Online Cost Minimization for Operating Geo-distributed Cloud CDNs

Xiaoxi Zhang
Department of
Computer Science
The University of Hong Kong
Email: xxzhang2@cs.hku.hk

Chuan Wu
Department of
Computer Science
The University of Hong Kong
Email: cwu@cs.hku.hk

Zongpeng Li
Department of
Computer Science
University of Calgary
Email: zongpeng@ucalgary.ca

Francis C.M. Lau
Department of
Computer Science
The University of Hong Kong
Email: fcmlau@cs.hku.hk

***Abstract*—Cloud-based content delivery networks (Cloud CDN) cache and deliver contents from geo-distributed cloud data centers to end users across the globe, exploiting "infinite" on-demand cloud resources to address volatile user demands. It is critically important to efficiently manage cloud resources in different locations over time, for minimization of the operational cost of the CDN provider, while delivering short response delay to user requests. Although many have studied cost-aware replica placement and request redirection in CDN systems, most are restricted to an offline or one-time setting, or resort to greedy heuristics for online operation. This work proposes an efficient online algorithm for dynamic content replication and request dispatching in cloud CDNs operating over a long time span, targeting overall cost minimization with performance guarantees. Our online algorithm consists of two main modules: (1) a regularization method from the online learning literature to convert the offline cost-minimization optimization problem into a sequence of regularized problems, each to be efficiently solvable in one time slot; (2) a randomized approach to convert the optimal fractional solutions from the regularized problems to integer solutions of the original problem, achieving a good competitive ratio. The effectiveness of our online algorithm is validated through solid theoretical analysis and trace-driven simulations.**

I. Introduction

Content Delivery Networks (CDNs) [1] has been one of the most important Internet-scale distributed systems in the past decade, and is foreseen to continue its profound impact in the next 20 years or so. Among the 28 [2] commercial CDNs nowadays, leading systems (*e.g.*, Akamai [3], Limelignt, Cloudfare) deliver web contents, web and IP-based applications, downloads, and streaming media to a global audience of end-users (*i.e.*, clients). With the fast growing of contents and surges of requests from users, CDN residing in a geo-distributed cloud infrastructure (cloud CDN) [4] has been a rising trend, together with the booming of cloud systems. Spanning a collection of data centers in different geographical locations, such cloud systems support CDN services with "infinite" on-demand resources, catering to the volatile storage and bandwidth demands of these services.

The delivery quality of cloud-based CDN services is subject to a trade-off with the CDN provider's cost, due to operating in a cloud platform. If the CDN service is widely distributed into data centers almost in all the major geographic areas and ISPs across the world, the proximity [3] between the server and the clients guarantees low latency and low package loss rate, yet possibly leading to huge operational costs for resource rental in these data centers [5]. On the other hand, if the CDN service is only deployed at large data centers located in a handful of critical regions, the cost is lower but higher delays could be experienced by the clients. It is therefore practically important to jointly optimize the user experience and minimize the total operational cost, especially for cloud CDNs where resources are acquired in a pay-per-use fashion. The challenge is pivotal: How to dynamically decide content deployment in available data centers located in different geographic locations on the go, striking a balance between costs due to frequent replica migration and replica maintenance in a data center, while guaranteeing acceptable levels of user experience for the huge number of dynamically arriving user requests?

Although a substantial body of work have studied cost-aware *replica placement* and *request redirection*, most of them are restricted to an offline or one-time setting (a detailed discussion is given in Sec. II). The studies addressing online operation largely resort to greedy heuristics without performance guarantee, or assume predictable future information, or ignore content migration cost during dynamic replication to simplify the model.

Contribution. This work proposes a randomized online algorithm for dynamic and optimal cost-effective replication of heterogenous contents, to well balance the request traffic on the fly, in a CDN residing in geo-distributed cloud data centers.

First, we formulate a practical online cost minimization problem, enabling dynamic migration of each content from the *origin* (*e.g.*, storage servers [6] maintaining one copy of each content for reliability, as in real CDNs such as Akamai [3]) to different data centers and removal of the content from a caching data center, as well as request mapping to data centers on the fly. Heterogenous migration costs are considered to differentiate the connection condition (*e.g.*, bandwidth) from the *origin* to each data center. Moreover, different storage and bandwidth costs at different data centers are also considered, as well as variant latencies for request mapping.

Second, we leverage a *regularization* method from the field of online learning [7][8][9] to transform the relaxation of the integer offline optimization problem into a sequence of regularized sub-problems, each of which can be optimally solved in each time slot, for timely replica management and request dispatching. In particular, the regularization removes

978-1-4673-7113-1/15/$31.00 ©2015 IEEE

the time correlation among decisions in the offline problem by lifting the precedence constraints relating to successive time slots into the objective function. By solving each subproblem in each time slot, a feasible (fractional) solution to the relaxed offline problem is obtained with polynomial running time, achieving an upper-bounded overall cost as compared to the optimal offline solution based on competitive analysis using the KKT optimality conditions.

Third, we design a randomized algorithm for rounding the fractional solution to a feasible integer solution to the original problem. This randomized algorithm contains two parts: 1). a rounding algorithm which treats each solved fractional solution for replica placement as probability; 2). a greedy strategy for request redirection. Such a simple redirection strategy turns out to be efficient which observes the inner connection between request mapping and replica management, *i.e.*, once the replica placement is determined in the current time slot, greedy mapping for each area is the best solution to save cost. In this way, the rounding well balances the minimization among three parts of the overall cost, namely content storage cost, request serving cost, and content migration cost. Without relying on any future information, it together with the regulation method yields a good competitive ratio which is irrelevant to the total number of requests nor the time horizon, as compared to the offline optimum.

Organization. We discuss related work in Sec. II and define the system model in Sec. III. Sec. IV and Sec. V give the online algorithm design. Sec. VI presents the trace-driven simulation results and Sec. VII concludes the paper.

II. Related Work

A sustainable body of existing work in the last decade extensively studied cost-effective mechanisms of replica placement problem in CDNs. However, most of them [10][11][12][13][14][15][16] proposed static mechanisms where requests are ideally assumed to be following historical patterns or uniformly distributed among all areas. And the cost considered in the models is apparently over-simplified since no migration (or upload) cost is involved in static settings.

Dynamic replica deployment (in an online fashion) is introduced to better meet the real-time demand. Bartolini *et al.* [17] defined the replica placement model as a Markovian decision process and proposed a corresponding dynamic approach. Chen *et al.* [18] proposed a replica placement protocol to build *dissemination tree*, a dynamic content distribution system on top of a peer-to-peer location service while satisfying QoS requirements. However, the proposed strategies in the aforementioned work were only validated by simulations without competitive analysis.

Chen *et al.* [19] advocated to deploy CDNs supported by the Cloud paradigm to take advantage of the elastic resource provisioning and to save the effort for deploying and provisioning their own infrastructure. They investigated the joint problem of dynamically building distribution paths and placing web server replicas in cloud CDNs to minimize the cost incurred while satisfying QoS requirements for user requests. Unfortunately, both the offline and online algorithms were also heuristics without performance guarantee in the worst case.

Some latest work further promoted the development of the cost-effective mechanism design in CDNs. Liu *et al.* [20] pioneered the systematic study in optimizing content multihoming where multiple CDNs cooperate to serve client requests. Besides the restrictions on offline setting, content migration and request redirection were not considered in their formulation. However, our work conducts a more theoretical approach to handle various costs for online replica deployment and migration as well as serving requests in an online fashion. Mathew *et al.* [21] proposed an online algorithm where CDN servers may be turned off and on upon low loads, rendering a trade-off among CDN power consumption and server on/off transition cost reduction, as well as the service availability guarantee. Yet, they did not prove an upper-bounded competitive ratio. Lin *et al.* [22] investigated the problem of dynamically "right-sizing" data centers by turning off servers by leveraging a "lazy"online algorithm. Moreover, Wu *et al.* [23] proposed proactive algorithms for dynamic scaling of a social media application in geo-distributed clouds. Unlike their work where a lookahead window is assumed based on prediction of limited future requests, our work aims to design an online mechanism without any future information, which is also the key challenge in online algorithm design with good competitive ratio guarantee.

