Resource Reservation and Request Routing for a Cloud-Based Content Delivery Network

Qilin Fan[1,2], Yuming Jiang[3], Hao Yin[4], Yongqiang Lyu[4], Sen Wang[1,2], Haojun Huang[5], and Xu Zhang[6]

[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society,
Ministry of Education, Chongqing University, Chongqing, China
[2]School of Big Data and Software Engineering, Chongqing University, Chongqing, China
[3]Department of Information Security and Communication Technology (IIK),
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
[4]Research Institute of Information Technology (RIIT), Tsinghua University, Beijing, China
[5]College of Computer, China University of Geosciences, Wuhan, China
[6]School of Electronic Science and Engineering, Nanjing University, Jiangsu, China

***Abstract*—Content delivery networks (CDNs) built on clouds constitute a promising content-distribution-as-a-service alternative. Exploiting the advantages of the cloud, such as the pay-as-you-go business model and geographical dispersion of resources, a cloud-based CDN can provide a flexible and cost-effective solution to realize content delivery without investments on installing and maintaining the infrastructures. However, resource reservation and request routing are critical issues to be addressed in such systems due to the geographical diversity of cloud resource prices and the dynamic nature of traffic demands. In this paper, we investigate the optimal resource reservation problem of cloud resources and explore the best trade-off between rental cost and user experience, and formulate the problem and develop a heuristic approach for it. To deal with fluctuating user demands, we further investigate the request routing optimization problem and design an online heuristic algorithm to redirect user requests. The simulation experiments demonstrate the effectiveness of our algorithms.**

***Index Terms*—content delivery network, resource reservation, request routing, cloud, heuristic algorithm**

I. Introduction

Content delivery networks (CDNs) are large distributed infrastructures of replica servers placed in strategic locations. By replicating contents of origin server on replica servers, the contents are delivered to end-users over the Internet as closely as possible with high-performance guarantee. Today, enterprises, regardless of their scale, are highly dependent on CDNs to maintain and develop their business [1].

Traditional CDNs are statically deployed [2, 3]. To deal with the challenge that user requests may fluctuate dramatically over time, traditional CDNs typically pre-deploy excessive service resource to maintain reliable system performance. However, this can result in low resource utilization in replica servers in the idle time [4] and consequently increased cost.

The recent emergence of cloud services such as Amazon S3 opens up new opportunities to enable cost-effective CDNs, by basing the CDNs on the cloud. For example, Netflix has moved its streaming servers, data stores, and other customer-oriented APIs to Amazon Web Services (AWS). As a customer, one may build a CDN on the cloud to provide CDN services without the need of investing on installing and maintaining the infrastructures. In particular, a cloud-based CDN can benefit from the cloud's pay-as-you-go business model [5]. Specifically, the cloud-based CDN can dynamically adjust the lease of CPU, memory, bandwidth and storage resources from the cloud based on the traffic demand to reduce the total rental cost without severely sacrificing the service performance.

To implement a cloud-based CDN, two fundamental issues have to be addressed, which are *resource reservation*, i.e. how many resources should be reserved on the cloud, and *request routing*, i.e. how to route user requests to use the reserved resources in the cloud, for the cloud-based CDN. In the literature, several recent works have published their results concerning building cloud-based CDNs [6–9]. However, they either only considered one content [6–8] or worked on solving linear programming problems [9], which may be impractical for large scale systems with rapidly varying demand patterns.

The objective of this work is to design resource reservation and request routing algorithms that are practical for use in large scale cloud-based CDNs and can achieve the best user experience at the least cost. This involves two critical challenges. First, since users spread over multiple geographically distributed regions, accordingly in order to improve user experience at these regions, a cloud-based CDN should deploy its delivery service in data centers located in or closer to these regions and the users there. However, such data centers possibly charge with different prices. Thus, for this cloud-based CDN, it needs to answer: How many resources (e.g., storage and bandwidth) should be reversed at each location? Second, the user requests can be highly dynamic and evolutionary over time. Thus, request routing is also a dynamic process and it needs to answer: How should user requests be directed to different geo-distributed data centers where the resources have been reserved, online?

