{(u)}(\omega) + X_{rult}^{(o)}(\omega)\Big) = 0, \quad \forall r\in\mathcal{R}, \forall u\in\mathcal{U}, \forall t\in\mathcal{T}, \tag{23}$$

$$\sum_{e\in\mathcal{E}_p^{out}}\sum_{l\in\mathcal{L}} f_{eult}(\omega) \geq \sum_{V_i\in\mathcal{V}}\sum_{k\in\mathcal{K}} d_i^{(b)}\Big(Y_{uipkt}^{(u)}(\omega) + Y_{uipkt}^{(o)}(\omega)\Big), \quad \forall p\in\mathcal{P}, \forall u\in\mathcal{U}, \forall t\in\mathcal{T}, \tag{24}$$

$$\sum_{e\in\mathcal{E}_u^{in}}\sum_{l\in\mathcal{L}} f_{eult}(\omega) \geq \sum_{p\in\mathcal{P}}\Big(\sum_{e\in\mathcal{E}_p^{out}}\sum_{l\in\mathcal{L}} f_{eult}(\omega)\Big), \quad \forall u\in\mathcal{U}, \forall t\in\mathcal{T}. \tag{25}$$

- (22) is a flow conservation constraint. The variable $f_{eult}(\omega)$ denotes the amount of network flow provisioned on edge e, for client u, under contract l, at time t. The set $\mathcal{E}_r^{out}$ contains the edges that connect directly to router r in the outward direction. Similarly, $\mathcal{E}_r^{in}$ denotes the set of edges that connect directly to router r in the inward direction. Thus we ensure that the sum of all traffic flowing into a router is equal to the sum of all traffic leaving the router—no bandwidth use is gained or lost between cloud provider and client throughout the network.
- (23) relates the flow variables defined on edges, to the bandwidth allocated to routers. Bandwidth is obtained from routers in the network. Consequently, this constraint ensures that the traffic flowing through the router is equal to the total bandwidth allocated to the router in a given time stage.
- (24) ensures that the bandwidth requirements of all VMs on all cloud providers are satisfied by the edges that connect directly to those cloud providers (denoted by the set $\mathcal{E}_p^{out}$). The total bandwidth demand on each cloud provider in scenario ω is the sum of the individual bandwidth requirements of each of the VMs provisioned from that cloud provider.
- (25) serves a similar purpose to (23), guaranteeing that the bandwidth requirements of all VMs associated with client u are satisfied by the edges that connect directly to the client, denoted by the set $\mathcal{E}_u^{in}$. Together with (22), this ensures that sufficient bandwidth is provisioned throughout the network, in a routing solution that is as cost effective as possible. This constraint offers a guarantee that the same amount of traffic that leaves a cloud provider reaches the client that it was intended for. Thus, demand is met and routing is optimized.

4.2 Deterministic Equivalent Formulation

If a stochastic program has finite support, the set of different uncertainty scenarios can be enumerated. By formulating the scenarios as another set in the domain, and their probabilities as weights for the cost coefficients in the objective function, we can formulate an equivalent deterministic optimization problem that can be solved using established methods. This is useful as the program can be solved as a mixed integer or linear program. However, even for a small number of demand scenarios, the size of the problem space can grow rapidly. Thus, the tractability of this method needs to be enhanced. Therefore, the virtue of Scenario Tree Reduction becomes apparent, in making a reasonable trade-off between optimality and complexity. In the following section, we introduce this technique in detail.

5 Scenario Tree Reduction

The uncertainty introduced in Section 3.3 can be modeled using a tree of scenarios. Assuming that there is a finite set of possible scenarios, a tree can be drawn like the one shown in Fig. 2a. Each level of the tree represents a time stage in which provisioning decisions are made with each possible sequence of events being mapped as a path from the root node to the leaf nodes. Each node in the tree represents a scenario in Ω, with each node label corresponding to a scenario subscript, such as ω_2. To solve the stochastic optimization formulated in (5)-(25), we can enumerate the set of possible scenarios and solve the problem as a deterministic mixed integer linear program, by assigning a probability coefficient, such as π_{w_2}, to each scenario. To enumerate each scenario, we can reformulate the tree as a fan, as shown in Fig. 2b. In the fan structure, each node in tree stage $t-1$ is replicated to match the number of scenarios in time stage t. With the exception of the root node, therefore, each node has a maximum of one ancestor and one descendant, with the number of tree branches equal to the number of leaf

nodes. The scope of a scenario realization is a single time stage, so a given scenario may occur in consecutive time stages. With the exception of the root node, the probability of each tree node is equal to its ancestor and descendant, found by multiplying together the probabilities of all scenarios in its branch. For simple enumeration of scenarios, we can relabel the tree as shown in Fig. 2c, with each node given a unique ID.

Solving a stochastic program is straightforward when the number of scenarios is small. However, the number of nodes in the tree can grow very large as the numbers of scenarios and time stages increase. In this case, the problem space can become enormous, leading to tractability problems. Thus, to improve the performance of the optimization, it is desirable to reduce the size of the scenario tree and reconstruct it as a reduced tree, similar to Fig. 2d.

5.1 Tree Reduction

In this section, we introduce an algorithm to reduce the size of the scenario tree heuristically whilst retaining a good approximation of the full tree's performance. The reduction algorithm creates two sets - Ω_t denoting the scenarios in time stage t that remain in the tree, and Φ_t containing the set of deleted scenarios from time stage t. Scenarios are deleted such that the *probability distance* measured between Ω_t and Φ_t is minimized. Different metrics can be used to measure the probability distance. In this case, we employ the Kantorovich distance [31], denoted by $\mathcal{L}[\Omega_t, \Phi_t]$, and is given as follows:

$$\mathcal{L}[\Omega_t, \Phi_t] = \sum_{\omega \in \phi_t} \pi_\omega \min_{\omega' \in \Omega_t / \Phi_t} \ell_t[\omega, \omega'], \tag{26}$$

where $\ell_t[\omega, \omega']$ is the probability distance between scenarios ω and ω', and is as follows:

$$\ell_t[\omega, \omega'] = \sum_{\bar{t} \in t_0, \ldots, t} \|\xi(\omega(\bar{t})) - \xi(\omega'(\bar{t}))\|^r, \tag{27}$$

where $\xi(\omega(\bar{t}))$ is the random vector of scenario $\omega(\bar{t})$ in time stage $\bar{t}$. Minimizing the Kantorovich distance gives an optimization problem, which provides the maximum cardinality of the removed scenario set. However, solving this can be NP-hard, which would be contrary to the objective of scenario reduction. Therefore, a heuristic algorithm can be formulated to obtain a set of removed scenarios. This algorithm offers a tradeoff between accuracy and execution time. The reduction heuristic algorithm is as follows:

