A Survey of Profit Optimization Techniques for Cloud Providers

PEIJIN CONG, GUO XU, and TONGQUAN WEI, East China Normal University, China
KEQIN LI, State University of New York, USA

As the demand for computing resources grows, cloud computing becomes more and more popular as a pay-as-you-go model, in which the computing resources and services are provided to cloud users efficiently. For cloud providers, the typical goal is to maximize their profits. However, maximizing profits in a highly competitive cloud market is a huge challenge for cloud providers. In this article, a survey of profit optimization techniques is proposed to increase cloud provider profitability through service quality improvement, service pricing, energy consumption reduction, and virtual network function (VNF) deployment. The strategy of improving user service quality is discussed first, followed by the pricing strategy for cloud resources to maximize revenue. Then, this article summarizes the techniques for cloud data centers to reduce server power consumption. Finally, various heuristic algorithms for VNF deployment in the cloud are further described to reduce the cost of cloud providers while maintaining performance. We classify research works based on components of profit and methods used to demonstrate similarities and differences in these studies. We hope this survey will provide researchers with insights into cloud profit optimization techniques.

CCS Concepts: • **Cloud** → Cloud computing;

Additional Key Words and Phrases: Cloud computing, profit maximization, quality of service (QoS), cloud service pricing, energy consumption, virtual network function (VNF) deployment

ACM Reference format:
Peijin Cong, Guo Xu, Tongquan Wei, and Keqin Li. 2020. A Survey of Profit Optimization Techniques for Cloud Providers. *ACM Comput. Surv.* 53, 2, Article 26 (March 2020), 35 pages.
https://doi.org/10.1145/3376917

1 INTRODUCTION

Cloud providers (CPs) virtualize hardware and software resources into a unified resource pool and provide users with needed resources on demand through the internet. In particular, these resources are provided to users in three different forms of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). As a business model, cloud providers aim to improve their profits as much as possible. Meanwhile, cloud customers expect to

This work was supported in part by National Key Research and Development Program of China under Grant 2018YFB2101300, in part by ECNU XingFuZhiHua Program, and in part by National Natural Science Foundation of China under Grant 61872147.

Authors' addresses: P. Cong, G. Xu, and T. Wei (corresponding author), East China Normal University, School of Computer Science and Technology, 3663 Zhongshan North Road, Shanghai, China; emails: 52184506011@stu.ecnu.edu.cn, xuguo0515@vip.qq.com, tqwei@cs.ecnu.edu.cn; K. Li, State University of New York, Department of Computer Science, 1 Hawk Drive, New Paltz, New York; email: lik@newpaltz.edu.

0360-0300/2020/03-ART26 $15.00
https://doi.org/10.1145/3376917

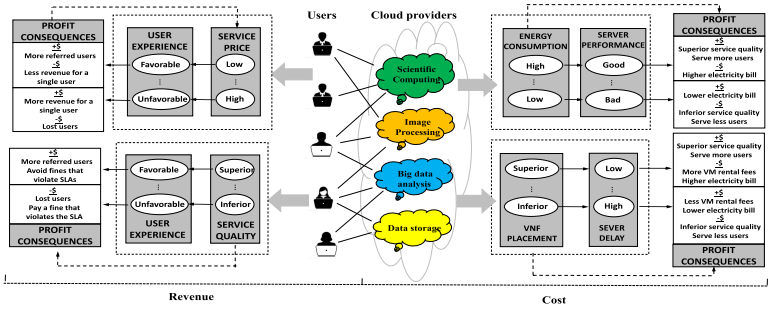

Fig. 1. A bird's eye view of the central idea of this article.

gain a satisfactory service from the cloud providers. However, as more cloud providers are available to the users, maximizing profits has become a big challenge for the cloud providers. Like all other businesses, the profit of a cloud provider is usually related to two components, that is, revenue and expenditure.

The revenue of a cloud provider is determined by not only quality of service (QoS) but also service price. Guaranteeing QoS for users can free the cloud provider from penalties for violating service level agreement (SLA), thereby increasing the cloud provider's income. As for service price, appropriate pricing strategies can not only maximize cloud resource value but also attract more users in the highly competitive cloud market, thus further increasing the cloud provider's income.

The expenditure of a cloud provider is mainly composed of the server rental fee from IaaS providers and the electricity fee incurred by server operations. In particular, for cloud providers that provide virtual network function (VNF), the placement cost of VNF instances should also be considered. Reducing electricity bills paid by cloud providers to power plants can reduce cost and increase the cloud providers' profit. Also, reducing placement cost of VNF instances can not only minimize resource consumption but also increase the number of users that cloud providers could serve, thus increasing the cloud provider's profit.

Contribution and article organization: In this article, a survey of profit optimization techniques for cloud providers is presented. We review the research on profit optimization for cloud providers from perspectives of service quality, service price, server power consumption, and VNF instance placement. Figure 1 shows the bird's eye view of the central idea of this article.

We first propose four factors including service quality, service price, server power consumption, and VNF instance placement that affect the profits of cloud providers, and discuss some techniques to improve profits from four perspectives (Section 3). Then, we review the techniques of improving service quality to increase revenue (Section 4), and summarize the work of adjusting service price to maximize the value of cloud resources (Section 5). Afterward, we discuss the techniques that reduce the costs of electricity (Section 6) and VNF instance placement (Section 7). Finally, this article is summarized in Section 8.

Scope of the article: The scope of profit optimization for cloud providers covers a broad range of techniques. We limit the scope of this article in the following way for ease of presentation. We concentrate on works that improve the profitability of cloud providers rather than works that focus primarily on reducing the cost of cloud users [93]. Since there is no model that relates security to profitability, in this study, we focus on the impact of performance (e.g., service response time) on cloud providers rather than the impact of security on cloud providers [41, 90].

2 RELATED WORK

Many surveys on cloud computing have been published during the last decade. These surveys review a large number of research works from perspectives of cloud pricing models [44, 100],

resource allocation and provisioning techniques [24, 107], task scheduling algorithms [62, 83], cloud security and privacy issues [40, 42, 88], and so on. In this section, we only focus on the survey works related to cloud computing economics. We summarize and compare the state-of-the-art surveys in the following.

For cloud service providers, profitability and revenue maximization are the most important goals pursued. Some survey works have been done for profit improvement in terms of cloud pricing models in the literature [5, 30, 44, 100]. For example, in [100], the authors provide a systematic review of cloud pricing in an interdisciplinary approach, in which various pricing models are analyzed from aspects such as cloud technologies, microeconomics, operations research, and value theory. Kumar et al. [44] investigate Amazon spot instances and provide an exhaustive survey of spot pricing in cloud ecosystem. These works have provided comprehensive analysis of existing cloud pricing mechanisms for revenue improvement in the current cloud market. However, energy-efficient cloud resource management also plays a very important role in cloud profit optimization, which is not discussed in the above reviews. There are numerous works that study energy-efficient resource management, for example, resource allocation, provisioning, scheduling, placement, and migration in cloud computing [24, 35, 53, 55, 60, 87, 107, 109]. The authors in [87, 107] summarize and make an assay of state-of-the-art resource scheduling approaches in cloud computing, such as real-time, adaptive dynamic, large-scale, multi-objective, and distributed and parallel scheduling, some of which are classified for energy conservation. In terms of cloud resource scheduling, these surveys have provided very comprehensive reviews, whereas in terms of cloud economy, their research is one-sided. Luong et al. [58] provide a novel comprehensive literature review for resource management in the context of cloud network based on economics and pricing models for sustainable cloud economic advantage achievement. This survey discusses the effects of various economic and pricing models on resource management. Nevertheless, it is more biased toward the cloud network economics rather than the cloud provider economics.

There are also some other surveys that research into issues surrounding Service Level Agreement (SLA) [98], Quality of Service (QoS) [25], trust [34], and brokerage [11, 21] in cloud computing. Obviously, this research only analyzes the factors affecting the cloud economy from a single perspective. Thus, there is a lack of comprehensive summarization and analysis of the factors related to cloud economics. In this article, we review state-of-the-art research to investigate various factors (e.g., service quality, pricing mechanisms, and resource management) that impact the profitability of cloud providers, provide comprehensive factor-based taxonomy, and make comparisons among these works from multiple aspects.

3 BACKGROUND AND MOTIVATION

Cloud computing has become increasingly popular by providing diverse resources and services to users in an effective and efficient way. In recent years, the number of cloud providers available to users has also increased dramatically. How to gain more profits in the competitive cloud markets is especially crucial for providers. Like other business, the profit of a cloud provider is usually composed of two parts, i.e., revenue and cost [69]. On one hand, the revenue is related to market demand, service pricing strategies, service quality, customer satisfaction, and so on. Service quality and customer satisfaction influence the market demand with respect to cloud services, which indirectly affects cloud providers' revenue, while service pricing strategies have a direct effect on cloud providers' revenue. On the other hand, the cost is mainly related to the energy consumption of the cloud service platform and the expense of platform resource deployment. Specifically, the user service demands grow so fast that more servers are required to guarantee QoS, thus leading to a drastic increase in energy consumption. Moreover, resource over-provisioning or under-provisioning could further cause wasted resources or lower service quality, thus increasing

resource deployment cost or decreasing revenue, leading to a reduction of profitability. Thus, in the following sections, we will investigate the impacts of service quality, service price, infrastructure energy consumption, and VNF placement on the profitability of cloud providers, respectively.

3.1 The Impact of Service Quality on Revenue

In the context of cloud computing, QoS is usually related to the predefined attributes such as response time, reliability, and the remedies for performance failures [67]. When the performance requirements are not met, cloud providers need to pay fines to cloud users due to the violation of SLA. Taking Alibaba Cloud [1] as an example, in its latest version of the SLA for Elastic Compute Service (ECS), for a single ECS instance, if the service availability is below a certain value (i.e., 99.975%), it needs to compensate the users for the corresponding voucher amount based on the service availability level [3]. Thus, we can see that if users do not get a timely response from the cloud provider, the cloud provider will be penalized for low QoS and the degree of penalty depends on the terms of SLA, which not only affects the current revenue of the cloud provider, but also has a negative impact on the market share of the cloud provider in the future. That is to say, QoS impacts the customer satisfaction with the service, and customer satisfaction further influences the future market demand of the service, thus affecting the future market share of the cloud provider. Thus, cloud providers need to increase their revenue by guaranteeing or improving QoS.

Extensive exploration has been conducted to improve QoS. A naive solution to improve QoS is over-provisioning of available resources to meet the peak user demands. However, this solution may lead to low resource utilization in the case of low user demands. Thus, a more effective solution is to guarantee QoS when the number of users is limited during peak workloads [32, 65]. The spare resources are provided to second-class users with discounted prices at the cost of low QoS. Nonetheless, this method results in unpredictable request delays, rejections, terminations, and price fluctuations [13].

Federated clouds of multiple providers improve QoS for users by sharing unused virtual machines (VMs) to the cloud federation during periods of low demands and borrowing VMs from the cloud federation during peak periods [80, 81]. Several strategies for resources sharing in cloud federation are studied in [31, 47]. These strategies help the cloud service providers in the cloud federation determine the resources capacity (i.e., computing capacity) and the timing of computing resource sharing and borrowing. However, these strategies increase the energy consumption of each cloud provider due to VM migration. Moreover, the information related to historical and future interactions among these cloud providers is not considered in these solutions when performing decision-making sharing.

3.2 The Impact of Service Price on Revenue

Service providers hope to obtain high profits with high service prices, but this will reduce user satisfaction and lead to a decline in demands in the future. While low service prices can improve user satisfaction and future demands, this may result in a loss of profits for cloud providers. Thus, service pricing can greatly affect the revenues of cloud providers.

Ghamkhari and Mohsenian-Rad [29] adopt a static pricing strategy and sell all services at a unified price. However, unified pricing is incompatible with service differentiation. Zhang et al. [111] and Amazon Incorpored [2] use a pricing strategy to periodically change service prices. However, users' demands are dynamic and sudden, which leads to changing service prices periodically cannot reflect the drastic fluctuations of supply and demand in time.

Cloud resources can also be priced by using auctions [94, 96]. Specifically, in [94], the authors formulate VM pricing as a multi-unit combined auction model and propose greedy allocation mechanisms to optimize the sum of declared valuations. Wang et al. [96] propose a dynamic

auction-based pricing strategy that determines the amount of auction resources and resource pricing for maximizing revenue based on the dynamic needs of users. However, some users could influence auction results and gain unfair benefits by malicious bidding or hiding their preferences for resources. These behaviors will finally destroy auction experience of other normal users, reduce auction efficiency, and hinder the participation of users.

3.3 The Impact of Server Energy Consumption on Cost

Service providers have to pay for the electricity consumed by servers that process user requests. On one hand, service providers need to allocate enough numbers of servers to meet QoS. On the other hand, they attempt to minimize the energy consumed by servers to increase profits. The challenge of optimizing energy consumption puts service providers in a dilemma since the number of user requests and service requirements grow so fast that more servers are required to guarantee QoS, leading to a drastic increase in electricity costs.

Numerous methods have been proposed to optimize server energy consumption, among which resource sharing is especially important for cloud providers. Lee et al. [48] present a resource sharing model for cloud providers through processor sharing and two profit-driven scheduling algorithms. However, the scheduling algorithms lead to an increase in SLA violations, thus, degrade QoS. Good VM placement strategies can improve resource utilization and reduce traffic costs in cloud [33, 39]. However, these works assume a known traffic between VMs, which is somewhat impractical.

3.4 The Impact of VNF Instance Deployment on Cost

Network function virtualization (NFV) dynamically configures virtualize network function (VNF) instances to provide users with fast and inexpensive network functions by using hardware resources. The deployment cost of VNF instances is related to the number of VMs running VNF. Thus, cloud providers need to minimize the required VMs while meeting users' needs. In addition, launching a new VNF instance will transfer a VM image to a new server. Frequent movement of VNF instances will result in additional transmission costs, increasing the cost of cloud providers. Thus, a good VNF instance placement policy can not only increase the utilization of resources but also greatly reduce the deployment costs, thus bringing more profits to cloud providers.

