CAWSAC: Cost-Aware Workload Scheduling and Admission Control for Distributed Cloud Data Centers

Haitao Yuan, Jing Bi, Member, IEEE, Wei Tan, Senior Member, IEEE, and Bo Hu Li

Abstract—Multiple heterogeneous applications concurrently run in distributed cloud data centers (CDCs) for better performance and lower cost. There is a highly challenging problem of how to minimize the total cost of a CDCs provider in a market where the bandwidth and energy cost show geographical diversity. To solve the problem, this paper first proposes a revenue-based workload admission control method to judiciously admit requests by considering factors including priority, revenue and the expected response time. Then, this paper presents a cost-aware workload scheduling method to jointly optimize the number of active servers in each CDC, and the selection of Internet service providers for the CDCs provider. Finally, trace-driven simulation results demonstrate that the proposed methods can greatly reduce the total cost and increase the throughput of the CDCs provider in comparison to existing methods.

Note to Practitioners—A cloud provider deploys its applications in geographically distributed CDCs to improve stability and reliability. For cost and performance, each CDC provides services through multiple ISPs that deliver traffic between millions of users and the CDCs provider. The geographical diversity of the bandwidth and energy cost brings the CDCs provider a big challenge of how to minimize the bandwidth and energy cost of the CDCs provider. This paper first proposes a revenue-based workload admission control method to selectively admit requests. Then, this paper proposes a cost-aware workload scheduling method to allocate requests among multiple available Internet service providers connecting to distributed CDCs. The scheduling strategy can intelligently dispatch requests, and achieve lower cost and higher throughput for the CDCs provider.

Index Terms—Admission control, cloud data centers (CDCs), traffic engineering, wide-area networks (WANs), workload scheduling.

I. INTRODUCTION

INFRASTRUCTURE resources in cloud data centers (CDCs) are shared to concurrently operate multiple applications that provide services to global users [1], [2]. Each CDC typically consumes tens of megawatts of power for running and cooling tens of thousands of servers [3]. To achieve low latency and high availability, applications are replicated and deployed in multiple CDCs distributed in different locations [4]. For cost and performance, each CDC connects to multiple Internet service providers (ISPs) that carry gigantic traffic between millions of users and distributed CDCs.

It has been shown that the energy and bandwidth cost account for a majority of the operational expense (OPEX) of the CDCs provider [3]. As requests of applications in distributed CDCs soar, the energy cost of the CDCs provider is skyrocketing. Recently, there have been works from both academia and industry focusing on the energy minimization problem [5], [6]. However, users’ requests must first go through the wide-area network (WAN) consisting of multiple available ISPs and then arrive in distributed CDCs. For example, Google's WAN provides a huge ISP bandwidth cost to deliver traffic. Besides, the bandwidth cost of each ISP is specified based on the service-level agreement (SLA) signed with the CDCs provider, with some ISPs being much cheaper than others. However, recent works ignore the diversity in the bandwidth cost and capacities of ISPs, and cause high cost and request loss. In addition, CDCs are located in different areas where the energy cost is regional, i.e., the energy cost in distributed CDCs also exhibits geographical diversity. Therefore, it is challenging to minimize the total cost of the CDCs provider in a market where the bandwidth and energy cost show geographical diversity.

In addition, the arrival of requests is dynamic and hard to accurately predict [8]. To solve the problem, this paper first proposes a revenue-based workload admission control method that considers priority, revenue and the expected response time of every request flow. This method inclines to accept higher priority requests while meeting the bandwidth constraint of ISPs. Furthermore, we propose cost-aware workload scheduling to minimize the total cost of the CDCs provider by exploiting the geographical diversity of the bandwidth and energy cost. The proposed scheduling method can provide joint optimization of the ISPs selection and the number of active servers in distributed CDCs for multiple competing applications.

This paper comprises two major stages. The first stage executes the revenue-based workload admission control method according to outside arrival requests, and provides the input...
for the second stage. The second stage runs the cost-aware workload scheduling to specify the optimal workload assignment that can minimize the total cost of the CDCs provider.

The remainder of this paper is organized as follows. Section II reviews the related work in the literature. Section III proposes the architecture of cost-aware workload scheduling in distributed CDCs. Section IV proposes the revenue-based workload admission control method. Section V formulates the cost-aware workload scheduling problem and then proposes the solution method. Extensive trace-driven experiments based on the real-life workload in Google production cluster [9] are conducted to evaluate the proposed cost-aware workload scheduling in Section VI. Finally, we conclude this paper and show the future work in Section VII.

II. RELATED WORK

Here, we discuss the related work and presents the contribution of the workload admission control and cost-aware workload scheduling in comparison to existing works.

A. Performance Modeling

There are several works focusing on the performance modeling and analysis of cloud infrastructure by considering virtual machines (VMs) [1], [10]–[12]. The authors in [1] present an analytical approach to evaluate the performance of cloud infrastructure by considering several metrics including system overhead rate, the rejection probability, and the expected completion time. Authors in [10] propose a stochastic analytic model to quantify the performance of cloud infrastructure. Compared with traditional analytic models, their interacting iterations of multilevel submodels can obtain the solution of the overall model. The authors in [11] formulate the resource allocation problem as an integer programming and propose a heuristic algorithm to maximize the profit of cloud infrastructure. The authors in [12] propose a stochastic reward nets-based analytical model to evaluate the performance of cloud infrastructure. The behavior of a cloud system is quantified and evaluated using the defined performance metrics. However, these works cannot accurately model the energy cost of distributed CDCs. In this paper, we model the energy cost based on servers and consider its geographical diversity for minimizing the total cost of the CDCs provider.

