Wireless Networks
https://doi.org/10.1007/s11276-021-02757-1

Total cost ownership optimization of private clouds: a rack minimization perspective

Yuanfang Chi[1] · Wei Dai[2,3] · Yuan Fan[1] · Jun Ruan[1] · Kai Hwang[2,3] · Wei Cai[2,3]

Accepted: 28 July 2021

Abstract
With the development of network infrastructure, a large volume of data will be exchanged with increased bandwidth. Many applications are connecting people to the rest of the world through the public network. Thus, privacy and security have become a concern. Under this circumstance, it becomes a trend that the enterprises tend to host their data and services on private clouds dedicated to their own use, rather than the public cloud services. However, in contrary to the well-investigated total cost of ownership (TCO) for public clouds, the analytic research on the cost of purchase and operation for private clouds is still a blank. In this work, we first review the state-of-the-art TCO literature to summarize the models, tools, and cost optimization techniques for public clouds. Based on our survey, we envision the TCO modeling and optimization for private clouds by comparing the differences of features between public and private clouds. Finally, we propose a heuristic algorithm, conflict-aware first-fit to optimize the total cost of ownership of private cloud by minimizing the number of racks when deploying servers.

Keywords Total cost of ownership · Cloud computing · Rack optimization · Private cloud

1 Introduction

Nowadays, cloud computing has been widely adopted by startups and well-established enterprises, thanks to its prominent elastic and on-demand features. The flexibility of on-demand usage reduces development, service deployment, and maintenance costs. Huge data computation, backup, and recovery tasks have become easier with the cloud. The cloud computing market is expanding. Many enterprises will choose to invest in cloud services [1] to enjoy cloud computing benefits, but risks often accompany the investment. For a company that provides cloud services or uses cloud services, maintaining development and revenue is also important. On the one hand, operating an enterprise with cloud-based IT infrastructure can improve the enterprise's stability and bring intuitive benefits. Several cloud computing business models, such as the VE model [2] and cloud business model framework (CBMF) [3] have been reported in the financial and business domain. More discussions and studies about business models can be found at [4–10]. On the other hand, analyzing Return On Investment (ROI) can also increase investment value. ROI analysis methods in cloud computing have been discussed in [11–13]. Cost analysis and optimization modelling can effectively help decision-making and increase profit under the premise of guaranteeing service quality [14].

From the users' perspective, the cost efficiency is always a key metric when considering cloudization. On the other hand, cloud providers also need to evaluate their cloud business profits. Hence, the total cost of ownership (TCO),

This paper was presented in part at EAI 6GN 2020: 3rd EAI International Conference on 6G for Future Wireless Network, Tianjin, China, Aug. 2020. The manuscript extends to demonstrate a cost model for the private cloud and a heuristic solution for the rack minimization problem to optimize the total cost of the private cloud.

This work was supported by Project 61902333 supported by National Natural Science Foundation of China, by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS).

✉ Wei Cai
caiwei@cuhk.edu.cn

[1] Alibaba Group, Beijing, China

[2] The Chinese University of Hong Kong, Shenzhen, China

[3] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China

Published online: 02 September 2021

a standard approach in analyzing the cost of purchase and operation of an asset, is a critical topic for both parties. By definition, the TCO involves all direct and indirect costs, such as the cost of cloud service providers purchasing, deploying, operating, maintaining assets, and the cost of cloud users renting cloud resources. Due to the importance of TCO, the analysis and optimization methods for public cloud and hybrid cloud have been well investigated. However, few works have been devoted to the cost analysis associated with private clouds. In addition, many cost optimization problems wildly exist in the real industry. In this work, we compare and contrast the similarities and differences between public and private clouds from a TCO perspective and demonstrate a server rack deployment problem. We further propose a heuristic algorithm to minimize the number of racks used.

The main contribution of our work is highlighted in the following:

1. We investigate the TCO analysis and optimization methods. By comparing and contrasting the similarities and differences between public and private clouds, we demonstrate a TCO evaluation method for the private cloud.
2. We tackle the novel and practical problem of rank minimization in private data center. To the best of our knowledge, it is the first work to minimize the total cost by minimizing the number of racks used in private cloud.
3. We present a novel heuristic scheme, conflict-aware first-fit (CAFF), to find a near optimal solution for the rack minimization problem. Extensive simulation experiments reveal that our method can find a feasible solution in a polynomial execution time.

The remainder of the paper is organized as follows. We present an overall review of existing works on TCO in Sect. 2. Afterwards, we compare the public and private cloud from TCO optimization perspective in Sect. 3. In Sects.4 and 5, we show a cost evaluation model and illustrate the optimization we want to solve. A heuristic method is proposed in Sect. 6. Finally, 8 concludes the work and envision the TCO research for private clouds.

2 Literature review

To draw a light on how to evaluate and optimize the TCO of private cloud for readers, we first survey the literature on public cloud TCO analysis methods and the optimization solutions that minimize the TCO of a cloud service.

TCO was first being studied in the business domain by Ellram in 1993 [15]. She claimed that TCO modeling is constructive for business parchment decision making, ongoing supplier management, and understanding of indirect costs. However, she also pointed out that the quantification and measurement of TCO is complex [16]. In 1999, Milligan suggested that accurate total cost measurement is elusive due to the lack of analysis methods [17]. Thus, one of the leading research directions in TCO is to find an accurate modeling and measurement method. Later, Degraeve and Roodhooft illustrated the application of TCO in the supply chain domain. They split purchasing activities into three levels corresponding to three different expenses. Then, they built a mathematical model to optimally select suppliers such that the total cost of ownership is minimized [18]. As the IT industry rising in 2004, few scholars began to study TCO in the information technology domain. They have worked on topics such as revenue analysis in IT by taking advantage of TCO [11], how to utilize TCO to decide whether to adopt open-source software in IT companies [19, 20], and optimization models to decrease the cost of IT infrastructure [21]. After cloud computing entered the market, some researchers analyzed the business value of cloud computing and mentioned that cloud providers should pay attention to the cost [22], while few studies took the modeling of TCO in cloud computing into consideration. Although Patel and Shah presented a cost model to calculate the cost of building a data center, they did not take indirect costs such as maintenance cost, operation cost, and labor cost into consideration [23]. TCO in cloud computing was finally studied by Li et al. in 2009. A cloud cost amortization model is built to make it possible to calculate the direct and indirect costs of cloud computing infrastructures. Their model can be divided into eight parts: server cost, software cost, network cost, support and maintenance cost, power cost, cooling cost, facilities cost, and real-estate cost. They also implemented an interactive tool for cloud providers to calculate cloud TCO [24]. Scholars begin to pay much more attention to models and tools to analyze TCO in cloud computing and the optimization to reduce TCO.

2.1 TCO from cloud service provider's perspective

The cost has been known as one of the most important factors for cloud providers, especially there is significant capital consumption, such as cloud data center construction, service software development and maintenance costs. So, cost evaluation will be helpful for revenue estimation, investment risk analysis and decision making.

The primary purpose of cloud total cost of ownership breakdown research is to provide cloud providers with a measurement of cost. It focuses on the modeling approaches of server cost, software cost, network cost, support

and maintenance cost, power cost, cooling cost, facilities cost, and real-estate cost.

In 2009, Li et al., as the pioneers, first solved the problem that the lack of a method to measure TCO. They proposed a cloud cost amortization model to calculate the total cost, and further use the model to find VM utilization cost given the number of running VMs [24]. Later, researchers studied cloud TCO breakdown under different business models. In 2015, Filiopoulou et al. applied the system of system (SoS) method to reorganize each part of the total cost. They regraded each type of cost as a subsystem of cloud computing and illustrated a cost modeling framework for each subsystem [25]. In 2016, Simonet et al. demonstrated a TCO breakdown modeling method for distributed cloud computing (DCC). Besides, they categorized DCC actors into five levels and presented related cost models [26]. It provides a detailed reference for cloud providers who intend to invest distributed could computing.