- The Backward Reduction Algorithm [32], shown in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSC.2015.2476812, is used to reduce the scenario tree. The algorithm functions as follows: given the desired number of scenarios, N_t, to be removed from time stage t, the algorithm iteratively selects a scenario for deletion. A scenario is chosen for deletion that minimizes the probability distance between the deleted scenario and the nearest remaining scenario in Ω_t. The algorithm terminates once N_t scenarios have been deleted from Ω_t and added to Φ_t.
- A second algorithm, given in Appendix B, available in the online supplemental material, is then run to redistribute the probabilities among the remaining scenarios in Ω_t. Algorithm 2 finds the closest scenario in Ω_t for each scenario in Φ_t. The probability of the scenario in Φ_t is added to the scenario in Ω_t. This is repeated N_t times.
- (26) is then calculated and compared to a threshold value ϵ_t. If the result is lower than the threshold, the scenario set is retained. Otherwise, N_t is reduced and the process is repeated.

5.2 Tree Construction

The algorithm outlined above is sufficient if there is only a single time stage below the root node. With more scenarios, the tree must be reconstructed as each branch of the scenario tree must be connected to the root node, with a uniform number of levels. For leaf nodes that are removed, its ancestors in the tree are also removed. For higher levels of the tree, its predecessors are removed, as for the leaf nodes, and its descendants are assigned as descendants of the nearest remaining scenario. Probabilities are recalculated as before, by adding the probability of the removed scenario to the nearest remaining scenario's probability. The algorithm described is given in Appendix C, available in the online supplemental material. The result is a reduced scenario tree, which is quicker to traverse yet maintains a good semantic approximation of the full scenario tree.

6 Sensitivity Analysis

The purpose of stochastic programming is to find an optimal solution given uncertainty in parameters. In this paper, we allow for uncertainty of pricing and demand. Sensitivity analysis allows the examination of a solution to discover its sensitivity to parameter changes. In [33], an analytical sensitivity analysis approach is derived for linear programming, which we employ here. We assume that VM and bandwidth demand has finite support, and can therefore treat the stochastic program outlined previously as a deterministic equivalent formulation with scenario probabilities becoming cost coefficients in the objective function. We relax the integer restriction on VM provisioning, making the primal formulation a linear program, convert any $\geq$ inequalities to be $\leq$ and replace all equalities with a pair of inequalities. The primal problem (given in (5)-(25)) is therefore of the standard form:

$$\min_x \{c^T x : Ax \geq b, x \geq 0\}, \tag{28}$$

and accompanying dual problem is of the form:

$$\max_{y,s} \{b^T y : A^T y + s \leq c, s \geq 0\}. \tag{29}$$

The dual formulation that accompanies the primal problem is given in detail in (30)-(41) as follows:

$$\begin{aligned} \max - & \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} W^{(13)}_{rt\omega} t_r \\ & - \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \left(W^{(14)}_{pt\omega} t^{(h)}_p + W^{(15)}_{pt\omega} t^{(s)}_p + W^{(16)}_{pt\omega} t^{(n)}_p \right) \\ & + \sum_{V_i \in \mathcal{V}} \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} W^{(17)}_{iut\omega} v_{iut\omega}, \end{aligned} \tag{30}$$

$$\sum_{\omega\in\Omega_1}\left(W^{(12LT)}_{rul\bar{t}\omega}-W^{(12GT)}_{rul\bar{t}\omega}\right)\leq c^{(R)}_{rl},\quad \bar{t}=1,\forall r\in\mathcal{R},\forall u\in\mathcal{U},\forall l\in\mathcal{L}, \tag{31}$$

$$\sum_{\omega\in\Omega_1}\left(W^{(11LT)}_{piuk\bar{t}\omega}-W^{(11GT)}_{piuk\bar{t}\omega}\right)\leq c^{(R)}_{ipk},\quad \bar{t}=1,\forall p\in\mathcal{P},\forall V_i\in\mathcal{V},\forall u\in\mathcal{U},\forall\in\mathcal{K}, \tag{32}$$

$$-W^{(12LT)}_{rul\bar{t}\omega}+W^{(12GT)}_{rul\bar{t}\omega}+W^{(10)}_{rul\bar{t}\omega}\leq c^{(re)}_{rl\bar{t}\omega}p(\omega),\quad \bar{t}=1,\forall r\in\mathcal{R},\forall u\in\mathcal{U},\forall l\in\mathcal{L},\forall\omega\in\Omega, \tag{33}$$

$$\sum_{\hat{t}\in\mathcal{N}_{lt}}W^{(10)}_{rul\hat{t}\omega}\leq c^{(re)}_{rlt\omega}p(\omega),\quad \forall r\in\mathcal{R},\forall u\in\mathcal{U},\forall l\in\mathcal{L},\forall t\in\mathcal{T}_l/\{t_1\},\forall\omega\in\Omega, \tag{34}$$

$$-W^{(11LT)}_{piuk\bar{t}\omega}+W^{(11GT)}_{piuk\bar{t}\omega}+W^{(9)}_{piuk\bar{t}\omega}\leq c^{(re)}_{ipk\bar{t}\omega}p(\omega),\quad \bar{t}=1,\forall p\in\mathcal{P},\forall V_i\in\mathcal{V},\forall u\in\mathcal{U},\forall k\in\mathcal{K},\forall\omega\in\Omega, \tag{35}$$

$$\sum_{\hat{t}\in\mathcal{N}_{kt}}W^{(9)}_{piuk\hat{t}\omega}\leq c^{(re)}_{ipkt\omega}p(\omega),\quad \forall p\in\mathcal{P},\forall V_i\in\mathcal{V},\forall u\in\mathcal{U},\forall k\in\mathcal{K},\forall t\in\mathcal{T}_k/\{t_1\},\forall\omega\in\Omega, \tag{36}$$

$$-W^{(10)}_{rul t\omega}-W^{(13)}_{rt\omega}-W^{(23LT)}_{rut\omega}+W^{(23GT)}_{rut\omega}\leq c^{(u)}_{rlt\omega}p(\omega),\quad \forall r\in\mathcal{R},\forall u\in\mathcal{U},\forall l\in\mathcal{L},\forall t\in\mathcal{T},\forall\omega\in\Omega, \tag{37}$$