B. Admission Control

The objective of admission control is to protect servers from overload and to guarantee the performance of applications. In [13], the authors provide joint response routing and request mapping in geographically distributed CDCs. The authors in [14] propose a coordinated method to provide admission control and to provision resources for multi-tier services in a shared platform. The reinforcement learning method and cascade neural networks are integrated to improve the scalability and agility of the system. In [15], the authors present a simple algorithm that drops excessive workload provided that the performance is met. However, these works only consider a single application and ignore the resource conflict among multiple applications. Our revenue-based workload admission control method can judiciously admit requests by considering priority, revenue and the expected response time of different request flows.

C. Traffic Engineering

Recently, there have been several works on traffic engineering algorithms [16]–[18]. In [16], the authors adopt an approximation algorithm to solve the virtual local area network (VLAN) assignment problem in different network topologies. Their result shows that the approximation algorithm can provide close-to-optimal traffic engineering and performance guarantee. In [17], the authors present a joint optimization of the workload scheduling and the virtual machine placement to improve traffic engineering in CDCs. However, these works do not adopt a centralized controller to provide traffic engineering. Authors in [18] apply a centralized controller to realize traffic engineering in a network where software-defined networking (SDN) is incrementally deployed. However, their work only applies to a network where only a few switches can be controlled by a centralized SDN controller. Our work considers a network where all switches can be remotely controlled. In addition, the proposed cost-aware workload scheduling can provide the joint optimization of the ISPs selection and the number of servers in each CDC.

III. ARCHITECTURE OF DISTRIBUTED CDCS

This section presents the architecture of distributed CDCs illustrated in Fig. 1. Every CDC hosts a great number of servers ranging from several hundreds to several thousands. Besides, for robustness and performance, multiple ISPs that deliver traffic between distributed CDCs and users connect to each CDC.

In Fig. 1, it is assumed that users around the world send hybrid requests to distributed CDCs where $N$ applications run. Then, users' requests are processed by three modules that are Request Classifier, Admission Control, and Cost-Aware Scheduling, respectively. The Request Classifier module classifies users’ hybrid requests into $N$ request flows, and determines the request
arrival rate for each application. The Admission Control module executes the revenue-based workload admission control to judiciously admit requests. Then, based on the admitted requests, the Cost-Aware Scheduling module can minimize the total cost of the CDCs provider by specifying the workload assignment between ISPss, and the number of active servers in each CDC. In addition, it is assumed that there are K available ISPs delivering traffic between users and distributed CDCs. What’s more, similar to the work in [19], we assume that replicas including programs and data that are indispensable for each application have been distributed across all CDCs. Therefore, applications and their corresponding essential data are strictly consistent with each other. In this way, admitted requests of every application can be independently served within any CDC.

It has been shown that the centralized control of workload in distributed CDCs is feasible [20]. Therefore, similar to the work in [20], as shown in Fig. 1, it is assumed that SDN controllers in the control plane can realize traffic engineering and specify routing paths for each request flow. In addition, it is assumed that SDN controllers can allocate the bandwidth resource of ISPs in the data plane among multiple competing applications. Fig. 1 illustrates that there are multiple available paths to each CDC by specifying routing paths and that there are multiple available paths to each CDC for every request flow corresponding to each application. Therefore, SDN controllers can realize the cost-aware workload scheduling by sending OpenFlow messages [21] and installing flow entries in OpenFlow-enabled switches. Based on the architecture, Section IV proposes the revenue-based workload admission control method. Section V presents the cost-aware workload scheduling that aims to minimize the total cost of the CDCs provider.

IV. WORKLOAD ADMISSION CONTROL PROBLEM

Here, we formulate the workload admission control problem and further presents the solution. The performance of the workload admission control plays an important role in the cost-aware workload scheduling in distributed CDCs. For clarity, we first summarize main notations used throughout this paper in Table I. Higher priority requests can bring more revenue to the CDCs provider than lower priority ones. Therefore, the revenue-based workload admission control method inclines to admit higher priority requests. However, this does not mean that lower priority requests cannot be admitted until higher priority ones have been completely admitted. For example, if there are not enough servers to execute higher priority requests, these requests may experience extremely long response time and bring less or no revenue to the CDCs provider. In this case, to maximize the total revenue, the revenue-based workload admission control method can refuse some of higher priority requests and intelligently admit lower priority ones that can bring more revenue.