Through the above studies, researchers showed an overview of cloud TCO breakdown. However, they either overlooked some detailed cost measurements for some terms in the formulation or overlooked the cost calculation under some specific business scenarios. In real business, the cost can be significantly altered by different business scenarios and other dynamic factors such as application migration, timely changed resource utilization rate, and satisfactory network dependability and availability. In 2013, Sun and Li illustrated a labor cost model to calculate labor efforts in a service migration scenario. They quantified the skill level of employees and built a probabilistic model to estimate the number of persons needed per day to finish migration [27]. Omana et al. proposed an analyzing method for cloud providers to determine whether to replace aged assets. By measuring the cost of power, cooling, physical space, asset attachments, and IT support, they utilized statistical methods to examine the relationship between asset cost and resource capacities, such as the number of CPU cores and memory. They finally mined some useful results to help decision making and device management [1]. Thanakornworakij et al. did a cloud TCO research on the premise of ensuring system availability and satisfying service-level agreement (SLA), and quality of service (QoS). They modeled the cost of servers, network, software, power, cooling, facilities, maintenance, and availability. Then, they obtained the relationship between the number of devices needed and revenue under 99% system availability, which is beneficial for cloud providers to define the right size of the data center [28]. Sousa et al. built a detailed maintenance cost model and derived a Stochastic Model Generator for Cloud Infrastructure Planning (SMG4CIP) which concerned both dependability and cost requirements [29, 30].

Furthermore, some studies presented modeling methods for dynamic factors in cloud TCO such as resource utilization cost. Although [24] tried to use the ratio of running VMs to model utilization cost at a given time point, they can only get a rough result because the cost of running VMs depends on other frequently changed variables like CPU rate. As a follow-up study of [24], in 2013, Vrček and Brumec pointed out that many cloud TCO related studies ignored some hidden variables to build cost models, such as CPU utilization rate, data transmission rate, system load. They utilized CPU rate per hour to build a cost model and analyzed how the CPU rate affects the total cost in cloud computing [31]. Their study offered some vital insight into cloud computing TCO. Since time dimension can be added into the cost model, cloud providers can generate flexible strategy by analyzing per hour cost. Molka and Byrne had a similar idea that an accurate prediction model of resource utilization rate can derive an accurate utilization cost calculation. They demonstrated a prediction model using an online nonlinear autoregressive method and then formulated a model to calculate the real-time cost using CPU utilization data [32]. In 2017, Singh et al. also took account of the resource utilization rate to calculate utilization cost. They captured the effect of the relinquishment of cost and revenue [33].

With the development of cloud computing and its convenience, the energy and electricity power consumed by cloud data center per year increased dramatically [34], which leads to a higher expense to operate a data center. An investigation indicated that energy cost account for 42% of all expenditure per month in data center until 2011 [35]. Thus, energy cost, as part of the total cost of ownership, gradually becomes a focus of cloud providers' concern. In 2010, Yu and Bhatti thought [24] ignored approaches to gain specific values in the energy cost formulas. They pointed out there is a lack of an information model that collects energy usage and resource usage data for devices for cloud providers to manage assets. They proposed a Scalable Energy Monitor (SEM) architecture to measure energy consumption and further to calculate energy cost [36]. Later in 2012, Uchechukwu et al. studied a detailed energy cost modeling method. They divided energy into static and non-static parts where energy consumed by storage, computation, and communication is modeled. By analyzing cost using their measurement, they found it is possible to save energy costs [37]. In the same year, Chen et al. had a similar idea. They built a model with a static part and a dynamic part, and further developed an analytic tool to measure and summarize energy consumed by different tasks [38]. Some researchers focus on establishing more advanced models. In 2018, Jawad et al. illustrated a smart Power Management Model (PMM) to schedule power by adapting a Nonlinear Autoregressive

Network with Exogenous Inputs and Neural Network based forecasting algorithm (NARX-NN) [39]. Recently, in 2021, Hu et al. considered a two-stage online algorithm with Bernstein approximation to minimize the energy cost [40].

2.2 TCO from the cloud users' perspective

In general, cloud users are small companies or startups who rent cloud services from cloud providers to operate their businesses. There are three types of service in cloud computing: Infrastructure as a Service (IaaS) that provides computation power and data storage, Platform as a Service (PaaS) that provides developers platforms and Software as a Service (SaaS) that allows users access software service by light-weight clients [6]. Besides, there are different cloud providers in the market with different pricing models [41], such as Amazon EC2, Microsoft Azure, Google Compute Platform. It is not trivial for cloud users to choose the optimal leasing solution from the intricate market. Several representative papers are selected to illustrate methods and tools to analyze TCO from the perspective of cloud users.

In 2009, Kondo et al. first found monthly cost can be decreased drastically by deploying tasks on cloud servers instead of on Volunteer Computing (VC) platform [42]. Later in 2011, Han introduced the concept of cloud TCO from the perspective of cloud users. He presented a detailed analysis of TCO by comparing cloud service with local storage and servers [43]. Inspired by [42, 43], in 2012, Martens et al. first contributed the TCO measurement for cloud users. They summarized cost variables to be considered by cloud users when renting cloud services, and proposed cost models for IaaS, PaaS and SaaS [14].In 2013, Martens and his team continued their research on TCO models. They demonstrated a deployment planning strategy for cloud users and presented more detailed cost models for IaaS, PaaS and SaaS [44].

Additionally, some researchers focused on developing tools and methods that compare different cloud providers and estimate resources acquired by users' services or applications to further provide an optimal leasing plan . Liew and Su proposed a tool called CouldGuide which can predict the cloud computing resources required, and helps users select the most suitable leasing scheme from multiple cloud service products with different pricing modes according to user policy, such as maximizing performance or minimizing cost [45]. Aniceto et al. gave a more detailed deployment strategy for small IT companies as cloud users. They abstracted the resource into the number of instances, and built a statistical model using historical data to predict the number of instances demanded at the moment. According to the pricing model, the authors further categorized cloud products into reserved instances and on-demand instances. They finally reduced 32% cost compared with only adapting on-demand deployment by formulating a mixed deployment strategy [46]. In 2018, Ghule and Gopal further proposed a comparison framework for cloud users to analyze the difference between IaaS cloud providers by defining suggestive comparison parameters, such as reliability, performance, serviceability [47].

2.3 Cost optimization

In the above sections, we reviewed modeling methods and tools to measure TCO. It may not be enough for providers or users to make an optimal decision. In order to sustain a thriving business in an enterprise, it is an effective way to maximize profit by optimizing and reducing the total cost of ownership. In cloud computing, many papers studied cost optimization from a different aspect. We survey and summarize several representative papers from the aspect of task scheduling, resource scheduling, and heterogeneous computing.

As cloud providers, all users' operation requests will be executed in the provider backend. Effectively scheduling tasks requested by users can improve service quality, reduce system latency, and reduce costs. We select several typical studies. They adopted different optimization methods in different application scenarios and finally achieved certain results. Pandey et al. optimized the general task scheduling problem in cloud computing and solved the Task-Resource Scheduling Problem using the Particle Swarm Optimization (PSO)-based Heuristic optimization method. They minimized the cost of task execution to reduce the operating costs of the TCO. The proposed PSO method can achieve lower cost than BRS (best resource selection) algorithm, and the PSO can converge faster than the GA (Genetic Algorithm) [48]. Zhang et al. further considered the scheduling strategy when tasks can be executed in parallel. They proposed the DeCloud architecture responsible for scheduling users' data requests from a centralized data storage center. They presented Generic Searching Algorithm (GSA) and Heures Searching Algorithm (HSA) to schedule parallel and non-parallel tasks, respectively. Finally, their method is better than greedy search, and random search [49]. In 2020, Ye et al. proposed a profiling-based consolidation scheduling scheme to reduce the resource cost. Their method can perform on large scale VMs with low latency [50].