With the two challenges in mind, our contributions are several-fold. First, for resource reservation, we develop a

978-1-7281-1442-2/19/$31.00 ©2019 IEEE
DOI 10.1109/SOSE.2019.00048

IEEE computer society

general optimization formulation of the problem, aimed to seek a trade-off between the rental cost and the user experience. We prove that the optimization problem is NP-hard. Due to this, we design a heuristic approach for resource reservation, which provides decisions on *what contents are to be replicated and where to place them, and how many storage and bandwidth resources should be reserved at each data center*. Second, for request routing, the corresponding optimization problem is also developed. Though this optimization problem may be solved using some existing (offline) linear programming approaches, their time complexity can be too high for online use that is critical when handling user demand fluctuation under given reserved resources. To this aim, we design *an online instead of offline algorithm to efficiently route user requests to optimize user experience*. Third, to validate the effectiveness of our proposed algorithms for the two challenges, we conduct extensive data-driven simulations. The results indicate that our algorithms behave closely to the theoretical optimum.

The rest of this paper is organized as follows. Sec. II formulates the problem and presents a heuristic approach for resource reservation. Sec. III presents the optimization problem and a greedy heuristic algorithm for online request routing. The proposed algorithms are evaluated in Sec. IV. Finally, Sec. V concludes the paper.

II. Resource Reservation

In this section, we formulate the resource reservation problem and present an effective heuristic algorithm for it.

A. Problem Formulation

1) Notations and definitions: The origin site aims to serve multiple contents (e.g., videos) to users residing on geo-distributed locations. Since the service capability of the origin site is limited, the CDN is built on resources leased from data centers of the cloud service provider, which are located across the Internet. When the leased resources at the data centers are unavailable, the user requests will be redirected to the origin site. The CDN distributes K contents $\mathcal{C} = \{C_1, \ldots, C_k, \cdots, C_K\}$, indexed with k; serves M geo-distributed locations, which are denoted as $\mathcal{L} = \{L_1, \ldots, L_i, \cdots, L_M\}$, indexed with i; rents N data centers from the cloud service provider. To facilitate the formulation, the origin site is denoted as DC_0. The set of data centers and the origin site is denoted as $\mathcal{DC} = \{DC_0, DC_1, \ldots, DC_j, \ldots, DC_N\}$, indexed with j. Let d_{ik} denote the demand or request of content k from location i.

2) Decision variables: For resource reservation, the first decision-making objective is about content deployment: what contents are to be replicated and where to place them. To this end, we introduce the first decision-making variable x_{jk} defined as:

$$x_{jk} = \begin{cases} 1 & C_k \text{ is stored in } DC_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The second decision-making objective of resource reservation is about flow allocation: what percentage of the requests should be assigned to each data center. It determines how much bandwidth resource should be reserved. For this, we introduce the second decision-making variable y_{ijk} representing the ratio of the requests on content k routed from location i to data center j, which is constrained as:

$$0 \leq y_{ijk} \leq x_{jk} \quad \forall i, j, k \tag{2}$$

$$\sum_{j=0}^{N} y_{ijk} = 1 \quad \forall i, k \tag{3}$$

where (2) demonstrates the fact that data center j can serve content k only when it has a copy locally, and (3) ensures that the total fraction of requests served is 1.

3) The objective: The overall objective is to provide satisfactory content delivery service to users under minimal cost. As for the cost, we focus on storage rental cost and bandwidth rental cost from the cloud service provider. For tractability, we assume **storage cost** S is linear on the amount of content replicas, and normalized **bandwidth cost** B is the average cost per request (\$/request). Here, for the convenience of representation, we adopt that all contents have normalized size 1 and the rental costs have been adapted to consider the content size normalization. Let p_j^s and p_j^b respectively denote the unit price for renting storage and bandwidth resources.

User perceived latency is the focused performance metric, since even small increments can have significant effect on the reputation of the CDN. CDNs can obtain statistically averaged latencies through active measurements [10] or latency prediction tools [11]. Let l_{ij} denote the latency from location i to data center j. Normalized **latency** L is defined as the average such latency per request (ms/request).

