# Smoothed Online Resource Allocation in Multi-Tier Distributed Cloud Networks

Lei Jiao*, Antonia Tulino§, Jaime Llorca§, Yue Jin*, Alessandra Sala*
*Bell Labs, Dublin, Ireland
§Bell Labs, New Jersey, USA

***Abstract*—In the emerging edge computing paradigm, small-scale highly distributed edge clouds are on the service path between end users and conventional large-scale clouds at the Internet core. A crucial problem that needs to be addressed in order to drive cost and performance in this multi-tier distributed infrastructure is the dynamic and joint allocation of cloud and network resources, which is particularly challenging due to the coexistence of several factors: the reconfiguration cost associated to changing resource allocation decisions over time, the constantly varying and often unpredictable nature of service demands, as well as the heterogeneity of distributed resources.**

**We study the problem of resource allocation and reconfiguration in the multi-tier resource pool from an online optimization perspective that addresses all the challenges above. Our approach decouples the original problem over time by constructing a series of subproblems that are solvable at each corresponding time slot using the output of the previous time slot. Via solid formal analysis, we prove that, without any lookahead beyond the current time slot, our online algorithm provides a solution with a parameterized competitive ratio for any arbitrarily dynamic workload and operating price. We conduct extensive evaluations in a variety of settings based on a number of clouds and real-world workloads with regular and flash crowd fluctuations, and demonstrate that our online algorithm performs well in practice, achieving up to $9\times$ total cost reduction than the sequence of one-shot optimizations and at most $3\times$ the offline optimum.**

## I. Introduction

Clouds are moving closer to end users [5], [19], [20], [22], which enables major improvements in key service performance metrics such as latency (via service proximity), reliability (via service redundancy), and privacy (via local or regional data storage). Small-scale highly distributed edge clouds can be built at network operators' existing points of presence, or implemented separately at metro, branch, or even customer premises. Introducing the edge cloud into the service path between end users and large commercial clouds at the Internet core results in a multi-tier hierarchic infrastructure, as shown in Fig. 1. A motivating scenario can be using this infrastructure to deploy Virtual Network Functions (VNFs) and service chains. The VNFs at lower-tier clouds are typically those that benefit from the proximity to end users, and the VNFs at upper-tier clouds are those that benefit more from the economy of scale of the large resource pool. User requests or flows are firstly processed at the lower-tier clouds, and afterwards forwarded upstream and processed at the upper-tier clouds.

To exploit the great potential of this multi-tier distributed infrastructure, a critical problem that needs to be addressed is the joint allocation of cloud and network resources across the hierarchic resource pool. Challenged by the reconfiguration cost, workload dynamics, and resource heterogeneity, this problem roots its difficulty in both time and space dimensions.

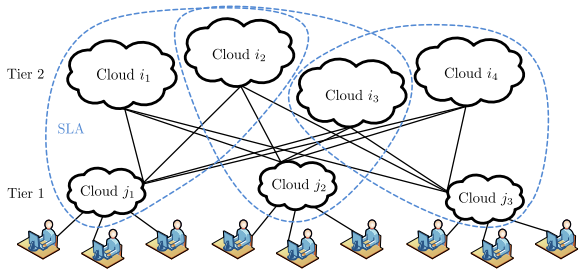



**Fig. 1:** The tiered resource pool of clouds and networks

In the time dimension, resource allocation needs to be balanced with resource reconfiguration over time, and striking this balance dynamically under unpredictable workloads requires online decision making. Allocation cost refers to the operating cost incurred by using physical and virtual resources such as Physical Machines (PMs), Virtual Machines (VMs), and networks. Reconfiguration cost is incurred by changing resource allocation decisions over time via switching on/off resources, and can capture service interruption [17], hardware wear and tear [11], risk and system instability [27], as well as resource lead time (*e.g.*, when booting a VM) [13]. On one hand, one prefers to allocate just sufficient resources to process the workload to avoid over provisioning and minimize the operating cost; on the other hand, one may desire the resource allocation decisions to be as smooth as possible to avoid sharp changes over time that can incur excessive reconfiguration cost. Striking the right balance is particularly hard in an online setting, where a resource allocation decision needs to be made on the fly without prior knowledge about the workload in the future. The operating price may also vary unpredictably and influence the resource allocation decisions as well.

In the space dimension, the heterogeneity and the geographic distribution of multi-tier resources requires the joint, multi-dimensional optimization of clouds and networks across locations and tiers, while respecting capacity limits [14], [24] and service quality requirements [9], [15]. Unlike gigantic upper-tier clouds where resources may be considered "infinite", lower-tier clouds and networks often impose limited capacities, and are diverse in resource prices. To process the incoming workload from an edge cloud, for instance, maybe only a particular subset of the upper-tier clouds can satisfy the

1530-2075/16 $31.00 © 2016 IEEE
DOI 10.1109/IPDPS.2016.35



IEEE computer society

specified Service Level Agreement (SLA) in terms of latency, security risk, reliability, and so on. Thus, at different upper-tier clouds, resources need to be allocated and reconfigured to handle workloads from different edge clouds. Such factors add additional complexities to the online optimization problem.

Existing researches do not capture the emerging distributed multi-tier cloud networks architecture, and more importantly, they either ignore the reconfiguration cost [7], [8], [12], [25], or assume the lookahead into the future so that their results highly depend on the capability of prediction [10], [23], [26], [27]. The gap remains: for the essentially unpredictable workloads, how to allocate resources online in a multi-tier cloud and network resource pool so that the total allocation and reconfiguration cost over time is minimized?

In this paper, we aim to fill this gap by developing formal models, online algorithms and competitive analysis to capture, solve and characterize the optimization problem of smoothly allocating resources in multi-tier cloud networks for unpredictably time-varying workloads. We make three contributions:

We build models that can capture a range of real-world costs and we formulate the smoothed online resource allocation problem. The allocation cost is modeled as affine functions of resource units of clouds and networks, which can capture, *e.g.*, the pay-as-you-go billing scheme for resource usage and electricity consumption. The reconfiguration cost is modeled as only paying for the increase of the amount of allocated resources from one time slot to the next, which can capture, *e.g.*, PM and VM booting and lead time. SLA is modeled using the subset approach, *i.e.*, for a lower-tier cloud, only a cloud in a specified subset of the upper-tier clouds can satisfy the SLA requirement. We do not enforce how such subsets are determined or what criterion is used. We also make no assumption on workload and operating price dynamics.

We design an online algorithm by exploiting the technique of regularization [4], and formally prove that our online algorithm provides a solution with a parameterized competitive ratio independent of workload and operating price dynamics. Fundamentally different from existing work, our approach, without any lookahead, decouples the original problem over time by constructing a series of subproblems where the optimal decision of a subproblem at a time slot depends on the workload at that time slot and the decision of the subproblem at the previous time slot, and uses the sequence of decisions to this series of subproblems as the solution to our original problem. The intuition behind our algorithm is that, when the workload increases, we allocate just enough resources to cover the workload, and when the workload decreases sharply, we do not reduce the resource allocation immediately to match the workload and instead we take a controlled exponential-decay reduction in the amount of allocated resources to avoid excessive reconfiguration cost as the workload may increase later. We derive the optimality guarantee for our algorithm via rigorous competitive analysis for two tiers of clouds, and generalize such a guarantee to arbitrary $N \geqslant 2$ tiers of clouds.

We conduct extensive numerical evaluations based on real-world data traces. Using the 18 AT&T clouds in North America as tier-2 clouds and one tier-1 cloud per continental US state, and using the realistic dynamic electricity price and the estimated bandwidth price as the operating prices, we run the sequence of one-shot optimizations, our online algorithm, and the offline optimization to allocate and reconfigure resources for the 2007 Wikipedia workload of 500 hours with regular dynamics and for the 1998 World Cup workload of 600 hours with large spikes, respectively. Through a number of different settings, we demonstrate that our online algorithm performs consistently well in practice, achieving up to $9\times$ total cost reduction over time than one-shot optimizations and at most $3\times$ the offline optimum.