$$-W^{(13)}_{rt\omega}-W^{(23LT)}_{rut\omega}+W^{(23GT)}_{rut\omega}\leq c^{(o)}_{rlt\omega}p(\omega),\quad \forall r\in\mathcal{R},\forall u\in\mathcal{U},\forall l\in\mathcal{L},\forall t\in\mathcal{T},\forall\omega\in\Omega, \tag{38}$$

$$\begin{aligned}&-W^{(9)}_{piukt\omega}-d^{(h)}_i W^{(14)}_{pt\omega}-d^{(s)}_i W^{(15)}_{pt\omega}-d^{(n)}_i W^{(16)}_{pt\omega}+W^{(17)}_{uit\omega}\\&-d^{(b)}_i W^{(24)}_{put\omega}\leq\left(c^{(u)}_{ipkt\omega}+c^{(s)}_{ipkt}(\omega)d^{(s)}_i+c^{(b)}_{ipkt}(\omega)d^{(b)}_i\right)p(\omega),\\&\quad\forall p\in\mathcal{P},\forall V_i\in\mathcal{V},\forall u\in\mathcal{U},\forall k\in\mathcal{K},\forall t\in\mathcal{T},\forall\omega\in\Omega,\end{aligned} \tag{39}$$

$$\begin{aligned}&-d^{(h)}_i W^{(14)}_{pt\omega}-d^{(s)}_i W^{(15)}_{pt\omega}-d^{(n)}_i W^{(16)}_{pt\omega}+W^{(17)}_{uit\omega}-d^{(b)}_i W^{(24)}_{put\omega}\\&\qquad\leq\left(c^{(o)}_{ipkt\omega}+c^{(s)}_{ipkt}(\omega)d^{(s)}_i+c^{(b)}_{ipkt}(\omega)d^{(b)}_i\right)p(\omega),\\&\qquad\forall p\in\mathcal{P},\forall V_i\in\mathcal{V},\forall u\in\mathcal{U},\forall k\in\mathcal{K},\forall t\in\mathcal{T},\forall\omega\in\Omega,\end{aligned} \tag{40}$$

$$\begin{aligned}&\sum_{r\in\mathcal{R}^{out}_e}\left(W^{(22LT)}_{rut\omega}-W^{(22GT)}_{rut\omega}+W^{(23LT)}_{rut\omega}-W^{(23GT)}_{rut\omega}\right)\\&+\sum_{r\in\mathcal{R}^{in}_e}\left(-W^{(22LT)}_{rut\omega}+W^{(22GT)}_{rut\omega}\right)\\&+\sum_{p\in\mathcal{P}^{out}_e}\left(W^{(24)}_{put\omega}-W^{(25)}_{ut\omega}\right)+\sum_{u\in\mathcal{U}^{in}_e}W^{(25)}_{ut\omega}\leq 0,\\&\forall e\in\mathcal{E},\forall u\in\mathcal{U},\forall l\in\mathcal{L},\forall t\in\mathcal{T},\forall\omega\end{aligned} \tag{41}$$

(30) is the dual objective function. The dual variables are given by W, where the superscript indicates the equation number from the primal problem that the dual variable corresponds to. In the case of the equality constraints, (11), (12), (22), and (23), in the original formulation, a suffix (either LT or GT) is added to the superscript to represent the two inequality constraints ($\leq$ and $\geq$, respectively) that replace the equalities. (31) corresponds to the primal variables $X^{(R)}$, (32) to the primal variables $Y^{(R)}$. (33) and (34) correspond to the variables $X^{(re)}$. (33) addresses the special case of the first time stage, whilst (34) corresponds to all subsequent time stages. (35) and (36) correspond similarly to the primal variables $Y^{(re)}$. (37)-(40) correspond to the primal variables X^u, X^o, Y^u, and Y^o, respectively. Finally, (41) is the dual equation for the primal flow variables, f. The parameter $p(\omega)$ is the probability of ω.

Given this dual formulation, from [33] we have the following rules for the shape of the optimal value function graph as the cost coefficient changes:

- The optimal value function, $f(\gamma)$, is defined as the change in value of the optimal solution as the cost coefficient of one of the variables, for example $c^{(R)}_{rl}$, varies by the offset value γ.
- The optimal value function is concave and piecewise linear.
- $\mathcal{P}^*_\gamma$ denotes the set of optimal solutions for the primal problem with the offset value ,γ, for example $c^{(R)}_{rl}+\gamma$. $\mathcal{P}^*_\gamma$ is constant for (γ_1,γ_2), if $f(\gamma)$ is linear across the interval $[\gamma_1,\gamma_2]$.
- If we have γ_1 and γ_2 such that $\mathcal{P}^*_{\gamma 1}=\mathcal{P}^*_{\gamma 2}$, we can define $\bar{\mathcal{P}}^*=\mathcal{P}^*_\gamma$ for all $\gamma\in[\gamma_1,\gamma_2]$. This is a linear interval of $f(\gamma)$.

The behaviour of the optimal value function within a linear interval is now defined. To obtain sensitivity analysis results, we must simply find the breakpoints dividing the intervals from each other. We denote the optimal value by z^*, found by solving the primal or dual problem. This allows the finding of the sensitivity results for any cost coefficient. Two linear programming problems can be solved to give γ_{min} and γ_{max}, which are the lower and upper bounds, respectively, for the perturbation variable γ, such that the optimal values of the primal variables X^* and Y^*, and the dual variables, W^* remain correct. We give the standard form of the two optimization problems in (42) and (43), where $e_j=1$ if j is the index of the c_j under analysis, else $e_j=0$,

$$\gamma_{min}=\min\{\gamma: A^Ty+s=c+\gamma e_j, b^Ty=c^Tx^*+\gamma x^*_j, s\geq 0\}, \tag{42}$$

$$\gamma_{max}=\max\{\gamma: A^Ty+s=c+\gamma e_j, b^Ty=c^Tx^*+\gamma x^*_j, s\geq 0\}. \tag{43}$$