III. Problem Model

A. The Cloud CDN System

We consider a cloud CDN hosted by a CDN service provider in a number of geo-distributed data centers, each of which contains two groups of interconnected and virtualized servers [24]. Data files are stored in *storage servers* while the virtual machines in each data center are running on *computing servers*. Conventionally, we use $[X]$ to denote the set $\{1, 2, ..., X\}$. A data center $i \in [I]$ may store a replica of content $k \in [K]$, which clients may download in the long time interval T. Suppose there are J geographic areas. Requests for downloading each type of content accumulated in each area $j \in [J]$ can be dispatched to multiple data centers in or out of the area. We handle requests on a time-slotted fashion, and consider a chunk-based download model, where the download of each chunk takes at most 1 time slot. Let n_{jt}^{k} denote the number of requests for content k from the area j at time slot t, which includes both newly arrived requests in t for the content and requests for the content which arrived earlier but have not completed the download.

One time slot after another, the CDN system timely redirects the real-time requests to the "right" data centers and manages the storage locations for each content. The goal is to minimize the total cost incurred by the CDN in the T-slot running span. Driven by this, replicas may be dynamically deleted for reducing the storage cost and migrated from the *origin* to data centers.Practically, bandwidth cost is incurred when the replica is migrated from the origin servers and uploaded to the target data center. Without loss of generality, no cost is incurred when a replica is deleted from a data center. To put it all together, the system dynamically handles content replica placement and client request redirection on the fly.

Dynamic content replica placement. Let c_{it}^k denote the cost to store a type-k replica in data center i during time slot t. Let y_{it}, a binary indicator, denote whether data center i stores the replica at t ($y_{it}^k = 1$) or not ($y_{it}^k = 0$). Let w_i denote the migration cost from the *origin* to data center i and z_{it}^k be a binary variable to indicate if migration of content k to data center i occurs in t or not. Since the bandwidth on the links of inter-datacenter networks is typically sufficient [25], we further assume that the delay of the content migration can be ignored.

Client request redirection. Let x_{ijt}^k denote the proportion of the n_{jt}^k requests to be dispatched to and served by data center i. We use r_{ijt}^k as a metric to characterize the cost and delay on data center i in serving each type-k request from area j at time slot t. Specifically, for each data center i and each time slot t, suppose *(i.)* the outgoing bandwidth cost to serve a chunk of content k is b_{it}^k, and therefore the total bandwidth cost for serving $x_{ijt}n_{jt}$ requests is $n_{jt}^k x_{ijt}^k b_{it}^k$; *(ii.)* the computation cost for renting one VM in data center i at time slot t is v_{it} and the number of type-k requests that each VM on data center i can serve is N_i^k, and hence the computation cost of data center i at t for serving area j is $\frac{n_{jt}^k x_{ijt} v_{it}}{N_i^k}$; *(iii.)* the delay in serving each type-k request from area j by data center i at time slot t is d_{ijt}^k. We use d_{ijt}^k multiplied by a delay-cost translation parameter α to describe the cost due to delay. In summary, the service cost incurred in serving area j at time slot t by data center i is

$$n_{jt}^k x_{ijt}^k \times (b_{it}^k + \frac{v_{it}}{N_{it}^k} + \alpha \times d_{ijt}^k) = n_{jt}^k x_{ijt}^k r_{ijt}^k$$

B. The Offline Content Replication and Request Dispatching Problem

We first formulate the offline problem in (1) where only one content is considered. The extension to K contents is straightforward as given in Theorem 1. The objective function (1) is designed to minimize the sum of the overall storage cost, service cost and migration cost in T.

$$P: \quad \min \sum_{t\in[T]}\sum_{i\in[I]} c_{it}y_{it} + \sum_{t\in[T]}\sum_{i\in[I]}\sum_{j\in[J]} n_{jt}x_{ijt}r_{ijt} + \sum_{t\in[T]}\sum_{i\in[I]} w_i z_{it} \qquad (1)$$

subject to:

$$\begin{aligned}
&\sum_{i\in[I]} x_{ijt} \geq 1, && \forall j \in [J], t\in[T] && (1a)\\
&x_{ijt} \leq y_{it}, && \forall j\in[J], i\in[I], t\in[T] && (1b)\\
&z_{it} \geq y_{it} - y_{i(t-1)}, && \forall i\in[I], t\in[T] && (1c)\\
&x_{ijt}\in[0,1], && \forall i\in[I], j\in[J], t\in[T] && (1d)\\
&y_{it}\in\{0,1\}, && \forall i\in[I], t\in[T] && (1e)\\
&z_{it}\in\{0,1\}, && \forall i\in[I], t\in[T] && (1f)
\end{aligned}$$

Constraint (1a) guarantees that at each time slot, requests from each area will be dispatched to multiple data centers and served completely. Constraint (1b) indicates that only when data center i stores a replica at time slot t can it serve requests for that replica at that time. Constraint (1c) illustrates that if data center i stores a replica at t but not at $t-1$, a content migration occurs at t (*i.e.*, $z_{it} = 1$); otherwise, $z_{it} = 0$.

Theorem 1. *In a cloud CDN deployed in a pool of data centers with 'infinite' resources, the minimal operational cost incurred for K contents is equal to the summation of the minimal operational cost incurred by each of the contents.*

The proof is given in our technical report [26].

Toward the design of an efficient online algorithm, we first relax the integrality constraint (1e) and (1f) to $y_{it}, z_{it} \in [0,1]$ and let P_f denote the relaxed program as follows:

$$P_f: \quad \min \sum_{t\in[T]}\sum_{i\in[I]} c_{it}y_{it} + \sum_{t\in[T]}\sum_{i\in[I]}\sum_{j\in[J]} n_{jt}x_{ijt}r_{ijt} + \sum_{t\in[T]}\sum_{i\in[I]} w_i z_{it} \qquad (2)$$

$$\begin{aligned}
\text{subject to:}\quad &(1a)-(1d)\\
&y_{it}\in[0,1], \quad \forall i\in[I], t\in[T] && (2e)\\
&z_{it}\in[0,1], \quad \forall i\in[I], t\in[T] && (1f)
\end{aligned}$$

Let a_{jt}, d_{ijt}, b_{it} denote the Lagrangian dual variables associated with (1a), (1b), and (1c) respectively. We next obtain the dual program [27] of P_f as follows:

$$D_f: \quad \max \sum_{t\in[T]}\sum_{j\in[J]} a_{jt} \qquad (3)$$

subject to:

$$\begin{aligned}
&b_{it} \leq w_i, && \forall i\in[I], t\in[T] && (3a)\\
&\sum_{j\in[J]} d_{ijt} + b_{i(t+1)} - b_{it} \leq c_{it}, && \forall i\in[I], t\in[T] && (3b)\\
&a_{jt} - d_{ijt} \leq R_{ijt}, && \forall i\in[I], t\in[T] && (3c)\\
&a_{jt}, b_{it}, d_{ijt} \geq 0, && \forall j\in[J], i\in[I], t\in[T] && (3e)
\end{aligned}$$

In an online setting, in each time slot $t \in [T]$, all the parameters, variables and constraints related to that time index emerges gradually. Due to the two sets of precedence constraints (1b) (1c) [9], decisions of one time slot are coupled with those in another. We apply a novel regularization method to remove the coupling and design an efficient online algorithm in the next section.