## II. Model Formulation

### A. Models and Notations

***System***: Clouds are geographically distributed and organized in tiers, as shown in Fig. 1. Tier-1 clouds, indexed by $j \in \mathcal{J}$, are edge clouds (*e.g.*, at metro points of presence) located in close proximity to the end users in each region. Tier-2 clouds, indexed by $i \in \mathcal{I}$, are larger clouds located at the Internet core, which are typically public clouds or enterprise clouds that host services offered to end users or customers. Note that tier-1 clouds are on the path between users and tier-2 clouds, *i.e.*, to reach a tier-2 cloud, a user's requests or flows must go through the regional tier-1 cloud. All users in a region are connected to their corresponding tier-1 cloud, and a tier-1 cloud can potentially connect to all the tier-2 clouds.

We model the cloud resources of tier-1 and tier-2 clouds, as well as the network resources between tier-1 and tier-2 clouds. Tier-2 cloud $i$ has capacity $C_i$, unit allocation cost (*i.e.*, the operating price) $a_{it}$ which may be time-varying, and unit reconfiguration cost (*i.e.*, the reconfiguration price) $b_i$. Analogously, tier-1 cloud $j$ has capacity $C_j$, unit allocation cost $e_{jt}$, and unit reconfiguration cost $f_j$. The network between tier-2 cloud $i$ and tier-1 cloud $j$ has capacity $B_{ij}$, unit allocation cost $c_{ijt}$, and unit reconfiguration cost $d_{ij}$. In a time-slotted system, the allocation cost pays for the amount of allocated resources at every time slot, such as energy and bandwidth expense; in contrast, the reconfiguration cost only pays for the increase of the amount of resources across consecutive time slots to capture the fact that, *e.g.*, booting PMs or VMs incurs considerable time while shutting them down is often fast.

***Workload***: We target web services workload and alike, and use $\lambda_{jt}$ to denote the aggregated workload, *e.g.*, in terms of the number of requests, received at edge cloud $j$ at time slot $t$. User requests are first processed at the local edge cloud and then at one of the clouds at the upper tier that host the target service. The workloads at different edge clouds can be different, and change over time. We make no assumption on workload dynamics and statistical distributions, and allow the workload of each edge cloud to vary arbitrarily and independently. We model a time-slotted system where each time slot $t \in \{1, 2, 3, ..., T\}$ corresponds to a resource allocation decision at all clouds and inter-cloud networks across tiers.

***SLA***: We model the SLA requirements as the selections of clouds at the upper tier. For each tier-1 cloud $j$, there exists

a subset of tier-2 clouds, denoted as $\mathcal{I}_j$, that satisfy the SLA requirement, meaning that the latency, security risk, reliability, and so on as in the SLA specification can be satisfied if user requests received at cloud $j$ are routed to any cloud in $\mathcal{I}_j$. Correspondingly, $\mathcal{J}_i$ refers to the subset of tier-1 clouds for which the tier-2 cloud $i$ can satisfy the SLA. Taking Fig. 1 as an example, we have $\mathcal{J}_{i_2} = \{j_1, j_2\}$, $\mathcal{I}_{j_1} = \{i_1, i_2\}$, $\mathcal{I}_{j_2} = \{i_2, i_3\}$. In case of a system with more than two tiers of clouds, an edge cloud receives the requests and sends them to a cloud at the top tier eventually for processing. Multiple paths may exist to satisfy the SLA and to reach one of the clouds at the top tier via different clouds at the intermediate tiers.

### *B. Problem Formulation*

Based on the models as described above, we formulate the total cost, including the allocation cost and the reconfiguration cost, at the two tiers of clouds and at the network between them. We use $x_{ijt}$ to denote the amount of resources allocated at cloud $i$ to process the incoming workload from cloud $j$ at time $t$, $y_{ijt}$ to denote the amount of resources allocated at the network between clouds $i$ and $j$ to transport the workload from cloud $j$ to cloud $i$ at time $t$, and $z_{ijt}$ to denote the amount of resources allocated at cloud $j$ to process the workload that is sent to cloud $i$ for processing at time $t$:

$$F_1 = \sum_t \sum_j \sum_{i\in\mathcal{I}_j} e_{jt} z_{ijt} + \sum_t \sum_j f_j \left[ \sum_{i\in\mathcal{I}_j} z_{ijt} - \sum_{i\in\mathcal{I}_j} z_{ijt-1} \right]^+,$$

$$F_{12} = \sum_t \sum_j \sum_{i\in\mathcal{I}_j} c_{ijt} y_{ijt} + \sum_t \sum_j \sum_{i\in\mathcal{I}_j} d_{ij} \left[ y_{ijt} - y_{ijt-1} \right]^+,$$

$$F_2 = \sum_t \sum_i \sum_{j\in\mathcal{J}_i} a_{it} x_{ijt} + \sum_t \sum_i b_i \left[ \sum_{j\in\mathcal{J}_i} x_{ijt} - \sum_{j\in\mathcal{J}_i} x_{ijt-1} \right]^+.$$

Then we formulate the dynamic resource allocation problem as follows, where $\sum_i \sum_{j\in\mathcal{J}_i} x_{ij} = \sum_j \sum_{i\in\mathcal{I}_j} x_{ij}$, $\forall x_{ij}$ and $[x]^+ \triangleq \max\{x, 0\}$, $\forall x$:

$$\begin{aligned}
\min \quad & F_1 + F_{12} + F_2 \\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}_j} \min\{x_{ijt}, y_{ijt}, z_{ijt}\} \geqslant \lambda_{jt}\,, \forall j\,, \forall t\,, && \text{(1a)} \\
& \sum_{j\in\mathcal{J}_i} x_{ijt} \leqslant C_i\,, \forall i\,, \forall t\,, && \text{(1b)} \\
& y_{ijt} \leqslant B_{ij}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,, && \text{(1c)} \\
& \sum_{i\in\mathcal{I}_j} z_{ijt} \leqslant C_j\,, \forall j\,, \forall t\,, && \text{(1d)} \\
& x_{ijt} \geqslant 0\,, y_{ijt} \geqslant 0\,, z_{ijt} \geqslant 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,. && \text{(1e)}
\end{aligned}$$

The objective is to minimize the total cost over time. Recall our workload model where user requests or flows are processed at each tier and transported by the network between tiers. So, Constraint (1a) ensures sufficient resources along the service path; Constraints (1b), (1c), and (1d) ensure that the resource allocation can only be done within the capacity.

By introducing the axillary variable $s_{ijt}$, we can rewrite the problem as follows:

$$\begin{aligned}
\min \quad & F_1 + F_{12} + F_2 \\
\text{s.t.} \quad & x_{ijt} \geqslant s_{ijt}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,, && \text{(2a)} \\
& y_{ijt} \geqslant s_{ijt}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,, && \text{(2b)} \\
& z_{ijt} \geqslant s_{ijt}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,, && \text{(2c)} \\
& \sum_{i\in\mathcal{I}_j} s_{ijt} \geqslant \lambda_{jt}\,, \forall j\,, \forall t\,, && \text{(2d)} \\
& s_{ijt} \geqslant 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \forall t\,, && \text{(2e)} \\
& \text{(1b)}\,, \text{(1c)}\,, \text{(1d)}\,.
\end{aligned}$$

For the problem to be feasible, the following inequalities must be satisfied: $C_j \geqslant \lambda_{jt}$, $\forall j$, $\forall t$; $\sum_{i\in\mathcal{I}_j} B_{ij} \geqslant \lambda_{jt}$, $\forall j$, $\forall t$; $\sum_i C_i \geqslant \sum_j \lambda_{jt}$, $\forall t$. These three inequalities correspond to constraints (1d), (1c), and (1b), respectively.