Addis et al. [4] first design a new NFV network model, then define the VNF placement optimization problem based on the model and formulated it as a mixed integer linear programming (ILP) problem. Bari et al. [6] decide the number and location of VMs for optimal VNF placement by using ILP. The optimization problem is solved by using a standard ILP solver. Mehraghdam et al. [66] propose a model to describe VNF requests, and design a mixed integer quadratic constrained program for VNF placement based on the model. These methods can reduce VNF deployment cost; however, the linear programming–based method requires a lot of computing resources and time to obtain optimal results, which may offset benefits obtained by using the linear programming–based techniques. These methods focus on offline VNF deployment and ignore fluctuations in user traffic.

4 INCREASE PROFIT BY IMPROVING SERVICE QUALITY

In this section, the research on improving QoS to maximize providers' profit is summarized. Table 1 summarizes the references with regard to the improvement of QoS. These works are classified based on the methods they use, as discussed below: (1) game theory approach (Section 4.1), (2) double queuing (Section 4.2), (3) resource reservation (Section 4.3), (4) resource scheduling (Section 4.4), (5) resource sharing (Section 4.5), and (6) bandwidth guarantee (Section 4.6).

Some common SLA models are shown in Figure 2. We can see from Figure 2(a) that when the submitted request can be executed within the specified deadline, the request is normally charged.

Table 1. Classification of Methods of Reducing Service Response Time

Classification	References
Using game theory to collaborate	[64], [85]
Using double queue	[69]
Using resource reservation	[52], [75], [10]
Using resource scheduling	[38], [82], [19], [46], [61]
Using resource sharing	[46], [48], [85]
Providing bandwidth guarantee	[106], [37], [56], [50]

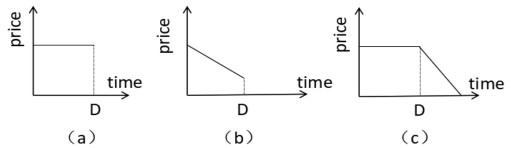

Fig. 2. Some commonly used SLA models.

Otherwise, the request will be free for penalty due to violation of SLA. From Figure 2(b), we can observe that as the waiting time increases, the charge will continue to decrease until the service is free. Figure 2(c) adopts a two-stage charge function. The first step is similar to that of Figure 2(a), that is, a service request is normally charged if it is processed on time. In the second step, the charge will continue to decrease along with the increase in waiting time until the service is free.

4.1 Using Game Theory Method

Lena et al. [64] focus on the problem that cloud providers may not have enough resources to fulfill the requirements of data-intensive applications. To tackle this problem, the authors propose a cooperation scheme for cloud providers, which increases cloud providers' dynamic resource expansion capability to meet demands of users. The proposed scheme allows cloud service providers to collaborate on resource expansion and dynamically form federations to provide users with the requested resources. The cloud federation formation scheme should achieve two major goals of fairness and stability. In order to achieve fairness, the method uses the estimated normalized Banzhaf value to calculate each cloud provider's revenue. Further, to achieve stability, the method guarantees that the cloud provider's revenue in the current federation is not less than that of other federations. Experimental results show that cloud federation is stable and can bring high profits to cloud providers. However, the federation formation problem does not take data privacy into account, and the impact of cloud providers' policies on the federation formation process is not studied in this work.

High workloads may result in increased latency and degraded QoS, while low workloads may lead to resource waste. Samaan [85] proposes an infrastructure capacity sharing model to deal with the problem of uncertainty workloads, as shown in Figure 3. The model optimizes cloud providers' profits by selling unused capacity to other cloud providers. They first introduce a series of cloud provider capacity sharing strategies based on multi-stage games. These strategies threaten cloud providers who refuse to share unused VMs by eliminating their future hosting of VMs of other cloud providers. Hence, individual cloud providers achieve less revenue compared to those in the federation. Then, they develop a dynamic program to obtain VM sharing decisions of each cloud provider. Simulation results show that their method can effectively increase the profits for cloud providers and VM utilization. However, the capacity sharing model of this work does not achieve

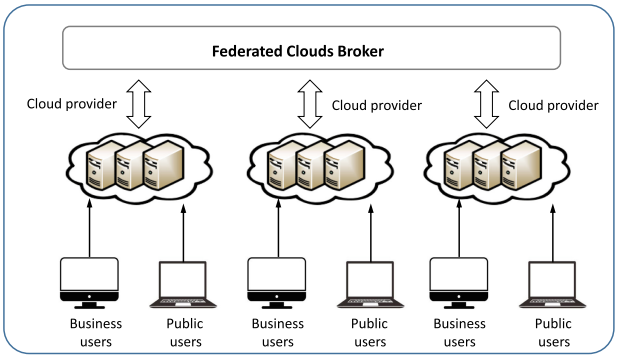

Fig. 3. The infrastructure capacity sharing model of the federated clouds [85].

full decentralization. The model can be further extended by adding in other constraints (e.g., energy consumption) for each cloud provider to solve different issues.

4.2 Using Double Queue Method

In order to optimize QoS and increase cloud providers' profit, Mei et al. [69] design a novel resource leases mechanism which combines short-term and long-term leases. In the proposed scheme, users' requests are first allocated to the waiting queue of the long-term rented server according to arrival time. Then, if a request in the waiting queue reaches its deadline, the cloud provider rents a temporary server from the infrastructure provider to process the request. The model can reduce service rejection rate and improve QoS. However, the cloud provider's cost will also increase due to the high temporary server rental price. Thus, they consider the tradeoff between the cost of rental and the revenue brought by the improvement of QoS, and establish the optimal configuration of service providers to maximize profits. To this end, the authors propose two optimal solutions: ideal solution and actual solution. The ideal solution assumes that the size and speed of servers are continuous while the actual solution assumes that the size and speed of servers are limited and discrete. The profit maximization problem studied in this work is carried out in a relatively simple homogeneous cloud environment, which is unrealistic in today's complex cloud computing environment. Profit optimization with double renting mechanisms in a heterogeneous cloud service environment is an interesting topic and needs to be further explored. Experiment results demonstrate that their novel renting mechanism performs better as compared with the single leases scheme in terms of both QoS and profit.

4.3 Using Resource Reservation Strategy

Liu et al. [52] minimize the payment costs of cloud providers under the constraints of customer service level objective (SLO) guarantees. To this end, the authors propose a SLO guaranteed economical cloud storage services (ES3) scheme. ES3 exploits three methods to minimize payment costs and guarantee SLO: a request allocation method, a request distribution adjustment method based on genetic algorithm, and a dynamic request redirection method. The request allocation method allocates user requests to data centers and utilizes all pricing strategies to determine resource reservations on data centers, as shown in Figure 4. The request distribution adjustment method based on a genetic algorithm maximizes the reserved revenue by reducing the data Get/Put rate difference in each data center. The dynamic request redirection method further reduces payments by dynamically redirecting data requests from reserved overused data centers to reserved under-utilized data centers. Experiments on supercomputing clusters show that ES3 achieves

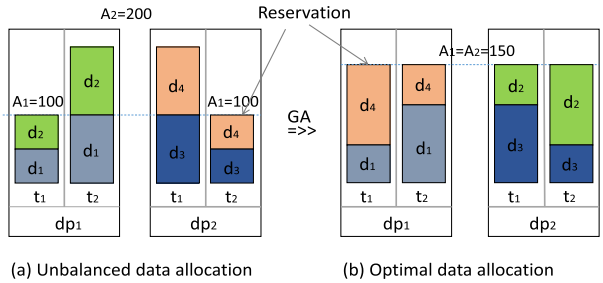

Fig. 4. Unbalanced and optimal data allocation schemes [52].

superior performance in terms of providing SLO guarantees and minimizing costs as compared to the benchmark scheme. However, this work does not take the dependency between data blocks into consideration for data allocation, which can further improve the speed of data retrieval.

In order to satisfy SLA and minimize total reservation cost, Qiu et al. [75] design a demand distribution system for cloud providers. The system focuses on the problem of how to reserve servers and assign service demands to these servers, and addresses it by two steps: demand forecasting and demand allocation. First, the system dynamically predicts the demands of different types of resources according to the historical data without having to assume a seasonal period. Subsequently, based on the prediction results, the authors formulate a probabilistic demand allocation problem and use the decentralized approach to solve it. In the demand forecasting stage, this work ignores the correlation between different tenants and different resource type requirements. This correlation can be utilized to better predict user resource demand. Extensive simulations demonstrate that their method can effectively reduce server reservation costs while guaranteeing QoS for users.

Reserving resources for cloud users leads to low resource utilization. Carvalho et al. [10] address this problem by re-selling users' unused resources; however, this will increase the risk of violating SLA. Thus, the authors first propose a confidence levels–based prediction method to forecast the number of unused resources that users will retain each month. Based on the predictions, they control the risk of SLA violations and trade with increased risk to provide more resources. Extensive experiments on clusters at Google show that the proposed approach can increase the profitability of cloud providers by 20%–60%. This work has a bias toward the prediction of the number of servers in an inactive state. Considering that idle machines may also act as storage servers, exploiting a hybrid approach to shutting down some machines and reselling idle resources from other machines is a better choice.

4.4 Performing Resource Scheduling

In order to meet users' SLA requirements, Rodriguez and Buyya [82] present a particle swarm optimization (PSO)–based scientific workflow scheme for cloud providers. The modeling of PSO issues requires solving two key issues: how to encode the problem and how to measure the "goodness" of a particle. For defining the encoding of the problem, they define the particle as a workflow and its tasks. Thus, the dimension of particles can be represented by the number of tasks in the workflow, and the range of particles allowed to move depends on the number of resources available running tasks. To measure the "goodness" of particles, they define particles' fitness function as the total execution cost associated with particles' positions. The evaluations using CloudSim and other well-known scientific workflows show that their method can produce schedules with lower execution costs compared with other algorithms when application deadlines are met. However, this work does not consider the data transfer cost between data centers in their resource model, which

can enable VMs to be deployed on different areas. Further, the current meta-heuristic optimization algorithm does not ensure enough VM memory for a task to complete, leaving improvement space for workflow scheduling algorithm design.

Huang et al. [38] investigate the problem of heterogeneous cloud resource allocation for cloud operational benefit optimization. The method gets great benefits by prioritizing key jobs at a specific time rather than jobs that are insensitive to completion. Specifically, the resource allocation problem is first formulated as an integer programming problem. Subsequently, by using the single-mode structure of the solution space, the integer programming problem is reconstructed into a linear programming problem, which could be efficiently and optimally solved. Finally, the authors implement the proposed method as the resource scheduler of the widely used Hadoop data processing framework. Extensive experiments demonstrate that the proposed method can effectively improve operational benefits in the cloud by maximizing the worst case utility and improving the subsequent worst case utility.

Du and De Veciana [19] propose an efficient scheduling scheme to meet users' QoS for real-time application workloads. For users with variable workloads, they prioritize the tasks based on the users' maximum QoS deficit for each period, and then handle the tasks from the highest to the lowest priority. For users with deterministic workloads, they prioritize the tasks based on the local remaining execution time, and then process the tasks that do not exceed the maximum workload of servers according to the priority. In particular, this work focuses on a single cloud computing system consisting of multiple resources but does not contain multiple types of resources. Thus, this computing system can be further extended by investigating different types of resources. Simulation results show that their proposed method can not only satisfy users' QoS requirements but also save substantial resources as compared to reservation-based designs.

Aiming at optimizing users' tail delay in a cloud content delivery network (CDN) while meeting cloud providers' cost constraints, Lai et al. [46] propose a request scheduling mechanism called TailCutter. TailCutter can reduce user delays by assigning user requests to different data centers based on workloads of the data centers. More specifically, TailCutter first regularly measures the delay distribution and workloads of different IP prefixes in each cloud data center. Subsequently, TailCutter assigns user requests and determines a specific download method for each user based on delay distribution, different cloud pricing strategies, and cost constraints. Finally, users receive the scheduling results and download the needed replica accordingly. Extensive experiments show that TailCutter can effectively reduce 68% delay of users without exceeding budgets of application providers.

4.5 Performing Resource Sharing

Cloud providers can improve resource utilization and profits by using resource sharing. However, resource sharing has negative impacts on QoS. Thus, Lee et al. [48] schedule the service requests by dynamically creating service instances, addressing the conflict between resource sharing and QoS. Specifically, they first design a pricing model on the basis of a processor sharing mechanism. In particular, the pricing model is defined as a time-varying utility function, where the charge for a request is related to its response time that is modeled based on the processor sharing mechanism. Subsequently, based on the proposed pricing model, the authors propose two profit-driven scheduling algorithms taking into account the priority constraints and deadline constraints of service requests. One algorithm maximizes profits by considering both the current service and other services handled by the same service instance. The other algorithm maximizes instance utilization and minimizes the cost of server renting from cloud providers. Experimental results show that their algorithms could optimize the resource utilization and the cloud provider's profit effectively.

In this work, the dynamic creation of service instances and the dependencies between tasks can be further taken into account to accommodate more general scenarios.

In order to relieve the bandwidth burden of cloud providers, Zhao et al. [115] design an online procurement auction scheme to share unused storage and network resources among users. In this scheme, cloud users are motivated to submit bid offers to cloud computing providers. The bid includes the number of unused resources that users plan to share, the available time of resources, and the expected rewards. After receiving the bid, the cloud provider will determine the amount of purchased resources to reduce the network traffic and storage burden of the data center server. Then, they extend Myerson's optimal auction design framework using marginal price–based allocation to guarantee the truthfulness of the proposed mechanism. Extensive simulation experiments show that the proposed online procurement auction can effectively reduce cloud provider's network traffic and storage burden and improve QoS. Moreover, the resource pool cost of the proposed online method is lower as compared with that of the offline Vickrey-Clarke-Groves auctions in most cases. However, this work does not consider the issue of time decoupling, which may lead to a suboptimal solution of their proposed scheme.

Marandi et al. [61] present a system that provides performance guarantee and efficient utilization of cloud resource for cloud providers. For performance guarantee, the authors first design an application program interface to allow users to indicate their performance requirements. Then, a novel distributed controller is proposed to coordinate servers' resource allocation and to fairly allocate unused capacity. For the efficient utilization of cloud resource, the authors exploit shared computing to reallocate reserved but underutilized resources to other users who need higher performance. A novel placement algorithm is used to consolidate users efficiently on a shared set of servers. Experiment results show that the distributed controller can achieve 95% efficiency of the centralized techniques while increasing speed by about five times, and the placement algorithm can effectively allocate more than 98% of reserved but underutilized resources.