As a typical resource scheduling problem in cloud computing, elastic computing has always been a concern of cloud providers. Over-provisioning will lead to lower resource utilization, thus increasing cloud computing operating costs. Under-provisioning will result in unmet user requirements and reduction of service quality. Each research team has different focuses and directions on this

issue. Wu et al. demonstrated an optimization model to schedule resources in SaaS. They minimized the cost by minimizing resources allocated to VMs while meeting SLA and QoS requirements. Their proposed algorithm can reduce the cost by 50% compared to the base algorithm, ProfminVio [51]. Mao et al. viewed VM (Virtual Machine) as the unit of recourse. They considered several factors, including the type of VM, the startup latency of the VM, and the deadline of the task to construct an auto-scaling schedule strategy. They performed different types of tasks by scheduling different kinds of VMs and finally got a lower cost compared with using fixed VM type under the condition of satisfactory performance, and deadline [52].

Companies may encounter business situations that require a mix of local computing resources and exogenous public cloud computing resources, which is common in hybrid clouds. Some studies have found that it is possible to optimize the cost of reducing the amount of investment required by the enterprise by optimally allocating the workload or task quota for local and heterogeneous resources. In 2010, Trummer et al. utilized the COP (Constraint optimization problem) method to minimize the rental cost of external cloud services such that the cost of the enterprise is minimized [53]. However, their research did not analyze the cost of local resources. Thus, it cannot give a global optimal cost solution. Bittencourt and Madeira proposed HCOC (Hybrid Cloud Optimized Cost) scheduling algorithm. It enables computing tasks in a hybrid cloud to be dynamically assigned to local resources (private clouds) or external sources (public clouds). Their method can effectively improve the efficiency of task completion, and reduce the cost compared to greedy schedule algorithm [54]. In 2015, Laatikainen et al. focused on the optimization of the storage costs of hybrid clouds. They first formulated the measurement of hybrid cloud storage costs and then analyzed the impact of refining the reassessment interval on the cost savings attainable by using hybrid cloud storage. Finally, they obtained that shortening reassessment interval and the acquisition of public cloud storage capacity allows the volume variability to be reduced, yielding a reduction of the overall costs [55].

3 Comparison between public and private cloud

Many cloud service providers start to deliver the entire data center as a whole cloud computing solution to customers, such as Alibaba Cloud Apsara Stack [56], Dell EMC [57], HPE Helion Open-Stack [58], Microsoft Azure Stack [59], which deploys public cloud software in a smaller private cloud, providing customers with customized, secure, low-latency private cloud services.

Most of the research works are related to cost modeling and optimization methods in public or hybrid clouds, while private cloud is seldomly mentioned. In this section, we hope that by briefly comparing and contrasting the public and private cloud environment, readers can draw a better understanding of the differences and similarities between public and private cloud, thus better understand the total cost of owning a private cloud and possible ways that can be attempted to optimize the cost. Table 1 gives a brief comparison of the characteristic of public and private cloud. As shown in Table 1, both public and private clouds can provide IaaS, PaaS, and SaaS services, but they have significant differences in the other nine areas. The private cloud services users are large organizations, communities, and enterprises, such as government departments, while public cloud orients to general public users, small companies, and startups. In general, private cloud users need to pay more in the initial stage because they need to purchase a complete cloud computing infrastructure. In contrast, public cloud users only pay for the rental service in the on-demand price. For private cloud users, high costs usually are accompanied by stable cloud service quality and high data security. Since the private cloud is deployed in the user's data center, the user can have an isolated network built within an intranet domain to run the service stably. Meanwhile, data security is guaranteed because of the network isolation and autonomous data management. In contrast, since the public cloud is deployed in the data center of the service provider, the public cloud user needs to access the cloud service through the public network (Internet), and the quality of service often depends on the network quality between the user side and the public cloud data center. Since public cloud users don't know where their data is physically stored, there is a hidden risk of data leakage. Although many data protection techniques have been wildly discussed for the public cloud, data may still suffer from potential attacks [60–62]. In terms of procurement, since private cloud users may need to carry some personalized services, their infrastructure will also need to be customized. Servers in the private cloud can be highly customized where users can select the model specification according to the business requirement, while the public cloud server has a low degree of customization. Generally, public providers purchase servers suitable for the multi-tenant technology.

3.1 Cloud evaluation

For public cloud users and private cloud users, due to the different charging standards of different providers, the choice of cloud service providers directly affects the cost of their investment. For public cloud users, there are some tools to help them choose the service provider and rental

Table 1 Comparison of characteristics between public cloud and private cloud

Characteristics	Private cloud	Public cloud
Service type	IaaS, PaaS, SaaS	IaaS, PaaS, SaaS
Service users	Large organizations	General public or Small startups
Device ownership	Users	Cloud service providers
Deployment location	User's data center	Cloud provider's data center
Quality of service	Stable	Unstable
Data security	High	Low
Use cost	High	Low
Scalability	Scale-out only	Elastic computation
Maintenance	Technician/service provider	Public cloud service
Procurement	High customization	Low customization

solution that best suits their requirements, such as [45, 47, 63]. However, there are no tools to provide users with a comparison of private cloud providers. This is not a trivial problem because private cloud users need to compare private cloud service offerings in terms of investment costs, business value benefits, security and stability, and IT cost savings. On the other hand, cloud users need to estimate the amount of computing and storage resources consumed by their own business before adopting cloud computing solutions. Public cloud and private cloud users can control their costs by evaluating and predicting the resource requirements. Rodrigo N. Calheiros et al. proposed EMUSIM that can help users evaluate and predict the resources consumed by their own business after migrating to cloud [64].

4 TCO evaluation in private cloud

Inspired by [24], we formulated a mathematical equation to describe the total cost of ownership of a typical private cloud.

Let $arp(t)$ represent the amortization rate, and let c_m denote the cost of machine type m, n_m denote the number of machine type m, the cost of server procurement can be formulated as:

$$S = \sum_{m} c_m * n_m * arp(t) \tag{1}$$

Different types of cloud services are charged differently. Some cloud services could be charged according to the number of CPU cores, while others could be charged according to the number of the node that the cloud service actually manages. We use S_i to denote types of cloud services toward this end, and the cost of software is formulated as follows:

$$SW = \sum_{S_i} c_{S_i} * n_{S_i} * arp(t) \tag{2}$$

Cost of network devices are:

$$N = \sum_{n} c_n * n_n * arp(t) + c_{cable}, \tag{3}$$

where c_n and n_n is the cost and quantity of a network switch, respectively

Data center facility is another factor that affects the cost of owning a private cloud. Servers and network devices need to be installed on racks, supporting facilities such as power distributed unit(PDU) and uninterruptible power system(UPS) also need to be taken into consideration. Thus, Thus, with $c_{f_{rack}}$ be the cost of a rack with PDU and UPS, R be the number of racks needed, we formulate the cost of the facility as:

$$F = \sum_{f} c_{f_{rack}} * R * arp(t) \tag{4}$$

Cost of real-estate is related to the size of facilities, with consideration that for a data center may be shared among small private clouds.

$$E = \frac{R}{N_{rack}} * c_{idc} \tag{5}$$

where N_{rack} is the total number of rack a data center could hold.

Power and cooling are two main contributors of operational costs of a data center. Because power consumption per rack of devices can usually be measure individually, we further formulate them in terms of R, with w_{rack} denotes as the power consumption per rack, c_{kw} denotes the cost per kilowatt, and t denotes the time.

$$P = c_{kw} * t * \sum_{i}^{R} w_{rack_i} \tag{6}$$

w_{rack_i} can be further split into two components: static part E_static and dynamic part $E_dynamic$. The static part is

composed with the power consumption of devices in idle state. The dynamic part is related to resource utilization rate. It is composed of power consumption when devices are reading/writing data, network communication, and computing. Then w_{rack_i} can be formulated as:

$$w_{rack_i} = \sum_{j \in D_i} E_{static_j} + E_{dynamic_j} \tag{7}$$

Assume the cooling coefficient is ω, the corresponding cooling cost is

$$C = \omega * P \tag{8}$$

To summarize, the total cost ownership described is

$$\sum S + SW + N + F + E + P + C \tag{9}$$

5 Problem formulation

In this section, we discuss how an optimal number of server racks leads to an optimal cloud TCO. Then, we introduce the server rack deployment problem and provide a formal definition of the problem. Finally, we present the optimization model.