4) The optimization problem: We are now in a position to formulate the resource reservation challenge as an optimization problem that minimizes S, B and L, where are given as follows:

$$S = \sum_{j=0}^{N} p_j^s \sum_{k=1}^{K} x_{jk} \tag{4}$$

$$B = \frac{\sum_{j=0}^{N} p_j^b \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik}}{d^{sum}} \tag{5}$$

$$L = \frac{\sum_{j=0}^{N} \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik} l_{ij}}{d^{sum}} \tag{6}$$

where d^{sum} denotes the total content requests or demands for ease of representation, which is:

$$d^{sum} = \sum_{i=1}^{M} \sum_{k=1}^{K} d_{ik} \tag{7}$$

However, the goals of minimizing the three objectives can often be at odds. For example, only minimizing storage cost would result in that data centers would not store contents and all requests would flow to the origin site. This would inevitably lead to higher bandwidth cost at the origin site and longer user perceived latency. In another instance, only optimizing user perceived latency would drive requests toward their closest data center, leading to increased storage cost at each data center, and in the meanwhile, the bandwidth cost could not be optimal since the bandwidth price had not yet been considered. Considering these, joint optimization on the three objectives should be performed.

Specifically, we utilize the linear weighting method for the three objective functions to optimize resource reservation by $w_1 S + w_2 B + w_3 L$. Since the weighting factors can be scaled by adjusting prices of p^s_j and p^b_j, the optimization objective function can be simplified and written as $S + B + L$. Finally, the optimization problem, denoted as $\mathcal{P}1$, can be expressed as:

$$
\begin{aligned}
min \quad & S + B + L && (8)\\
s.t. \quad & x_{jk} \in \{0, 1\} \quad \forall j, k && (9)\\
& x_{0k} = 1 \quad \forall k && (10)\\
& 0 \le y_{ijk} \le x_{jk} \quad \forall i, j, k && (11)\\
& \sum_{j=0}^{N} y_{ijk} = 1 \quad \forall i, k && (12)
\end{aligned}
$$

5) NP-hardness: The resource reservation optimization problem $\mathcal{P}1$, i.e. (9)–(13), is NP-hard:

Theorem 1. *$\mathcal{P}1$ is NP-hard.*

Proof. The original objective function of $\mathcal{P}1$ is equivalent to

$$
\begin{aligned}
min \quad & d^{sum} \sum_{j=0}^{N} \sum_{k=1}^{K} p^s_j x_{jk} + \sum_{j=0}^{N} p^b_j \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik} \\
& + \sum_{j=0}^{N} \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik} l_{ij}
\end{aligned} \tag{13}
$$

In order to simplify the problem, we have the case where $k = 1$ and obtain the objective function as follows.

$$
\begin{aligned}
min \quad & d^{sum} \sum_{j=0}^{N} p^s_j x_j + \sum_{j=0}^{N} p^b_j \sum_{i=1}^{M} y_{ij} d_i \\
& + \sum_{j=0}^{N} \sum_{i=1}^{M} y_{ij} d_i l_{ij}
\end{aligned} \tag{14}
$$

It is not difficult to find that the optimal solution of above simplest problem must meet $y_{i\tilde{j}} = 1, \exists 1 \le \tilde{j} \le N$ for arbitrary i.

Thus, this problem is a 0-1 integer linear Programming problem which is well-known as a NP-complete problem, then a NP-hard problem consequently. □

B. Our Heuristic Approach

While as shown above resource reservation is a NP-hard problem, due to the requirement of fine-grained update (e.g. 1 hour for Amazon EC2), the system needs a way to address resource reservation efficiently. To this aim, we propose a heuristic approach that has its foundation on the following:

Theorem 2. *Given the content replication strategy x_{jk}, let $\mathcal{Q}_k$ denote the set of data centers containing C_k, the solution of optimal flow assignment can be given as*

$$
y_{ijk} = \begin{cases} 1 & j = \arg\min_{j \in \mathcal{Q}_k} (l_{ij} + p^b_j) \\ 0 & j \ne \arg\min_{j \in \mathcal{Q}_k} (l_{ij} + p^b_j) \end{cases} \tag{15}
$$