Then, we first formulate the revenue-based workload admission control problem. We assume that requests of application n arrive in a Poisson process with rate of \( \lambda_n \). We model distributed CDCs as a \( M/M/m \) queueing system [22]. For application n, the average serving rate of each server is defined as the average serving rate of all servers in distributed CDCs, i.e., \( \sum_{c=1}^{C} \mu_{c,n} / \sum_{c=1}^{C} M_{c,n} \). We assume that active servers are busy all the time, i.e., there are always requests that wait in the queue. Therefore, the expected average response time for the arrival requests of application n, \( ERT_n \), is calculated as follows:

\[
ERT_n = \frac{1}{\left( \frac{\sum_{c=1}^{C} M_{c,n} \mu_{c,n}}{\sum_{c=1}^{C} M_{c,n}} \right) - \lambda_n} + \frac{1}{\sum_{c=1}^{C} M_{c,n} \mu_{c,n}} - \frac{1}{\sum_{c=1}^{C} M_{c,n}} \tag{1}
\]

A \( M/M/m \) queueing system can keep stable on the condition that the traffic intensity of application n, \( \rho_n = (\lambda_n / \sum_{c=1}^{C} M_{c,n} \mu_{c,n}) < 1 \). We denote the utility (revenue) of executing a request from application n in time \( t_n \) as \( u_n(t_n) \). Then, we define the utility function \( u_n(t_n) \) as

\[
u_n(t_n) = \begin{cases} R_n, & t_n < T_{n}^{\text{min}} \\ R_n - \alpha_n (t_n - T_{n}^{\text{min}}), & T_{n}^{\text{min}} < t_n < T_{n}^{\text{max}} \\ 0, & t_n > T_{n}^{\text{max}} \end{cases}
\tag{2}
\]

The time-varying utility function \( u_n(t_n) \) is negatively correlated with \( t_n \). \( R_n \) denotes the maximum revenue brought by executing a request of application n. Note that higher priority requests bring more revenue to the CDCs provider. The request flow of application n has a maximum acceptable average service response time, \( T_{n}^{\text{max}} \), i.e., requests expect to be executed within the time limit \( T_{n}^{\text{max}} \). \( T_{n}^{\text{min}} \) denotes the minimum response time required to execute a request of application n. Here, the value
of $T_{\text{min}}$ is equal to that of $RT_{\text{iner}}$, which denotes the user-defined response time constraint for application $n$. $\alpha_n$ denotes the revenue decay rate of application $n$. In this way, the revenue brought by a request is proportional to its average response time. Equation (2) shows that, if the actual average response time of application $n$, $t_n$, is less than $T_{\text{min}}$, the request flow will bring maximum revenue, $R_n$, to the CDCs provider. However, if $t_n$ is greater than $T_{\text{max}}$, the revenue brought by the request flow decreases. Furthermore, if $t_n$ is greater than $T_{\text{max}}$, no revenue is brought to the CDCs provider. This means that $u_n(t_n)$ is equal to 0 if $t_n$ is equal to $T_{\text{max}}$, i.e., $u_n(T_{\text{max}}) = 0$. Therefore, $R_n = \alpha_n(T_{\text{max}} - T_{\text{min}}) = 0$. Then, $\alpha_n$ can be calculated using

$$\alpha_n = \frac{R_n}{T_{n,\text{max}} - T_{n,\text{min}}}$$

(3)

Therefore, (2) can be rewritten as

$$u_n(t_n) = \begin{cases} R_n, & t_n \leq T_{n,\text{min}} \\ R_n - \frac{R_n}{T_{n,\text{max}} - T_{n,\text{min}}} (t_n - T_{n,\text{min}}), & T_{n,\text{min}} < t_n < T_{n,\text{max}} \\ 0, & t_n > T_{n,\text{max}}. \end{cases}$$

(4)

Let $IS_P W_{\text{Cap}}$ and $\lambda_{n,\text{arrival}}$ denote bandwidth capacity of ISP $k$, and request arrival rate of application $n$, respectively. Let $\lambda_{n,k}$ denote part of $\lambda_n$ allocated to ISP $k$. Let $\lambda_{n,c}$ denote part of $\lambda_n$ allocated to CDC $c$. Let $r_n$ denote the average revenue of a request of application $n$.

Our objective is to maximize the total revenue of the CDCs provider. Then, the revenue-based workload admission control problem can be formulated as follows:

$$\max_{\lambda_n} \text{Revenue} = \sum_{n=1}^{N} (r_n \cdot \lambda_n)$$

subject to

$$\lambda_n \leq \lambda_{n,\text{arrival}}$$

$$\sum_{n=1}^{N} (\lambda_n \cdot s_n) \leq \sum_{k=1}^{K} IS_P W_{\text{Cap}}$$

$$\lambda_n < \sum_{c=1}^{C} M_{c,n,\mu,c,n}$$

$$r_n = u_n(ERT_n)$$

$$ERT_n = \frac{1}{C} \sum_{c=1}^{C} M_{c,n,\mu,c,n} - \lambda_n$$

$$n \in \{1, \ldots, N\}.$$  

(5)

(6)

(7)

(8)

(9)

Constraint (5) ensures that the admitted request rate of application $n$ must be less than corresponding request arrival rate. Constraint (6) means that the total occupied bandwidth of all admitted request flows cannot exceed the total bandwidth of all ISPs. Constraint (7) ensures that a $M/M/m$ queueing system can keep stable. Constraint (8) shows the revenue calculated based on the proposed utility function in (4). Constraint (9) calculates the expected average response time of application $n$.

In this problem, the decision variables are $\lambda_n$ ($n \in \{1, \ldots, N\}$). Note that the objective function is nonlinear with respect to $\lambda_n$. In addition, constraints (5)–(7) are linear constraints with respect to $\lambda_n$. Therefore, this problem is a constrained nonlinear programming. There are several traditional deterministic algorithms to solve this problem, e.g., branch and bound [23], dynamic programming [24] and dynamic backtracking [25]. These algorithms usually rely on the structure of a specific problem and transform it into another one that can be directly solved. However, these algorithms usually obtain optimal solution at the cost of relatively long execution time depending on the complexity of problems.