5.1 Minimize cloud TCO with minimum racks

To build a data center, server racks are essential. In Sect. 4, from the Eqs. 4, 5, and 6, the number of server rack is closely related to the cost of data center facility, real-estate, and consumed power. There is a large potential benefit if the data center system is rightsized to the actual requirement over time. As reported in [65], it may offer the potential of reducing the cost of infrastructure by up to 60%. Therefore, cloud providers can further minimize their cloud TCO by minimizing the number of deployed server racks.

5.2 Racks deployment problem in real industry

We consider a cloud platform providing different cloud services to support different cloud products. Each cloud service runs on a virtual machine (VM), and multiple VMs are deployed on a server called a physical machine (PM). Multiple servers are deployed on a server rack in a data center where the rack provides the servers' power supply. In the next level, multiple server racks are connected to an access switch (ASW), where network data can be switched and transmitted. The relationship of servers, racks, and ASW is illustrated in Fig. 1, where a server is deployed on a rack, and a rack is connected to an ASW. For simplification, if one rack is connected to an ASW and one server is deployed on the rack, we call it connected to the ASW.

In real industry, the server racks deployment problem aims to find optimal number of racks to minimize the cloud TCO. The number of required racks for constructing the data center is determined by the organization of servers. Arbitrary deployment of servers may bring risk by overloading power because each server may consume certain energy and there is a power limitation on the rack. Additionally, the capacity of bandwidth on the ASW is limited. If too many communication intensive servers on the same rack are connected to one ASW, it may lower the quality of service (QoS) and user experience. Furthermore, there may be complex relationship between servers, dependency and conflict. On the one hand, one cloud product may consist of several cloud services that may require more than one servers to support a cloud product. Thus, some servers are considered dependent on each other, which means they have to be deployed on the same rack or connected to the same ASW. On the other hand, due to the different business requirements and functionality of cloud products, two servers may be in conflict with each other, which means they have to be deployed on different racks or connected to different ASWs.

In our problem, we consider the dependency and conflict relationship among servers exist on both rack level and ASW level. An example is also illustrated in Fig. 1. In the figure, server 1 and server 2 colored in yellow are dependent on each other on rack level, server 4 and server 5 colored in red conflict with each other on rack level, server 3 and server 6 colored in blue are dependent on each other on ASW level, and server 7 and server 9 colored in green conflict with each other on ASW level. We also consider each server consumes different energy and bandwidth usage. In addition, there are different rack types available when deploy servers, where different racks types refers to different power capacities of racks.

5.3 Optimization model

Next, we formally define our problem and present the optimization model.

Consider a cloud data center to be built. It is given a set of servers $S = \{s_1, \ldots, s_n\}$ with size n where server s_i consumes energy e_i and bandwidth usage b_i. There are unlimited number of ASWs with maximum bandwidth capacity, B, and racks with m types which refer to different maximum power capacity, $p_1, \ldots, p_m$ accordingly. The data center is also given four relationship graphs: Rack-level dependency graph $G^R(S, E_G^R)$, Rack-level conflict graph $\bar{G^R}(S, \bar{E_G^R})$, ASW-level dependency graph $G^W(S, E_G^W)$, and

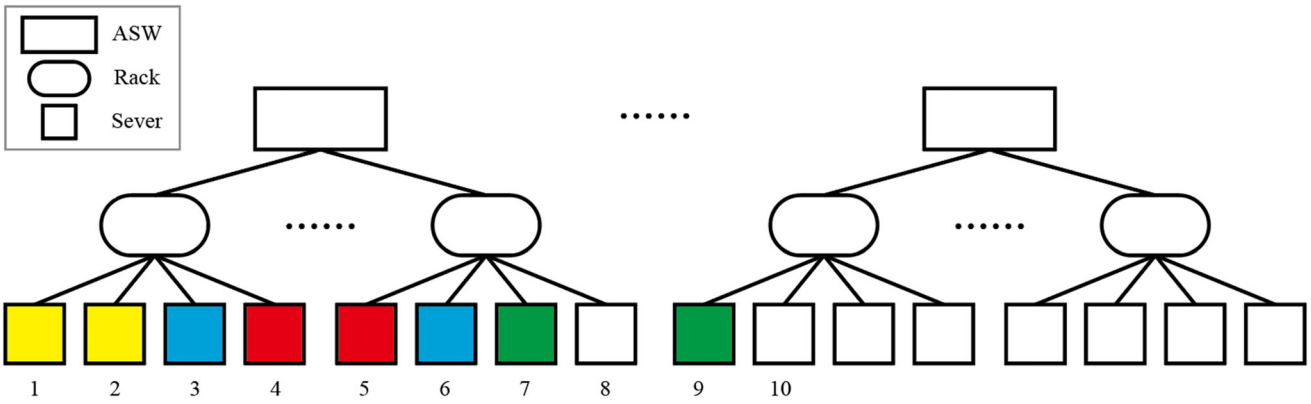

Fig. 1 Cloud data center architecture

ASW-level conflict graph $\bar{G}^W(S, \bar{E}_G^W)$. We further define $A^R, \bar{A}^R, A^W, \bar{A}^W \in \{0,1\}^{n\times n}$ to be the adjacency matrix of the four relationship graphs, where if there is corresponding relationship between two servers, the entry would be 1. The primary goal of the problem is to find an optimal server racks deployment which uses minimum racks and satisfies all relationship constraints and resource capacity constraints.

Then, we formulate the optimization model as

$$\textbf{Minimize}\ R = \sum_i r_i, \tag{10}$$

$$\textbf{s.}t. \tag{11}$$

$$O_{ij} + O_{kj} \le 1, \forall j, \forall (i,k) \in \bar{E}_G^R \tag{12}$$

$$O_{ij} - O_{kj} = 0, \forall j, \forall (i,k) \in E_G^R \tag{13}$$

$$W_{ij} + W_{kj} \le 1, \forall j, \forall (i,k) \in \bar{E}_G^W \tag{14}$$

$$W_{ij} - W_{kj} = 0, \forall j, \forall (i,k) \in \bar{E}_G^W \tag{15}$$

$$\sum_{j=1}^{n} O_{ij} = 1, \sum_{j=1}^{n} W_{ij} = 1, \forall i \tag{16}$$

$$\sum_{j=1}^{n} D_{ij} \le 1, \sum_{j=1}^{n} D_{ij} - r_i = 0, \forall i \tag{17}$$

$$\sum_{j=1}^{m} T_{ij} \le 1, \sum_{j=1}^{m} T_{ij} - r_i = 0, \forall i \tag{18}$$

$$\sum_{j=1}^{n} D_{jk} O_{ij} - W_{ik} = 0, \forall i,k \tag{19}$$

$$\sum_{i=1}^{n} O_{ij} e_i \le \sum_{k=1}^{m} T_{jk} p_k, \sum_{i=1}^{n} W_{ij} b_i \le B, \forall j, \forall k \tag{20}$$

$$\sum_{i=1}^{n} O_{ij} \le M(1 - y_j), \forall j \tag{21}$$

$$1 - \sum_{i=1}^{n} O_{ij} \le M y_j, \forall j \tag{22}$$

$$r_i + y_i = 1, \forall i \tag{23}$$

where $O, W, D \in \{0,1\}^{n\times n}$, $T \in \{0,1\}^{n\times m}$, O_{ij} represent server i is deployed on rack j; W_{ij} represent server i is deployed on ASW j; D_{ij} represent rack i is connected on rack j, and T_{ij} represent rack i is in type j. In the optimization model, we intent to minimize the number of used racks, R. Eq. 10 describes the objective of the model. In a worst case, only one server is deployed on each rack so that we need at most n racks. Thus, $r \in \{0,1\}^n$ is a size n vector where r_i indicate whether the i-th rack is used. In the constraints part, Eq. 12–Eq. 15 describe the relationship of rack-level conflict, rack-level dependency, ASW-level conflict, and ASW-level dependency, respectively. Eq. 16–Eq. 19 describe the pigeonhole principles in this model. Eq. 16 show that one server can only be deployed on one rack and ASW, respectively. Eq. 17 show that one rack can only connect to one ASW and if and only if the i-th rack is deployed, the i-th entry in r would be 1. Similarly, Eq. 18 describe one rack can only have one type. Eq. 19 show that if and only if the i-th server is deployed on the k-th ASW, the i-th server must be deployed on one rack that is connected to the k-th ASW. Eq. 20 describe the constraints that the total power and bandwidth consumption of the deployed servers should not exceed the capacity of rack and ASW, respectively. Eq. 21–Eq. 23 show that if and only if there are non-zero entries in the j-th column of O, the j-th entry of r should be 1, in which a helper vector $y \in \{0,1\}$ has the inverse value of r.