Proof. Due to the content replication strategy x_{jk} is known in advance, namely the storage cost S is already determined, the optimization problem $\mathcal{P}1$ can be rewritten as, by removing S and d^{sum}:

$$
\begin{aligned}
min \quad & \sum_{j=0}^{N} \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik} (p^b_j + l_{ij}) && (16)\\
s.t. \quad & 0 \le y_{ijk} \le x_{jk} \quad \forall i, j, k && (17)\\
& \sum_{j=0}^{N} y_{ijk} = 1 \quad \forall i, k && (18)
\end{aligned}
$$

Assume that $DC_{j_1} \in \mathcal{DC}, DC_{j_2} \in \mathcal{DC}, x_{j_1 k} = 1, x_{j_2 k} = 1$. There exists a optimal solution of flow assignment in which $y_{ij_1k}{>}0, y_{ij_2k}{>}0, y_{ij_1k} + y_{ij_2k} = 1$. So we could deduce:

$$
p^b_{j_1} + l_{ij_1} > y_{ij_1k}(p^b_{j_1} + l_{ij_1}) + y_{ij_2k}(p^b_{j_2} + l_{ij_2}) \tag{19}
$$

$$
(1 - y_{ij_1k})(p^b_{j_1} + l_{ij_1}) > y_{ij_2k}(p^b_{j_2} + l_{ij_2}) \tag{20}
$$

$$
y_{ij_1k}(p^b_{j_1} + l_{ij_1}) > y_{ij_2k}(p^b_{j_2} + l_{ij_2}) \tag{21}
$$

$$
p^b_{j_1} + l_{ij_1} > p^b_{j_2} + l_{ij_2} \tag{22}
$$

Similarly, we can prove that:

$$
p^b_{j_2} + l_{ij_2} > p^b_{j_1} + l_{ij_1} \tag{23}
$$

Due to the conflict between (22) and (23), the optimal solution of flow assignment does not exist for any condition that $y_{ij_1k} > 0, y_{ij_2k} > 0, y_{ij_1k} + y_{ij_2k} = 1$. Thus, the optimal solution of flow assignment is obtained when $y_{ijk} = 1$, where $j = \arg\min_{j \in \mathcal{Q}_k} (l_{ij} + p^b_j)$. □

Based on Theorem 2, the variables x_{jk} and y_{ijk} in the resource reservation problem are decoupled, which can be solved through two steps: *We can first optimize content replication to book storage resource, and then conduct flow assignment to reserve bandwidth resource*.

For content replication, the detailed algorithm is presented in Algorithm 1. In this algorithm, we define $gain_{jk}$ as the reduced $S + B + L$ after a newly added replica C_k in data center D_j. Initially, there is no content replica in any data center. First, we calculate the gain $gain_{jk}$ for each content C_k stored at DC_j, and record the best gain $best\_gain_{jk}$ that C_k can obtain, as well as the corresponding optimal data center $best\_DC_j$. Then, we select C_k with the maximal gain value. If the value is greater than 0, we repeat the following process: replicate it at the corresponding optimal data center, and update its best gain and optimal data center. Specifically, lines 4-6 represent the gain content C_k could obtain if it is replicated at data center compared with only at origin site DC_0. Lines 14-25 illustrate that C_k has already be replicated at a set of data centers Q_k. If a new data center is selected to store this content, how much value of $S + B + L$ could be reduced for content C_k.

Finally, storage resource reservation for each data center can be calculated based on Algorithm 1 as:

$$
s_j = \sum_{k=1}^{K} x_{jk}. \tag{24}
$$

In addition, based on Theorem 2, we can obtain bandwidth resource reservation for each data center as follows:

$$b_j = \sum_{i=1}^{M} \sum_{k=1}^{K} y_{ijk} d_{ik}. \tag{25}$$

Algorithm 1 Content Replication Algorithm

```
Input: $M$, $N$, $K$; traffic demand $d_{ik}$; latency $l_{ij}$; unit storage cost $p_j^s$; unit bandwidth cost $p_j^b$
Output: Content replication indicator $x_{jk}$
Initial $x_{jk} \leftarrow 0$, $Q_k \leftarrow \varnothing$
$d^{sum} = \sum_{i=1}^{M} \sum_{k=1}^{K} d_{ik}$
for $k = 1 \rightarrow K$ do
    for $j = 1 \rightarrow N$ do
        $gain_{jk} \leftarrow \sum_{i=1}^{M} d_{ik}(l_{i0} - l_{ij} + p_0^b - p_j^b) - d^{sum} p_j^s$
    end for
    $j^* \leftarrow \arg\max_j gain_{jk}$, $best\_DC_k \leftarrow j^*$, $best\_gain_k \leftarrow gain_{j^*k}$
end for
while true do
    $k^* \leftarrow \arg\max_k best\_gain_k$
    if $best\_gain_{k^*} \leq 0$ then return
    else
        $j^* \leftarrow best\_DC_{k^*}$, $x_{j^*k^*} \leftarrow 1$, $\mathcal{Q}_{k^*} \leftarrow \mathcal{Q}_{k^*} + j^*$
        for $j = 1 \rightarrow N$ do
            if $j \notin \mathcal{Q}_{k^*}$ then
                $\mathcal{Q}_{k^*}^{'} \leftarrow \mathcal{Q}_{k^*}^{'} + j$
                for $i = 1 \rightarrow M$ do
                    $serve\_DC_i^{old} \leftarrow \arg\min_{j \in \mathcal{Q}_{k^*}} (l_{ij} + p_j^b)$
                    $serve\_DC_i^{new} \leftarrow \arg\min_{j \in \mathcal{Q}'_{k^*}} (l_{ij} + p_j^b)$
                end for
                $gain_{jk^*} \leftarrow \sum_{i=1}^{M} d_{ik^*}(l_{i,serve\_DC_i^{old}} - l_{i,serve\_DC_i^{new}} + p^b_{serve\_DC_i^{old}} - p^b_{serve\_DC_i^{new}}) - d^{sum} p_j^s$
            else
                $gain_{jk^*} \leftarrow 0$
            end if
        end for
        $j^* \leftarrow \arg\max_j gain_{jk^*}$, $best\_DC_{k^*} \leftarrow j^*$, $best\_gain_{k^*} \leftarrow gain_{j^*k^*}$
    end if
end while
```

III. Request Routing

In the previous section, the resource reservation problem has been addressed. Specifically, storage and bandwidth resource at each data center have been reserved. This reservation is based on the assumption that content demand could be estimated from previous knowledge or measurement. However, user requests are dynamic and evolutionary over time. Due to this, an online algorithm is required to route user requests to appropriate data centers effectively. The focus of this subsection is to address this request routing challenge.

A. The Optimization Problem

Let $\hat{d}_{ik}$ denote the actual demands of content k at location i, $\hat{y}_{ijk}$ denote the actual ratio of demands on content k routed from location i to data center j. The optimization problem of request routing, denoted as $\mathcal{P}2$, can be formulated as:

$$min \quad \sum_{j=0}^{N} \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{y}_{ijk} \hat{d}_{ik} l_{ij} \tag{26}$$

$$s.t. \quad 0 \leq \hat{y}_{ijk} \leq x_{jk} \quad \forall i, j, k \tag{27}$$

$$\sum_{j=0}^{N} \hat{y}_{ijk} = 1 \quad \forall i, k \tag{28}$$

$$\sum_{i=1}^{M} \sum_{k=1}^{K} \hat{y}_{ijk} \hat{d}_{ik} \leq b_j \quad \forall j \tag{29}$$

where x_{jk} and b_j are given in Sec. II from resource reservation.

Theoretically, the global optimal solution of problem $\mathcal{P}2$ can be solved using linear programming approach such as primal simplex and dual simplex algorithm. However, $\mathcal{P}2$ has large number of variables $|N| \cdot |M| \cdot |K|$ and the linear programming approach requires large iterations and has high computational complexity, thus it is difficult for online use.