4.6 Providing Bandwidth Guarantee

Yu and Cai [106] design a scheme to guarantee network bandwidth as the virtual network expands. First, new VMs are allocated for cluster scaling by traversing the network topology step by step in a bottom-up manner. Then, they propose an algorithm to further reduce communication latency and network overhead by adjusting the new VM as close as possible to the pre-existing VM. Considering that VM cluster may not be able to scale without changing the original VM location, the authors further exploit VM migration and develop a best algorithm to allocate the scaled VM cluster for VM migration cost minimization. The algorithm transforms the VM migration problem to the minimum weight perfect matching problem and finds the minimum number of hops for VM migration to reduce service downtime and network overheads. In particular, this scheme can be improved by utilizing additional resources to recover the service quickly in the event of a failure, enhancing performance guarantees. Extensive simulations show that their method can greatly reduce service rejection rate and improve the scalability with minimal migration costs.

Hu et al. [37] propose Trinity to guarantee bandwidth while providing work conservation, as shown in Figure 5. To achieve this goal, Trinity distinguishes between short and long traffic and prioritizes them in the network. Through this distinction and prioritization, Trinity eliminates the tradeoff between bandwidth guarantee and work protection, and actively designs work protection without affecting bandwidth guarantees. Experiments show that Trinity can reduce the average completion time of short and long traffic by 82% and 78% while providing bandwidth guarantee, respectively.

To achieve both bandwidth guarantee and work protection, the authors in [56] propose a network-based solution called QShare, as shown in Figure 6, which includes resources placement

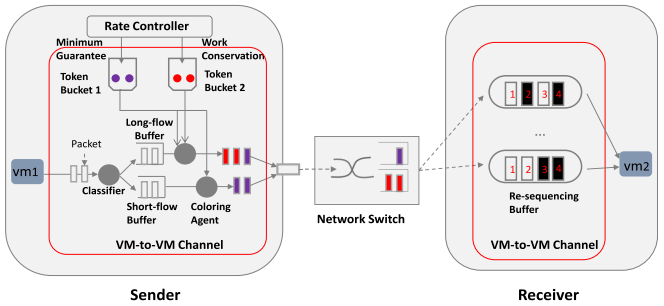

Fig. 5. Trinity system framework [37].

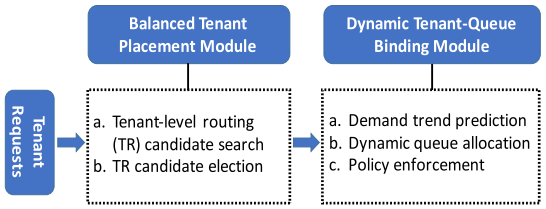

Fig. 6. The architecture of QShare [56].

and dynamic queue binding. The resources placement module allocates cloud resources to users and balances the use of switch ports between users to provide bandwidth guarantees. The dynamic queue binding module considers the users' traffic demand and their payment factors and dynamically adjusts the high-demand user private queue to achieve work protection. Experimental results show that even under unpredictable traffic demand, QShare can ensure bandwidth guarantee and increase network utilization to over 91%.

In order to guarantee QoS of cloud services, the authors in [50] propose an SDN-based application identification and queue scheduling method. The application identification method first identifies the type of cloud services based on decision tree, and then decides the QoS level required for each type of services to satisfy their QoS requirements. The queue scheduling method dispatches the identified cloud services to different queues and allows for priority execution of delay sensitive data. Experimental results show that their approach achieves reduction of the average delay by 28%. However, this work lacks a theoretical analysis and discussion of the effectiveness of the proposed scheduling algorithms. Moreover, the current classification granularity of the application flows is a little coarse, and can be further improved at finer granularity.

4.7 Discussion

Table 2 provides a comparison summary of various service quality improvement techniques. Based on the summary, comparison, and analysis of the above research works, we know that service quality plays a crucial role in cloud profit optimization due to the close and inseparable relationship between service quality and customer retention. Service quality can be optimized in two scenarios, i.e., resource over-provisioning and resource under-provisioning. In terms of resource over-provisioning, researchers propose various methods, for example, cloud provider federation, workload demand prediction, resource reclaim and sharing, and adaptive dynamic resource scheduling, to improve resource utilization. In terms of resource under-provisioning, some novel approaches also have been presented to solve this problem for satisfying service quality, such as double resource renting mechanism. These research topics are more inclined to solve the service quality

Table 2. A Comparison Summary of Various Service Quality Improvement Methods in Terms of Multiple Aspects, such as Method, User Demand, Resource Type, SLA Model Used, Optimization Goal, Decision, Resource Model, and Constraints

Ref.	Method	User demand	Resource type	SLA model
[64]	Game theory	Known	Homogeneous	Decreased charge model
[85]	Game theory	Unknown	Homogeneous	Other charge model
[69]	Double queuing	Known	Heterogeneous	Stepwise charge model
[52]	Pricing strategy utilization	Unknown	Homogeneous	×
[75]	Pricing strategy utilization	Unknown	Heterogeneous	×
[115]	Resource recycling	Known	Homogeneous	Other charge model
[10]	Resource recycling	Unknown	Homogeneous	Stepwise charge model
[82]	Resource scheduling	Known	Homogeneous	Two stages charge model
[38]	Resource scheduling	Known	Heterogeneous	Stepwise charge model
[19]	Resource scheduling	Unknown	Heterogeneous	Stepwise charge model
[46]	Resource scheduling	Unknown	Homogeneous	×
[48]	Resource sharing	Known	Homogeneous	Two stages charge model
[61]	Resource sharing	Unknown	Heterogeneous	Other charge model
[106]	Bandwidth guarantee	Unknown	Homogeneous	×
[37]	Bandwidth guarantee	Known	Homogeneous	×
[56]	Bandwidth guarantee	Unknown	Homogeneous	×
[50]	Bandwidth guarantee	Unknown	Heterogeneous	×
Ref.	Goal	Decision	Resource model	Constraint(s)
[64]	QoS, profit	Dynamic	Self-owned	Fairness, stability
[85]	QoS, resource utilization	Dynamic	Self-owned	Capacity, commitment
[69]	QoS, profit	Static	Rent from the infrastructure provider	QoS
[52]	QoS, cost	Dynamic	Self-owned	Service level objective
[75]	QoS, cost	Dynamic	Rent from cloud providers	Service level agreement
[115]	QoS, cost	Static	Procure from users	Truthfulness
[10]	QoS, profit	Dynamic	Self-owned	Service level objective
[82]	QoS, cost	Static	Self-owned	Job deadline
[38]	QoS, profit	Static	Self-owned	Job completion time
[19]	QoS, cost	Dynamic	Self-owned	QoS
[46]	QoS	Dynamic	Self-owned	Application provider cost
[48]	QoS, profit	Static	Self-owned	Service level agreement
[61]	QoS	Dynamic	Self-owned	Throughput
[106]	QoS, cost	Dynamic	Self-owned	Bandwidth
[37]	QoS	Static	Self-owned	Bandwidth, work conservation, latency
[56]	QoS, resource utilization	Dynamic	Self-owned	Bandwidth, work conservation
[50]	QoS	Dynamic	÷	QoS

× indicates that the literature does not consider the factor. ÷ implies that the literature does not give a clear explanation of the factor.

Table 3. Classification of Methods of Adjusting the Price

Classification	References
Pricing based on VM location	[114], [116]
Pricing based on resource auction	[94], [96], [63], [108], [114], [115], [72]
Pricing based on users demands	[103], [43], [112], [18], [92]
Pricing based on market competition	[59], [73], [101]

issues from the perspective of cloud providers, ignoring the important role that customers play in cloud profit optimization.

Service quality has a direct impact on customers' purchase behaviors. However, few existing works explore and analyze customers' purchase behaviors when optimizing profitability. Cong et al. [14, 15] and Wang et al. [95] try to model customer perceived value from a psychological perspective to explore customer's psychological activities when purchasing cloud services, while Mei et al. [68] adopt the definition of customer satisfaction in economics and developed a formula to model customer satisfaction in cloud. Thus, when optimizing service quality, researchers can explore the effects of cloud customers' purchase behaviors and psychological activities on cloud profit improvement from perspectives of customer perceived value, customer satisfaction, customer lifetime value, and customer retention rate through interdisciplinary research. We have known that improving service quality can protect providers from penalties for SLA violations, thus improving revenue. Another factor that cannot be ignored, that is, cloud pricing strategy, also has a great influence on cloud providers' revenue. Thus, in the next section, we will take a closer look at cloud pricing mechanisms that optimize cloud provider's profitability by maximizing the value of cloud resources.

5 INCREASE PROFIT BY ADJUSTING PRICES

In this section, the research works on resources pricing to maximize cloud provider's profit are reviewed. Table 3 summarizes references with regard to resources pricing. These works are classified based on the methods they use, as discussed below: (1) pricing based on VM location (Section 5.1), (2) pricing based on resource auction (Section 5.2), (3) pricing based on users demands (Section 5.3), and (4) pricing based on market competition (Section 5.4).

5.1 Pricing Based on VM Location

Zhao et al. [116] optimize cloud provider's profit by dynamically pricing VM resources in data centers of different locations. They develop an online profit maximization approach to solve the optimization problem. The online algorithm sets the optimal price for VMs of different data centers and configures the optimal amount of resources for VMs. In the online algorithm, the pricing decisions are made dynamically according to the current system state without historical data. Experiments show that the algorithm can achieve an average total profit close to the offline maximum and can obtain a more stable profit over time. Actually, this work can be further improved by designing an effective future workload predictive algorithm rather than assuming that the future workload information is known.

Zhang et al. [114] present an auction scheme for dynamic VMs configuration and pricing in a geographically distributed data center to optimize profit. They first convert the profit maximization issue into the ILP model. The ILP problem is then resolved by their proposed smooth analysis-based dynamic analysis algorithm. Afterward, they design a resource allocation scheme based on random reduction to convert the profit maximization solution into the auction resource allocation scheme. Finally, they complete the auction design by combining the random resource allocation scheme

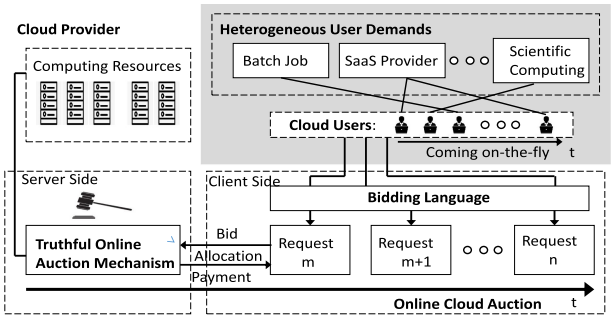

Fig. 7. An overview of online cloud auction framework [108].

with the well-known offline Vickrey-Clarke-Groves method. Simulation experiments show that their scheme is superior to the original dual approximation scheme in terms of profit and user satisfaction.

Ekwe-Ekwe and Barker [20] investigate the effect of geographic location on the deployment cost of a Spot instance of Amazon EC2 in terms of pricing and reliability factors. A detailed analysis of the impact of different location on the deployment cost of a Spot instance have been conducted in this work. More specifically, they first collect all available pricing data of various spot instance types from all available Amazon Web Services (AWS) areas over 60 days and associated the data with all the AWS areas and their Available Zone (AZ). Then, a histogram analysis is performed to obtain the frequency of all price points under different areas. Based on this histogram analysis, they demonstrate the possibility of decreasing deployment cost by rationally selecting Spot instances of different locations. For example, they conclude that deploying a powerful instance in a cheaper available zone that is located in a more expensive area is a practicable and possible choice. However, this work only provides a theoretical basis and lacks specific design of a geographic location-based pricing algorithm, which is more helpful for cloud providers to achieve profit improvement.

5.2 Pricing Based on Resource Auction

Aiming at effectively selling cloud resources, Zhang et al. [108] design an online cloud resource auction architecture, as shown in Figure 7. They first design a bidding language to unify the different needs of users into a standardized and consistent form. Through this language, users' service requests can be expressed in a more concise and standardized way. Then, based on the bidding language, the authors propose an online resource auction scheme, which is mainly composed of an allocation rule and a payment rule. In particular, the former rule aims to maximize bidders' utility while the latter rule determines the allocation result based on user request form and the allocation rule. Further, in order to ensure the truthfulness of the auction framework, a non-decreasing auxiliary pricing function is introduced to capture the current supply and demand relationship. Experimental results show that their method achieves comparable performance as compared to the well-known offline VCG method.

In order to optimize cloud providers' profits, Nejad et al. [72] propose a series of auction-based VM allocation mechanisms. The mechanisms first receive bids from each participating user. The bids include the amount and valuation of requested resources. Then, the allocation and payment functions are exploited to determine the allocation of VMs and user charges. In order to ensure the truthfulness of auction schemes, the allocation function is designed to be monotonous, and the payment function is designed based on threshold values. The authors convert this problem into an integer programming model. To this end, a truthful greedy approximation method is designed.

Experiment results show that their method can determine the allocation for the approximately maximum profit in a very short time while satisfying the truthfulness. However, a prototype VM allocation system is not implemented in an experimental cloud computing context, which is also important for method evaluation.

Aiming at maximizing profit and resource utilization of cloud providers, Lena et al. [63] design a VM configuration and allocation scheme. In this scheme, users submit service requests to express the number and bid of VM instances they need, and cloud providers will select some users to allocate resources for maximizing revenue due to limited resources. In order to obtain the highest revenue, a dynamic programming algorithm and VCG-based optimal selection scheme is designed. The dynamic programming–based method that provides the best solution is equivalent to the multi-dimensional knapsack problem of NP difficulty, thus a polynomial time approximation scheme is designed to obtain an approximate optimal solution. Experiment results demonstrate that their scheme can achieve approximate optimal resource allocation while satisfying truthfulness.

5.3 Pricing Based on User Demands

The authors in [43] propose a dynamic bidding scheme to maximize the profitability of cloud providers by an auction-based cloud spot pricing method. A hidden Markov model is first designed to explore cloud spot market dynamics, which characterizes user demand in the market by using spot prices as a function of the potential state. The model predicts the future cloud spot prices based on the past spot prices, enabling cloud providers to derive user bid strategies. Based on the model, the authors obtain the cloud spot pricing strategy based on the historical behavior of the spot price. Experiments show that their method can achieve up to four times closer to the optimal strategy than the baseline regression method.