In (44), we give an example of the objective function for finding the lower bound on the cost coefficient of one of the decision variables, X_j, which can be any of the decision variables, for example, one of the $X^{(R)}_{rul}$ variables,

$$\gamma_{min} = \min\Big\{\gamma : -\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}}\sum_{\omega\in\Omega} W^{(13)}_{rt\omega} t_r - \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}}\sum_{\omega\in\Omega}\big(W^{(14)}_{pt\omega} t^{(h)}_p + W^{(15)}_{pt\omega} t^{(s)}_p + W^{(16)}_{pt\omega} t^{(n)}_p\big) + \sum_{V_i\in\mathcal{V}}\sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{T}}\sum_{\omega\in\Omega} W^{(17)}_{iut\omega} v_{iut\omega}$$
$$= \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}}\Big[c^{(R)}_{rl} X^{(R)*}_{rul} + \sum_{\omega\in\Omega} p(\omega)\Big(\sum_{t\in\mathcal{T}_l} c^{(re)}_{rlt\omega} X^{(re)*}_{rult\omega} + \sum_{t\in\mathcal{T}}\Big(c^{(u)}_{rlt\omega} X^{(u)*}_{rult\omega} + c^{(o)}_{rlt\omega} X^{(o)*}_{rult\omega}\Big)\Big)\Big]$$
$$+ \sum_{u\in\mathcal{U}}\sum_{V_i\in\mathcal{V}}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\Big[c^{(R)}_{ipk} Y^{(R)*}_{uipk} + \sum_{\omega\in\Omega} p(\omega)\Big(\sum_{t\in\mathcal{T}_k} c^{(re)}_{ipkt\omega} Y^{(re)*}_{uipkt\omega} + \sum_{t\in\mathcal{T}}\Big(c^{(s)}_{ipkt}(\omega) d^{(s)}_i + c^{(b)}_{ipkt}(\omega) d^{(b)}_i\Big)\Big(c^{(u)}_{ipkt\omega} Y^{(u)*}_{uipkt\omega} + c^{(o)}_{ipkt\omega} Y^{(o)*}_{uipkt\omega}\Big)\Big)\Big] + \gamma X_j\Big\}. \quad (44)$$

(45) gives an example of the modified dual problem constraint in (31) corresponding to the coefficient under analysis, in this case for one of the coefficients of $X^{(R)}_{rul}$. In practice, when implementing the sensitivity analysis, the other dual constraints can be left in their original form with no modification required. The constraint is as follows:

$$\sum_{\omega\in\Omega_1} W^{(12)}_{rul\bar{t}\omega} \le c^{(R)}_{rl} + \gamma e_j, \bar{t} = 1, \forall r \in \mathcal{R}, \forall u \in \mathcal{U}, \forall l \in \mathcal{L}, \quad (45)$$

where, as in (42) and (43), $e_j = 1$ if j denotes the variable whose coefficient is under analysis, else $e_j = 0$.

These optimization problems can be encoded for a solver such as CPLEX [34], and solved as a linear program. A constraint of this method is that it is limited to modifying a single variable at a time. When studying the cost coefficient of a decision variable in the deterministic linear programming equivalent formulation of our stochastic program, it is also necessary to divide by the probability weight of the variable, to find the true cost sensitivity range, after the linear programs have been solved. Nonetheless, despite these restrictions, this remains a useful technique for examining the tolerance of the stochastic program solution, as well as the interaction between decision variables.

7 Performance Evaluation

We simulate a cloud environment to test the effectiveness of our stochastic optimization solution and examine the success of scenario tree reduction and sensitivity analysis.

7.1 Parameter Settings

The parameters are designed to provide a reasonable and interesting environment for yielding numerical results as follows.

7.1.1 VMs and Cloud Providers

In our test environment we consider three distinct VM applications, representing three classes. The computing requirements for $V1$, $V2$, and $V3$ are 24, 24, and 48 CPU-hours per day, respectively. The daily storage requirements for storage are 160, 410, and 840 GBs for $V1$, $V2$, and $V3$, respectively. Daily internal network requirements are 5, 10, and 100 Mbps for $V1$, $V2$, and $V3$, respectively. Our simulated environment is divided into three distinct geographic locations in the UK: North, South, and West. We include four cloud providers, all four are large scale commercial providers, and thus can be considered exempt from capacity constraints. Cloud providers offer two contract lengths: 1 hour, and 2 hours. We choose these time periods for optimal illustration, but any time stage length and number of time stages could be used.

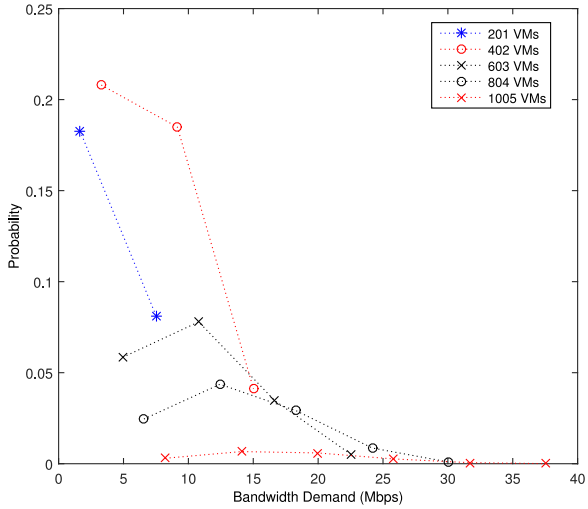


Fig. 3. Probability distribution of VM demand test data. Each curve represents a different VM demand scenario.

7.1.2 Network Provider Routers

Our test environment contains 17 routers, arranged according to Fig. 1. Routers belong to one of two classes, with gateway routers connecting regions having higher capacity than those within regions. The capacities of the router classes are 50 Mbps for regional routers, and 100 Mbps for gateway routers. The price of provisioning bandwidth from each geographic region is different, with bandwidth also obtainable under two contracts, as with VMs.

7.1.3 Clients

We allow for two clients, one based in the North, the other in the West. Each client has their own VM demands. For simplicity of illustration, we make the number of required VMs from each class the same for each client and cloud provider in a given demand realization.

7.1.4 Uncertainty Parameters

We base the uncertainty of VM demand on test data obtained from Google Cluster Trace files [35], as shown in Fig. 3. Bandwidth demand for $V1$ and $V2$ is reasonably synthesized based on IP traces for web traffic [36], with two demand levels for each class. Bandwidth demand for V_3 is based on usage estimates for Netflix Video on Demand [37], representing a different class of network traffic. Demand for class $V3$ varies from 1 to 5, and 100 to 500 (in units of 100) for classes $V1$ and $V2$ (representing more popular but less bandwidth-intensive activities), for each client on a cloud provider. For simplicity, for a given realization, VM demand level is taken to be equivalent for

TABLE 2
VM Costs for each Cloud Provider and Provisioning Phase

Provider	VM, Contract, and Phase														
	V_1					V_2					V_3				
	$1-hour$			$2-hour$		$1-hour$			$2-hour$		$1-hour$			$2-hour$	
	R	U	O	R	U	R	U	O	R	U	R	U	O	R	U
P_1	0.013	0.077	0.154	0.016	0.063	0.025	0.154	0.308	0.031	0.126	0.050	0.308	0.616	0.063	0.252
P_2	0.014	0.087	0.158	0.018	0.069	0.028	0.175	0.315	0.036	0.139	0.056	0.349	0.630	071	0.277
P_3	0.013	0.064	0.113	0.016	0.050	0.025	0.127	0.225	0.031	0.101	0.050	0.254	0.450	0.063	0.201
P_4	0.013	0.064	0.113	0.016	0.050	0.025	0.127	0.225	0.031	0.101	0.050	0.254	0.450	0.063	0.201

each VM class. For example, if the demand for $V3$ is 1, the demand for $V1$ and $V2$ will be 100. Prices for VMs are given in Table 2, listing upfront costs for reservation and hourly costs for utilization and on-demand. Bandwidth prices are given in Table 3. VM pricing varies according to the cloud provider location, with prices based on Amazon EC2 costs. On-demand prices are identical regardless of contract length so are only stated once. Router prices in each region are listed per Mbps, and are based on average prices according to NetIndex [38]. All prices are listed in USD.