IV. An Online Fractional Algorithm

A. Online Algorithm based on Regularization Method

The *Regularization* technique is adopted by adding a smooth convex function to the original objective function and subsequently deriving the optimal fractional solution to the new problem. The basic idea of our online algorithm is to

TABLE I: Summary of Key Notation

I	# of data centers (DCs) in the cloud CDN
J	# of areas where requests are generated
T	# of time slots during the long time interval
c_{it}	storage cost of DC i for time slot t
n_{jt}	# of requests from area j at time slot t
r_{ijt}	service cost of DC i per request from area j
R_{ijt}	service cost of n_{jt} requests served by DC i
w_i	migration cost for copying a replica to DC i
x_{ijt}	fraction of n_{jt} requests directed to DC i
y_{it}	binary indicator: whether DC i stores a replica at t ($y_{it}=1$) or not ($y_{it}=0$)
z_{it}	binary indicator: whether a replica is copied to DC i at t ($z_{it}=1$) or not ($z_{it}=0$)

lift the precedence constraint (1c) to the objective function in order to remove the correlation between time slot $t-1$ and t. In other words, we intend to decompose the offline optimization problem into a set of simpler sub-problems for each time slot. Then it fits the online setting where decisions are made in each time slot.

Main idea. Let $\tilde{P}_f$ denote the new problem in which time correlation is removed. Let $\tilde{P}_{ft}$ denote the sub-problem our online algorithm is trying to solve in time slot t. We should have (i.) the solution solved from $\tilde{P}_f$ should be feasible in P_f; (ii.) an online algorithm should be designed to produce feasible solutions of P from that of P_f such that the ultimate problem P can be solved; (iii.) $\tilde{P}_f$ is an approximation of P_f which intuitively guarantees a good competitive ratio; and (iv.)

$$\tilde{P}_f = \sum_{t\in[T]} \tilde{P}_{ft}, \tag{4}$$

where each $\tilde{P}_{ft}$ has no time correlation constraint (1c). Note that (ii.) will be achieved in Section V and we focus on (i.), (iii.) and (iv.) in this section.

We apply the idea of regularization in online learning to achieve the approximation from P to $\tilde{P}_f$. Observe that (1c) can be rewritten as an equality constraint $z_{it} = \max\{0, y_{it} - y_{i(t-1)}\}$, while z_{it} in the third term of (1) can be substituted. A commonly used regularizer Relative Entropy Function is as follows:

$$\Delta(\mathbf{y}_t||\mathbf{y}_{(t-1)}) = \sum_{i\in[I]} (y_{it} \ln \frac{y_{it}}{y_{i(t-1)}} + y_{i(t-1)} - y_{it}) \tag{5}$$

First, we regularize P_f to $\tilde{P}_f$ by using $\Delta(\mathbf{y}_t||\mathbf{y}_{(t-1)})$ to approximate $\max\{0, y_{it} - y_{i(t-1)}\}$. Second, we define $R_{ijt} = n_{jt}r_{ij}$ to denote the service cost of data center i for serving area j at time slot t. Hence, $n_{jt}r_{ij}$ can be replaced by R_{ijt}. Third, when (5) is applied as an approximation, we define an approximation weight parameter η to be determined in Sec. IV-B. Moreover, we add a constant term $\frac{\epsilon}{I}$ on both the denominator and the nominator of the fraction within the ln operator to ensure the feasibility when $y_{i(t-1)} = 0$. Note that $y_{i(t-1)}$ for all i are solved in $\tilde{P}_{f(t-1)}$ such that they are fixed as constant in $\tilde{P}_{ft}$. As a result, we have:

Algorithm 1: An Online Regularization-based Fractional Algorithm $ORFA$

```
Input: $I$, $J$, $\mathbf{w}$, $\epsilon$
Output: $\mathbf{x}, \mathbf{y}$
INITIALIZE $\mathbf{x} = \mathbf{0}, \mathbf{y} = \mathbf{0}$;
while time slot $t$ starts, do
    Get $\mathbf{c}_t, \mathbf{r}_t, \mathbf{n}_t, \mathbf{y}_{t-1}$;
    Adopt Interior Point Method to solve
      $\tilde{P}_{ft}(I, J, \mathbf{c}_t, \mathbf{r}, \mathbf{w}, \mathbf{n}_t, \epsilon)$, according to (6);
    Return $\mathbf{x}_t, \mathbf{y}_t$.
end
```

$$\tilde{P}_{ft}: \quad \min \sum_{i\in[I]} c_{it}y_{it} + \sum_{i\in[I]}\sum_{j\in[J]} x_{ijt}R_{ijt}$$
$$+ \sum_{i\in[I]} \frac{1}{\eta} w_i \left[(y_{it} + \frac{\epsilon}{I}) \ln(\frac{y_{it} + \frac{\epsilon}{I}}{y_{i(t-1)} + \frac{\epsilon}{I}}) + y_{i(t-1)} - y_{it} \right] \tag{6}$$

subject to:

$$\sum_{i\in[I]} x_{ijt} \geq 1, \qquad \forall j \in [J] \qquad (6a)$$
$$x_{ijt} \leq y_{it}, \qquad \forall j \in [J], i \in [I] \qquad (6b)$$
$$x_{ijt}, y_{it} \in [0,1], \qquad \forall i \in [I], j \in [J] \qquad (6f)$$

Algorithm Interpretation Our algorithm to solve $\tilde{P}_{ft}$ is given in Alg. 1. Since $\tilde{P}_{ft}$ is a standard convex problem [27], it can be optimally solved in polynomial time, *e.g.*, by the interior point method [27]. Let $\tilde{\mathbf{x}}^\star_{ft}$ and $\tilde{\mathbf{y}}^\star_{ft}$ denote the optimal solution of $\tilde{P}_{ft}$. At each time slot t, the fractional solution $(\mathbf{x}_t, \mathbf{y}_t)$ is determined based on the solution independently of the rounds prior to $t-1$. In fact, such a fractional solution will be used by the algorithm in Sec. V as the base of the final integer solution.

Theorem 2. *Our online algorithm $ORFA$ obtains the optimal solution of $\tilde{P}_{ft}$ in polynomial time.*

Proof: Our online algorithm $ORFA$ adopts Interior Point Method [27] to solve the standard convex optimization $\tilde{P}_{ft}$. ■

Theorem 3. *A feasible solution of P_f is obtained by running our online algorithm $ORFA$ in each of the T time slots.*

Proof: Let $\tilde{S}_{ft}$ and S_f denote the feasible region of $\tilde{P}_{ft}$ and P_f respectively, each of which is a polyhedron defined by the intersection of the constraints of themselves. Then we have $S_f = \cup_{t\in[T]}(\tilde{S}_{ft} \cap (1c))$. We also have that $\{\tilde{\mathbf{x}}^\star_{ft}, \tilde{\mathbf{y}}^\star_{ft}\} \in \tilde{S}_{ft}$ and each $\tilde{z}^\star_{ft}$ can be easily computed by (1c). Thus, taking $(\tilde{\mathbf{x}}^\star_{ft}, \tilde{\mathbf{y}}^\star_{ft}, \tilde{\mathbf{z}}^\star_{ft})$ for all t together is a feasible solution of P_f. ■

B. Competitive Analysis of $ORFA$ via a Primal-dual Framework

Intuition. Let $P^\star_f$ and $D^\star_f$ denote the minimum (optimum) of P_f and maximum (optimum) of D_f, respectively. We abuse