Recently, stochastic optimization algorithms have been demonstrated to be an efficient tool for tackling constrained nonlinear programming problems. They do not require any knowledge about the mathematical structure of the problems. Besides, the robustness and easy implementation of stochastic algorithms make them widely adopted to solve constrained nonlinear problems. Therefore, to tackle the drawbacks of deterministic algorithms, this paper adopts a hybrid heuristic algorithm based on simulated annealing (SA) [26] and particle swarm optimization (PSO) [27] to solve the formulated problem.

We first apply the penalty function method to transform the formulated problem into an unconstrained one. Let $\text{Penalty}$ denote the value of penalty function defined in

$$\text{Penalty} = \sum_{u=1}^{p} (\max \{0, -g_u(x)\})^\gamma + \sum_{w=1}^{q} |h_w(x)|^\delta.$$  

(10)

Each equality or inequality constraint in the formulated problem corresponds to a penalty added to the objective function, $\text{Revenue}$.

In (10), $x$ denotes a vector of decision variables consisting of $\lambda_n$ ($n \in \{1, \ldots, N\}$). In addition, $\gamma$ and $\delta$ are two constant parameters. Given $p$ inequality constraints, constraint $v$ can be converted into $g_v(x) \geq 0, 1 \leq v \leq p$. Similarly, given $q$ equality constraints, constraint $w$ can be converted into $h_w(x) = 0, 1 \leq w \leq q$. For example, constraint (7) can be converted into $\sum_{c=1}^{C} M_{c,n,\mu,c,n} - \lambda_n > 0$. Then, the corresponding penalty of this constraint is $(\max \{0, -((\sum_{c=1}^{C} M_{c,n,\mu,c,n} - \lambda_n)\})^\gamma$. In this way, the formulated problem can be converted into an unconstrained one described as follows:

$$\min_{\lambda_n} \text{augRevenue} = \min_{\lambda_n} (-\text{Revenue} + \sigma \cdot \text{Penalty}).$$

$\text{augRevenue}$ denotes the augmented objective function. In addition, parameter $\sigma$ is an extremely large positive number, which exaggerates the impact of $\text{Penalty}$ on $\text{augRevenue}$. If a solution is not valid, $\text{Penalty}$ is greater than 0. Therefore, the minimization of $\text{augRevenue}$ can not only find a valid solution that does not cause any penalty, but also maximize $\text{Revenue}$.

The SA algorithm provides a chance to escape from local optima by allowing moves that worsen the objection value in the hope of finding global optima. However, the running time of SA algorithm is relatively long especially when the searching space is large [26]. The PSO algorithm converges to its final solution much quicker than SA algorithm. Nevertheless, PSO algorithm easily traps into local optima in solving constrained...
nonlinear programming [27]. Therefore, our work applies a hybrid heuristic algorithm, which integrates strengths of SA and PSO algorithms. In this hybrid algorithm, old solution of each particle and new one are compared. Better solutions are immediately accepted while inferior solutions are accepted according to the Metropolis criterion that is a typical characteristic of SA algorithm. Therefore, the hybrid algorithm can escape from local optima and eventually find global optima in solving the converted unconstrained problem.

The hybrid heuristic algorithm takes $\lambda_{n, \text{arriv}}$, $IS_{k}^{BW\text{Cap}}$, $s_{n}$, $M_{c,n}$, and $\mu_{c,n}$ as input, and $\lambda_{n}$ as output. For clarity, the detail of this algorithm is omitted from this paper. In addition, it is assumed that although users' requests arrive asynchronously, the hybrid heuristic algorithm is periodically executed at fixed time slots.

V. COST-AWARE WORKLOAD SCHEDULING

This section describes the cost-aware workload scheduling problem in distributed CDCs. Similar to the work [2], each application in a specific CDC is modeled as a $M/M/m$ queueing system. It is assumed that the service time of every server conforms to exponential distribution. Besides, it is assumed that the arrival process is Poisson.

We denote the average serving rate and the average request arrival rate of a server for application $n$ by $\mu_{c,n}$ and $\lambda_{n}$, respectively. Besides, we denote the average number of active servers for application $n$ in CDC $c$ by $m_{c,n}$. The workload intensity in a $M/M/m$ queueing system is denoted by $\rho_{c,n}$, i.e., $\rho_{c,n} = (\lambda_{n} / m_{c,n} \mu_{c,n})$. Based on the queueing theory [22], the condition that a $M/M/m$ queueing system can keep stable is $\rho_{c,n} < 1$. This means that $\lambda_{n}$ must be less than $m_{c,n} \mu_{c,n}$. Let $P_{c,n}^{Q}$ denote the probability of requests waiting in the queue. Let $ART_{c,n}$ denote the average response time in application $n$ of CDC $c$. Therefore, $ART_{c,n}$ can be calculated according to (11).

$$ART_{c,n} = \frac{P_{c,n}^{Q}}{m_{c,n} \mu_{c,n} - \lambda_{c,n}} + \frac{1}{\mu_{c,n}}.$$  \hfill (11)