Due to the highly analogous structure of $F_2$ and $F_1$, we remove $F_1$ and its corresponding constraints (2c) and (1d) from our problem for the ease of presentation. All the techniques that we develop in this paper are naturally applicable to the problem that has $F_1$, (2c) and (1d). In the rest of this paper, we focus on the following problem that we name $\mathbf{P_1}$:

$$\begin{aligned}
\min \quad & F_{12} + F_2 \\
\text{s.t.} \quad & \text{(2a)}\,, \text{(2b)}\,, \text{(2d)}\,, \text{(2e)}\,, \text{(1b)}\,, \text{(1c)}\,.
\end{aligned}$$

## III. Algorithm and Optimality

### *A. Key Idea*

In a typical online setting, the "competitive ratio" is often used to quantify the quality of the solution produced by an online algorithm. To make decisions for a series of time slots, an online algorithm, to which the input is revealed incrementally and only a piece at a time, makes a decision for the current time slot on the fly; an offline algorithm, to which the entire input is assumed to be revealed all at once, makes decisions for all time slots at one time. The competitive ratio, independent of the input, refers to the ratio of the over time cost incurred by the online decisions over that incurred by the offline optimal decisions. We aim to propose an online algorithm and also analyze its competitive ratio in this paper.

The major difficulty in solving the problem $\mathbf{P_1}$ online lies in the reconfiguration cost which couples every two consecutive time slots. The resource allocation decision for a time slot can influence the reconfiguration cost between this time slot and its next time slot—without knowing the workload of the next time slot, it is hard to make a good decision for this time slot.

To conquer such difficulty, we exploit the regularization technique to decouple the original problem $\mathbf{P_1}$ by constructing a series of subproblems $\{\mathbf{P_2^{(1)}}, \mathbf{P_2^{(2)}}, \ldots, \mathbf{P_2^{(t)}}, \ldots, \mathbf{P_2^{(T)}}\}$. Denoting by $(x_t^*, y_t^*)$ the optimal solution to $\mathbf{P_2^{(t)}}$, we use the sequence $\{x_1^*, y_1^*, x_2^*, y_2^*, \ldots, x_t^*, y_t^*, \ldots, x_T^*, y_T^*\}$ as the solution to $\mathbf{P_1}$ (Lemma 1 in the next section shows this sequence is feasible for $\mathbf{P_1}$). Without knowing the workload of the next time slot, this approach enables us to make an appropriate decision for the current time slot with bounded

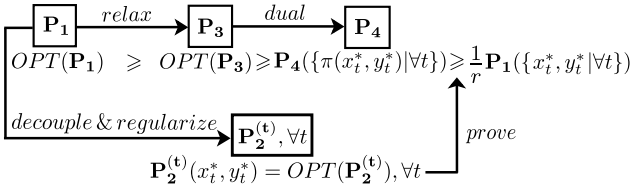



**Fig. 2:** Key idea

proximity to the offline optimum, based on the decision of the previous time slot and the workload of the current time slot.

Our key idea for algorithm design and competitive analysis is illustrated in Fig. 2. We proceed through the following steps:

- **Step 1:** Construct $\mathbf{P_2^{(t)}}$ whose optimal solution $(x_t^*, y_t^*)$ is feasible for $\mathbf{P_1}$ at $t$;
- **Step 2:** Construct $\mathbf{P_3}$ by relaxing $\mathbf{P_1}$, and derive $\mathbf{P_4}$, the Lagrange dual problem of $\mathbf{P_3}$;
- **Step 3:** Construct the mapping $\pi$ which maps $(x_t^*, y_t^*)$ to a solution feasible for $\mathbf{P_4}$ at $t$;
- **Step 4:** Prove $\mathbf{P_1}(\{x_t^*, y_t^*|\forall t\}) \leqslant r\mathbf{P_4}(\{\pi(x_t^*, y_t^*)|\forall t\})$.

Let us denote by $\mathbf{P_i}(x)$ the objective function value of the problem $\mathbf{i}$ evaluated at $x$ and denote by $OPT(\cdot)$ the offline optimal objective function value. From **Steps 2** and **3**, it naturally follows $\mathbf{P_4}(\{\pi(x_t^*, y_t^*)|\forall t\}) \leqslant OPT(\mathbf{P_3}) \leqslant OPT(\mathbf{P_1})$ due to weak duality and relaxation, respectively. By **Step 1**, we achieve **Step 4**, from which it follows $\mathbf{P_1}(\{x_t^*, y_t^*|\forall t\}) \leqslant rOPT(\mathbf{P_1})$, and consequently $r$ is the competitive ratio.

### *B. Algorithm Design*

Our online algorithm solves $\mathbf{P_2^{(t)}}$, $\forall t \in \{1, \ldots, T\}$, taking the optimal solution of $\mathbf{P_2^{(t-1)}}$ and the workload at $t$ as input. At $t = 0$ where $\mathbf{P_2^{(0)}}$ is undefined, we set its "optimal solution" to zero. We construct the following formulation as $\mathbf{P_2^{(t)}}$:

$$\begin{aligned}
\min \quad & F_t = \sum_i \sum_{j\in\mathcal{J}_i} a_{it}x_{ijt} + \sum_j \sum_{i\in\mathcal{I}_j} c_{ijt}y_{ijt} \\
& + \sum_i \frac{b_i}{\eta_i}\left(\left(\sum_{j\in\mathcal{J}_i} x_{ijt} + \varepsilon\right)\ln\frac{\sum_{j\in\mathcal{J}_i} x_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \sum_{j\in\mathcal{J}_i} x_{ijt}\right) \\
& + \sum_j \sum_{i\in\mathcal{I}_j} \frac{d_{ij}}{\eta'_{ij}}\left((y_{ijt} + \varepsilon')\ln\frac{y_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} - y_{ijt}\right) \\
\text{s.t.} \quad & x_{ijt} \geqslant s_{ijt}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, && \text{(4a)} \\
& y_{ijt} \geqslant s_{ijt}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, && \text{(4b)} \\
& \sum_{i\in\mathcal{I}_j} s_{ijt} \geqslant \lambda_{jt}\,, \forall j\,, && \text{(4c)} \\
& \sum_{\substack{k\in\mathcal{I}\\k\neq i}} \sum_{j\in\mathcal{J}_i} x_{kjt} \geqslant \sum_j \lambda_{jt} - C_i\,, \forall i\,, && \text{(4d)} \\
& \sum_{\substack{k\in\mathcal{I}_j\\k\neq i}} y_{kjt} \geqslant \lambda_{jt} - B_{ij}\,, \forall i \in \mathcal{I}_j\,, \forall j\,, && \text{(4e)} \\
& s_{ijt} \geqslant 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, && \text{(4f)}
\end{aligned}$$

where $(x^*_{ijt-1}, y^*_{ijt-1})$, satisfying $x^*_{ij0} = y^*_{ij0} = 0$, is the optimal solution to $\mathbf{P_2^{(t-1)}}$, and $\varepsilon$, $\varepsilon'$, $\eta_i$, $\eta'_{ij}$ are the parameters:

$$\varepsilon > 0\,, \varepsilon' > 0\,, \eta_i = \ln\left(1 + \frac{C_i}{\varepsilon}\right)\,, \eta'_{ij} = \ln\left(1 + \frac{B_{ij}}{\varepsilon'}\right)\,.$$

Note that $\mathbf{P_2^{(t)}}$ is a convex optimization problem. When formulating the objective of $\mathbf{P_2^{(t)}}$, we "regularize" the reconfiguration cost by replacing the function $[\cdot]^+$ (recall $[x]^+ = \max\{x, 0\}$) with a logarithmic function. Furthermore, we reformulate constraints (2a), (2d) and (1b) in $\mathbf{P_1}$, introducing (4d) in $\mathbf{P_2^{(t)}}$, and analogously for (2b), (2d) and (1c) in $\mathbf{P_1}$, we introduce (4e) in $\mathbf{P_2^{(t)}}$.