TABLE II: KKT Optimality Conditions of $\tilde{P}_f$ and $\tilde{D}_f$

$\forall\ j \in [J], t \in [T]:$	
$1 - \sum_{i\in[I]} \tilde{x}^\star_{ijt} \leq 0,$	(7.1)
$\tilde{a}^\star_{jt}(\sum_{i\in[I]} \tilde{x}^\star_{ijt} - 1) = 0,$	(7.2)
$\forall\ j \in [J], i \in [I], t \in [T]:$	
$\tilde{x}^\star_{ijt} - \tilde{y}^\star_{it} \leq 0,$	(7.3)
$\tilde{d}^\star_{ijt}(\tilde{y}^\star_{it} - \tilde{x}^\star_{ijt}) = 0,$	(7.4)
$R_{ijt} + \tilde{d}^\star_{ijt} - \tilde{a}^\star_{jt} \geq 0,$	(7.5)
$\tilde{x}^\star_{ijt}(R_{ijt} + \tilde{d}^\star_{ijt} - \tilde{a}^\star_{jt}) = 0,$	(7.6)
$\forall\ i \in [I], t \in [T]:$	
$c_{it} + \frac{w_i}{\eta} \ln(\frac{\tilde{y}^\star_{it} + \frac{\epsilon}{I}}{\tilde{y}^\star_{i(t-1)} + \frac{\epsilon}{I}}) - \sum_{j\in[J]} \tilde{d}^\star_{ijt} \geq 0,$	(7.7)
$\tilde{y}^\star_{it}(c_{it} + \frac{w_i}{\eta} \ln(\frac{\tilde{y}^\star_{it} + \frac{\epsilon}{I}}{\tilde{y}^\star_{i(t-1)} + \frac{\epsilon}{I}}) - \sum_{j\in[J]} \tilde{d}^\star_{ijt}) = 0,$	(7.8)

P_f and D_f to denote the objective value on a feasible solution of P_f and D_f, respectively. According to Theorem 3, P_f can be obtained by $ORFA$. Given P_f, *a.k.a.* $P_f(OFRA)$, in order to obtain an upper bound of the ratio of $P_f(OFRA)$ to $P^\star_f$, we still need a lower bound of $P^\star_f$. However, it is not easy to solve the minimization problem (1) exactly. Since strong duality [27] guarantees that $P^\star_f = D^\star_f$, we resort to the lower bound of $D^\star_f$. Note that the dual fractional problem in (3) is a maximization, thus, any feasible solution produces a lower bound of $P^\star_f$, i.e., $D_f \leq D^\star_f = P^\star_f$. Therefore, the key point is to seek a feasible dual solution of (3) as the lower bound of the optimum of P_f, which will be proved in Theorem 4.

Looking into (3), our goal is to assign values of a set of dual variables a_{jt}, b_{it} and d_{ijt} within the feasible region defined by the constraints. Still, we explore the regularized problem $\tilde{P}_f$. Now we define $\tilde{D}_f$, the dual problem of $\tilde{P}_f$. Let $\tilde{a}_{jt}$ (≥ 0) and $\tilde{d}_{ijt}$ (≥ 0) be dual variables of $\tilde{D}_f$ associated with constraints (6a) and (6b), respectively. Based on Theorem 2 and Theorem 3, $\tilde{\mathbf{y}}^\star_{ft}$ and $\tilde{\mathbf{x}}^\star_{ft}$ satisfy the KKT Optimality Condition [27], which is sufficient and necessary for optimality of a convex optimization. We derive and present

$$\text{KKT Optimality Conditions} \tag{7}$$

of $\tilde{P}_f$ and $\tilde{D}_f$ in Table II. Directed by the KKT Optimality Condition, a feasible solution can be derived as follows:

$$a^0_{jt} = \tilde{a}^\star_{jt}, \quad d^0_{ijt} = \tilde{d}^\star_{ijt} \quad b^0_{it} = \frac{w_i}{\eta} \ln(\frac{1 + \frac{\epsilon}{I}}{\tilde{y}^\star_{i(t-1)} + \frac{\epsilon}{I}}) \tag{8}$$

In the following, to be clear, we use D^0_f to denote the dual problem constructed in (8).

Theorem 4. *$\sum_{t\in[T]} \sum_{j\in[J]} \tilde{a}^\star_{jt}$ is a lower bound of $P^\star_f$.*

Proof: According to (8), D^0_f is obtained by plugging each a^0_{jt} into the objective function of (3). In the following, we will further show that (8) is feasible to (3). Given $0 \leq \tilde{y}^\star_{i(t-1)} \leq 1$, we choose $\eta = \ln(1 + \frac{n}{\epsilon})$ and plug them into , we have

$$\begin{aligned} b_{it} &= \frac{w_i}{\ln(1 + \frac{I}{\epsilon})} \ln(\frac{1 + \frac{\epsilon}{I}}{\tilde{y}^\star_{i(t-1)} + \frac{\epsilon}{I}}) && \text{(by)} \\ 0 &\leq b_{it} \leq w_i. && \text{(by (1e))} \end{aligned}$$

which shows that (3a) and (3e) are feasible. We also plug into (3b) and have

$$\begin{aligned} &\sum_{j\in[J]} \tilde{d}^\star_{ijt} + b_{i(t+1)} - b_{it} \\ &= \sum_{j\in[J]} \tilde{d}^\star_{ijt} - \frac{w_i}{\eta} \ln(\frac{\tilde{y}^\star_{it} + \frac{\epsilon}{I}}{\tilde{y}^\star_{i(t-1)} + \frac{\epsilon}{I}}) && \text{(by)} \\ &\leq c_{it} && \text{(by (7.5))} \end{aligned}$$

which indicates (3b) is feasible. Feasibility of (3c) is naturally guaranteed by (7.5). Then we have $D^0_f \leq D^\star_f \leq P^\star_f$ according to weak duality [27]. Since $\sum_{t\in[T]} \sum_{j\in[J]} \tilde{a}^\star_{jt}$ is the objective value of D^0_f, which the theorem follows.

■

Based on Theorem 4, we then compare the total cost of fractional dynamic replica placement and request redirection problem using online algorithm $ORFA$ with that of the offline optimal algorithm. The total cost in (2) can be divided into three parts: total storage cost, total service cost and total migration cost. We compute the ratio of each of them to D_f respectively and then the competitive ratio can be bounded by summing up the three sub-ratios. In the following, we simplify $\tilde{\phi}^\star$ to ϕ where ϕ generally denotes all the variables.

Lemma 1. *The storage cost of $P_f(y_{it})$ is no larger than D^0_f.*

The proof is given in our technical report [26].

Lemma 2. *The service cost of $P_f(\tilde{y}^\star_{it})$ is no larger than D^0_f.*

The proof is given in our technical report [26].

Lemma 3. *The migration cost of $P_f(\tilde{x}^\star_{it})$ is no larger than $(1 + \epsilon)\ln(1 + \frac{I}{\epsilon})$ times of D^0_f.*

The proof is given in our technical report [26].

Theorem 5. *Our online algorithm $ORFA$ achieves a $((1 + \epsilon)\ln(1 + \frac{I}{\epsilon}) + 2)$-ratio compared to the offline optimum of the original problem in* (1).

Proof: Taking the sub-ratios in Lemma 1, 2, and 3 together, we have

$$\frac{P_f(OFRA)}{P^\star_f} \leq (1+\epsilon)\ln(1+\frac{I}{\epsilon})+1+1 = (1+\epsilon)\ln(1+\frac{I}{\epsilon})+2, \tag{9}$$

Since $P^\star_f$ is the optimum of the relaxation of P^{star}, we have

$$P_f(OFRA) \le \left((1+\epsilon)\ln(1+\frac{I}{\epsilon})+2\right) P_f^{\star} \le \left((1+\epsilon)\ln(1+\frac{I}{\epsilon})+2\right) P^{\star}. \tag{10}$$

■

V. An Online Algorithm for Replica Placement and Request Redirection

A. A Randomized Online Rounding Algorithm

In Sec. IV, our proposed $OFRA$ obtains a fractional solution of (2) which guarantees a ratio of $((1+\epsilon)\ln(1+\frac{n}{\epsilon})+2)$ compared to (1), the original problem. In order to satisfy (1e), the solution of $\mathbf{y}$ should be binary while $\mathbf{x}$ can be fractional but may require adjustment due to the rounding of $\mathbf{y}$. Let $\hat{y}_{it}$ denote the rounding solution of y_{it} and $\hat{x}_{ijt}$ denote the final solution of x_{ijt}. The main ideas to determine $\hat{y}_{it}$ and $\hat{x}_{it}$ are as follows.