Since the formulation is a mixed integer programming (MIP) whose solution is in NP-Complete [66], we propose

a heuristic methods to find a feasible solution in the next section.

6 Proposed algorithm

This section will illustrate the proposed algorithm, conflict-aware first-fit (CAFF). Our problem has some similar characteristics with the existing bin packing problem with conflict graph (BPPC), also known as an NP-Complete problem [67]. However, our problem has more constraints in two aspects: (1) there is not only the conflicted relationship but also the dependent relationship; (2) the relationships exist on both rack and ASW level in the data center hierarchy. We present a pre-processing method to simplify the dependent relationship to counter these. Finally, we propose a greedy algorithm to solve the server racks deployment problem.

6.1 Pre-processing

To address the problems listed above, we introduce a simple graph based pre-processing method. As introduced in Sec. 5.3, we consider the format of input data is graph. In order to make our problem close to the BPPC problem, we will try to eliminate the dependent relationship before we solve the problem.

Observation 1 For servers, A, B, and C, if A is dependent on B and B is dependent on C, then A is dependent on C.

Based on the observation 1 above, we can induct that in a server dependency graph, each connected component can be considered as a fully connected sub-graph. We can then consider one connected component as a whole part so that the dependent relationship is eliminated in a higher level. In this problem, we define such connected component as a hyper-server in Definition 1. Furthermore, since the server-dependent relationship exists on both rack and ASW levels, we need to eliminate the dependent relationship on both levels.

Definition 1 Given a dependency graph $G(V, E)$ where each vertex represents a server, a *hyper-server* of G is a connected component in the graph. Let S be the set of hyper-server of G. Then, S is the set of the connected components. The number of hyper-servers, $|S|$ is the number of connected components in G. Let S_i be the i-th hyper-server in G. Then, S_i contains all servers that belong to the corresponding connected component.

6.1.1 Rack-level pre-processing

Given the input rack-level dependency graph $G^R(S, E_G^R)$, we can have the set of *rack-level hyper-servers*, S^R, by Definition 1. If we take one rack-level hyper-serer as a whole, then there is no rack-level dependency relation anymore. After constructing the rack-level hyper-servers, the dependency and conflict relationship between two rack-level hyper servers can be inferred by Observation 2.

Observation 2 For two rack-level hyper-servers, A and B, if any sever in A has a conflict or dependency relationship with any server in B on rack level, then rack-level hyper-server A conflicts have conflict or dependency relationship with B on rack-level, respectively.

Then, we can merge the relationship and construct the rack-level conflict graph of rack-level hyper-servers, $\bar{K}^R(S^R, \bar{E}_K^R)$, and the rack-level dependency graph of rack-level hyper-servers, $K^R(S^R, E_K^R)$. By the similar observation, we can further construct the ASW-level conflict graph of rack-level hyper-servers, $\bar{K}^W(S^R, \bar{E}_K^W)$, and the ASW-level dependency graph of rack-level hyper-servers, $K^W(S^R, E_K^W)$. For example, the dependency relationships represented by a black dashed line between servers are merged and become the dependency relationship between rack-level hyper-servers.

6.1.2 ASW-level pre-processing

Moreover, we can construct a higher level hyper-server set, ASW-level hyper-server set S^W, using Definition 1 and the ASW-level dependency graph of rack-level hyper-servers K^W. The dependency relationship between rack-level hyper-servers will also be eliminated if we regard each ASW-level hyper-server as a whole part. Similar to the rack-level pre-processing in Sect. 6.1.1, we can merge the conflicted relationship between rack-level hyper-servers into ASW level. Then, we construct the ASW-level conflict graph of ASW-level hyper-servers, $\bar{L}^W(S^W, \bar{E}_L^W)$.

6.2 Conflict-aware first-fit (CAFF)

After performing the pre-processing in Sect. 6.1, there only remains the conflicted relationship between hyper-servers. Then, we solve BPPC problems on both rack and ASW level using a modified first-fit method. The number of bins used in the ASW level refers to the number of used ASW when building the data center. Accordingly, the number of

bins used in rack-level refers to the number of the used rack when deploying the servers. In our scheme, we first organize the rank deployment problem into several sub-problems. After pre-processing at rack-level and ASW-level, the dependency relationships are eliminated. We then consider the sub-problem as BPPC problem. By solving the sub-problems in a top-down hierarchical way, we can find our solution. The detailed procedure is illustrated in Algorithm 1. In CAFF, Line 3 follows the pre-processing introduced in Sect. 6.1, where it returns four corresponding merged relationship graphs. In-Line 7 and Line 15, it solves the sub-problems on ASW level and rack level respectively using Bin Packing with Conflict Graph First Fit (BPCGFF) function. The idea of BPCGFF is simple. As described in Algorithm 2, it considers the items are placed by a sequence. When a new item comes, it always checks whether the new item can be placed in the used bin. Otherwise, a new empty bin is used to place the new item. This process is used in Line 8 to 10 in Algorithm 2. Moreover, BPCGFF returns the number of used bins and the server deployments. In-Line 11, *Segment_Graph* detailed in Algorithm 3 returns a sub-graph containing the required vertices to efficiently solve sub-problems.

Algorithm 2: Bin_Packing_with_Conflict_Graph_First_Fit (BPCGFF)

Input: The sequence of items in arbitrary order $item\_seq$, the list of corresponding weights $weights$, maximum capacity of bins C, conflict graph $\bar{G}$;

Output: The number of used bins N, the bin packing mapping dep where the key is the index of bins and the value is the set of items;

1 Initialize the mapping dep;
2 Initialize a empty list $opened\_bin$;
3 $unopened\_bin = 1, 2, ..., item\_seq.size$;
4 Let $remained\_capacities$ be a list with size $item\_seq$ in which all values equals to C;
5 **for** $i = 1; i \leq item\_seq.size; i++$ **do**
6 $\quad weight = weights[i]$;
7 $\quad item = item\_seq[i]$;
8 $\quad$ Check whether $item$ can be placed in any $opened\_bin$ in terms of $weight$ and conflict relationship;
9 $\quad$ If so, we find the $candidate\_bin$;
10 $\quad$ If not, move $unopened\_bin[0]$ into $opened\_bin$ and take $candidate\_bin = unopened\_bin[0]$;
11 **end**
12 $N =$ opened_bin.size;
13 **return** N, dep;

Algorithm 1: Conflict-Aware First-fit (CAFF)

Input: The set of servers S, maximum bandwidth capacity of ASWs B, maximum power capacity of different types of racks P,rack-level dependency graph G^R, rack-level conflict graph $\bar{G}^R$, ASW-level dependency graph G^W, ASW-level conflict graph $\bar{G}^W$;

Output: the number of used racks R, the number of used ASWs W, server-to-rack deployment mapping D^R, rack-to-ASW deployment mapping D^W;