Without loss of generality, it is assumed that active servers are always busy in each CDC. This means that there are requests waiting in a first-come, first-served (FCFS) queue all the time. Therefore, we assume that $P_{c,n}^{Q}$ in (11) equals 1. Then, we obtain

$$ART_{c,n} = \frac{1}{m_{c,n} \mu_{c,n} - \lambda_{c,n}} + \frac{1}{\mu_{c,n}}.$$  \hfill (12)

Let $RT_{n}^{user}$ denote the user-defined response time constraint for application $n$. Therefore, $ART_{c,n}$ must be less than $RT_{n}^{user}$. Therefore, we have

$$\frac{1}{m_{c,n} \mu_{c,n} - \lambda_{c,n}} + \frac{1}{\mu_{c,n}} \leq RT_{n}^{user}.$$  \hfill (13)

We denote the request rate admitted in CDC $c$ by $\Lambda_{c}$, which can be calculated as follows:

$$\Lambda_{c} = \sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_{n,k,c}, \quad c = 1, 2, \cdots, C.$$  \hfill (14)

In addition, we denote the total number of available servers for application $n$ in CDC $c$ by $M_{c,n}$. Similarly, we denote the total number of available servers for all applications in CDC $c$ by $M_{c}$. Therefore, the total number of available servers for all applications in CDC $c$ must be equal to $M_{c}$. Then, we obtain

$$\sum_{n=1}^{N} M_{c,n} = M_{c}, \quad c = 1, 2, \cdots, C.$$  \hfill (15)

Let $price_{k}$ denote the price of unit bandwidth of ISP $k$. In addition, we denote the energy cost of an active server for application $n$ in CDC $c$ by $b_{c,n}^{a}$. Similarly, the energy cost of a spare server for application $n$ in CDC $c$ is denoted by $b_{c,n}^{s}$. Therefore, the optimization problem that we aim to tackle, Problem One, can be formulated as follows:

$$\min \sum_{c=1}^{C} \sum_{n=1}^{N} \lambda_{n,k,c} \cdot s_{n} \leq ISB_{k}^{BW\text{Cap}}$$

$$\frac{1}{m_{c,n} \mu_{c,n} - \lambda_{c,n}} + \frac{1}{\mu_{c,n}} \leq RT_{n}^{user}$$

$$m_{c,n} \leq M_{c,n}$$

$$\lambda_{c,n} = \sum_{k=1}^{K} \lambda_{n,k,c} < m_{c,n} \mu_{c,n}$$

$$\lambda_{n} = \sum_{c=1}^{C} \lambda_{n,c,n} - \sum_{k=1}^{K} \sum_{c=1}^{C} \lambda_{n,k,c}$$

$$m_{c,n} \in N^{+}$$

$$\lambda_{n,k,c} \geq 0$$

$$n = 1, 2, \cdots, N, \quad k = 1, 2, \cdots, K, \quad c = 1, 2, \cdots, C$$

In this problem, the objective function $TotalCost$ denotes the total cost of the CDCs provider including the ISP bandwidth cost and the energy cost. Constraint (16) shows that the total occupied bandwidth of all admitted requests that traverse ISP $k$ must be less than the bandwidth capacity of the corresponding ISP. Constraint (17) guarantees that the average response time of requests corresponding to application $n$ must be less than the response time constraint of that application. Constraint (18) shows that the number of active servers for application $n$ in CDC $c$ cannot exceed the corresponding limit, $M_{c,n}$. Constraint (19) guarantees that the request arrival rate of application $n$ in CDC $c$ must be less than the total capacity of all corresponding servers. Besides, constraint (20) guarantees that all arrival requests of application $n$ have been allocated to execute in distributed CDCs. Constraints (21) and (22) specify the valid ranges of decision variables including $m_{c,n}$ and $\lambda_{n,k,c}$.

Note that in the problem, the objective function and the corresponding constraints are both linear. Besides, decision variables include continuous variables $\lambda_{n,k,c}$ and discrete variables $m_{c,n}$. Thus, Problem One is a typical mixed integer linear programming (MILP) [28]. To solve this problem, this paper applies the rounding method [29] to tackle the MILP problem using a safe bound. According to [29], constraint (17)
can be converted to $m_{c,n} \geq (1/\mu_{c,n}RT_{\text{user}} - 1) + (\lambda_{c,n}/\mu_{c,n})$. Besides, $m_{c,n}$ integer variables, and therefore, we have $m_{c,n} - \left[(1/\mu_{c,n}RT_{\text{user}} - 1) + (\lambda_{c,n}/\mu_{c,n})\right]$. Then, we can replace $m_{c,n}$ with $\left[(1/\mu_{c,n}RT_{n} - 1) + (\lambda_{c,n}/\mu_{c,n})\right]$ in constraint (18). In this way, we can directly have $\left[(1/\mu_{c,n}RT_{\text{user}} - 1) + (\lambda_{c,n}/\mu_{c,n})\right] \leq M_{c,n}$. Then, it is easy to obtain $(1/\mu_{c,n}RT_{n} - 1) + (\lambda_{c,n}/\mu_{c,n}) \leq M_{c,n}$. Therefore, constraint (18) is converted to $\lambda_{c,n} \leq M_{c,n} - \left[(1/\mu_{c,n}RT_{\text{user}} - 1)/\mu_{c,n}\right]$.