We state the following lemma to show the feasibility of the sequence $\{x_1^*, y_1^*, x_2^*, y_2^*, \ldots, x_t^*, y_t^*, \ldots, x_T^*, y_T^*\}$ for $\mathbf{P_1}$:

**Lemma 1.** *$(x_t^*, y_t^*)$ is feasible for $\mathbf{P_1}$ at $t$.*

*Proof:* We prove this lemma by showing that $(x_t^*, y_t^*)$, as the optimal solution to $\mathbf{P_2^{(t)}}$, while satisfying $\mathbf{P_2^{(t)}}$'s constraints (4a)-(4f), also satisfies $\mathbf{P_1}$'s constraints (2a), (2b), (2d), (2e), (1b) and (1c) at the same $t$. Note that $x_t^*$ and $y_t^*$, for the ease of presentation, actually refer to $x^*_{ijt}$ and $y^*_{ijt}$, $\forall i \in \mathcal{I}_j$, $\forall j$.

$$\frac{\partial F_t}{\partial x_{ijt}} = a_{it} + \frac{b_i}{\eta_i}\ln\frac{\sum_{j\in\mathcal{J}_i} x_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \geqslant 0\,,$$

when $x_{ijt} \geqslant x^*_{ijt-1}\,, \forall i \in \mathcal{I}_j\,, \forall j$, and

$$\frac{\partial F_t}{\partial y_{ijt}} = c_{ijt} + \frac{d_{ij}}{\eta'_{ij}}\ln\frac{y_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \geqslant 0\,,$$

when $y_{ijt} \geqslant y^*_{ijt-1}\,, \forall i \in \mathcal{I}_j\,, \forall j$. That is, $F_t$ increases monotonically for $x_{ijt} \geqslant x^*_{ijt-1}$ and $y_{ijt} \geqslant y^*_{ijt-1}$, and drops when we reduce $x_{ijt}$ to $x^*_{ijt}$ from a value that is larger than $x^*_{ijt}$ and reduce $y_{ijt}$ to $y^*_{ijt}$ from a value that is larger than $y^*_{ijt}$. With $\sum_{j\in\mathcal{J}_i} x^*_{ij0} = 0 \leqslant C_i$ and $y^*_{ij0} = 0 \leqslant B_{ij}$, the value of $F_t$ is reduced when we reduce $x_{ij1}$ until $\sum_{j\in\mathcal{J}_i} x_{ij1} = C_i$ holds and reduce $y_{ij1}$ until $y_{ij1} = B_{ij}$ holds, *i.e.*, we will have $\sum_{j\in\mathcal{J}_i} x^*_{ij1} \leqslant C_i$ and $y^*_{ij1} \leqslant B_{ij}$, as required by (1b) and (1c) at $t = 1$. Analogously, $\forall t \geqslant 2$, (1b) and (1c) hold. ■

### *C. Geometric Interpretation*

To understand how the optimal decisions from $\mathbf{P_2^{(t)}}$, $\forall t$ actually dictate the resource allocation, we consider a simplified version of our smoothed online resource allocation problem at a single data center with a time-varying workload, which is formulated as

$$\begin{aligned}
\min \quad & \sum_t a_t x_t + \sum_t b[x_t - x_{t-1}]^+ \\
\text{s.t.} \quad & x_t \geqslant \lambda_t\,, \forall t\,, && \text{(5a)} \\
& x_t \leqslant C\,, \forall t\,. && \text{(5b)}
\end{aligned}$$

Replacing $b[x_t - x_{t-1}]^+$, we have

$$a_t x_t + \frac{b}{\eta}\left((x_t + \varepsilon)\ln\frac{x_t + \varepsilon}{x^*_{t-1} + \varepsilon} - x_t\right) \qquad (6)$$

where $\eta = \ln(1+C/\varepsilon)$. The problem is further decoupled over time slots. At each time slot $t \geqslant 1$, we minimize (6) subject to (5a) and (5b) at the corresponding time slot, with $x_0^* = 0$.

By setting the derivative of (6) to zero, we get its constraint-free minimizer $\widetilde{x_t}$ as

$$\widetilde{x_t} = \left(1+\frac{C}{\varepsilon}\right)^{-\frac{a_t}{b}}(x_{t-1}^*+\varepsilon)-\varepsilon \leqslant x_{t-1}^*. \tag{7}$$

With constraints (5a) and (5b), we know that at $t$, if $\lambda_t > \widetilde{x_t}$, then $x_t^* = \lambda_t$; if $\lambda_t \leqslant \widetilde{x_t}$, then $x_t^* = \widetilde{x_t}$.

Let us consider $w+1$ (where $w$, and $w'$ in the following, are integers) consecutive time slots $t$, $t+1$, ..., $t+w$ with the workload $\lambda_t < \lambda_{t+1} < ... < \lambda_{t+w}$:

- In the case of $\lambda_t > \widetilde{x_t}$, we have $x_{t+w'}^* = \lambda_{t+w'}$, $\forall w'$, where $1 \leqslant w' \leqslant w$. This is because $\lambda_t > \widetilde{x_t}$ gives $x_t^* = \lambda_t$ and further gives $\lambda_{t+1} > \lambda_t = x_t^* \geqslant \widetilde{x_{t+1}}$, and afterwards, $\lambda_{t+1} > \widetilde{x_{t+1}}$ gives $x_{t+1}^* = \lambda_{t+1}$. This procedure can continue for any $w'$, where $1 < w' \leqslant w$. Here, the resource allocation follows the workload.
- In the case of $\lambda_t \leqslant \widetilde{x_t}$, by applying the equation in (7) iteratively, we have

$$x_{t+w'}^* = \widetilde{x_{t+w'}} = \left(1+\frac{C}{\varepsilon}\right)^{-\frac{1}{b}\sum_{t'=1}^{w'} a_{t+t'}}(\widetilde{x_t}+\varepsilon)-\varepsilon,$$

  if $\lambda_{t+w'} \leqslant \widetilde{x_{t+w'}}$, $\forall w'$, where $1 \leqslant w' \leqslant w$. Here, if $a_t$ does not vary with $t$, the resource allocation follows the exponential decay; if $a_t$ varies but is bounded by some constant, the resource allocation is also bounded by the corresponding exponential decay curve.

Our online algorithm always tries to allocate resources following an exponential decay curve (or a curve bounded by the exponential decay as explained in the above) for an arbitrarily time-varying workload. At a time slot, the actual amount of allocated resources depends on which is larger: the "expected" amount of resources calculated according to the current exponential decay or the actual workload at the current time slot. If the former is larger, then what has been calculated is the amount to allocate; if the latter is larger, then it allocates just enough resources to cover the workload. Note that in the latter case, the decay curve changes correspondingly. At the next time slot, our algorithm will calculate the "expected" amount of resources following the new decay curve, and compare it with the actual workload of the next time slot.

### *D. Competitive Analysis*

**Theorem 1.** *Our online algorithm produces a solution to* $\mathbf{P_1}$ *with a competitive ratio of* $r = 1+|\mathcal{I}|\left(C(\varepsilon)+B(\varepsilon')\right)$, *where* $C(\varepsilon) = \max_{i\in\mathcal{I}}\left\{(C_i+\varepsilon)\ln\left(1+\frac{C_i}{\varepsilon}\right)\right\}$ *and* $B(\varepsilon') = \max_{i\in\mathcal{I}_j, j\in\mathcal{J}}\left\{(B_{ij}+\varepsilon')\ln\left(1+\frac{B_{ij}}{\varepsilon'}\right)\right\}$.

The rest of this section, following the steps described in Section III-A, analyzes why and how we get such a competitive ratio, which also serves as the proof to the above theorem. **Step 1** has been addressed in Section III-B, so we start with **Step 2** and break every step into two substeps for clarity.