Xu and Li [103] propose an optimal dynamic pricing strategy to optimize long-term cloud providers' profits. Considering the stochastic demand and perishable resources, the authors exploit the economics-based revenue management framework to model the profit maximization problem as a finite-time stochastic dynamic program. Subsequently, the authors prove that the optimal pricing strategy has the monotony of time and utilization, that is, the optimal profit has a concave structure. Based on this conclusion, they balance current pricing with future demand to attract more profit from future demand and generate more profit from existing customers. Simulation results show that dynamic pricing achieves greater improvement for the profit of cloud providers than static pricing. This work assumes that the cloud provider is in a monopoly environment, which is somewhat unrealistic. Moreover, this work lacks an analysis of the cloud market environment and customer behaviors.

To maximize the profitability of cloud providers, the authors in [112] propose a pricing strategy that iteratively updates the multiple classes of VM prices. Based on genetic and hill climbing algorithms, the pricing model periodically updates the VM price according to the user's selection of the VM category and the required amount until the cloud provider's profit converges. The genetic-based algorithm enables near-optimal pricing, but slower convergence speeds increase the cost of communication for cloud providers to receive user feedback. The hill climbing–based algorithm has slightly lower profitability than the genetic-based algorithm, but greatly reduces the communication costs of cloud providers. Experiments show that their pricing model achieves profit improvement for cloud providers, whereas the stability of the cloud systems has not been analyzed and discussed in this work, and thus needs to be further investigated in the future.

The authors in [18] propose a resource pricing and allocation scheme to improve resource utilization and optimize cloud provider's profitability. The scheme leverages modeless deep reinforcement learning (DRL) to capture the dynamics of cloud users for developing cloud resource

pricing and allocation decisions to maximize profits. The output of the DRL model is the probability distribution of choices between different servers and the unit time usage price of the server. Through this model, the cloud provider selects the server for the user service request and publishes the price of the corresponding server. Experiments show that compared with the basic DRL algorithm and the state-of-the-art white-box online cloud resource allocation/pricing algorithm, the proposed DRL method can increase the profit by at least 25% and increase the number of users by at least 15%. However, the rapid growth of cloud users and tasks will greatly increase the number of environmental states, which will put a heavy burden on their proposed algorithm.

In order to optimize long-term profit for cloud providers, the authors in [92] propose a reactive pricing (RP) algorithm based on Lyapunov optimization. RP optimizes the profit of cloud providers from three aspects: server pricing, battery management, and power procurement. For server pricing, the RP dynamically adjusts server prices in response to demand changes by capturing energy consumption costs and server supply and demand relationships. For battery management, the RP determines when the server battery is charging and discharging and considers charging the battery with renewable energy. For power purchases, RP reduces the cost of cloud providers by purchasing low-cost power for future use. Experiments show that their method can effectively increase cloud provider's profit by utilizing user demand, renewable energy, and power price fluctuations.

5.4 Pricing Based on Market Competition

Macías and Guitart [59] propose a genetic-based pricing model to optimize cloud providers' profits in the competitive cloud market. They first define the general pricing model as a chromosome, parameters of pricing model as genes, and then randomly generate chromosomes with different genes. Based on the profit generated by chromosomes, the authors select the most profitable chromosome and then replicate and mutate by simulating the natural evolutionary process. After multiple iterations, the chromosomal population will generate a pricing model that maximizes cloud provider's profit. In addition, the proposed model assumes that the cloud market is uncertain and can adapt to changes of market. In particular, the proposed method can be further improved by designing a mechanism that dynamically adjusts the number of chromosomes, mutation rate, and other data to improve the solution. Experimental results show that their method generates 100% higher profit than the utility-based dynamic pricing model and 1000% higher than the typical fixed price model in the competitive environment. However, their method should also be tested in real cloud market environments.

The authors in [73] consider the competitive market for multiple cloud providers and designed a resource pricing scheme to form a fair and profitable cloud services market. The pricing scheme uses price regulators to coordinate the price of resources for all cloud providers to achieve a trade-off between profit optimization and user satisfaction improvement. Through this price regulator, cloud providers can iteratively decide the optimal price by taking into account the prices charged by other competitors. In addition, in order to prevent cloud providers from becoming market monopolies, the market regulator is also in charge of reviewing QoS served by cloud providers. Simulation results show that their pricing strategy is better than the benchmarks in terms of economic efficiency and fairness.

The authors in [101] study how to provide competitive cloud resource pricing in the markets of multiple cloud providers. First, the competition between cloud providers is modeled as a limited continuous game. At each stage of the game, one cloud provider changes the price of resources, and then other cloud providers passively update their prices based price changes. Then, the optimal pricing model is converted into a Markov decision process (MDP) model, and the Q-learning-based algorithm is further designed to find the optimal pricing strategy. Experiments show that their method can achieve more provider profit improvement as compared with the two benchmark

Table 4. A Comparison Summary of Various Resources Pricing Methods in Terms of Multiple Aspects, such as Pricing Based, User Demand, Resource Type, Optimization Goal, Profit Improvement, Time Consumption, Benchmarks, and Constraints

Ref.	Pricing based	User demand	Resource type	Goal
[116]	VM location	Known	Heterogeneous	Profit
[114]	VM location	Known	Heterogeneous	Profit, user satisfaction
[108]	Resource auction	Unknown	Heterogeneous	Profit
[72]	Resource auction	Known	Homogeneous, heterogeneous	Profit, execution time
[63]	Resource auction	Known	Heterogeneous	Profit, execution time
[43]	User demands	Unknown	Heterogeneous	Profit
[103]	User demands	Known	Homogeneous, heterogeneous	Profit
[112]	User demands	Known	Heterogeneous	Profit
[18]	User demands	Known	Heterogeneous	Profit
[92]	User demands	Known	Homogeneous	Profit
[59]	Market competition	Unknown	Homogeneous	Profit
[73]	Market competition	Known	Homogeneous	Profit
[101]	Market competition	Known	Homogeneous	Profit
Ref.	Profit	Time	Benchmarks	Constraint(s)
[116]	÷	×	Heuristic and static pricing	Job deadline
[114]	272%	×	Primal-dual approximation pricing	Truthfulness
[108]	÷	÷	Offline optimal pricing	Truthfulness
[72]	÷	100×	Offline optimal pricing	Truthfulness
[63]	÷	100×	Offline optimal pricing	Truthfulness
[43]	400%	×	Auto-regressive pricing	Capacity
[103]	÷	÷	×	Capacity
[112]	15%–21%	×	Genetic based pricing	÷
[18]	25%	×	Online dynamic pricing	÷
[92]	200%	×	Online dynamic pricing	Battery capacity
[59]	100%–1000%	×	Static and dynamic pricing	Service level agreement
[73]	20.54%–26.33%	×	Fair pricing scheme	QoS, operational feasibility
[101]	45%	×	Price reduction policies	Evolutionary cloud computing market

× indicates that the literature does not consider the factor. ÷ implies that the literature does not give a clear explanation of the factor.

pricing strategies. This work assumes that there is only one cloud provider in the market, which is not realistic and cannot be applied to real cloud market environments. Correspondingly, it also cannot be used to solve the optimal pricing problem in a cloud market with multiple cloud providers simultaneously.

5.5 Discussion

Table 4 summarizes and compares various cloud pricing methods from multiple aspects. In the current cloud market, there are a large number of pricing strategies, which can be divided into two main categories, that is, static/fixed pricing and dynamic/adaptive pricing. Static/fixed pricing strategy is the current mainstream pricing strategy adopted by most cloud providers. However, due to the inflexibility, this kind of pricing strategy cannot adapt to the dynamic and volatile market demand, thus missing opportunities for revenue improvement. To this end, a variety of

Table 5. Classification of Methods to Reduce Electricity Bills

Classification	References
Reducing communication costs of data centers	[51], [16], [17], [91]
VMs provisioning and allocation	[8], [28], [72], [79], [27]
Load distribution and balancing	[9], [54], [61], [117], [26], [102]
Approximate computing	[36], [12]
Dynamic power management	[104], [99]

dynamic/adaptive pricing strategies have emerged as necessary designs for different scenarios, such as user market demand–based, resource/service-based, geographic location-based, market competitor–based, and auction-based. These novel pricing mechanisms reflect that when pricing resources or services, cloud service providers cannot just raise prices or lower prices. The former will reduce the number of customers while the latter will increase the number of customers; however, the total revenue will not necessarily increase. Thus, an appropriate pricing model not only needs to bring considerable profits to cloud service providers, but also needs to bring a positive service experience to customers.

How to build such win-win cloud pricing models is an urgent problem to be solved for cloud service providers in the cloud market. There are too many factors that need to be taken into account when building pricing models, such as heterogeneity of resources, services, and customers, volatility of market demand, and diversity of competitors; how to measure the importance of these factors in profit optimization is crucial. A suitable pricing strategy can increase revenue by maximizing the value of cloud resources, while an improved service quality can avoid revenue reduction due to the penalty of SLA violation. So far, we have explored how to optimize profits from the perspective of increasing revenue in terms of service quality and pricing models. In addition to profit optimization by increasing revenue, reducing server energy consumption and VNF deployment costs can also enhance cloud providers' profits. In the next section, we discuss state-of-the-art optimization techniques for server energy consumption and VNF deployment to further improve cloud providers' profits.

6 INCREASE PROFIT BY REDUCING ELECTRICITY BILLS

In this section, the research works on reducing electricity bills to maximize cloud providers' profits are reviewed. Table 5 summarizes the references focusing on server energy consumption optimization. These works are classified based on the methods they use, as discussed below: (1) reducing communication energy of data centers (Section 6.1), (2) VMs provisioning and allocation (Section 6.2), (3) load distribution and balancing (Section 6.3), (4) approximate computing (Section 6.4), and (5) dynamic power management (Section 6.5).

6.1 Reducing Communication Energy of Data Centers

Li et al. [51] investigate VM placement to minimize network traffic costs (N-cost) and physical machine costs (PM-cost), as shown in Figure 8. They define N-cost based on three communication models, formulate the VM placement problem, and solve the problem from the perspectives of fixed PM-cost and varying PM-cost. First, they minimize N-cost when PM-cost is set fixed. The problem is studied under two scenarios, that is, homogeneous and heterogeneous. For the homogeneous scenario, users demand the same numbers of VMs. Further, three communication models–based optimal algorithms are proposed to solve the cost minimization problem. For the heterogeneous scenario, users demand different numbers of VMs and approximate algorithms based on three communication models are proposed to obtain an approximate solution. Then, they minimize the

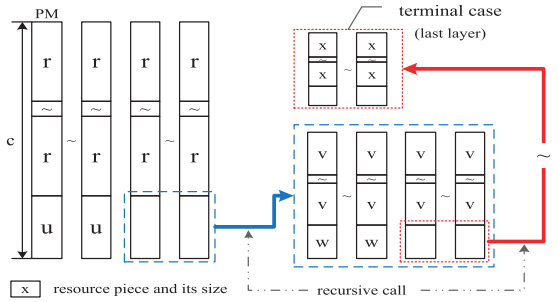

Fig. 8. An overview of the solution architecture [51].

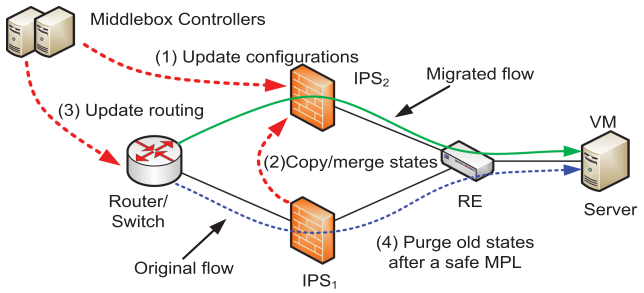

Fig. 9. An illustration of policy migration [16].

sum of PM-cost and N-cost when PM-cost is changing dynamically. A binary search–based heuristic method is proposed to minimize the sum of PM-cost and N-cost by obtaining the number of PMs. Experimental results show that their method can achieve power saving while improving performance.

Cui et al. [16] reduce network communication costs by optimizing dynamic reallocation of VMs and network policies, as shown in Figure 9. First, they formulate the joint optimization problem and prove it is NP-Hard. Then, they design Sync, a migration scheme that minimizes communication costs in two stages. In stage 1, they migrate policies and prepare VM migration by building preference matrices of servers. In stage 2, they model the above VM migration problem as a many-to-one matching, and decide the migration goal of each VM based on the preference matirces of servers. Experimental results show that Sync not only reduces the communication cost of the data center by 50% but also decreases end-to-end delay by 38.8%. In particular, this work can be further improved by taking dynamic VM configuration into account to address the dynamic changes in terms of VM resource requirements.

6.2 VM Provisioning and Allocation

Ghaderi et al. [28] design a VM schedule scheme to save cloud providers' energy consumption in multi-server systems. The VM schedule scheme exploits random algorithms to place VMs for maximizing system throughput in a data center. The main idea is to generate the VM configuration by using a loss system, which is composed of a set of servers, a set of VM types, and a vector of weights. The VM arrivals of the loss system are governed by dedicated Poisson clocks. By assigning a dedicated Poisson clock to centralized and distributed queues of VMs, the proposed randomized algorithms can stabilize the queues and maximize the throughput without preemptions of ongoing services. Simulation results show that their method achieves optimal throughput and shorter delay

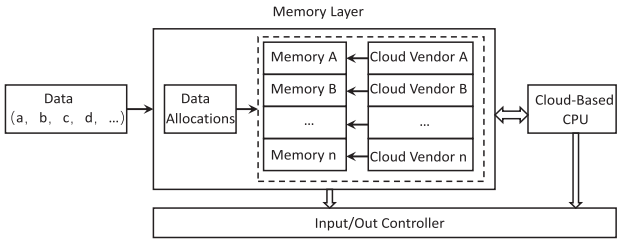

Fig. 10. The structure of cloud-based heterogeneous memories [27].

compared to the existing throughput-optimal algorithms. However, it is worth noting that the sampling rate of the proposed random sampling method depends on the size of the waiting queue, and it will take a long time to converge when the queue grows.