B. Our Heuristic Algorithm

In this subsection, we propose a greedy heuristic algorithm for online request routing. Our algorithm is elaborated in Algorithm 2. We assume all requests are first routed to the origin site. In addition, the residual bandwidth of data b_j^r is initialized to b_j, and the unassigned requests $\hat{d}_{ik}^r$ is initialized to $\hat{d}_{ik}$. Then, C_k is arranged based on its number of copies in the system in increasing order. We associate geographical location L_i with data center DC_j to form pairs, and arrange them according to the latency l_{ij} in increasing order. Finally, we allocate actual bandwidth to satisfy the requirement of each C_k in pair (L_i, DC_j). In this way, $\hat{y}_{ijk}$ becomes the decision result of request routing, and the ratio of demands allocated to the origin site is $\hat{y}_{i0k} = 1 - \sum_{j=1}^{N} \hat{y}_{ijk}$.

Algorithm 2 Request Routing Algorithm

```
Input: $M$, $N$, $K$; traffic demand $\hat{d}_{ik}$; content replication indicator $x_{jk}$; bandwidth reservation $b_j$
Output: Ratio of request routing $\hat{y}_{ijk}$
 1: Initial $\hat{y}_{ijk} \leftarrow 0$, $b_j^r \leftarrow b_j$, $\hat{d}_{ik}^r \leftarrow \hat{d}_{ik}$
 2: Sort $C_k$ in increasing order of replication quantity $\left|\sum_{j=1}^{N} x_{jk}\right|$
 3: Sort pair $(L_i, DC_j)$ in increasing order of $l_{ij}$
 4: for all pair $(L_i, DC_j)$ do
 5:     for all $C_k$ do
 6:         if $x_{jk} = 1$ and $\hat{d}_{ik}^r > 0$ then
 7:             $d \leftarrow \min(\hat{d}_{ik}^r, b_j^r)$
 8:             $\hat{y}_{ijk} \leftarrow \hat{y}_{ijk} + \frac{d}{\hat{d}_{ik}}$
 9:             $\hat{d}_{ik}^r \leftarrow \hat{d}_{ik}^r - d$
10:             $b_j^r \leftarrow b_j^r - d$
11:         end if
12:     end for
13: end for
```

The time complexity of line 2, line 3, and lines 4-13 in Algorithm 2 is $O(KlogK)$, $O(MNlog(MN))$, $O(MNK)$, respectively. Due to $K > MN > logK$, the time complexity of our algorithm is $O(MNK)$.

IV. Performance Evaluation

A. Experiment Setup

In this section, we present a performance evaluation study to assess the effectiveness of the proposed algorithms. In the experiments, the locations of data centers and user locations are represented as the longitude and latitude of Chinese cities after clustering. It is considered that the number of data centers is less than the number of user locations of incoming requests.

We assume that the prices of storage and bandwidth of data centers follow truncated normal distributions $N(\mu_s, (\frac{1}{4}\mu_s)^2)$ and $N(\mu_b, (\frac{1}{4}\mu_b)^2)$, where μ_s and μ_b are the average price of storage and bandwidth of the experimental set, respectively. Authors of [12] observed that network latency has substantial correlation with geographical distance. We introduce conversion factor β to approximate network latency by geographical distance (in kilometers). That is, a request with 1km geographical distance translates to βms network latency. We assume that the popularity of contents is governed by a Zipf-Mandelbrot distribution [13] with shape parameter α and plateau parameter q. The normalized population p_k of content k is proportional to $(q+k)^{-\alpha}$. The geographical distribution of each content is randomly generated. The detailed parameters in experiments are listed in Table I.

TABLE I
Experiment parameters setup

Parameter	Default Value	Range
K	500	None
M	40	[20,60]
N	10	[6,14]
μ_s	0.002	[0.001,0.005]
μ_b	40	[20,60]
β	0.04	[0.02,0.06]
α	1	[0.6,1.4]
q	0.5	None
$\sum_{i=1}^{M}\sum_{k=1}^{K} d_{ik}$	1000000	None

B. Simulation Results

First, we randomly set parameters in the variable range and conducted 50 experiments. Metric $Relative\ Optimal = \frac{Optimal}{Optimal^*}$ is defined to verify the validity of our resources allocation algorithm, where $Optimal^*$ is the theoretical optimum calculated by open source optimization tool[1], and $Optimal$ is the optimum calculated by our Algorithm 1. Fig. 1 shows the cumulative distribution function of $Relative\ Optimal$. It illustrates that the relative optimal metric of 95% experiments is less than 1.003. It means that the result calculated by our algorithm is close to theoretical optimum.