7.2 Numerical Results

We encode the deterministic equivalent model in GAMS with the listed parameters and solve it with CPLEX [34].

7.2.1 Effects of Reservation

The purpose of using stochastic optimization is to achieve resource reservation that minimizes both oversubscription and undersubscription. We first vary the total quantity of reserved VMs in the system and observe the effects on total cost, shown in Fig. 4. As expected, the first stage cost increases and the second stage cost decreases, as the increased reservation requires less on-demand provisioning. The lowest point on the Total Cost curve shows the ideal tradeoff point to allow for multiple demand scenarios. It should be noted that there is still some on-demand usage at this point, as indicated by the second stage graph, which is still decreasing. This is the consequence of accounting for differing amounts of anticipated demand. We observe a similar pattern for the bandwidth results, which can be found in Appendix D, available in the online supplemental material. The difference is less dramatic due to the difference in cost and demand values compared to VMs, but the effect is the same. The second stage cost becomes constant as the reservation amount passes the point where all possible demand scenarios are accounted for, and the only increase in cost is due to gratuitous reservation. We combine the two and observe a joint effect in Fig. 5. This figure shows the importance of optimizing both VM and bandwidth reservation, as well as showing that erring on the side of overprovisioning is preferable to underprovisioning, as the cost penalties are more severe in the latter case.

TABLE 3
Cost per Mbps of Bandwidth Provisioning through Network Provider Routers

Router Region	Contract, Provisioning Phase				
	$1-hour$			$2-hour$	
	R	U	O	R	U
$R_1(West)$	0.110	0.038	0.625	0.213	0.029
$R_2(North)$	0.120	0.025	0.500	0.233	0.021
$R_3(South)$	0.154	0.013	0.375	0.296	0.010

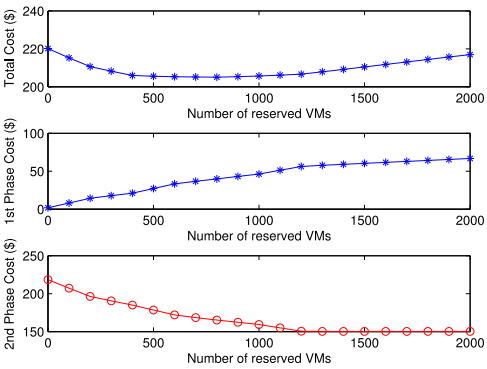


Fig. 4. Variation of cost with VM reservation.

7.2.2 Importance of Joint Approach

The problem formulation in this paper is based on the premise that a joint approach to VM and bandwidth allocation is superior to attempting to provision these resources separately. To test this hypothesis, we conduct two tests to

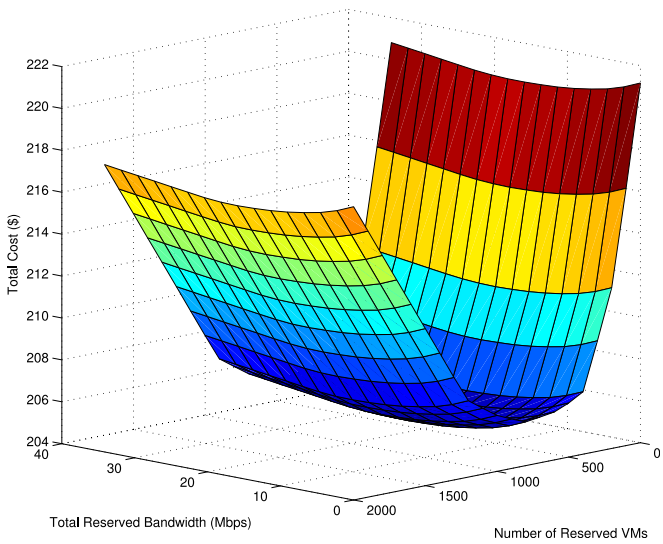


Fig. 5. Variation of cost with VM and bandwidth reservation.

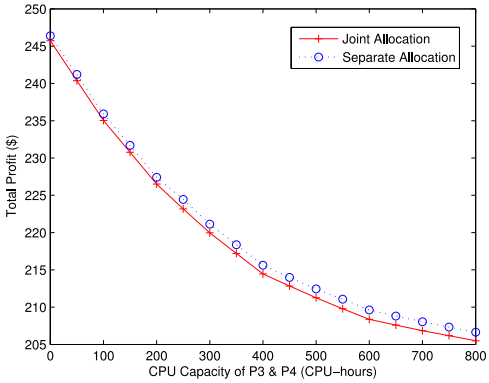


Fig. 6. Comparison of joint and separate provisioning solutions when provider CPU capacity varies.