Random, Dynamic Replica Placement. Looking into (2), the value of y_{it} is positively correlated with the tendency that data center i stores a replica at time slot t. The basic idea to round fractional y_{it} to binary $\hat{y}_{it}$ is to treat y_{it} as the probability $Pr[\hat{y}_{it} = 1]$. One of the state-of-the-art in random rounding algorithm design for online set covering problem (e.g., [28]) sheds light on our design of the online algorithm with random rounding, $RORA$. Unfortunately, directly applying their method on the rounding of y_{it} can not guarantee that all the requests are served, *i.e.*, with a small probability, none of the data centers store a replica. To address this, we force a selected data center i^0 to store the replica in the long time interval. However, it incurs extra storage cost and migration cost in data center i^0. To bound the total migration cost, we choose the data center with the smallest w_i as data center i^0. The ratio of storage cost is bounded by U_t and L_t.

Greedy Adaptive Request Redirection. Suppose data center i is an *available data center* if and only if $\hat{y}_{it} = 1$. We further define $\mathcal{D}_t$ to be the set of all *available data centers* at time slot t. The efficacy of our proposed request redirection strategy is presented in Proposition 1 as follows.

Proposition 1. *Given $\mathcal{D}_t$ at any time slot $t \in [T]$, for each $j \in [J]$, greedily redirecting all the requests from area j to the* available data center *with the smallest r_{ijt} is the optimal redirection strategy.*

Proof:

Suppose $\hat{y}_{it}$ for all i is given at time slot t, the storage cost of time slot t ($\sum_{i\in[I]} c_{it}\hat{y}_{it}$), is determined. Since $\hat{y}_{i(t-1)}$ for all i are known at time slot $t-1$, the migration cost of time slot t ($\sum_{i\in[I]} w_i \max\{0, \hat{y}_{it} - \hat{y}_{i(t-1)}\}$), can also be fixed. In such sense, no matter what the solutions of each $\hat{y}_{it}$ are, the solutions of each $\hat{x}_{ijt}$ can only influence the service cost at time slot t ($\sum_{j\in[J]}\sum_{i\in[I]} R_{ijt}\hat{x}_{ijt}$).

Combining the minimization goal with constraint (1a), we have $\sum_{i\in[I]} \hat{x}_{ijt} = 1$. According to constraint (1b), only for those available data centers $i \in \mathcal{D}_t$, we could have $\hat{x}_{ijt} > 0$ for any $j \in [J]$. Hence, $\sum_{j\in[J]}\sum_{i\in[I]} R_{ijt}\hat{x}_{ijt} = \sum_{j\in[J]}\sum_{i\in\mathcal{D}_t} R_{ijt}\hat{x}_{ijt}$. Thus $\sum_{i\in\mathcal{D}_t} R_{ijt}\hat{x}_{ijt}$ under any feasible solution of $\{\hat{x}_{ijt}\}_{i\in[\mathcal{D}_t]}$ is a convex combination of $\{R_{ijt}\}_{i\in\mathcal{D}_t}$. We know that all the convex combinations are within the convex hull [27], a line segment constructed by $\{R_{ijt}\}_{i\in\mathcal{D}_t}$. Then the critical points [27] of such a convex hull are end points of the line segment. Hence, the maximum of $\sum_{i\in\mathcal{D}_t} R_{ijt}\hat{x}_{ijt}$ is the one-dimensional coordinate of the right end point of the line segment, which is $\max_{i\in\mathcal{D}_t} R_{ijt}$. Note $R_{ijt} = r_{ij}n_{jt}$, thus we have $\max_{i\in\mathcal{D}_t} R_{ijt} = \max_{i\in\mathcal{D}_t} r_{ij}$

■

Algorithm 2: Randomized Online Rounding Algorithm $RORA$

```
   Input: $I$, $J$, $\mathbf{w}$, $\epsilon$
   Output: $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$
   Initialize: $\mathbf{x} = \hat{\mathbf{x}} = \mathbf{0}, \mathbf{y} = \hat{\mathbf{y}} = \mathbf{0}, \hat{\mathbf{z}} = \mathbf{0}, \mathcal{D}_t = \emptyset, \theta = \mathbf{0}$
1  foreach $i \in [I]$ do
2      Generate $i.i.d.$ $U(0,1)$ random variables:
       $Y(i,1), Y(i,2), ..., Y(i, 3\ln J)$;
3      $\Theta_i = \min_{m=1}^{3\ln J} Y(i,m)$;
4  end
5  Search the data center $i^0 = \mathrm{argmin}_{i\in[I]} w_i$;
6  while time slot $t$ starts, do
7      Get $\mathbf{c}_t, \mathbf{r}_t, \mathbf{n}_t, \mathbf{y}_{t-1}$;
8      $(\mathbf{x}_t, \mathbf{y}_t) = \tilde{P}_{ft}(I, J, \mathbf{c}_t, \mathbf{r}_t, \mathbf{w}, \mathbf{n}_t, \epsilon, \mathbf{y}_{t-1})$;
9      for $i = 1, ..., I$ do
10         if $\Theta_i \le y_{it}$ then
11             $\hat{y}_{it} = 1$;
12             $\mathcal{D}_t = \mathcal{D}_t \cup i$;
13         else
14             $y_{it} = 0$;
15         end
16     end
17     $\hat{y}_{i^0 t} = 1$;
18     for $j = 1, ..., J$ do
19         $i^\star = \min_{i\in\mathcal{D}_t\cup\{i^0\}} r_{ijt}$;
20         $\hat{x}_{i^\star jt} = 1$;
21         $\hat{x}_{ijt} = 0,\ \forall i \in [I]/i^\star$;
22     end
23     Return $\mathbf{y}_t, \hat{\mathbf{y}}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{z}}_t$;
24 end
```

Algorithm Intepretation. Directed by the discussion above, we design the online algorithm $RORA$, as given in Alg. 2, where the main steps of $OFRA$ are embedded in. Upon the arrival of the aggregated requests from all j at time slot t, a fractional solution is derived by $OFRA$ (line 8). Then y_{it} for all i are randomly rounded into $\hat{y}_{it}$ by treating the fractional y_{it} as probability (line $9-15$). Data center i^0 always stores a replica for reliability (line 17) since (i.) with a small probability, no data centers store a replica after the rounding (line $9-15$), and (ii.) the *origin* can only handle a restricted number of requests in practice. For each area $j \in [J]$, data center $i^\star$ with the minimal service cost r_{ijt} among all the *available data centers* is chosen to serve the requests (line $18-22$).

B. Competitive Analysis of $RORA$

Bounding the storage cost. Let U_t and L_t denote the upper bound and lower bound of c_{it} for all $i \in [I]$ and $t \in [T]$. The storage cost can be upper bounded as in Theorem 6.