In addition, we can replace $m_{c,n}$ with $\left[(1/\mu_{c,n}RT_{\text{user}} - 1) + (\lambda_{c,n}/\mu_{c,n})\right]$ in constraint (19). Then, we can further convert constraint (19) to $\lambda_{c,n} \leq \mu_{c,n}(1/\mu_{c,n}RT_{\text{user}} - 1) + (\lambda_{c,n}/\mu_{c,n}) = (1/\mu_{c,n}RT_{n} - 1) + (\lambda_{c,n}/\mu_{c,n})$. Constraint (17) shows that $RT_{\text{user}}$ must be greater than $1/\mu_{c,n}$. Thus, constraint (19) is obviously met, and we can remove it directly.

In this way, Problem One can be rewritten as Problem Two, which is a linear programming problem described as follows:

$$
\min_{\lambda, b} \text{TotalCost} = \left( \sum_{k=1}^{K} \left( \sum_{n=1}^{N} \sum_{c=1}^{C} \mu_{n,k,c} \cdot s_{n} \right) \right) \cdot \frac{1}{\mu_{c,n}} \cdot \frac{\mu_{c,n}RT_{\text{user}} - 1}{\mu_{c,n}} + \sum_{c=1}^{C} \left( \mu_{c,n}RT_{\text{user}} - 1 \right) \cdot \lambda_{c,n} \cdot b_{c,n} + \sum_{n=1}^{N} \left( 1 - \frac{1}{\mu_{c,n}RT_{\text{user}} - 1} \right) \cdot \lambda_{c,n} \cdot b_{c,n} \right)
$$

subject to

$$
\sum_{n=1}^{N} \sum_{c=1}^{C} \lambda_{n,k,c} \cdot s_{n} \leq ISP_{k}^{\text{BW Cap}} \quad (23)
$$

$$
\lambda_{c,n} \leq \mu_{c,n} \cdot b_{c,n} - \frac{1}{\mu_{c,n}RT_{\text{user}} - 1} \quad (24)
$$

$$
\lambda_{c,n} = \sum_{c=1}^{C} \lambda_{n,k,c} - \sum_{k=1}^{K} \sum_{c=1}^{C} \lambda_{n,k,c} \quad (25)
$$

$$
\lambda_{n,k,c} \geq 0 \quad (26)
$$

$$
m_{c,n} = \left[ \frac{1}{\mu_{c,n}RT_{\text{user}} - 1} \right] + \frac{\lambda_{c,n}}{\mu_{c,n}} \quad (27)
$$

$k = 1, 2, \cdots, K, c = 1, 2, \cdots, C, n = 1, 2, \cdots, N.$

Therefore, we can first obtain $\lambda_{n,k,c}$ by directly solving Problem Two. Then, we can further obtain the average number of active servers $m_{c,n}$ for application $n$ in CDC $c$ according to (27). In this way, the total cost of the CDCs provider can be minimized by specifying the workload assignment between ISPs, and the number of active servers in every CDC.

VI. EVALUATION

A. Simulation Setting

This section evaluates the effectiveness of our methods by adopting the real-life workload traces in Google production cluster [9] shown in Fig. 2. The dataset contains arrival requests of four applications in Google production cluster for 370 min in May 2011. In addition, the length of sampling period in Fig. 2 is set to 5 min, i.e., the arrival rate of each application is calculated every five minutes. For clarity, in the following part of the paper, the sampling period and the time slot can be used interchangeably. There are existing works on workload prediction [8], therefore, this experiment simply adopts the actual request arrival rates during the 370 minutes. Besides, the priority values of request flows (type 1, 2, 3, and 4) are set to 1, 2, 3, and 4, respectively. A greater priority value means higher priority. A request flow with a greater priority value brings more revenue to the CDCs provider and has the privilege to be admitted to execute.

In this experiment, the arrival requests of four applications are allocated to three distributed CDCs (i.e., $N = 4, C = 3$). All the arrival requests can be delivered to distributed CDCs through three available ISPs (i.e., $K = 3$). We denote the size of each request corresponding to application $n$ by $s_{n}$. It is assumed that a request corresponding to application $n$ ($n = 1, 2, 3, 4$) contains 2, 5, 8, and 10 MB of data on average, respectively, i.e., $s_{1} = 2 \text{ MB}, s_{2} = 5 \text{ MB}, s_{3} = 8 \text{ MB}$ and $s_{4} = 10 \text{ MB}$. In addition, parameter $\sigma$ is set to $10^{10}$. Parameters $\gamma$ and $\delta$ are both set to 2. $T_{\text{max}}$ is set to 1.5 times of that of $T_{\text{min}}$ that is equal to $RT_{\text{user}}$. $ISP_{1}^{\text{BW Cap}}, ISP_{2}^{\text{BW Cap}}, ISP_{3}^{\text{BW Cap}},$ and $ISP_{4}^{\text{BW Cap}}$ are set to $10^{9} \text{ Mbps}, 0.8 \times 10^{9} \text{ Mbps}$, and $0.6 \times 10^{9} \text{ Mbps}$, respectively. $\text{price}_{1}, \text{price}_{2}$, and $\text{price}_{3}$ are set to $2 \times 10^{-10} \$/Mbps, $3 \times 10^{-10} \$/Mbps$, and $6 \times 10^{-10} \$/Mbps$, respectively. In addition, according to the work [3], [7], the setting of other parameters are shown in Table II.