Ren and He [79] propose an online algorithm to minimize operating costs including energy costs and delay costs. The online algorithm minimizes the operating cost of data centers through server speed control and load allocation. For speed control, they first build energy consumption queues for servers and then leverage Lyapunov optimization to balance server speed and latency. For load allocation, the authors propose an algorithm based on Gibbs sampling to iteratively update the optimal load allocation decisions of servers. Specifically, based on the proposed online algorithm, each server can automatically adjust its speed and determine the optimal load allocation to improve data center operating costs. Experimental results demonstrate that their method achieves more than 25% reduction of the operating cost of a data center compared to the perfect hourly prediction heuristic method.

Gai et al. [27] develop a task allocation method to allocate tasks to various types of cloud storage to minimize the cost of cloud providers, as shown in Figure 10. The authors first design a cost-aware heterogeneous cloud storage model that considers key factors affecting cost, such as communication, data migration, performance, and time constraints. Based on the proposed cloud storage model, a dynamic genetic-based data task allocation algorithm is designed to determine the distribution of data tasks on cloud storage for minimizing the total cost of cloud providers. Experimental results show that the task allocation scheme can output the optimal solution at high running speed. However, the proposed solution cannot perform data allocation dynamically under changing memory service requirements.

6.3 Load Distribution and Balancing

In order to optimize power consumption for cloud providers, Cao et al. [9] propose load distribution energy optimization methods under performance constraints. They explore the balance between energy consumption and performance by formulating the optimization problems as two constrained optimization problems, that is, a performance optimization problem under the constraint of power consumption and a power optimization problem under the constraint of performance. By solving the former performance optimization problem, cloud providers maximize QoS while saving data center power consumption. By solving the latter power optimization problem, cloud providers achieve power consumption minimization while meeting QoS. The authors solve the above two optimization problems under two different speed models, one of which assumes that servers are running at zero speed when idle, while the other assumes that servers are running at a constant speed. Experimental results show that their method effectively achieves power reduction for cloud providers while ensuring performance under different speed models. This work only has theoretical calculations and needs to be implemented and verified in real cloud computing environments.

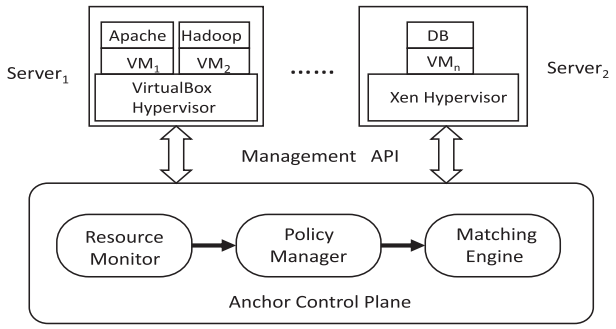

Fig. 11. The framework of Anchor [102].

To maximize profits of cloud providers in the context of the multi-electricity market, Liu et al. [54] propose a service request scheduling and resource assignment algorithm to optimize energy consumption for cloud providers. They first propose a resource management framework that takes multi-electricity market, SLA, and net profit into account. The framework models profits obtained by cloud providers as a multi-level down utility function that can simulate various situations. The framework then determines the allocation of user service requests, the number of servers, and the resource allocation scheme. The authors formulate the problems as a constrained optimization problem by transforming the utility function into well-defined constraints, which can be solved with existing solvers. Simulation results show that their method greatly improves profit through effective utilization of energy and resources.

To save data center energy consumption, Fu et al. [26] propose a heuristic allocation scheme to optimize the ratio of throughput to energy consumption. Their proposed energy-efficiency (EE) strategy saves energy by exploiting server heterogeneities from the perspectives of server speed and power consumption. In EE, the most energy-efficient servers are aggregated as a virtual server and the utilization of this virtual server is prioritized so that the system throughput and energy efficiency can be increased. In addition, EE strategy can ensure the predictability and robustness of data centers when the job size distribution is unpredictable. Extensive experiments have shown that EE can increase energy efficiency by 70% while maintaining system throughput compared to the slowest server first (SSF) method.

In order to efficiently manage heterogeneous resources in cloud, Xu and Li [102] design Anchor to effectively match VM requirements to servers, as shown in Figure 11. Anchor is mainly composed of a resource monitor module, a policy manager module, and a matching engine module. The first module manages resource according to cost, performance, and other factors required by cloud providers and users. When the VM placement request arrives, the second module polls the information from the resource monitor and provides the information to the third module. When the third module receives the information from the second module, it solves the conflict among the stakeholders based on a stable economic matching framework, and outputs the matching between VMs and servers. In summary, Anchor can efficiently match VMs of heterogeneous resource requirements to servers through a many-to-one stable matching framework. However, this work dose not consider dynamic resource requirements, which is critical in real-world cloud environments and can lead to VM relocation and migration. Experiment results show that this architecture can provide near-optimal performance for large-scale data centers.

6.4 Approximate Computing

Aiming at reducing the resource cost of mining big data in the cloud, the authors in [36] study how to obtain sufficient accuracy results at a lower computational cost. The well-known k-means

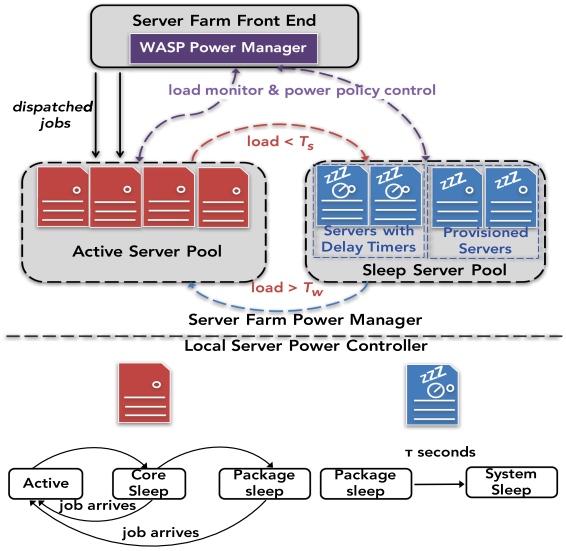

Fig. 12. Power Management Framework [104].

algorithm is used to explore and demonstrate the cost-effectiveness of big data mining in the cloud. They first use the Lloyd k-means algorithm to divide the dataset and used different k in different experiments. Then, they calculate the accuracy and time of the different partitions and analyze and discuss the results. Experiments show that the k-means algorithm can achieve 99% accuracy, and the computational cost is 0.32%–46.17%.

The authors in [12] take advantage of the demand elasticity of data analytic to reduce cloud resource consumption. Demand resilience allows work to run in much less than ideally needed resources with modest performance losses. They propose a performance awareness fair (PAF) scheduler to leverage demand elasticity to optimize resource consumption while achieving near-fair guarantees. PAF first distributes resources fairly and then iteratively updates the allocation of resources to increase the average performance of cloud providers. Experimental results show that PAF increases the average performance of cloud providers by 13% as compared with fair distribution, while the penalty for SLA violations does not exceed 1%. However, due to the interferences incurred by resource contention and sharing mechanisms, the proposed solution can cause unpredictable performance changes.

6.5 Dynamic Power Management

The authors in [104] propose an energy optimization framework for server farms that adaptively adjusts server power based on workload to minimize energy consumption while meeting QoS constraints, as shown in Figure 12. The proposed energy optimization framework includes a global power manager and a local power controller. The global power manager sends a power mode translation request to the server that needs to be shut down based on the user request and the current load of each server. Upon receiving the power mode transition request, the local power controller first processes all tasks in the local task queue, then starts the delay timer and enters the system sleep state when the delay timer expires. When facing a surge in workload, the timer will be reset and the server will switch to the active mode when the task arrives before the timer expires. Experimental results demonstrate that their scheme can save up to 57% and 39% energy as compared to the simple strategy using only shallow processor sleep states and the delay

Table 6. A Comparison Summary of Some Server Energy Optimization Methods in Terms of Multiple Aspects, such as Method, Resource Type, Granularity, Constraints, Optimization Goal, Energy Saving, Performance Improvement, and Benchmarks

Ref.	Method	Resource type	Granularity	Constraint(s)
[51]	Reducing communication energy	Homogeneous, heterogeneous	Request	Resource capacity
[16]	Reducing communication energy	Heterogeneous	Flow	Network policy requirements
[28]	VMs provisioning and allocation	Homogeneous	Job	Packing and non-preemption
[79]	VMs provisioning and allocation	Heterogeneous	Workload	Carbon neutrality
[27]	VMs provisioning and allocation	Heterogeneous	Task	Communication and data move costs, energy performance, and time
[9]	Load distribution and balancing	Heterogeneous	Task	Power or performance
[54]	Load distribution and balancing	Homogeneous	Request	QoS
[26]	Load distribution and balancing	Heterogeneous	Job	Energy efficiency
[102]	Load distribution and balancing	Homogeneous	Workload	VM placement
[12]	Approximate computing	Homogeneous	Job	Fairness
[104]	Dynamic power management	Heterogeneous	Job	QoS
Ref.	Goal	Energy	Performance	Benchmarks
[51]	Energy	−28%	×	Greedy placement algorithm
[16]	Energy, performance	−50%	+38.8%	S-CORE [91]
[28]	Energy	÷	÷	÷
[79]	Energy	−25%	×	Prediction-based method [78]
[27]	Energy, performance	÷	÷	÷
[9]	Energy, performance	÷	÷	÷
[54]	Energy, profit	÷	×	÷
[26]	Energy	−70%	×	Slowest-server-first policy [84]
[102]	Energy, performance	×	+60%	First fit algorithm [70]
[12]	Performance	×	+13%	Fair allocation
[104]	Energy, service quality	−57%	÷	Naive policy

× indicates that the literature does not consider the factor. ÷ implies that the literature does not give a clear explanation of the factor.

timer–based methods, respectively. However, it is worth noting that switching a server to the idle state requires that all cores of this server are idle at the same time, which makes the proposed solution more difficult to solve.

In order to save power while satisfying a specified SLA, the authors in [99] propose a predictive scheduling method for energy savings in computing infrastructure by using a private cloud. This method monitors cloud activity and uses quantile forecasts to estimate the number of servers that will be requested in the next time period. Based on the predicted results and SLA between the cloud provider and users, the method then shuts down some machines to save power. Experiment results show that their method produces significant power savings in the tradeoff between energy savings and reduced user experience.

6.6 Discussion

Table 6 compares the above server energy consumption optimization methods in detail from aspects of method used, resource type, service granularity, energy saving, and so on. As discussed in [76], millions of dollars are needed to be spent annually on the electricity to power a

Table 7. Classification of Methods of Reducing the Cost of Deployment

Classification	References
Reduce deployment cost via machine learning method	[105], [71], [23], [113]
Reduce deployment cost via greedy strategy	[45], [49], [86]
Reduce deployment cost via Monte Carlo and Markov Model	[22], [89], [74]
Reduce deployment cost via other common optimization methods	[7], [97], [110], [77], [57]

geographically distributed cloud system, which is made up of hundreds of thousands of servers and several data centers. In order to reduce tremendous infrastructure energy consumption, a large number of efficient methods have been proposed, such as novel cooling technologies, low-power hardware design, various energy-efficient task and resource scheduling algorithms, energy-aware load distribution balancing algorithms, dynamic and adaptive power management mechanisms, and energy-saving communications under scenarios of resource-resource, data center–data center, and customer-data center. Existing works have provided a broad range of ideas for data center energy savings from different perspectives. Unfortunately, most of these works save energy, but at the expense of performance.

Renewable energy sources (e.g., wind and solar), as an effective mechanism, could be subtly utilized to power energy-consuming data centers. Several works use renewable energy sources as an alternative energy source for electricity [92]. The authors in [76] use the volatility of electricity prices in different geographic regions to find opportunities for data center energy conservation. In the future, higher accuracy workload prediction algorithms, as well as customer-based personalized task and resource scheduling solutions will have indispensable impacts on the flexible provision and scheduling of data center resources, thus further bringing considerable benefits to energy consumption reduction. The above discussed methods aim to reduce operating costs of cloud providers by minimizing electricity bills of servers in data centers. In addition to the operating costs, cloud service providers deploy VNF to provide users with fast and inexpensive network capabilities, resulting in non-negligible deployment costs that also have a significant impact on cloud service providers' profits. In the next section, the VNF instance deployment strategies are described in detail. These VNF instance deployment strategies can increase resource utilization and remarkably reduce deployment costs, bringing more profits for cloud service providers.

7 INCREASE PROFIT BY REDUCING DEPLOYMENT COST

In this section, we review works on reducing VNF deployment cost to maximize cloud providers' profits. Table 7 summarizes references with regard to cost reduction of VNF deployment. These works are roughly divided into four categories based on techniques adopted: (1) machine learning (Section 7.1), (2) greedy based (Section 7.2), (3) Monte Carlo and Markov Model based (Section 7.3), and (4) other optimization techniques (Section 7.4).

7.1 Using Machine Learning Method

In order to deploy VNF to provide network services more efficiently, Ye et al. [105] present a heuristic algorithm for VNF combining and service function chaining (SFC) mapping to achieve bandwidth cost minimization. The heuristic constructs a decision tree in descending order of bandwidth savings and evaluates mapping costs in all possible combined strategies to determine VNF combination, as shown in Figure 13. After VNF combination is established, the SFC mapping is determined by a link mapping priority algorithm. Experimental results show that their method can save network reconfiguration cost by 38.2% as compared with baseline algorithms while improving

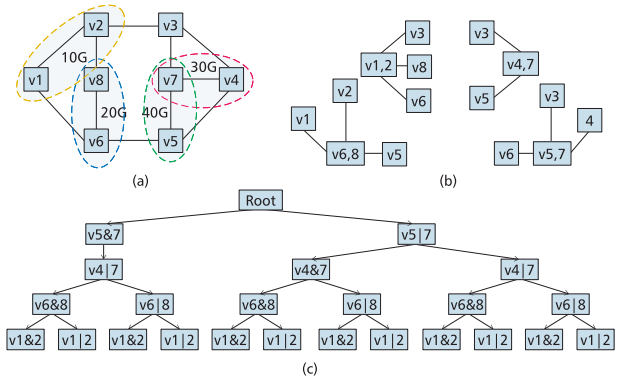

Fig. 13. The decision tree [105].

service reliability by 12.2%. However, the type of resources studied in this work is somewhat single. The proposed solution needs to consider more types of physical resources and requires solid theoretical work to support.