Next, through Fig. 2(a) - Fig. 2(d), we investigate how the variable values of parameters impact the results in terms of the optimal value of $S+B+L$ and the corresponding values of S, B and L.

Specifically, Fig. 2(a) illustrates how the optimal value varies with the shape parameter of content popularity distribution α increasing from 0.6 to 1.4. It is clear from Fig. 2(a) that, with the increase of α, the storage cost gradually reduces. The reason is that with the larger diversity of content popularity, storing less popular contents can make the system obtain the same gain. Therefore, the system with flatter content popularity distribution requires more storage resource to be allocated.

Fig. 2(b) displays the optimal value with the different geographical conversion factor β. The default β is set as 0.004. It is clear that, not only user perceived latency but also storage cost increase as the conversion factor β increases from 0.02 to 0.06. This is because the system needs to deploy more content copies at the edge to reduce user perceived latency.

Fig. 2(c) and Fig. 2(d) indicate the optimal value with varied storage price μ_s and bandwidth price μ_b. The default storage price μ_s and bandwidth price μ_b are set as 0.002 and 40, and vary in the range [0.001,005] and [20,60], respectively. As can be seen from Fig. 2(c), both storage cost and user perceived latency increase slightly with the increasing storage price. This is because storage cost and user perceived latency affect each other, as having been discussed for Fig. 2(b). When storage cost increases, the system may reduce the number of content copies stored, leading to the increase of user perceived latency. Nevertheless, as shown in Fig. 2(d), the rising in the bandwidth price of data centers has few effect on user perceived latency and storage cost.

Furthermore, we verify the effectiveness of the proposed heuristic routing request algorithm under different value of γ. For this, we utilize truncated Gaussian distribution to simulate user demands $\hat{d}_{ik} \sim N(d_{ik}, (\gamma d_{ik})^2)$. Fig. 3(a) shows the bandwidth utilization and average latency of data centers when the number of contents is $K = 500$. As can be seen, the bandwidth utilization and average latency are close to the optimal results, with the relative errors 0.2% and 4%, respectively. In addition, we increase the number of contents to $K = 1000$. In this case, the number of copies of unpopular contents stored in the data centers will reduce significantly. Thus, the average latency of data centers greatly increases, as illustrated in Fig. 3(b). Nevertheless, it is observed that the bandwidth utilization and average latency of the proposed algorithm are still close to the optimal results with relative error less than 5%.

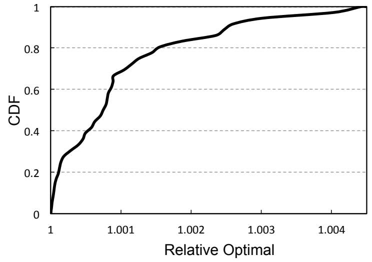
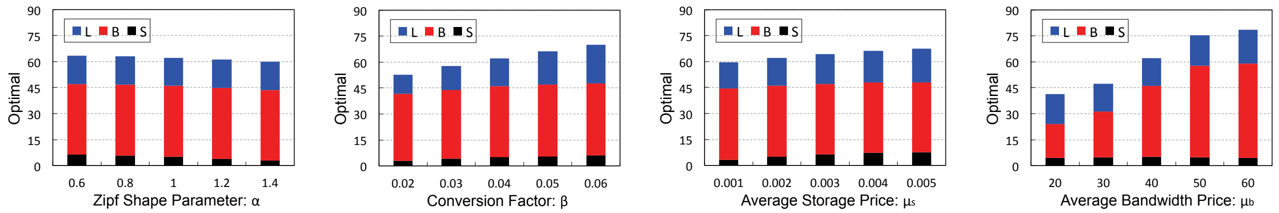