Theorem 6. *The expected storage cost under the binary solution output by $RORA$ is no larger than $2\ln J + \frac{U_t}{L_t}$ times of that under the fractional solution output by $OFRA$.*

Proof:

We first analyze line 16 to line 22 of $RORA$. For each i and each single sample $1 \leq m \leq 2\ln J$, the probability that $Y(i,m) \leq y_{it}$ is exactly y_{it} . The probability that data center i is rounded to be 1 is the probability that there exists an m, $1 \leq m \leq 2\ln J$, such that $Y(i,m) \leq y_{it}$, i.e. $\Theta_i \leq y_{it}$. Define events $A_{imt} = \{Y(i,m) \leq y_{it}\}$, $A_{it} = \{\Theta_i \leq y_{it}\}$, and $A_{jt} = \{\exists i \in [\mathcal{B}_{jt}], A_{it} \text{ happens}\}$, respectively. Thus, the probability that i is rounded is to be 1 by line 15 is

$$Pr[A_{it}] = Pr[\bigcup_{m=1}^{2\ln J} A_{imt}] \tag{11}$$

By the union bound this probability is at most the sum of each of the probability of A_{imt}, we have

$$(11) \leq 2\ln J y_{it} \tag{12}$$

Thus for each area j at time slot t, we have

$$Pr[A_{jt}] \leq \sum_{i \in \mathcal{B}_{jt}} Pr[A_{it}] \leq 2\ln J \sum_{i \in \mathcal{B}_{jt}} y_{it} \tag{13}$$

We next consider the contribution of i^0 to the the expected storage cost at time slot t. According to line 5, we have

$$E_1[\hat{y}_{i^0 t} c_{i^0 t}] = c_{i^0 t} Pr[\hat{y}_{i^0 t}] = c_{i^0 t} \tag{14}$$

Due to constraint (1a) and (1b), we have

$$\sum_{i \in [I]} y_{it} \geq \sum_{i \in [I]} x_{ijt} \geq 1, \quad \forall j \in [J], t \in [T] \tag{15}$$

Thus the ratio of expected storage cost of time slot t under the rounding solution to that under the fractional solution is

$$\begin{aligned} E[\sum_{i \in [I]} c_{it}\hat{y}_{it}] &\leq \quad \frac{2\ln J \sum_{i \in [I]} c_{it} y_{it} + U_t \sum_{i \in [I]} y_{it}}{\sum_{i \in [I]} c_{it} y_{it}} \\ &\leq \quad 2\ln J + \frac{U_t \sum_{i \in [I]} y_{it}}{L_t \sum_{i \in [I]} y_{it}} \\ &= \quad 2\ln J + \frac{U_t}{L_t} \end{aligned} \tag{16}$$

■

Bounding the service cost. Let $U_s = \max_{i \in [I], j \in [J]} r_{ij}$ and $L_s = \min_{i \in [I], j \in [J]} r_{ij}$. In the following, we first compute the service cost of each area j at each time slot t under the fractional solution output by $OFRA$ and under the binary solution output by $RORA$, respectively. Based on Lemma 1, we look deeper into the structures of each x_{ijt} output by $OFRA$. For each i and t, suppose data center i is a *responsible data center* of j at t if and only if $x_{ijt} > 0$. Let $\mathcal{B}_{jt}$ denote the set of all *responsible data centers* of j at t. We further define $|\mathcal{B}_{max}|$ to denote the upper-bound of the number of data centers each area can be served. Note that $|\mathcal{B}_{max}| \leq I$. According to constraints (6a) and (6b), the optimal fractional solution of (6) has the following properties:

$$\begin{aligned} &\forall j \in [J], t \in [T]: \\ &\text{(i.)} \sum_{i \in \mathcal{B}_{jt}} y_{it} \geq 1; \\ &\begin{cases} \text{(ii.)}\ x_{ijt} \leq y_{it}, & \text{if } r_{ij} = \max_{i \in \mathcal{B}_{jt}} r_{ij} \\ \text{(iii.)}\ x_{ijt} = y_{it}, & \text{otherwise} \end{cases} \end{aligned} \tag{17}$$

We use *spare data center* to denote a data center which satisfies (ii.) of (17) and let σ_{jt} denote the index of *spare data center* for each j and t. Thus, the service cost of time slot t under the fractional solution output by $OFRA$ is

$$\sum_{j \in [J]} [\sum_{i \in \mathcal{B}_{jt}/\sigma_{jt}} y_{it} r_{ijt} + (1 - \sum_{i \in \mathcal{B}_{jt}/\sigma_{jt}} y_{it}) r_{\sigma_{jt}}] \tag{18}$$

For each i at each time slot j, we sort i with r_{ij} in an ascending order and let $\{i_{[q]}\}_{q \in [I]}$ to denote the index of a data center in the sorted sequence with $r_{i_{[q]}j} \leq r_{i_{[q+1]}j}$. Now we analyze the service cost for each area j at each time slot t produced by $RORA$. Given the rounding solution $\hat{y}_{it}$ for all i, greedy x_{ijt} for all j are optimal according to Lemma 1. We may and may not map all the requests from area j to some data center $i \in \mathcal{B}_{jt}$.

Lemma 4. *For each $k \in [|\mathcal{B}_{jt}|]$, we have*

$$Pr\left[\bigcup_{q=1}^{k} \left\{y_{i_{[q]}t=1}\right\}\right] \geq \left(1 - \frac{1}{J^2}\right) \sum_{q=1}^{k} y_{i_{[q]}t} \tag{19}$$

The proof is given in our technical report [26]. Further, the expected service cost will be bounded in Theorem 7 based on Lemma 4.

Theorem 7. *The expected service cost under the binary solution output by $RORA$ is no larger than $1 + \frac{|\mathbf{B}_{max}|}{J^2} \frac{U_s}{L_s}$ times of that under the fractional solution output by $OFRA$*

The proof is given in our technical report [26].

Bounding the migration cost. Since migration cost under the fractional solution is incurred for any data center i at any $t+1$ when $y_{i(t+1)} > y_{it}$, we define $\Delta_{i(t+1)} = y_{i(t+1)} - y_{it}$ for all i and t. Thus the total migration cost under the fractional solution is

$$\sum_{t \in [T]} \sum_{i \in [I]} w_i \max\{0, y_{i(t+1)} - y_{it}\} = \sum_{t \in [T]} \sum_{i \in [I]} w_i \max\{0, \Delta_{i(t+1)}\} \tag{20}$$

while the expected migration cost at time slot t is

$$E\left[\sum_{i\in[I]} w_i \max\{0, \hat{y}_{i(t+1)} - \hat{y}_{it}\}\right] \tag{21}$$

Theorem 8. *The expected migration cost under the binary solution output by $RORA$ is no larger than $2\ln J$ times of that under the fractional solution output by $OFRA$.*

The proof is given in our technical report [26].

Theorem 9. *The total cost under the binary solution output by $RORA$ is no larger than $\max\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s}\}$ times of that under the fractional solution output by $OFRA$.*

Proof: To simplify, let A, B, C denote the total storage cost, total service cost, and total migration cost under the fractional solution respectively. Let $E[COST]$ denote the expected total cost under the binary solution output by $RORA$. Taking Theorem 6, 7, and 8 together, we have

$$\begin{aligned}
&\frac{E[COST]}{A+B+C}\\
\leq&\frac{(2\ln J + \frac{U_t}{L_t})A + (1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s})B + (2\ln J + 1)C}{A+B+C}\\
\leq&\frac{\max\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s}, 2\ln J + 1\}(A+B+C)}{A+B+C}\\
=&\max\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s}\}
\end{aligned} \tag{22}$$

■

Theorem 10. *The final competitive ratio our $RORA$ is* $\max\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s}\} \times \left[(1+\epsilon)\ln(1+\frac{I}{\epsilon}) + 2\right]$.