In order to effectively utilize physical resources, Mijumbi et al. [71] design a graph neural network (GNN) based scheme to dynamically arrange VNFs for demand fluctuations. The authors first model a virtual network functional component as two parameter functions. Then, these two parameter functions are carried out by a feedforward neural network (FNN). The FNN learns the resource demand trend of a local virtual network function component (VNFC) by combining historical VNFC with its neighbor's resource configuration information. Based on the resource demand trend obtained by FNN, the algorithm determines the opening and configuration of a new VNFC and the shutdown of an old VNFC. The proposed FNN algorithm needs to store the state of all parameter functions. Such a large-scale VNF will impose a large memory burden on the cloud provider. Experiments show that their method achieves an average prediction accuracy of 90% for future resource requirements of VNFC.

To minimize cloud providers' cost due to VNF deployments, Fei et al. [23] propose an online VNF provisioning scheme to dynamically deploy VNF and reroute traffic demands of users. The online scheme predicts traffic demands of users and deploys new instances for overloaded VNFs according to predicted traffic. With regard to the prediction of users demands, the authors design a regularization-based online learning method to predict upcoming traffic. With regard to the assignment of new VNF instances, they propose two online algorithms to allocate these new VNF instances to servers with sufficient space capacity and assign route traffic along the service links to VNF instances on the proper network links. Experiment results show that compared to other benchmarking schemes, their online VNF provisioning scheme can reduce deployment costs while achieving more accurate prediction. This work can take the dynamic scalability of VNF into account to improve resource flexibility.

Zhang et al. [113] minimize resource costs of cloud providers by efficiently estimating upcoming traffic rates and adjusting VNF deployments. In order to estimate the upcoming traffic rates effectively, they first exploit a stochastic convexity technique to formally define the cost minimization problem as an convex optimization model. Then, an online gradient descent approach is used to predict the upcoming flow demands along VNF chains to minimize the prediction errors. Based on the predicted user traffic requirements, the authors design an online algorithm to purchase VMs and deploy VNFs based on the ski rental algorithm. Experiments show that their method is superior to the benchmarks in terms of user demands forecasting and costs minimization.

7.2 Using Greedy Strategy

To reduce deployment and operating expenses, Sang et al. [86] minimize the cost of VNF instances in a network. They first convert the cost minimization issue into a mixed integer linear programming (MILP) model. Then, two simple greedy algorithms are designed to solve the MILP problem. The first one is a flow number–based greedy mechanism that iteratively chooses the node which has the largest amount of unprocessed flows. The second one is a flow rate–based greedy mechanism that chooses the node that has the largest amount of unprocessed data. Based on randomly generated dense graphs and real backbone network topology of Internet MCI, experiment results show that their method can perform well in all cases. However, this work ignores the effect of dynamic decision and scalability of VNF on the solution. These two factors need to be taken into consideration to further optimize the deployment and operating expenses.

In order to optimize resource utilization, Li et al. [49] solve the VNF deployment cost minimization problem for cloud providers considering the time-varying characteristics of workloads of NF requests. They first formulate the VNF deployment cost minimization as an ILP problem to achieve minimization of the amount of used resources. To this end, a two-stage heuristic scheme is proposed. In stage 1, the scheme maps all service function chain (SFC) requests to physical machines (PMs) one by one using a correlation-based greedy method. In stage 2, they develop an adjustment method to save resources by sharing the basic resource consumptions for VNF requests on the VNF with multi-tenancy capabilities. Simulation results show that the method performs better than benchmarking schemes.

Since users need to invoke multiple VNFs in the order determined by the routing path, Kuo et al. [45] collaboratively optimize VNF deployment and path selection for resource cost minimization. Aiming at determining the routing path for demands of user, the authors propose a system approach based on stress testing that dynamically adjusts the path and resource usage of each demand according to system state and demand attributes. In order to determine the VNF deployment, a link deployment algorithm based on greedy strategies and dynamic programming is proposed to deploy routing paths required by user requirements. Experiments show that the algorithm is superior to other greedy and shortest path–based heuristics and can better utilize limited resources to meet larger scale requirements. Moreover, the algorithm can also make resource allocation better adapt to network dynamics.

7.3 Using Monte Carlo and Markov Model

Soualah et al. [89] propose a Monte Carlo tree search–based optimal VNF deployment and linking algorithm that considers energy efficiency and cost savings while meeting multi-tenant requirements. The proposed algorithm optimizes power consumption at hardware and software levels. At hardware level, the proposed approach minimizes energy consumption by providing privileges to the servers which are more energy-efficient. At software level, the proposed approach achieves energy savings that depend on the best VNF sharing across multiple tenants. Experimental results show that their method can minimize the power consumption of cloud providers while improving the scalability of NFV.

Pham et al. [74] address the issue of minimizing capital and operating expenses of VNF placement for cloud providers, as shown in Figure 14. They first formulate the problem of minimizing operational costs and network traffic costs as a joint optimization problem. Then, they design a sample-based Markov approximation scheme to solve this joint optimization problem, which can produce an approximate optimal solution. To further reduce the cost of calculations, a many-to-one matching game is proposed to reduce the space of feasible solutions. Experiment results show that their method can save up to 19% total cost for cloud providers as compared with the

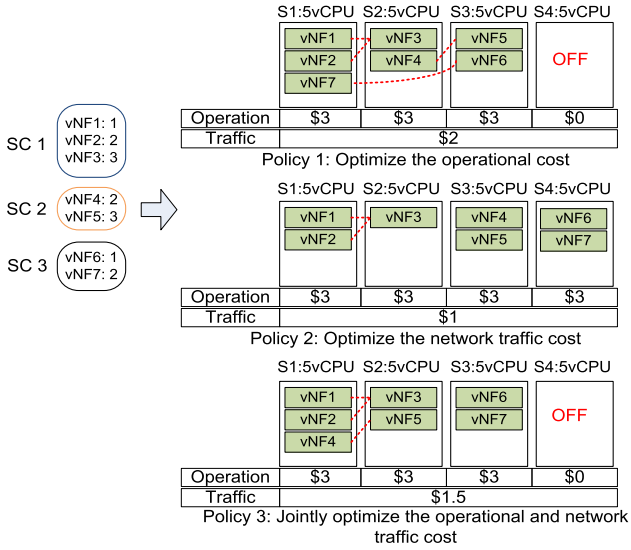

Fig. 14. An example of VNF placement with different policies [74].

non-coordinated schemes. However, the proposed solution may lead to sub-optimal or even useless scenarios in the case of unknown network traffic.

Eramo et al. [22] propose a migration scheme to save total cost of VNF instance migration, including the energy cost and reconfiguration cost of VNF instance moves. The strategy uses the Markov decision process theory to determine the location and amount of resources of each VNF instance after migration to cope with changing service chain requests. Through this migration strategy, cloud providers can complete VNF migration with minimal resources to minimize placement costs while minimizing the rejection of service function chain requests. In addition, they consider the loss of revenue due to loss of user information during the migration. Experimental results show that their strategy can be improved by approximately 27% compared to a simple strategy that does not consider future reconfiguration costs. However, this work does not consider the bandwidth consumption of the VNF, which also has an impact on the optimization results.

7.4 Using Other Optimization Methods

Bhamare et al. [7] study virtual network functional placement issues to minimize latency and resource costs in geographically distributed clouds. They improve user latency by optimizing the placement of VNFs in a cloudy environment under important constraints such as overall deployment costs and SLAs. First, the link queues and the server queues are modeled as M/D/1 and M/M/1, respectively, to accurately estimate user delays in cloudy scenarios. Then, an ILP method is used to obtain an optimal solution under the total deployment cost and SLA constraints. Since ILP methods are often not scalable due to their computational complexity, the authors propose an affinity-based heuristic with short execution time and little impact on the solution quality. Experiment results show that affinity-based methods have lower overall latency and total resource costs than benchmarking methods.

Wang et al. [97] present two efficient online methods to optimize VNF deployment cost in the absence of information about future traffic rates. The algorithms determine the configured instances number of each type of VNF each time, while considering both the server capacity and the traffic rate between near VNFs. For a single service chain, they design a randomized online

method while for multiple concurrent service chains, they design a minimal weight matching based heuristic method to optimize the deployment cost of cloud providers. Experiments show that their method can achieve significant cost savings for cloud providers. However, the proposed scheme is a reactive VNF scaling scheme, which can result in delays, packet loss, and degrade the quality of service.

Zhang et al. [110] improve resource utilization and optimize service request response times through VNF chain placement and service request scheduling. The authors model the VNF chain through Jackson's network theory to capture network traffic characteristics such as network congestion and request rejection rates. The placement of the VNF chain is formulated as a variant packaging problem and a priority-driven algorithm is designed to achieve an approximate optimal placement of the VNF chain. The service request scheduling problem is formulated as a multipath partitioning problem, which is subsequently solved by a heuristic method. It is worth noting that the vertical scaling technique employed by the proposed solution cannot avoid service interruption and rejection. Experiment results show that compared with advanced algorithms, their method optimizes the resource utilization and the delay by 33.4% and 19.9%, respectively.

To minimize resource consumption of the cloud provider, Rankothge et al. [77] design a VNF placement scheme to determine the initial placement of VNF and the VNF scaling caused by traffic fluctuations. For the initial placement of the VNF, the authors minimize the amount of servers and links required. For VNF scaling, the authors dynamically change resources to meet traffic fluctuations while minimizing the number of configuration changes to decrease service disruption. In order to achieve these two goals, the authors solve the VNF placement problem by using integer linear programming and a genetic algorithm in three network architectures. Experiment results demonstrate that their method achieves more resource reduction of VNF initial allocation and the configuration changes of VNF scaling than traditional integer linear programming approaches.

Luizelli et al. [57] minimize the required resource allocation by optimizing VNFs placement and chaining while ensuring the scalability of VNFs. They first express the optimization problem as a virtual network function placement and chaining (VNFPC) problem. Then, a repair and optimization–based heuristic algorithm is proposed to effectively explore the placement and chaining solution space. The heuristic algorithm combines ILP and variable neighborhood search (VNS) to generate low-demand resource solutions for large-scale network configurations. In particular, other optimization goals (e.g., network service latency) can be further taken into account when optimizing resource consumption. Experiment results have shown that even in the case of expansion to hundreds of VNFs, their method can effectively find a viable high-quality solution.

7.5 Discussion

Table 8 provides a comparison summary of various deployment cost optimization techniques from different aspects. In this section, different VNF deployment strategies of cloud providers have been studied in detail. A good VNF instance deployment strategy can increase resource utilization and greatly reduce deployment costs, thus bringing more profits to cloud providers. Generally speaking, the VNF deployment cost mainly consists of two parts,that is, the initial cost and the adjustment cost. The former refers to the resource consumption cost when the VNF is deployed for the first time, while the latter refers to the resource consumption cost incurred by the VNF adjustment when users' demand changes. Next, we summarize and analyze the VNF cost optimization techniques from these two parts.

The initial VNF deployment is a well-known NP-hard problem, thus, the main goal of most works in VNF deployment optimization problems is to find an approximate solution that is close to the optimal solution quickly and efficiently. Fortunately, there are many optimization methods that can solve the above NP-hard problems, such as machine learning methods, greedy strategy-based

Table 8. A Comparison Summary of Some Deployment Cost Optimization Techniques in Terms of Multiple Aspects, such as Method, Future Traffic, VM Migration/Scalability, Constraints, Optimization Goal, Decision, Cost Saving, and Benchmarks

Ref.	Method	Future traffic	VM migration/ Scalability	Constraint(s)
[105]	Decision tree	Known	No/No	Resource capacity, security
[71]	Graph neural network	Unknown	Yes/Yes	÷
[23]	Follow the regularized leader	Unknown	No/No	Bandwidth
[113]	Gradient descent method, ski-rental	Unknown	No/No	Flow service quality
[86]	Greedy strategy	Known	No/No	Flow processing time
[45]	Greedy strategy, dynamic programming	Unknown	Yes/Yes	Path, link/VM capacity, VNF placement, chaining
[89]	Monte Carlo tree search method	Known	No/Yes	Bandwidth
[74]	Sample-based Markov approximation	Known	No/No	Resource
[22]	Markov decision process theory	Unknown	Yes/Yes	Link, bandwidth, capacity
[7]	Affinity-based method	Known	No/Yes	Deployment cost and service level agreement
[97]	Minimal weight matching algorithm	Unknown	No/Yes	Server capacity and flow conservation
[110]	Priority driven algorithm	Known	No/No	Resource capacity
[77]	Genetic Algorithm	Unknown	Yes/Yes	Server capacity, physical network, link
[57]	Variable neighborhood search	Known	No/Yes	Network flow requirement
Ref.	Goal	Decision	Cost	Benchmarks
[105]	Deployment cost, service reliability	Static	−38.2%	No protection (NP)
[71]	Deployment cost, resource utilization	Dynamic	÷	÷
[23]	Deployment cost	Dynamic	−45%	Constant capacity allocation (CCA)
[113]	Deployment cost	Dynamic	÷	÷
[86]	Deployment cost	Static	÷	÷
[45]	Deployment cost	Dynamic	÷	÷
[89]	Energy Consumption	Dynamic	−90%	Non-energy-aware method
[74]	Deployment cost, operating cost	Dynamic	−19%	Non-coordinated approach
[22]	Deployment cost, energy cost	Dynamic	−27%	Benefit/cost evaluation
[7]	Deployment cost, overall latency	Dynamic	−62.5%	Greedy approach
[97]	Deployment cost	Dynamic	÷	÷
[110]	Resource utilization, overall latency	Static	−33.4%	Node assignment heuristic algorithm
[77]	Deployment cost	Dynamic	÷	÷
[57]	Deployment cost	Static	÷	÷

÷ implies that the literature does not give a clear explanation of the factor.