1 $R, W = 0$;
2 Initialize deployment mapping D^R, D^W ;
3 $\bar{K}^R, K^W, \bar{K}^W, \bar{L}^W,$ = Pre-processing$(S, G^R, \bar{G}^R, G^W, \bar{G}^W)$;
4 Let $ASW\_hyper\_servers$ be the list of ASW-level hyper servers in the graph $\bar{L}^W$;
5 Let $ASW\_hyper\_server\_BD$ be a list of the ASW-level hyper server bandwidth consumption;
6 Sort $ASW\_hyper\_servers$ and $ASW\_hyper\_server\_BD$ by the bandwidth consumption in descending order;
7 ASW_num, ASW_dep = BPCGFF($ASW\_hyper\_servers$, $ASW\_hyper\_server\_BD$, B, $\bar{L}^W$) ;
8 **for** $i = 1; i \leq ASW\_num; i++$ **do**
9 $\quad W$ += 1;
10 $\quad D^W[W] = ASW\_dep[i]$;
11 $\quad filtered\_graph$ = Segment_Graph($ASW\_dep[i], \bar{K}^R$);
12 $\quad$ Let $rack\_hyper\_servers$ be the list of rack-level hyper servers in the graph $filtered\_graph$;
13 $\quad$ Let $rack\_hyper\_server\_P$ be a list of the rack-level hyper-server power consumption;
14 $\quad$ Sort $rack\_hyper\_servers$ and $rack\_hyper\_server\_P$ by the power consumption in descending order;
15 $\quad rack\_num, rack\_dep$ = BPCGFF($rack\_hyper\_servers$, $rack\_hyper\_server\_P$, P, $filtered\_graph$) ;
16 $\quad$ **for** $j = 1; j \leq rack\_num; j++$ **do**
17 $\quad\quad R$ += 1;
18 $\quad\quad D^R[R] = rack\_dep[j]$;
19 $\quad$ **end**
20 **end**
21 **return** R, D^R, W, D^W;

Algorithm 3: Segment_Graph

Input: A set of node S, A graph $G(N,E)$ that contains more nodes than S;
Output: A filtered graph $\hat{G}(S,\hat{E})$ that only contains nodes in S;

```
Initialize a new edge set Ê;
for (N_i, N_j) in E do
    if N_i in S and N_j in S then
        Add (N_i, N_j) into Ê;
    end
end
return Ĝ(S, Ê);
```

7 Numerical Evaluation

7.1 Simulation Setup

In this section, we demonstrate how we generate the simulated data. We simulate a cloud data center with n servers and m types of racks to be built. Typically, we assume that the power consumption of each server is generated uniformly at random from the range [25, 50], and we consider $m = 4$. Hence, we can set $p_1 = 200, p_2 = 250, p_3 = 300, p_4 = 350$ which refers that we can select 8U, 10U, 12U, and 14U rack. So, the maximum power capacity $P = 350$. The cooling power of the rack is neglected for simplicity. In order to simulate different network requirement, the ASW bandwidth capacity B is set to 30, and the network bandwidth requirement of each server is randomly generated in the range [1, 3]. The dependency and conflict relationships are simulated level by level. In real industry, the dependency relationship tends to be sparser than the conflict relationship. So, we assume each rack-level hyper-server is with size up to 4 and each ASW-level hyper-server is with size up to 3. We randomly group 1 to 4 servers together to form a rack-level hyper-server, and randomly group 1 to 3 rack-level hyper-servers to form an ASW-level hyper-server. In the meantime, we ensure the power and bandwidth requirements of each hyper-server do not exceed the capacities. For the conflict relationship, we generate it based on a *Conflict Complexity Factor*, $\alpha \in \{0, 1\}$. At each level, suppose there are $\tilde{n}$ hyper-servers, then each hyper-server conflicts with other $\alpha * \tilde{n}$ hyper-servers.

To evaluate the performance of the proposed algorithms, we directly use the number of used racks as the metric. We will evaluate the performance with varying conflict complexity and resources capacity in Sect. 7.2. The programs for all algorithms are coded in Python language. A complexity analysis in shown in Sect. 7.3.

7.2 Performance evaluation of CAFF

We explore the performance of the CAFF algorithm with varying conflict complexity in this section. We repeat each experiment with the same setting 500 times in case of the randomly generated conflicted relationship. The result is illustrated in Fig. 2 where the error bar is drawn with 95% confidence. In the results, the number of racks used does not have a strictly linear relationship with the number of servers because the number of conflict relationship does grow linearly as the number of servers increase rather that the number of rack-level servers. We statically generate the same dependency relationship for all repeat runs. So, the number of rack-level servers is fixed for certain points in the result. Overall, the simulation results reveal that as the number of servers increasing, the number of racks used also increases.

We further explore the impact of different resource capacity setting on the performance of CAFF. We consider four resource capacity settings: high power high bandwidth, high power low bandwidth, low power high bandwidth, and low power low bandwidth, where high power refers to $P = 550$, low power refers to $P = 350$, high bandwidth refers to $B = 80$, and low bandwidth refers to $B = 30$. Similar to above, each experiment is repeated 500 times with the conflicted complexity $\alpha = 0.1$. The result is illustrated in Fig. 3. In the figure, we see that poor resource condition results in more racks required, and vice versa. Additionally, it seems the number of racks used is more sensitive to bandwidth capacity in our setting.

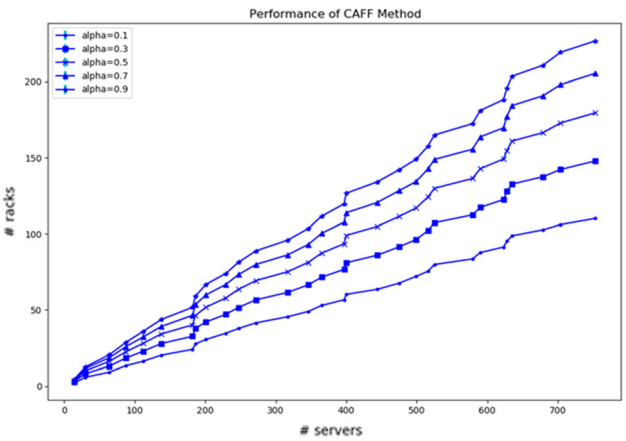

Fig. 2 Avg. number of racks used as server number increase

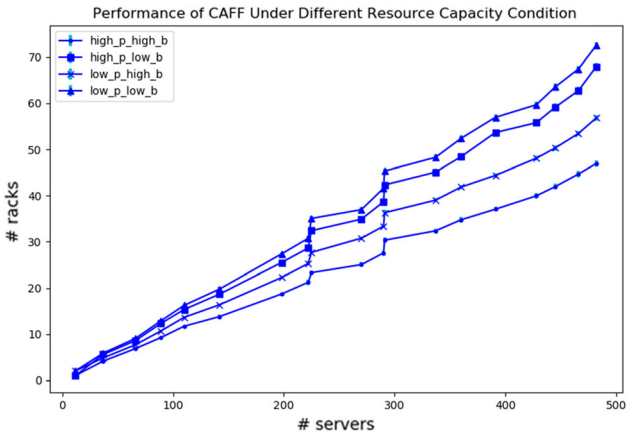

Fig. 3 Avg. number of racks used as server number increase under different resource capacity setting

7.3 Computational complexity

The proposed scheme can be executed in a polynomial time. Through the pre-processing, the proposed heuristics can be decomposed into many in packing sub-problems with conflicts. In a worse case, the BPCGFF has complexity $O(n^2)$ [67]. In Algorithm 1, since the total number elements to be processed in line 8 with for loop is $O(n)$, the computational complexity of CAFF is also $O(n^2)$.

8 Conclusion

In this paper, we first study current researches related to the total cost of ownership (TCO) in cloud computing and compare the public and private cloud in the view of TCO. Then, we demonstrate a TCO evaluation model for Alibaba Dedicated Cloud and further describe a server rack deployment problem to optimize the facility, real estate, and power cost. Finally, we propose a heuristic algorithm, conflict-aware first-fit, to solve the problem. Our simulation results show that the CAFF method can find a feasible solution under different experiment settings.