Proof: The final solution output by $RORA$ is feasible to the original problem (1). Let $E[COST(RORA)]$ denote the expected total cost under the binary solution output by $RORA$, $COST(OFRA)$ denote the total cost under the fractional solution output by $OFRA$, and OPT denote the cost under the optimal solution of (1). Combining Theorem 9 and Theorem 5, the final competitive ratio is

$$\begin{aligned}
&\frac{E[COST(RORA)]}{OPT} = \frac{E[COST(RORA)]}{COST(OFRA)} \times \frac{COST(OFRA)}{OPT}\\
=&\max\left\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{|\mathcal{B}_{max}|}{J^2}\frac{U_s}{L_s}\right\}\left[(1+\epsilon)\ln(1+\frac{I}{\epsilon}) + 2\right]
\end{aligned} \tag{23}$$

■

In the following section, we will show the practical performance of the ratio by trace-driven simulations.

VI. Performance Evaluation

We conduct trace-driven simulations to evaluate the performance of Alg. 2. Based on the Amazon CloudFront pricing structure [29], we choose US, EU, South America (SA), Japan (JP), Singapore/Hong Kong (SHK), and Australia (AU) as the 6 request areas. The number of data centers located in each area is derived according to the distribution of zones of Amazon CloudFront [30].

We also compare the effectiveness of our Algorithm 2 with extended versions of two existing heuristics, *Greedy data center (GC)*, *Greedy Area (GA)*, proposed in [19] and a one-shot optimization. The basic idea of *GC* is: for each time slot t, we iteratively decide to place the replica on a data center with the maximum *utility* and assign to it with all the potential requests. The potential requests are those from all the areas within the QoS distance to the selected data center while not yet served in the current time slot. The *utility* of data center i at time slot t is equal to $\frac{\sum_{j\in\mathcal{Q}_{it}} n_{jt}}{\sum_{j\in\mathcal{Q}_{it}} n_{jt}r_{ijt}+c_{it}+w_i(1-y_{i(t-1)})}$, where $\mathcal{Q}_{it}$ is the set of all the areas within the QoS distance to data center i. The basic idea of *GA* is: for each time slot t and each area j, we redirect all the requests in area j with the lowest *potential cost*, where the *potential cost* of data center i for serving area j is equal to $n_{jt}r_{ijt} + c_{it} + w_i(1 - y_{i(t-1)})$.

A. Evaluation Setup and Prameter Settings.

Traffic Collection and Content Chunking We have proved in Theorem 1 that without the constraints of storage capacity and bandwidth capacity, optimizing on each unique content leads to an overall optimization over all the contents. Thus we take as input both VoD traces and downloads traces into our online algorithm respectively to evaluate the performance of our algorithm. The trace data is extracted from the pattern of user requests on Oct. 1, 2004 where each of the request demands for a VoD content of 100 minutes, 300 MB [31]. We also extract the downloads traces from the pattern of web traffic in [32]. In accordance with the model, we use content chunking [6] to split a content into multiple chunks, each of which is translated into a request in the corresponding time slot. Moreover, we split a VoD content into chunks of 5 minute duration, which is also the length of each time slot. As for the downloads traces, we split content into chunks of 50 MB in size. Subsequently, the length of each time slot is equal to $\frac{\text{size of a web page volume}}{\text{50MB}}$. Furthermore, we manually divide the overall requests into US, EU, SA, Asia & Pacific and Japan according to CloudFront Pricing Model [29], following a poisson distribution.

Cost Parameters. Based on the real pricing structure of Amazon CloudFront [29][33], we set the cost parameters as in Table III with minor adjustments. Our parameters are determined as follows. We believe that the pricing structure of CloudFront reflects the cost difference among areas such that it can be used as cost parameters in our problem. Storage cost and bandwidth cost for serving requests and migration are determined based on *storage cost* [33], *Regional Data Transfer Out to Internet* [29], and *Regional Data Transfer Out to Origin* [29] as follows:

TABLE III: Cost Parameters

requst area		US	EU	SA	JP	SHK	AU
$\bar{c}_{it}$		0.03	0.03	0.041	0.033	0.03	0.033
$\bar{r}_{ijt}$	US	0.085					
	EU		0.085				
	SA			0.140			
	JP				0.140		
	SHK					0.250	
	AU						0.140
w_i		0.020	0.020	0.060	0.060	0.125	0.100

Note that each area is indexed by a unique j in our model, which may contain multiple i's since a single location can host more than one data centers (*e.g.*, US). To be consistent with the interpretation in Sec. III, the parameter of *service cost per request* is set as r_{ijt} = unit bandwidth cost + $\alpha \times$ RTT. The round-trip time (RTT) between each pair of data center and request area are emulated using manually injected delays in programs following the formula RTT(ms) $= 0.02 \times$ Distance(km) $+ 5$ from [34]. The unit bandwidth is set based on the corresponding entry along the diagonal of the matrix in Table III. The RTT is ignored when request area is the same as the data center location. Since a latency up to 200ms [35] will deteriorate the user experience significantly, some r_{ijt} are set to be $+\infty$ to force the rejection of redirection such that the user experience is guaranteed. For example, the distance from Sao Paulo to France is about 9440 km, then average r_{ijt} from data centers in EU to request areas in SA is $(9440 \times 0.02 + 5) \times 0.1\% + 0.085 = 0.0.2788$, while it is $(9440 \times 0.02 + 5) \times 0.1\% + 0.140 = 0.3338$ from data centers in Sao Paulo (SA) to request areas in EU.

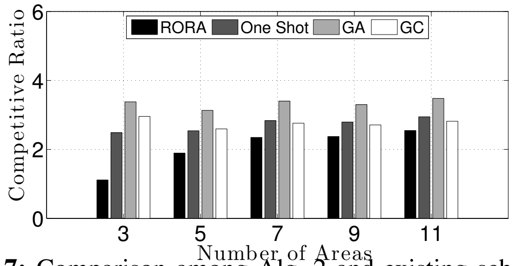

Fig. 7: Comparison among Alg. 2 and existing schemes

B. Comparison with Offline Optimum

We first study the competitive ratio achieved by $RORA$,computed by dividing the offline minimal cost (incurred by solving (1) exactly using MOSEK Optimizer) by the cost output by Alg. 2. Due to the complexity of solving the offline problem with a large number of variables, we set the default number of time slots to be 300.

Fig. 1 and Fig. 2 together illustrate how ϵ and I influence the performance of Alg. 1 and Alg. 2 respectively. The ratio of Alg. 1 is obtained through dividing the objective value of (1) under the fractional solution of Alg. 1 by the offline minimal cost. A larger ratio of both Alg. 1 and Alg. 2 comes with a larger number of data centers, similar to the impact of I on the theoretical ratio obtained in (9). Nevertheless, the impact of ϵ is different in Alg. 1 and Alg. 2, which can be explained as follows. The deterministic Alg. 1 optimally solves a well designed approximation of the relaxed original problem, therefore achieving a relatively stable performance. On the contrary, the randomized algorithm relies heavily on the repeated number of our simulation and thus more unstable. Moreover, both the ratios of Alg. 1 and Alg. 2 are under the average cases, which do not fit into the variation trend of the theoretical competitive ratio.

Deriving from the result from the two figures above, we choose a relative small ϵ rather than the optimal ϵ to minimize the theoretical ratio. More specifically, ϵ is equal to 0.1 in the following simulations. Fig. 3 further indicates that the more number of areas, the larger the competitive ratio. The reason is that, by using the trace data as input, we have $max\{2\ln J + \frac{U_t}{L_t}, 1 + \frac{I}{J^2}\frac{U_s}{L_s}\} = 2\ln J + \frac{U_t}{L_t}$, *i.e.*, a larger number of areas J leads a larger competitive ratio. Fig. 4 shows that the performance of Alg. 2 is not compromised with increasing number of time slots.

C. Comparison with Existing Schemes

We next compare the competitive ratios achieved by *RORA* and two heuristics (*Greedy data center (GC)* and *Greedy Area (GA)*) which are modified to the online fashion. We also use the MOSEK Optimizer to optimally solve the one-shot optimization at each time slot (we identify a lack of comparable online algorithms from the CDN cost-aware optimization literature). Fig. 5, 6, and 7 indicate that our proposed online algorithm $RORA$ can achieve better performance than the online version of the three offline algorithms. Additionally, we believe that both *GC* and *GA* perform well in offline setting since they are not surpassed much by the one-shot optimum.