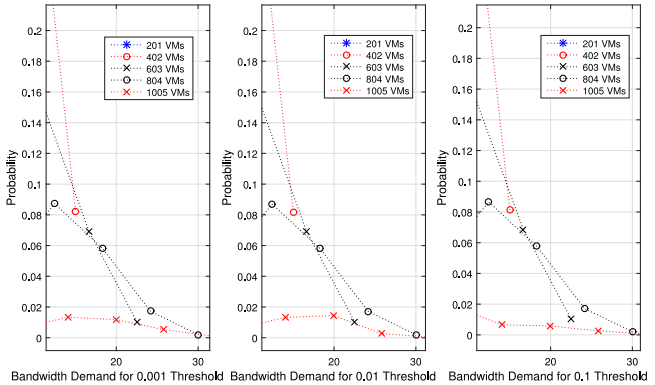


Fig. 8. Magnified view of changed probability distributions to highlight shape changes.

compare the performance of our joint solution with a separate approach. The first test case, given in Fig. 6, limits the CPU capacity on cloud providers P3 and P4. These two providers offer the same pricing plan and are geographically close. The joint solution is calculated and compared to an alternative solution that provisions VM and bandwidth separately. The separate solution uses expected demand values for VM demand (with bandwidth derived from the expected VM demand) as the basis for VM and bandwidth reservation decisions. To compare the solutions, we apply the reservation decisions of each method to the same problem. The resulting cost margin demonstrates the superior solution to the chosen problem. In this case, despite the relatively low costs of bandwidth used in our parameter settings, we find that the joint solution is clearly superior at all levels of CPU capacity. This shows that the reservation choice made in the joint solution allows the repositioning of bandwidth to match the limitations on available CPU. In contrast, the separate solution reserves bandwidth so as to minimize the expected bandwidth costs, which involves reserving bandwidth in the cheaper South region, where providers P3 and P4 are located. This reserved bandwidth becomes essentially wasted, as VM allocation switches to the other providers. The joint solution is flexible enough to reroute bandwidth reservation to avoid the redundant South region routers, and thus can minimize the increase in cost. Increasing the relative cost of bandwidth would highlight this further.

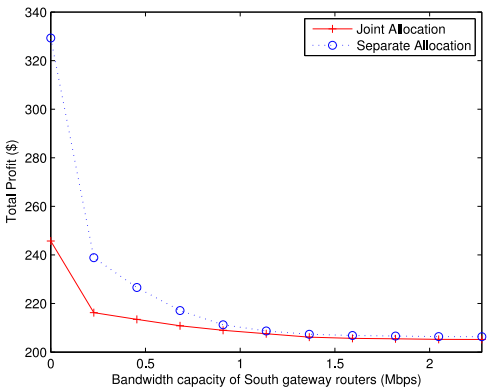


Fig. 7. Comparison of joint and separate provisioning solutions when South gateway router bandwidth capacity varies.

In the second case, we modify the bandwidth capacity of the gateway routers for accessing the southern region, which contains P3 and P4. We choose these routers as P3 and P4 are a popular choice for VM allocation, and limiting bandwidth access highlights the importance of a joint approach. These results are shown in Fig. 7. There is an initial significant drop in cost when the bandwidth capacity exceeds zero, as P3 and P4 become useable. However, particularly at low capacities, the cost difference between the joint and separate solutions are significant. In the separate solution, VMs are reserved from providers without knowledge of the bandwidth available to communicate with those providers. With this lack of information, the cheaper providers P3 and P4 are preferred in advance, but at the time of use, the reserved resources are inaccessible, leading to the difference in price. The additional knowledge of the bandwidth routing allows the circumvention of the bandwidth bottlenecks whilst minimizing the increase in cost. Contrastingly, the joint solution is less affected by the variation in demand, as it is able to rearrange VM reservation to compensate for the bandwidth bottlenecks, which the separate allocation cannot match. An increase in bandwidth demand per VM would highlight this even further.

7.2.3 *Scenario Tree Reduction Performance*

To test the performance of the Scenario Tree Reduction algorithm outlined earlier, we format the resulting data as a new set of probability distributions and use them as the demand input to generate the reservation decisions in our stochastic programs. The algorithm can be run to different probability distance thresholds, in this case: 0.1, 0.01, and 0.001. The smaller threshold values yield probabilities that are closer to the original distribution, but even so, the differences are subtle. In Fig. 8 we highlight the clearest example of difference between the distributions. The subtlety of the differences demonstrates the ability of the algorithm to exclude scenarios that do not contribute significantly to the nature of the scenario tree. The statistical details of the scenario reductions are given in Table 4, along with figures showing the runtime and memory improvements as the tree becomes smaller. The cost rises as the tree is reduced, and thus the user must choose an acceptable tradeoff of cost and speed. However, the algorithm manages to offer scenario reduction that differs only slightly from the optimum, even at a probability distance of 0.1.

TABLE 4
Impact of Scenario Reduction on Optimization Performance

Threshold	T1 Scenarios Removed	T2 Scenarios Removed	Cost	Iterations	Memory
0.001	72	33	205.0588	29,170	66
0.01	108	29	205.0667	25,260	60
0.1	144	25	205.2798	21,437	53

7.2.4 Study of Bandwidth Allocation under Parameter Changes

An interesting factor in bandwidth optimization is studying how the distribution of bandwidth varies under different scenarios and parameter settings. Similarly, it is also interesting to observe what influences bandwidth reservation decisions. In Fig. 9 we observe the realized bandwidth usage across the network under different VM demand scenarios. We highlight the 'gateway' routers that connect geographic regions together. Allocation is particularly high for routers connecting to region 3, which contains two cloud providers, offering the best prices, and therefore is the most influenced by increases in VM demand. In Fig. 10, we alter the pricing on routers $R3$ and $R8$, the gateway between regions 1 and 3, and observe the redistribution of traffic across the other gateway routers to compensate for the increased cost. If the implementation was a separate one, it would be logical to expect more traffic through the North region. However, the joint approach means that much of each user's demand is relocated to their own local providers, as evidenced by the rise in $R1$, which is adjacent to the West cloud provider. This responsiveness to price change is important, and shows that well-placed reservations can compensate for increased costs. It also highlights the importance of a joint approach for finding the optimal cost.

7.2.5 Performance of Alternative Methods under Different Probability Distribution Shapes

Stochastic optimization is a more complex method than some alternatives. Thus, to demonstrate its usefulness, we compare it against alternative methods. The first alternative is an on-demand only method, with no reservation, representing an extreme case of undersubscription. The second alternative is a simple Expected Value Function (EVF)—using the mean of the probability distribution as the reservation amount. This is an important comparison, as stochastic optimization would be needlessly complex if it could not improve on EVF. Finally we take the 0.1 threshold scenario tree reduction solution to compare the reduced tree's performance. We expect stochastic optimization to be most effective. However, to prove that it is the best method, we demonstrate that it works even when the input parameters change. In Fig. 11, we change the variance of the input scenario distribution and compare the performance of the four methods. The patterns are the same with some interesting caveats. The exclusively on-demand approach is largely immune to the change in distribution, as it is purely reactive, and is thus not influenced by probability. EVF performs markedly better with a narrower variance, as the expected value is closer to the possible outcomes. Surprisingly, however, it is noticeably worse than the on-demand only option. This is a consequence of the probability distribution, with the mean allocation resulting in oversubscription. This proves the necessity of a more sophisticated approach. Given the scenario tree reduction results already highlighted, it is not surprising that the reservation based on a reduced tree performs very close to the level of the full size stochastic optimization.