Table 9. A Classification of Profit Optimization Techniques Based on Application and Service Types

Application or service types	References
Data-intensive applications	[52], [46], [64], [37], [50], [92], [59], [27]
Web applications	[52], [46], [50], [92], [77]
Scientific workflows application	[82]
Large-scale data processing applications	[38], [103], [105]
Soft real-time applications	[19]
Cloud storage services	[46], [115]
Composite services	[48]
Throughput-intensive applications	[37], [50], [92]
Delay sensitive applications	[37], [50], [92], [59]
Computation-intensive applications	[36], [12]
Network security and analytic applications	[86]
Network services (e.g., load balancers and firewalls)	[71], [23], [113], [45], [89], [74], [22], [7], [97], [110]

methods, Monte Carlo–based and Markov model–based methods, and other optimization techniques. Since the prior art can well solve the NP-Hard problem of VNF deployment, multi-objective optimization of deployment cost and performance will be the main research issue in the VNF deployment domain. In particular, we also review some literature that achieves multi-objective optimization for VNF deployment under the premise of minimizing costs, such as from aspects of service reliability, resource utilization, and network latency. These works achieve a good balance between VNF deployment costs and performance under different scenarios. After the VNF is initially deployed, the fluctuation of network requests will make the original deployment scheme unable to meet user demand. Re-deploying VNF every time the network fluctuates is undoubtedly time-consuming and unrealistic. Thus, it is very crucial for cloud providers to minimize the adjustment cost while achieving adaptive adjustment of VNF in an environment in which the user demand is dynamically changed due to network fluctuations.

8 CONCLUSIONS

In this article, a survey on cloud provider profit optimization techniques is presented. In particular, Table 9 provides a classification of profit optimization techniques based on application and service types. We review and summarize profit optimization methods from aspects of service quality, service price, server energy consumption, and VNF deployment. At the end of this article, we briefly summarize the challenges in this area.

- The SLA signed with users forces cloud providers to pay fines due to low QoS, but the volatility of user demand makes it difficult for cloud providers to satisfy QoS of users at all times without wasting resources. This has prompted researchers to explore new resource usage strategies to meet user service quality without wasting resources.
- Cloud computing, as an attractive computing paradigm has facilitated the emergence and development of more and more cloud providers in the cloud service market. This forces cloud providers to develop pricing strategies by considering both market demand and peer competition, which is a huge challenge for cloud providers.
- Electricity fee is a crucial component of a service provider's profit. The large increase in user demand has made it extremely difficult to reduce infrastructure power consumption while ensuring service quality.

— The deployment of VNF as an NP-hard problem is difficult to solve when the demand of users for network increases. This motivates researchers to develop a large number of heuristic algorithms to get an approximate solution to VNF deployment.

These works demonstrate that by carefully designing service quality improvement, service price adjustment, server energy reduction, and VNF deployment schemes, the profit of cloud providers is controllable. However, taking into account multiple profit-affected factors, it is still challenging for profit optimization of cloud providers in the current cloud market.

REFERENCES

[1] 2018. Alibaba Cloud. Retrieved from https://cn.aliyun.com.

[2] 2018. AmazonEC2SpotInstances. Retrieved from http://aws.amazon.com/ec2/spot-instances/.

[3] 2018. Elastic Compute Service (ECS)_Service Level Agreement. Retrieved from http://terms.aliyun.com/legal-agreement/terms/suit_bu1_ali_cloud/suit_bu1_ali_cloud201909241949_62160.html?spm=a2c4g.11186623.2.11.65491d94fphcD5.

[4] Bernardetta Addis, Dallal Belabed, Mathieu Bouet, and Stefano Secci. 2015. Virtual network functions placement and routing optimization. In *IEEE CloudNet (2015)* , 171–177.

[5] Sahar Arshad, Saeed Ullah, Shoab Ahmed Khan, M. Daud Awan, and M. Sikandar Hayat Khayal. 2015. A survey of cloud computing variable pricing models. In *ENASE (2015)*.

[6] Md Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba. 2015. On orchestrating virtual network functions. In *CNSM (2015)*, 50–56.

[7] Deval Bhamare, Mohammed Samaka, Aiman Erbad, Raj Jain, Lav Gupta, and H. Anthony Chan. 2017. Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications* 102 (2017), 1–16.

[8] Junwei Cao, Kai Hwang, Keqin Li, and Albert Y. Zomaya. 2012. Optimal multiserver configuration for profit maximization in cloud computing. *IEEE Transactions on Parallel and Distributed Systems* 24, 6 (2012), 1087–1096.

[9] Junwei Cao, Keqin Li, and Ivan Stojmenovic. 2013. Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. *IEEE Transactions on Computers* 63, 1 (2013), 45–58.

[10] Marcus Carvalho, Walfredo Cirne, Francisco Brasileiro, and John Wilkes. 2014. Long-term SLOs for reclaimed cloud computing resources. In *ACM SoCC (2014)*, 1–13.

[11] Sameer Singh Chauhan, Emmanuel S. Pilli, R.C. Joshi, Girdhari Singh, and M.C. Govil. 2019. Brokering in interconnected cloud computing environments: A survey. *Journal of Parallel and Distributed Computing* 133 (2019), 193–209.

[12] Chen Chen, Wei Wang, and Bo Li. 2018. Performance-aware fair scheduling: Exploiting demand elasticity of data analytics jobs. In *IEEE INFOCOM (2018)*, 504–512.

[13] Junliang Chen, Chen Wang, Bing Bing Zhou, Lei Sun, Young Choon Lee, and Albert Y. Zomaya. 2011. Tradeoffs between profit and customer satisfaction for service provisioning in the cloud. In *International Symposium on High Performance Distributed Computing (2011)*, 229–238.

[14] Peijin Cong, Liying Li, Gaoyuan Shao, Junlong Zhou, Mingsong Chen, Kai Huang, and Tongquan Wei. 2017. User perceived value-aware cloud pricing for profit maximization of multiserver systems. *IEEE ICPADS (2017)* , 537–544.

[15] Peijin Cong, Liying Li, Junlong Zhou, Tongquan Wei, Mingsong Chen, and Shiyan Hu. 2018. Developing user perceived value based pricing models for cloud markets. *IEEE Transactions on Parallel and Distributed Systems* 29, 12 (2018), 2742–2756.

[16] Lin Cui, Richard Cziva, Fung Po Tso, and Dimitrios P. Pezaros. 2016. Synergistic policy and virtual machine consolidation in cloud data centers. In *IEEE INFOCOM (2016)*, 1–9.

[17] Lin Cui, Fung Po Tso, Dimitrios P. Pezaros, Weijia Jia, and Wei Zhao. 2016. Plan: Joint policy-and network-aware VM management for cloud data centers. *IEEE Transactions on Parallel and Distributed Systems* 28, 4 (2016), 1163–1175.

[18] Bingqian Du, Chuan Wu, and Zhiyi Huang. 2019. Learning resource allocation and pricing for cloud profit maximization. In *AAAI (2019)*.

[19] Yuhuan Du and Gustavo De Veciana. 2017. Scheduling for cloud-based computing systems to support soft real-time applications. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 2, 3 (2017), 13.

[20] Nnamdi Ekwe-Ekwe and Adam Barker. 2018. Location, location, location: Exploring Amazon EC2 spot instance pricing across geographical regions - extended version. *arXiv:1807.10507v1*.

[21] Abdessalam Elhabbash, Faiza Samreen, James Hadley, and Yehia Elkhatib. 2019. Cloud brokerage: A systematic survey. *ACM Computing Surveys* 51, 6 (2019).

[22] Vincenzo Eramo, Emanuele Miucci, Mostafa Ammar, and Francesco Giacinto Lavacca. 2017. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking* 25, 4 (2017), 2008–2025.
[23] Xincai Fei, Fangming Liu, Hong Xu, and Hai Jin. 2018. Adaptive VNF scaling and flow routing with proactive demand prediction. In *IEEE INFOCOM (2018)*, 486–494.
[24] Manoel C. Silva Filho, Claudio C. Monteiro, Pedro R. M. Incio, and Mrio M. Freire. 2018. Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. *Journal of Parallel and Distributed Computing* 111 (2018), 222–250.
[25] Mohamed Firdhous, Suhaidi Hassan, and Osman Ghazali. 2013. A comprehensive survey on quality of service implementations in cloud computing. *International Journal of Scientific and Engineering Research* 4, 5 (2013), 118–123.
[26] Jing Fu, Jun Guo, Eric W. M. Wong, and Moshe Zukerman. 2015. Energy-efficient heuristics for job assignment in processor-sharing server farms. In *IEEE INFOCOM (2015)*, 882–890.
[27] Keke Gai, Meikang Qiu, and Hui Zhao. 2016. Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. *IEEE Transactions on Cloud Computing* (2016). DOI: 10.1109/TCC.2016.2594172
[28] Javad Ghaderi. 2016. Randomized algorithms for scheduling VMs in the cloud. *IEEE INFOCOM (2016)*, 1–9.
[29] Mahdi Ghamkhari and Hamed Mohsenian-Rad. 2013. Energy and performance management of green data centers: A profit maximization approach. *IEEE Transactions on Smart Grid* 4, 2 (2013), 1017–1025.
[30] Atul Gohad, Nanjangud C. Narendra, and Parathasarthy Ramachandran. 2013. Cloud pricing models: A survey and position paper. In *IEEE CCEM (2013)*.
[31] Íñigo Goiri, Jordi Guitart, and Jordi Torres. 2012. Economic model of a cloud provider operating in a federated cloud. *Information Systems Frontiers* 14, 4 (2012), 827–843.
[32] Íñigo Goiri, Ferran Julià, J. Oriol Fitó, Mario Macías, and Jordi Guitart. 2012. Supporting CPU-based guarantees in cloud SLAs via resource-level QoS metrics. *Future Generation Computer Systems* 28, 8 (2012), 1295–1302.
[33] Yang Guo, Alexander L. Stolyar, and Anwar Walid. 2015. Shadow-routing based dynamic algorithms for virtual machine placement in a network cloud. *IEEE Transactions on Cloud Computing* 6, 1 (2015), 209–220.
[34] Sheikh Mahbub Habib, Sascha Hauke, Sebastian Ries, and Max Mühlhäuser. 2012. Trust as a facilitator in cloud computing: A survey. *Journal of Cloud Computing: Advances, Systems and Applications* 1, 1 (2012), 1–18.
[35] Abdul Hameed, Alireza Khoshkbarforoushha, et al. 2016. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing* 98, 7 (2016), 751–774.
[36] Qiang He, Xiaodong Zhu, Dongwei Li, Shuliang Wang, Jun Shen, and Yun Yang. 2017. Cost-effective big data mining in the cloud: A case study with K-means. In *IEEE CLOUD (2017)*, 74–81.
[37] Shuihai Hu, Wei Bai, Kai Chen, Chen Tian, Ying Zhang, and Haitao Wu. 2018. Providing bandwidth guarantees, work conservation and low latency simultaneously in the cloud. In *IEEE Transactions on Cloud Computing (2018)*.
[38] Zhe Huang, Bharath Balasubramanian, Michael Wang, Tian Lan, Mung Chiang, and Danny HK Tsang. 2015. Need for speed: Cora scheduler for optimizing completion-times in the cloud. In *IEEE INFOCOM (2015)*, 891–899.
[39] Joe Wenjie Jiang, Tian Lan, Sangtae Ha, Minghua Chen, and Mung Chiang. 2012. Joint VM placement and routing for data center traffic engineering. In *IEEE INFOCOM (2012)*, 2876–2880.
[40] Issa M. Khalil, Abdallah Khreishah, and Muhammad Azeem. 2014. Cloud computing security: A survey. *Computers* 3, 1 (2014), 1–35.
[41] Abdul Nasir Khan, M. L. Mat Kiah, Samee U. Khan, and Sajjad A. Madani. 2013. Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems* 29, 5 (2013), 1278–1299.
[42] Minhaj Ahmad Khan. 2016. A survey of security issues for cloud computing. *Journal of Network and Computer Applications* 71 (2016), 11–29.
[43] Mikhail Khodak, Liang Zheng, Andrew S. Lan, Carlee Joe-Wong, and Mung Chiang. 2018. Learning cloud dynamics to optimize spot instance bidding strategies. In *IEEE INFOCOM (2018)*, 2762–2770.
[44] Dinesh Kumar, Gaurav Baranwal, Zahid Raza, and Deo Prakash Vidyarthi. 2018. A survey on spot pricing in cloud computing. *Journal of Network and Systems Management* 26, 4 (2018), 809–856.
[45] Tung-Wei Kuo, Bang-Heng Liou, Kate Ching-Ju Lin, and Ming-Jer Tsai. 2018. Deploying chains of virtual network functions: On the relation between link and server usage. *IEEE/ACM Transactions on Networking* 26, 4 (2018), 1562–1576.
[46] Zeqi Lai, Yong Cui, Minming Li, Zhenhua Li, Ningwei Dai, and Yuchi Chen. 2016. TailCutter: Wisely cutting tail latency in cloud CDN under cost constraints. In *IEEE INFOCOM (2016)*, 1–9.
[47] Kien Le, Ricardo Bianchini, Jingru Zhang, Yogesh Jaluria, Jiandong Meng, and Thu D. Nguyen. 2011. Reducing electricity cost through virtual machine placement in high performance computing clouds. In *SC(2001)* .
[48] Young Choon Lee, Chen Wang, Albert Y. Zomaya, and Bing Bing Zhou. 2012. Profit-driven scheduling for cloud services with data access awareness. *Journal of Parallel and Distributed Computing* 72, 4 (2012), 591–602.