**Step 2.1**: By deriving (8d) from (2a), (2d) and (1b), and deriving (8e) from (2b), (2d) and (1c), we relax $\mathbf{P_1}$ to $\mathbf{P_3}$:

$$\begin{aligned}
\min\quad & \sum_t\sum_i\sum_{j\in\mathcal{J}_i} a_{it}x_{ijt} + \sum_t\sum_i b_i v_{it} \\
& + \sum_t\sum_j\sum_{i\in\mathcal{I}_j} c_{ijt}y_{ijt} + \sum_t\sum_j\sum_{i\in\mathcal{I}_j} d_{ij}w_{ijt} \\
\text{s.t.}\quad & v_{it} \geqslant \sum_{j\in\mathcal{J}_i} x_{ijt} - \sum_{j\in\mathcal{J}_i} x_{ijt-1}, \forall i, \forall t, && \text{(8a)}\\
& w_{ijt} \geqslant y_{ijt} - y_{ijt-1}, \forall i\in\mathcal{I}_j, \forall j, \forall t, && \text{(8b)}\\
& v_{it} \geqslant 0, w_{ijt} \geqslant 0, \forall i, \forall j, \forall t, && \text{(8c)}\\
& \sum_{\substack{k\in\mathcal{I}\\k\neq i}}\sum_{j\in\mathcal{J}_i} x_{kjt} \geqslant \left[\sum_j \lambda_{jt} - C_i\right]^+, \forall i, \forall t, && \text{(8d)}\\
& \sum_{\substack{k\in\mathcal{I}_j\\k\neq i}} y_{kjt} \geqslant \left[\lambda_{jt} - B_{ij}\right]^+, \forall i\in\mathcal{I}_j, \forall j, \forall t, && \text{(8e)}\\
& \text{(2a), (2b), (2d), (2e),}
\end{aligned}$$

where $v_{it}$ and $w_{ijt}$ are auxiliary variables. Note $x_{ijt} \geqslant 0$, $y_{ijt} \geqslant 0$ due to (2a), (2b), (2e), and thus we can apply $[\cdot]^+$ to the right-hand sides of (8d) and (8e).

**Step 2.2**: We derive the Lagrange dual problem of $\mathbf{P_3}$. Let $\alpha_{it}$, $\beta_{ijt}$, $\delta_{it}$, $\theta_{ijt}$ be the dual variables associated with (8a), (8b), (8d) and (8e), respectively; let $\rho_{ijt}$, $\phi_{ijt}$, $\gamma_{jt}$ be the dual variables associated with (2a), (2b) and (2d), respectively. We have the dual problem $\mathbf{P_4}$:

$$\begin{aligned}
\max\quad & D = \sum_t\sum_j \lambda_{jt}\gamma_{jt} + \sum_t\sum_i\left[\sum_j\lambda_{jt}-C_i\right]^+\delta_{it} \\
& + \sum_t\sum_j\sum_{i\in\mathcal{I}_j}\left[\lambda_{jt}-B_{ij}\right]^+\theta_{ijt} && \text{(9)}\\
\text{s.t.}\quad & a_{it}+\alpha_{it}-\alpha_{it+1}-\rho_{ijt}+\sum_{\substack{k\in\mathcal{I}\\k\neq i}}\delta_{kt} \geqslant 0, \\
& \qquad \forall i\in\mathcal{I}_j, \forall j, \forall t, && \text{(9a)}\\
& c_{ijt}+\beta_{ijt}-\beta_{ijt+1}-\phi_{ijt}+\sum_{\substack{k\in\mathcal{I}_j\\k\neq i}}\theta_{kjt} \geqslant 0, \\
& \qquad \forall i\in\mathcal{I}_j, \forall j, \forall t, && \text{(9b)}\\
& \rho_{ijt}+\phi_{ijt}-\gamma_{jt} \geqslant 0, \forall i\in\mathcal{I}_j, \forall j, \forall t, && \text{(9c)}\\
& b_i-\alpha_{it} \geqslant 0, \forall i, \forall t, && \text{(9d)}\\
& d_j-\beta_{jt} \geqslant 0, \forall j, \forall t, && \text{(9e)}\\
& \alpha_{it}\geqslant 0, \delta_{it}\geqslant 0, \gamma_{jt}\geqslant 0, \forall i, \forall j, \forall t, \\
& \beta_{ijt}\geqslant 0, \theta_{ijt}\geqslant 0, \rho_{ijt}\geqslant 0, \phi_{ijt}\geqslant 0, \\
& \qquad \forall i\in\mathcal{I}_j, \forall j, \forall t. && \text{(9f)}
\end{aligned}$$

**Step 3.1**: We write the following KKT conditions that characterize the optimal solution $x_{ijt}^*$, $y_{ijt}^*$ of $\mathbf{P_2^{(t)}}$, where $\rho'_{ijt}$, $\phi'_{ijt}$, $\gamma'_{jt}$ are the dual variables associated with (4a), (4b), (4c), respectively, $\delta'_{it}$, $\theta'_{ijt}$ are the dual variables associated with (4d), (4e), respectively, and $p_{ijt}$ is the dual variable for (4f).

We will use these equations and inequalities later:

$$a_{it} + \frac{b_i}{\eta_i} \ln \frac{\sum_{j \in \mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \rho'_{ijt} + \sum_{\substack{k \in \mathcal{I} \\ k \neq i}} \delta'_{kt} = 0\,, \quad \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10a}$$

$$c_{ijt} + \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} - \phi'_{ijt} + \sum_{\substack{k \in \mathcal{I}_j \\ k \neq i}} \theta'_{kjt} = 0\,, \quad \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10b}$$

$$\rho'_{ijt} + \phi'_{ijt} - \gamma'_{jt} - p_{ijt} = 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10c}$$

$$\rho'_{ijt}(s^*_{ijt} - x^*_{ijt}) = 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10d}$$

$$\phi'_{ijt}(s^*_{ijt} - y^*_{ijt}) = 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10e}$$

$$\gamma'_{jt}\left(\lambda_{jt} - \sum_{i \in \mathcal{I}_j} s^*_{ijt}\right) = 0\,, \forall j\,, \tag{10f}$$

$$p_{ijt} s^*_{ijt} = 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,, \tag{10g}$$

$$\rho'_{ijt} \geqslant 0\,, \phi'_{ijt} \geqslant 0\,, \theta'_{ijt} \geqslant 0\,, p_{ijt} \geqslant 0\,, \forall i \in \mathcal{I}_j\,, \forall j\,;$$
$$\gamma'_{jt} \geqslant 0\,, \delta'_{it} \geqslant 0\,, \forall j\,, \forall i\,. \tag{10h}$$

**Step 3.2**: We map $x^*_{ijt}$, $y^*_{ijt}$ and the dual variables in the KKT conditions to a solution that is feasible for $\mathbf{P_4}$ at $t$:

$$\alpha_{it} = \frac{b_i}{\eta_i} \ln \frac{C_i + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon}\,, \beta_{ijt} = \frac{d_{ij}}{\eta'_{ij}} \ln \frac{B_{ij} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'}\,,$$
$$\rho_{ijt} = \rho'_{ijt}\,, \phi_{ijt} = \phi'_{ijt}\,, \gamma_{jt} = \gamma'_{jt}\,, \delta_{it} = \delta'_{it}\,, \theta_{ijt} = \theta'_{ijt}\,.$$

To see the feasibility, let us take the constraint (9a) as an example. Putting them into the left-hand side of (9a), we get

$$\begin{aligned} & a_{it} + \alpha_{it} - \alpha_{it+1} - \rho_{ijt} + \sum_{\substack{k \in \mathcal{I} \\ k \neq i}} \delta_{kt} \\ &= a_{it} + \frac{b_i}{\eta_i} \ln \frac{C_i + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \frac{b_i}{\eta_i} \ln \frac{C_i + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt} + \varepsilon} - \rho'_{ijt} \\ &+ \sum_{\substack{k \in \mathcal{I} \\ k \neq i}} \delta'_{kt} \geqslant 0\,. \end{aligned}$$

The above holds due to (10a) and (10h). Analogously, (9b) holds due to (10b) and (10h); (9c) holds due to (10c) and (10h); (9d), (9e) hold due to $x^*_{ijt} \geqslant 0$, $y^*_{ijt} \geqslant 0$, as in (4a), (4b), (4f). In (9f), $\alpha_{it} \geqslant 0$, $\beta_{ijt} \geqslant 0$ hold due to $\sum_{j \in \mathcal{J}_i} x^*_{ijt} \leqslant C_i$, $y^*_{ijt} \leqslant B_{ij}$, $\forall t$, as in Lemma 1; the others hold due to (10h).

**Step 4**: In this step, we demonstrate that, using the sequence of $\{x^*_1, y^*_1, x^*_2, y^*_2, \ldots, x^*_t, y^*_t, \ldots, x^*_T, y^*_T\}$ as the solution to $\mathbf{P_1}$, its objective function value is bounded by a constant (*i.e.*, the competitive ratio) times the objective function value of $\mathbf{P_4}$ evaluated with the constructed solutions $\alpha_{it}$, $\beta_{ijt}$, $\rho_{ijt}$, $\phi_{ijt}$, $\gamma_{jt}$, $\delta_{it}$, $\theta_{ijt}$, $\forall t$. To this end, we bound the allocation cost and the reconfiguration cost in $\mathbf{P_1}$'s objective, respectively.