References

1. Iglesias, J.O., Perry, P., Stokes, N., Thorburn, J., Murphy, L. (2013). A cost-capacity analysis for assessing the efficiency of heterogeneous computing assets in an enterprise cloud. In: IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC 2013, Dresden, Germany, December 9-12, 2013, pp. 107–114. IEEE Computer Society . https://doi.org/10.1109/UCC.2013.32
2. Mvelase, P., Sibiya, G., Dlodlo, N., Oladosu, J., Adigun, M. (2013). A comparative analysis of pricing models for enterprise cloud platforms. In: 2013 Africon, pp. 1–7. https://doi.org/10.1109/AFRCON.2013.6757870
3. Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., & Stößer, J. (2009). Cloud computing-a classification, business models, and research directions. *Business and Information Systems Engineering, 1*(5), 391–399.
4. Etro, F. (2010). The economic consequences of the diffusion of cloud computing
5. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing - the business perspective. *Decision Support Systems, 51*(1), 176–189. https://doi.org/10.1016/j.dss.2010.12.006.
6. Mell, P., & Grance, T. (2009). The nist definition of cloud computing. *National Institute of Standards and Technology, 53*(6), 50.
7. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer System, 25,* 599–616.
8. Pal, R., & Hui, P. (2013). Economic models for cloud service markets: Pricing and capacity planning. *Theor. Comput. Sci., 496,* 113–124.
9. For economics, C., Business research Ltd, C. (2010). The cloud dividend: Part one. the economic benefits of cloud computing to business and the wider emea economy-France, Germany, Italy, Spain and the UK. Business& Information Systems Engineering.
10. Chang, V.I., Wills, G.B., Roure, D.D. (2010). A review of cloud business models and sustainability. 2010 IEEE 3rd International Conference on Cloud Computing pp. 43–50.
11. Brocke, J.V., Lindner, M. (2004). Service portfolio measurement - a framework for evaluating the financial consequences of outtasking decisions. pp. 203–211 . https://doi.org/10.1145/1035167.1035197
12. Skilton, M., Director, C. (2009) Other members of the Cloud Business Artifacts Project: Building return on investment from cloud computing. White Paper.
13. Tsalis, N., Theoharidou, M., Gritzalis, D. (2013). Return on security investment for cloud platforms. 2013 IEEE 5th International Conference on Cloud Computing Technology and Science **2**, 132–137
14. Martens, B., Walterbusch, M., Teuteberg, F. (2012). Costing of cloud computing services: A total cost of ownership approach. 2012 45th Hawaii International Conference on System Sciences pp. 1563–1572.
15. Ellram, L.M. (1993). Total cost of ownership: Elements and implementation
16. Ellram, L.M. (1993). Total cost of ownership: a key concept in strategic cost management decisions
17. Milligan, B. (1999). Tracking total cost of ownership proves elusive. *Purchasing, 127,* 22–23.
18. Degraeve, Z., & Roodhooft, F. (1999). Improving the efficiency of the purchasing process using total cost of ownership in-formation: The case of heating electrodes at cockerill sambre SA. *European Journal of Operational Research*. https://doi.org/10.1016/S0377-2217(97)00383-4.
19. Larsen, M.H., Holck, J., Pedersen, M.K. (2004). The challenges of open source software in it adoption: Enterprise achitecture versus total cost of ownership.
20. Moyle, K.(2004). Total cost of ownership and open source software.
21. Ardagna, D., Francalanci, C., Trubian, M. (2004). A cost-oriented approach for infrastructural design. In: SAC.
22. Klems, M., Nimis, J., Tai, S. (2008). Do clouds compute? a framework for estimating the value of cloud computing. In: WEB.

23. Patel, C.D., Shah, A. (2005). Cost model for planning, development and operation of a data center.
24. Li, X., Li, Y., Liu, T., Qiu, J., Wang, F. (2009). The method and tool of cost analysis for cloud computing. In: 2009 IEEE International Conference on Cloud Computing, pp. 93–100. https://doi.org/10.1109/CLOUD.2009.84
25. Filiopoulou, E., Mitropoulou, P., Tsadimas, A., Michalakelis, C., Nikolaidou, M., Anagnostopoulos, D. (2015). Integrating cost analysis in the cloud: A sos approach. 2015 11th International Conference on Innovations in Information Technology (IIT) pp. 278–283.
26. Simonet, A., Lebre, A., Orgerie, A. (2016). Deploying distributed cloud infrastructures: Who and at what cost? In: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), pp. 178–183. https://doi.org/10.1109/IC2EW.2016.48
27. Sun, K., Li, Y. (2013) Effort estimation in cloud migration process. In: 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, pp. 84–91. https://doi.org/10.1109/SOSE.2013.29
28. Thanakornworakij, T., Nassar, R., Leangsuksun, C., Paun, M. (2012). An economic model for maximizing profit of a cloud service provider. 2012 Seventh International Conference on Availability, Reliability and Security pp. 274–279.
29. Sousa, E., Maciel, P., Medeiros, L., Lins, F., Tavares, E., Medeiros, E. (2013). Stochastic model generation for cloud infrastructure planning. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4098–4103 . https://doi.org/10.1109/SMC.2013.699
30. Sousa, E., Lins, F., Tavares, E., Cunha, P., & Maciel, P. (2015). A modeling approach for cloud infrastructure planning considering dependability and cost requirements. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45*(4), 549–558. https://doi.org/10.1109/TSMC.2014.2358642.
31. Vrček, N., Brumec, S. (2013). Role of utilization rate on cloud computing cost effectivness analysis. In: International Conference on Information Society (i-Society 2013), pp. 177–181.
32. Molka, K., Byrne, J. (2013). Towards predictive cost models for cloud ecosystems: Poster paper. In: IEEE 7th International Conference on Research Challenges in Information Science (RCIS), pp. 1–2 . https://doi.org/10.1109/RCIS.2013.6577736
33. Singh, S., Aazam, M., St-Hilaire, M. (2017). Race: Relinquishment-aware cloud economics model. In: 2017 24th International Conference on Telecommunications (ICT), pp. 1–6 . https://doi.org/10.1109/ICT.2017.7998279
34. Project, B. (2009). Wp 21 tropical project green optical networks: Report on year 1 and unpdate plan for activities. FP7-ICT-2007-1216863 BONE project.
35. Hamilton, J. (2009). Cooperative expendable micro-slice servers (cems): Low cost, low power servers for internet-scale services.
36. Yu, Y., Bhatti, S.N. (2010) Energy measurement for the cloud. International Symposium on Parallel and Distributed Processing with Applications pp. 619–624.
37. Uchechukwu, A., Li, K., Shen, Y. (2012) Improving cloud computing energy efficiency. 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC) pp. 53–58.
38. Chen, F., Schneider, J., Yang, Y., Grundy, J., He, Q. (2012). An energy consumption model and analysis tool for cloud computing environments. In: 2012 First International Workshop on Green and Sustainable Software (GREENS), pp. 45–50. https://doi.org/10.1109/GREENS.2012.6224255
39. Jawad, M., Qureshi, M. B., Khan, U., Ali, S. M., Mehmood, A., Khan, B., et al. (2018). A robust optimization technique for energy cost minimization of cloud data centers. *IEEE Transactions on Cloud Computing*. https://doi.org/10.1109/TCC.2018.2879948.
40. Hu, X., Li, P., & Sun, Y. (2021). Minimizing energy cost for green data center by exploring heterogeneous energy resource. *Journal of Modern Power Systems and Clean Energy, 9*(1), 148–159.
41. Sharma, U., Shenoy, P., Sahu, S., Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In: 2011 31st International Conference on Distributed Computing Systems, pp. 559–570. https://doi.org/10.1109/ICDCS.2011.59
42. Kondo, D., Javadi, B., Malecot, P., Cappello, F., Anderson, D.P. (2009). Cost-benefit analysis of cloud computing versus desktop grids. In: 2009 IEEE International Symposium on Parallel Distributed Processing, pp. 1–12 . https://doi.org/10.1109/IPDPS.2009.5160911
43. Han, Y. (2011). Cloud computing: Case studies and total cost of ownership.
44. Walterbusch, M., Martens, B., & Teuteberg, F. (2013). Evaluating cloud computing services from a total cost of ownership perspective. *Management Research Review, 36,* 613–638.
45. Liew, S., Su, Y.Y. (2012). Cloudguide: Helping users estimate cloud deployment cost and performance for legacy web applications. pp. 90–98 . https://doi.org/10.1109/CloudCom.2012.6427577
46. Orbegozo, I.S.A., Moreno-Vozmediano, R., Montero, R.S., Llorente, I.M. (2011) Cloud capacity reservation for optimal service deployment. In: CLOUD 2011.
47. Ghule, D., Gopal, A. (2018). Comparison parameters and evaluation technique to help selection of right iaas cloud. In: 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp. 1–6 . https://doi.org/10.1109/UPCON.2018.8597059
48. Pandey, S., Wu, L., Guru, S.M., Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 400–407 . https://doi.org/10.1109/AINA.2010.31
49. Zhang, P., Han, Y., Zhao, Z., Wang, G. (2012). Cost optimization of cloud-based data integration system. In: 2012 Ninth Web Information Systems and Applications Conference, pp. 183–188 . https://doi.org/10.1109/WISA.2012.13
50. Ye, K., Shen, H., Wang, Y., & Xu, C. (2020). Multi-tier workload consolidations in the cloud: Profiling, modeling and optimization. *IEEE Transactions on Cloud Computing*. https://doi.org/10.1109/TCC.2020.2975788.
51. Wu, L., Garg, S.K., Buyya, R. (2011). Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In: 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 195–204 . https://doi.org/10.1109/CCGrid.2011.51
52. Mao, M., Li, J., Humphrey, M. (2010). Cloud auto-scaling with deadline and budget constraints. In: 2010 11th IEEE/ACM International Conference on Grid Computing, pp. 41–48 . https://doi.org/10.1109/GRID.2010.5697966
53. Trummer, I., Leymann, F., Mietzner, R., Binder, W. (2010) Cost-optimal outsourcing of applications into the clouds. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science, pp. 135–142. https://doi.org/10.1109/CloudCom.2010.64
54. Bittencourt, L. F., & Madeira, E. R. M. (2011). Hcoc: A cost optimization algorithm for workflow scheduling in hybrid clouds. *Journal of Internet Services and Applications, 2,* 207–227.
55. Laatikainen, G., Tyrväinen, P. (2015). Cost efficiency of hybrid cloud storage cost efficiency of hybrid cloud storage: Shortening acquisition cycle to mitigate volume variation.
56. Group, A. (2009). Apsara Stack (accessed October 15, 2019). https://www.alibabacloud.com/product/apsara-stack