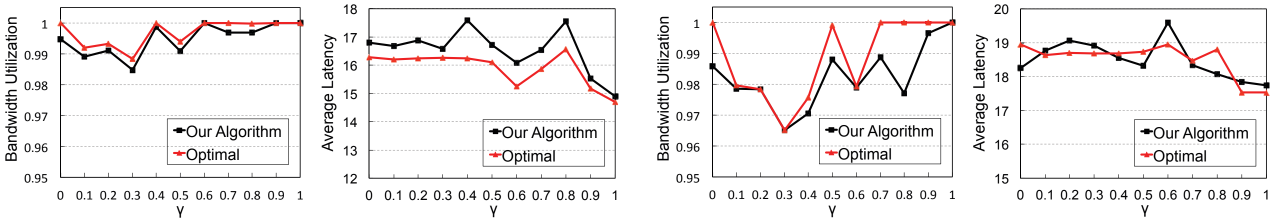
Fig. 1. CDF of $Relative\ Optimal$ of resources allocation algorithm

[1] http://web.mit.edu/lpsolve/doc/Python.htm



(a) (b) (c) (d)

Fig. 2. Optimal value of $S + B + L$ and the corresponding values of S, B and L



(a) $K = 500$ (b) $K = 1000$

Fig. 3. Bandwidth utilization and average latency of data centers

V. Conclusions

As the cloud vendors provide on-demand and cost-effective content storage and delivery capabilities, one can build CDN on the cloud to adaptively meet the dynamic user request requirements. In this paper, we have proposed a multi-objective optimization problem on resources reservation for a cloud-based CDN, which is formulated to seek a trade-off between the rental cost and the user experience. Furthermore, in order to adapt to user demands fluctuation, we have designed an efficient online request routing algorithm. The data-driven simulation experiments have shown that our algorithms achieve similar performance compared to the theoretical optimum.

Acknowledgment

This work was supported in part by the National Key Research and Development Program under Grant no. 2016YFB1000102, in part by the National Natural Science Foundation of China under Grant no. 61631013, 61672318 and 61602071, and in part by the Fundamental Research Funds for the Central Universities under Grant no. 2018CDXYRJ0030.

References

[1] CISCO, "Cisco visual networking index: Forecast and trends, 2017-2022," *CISCO White paper*, pp. 1–38, 2018.

[2] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1002–1026, 2016.

[3] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of cdns," in *Proceedings of INFOCOM*, 2014, pp. 460–468.

[4] P. Wendell and M. J. Freedman, "Going viral: flash crowds in an open cdn," in *Proceedings of IMC*. ACM, 2011, pp. 549–558.

[5] C. Barba-Jimenez, R. Ramirez-Velarde, A. Tchernykh, R. Rodríguez-Dagnino, J. Nolazco-Flores, and R. Perez-Cazares, "Cloud based video-on-demand service model ensuring quality of service and scalability," *Journal of Network and Computer Applications*, vol. 70, pp. 102–113, 2016.

[6] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "Dynamic service placement in geographically distributed clouds," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 762–772, 2013.

[7] F. Chen, K. Guo, J. Lin, and T. La Porta, "Intra-cloud lightning: Building cdns in the cloud," in *Proceedings of INFOCOM*, 2012, pp. 433–441.

[8] F. Wang, J. Liu, and M. Chen, "Calms: Cloud-assisted live media streaming for globalized demands with time/region diversities," in *Proceedings of INFOCOM*, 2012, pp. 199–207.

[9] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, "Scaling social media applications into geo-distributed clouds," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 689–702, 2015.

[10] X. Zhang, H. Yin, D. O. Wu, G. Min, H. Huang, and Y. Zhang, "Ssl: A surrogate-based method for large-scale statistical latency measurement," *IEEE Transactions on Services Computing*, 2017.

[11] D. Mirkovic, G. Armitage, and P. Branch, "A survey of round trip time prediction systems," *IEEE Communications Surveys & Tutorials*, 2018.

[12] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, 2001, pp. 173–185.

[13] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proceedings of INFOCOM*, 2010, pp. 1–9.