**Step 4.1**: Firstly, we bound the allocation cost.

$$\sum_t \sum_j \sum_{i \in \mathcal{I}_j} a_{it} x^*_{ijt} + \sum_t \sum_j \sum_{i \in \mathcal{I}_j} c_{ijt} y^*_{ijt} \tag{13}$$

$$\begin{aligned} &= \sum_t \sum_j \sum_{i \in \mathcal{I}_j} x^*_{ijt} \left( \rho_{ijt} - \frac{b_i}{\eta_i} \ln \frac{\sum_{j \in \mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \sum_{\substack{k \in \mathcal{I} \\ k \neq i}} \delta_{kt} \right) \\ &+ \sum_t \sum_j \sum_{i \in \mathcal{I}_j} y^*_{ijt} \left( \phi_{ijt} - \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} - \sum_{\substack{k \in \mathcal{I}_j \\ k \neq i}} \theta_{kjt} \right) \end{aligned} \tag{13a}$$

$$\begin{aligned} &\leqslant \sum_t \sum_j \sum_{i \in \mathcal{I}_j} x^*_{ijt} \rho_{ijt} + \sum_t \sum_j \sum_{i \in \mathcal{I}_j} y^*_{ijt} \phi_{ijt} \\ &- \sum_t \sum_j \sum_{i \in \mathcal{I}_j} x^*_{ijt} \frac{b_i}{\eta_i} \ln \frac{\sum_{j \in \mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \\ &- \sum_t \sum_j \sum_{i \in \mathcal{I}_j} y^*_{ijt} \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \end{aligned} \tag{13b}$$

$$\leqslant \sum_t \sum_j \sum_{i \in \mathcal{I}_j} s^*_{ijt} (\rho_{ijt} + \phi_{ijt}) \tag{13c}$$

$$= \sum_t \sum_j \sum_{i \in \mathcal{I}_j} s^*_{ijt} \gamma_{jt} \tag{13d}$$

$$= \sum_t \sum_j \lambda_{jt} \gamma_{jt} \tag{13e}$$

$$\leqslant D \tag{13f}$$

(13a) follows from (10a) and (10b). (13b) follows from (10h). (13c) follows from (10d), (10e) and these two inequalities: $\sum_t \sum_j \sum_{i \in \mathcal{I}_j} x^*_{ijt} \frac{b_i}{\eta_i} \ln \frac{\sum_{j \in \mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j \in \mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \geqslant 0$ and $\sum_t \sum_j \sum_{i \in \mathcal{I}_j} y^*_{ijt} \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \geqslant 0$. (13d) follows from (10c) and (10g). (13e) follows from (10f). (13f) follows from (9). As an example, in the following we show that the latter of the above two inequalities holds, and the former can be shown analogously. Note that proving the latter inequality is equivalent to proving that the sum of (14a) and (14e) is no less than zero:

$$\sum_t (y^*_{ijt} + \varepsilon') \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \tag{14a}$$

$$\geqslant \left( \sum_t (y^*_{ijt} + \varepsilon') \right) \ln \frac{\sum_t (y^*_{ijt} + \varepsilon')}{\sum_t (y^*_{ijt-1} + \varepsilon')} \tag{14b}$$

$$\geqslant \sum_t (y^*_{ijt} + \varepsilon') - \sum_t (y^*_{ijt-1} + \varepsilon') \tag{14c}$$

$$= y^*_{ijT} - y^*_{ij0} \tag{14d}$$

$$- \sum_t \varepsilon' \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \tag{14e}$$

$$= (y^*_{ij0} + \varepsilon') \ln \frac{y^*_{ij0} + \varepsilon'}{y^*_{ijT} + \varepsilon'} \tag{14f}$$

$$\geqslant y^*_{ij0} - y^*_{ijT} \tag{14g}$$

(14b) follows from (15b) as below. (14c) and (14g) follow from (15a) as below. (14f) follows due to $y^*_{ij0} = 0$. (15a) and

(15b) are two facts that we exploit.

$$m - n \leqslant m \ln \frac{m}{n}\,, \forall m, n > 0\,, \tag{15a}$$

$$(\sum\nolimits_i m_i) \ln \frac{\sum_i m_i}{\sum_i n_i} \leqslant \sum\nolimits_i m_i \ln \frac{m_i}{n_i}\,, \forall m, n > 0\,. \tag{15b}$$

**Step 4.2**: Afterwards, we bound the reconfiguration cost. We have the following two definitions for the index sets, $\forall t \geqslant 1$:

$$\mathcal{I}_t^+ \triangleq \{i | \sum\nolimits_{j\in\mathcal{J}_i} x^*_{ijt} > \sum\nolimits_{j\in\mathcal{J}_i} x^*_{ijt-1}, \forall i \in \mathcal{I}\}, \tag{16a}$$

$$\{\mathcal{I}_j \times \mathcal{J}\}_t^+ \triangleq \{(i,j) | y^*_{ijt} > y^*_{ijt-1}, \forall i \in \mathcal{I}_j, \forall j \in \mathcal{J}\}. \tag{16b}$$

We bound the first part of the reconfiguration cost:

$$\sum_t \sum_{i\in\mathcal{I}} b_i \left[\sum_{j\in\mathcal{J}_i} x^*_{ijt} - \sum_{j\in\mathcal{J}_i} x^*_{ijt-1}\right]^+ \tag{17}$$

$$= \sum_t \sum_{i\in\mathcal{I}_t^+} b_i \left(\sum_{j\in\mathcal{J}_i} x^*_{ijt} - \sum_{j\in\mathcal{J}_i} x^*_{ijt-1}\right) \tag{17a}$$

$$\leqslant \sum_t \sum_{i\in\mathcal{I}_t^+} b_i \left(\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon\right) \ln \frac{\sum\limits_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum\limits_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \tag{17b}$$

$$\leqslant \max_i\{(C_i+\varepsilon)\eta_i\} \sum_t \sum_{i\in\mathcal{I}_t^+} \frac{b_i}{\eta_i} \ln \frac{\sum\limits_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum\limits_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \tag{17c}$$

$$\leqslant C(\varepsilon) \sum_t \sum_{i\in\mathcal{I}_t^+} \rho_{ijt}\Big|_{j\in\{j|x^*_{ijt}>0, j\in\mathcal{J}_i\}} \tag{17d}$$

$$= C(\varepsilon) \sum_t \sum_{\substack{i\in\mathcal{I}_t^+ \\ \rho_{ijt}>0}} (\gamma_{jt} + p_{ijt} - \phi_{ijt})\Big|_{j\in\{j|x^*_{ijt}>0, j\in\mathcal{J}_i\}} \tag{17e}$$

$$\leqslant C(\varepsilon) \sum_t \sum_{\substack{i\in\mathcal{I}_t^+ \\ \rho_{ijt}>0}} \gamma_{jt}\Big|_{j\in\{j|x^*_{ijt}>0, j\in\mathcal{J}_i\}} \tag{17f}$$

$$\leqslant C(\varepsilon)|\mathcal{I}|D \tag{17g}$$

(17a) follows from (16a). (17b) follows from (15a). (17c) follows, due to $\sum_{j\in\mathcal{J}_i} x^*_{ijt} \leqslant C_i$. (17d) follows from (10a). Note that in (17d), for any given $i \in \mathcal{I}_t^+$, we can choose to use any $\rho_{ijt}$, $j \in \mathcal{J}_i$; however, we choose the particular $\rho_{ijt}$ that has the corresponding $x^*_{ijt} > 0$. Such a $j$ always exists, because $i \in \mathcal{I}_t^+$ indicates $\sum_{j\in\mathcal{J}_i} x^*_{ijt} > \sum_{j\in\mathcal{J}_i} x^*_{ijt-1} \geqslant 0$ and thus there exists at least one $j \in \mathcal{J}_i$ such that $x^*_{ijt} > 0$ holds. We continue to (17e) only for those $i$ where $\rho_{ijt} > 0$; if $\rho_{ijt} = 0$, $\forall i \in \mathcal{I}_t^+$, we can directly reach (17g) from (17d). (17e) follows from (10c). (17f) follows, because of $p_{ijt} = 0$. Applying $x^*_{ijt} > 0$, $\rho_{ijt} > 0$ to (10d), we have $s^*_{ijt} > 0$; applying $s^*_{ijt} > 0$ to (10g), we have $p_{ijt} = 0$. (17g) follows, because of (9), $\gamma_{jt} > 0$ and $\lambda_{jt} \geqslant 1$. $\gamma_{jt} > 0$ is due to (10c), $\rho_{ijt} > 0$ and $p_{ijt} = 0$. $\lambda_{jt} > 0$ is due to (10f), $\gamma_{jt} > 0$ and $s^*_{ijt} > 0$; $\lambda_{jt} \geqslant 1$ holds becasue $\lambda_{jt}$ is an integer.