57. Inc., D. (2019) Dell EMC (accessed October 15, 2019). https://www.dellemc.com/en-us/solutions/cloud/vmware-cloud-on-dellemc.htm#scroll=off
58. Development, H.P.E. (2019). HPE Helion OpenStack (accessed October 15, 2019). https://www.hpe.com/us/en/product-catalog/detail/pip.hpe-helion-openstack-cloud-software.1010838414.html
59. Microsoft: Azure Stack (2009) (accessed October 15, 2019). https://docs.microsoft.com/en-us/azure-stack/operator/azure-stack-overview?view=azs-1908
60. Qiu, H., Noura, H., Qiu, M., Ming, Z., & Memmi, G. (2019). A user-centric data protection method for cloud storage based on invertible dwt. *IEEE Transactions on Cloud Computing*. https://doi.org/10.1109/TCC.2019.2911679.
61. Qiu, H., Kapusta, K., Lu, Z., Qiu, M., & Memmi, G. (2019). All-or-nothing data protection for ubiquitous communication: Challenges and perspectives. *Information Sciences, 502,* 434–445.
62. Qiu, H., Qiu, M., Liu, M., & Ming, Z. (2019). Lightweight selective encryption for social data protection based on ebcot coding. *IEEE Transactions on Computational Social Systems, 7*(1), 205–214.
63. Li, A., Yang, X., Kandula, S., Zhang, M. (2010). Cloudcmp: Comparing public cloud providers. In: Internet Measurement Conference.
64. Calheiros, R., Netto, M., De Rose, C., & Buyya, R. (2013). Emusim: An integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications. *Software Practice and Experience, 43,* 595–612. https://doi.org/10.1002/spe.2124.
65. Rasmussen, N. (2011) Determining total cost of ownership for data center and network room ddinfrastructure. Tech. rep. https://download.schneider-electric.-com/files?p_File_Name=CMRP-5T9PQG_R&p_Doc_Ref=SPD_CMRP-5T9PQG_EN
66. Kannan, R., & Monma, C. L. (1978). On the computational complexity of integer programming problems. In R. Henn, B. Korte, & W. Oettli (Eds.), *Optimization and Operations Research* (pp. 161–172). Berlin, Heidelberg: Springer.
67. Gendreau, M., Laporte, G., & Semet, F. (2004). Heuristics and lower bounds for the bin packing problem with conflicts. *Computers and Operations Research, 31*(3), 347–358. https://doi.org/10.1016/s0305-0548(02)00195-8.

Yuanfang Chi received her B.S. and M.S. degree from The University of British Columbia. She is working with Alibaba Cloud on the digitalization and AI-based operation of the Alibaba Dedicated Cloud. Her research interests includes cloud computing, AI-based operations, and distributed machine learning.

Wei Dai received B.S. degree from The Chinese University of Hong Kong, Shenzhen in 2019 and the Master degree from University of Minnesota, Twin Cities in 2020. He is currently pursuing Ph.D. degree in Computer and Information Engineering at The Chinese University of Hong Kong, Shenzhen. His research interests include distributed cloud/edge computing, federated learning, and artificial intelligence.

Yuan Fan is a staff engineer at Alibaba Cloud and received his master's degree from Peking University. He is working on the Cloud infrastructure architect and development, He focuses on hybrid cloud platform development, including resource scheduling, resource optimization, and resource management.

Jun Ruan received his B.S. and M.S. degree from the East China Normal University. He is a senior technology expert at Alibaba and working on hardware and network architecture, site reliability engineering around cloud computing.

Kai Hwang is a Presidential Chair Professor at The Chinese University of Hong Kong (CUHK), Shenzhen, China. He received the Ph.D. in EECS from the University of California at Berkeley. He has worked at Purdue University and University of Southern California for many years prior joining the CUHK in 2018. Dr. Hwang has published 10 scientific books and over 280 scientific papers. An IEEE Life Fellow, He has received the Outstanding Achievement Award in 2005 from China Computer Federation and the Lifetime Achievement Award from IEEE CloudCom 2012. In 2020, he received the Tenth Wu Wenjun Artificial Intelligence

Natural Science Award from China's Artificial Intelligence Association for his recent work on AI-oriented clouds/datacenters.

Wei Cai is currently an assistant professor of Computer Engineering, School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen. He is serving as the director of the Human-Cloud Systems Laboratory, as well as the director of the CUHK(SZ)-White Matrix Joint Metaverse Laboratory. Wei received Ph.D., M.Sc. and B.Eng. from The University of British Columbia (UBC), Seoul National University and Xiamen University in 2016, 2011 and 2008, respectively. Before joining CUHK-Shenzhen, he was a postdoctoral research fellow in Wireless Networks and Mobile Systems (WiNMoS) Laboratory at UBC. He has completed research visits at Academia Sinica (Taiwan), The Hong Kong Polytechnic University and National Institute of Informatics, Japan. Dr. Cai has co-authored more than 50 journal and conference papers in the area of cloud computing, edge computing, interactive multimedia and blockchain systems. He is serving as an associate editor of IEEE Transactions on Cloud Computing. He was a recipient of the 2015 Chinese Government Award for the Outstanding Self-Financed Students Abroad, the UBC Doctoral Four-Year-Fellowship from 2011 to 2015, and the Brain Korea 21 Scholarship. He also received the best student paper award from ACM BSCI2019 and the best paper awards from CCF CBC2018, IEEE CloudCom2014, SmartComp2014, and CloudComp2013.