B. Workload Admission Control

The proposed revenue-based admission control method can maximize the total revenue of the CDCs provider by selectively admitting the arrival requests. Fig. 3 illustrates the admitted requests of four applications using the proposed method. It can be clearly observed that requests of application 3 are almost directly admitted into distributed CDCs. The proposed method inclines to admit higher priority requests. However, it is worth noting that though the priority of requests corresponding to application 4 is greater than...
that of requests corresponding to application 3, requests of application 3 are preferred to that of application 4. The reason is that there are not enough servers to execute requests of application 4. In this case, requests of application 4 may experience relatively long response time and will bring less or no revenue to the CDCs provider.

Fig. 4 illustrates the revenue of the CDCs provider and the value of penalty function in every time slot using the hybrid heuristic algorithm that is proposed to solve the admission control problem. In Fig. 4, the revenue of all admitted requests denotes the value of the objective function in the admission control problem formulated in Section IV. This result illustrates that the penalty of the final solution determined by the hybrid heuristic algorithm is zero in every time slot. Therefore, this means that our method can find an effective solution that can bring revenue to the CDCs provider in each time slot.

Then, we evaluate the performance of our revenue-based admission control method by comparing it with the priority-based admission control method [30]. The priority-based admission control method ensures that lower priority requests cannot be admitted until all higher priority ones have been admitted. Fig. 5 shows the revenue of two methods. We can observe that the revenue of the proposed revenue-based workload control method is obviously greater than that of the priority-based admission control method in each time slot. The reason is that the proposed control method estimates the expected revenue brought by admitted requests and admits requests that can maximize the total revenue of the CDCs provider.

C. Cost-Aware Workload Scheduling

Here, we adopt the same parameter setting and further evaluate the performance of the proposed cost-aware workload scheduling method by comparing it with typical average workload scheduling method [31]. The average workload scheduling method equally allocates requests among multiple ISPs and does not consider the bandwidth capacities of ISPs. If the total occupied bandwidth of requests scheduled to a specific ISP exceeds the bandwidth capacity of the ISP, some requests must be refused and cannot traverse this ISP. Besides, requests

<table>
<thead>
<tr>
<th>Table II</th>
<th>Parameters of Three CDCs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>n</td>
</tr>
<tr>
<td>1</td>
<td>n=1</td>
</tr>
<tr>
<td>n=2</td>
<td>800</td>
</tr>
<tr>
<td>n=3</td>
<td>800</td>
</tr>
<tr>
<td>n=4</td>
<td>1200</td>
</tr>
<tr>
<td>$M_2$=3800</td>
<td></td>
</tr>
<tr>
<td>c=2</td>
<td>n=1</td>
</tr>
<tr>
<td>n=2</td>
<td>1800</td>
</tr>
<tr>
<td>n=3</td>
<td>1200</td>
</tr>
<tr>
<td>n=4</td>
<td>2500</td>
</tr>
<tr>
<td>$M_2$=7300</td>
<td></td>
</tr>
<tr>
<td>c=3</td>
<td>n=1</td>
</tr>
<tr>
<td>n=2</td>
<td>1000</td>
</tr>
<tr>
<td>n=3</td>
<td>1200</td>
</tr>
<tr>
<td>n=4</td>
<td>2000</td>
</tr>
<tr>
<td>$M_2$=5200</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Admitted request rates of four applications.

Fig. 4. Revenue and penalty of the CDCs provider.

Fig. 5. Revenue comparison between the proposed admission control and the priority-based admission control.
Fig. 6. Throughput comparison between cost-aware and average workload scheduling.

Fig. 7. Total cost comparison between cost-aware and average workload scheduling.

Fig. 8. Occupied bandwidth of ISPs.

Fig. 9. Requests admitted in distributed CDCs.

Admitted by each ISP are also equally allocated to multiple distributed CDCs. Therefore, requests scheduled to each CDC may exceed the capacity of available servers in that CDC. Fig. 6 illustrates that the throughput of the cost-aware workload scheduling method is greater than that of the average workload scheduling method by the maximum amount of 10.08%.

Note that the average workload scheduling method may cause that some admitted requests cannot be scheduled to execute in distributed CDCs. Therefore, to fairly compare two workload scheduling methods, we apply a penalty cost for each request refused to execute in distributed CDCs. We denote the penalty cost per unit bandwidth corresponding to type \( n \) requests by \( \text{penalty}_n \). The penalty cost for lower priority requests is less than that of higher priority requests. If more admitted requests are allocated to execute in distributed CDCs, these requests will add additional bandwidth and energy cost to the total cost of the CDCs provider. Therefore, the value of \( \text{penalty}_n \) should be greater than or at least equal to that of \( \text{price}_k \). In this experiment, \( \text{penalty}_1, \text{penalty}_2, \text{penalty}_3, \) and \( \text{penalty}_4 \) are set to \( 2 \times 10^{-7} \text{$/Mbps}$, \( 2.5 \times 10^{-7} \text{$/Mbps}$, \( 3 \times 10^{-7} \text{$/Mbps}$, and \( 3.5 \times 10^{-7} \text{$/Mbps}$, respectively. Fig. 7 shows the total cost comparison between the proposed cost-aware and the average workload scheduling method. We can observe that compared with the average workload scheduling method, the total cost of the proposed workload scheduling method can be reduced significantly almost in every time slot.