We bound the second part of the reconfiguration cost:

$$\sum_t \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}_j} d_{ij}[y^*_{ijt} - y^*_{ijt-1}]^+ \tag{18}$$

$$= \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} d_{ij}(y^*_{ijt} - y^*_{ijt-1}) \tag{18a}$$

$$\leqslant \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} d_{ij}(y^*_{ijt} + \varepsilon') \ln \frac{y^*_{ijt} + \varepsilon}{y^*_{ijt-1} + \varepsilon'} \tag{18b}$$

$$\leqslant \max_{i,j}\{(B_{ij}+\varepsilon')\eta'_{ij}\} \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon}{y^*_{ijt-1} + \varepsilon'} \tag{18c}$$

$$\leqslant B(\varepsilon') \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} \phi_{ijt} \tag{18d}$$

$$= B(\varepsilon') \sum_t \sum_{\substack{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+ \\ \phi_{ijt}>0}} (\gamma_{jt} + p_{ijt} - \rho_{ijt}) \tag{18e}$$

$$\leqslant B(\varepsilon') \sum_t \sum_{\substack{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+ \\ \phi_{ijt}>0}} \gamma_{jt} \tag{18f}$$

$$\leqslant B(\varepsilon')|\mathcal{I}|D \tag{18g}$$

(18a) follows from (16b). (18b) follows from (15a). (18c) follows, due to $y^*_{ijt} \leqslant B_{ij}$. (18d) follows from (10b). We continue to (18e) only for those $(i,j)$ such that $\phi_{ijt} > 0$; if $\phi_{ijt} = 0$, $\forall (i,j) \in \{\mathcal{I}_j \times \mathcal{J}\}_t^+$, we can directly reach (18g) from (18d). (18e) follows from (10c). (18f) follows, because of $p_{ijt} = 0$. Applying $y^*_{ijt} > y^*_{ijt-1} \geqslant 0$, $\phi_{ijt} > 0$ to (10e), we have $s^*_{ijt} > 0$; applying $s^*_{ijt} > 0$ to (10g), we have $p_{ijt} = 0$. Finally, (18g) is analogous to (17g).

## IV. Generalization

Our models, online algorithm, and competitive analysis can be generalized to arbitrary $N \geqslant 2$ tiers of clouds. We can, in fact, bound the allocation cost and the reconfiguration cost in the $N$-tier problem, and thus prove the following theorem for the competitive ratio, where $\mathcal{I}_n$ denotes the set of clouds at the $n$-th tier, $C_i$ is the capacity of cloud $i$, and $B_{ij}$ is the capacity of the network between clouds $i$ and $j$:

**Theorem 2.** *For arbitrary $N \geqslant 2$ tiers of clouds, the competitive ratio is $r = 1 + \sum_{n=1}^{N} |\mathcal{I}_n| \left(C_n(\varepsilon_n) + B_{n,n-1}(\varepsilon'_n)\right)$, where $C_n(\varepsilon_n) = \max\limits_{i\in\mathcal{I}_n} \left\{(C_i+\varepsilon_n)\ln\left(1+\frac{C_i}{\varepsilon_n}\right)\right\}$, $\forall n \geqslant 1$; $B_{n,n-1}(\varepsilon'_n) = \max\limits_{i\in\mathcal{I}_n, j\in\mathcal{I}_{n-1}} \left\{(B_{ij}+\varepsilon'_n)\ln\left(1+\frac{B_{ij}}{\varepsilon'_n}\right)\right\}$, $\forall n \geqslant 2$; $B_{1,0}(\varepsilon'_1) = 0$.*

A sketch of the proof is that, in a $N$-tier problem, we have additional constraints and thus more KKT conditions that we can exploit for deduction. For every two consecutive tiers, there exists a particular KKT equation that involves the dual variables of both tiers, enabling us to reformulate an upper-tier problem in terms of a lower-tier problem. The details of the proof are not presented here due to the page limit.

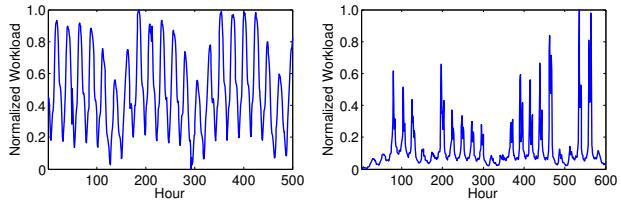
## V. Numerical Evaluation

We evaluate our online algorithm using real-world data traces. The evaluation has two purposes. Firstly, having proved the worst-case guarantee, we investigate the performance of our online algorithm in a realistic case and compare it with other approaches. Secondly, to understand the difference between the results, we characterize how the resources are actually allocated and reconfigured over time by different approaches.

### A. Inputs

***Clouds*** $\mathcal{I}$, $\mathcal{J}$ ***and SLA*** $\mathcal{I}_j$, $\mathcal{J}_i$: We use the 18 AT&T North American data center locations [2] as the locations of tier-2 clouds, and use the locations of the capital cities of the 48 continental US states as the locations of tier-1 clouds. Having the location of each cloud, we use the geographic distance to define SLA [9], [15]: for a tier-1 cloud, we assume that the $k$ tier-2 clouds that are geographically closest to this tier-1 cloud can satisfy the SLA requirement. For different tier-1 clouds, these $k$ closest tier-2 clouds can be different.

***Workload*** $\lambda_{jt}$: We use the workload of Wikipedia in October 2007 [18] and the workload of the HTTP servers from April to July 1998 during the World Cup'98 period [3]. The former has more regular dynamics and the latter is more bursty, as shown in Fig. 3a and 3b, respectively. While the original workload files record the URL requests at a second granularity, we aggregate the number of requests by hour and treat one hour as one time slot in our evaluations. There are 500 hours for Wikipedia. There are 2089 hours for the original World Cup workload, however, in our evaluations we only adopt the most bursty 600 hours, starting at the 901st hour and ending at the 1500th hour. We replicate the workload across all tier-1 clouds to simulate the workload of each cloud.




**(a)** Wikipedia **(b)** World Cup

**Fig. 3:** The time-varying workload

***Operating Price*** $a_{it}$, $c_{ijt}$: We use energy and WAN bandwidth prices respectively. Energy and bandwidth are among the most significant operating expense for data centers. In the wholesale electricity markets in US, prices vary temporally and spatially. The hourly real-time electricity prices of different states, administered by different RTOs (Regional Transmission Organizations), follow Gaussian distributions with different means and standard deviations [15]. In our case, across all 18 tier-2 cloud locations, for those where there is an hourly real-time electricity market, we synthesize the dynamic price for each hour following the Gaussian distribution with the mean and the standard deviation of the corresponding market, as shown in Table I; for those without an hourly real-time electricity market, we assume the price is fixed and equals the mean price of its geographically closest real-time market [16].

Cloud WAN bandwidth price is estimated based on network capacity [14], [24]. We estimate the price of a given network capacity by the tiered pricing scheme of Amazon EC2 [1], summarized as Table II. Bandwidth price does not vary much with time in a short term, and is thus considered a constant.