7.2.6 Benefits of Sensitivity Analysis

We test our sensitivity analysis method by examining price sensitivity of VMs. To examine the effects of changing cost on reservation, we define a VM demand for each user on each provider. By using the sensitivity analysis as outlined in Section 6, we can find the precise cost thresholds that trigger reservation changes. This result is given in Fig. 12. Sensitivity analysis ensures we find exact price points without having to make unnecessary measurements. This graph can be reversed to find the upper and lower cost bounds for the whole provisioning range of a variable.

We also test the relevance of sensitivity analysis for bandwidth allocation. Close examination of the bandwidth pricing in Table 3 reveals that the longer contract, $L2$, is

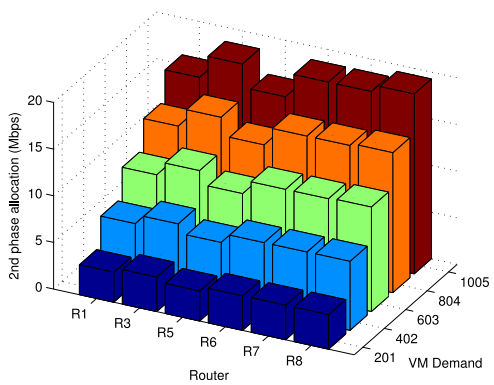


Fig. 9. Effect of varying VM demands on network router bandwidth allocation across the network.

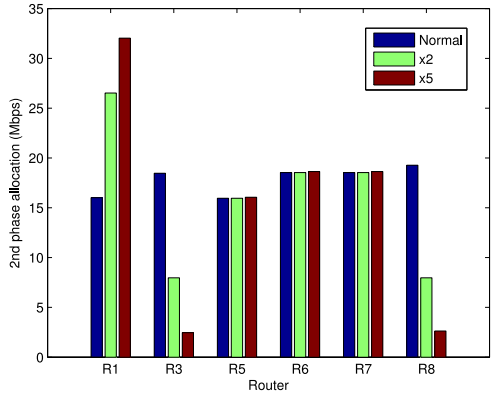


Fig. 10. Effect of price change on bandwidth allocation to gateway router between network regions 3 and 1.

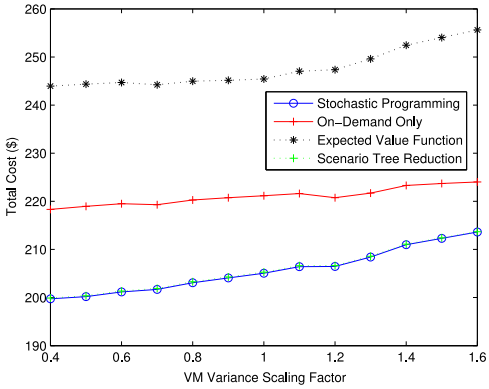


Fig. 11. Total cost of different solution methods under different demand probability distributions.

undesirable. This contract should be more expensive to reserve, but reduce long term costs, thus rewarding forward planning. However, it is apparent that the prices are currently too high, making contract $L1$ more desirable. Examination of the optimization results supports this observation. We can employ sensitivity analysis to find the necessary reduction in cost for $L2$, such that it becomes financially viable again. In Fig. 13, we examine the router prices from each location and contrast the current, poorly chosen, reservation cost for $L2$, with the new, optimal reservation cost found through sensitivity analysis. Thus we can see that sensitivity analysis has useful potential applications not only for the client, but also for the cloud provider, who can ensure that their profits are optimized.

8 Conclusion

In this paper, we have presented a stochastic programming formulation to optimally reserve virtual machines and bandwidth in a cloud computing environment. The formulation makes optimal decisions to reserve resources across multiple time stages despite uncertain demand. The joint VM and bandwidth provisioning is shown to be necessary, as each is interdependent. In a real world application, the problem space can quickly become very large, so we have applied a scenario tree reduction algorithm to find a reasonable heuristic that can be solved efficiently. We have found that whilst reducing the scenario tree increases total cost, the heuristic retains enough accuracy to give a desirable cost-performance tradeoff. Finally we have performed a sensitivity analysis of the stochastic programming problem to examine the solution's tolerance to parameter change. The optimization has sizeable intervals, and we have shown

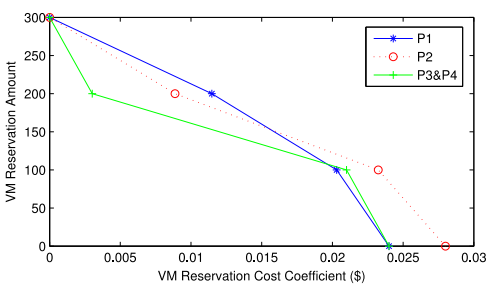


Fig. 12. Precise cost coefficient thresholds triggering VM reservation changes.

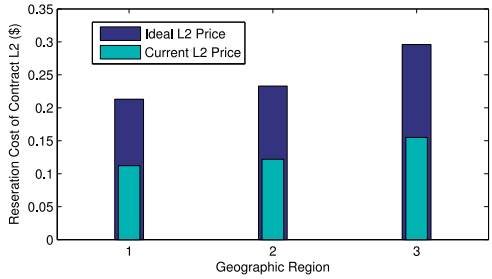


Fig. 13. Full price of contract $L2$ in each network region, with the price needed for the contract to become useful.

that sensitivity analysis is useful for providers in setting system parameters, as well as for clients.

Future work should improve the realism of the problem formulation, by considering factors such as random network delay and VM migration. The model could also be extended to consider inter-VM communication and service composition. Further attempts to improve computational performance should be considered, for example by applying a distributed approach. The sensitivity analysis results show promise, but can be extended to further examine the interaction between system elements, such as the significance of constraints, and the priority of variables.

Acknowledgments

This work was supported by Singapore MOE Tier 1 (RG18/13 and RG33/12) and MOE Tier 2 (MOE2014-T2-2-015 ARC 4/15).