[49] Defang Li, Peilin Hong, Kaiping Xue, et al. 2018. Virtual network function placement considering resource optimization and SFC requests in cloud datacenter. *IEEE Transactions on Parallel and Distributed Systems* 29, 7 (2018), 1664–1677.
[50] Fuliang Li, Jiannong Cao, Xingwei Wang, Yinchu Sun, and Yuvraj Sahni. 2017. Enabling software defined networking with QoS guarantee for cloud applications. In *IEEE CLOUD (2017)*, 130–137.
[51] Xin Li, Jie Wu, Shaojie Tang, and Sanglu Lu. 2014. Let's stay together: Towards traffic aware virtual machine placement in data centers. In *IEEE INFOCOM (2014)*, 1842–1850.
[52] Guoxin Liu, Haiying Shen, and Haoyu Wang. 2017. An economical and SLO-guaranteed cloud storage service across multiple cloud service providers. *IEEE Transactions on Parallel and Distributed Systems* 28, 9 (2017), 2440–2453.
[53] Lin Liu, Yuchen Zhou, Yang Liu, and Shiyan Hu. 2014. Dynamic programming based game theoretic algorithm for economical multi-user smart home scheduling. In *MWSCAS (2014)*.
[54] Shuo Liu, Shaolei Ren, Gang Quan, Ming Zhao, and Shangping Ren. 2013. Profit aware load balancing for distributed cloud data centers. *International Symposium on Parallel and Distributed Processing (2013)*, 611–622.
[55] Yang Liu and Shiyan Hu. 2015. Cyberthreat analysis and detection for energy theft in social networking of smart homes. *IEEE Transactions on Computational Social Systems* 2, 4 (2015), 148–158.
[56] Zhuotao Liu, Kai Chen, Haitao Wu, Shuihai Hu, Yih-Chun Hut, Yi Wang, and Gong Zhang. 2018. Enabling work-conserving bandwidth guarantees for multi-tenant datacenters via dynamic tenant-queue binding. In *IEEE INFOCOM (2018)*, 1–9.
[57] Marcelo Caggiani Luizelli, Weverton Luis da Costa Cordeiro, Luciana S Buriol, and Luciano Paschoal Gaspary. 2017. A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. *Computer Communications* 102 (2017), 67–77.
[58] Nguyen Cong Luong, Ping Wang, Dusit Niyato, Yonggang Wen, and Zhu Han. 2017. Resource management in cloud networking using economic analysis and pricing models: A survey. *IEEE Communications Surveys and Tutorials* 19, 2 (2017), 954–1001.
[59] Mario Macías and Jordi Guitart. 2011. A genetic model for pricing in cloud computing markets. In *ACM Symposium on Applied Computing (2011)*, 113–118.
[60] Sunilkumar S. Manvi and Gopal Krishna Shyam. 2014. Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications* 41 (2014), 424–440.
[61] Parisa Jalili Marandi, Christos Gkantsidis, Flavio Junqueira, and Dushyanth Narayanan. 2016. Filo: Consolidated consensus as a cloud service. In *USENIX Annual Technical Conference (USENIX ATC'16)*, 237–249.
[62] Mohammad Masdari, Farbod Salehi, Marzie Jalali, and Moazam Bidaki. 2017. A survey of PSO-based scheduling algorithms in cloud computing. *Journal of Network and Systems Management* 25, 1 (2017), 122–158.
[63] Lena Mashayekhy, Mahyar Movahed Nejad, and Daniel Grosu. 2013. A truthful approximation mechanism for autonomic virtual machine provisioning and allocation in clouds. In *ACM CAC (2013)*.9.
[64] Lena Mashayekhy, Mahyar Movahed Nejad, and Daniel Grosu. 2014. Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing* 3, 1 (2014), 14–27.
[65] Michael Mattess, Christian Vecchiola, and Rajkumar Buyya. 2010. Managing peak loads by leasing cloud infrastructure services from a spot market. In *IEEE HPCC (2010)*, 180–188.
[66] Sevil Mehraghdam, Matthias Keller, and Holger Karl. 2014. Specifying and placing chains of virtual network functions. In *IEEE CloudNet (2014)*, 7–13.
[67] Jing Mei, Kenli Li, Jingtong Hu, Shu Yin, and Edwin H.-M. Sha. 2013. Energy-aware preemptive scheduling algorithm for sporadic tasks on DVS platform. *Microprocessors and Microsystems* 37, 1 (2013), 99–112.
[68] Jing Mei, Kenli Li, and Keqin Li. 2017. Customer-satisfaction-aware optimal multiserver configuration for profit maximization in cloud computing. *IEEE Transactions on Sustainable Computing* 2, 1 (2017), 17–29.
[69] Jing Mei, Kenli Li, Aijia Ouyang, and Keqin Li. 2015. A profit maximization scheme with guaranteed quality of service in cloud computing. *IEEE Transactions on Computers* 64, 11 (2015), 3064–3078.
[70] Xiaoqiao Meng, Canturk Isci, Jeffrey Kephart, Li Zhang, Eric Bouillet, and Dimitrios Pendarakis. 2010. Efficient resource provisioning in compute clouds via VM multiplexing. In *IEEE ICAC (2010)*, 11–20.
[71] Rashid Mijumbi, Sidhant Hasija, Steven Davy, Alan Davy, Brendan Jennings, and Raouf Boutaba. 2016. A connectionist approach to dynamic resource management for virtualised network functions. In *CNSM (2016)*, 1–9.
[72] Mahyar Movahed Nejad, Lena Mashayekhy, and Daniel Grosu. 2013. A family of truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. In *IEEE CLOUD (2013)*, 188–195.
[73] Debdeep Paul, Wen-De Zhong, and Sanjay K. Bose. 2016. Energy efficient cloud service pricing: A two-timescale optimization approach. *Journal of Network and Computer Applications* 64 (2016), 98–112.
[74] Chuan Pham, Nguyen H. Tran, Shaolei Ren, Walid Saad, and Choong Seon Hong. 2017. Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach. *IEEE Transactions on Services Computing* 13, 1 (2017), 172–185.

[75] Chenxi Qiu, Haiying Shen, and Liuhua Chen. 2016. Probabilistic demand allocation for cloud service brokerage. In *IEEE INFOCOM (2016)*, 1–9.
[76] Asfandyar Qureshi, Rick Weber, and Hari Balakrishnan. 2009. Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM (2009)*, 123–134.
[77] Windhya Rankothge, Franck Le, Alessandra Russo, and Jorge Lobo. 2017. Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms. *IEEE Transactions on Network and Service Management* 14, 2 (2017), 343–356.
[78] Chuangang Ren, Di Wang, Bhuvan Urgaonkar, and Anand Sivasubramaniam. 2012. Carbon-aware energy capacity planning for datacenters. In *MASCOTS (2012)*, 391–400.
[79] Shaolei Ren and Yuxiong He. 2013. Coca: Online distributed resource management for cost minimization and carbon neutrality in data centers. In *SC (2013)*, 39.
[80] Benny Rochwerger, David Breitgand, Amir Epstein, David Hadas, Irit Loy, Kenneth Nagin, Johan Tordsson, Carmelo Ragusa, Massimo Villari, Stuart Clayman, et al. 2011. Reservoir-when one cloud is not enough. *Computer* 44, 3 (2011), 44–51.
[81] Benny Rochwerger, David Breitgand, Eliezer Levy, Alex Galis, Kenneth Nagin, Ignacio Martín Llorente, Rubén Montero, Yaron Wolfsthal, Erik Elmroth, Juan Caceres, et al. 2009. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development* 53, 4 (2009), 4:1–4:11.
[82] Maria Alejandra Rodriguez and Rajkumar Buyya. 2014. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing* 2, 2 (2014), 222–235.
[83] Maria Alejandra Rodriguez and Rajkumar Buyya. 2017. A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments. *Concurrency and Computation: Practice and Experience* 29, 8 (2017).
[84] Zvi Rosberg, Yu Peng, Jing Fu, Jun Guo, W. M. Eric Wong, and Moshe Zukerman. 2014. Insensitive job assignment with throughput and energy criteria for processor-sharing server farms. *IEEE/ACM Transactions on Networking* 22, 4 (2014), 1257–1270.
[85] Nancy Samaan. 2013. A novel economic sharing model in a federation of selfish cloud providers. *IEEE Transactions on Parallel and Distributed Systems* 25, 1 (2013), 12–21.
[86] Yu Sang, Bo Ji, Gagan R. Gupta, Xiaojiang Du, and Lin Ye. 2017. Provably efficient algorithms for joint placement and allocation of virtual network functions. In *IEEE INFOCOM (2017)*, 1–9.
[87] Sukhpal Singh and Inderveer Chana. 2016. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing* 14, 2 (2016), 217–264.
[88] Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park. 2016. A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications* 75 (2016), 200–222.
[89] Oussama Soualah, Marouen Mechtri, Chaima Ghribi, and Djamal Zeghlache. 2017. Energy efficient algorithm for VNF placement and chaining. In *IEEE/ACM CCGrid (2017)*, 579–588.
[90] Subashini Subashini and Veeraruna Kavitha. 2011. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications* 34, 1 (2011), 1–11.
[91] Fung Po Tso, Konstantinos Oikonomou, Eleni Kavvadia, and Dimitrios P. Pezaros. 2014. Scalable traffic-aware virtual machine management for cloud data centers. In *IEEE ICDCS (2014)*, 238–247.
[92] Jianxiong Wan, Ran Zhang, Xiang Gui, and Baoqing Xu. 2016. Reactive pricing: An adaptive pricing policy for cloud providers to maximize profit. *IEEE Transactions on Network and Service Management* 13, 4 (2016), 941–953.
[93] Juntao Wang, Xun Xiao, Jianping Wang, Kejie Lu, Xiaotie Deng, and Ashwin A. Gumaste. 2016. When group-buying meets cloud computing. In *IEEE INFOCOM (2016)*, 1–9.
[94] Qian Wang, Kui Ren, and Xiaoqiao Meng. 2012. When cloud meets ebay: Towards effective pricing for cloud computing. In *IEEE INFOCOM (2012)*, 936–944.
[95] Tian Wang, Junlong Zhou, Gongxuan Zhang, Tongquan Wei, and Shiyan Hu. 2019. Customer perceived value- and risk-aware multiserver configuration for profit maximization. *IEEE Transactions on Parallel and Distributed Systems* 31, 5 (2019), 1074–1088.
[96] Wei Wang, Ben Liang, and Baochun Li. 2013. Revenue maximization with dynamic auctions in IaaS cloud markets. In *International Symposium on Quality of Service (2013)*, 1–6.
[97] Xiaoke Wang, Chuan Wu, Franck Le, Alex Liu, Zongpeng Li, and Francis Lau. 2016. Online VNF scaling in datacenters. In *IEEE CLOUD (2016)*, 140–147.
[98] Usman Wazir, Fiaz Gul Khan, and Sajid Shah. 2016. Service level agreement in cloud computing: A survey. *International Journal of Computer Science and Information Security* 14, 6 (2016), 324–330.
[99] Rich Wolski and John Brevik. 2017. QPRED: Using quantile predictions to improve power usage for private clouds. In *IEEE CLOUD (2017)*, 179–187.

[100] Caesar Wu, Rajkumar Buyya, and Kotagiri Ramamohanarao. 2019. Cloud pricing models: Taxonomy, survey and interdisciplinary challenges. *ACM Computing Surveys* 52, 6 (2019).
[101] Bolei Xu, Tao Qin, Guoping Qiu, and Tie-Yan Liu. 2015. Optimal pricing for the competitive and evolutionary cloud market. In *IJCAI (2015)*.
[102] Hong Xu and Baochun Li. 2012. Anchor: A versatile and efficient framework for resource management in the cloud. *IEEE Transactions on Parallel and Distributed Systems* 24, 6 (2012), 1066–1076.
[103] Hong Xu and Baochun Li. 2013. Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing* 1, 2 (2013), 158–171.
[104] Fan Yao, Jingxin Wu, Suresh Subramaniam, and Guru Venkataramani. 2017. WASP: Workload adaptive energy-latency optimization in server farms using server low-power states. In *IEEE CLOUD (2017)*, 171–178.
[105] Zilong Ye, Xiaojun Cao, Jianping Wang, Hongfang Yu, and Chunming Qiao. 2016. Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization. *IEEE Network* 30, 3 (2016), 81–87.
[106] Lei Yu and Zhipeng Cai. 2016. Dynamic scaling of virtual clusters with bandwidth guarantee in cloud datacenters. In *IEEE INFOCOM (2016)*, 1–9.
[107] Zhi-hui Zhan, Xiao-fang Liu, Yue-jiao Gong, Jun Zhang, Henry Shu-hung Chung, and Yun Li. 2015. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys* 47, 4 (2015).
[108] Hong Zhang, Hongbo Jiang, Bo Li, Fangming Liu, Athanasios V. Vasilakos, and Jiangchuan Liu. 2015. A framework for truthful online auctions in cloud computing with heterogeneous user demands. *IEEE Transactions on Computers* 65, 3 (2015), 805–818.
[109] Jiangtao Zhang, Hejiao Huang, and Xuan Wang. 2016. Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications* 64 (2016), 23–42.
[110] Qixia Zhang, Yikai Xiao, Fangming Liu, John C. S. Lui, Jian Guo, and Tao Wang. 2017. Joint optimization of chain placement and request scheduling for network function virtualization. In *ICDCS (2017)*, 731–741.
[111] Qi Zhang, Quanyan Zhu, and Raouf Boutaba. 2011. Dynamic resource allocation for spot markets in cloud computing environments. In *UCC (2011)*, 178–185.
[112] Shuo Zhang, Li Pan, Shijun Liu, Lei Wu, Lizhen Cui, and Dong Yuan. 2016. An optimal and iterative pricing model for multiclass IaaS cloud services. In *ICSOC (2016)*, 597–605.
[113] Xiaoxi Zhang, Chuan Wu, Zongpeng Li, and Francis C. M. Lau. 2017. Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization. In *IEEE INFOCOM (2017)*, 1–9.
[114] Xiaoxi Zhang, Chuan Wu, Zongpeng Li, and Francis C. M. Lau. 2018. A truthful (1-ϵ)-optimal mechanism for on-demand cloud resource provisioning. *IEEE Transactions on Cloud Computing* (2018). DOI: 10.1109/TCC.2018.2822718
[115] Jian Zhao, Xiaowen Chu, Hai Liu, Yiu-Wing Leung, and Zongpeng Li. 2015. Online procurement auctions for resource pooling in client-assisted cloud storage systems. In *IEEE INFOCOM (2015)*, 576–584.
[116] Jian Zhao, Hongxing Li, Chuan Wu, Zongpeng Li, Zhizhong Zhang, and Francis C. M. Lau. 2014. Dynamic pricing and profit maximization for the cloud with geo-distributed data centers. In *IEEE INFOCOM (2014)*, 118–126.
[117] Zizhan Zheng and Ness B. Shroff. 2016. Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud. In *IEEE INFOCOM (2016)*, 1–9.

Received June 2019; revised November 2019; accepted December 2019