VII. Conclusion

This work aims to efficiently manage cloud resources in different locations over time to minimize the operational cost of the CDN provider, while delivering short response delay to user requests. We first leverage a *regularization* method borrowed from online learning to reshape the relaxation of the original problem. Therefore the new problem can be decomposed into a set of sub-problems, each of which can be solved optimally in polynomial time. Such a fractional solution of the new problem is feasible to the relaxation of the original problem. Furthermore, we design a randomized rounding algorithm to obtain the final randomized binary solution. We show that the final solution to the original problem yields a competitive ratio which is independent of the number of requests or time horizon, based on the analysis via a primal-dual framework. The cost effectiveness is further validated by both theoretical proof and a series of trace-driven simulations.

Acknowledgments

This work was supported in part by a grant from Hong Kong RGC under the contract HKU 717812, the Natural Science and Engineering Research Council of Canada (NSERC) and Wedge Networks.

References

[1] J. Dilley, B. Maggs, J. Parihk, H. Prohop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Computing*, vol. 6, no. 5, pp. 50–58, 2002.

[2] C. Huang, A. Wang, J. Li, and K. W. Ross, "Measuring and Evaluating Large-Scale CDNs," in *Proc. of ICDCS*, 2014.

[3] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance internet applications," in *Proc. of ACM SIGOPS Operating Systems Review*, 2010.

[4] J. Broberg, R. Buyya, and Z. Tari, "MetaCDN: Harnessing Storage Clouds for high performance content delivery," *Journal of Network and Computer Applications*, vol. 32, no. 5, p. 10121022, 2009.

[5] "CDN Provider Akamai Expands Sercices to Costa Rica, Names Asia-Pacific Executives," goo.gl/BRvvk.

[6] A. Sharmar, A. Venkataramani, and R. K. Sitaraman, "Distributing Content Simplified ISP Traffic Engineering," in *Proc. of ACM SIGMETRICS*, 2013.

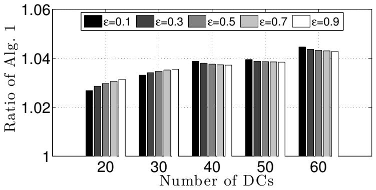

Fig. 1: Ratio of Alg. 1 to the relaxed original problem

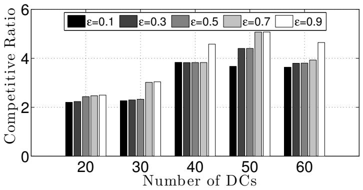

Fig. 2: Competitive Ratio of Alg. 2

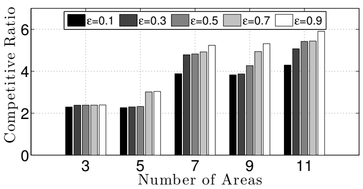

Fig. 3: Competitive Ratio of Alg. 2

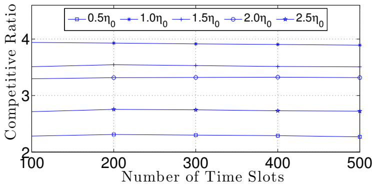

Fig. 4: Competitive Ratio of Alg. 2

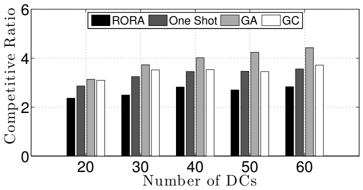

Fig. 5: Comparison among Alg. 2 and existing schemes

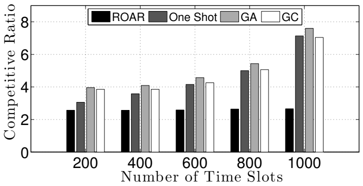

Fig. 6: Comparison among Alg. 2 and existing schemes

[7] N. Cese-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge University Press, 2006.

[8] A. Rakhlin, "Lecture notes on online learning draft," 2009.

[9] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive analysis via regularization," in *Proc. of ACM SODA*, 2014.

[10] B. Li, M. Golin, G. Italiano, and X. Deng, "On the optimal placement of web servers in the Internet," in *Proc. of IEEE INFOCOM*, 1999.

[11] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 5, pp. 568–582, 2000.

[12] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *Proc. of IEEE INFOCOM*, 2001.

[13] S. Jamin, C. Jin, A. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the Internet," in *Proc. of IEEE INFOCOM*, 2001.

[14] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-informed Internet replica placement," *Computer Communications*, vol. 25, no. 4, pp. 384–392, 2002.

[15] J. Xu, B. Li, and D. Lee, "Placement problems for transparent data replication proxy services," *IEEE JSAC*, vol. 20, no. 7, pp. 1383–1393, 2002.

[16] I. Cidon, S. Kutten, and R. Soffer, "Optimal allocation of electronic content," *Computer Networks*, vol. 40, no. 2, pp. 205–218, 2002.

[17] N. Bartolini, F. Lo, and P. Petrioli, "Optimal dynamic replica placement in content delivery networks," in *Proc. of IEEE Int. Conf. on Networking*, 2003.

[18] Y. Chen, R. Katz, and J. Kubiatowicz, "Dynamic Replica Placement for Scalable Content Delivery," in *Proc. of Intl Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[19] F. Chen, K. Guo, J. Lin, and T. Porta, "Intra-cloud Lightning: Building CDNs in the Cloud," in *Proc. of IEE INFOCOM*, 2010.

[20] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian, "Optimizing Cost and Performance for Content Multihoming," in *Proc. of ACM SIGCOMM*, 2012.

[21] V. Mathew, R. K. Sitaraman, and P. Shenoy, "Energy-Aware Load Balancing in Content Delivery Networks," in *Proc. of IEEE INFOCOM*, 2012.

[22] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic Right-Sizing for Power-Proportional Data Centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[23] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. Lau, "Scaling Social Media Applications into Geo-Distributed Clouds," *to apear in IEEE/ACM Transactions on Networking*.

[24] "Intel Cloud Builders Guide: Cloud Design and Deployment on Intel Platforms," White Paper, 2011.

[25] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. of ACM SIGCOMM*, 2011.

[26] "Online Cost Minimization for Operating Geo-distributed Cloud CDNs," Tech. Rep., `https://www.dropbox.com/s/pkylfngjevbwyjt/IWQoS15.pdf`.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[28] N. Buchbinder and J. Naor, "The design of online algorithms via a primal-dual approach," *Foundation and Trends in Theoretical Computer Science*, vol. 2-3, no. 3, pp. 93–263, 2009.

[29] "Amazon CloudFront Pricing," http://aws.amazon.com/cloudfront/pricing/.

[30] "distribution of data centers of Amazon CloudFront," http://aws.amazon.com/about-aws/global-infrastructure/.

[31] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-scale Video-on-Demand Systems," in *Proc. of ACM SIGOPS/EuroSys*, 2006.

[32] X. Wang, A. Abraham, and K. A. Smith, "Intelligent Web Traffic Mining and Analysis," *IEEE Transactions on Cloud Computing*, vol. 28, no. 2, pp. 147–165, 2005.

[33] "Amazon Simple Storage Service," http://aws.amazon.com/s3/pricing/.

[34] A. Qureshi, "Power-Demand Routing in Massive Geo-Distributed Systems," *Ph.D. dissertation, Massachusetts Institute of Technology*, 2010.

[35] R. Kuschnig, I. Kofler, and H. Hellwagner, "Improving Internet Video Streaming Performance by Parallel TCP-based Request-Response Streams," in *Proc. of CCNC*, 2010.