References

[1] (2013, Aug.). Amazon EC2. [Online]. Available: http://aws.amazon.com/ec2/pricing/

[2] A. Shapiro, D. Dentcheva, and A. Ruszcynski. (2009). *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics Philadelphia. [Online]. Available: http://www2.isye.gatech.edu/people/faculty/Alex_Shapiro/SPbook.pdf

[3] (2014). AWS Case Study: Netflix [Online]. Available: http://aws.amazon.com/solutions/case-studies/netflix/

[4] (2012, Apr.). Software-Defined Networking: The New Norm for Networks [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[5] (2014). Enter the Andromeda Zone-Google Cloud Platforms Latest Networking Stack [Online]. Available: http://googlecloudplatform.blogspot.sg/2014/04/enter-andromeda-zone-google-cloud-platforms-latest-networking-stack.html

[6] (2014). Mirage OS A Cloud Operating System [Online]. Available: http://www.xenproject.org/developers/teams/mirage-os.html

[7] (2015). OpenFlow [Online]. Available: https://www.opennetworking.org/ja/sdn-resources-ja/onf-specifications/openflow

[8] H. N. Van, F. Tran, and J.-M. Menaud, "Autonomic virtual resource management for service hosting platforms," in *Proc. Softw. Eng. Challenges Cloud Comput.*, May 2009, pp. 1–8.

[9] H. N. Van, F. Tran, and J.-M. Menaud, "SLA-aware virtual resource management for cloud infrastructures," in *Proc. 9th IEEE Int. Conf. Comput. Inf. Technol.*, Oct 2009, vol. 1, pp. 357–362.

[10] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing SLA violations," in *Proc. 10th IFIP/IEEE Int. Symp. Integrated Netw. Manage.*, May 2007, pp. 119–128.

[11] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An energy-saving application live placement approach for cloud computing environments," in *Proc. IEEE Int. Conf. Cloud Comput.*, Sep. 2009, pp. 17–24.

[12] R. Cohen, L. Lewin-Eytan, J. Naor, and D. Raz, "Almost optimal virtual machine placement for traffic intense data centers," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 355–359.
[13] X. Xu, K. Yao, S. Wang, and X. Zhou, "A VM migration and service network bandwidth analysis model in IaaS," in *Proc. 2nd Int. Conf. Consum. Electron., Commun. Netw.*, Apr. 2012, pp. 123–125.
[14] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments Via lookahead control," in *Proc. Int. Conf. Autonomic Comput.*, Jun. 2008, pp. 3–12.
[15] K. Zamanifar, N. Nasri, and M. Nadimi-Shahraki, "Data-aware virtual machine placement and rate allocation in cloud environment," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Jan. 2012, pp. 357–360.
[16] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[17] M. Alicherry and T. Lakshman, "Network aware resource allocation in distributed clouds," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 963–971.
[18] V. Justafort and S. Pierre, "Performance-aware virtual machine allocation approach in an intercloud environment," in *Proc. 25th IEEE Canadiam Conf. Elect. Comput. Eng.*, Apr. 2012, pp. 1–4.
[19] L. Yu and H. Shen, "Bandwidth guarantee under demand uncertainty in multi-tenant clouds," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, Jun. 2014, pp. 258–267.
[20] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "SecondNet: A data center network virtualization architecture with bandwidth guarantees," in *Proc. 6th Int. Conf.*, 2010, pp. 15:1–15:12.
[21] J. Zhu, D. Li, J. Wu, H. Liu, Y. Zhang, and J. Zhang, "Towards bandwidth guarantee in multi-tenancy cloud computing networks," in *Proc. 20th Int. Conf. Netw. Protocols*, Oct. 2012, pp. 1–10.
[22] A. Dalvandi, M. Gurusamy, and K. C. Chua, "Time-Aware VM-placement and routing with bandwidth guarantees in green cloud data centers," in *Proc. IEEE 15th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2013, vol. 1, pp. 212–217.
[23] H. Xu and B. Li, "Joint request mapping and response routing for geo-gistributed cloud services," in *Proc. IEEE INFOCOM*, 2013, pp. 854–862.
[24] J. Cao, W. Zhang, and W. Tan, "Dynamic control of data streaming and processing in a virtualized environment," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 365–376, Apr. 2012.
[25] R.-H. Hwang, C.-N. Lee, Y.-R. Chen, and D.-J. Zhang-Jian, "Cost optimization of elasticity cloud resource subscription policy," *IEEE Trans. Serv. Comput.*, vol. 7, no. 4, pp. 561–574, Oct. 2014.
[26] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 2, pp. 164–177, Apr. 2012.
[27] D. Breitgand and A. Epstein, "Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2861–2865.
[28] D. Divakaran and M. Gurusamy, "Towards flexible guarantees in clouds: Adaptive bandwidth allocation and pricing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1754–1764, Jun. 2015.
[29] D. Niu, C. Feng, and B. Li, "A theory of cloud bandwidth pricing for video-on-demand providers," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 711–719.
[30] J. Chase, R. Kaewpuang, W. Yonggang, and D. Niyato, "Joint virtual machine and bandwidth allocation in software defined network (SDN) and cloud computing environments," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 2969–2974.
[31] C. Villani, "The Wasserstein distances," in *Optimal Transport* (series Grundlehren der mathematischen Wissenschaften). Berlin, Germany, Springer, 2009, vol. 338, pp. 93–111.
[32] H. Heitsch and W. Rmisch, "Scenario reduction algorithms in stochastic programming," *Comput. Optim. Appl.*, vol. 24, no. 2-3, pp. 187–206, 2003.
[33] B. Jansen, J. de Jong, C. Roos, and T. Terlaky. (1997). Sensitivity analysis in linear programming: Just be careful! *Eur. J. Oper. Res.* [Online]. *101(1)*, pp. 15–28. Available: http://www.sciencedirect.com/science/article/pii/S0377221796001725
[34] (2015). GAMS Solvers [Online]. Available: http://www.gams.com/dd/docs/solvers/allsolvers.html
[35] J. Wilkes and C. Reiss. (2014). Google Cluster Data [Online]. Available: https://code.google.com/p/googleclusterdata/wiki/ClusterData2011_2
[36] (1993). LBL-CONN-7 [Online]. Available: ftp://ita.ee.lbl.gov/html/contrib/LBL-CONN-7.html
[37] (2014). How Can I Control How Much Data Netflix Uses? [Online]. Available: https://help.netflix.com/en/node/87
[38] (2014). Ookla Net Index [Online]. Available: http://www.netindex.com/

Jonathan Chase received the MEng degree in the Department of Computer Science from the University of Warwick, United Kingdom in 2011. He is currently working toward the PhD degree in the School of Computer Engineering, at the Nanyang Technological University, Singapore. His research interests include optimization in cloud computing and mobile application migration.

Dusit Niyato received the PhD degree in electrical and computer engineering from the University of Manitoba, Canada in 2008. He is currently an associate professor in the School of Computer Engineering, at the Nanyang Technological University, Singapore. His research interests are in the area of the optimization of wireless communication and mobile cloud computing, smart grid systems, and green radio communications. He is a member of the IEEE.