We then consider the occupied bandwidth of each ISP connecting to distributed CDCs. The charging policies of ISPs are heterogeneous, therefore, the unit bandwidth cost of ISP \( k \), \( \text{price}_k \) is varying. Fig. 8 shows that the occupied bandwidth of each ISP varies a lot due to the difference in charging policies of ISPs. The reason is that the proposed workload scheduling method can intelligently determine suitable ISPs for admitted requests and minimize the total cost of the CDCs provider. We can observe that the number of requests that traverse ISP 1 is the largest among three ISPs. Besides, the number of requests that traverse ISP 3 is the smallest among three ISPs. The result in Fig. 8 is consistent with charging policies of three ISPs, i.e., the unit bandwidth price of ISP 1, \( \text{price}_1 \), is the smallest among three ISPs while the unit bandwidth price of ISP 3, \( \text{price}_3 \), is the largest among three ISPs.

Fig. 9 shows the number of requests admitted into each CDC based on the proposed cost-aware workload scheduling method. The number of available servers for each application and the total number of available servers in every CDC are both limited. Thus, the number of requests admitted into each CDC cannot exceed the total capacity of the corresponding servers in that CDC. Besides, the energy cost of active (spare) servers in different CDCs are varying. Therefore, to minimize the total cost of the CDCs provider, the number of requests admitted into each CDC is also varying. We can clearly see that the number of requests admitted into CDC 1 is the smallest among three CDCs. The reason is that \( b_{1,n} \) \( (b_{1,n}) (n = 1, 2, 3, 4) \) is the largest among three CDCs. The result in Fig. 9 is also consistent with the parameter setting of three CDCs in Table II.

In addition, the total number of active servers for application \( \pi \) in CDC \( c \) must be less than the corresponding limit \( M_{c,n} \). To meet the corresponding response time constraint of each
application, the proposed cost-aware workload scheduling method can determine the optimal number of active servers for each application in each CDC. Fig. 10, 11 and 12 illustrate the number of active servers in CDC 1, 2, and 3, respectively. We can see that the number of active servers for each application in every CDC does not exceed the corresponding limit of available servers in that CDC. Besides, it can be also observed that the number of active servers for the same application in different CDCs varies a lot. For example, the number of active servers for application 4 in CDC 2, $m_{2,4}$, is obviously greater than the number of active servers for application 4 in CDC 1 and 3. The reason is that the energy cost of an active (spare) server for application 4 in CDC 2, $l_{2,4}^1$, is the smallest among three CDCs. Therefore, the result demonstrates that the proposed cost-aware workload scheduling method can determine the optimal number of active servers in each CDC, and minimize the total cost of the CDCs provider.

VII. CONCLUSION AND FUTURE WORK

Distributed CDCs require gigantic bandwidth and energy cost to support multiple applications. Existing works focus on the energy cost minimization problem in CDCs. However, the geographical diversity of bandwidth and energy cost brings a challenge to minimize the total cost of a CDCs provider. This paper studies the total cost minimization problem for the CDCs provider in a market where the bandwidth and energy cost show geographical diversity. To solve this problem, we first propose a revenue-based workload admission control method. Furthermore, we present a cost-aware workload scheduling method for distributed CDCs. Finally, simulation results demonstrate that compared with existing methods, the proposed scheduling method can greatly reduce the total cost and increase the throughput of the CDCs provider. In future work, we would like to extend our work to consider finer-grained metrics, e.g., memory and storage constraints. In addition, we also would like to consider VM scheduling in distributed CDCs where multiple heterogeneous VMs concurrently run in a server.

REFERENCES


Haitao Yuan received the B.S. and M.S. degrees from Northeastern University, Shenyang, China, in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree at the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

His research interests include cloud computing, resource allocation, software-defined networking, and energy efficiency.

Mr. Yuan was the recipient of the 2011 Google Excellence Scholarship.

Jing Bi (M’13) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2011. From 2009 to 2010, she participated in research on cloud computing at IBM China Research Laboratory. She is currently a Postdoctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China. Her research interests include service computing, cloud computing, large-scale data analytics, and resource optimization.

Dr. Bi was the recipient of the 2009 IBM Ph.D. Fellowship Award.

Wei Tan (M’12–SM’13) received the Ph.D. degree in automation from Tsinghua University, Beijing, China in 2008.

He is currently a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. From 2008 to 2010, he was a Researcher with the Computation Institute, University of Chicago, and Argonne National Laboratory, USA. His research interests include large-scale data analytics, NoSQL, cloud computing, business and scientific workflows, and Petri nets.

Dr. Tan is member of the Association for Computing Machinery. He is an associate editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

Bo Hu Li graduated from Tsinghua University, Beijing, China, in 1961.

He was a Visiting Scholar with the University of Michigan, Ann Arbor, MI, USA, from 1980 to 1982. His research interests include distributed simulation and cloud computing.

Dr. Li is a member of the Chinese Academy of Engineering. He was the recipient of the Lifetime Achievement Award from the Society for Modeling & Simulation International (SCS).