**TABLE I:** Electricity price statistics [15]

| Location | State | RTO | Mean ($/MWh) | StDev ($/MWh) |
|---|---|---|---|---|
| Annapolis | MD | PJM | 40.6 | 26.9 |
| Chicago | IL | | | |
| Washington DC | DC | | | |
| San Francisco | CA | CAISO | 54.0 | 34.2 |
| San Jose | | | | |
| Albany | NY | NYISO | 77.9 | 40.3 |
| New York City | | | | |
| Boston | MA | ISONE | 66.5 | 25.8 |

**TABLE II:** Bandwidth price [1]

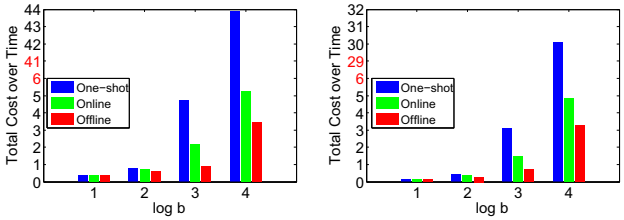
| Network Capacity (TB/month) | Price ($/GB) |
|---|---|
| $\leqslant$ 10 | 0.09 |
| 10 – 50 | 0.085 |
| 50 – 150 | 0.07 |
| 150 – 500 | 0.05 |

***Cloud and Network Capacities*** $C_i$, $B_{ij}$: Cloud capacity and network capacity are estimated based on workload [12], [14]. We assume the cloud capacity is provisioned so that the peak workload consumes 80% of it. If every tier-1 cloud uses its closest tier-2 cloud to satisfy the SLA, then the capacity of a tier-2 cloud is set to $1.25$ times its peak workload which is the sum of the peak workloads of those tier-1 clouds that use this tier-2 cloud as their closest cloud; if every tier-1 cloud uses its $k$ closest tier-2 clouds to satisfy the SLA, then we evenly split the peak workload of every tier-1 cloud across its tier-2 clouds, and thus the capacity of a tier-2 cloud is set by the same approach as above while replacing $1.25$ with $1.25/k$. We set the capacity of the network between a tier-1 and a tier-2 clouds to the capacity of the incident tier-2 cloud.

***Algorithms for Comparison***: We compare our online algorithm, which solves $\mathbf{P_2^{(t)}}$ at every time slot, the sequence of one-shot optimizations, which solves the one-shot slice of $\mathbf{P_1}$ at every time slot, and the offline optimum, which solves $\mathbf{P_1}$ assuming the workload of the entire future is known in prior. We use AMPL [6] for formulations and invoke IPOPT [21], the interior point method, for the three algorithms.

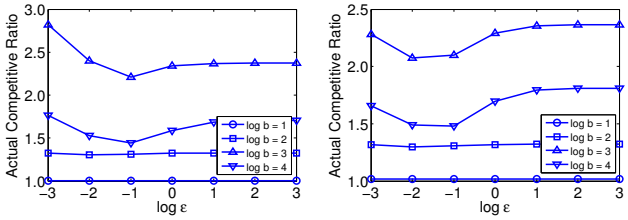
### B. Control Knobs

***Reconfiguration Price*** $b_i$, $d_{ij}$: We vary $b_i$, $d_{ij}$ to reveal a spectrum of how different reconfiguration prices may influence the results. Instead of estimating an absolute value of the reconfiguration price, we use a relative weight over the operating price. For instance, a weight of $10$ means the absolute reconfiguration price is an order of magnitude larger than the absolute operating price in value. In our evaluations, we always set $b_i = d_{ij}, \forall i, j$. We denote this value simply as $\mathbf{b}$ in our figures and vary it as $10$, $10^2$, $10^3$ and $10^4$, respectively.



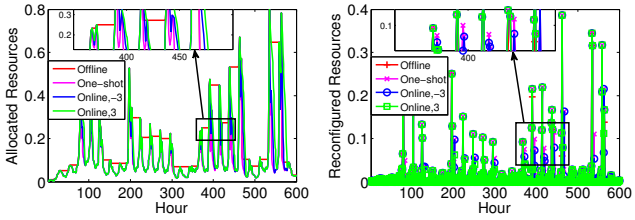

**(a)** Wikipedia **(b)** World Cup

**Fig. 4:** Total cost comparison for different reconfiguration prices



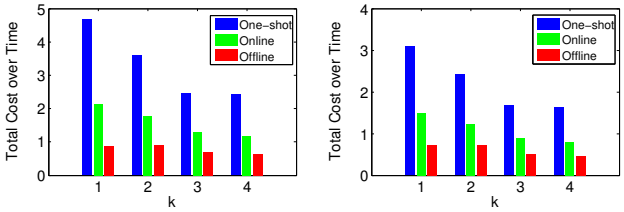

**(a)** Wikipedia **(b)** World Cup

**Fig. 5:** Actual competitive ratios




**(a)** Allocation **(b)** Reconfiguration

**Fig. 6:** Dynamic resource allocation and reconfiguration




**(a)** Wikipedia **(b)** World Cup

**Fig. 7:** Total cost comparison for different SLAs

***Other Parameters*** $\varepsilon$, $\varepsilon'$, $k$: We set $\varepsilon = \varepsilon'$, where $\varepsilon$, $\varepsilon' > 0$ are parameters of our online algorithm, and vary $\varepsilon$ from $10^{-3}$ to $10^3$ in a logarithmic scale so that we see how it may affect the results and how to tune its value to achieve the largest benefit for a workload. We also vary $k$ as 1, 2, 3, 4, meaning that the number of the closest clouds chosen by every tier-1 cloud varies, and see how this variation may affect the results.

### C. Results

Fig. 4 demonstrates the normalized total cost over time when the cloud and the network resources are allocated and reconfigured by one-shot optimizations, our online algorithm, and the offline optimal approach for the Wikipedia workload and the World Cup workload, respectively. In this figure we set $\varepsilon = 10^{-2}$, $k = 1$ and vary the reconfiguration price. It is natural that if the reconfiguration price is low one-shot optimizations perform quite close to the offline optimum. For a low reconfiguration price, our online algorithm preserves the same performance as one-shot optimizations. As the reconfiguration price increases, one-shot optimizations, which essentially neglect the reconfiguration cost, have much larger total cost than the offline optimum, while our online algorithm has the total cost just moderately larger than the offline optimum. Note the jumps (marked red) in the vertical axes that show the comparison on the lower end of the scale and also capture the larger values. This figure indicates that our algorithm behaves consistently well for the two workloads.

Fig. 5 visualizes how the "actual" competitive ratio, *i.e.*, the ratio of the total cost incurred by our online algorithm over what is incurred by the offline optimal solution in the realistic case, varies along with the algorithmic parameter $\varepsilon$ for the two workloads. In this figure we set $k = 1$. Firstly and overall, this ratio is reasonably good for both workloads as it is always below 3. Secondly, this ratio does not always increase with the reconfiguration price, *e.g.*, the reconfiguration price of $10^4$ has smaller ratios than $10^3$. This is because the offline optimum in the former case is larger than in the latter (*cf.* Fig. 4). Thirdly, the curve of the actual competitive ratio has a valley. Note that our worst-case theoretical competitive ratio always decreases as $\varepsilon$ grows, but this figure implies that in practice a lower $\varepsilon$ may achieve a lower actual competitive ratio.

Fig. 6 depicts the normalized amount of resources allocated and reconfigured over time at one of the tier-2 clouds for the World Cup workload. We compare the offline approach, one-shot optimizations, and our online algorithm with $\varepsilon = 10^{-3}$ and $\varepsilon = 10^3$, respectively. In this figure we set $k = 1$, and the reconfiguration price is $10^2$. First, the offline optimum tends to stay (*cf.* Fig. 6a) for workload valleys so that the corresponding reconfiguration cost is zero (*cf.* Fig. 6b) during these time slots. Second, one-shot and online approaches are quite close in total cost (*cf.* Fig. 4b), but the cost breakdown is different: the former allocates resources no more than the latter (*cf.* Fig. 6a) at every time slot but reconfigures resources no fewer than the latter (*cf.* Fig. 6b) at every time slot. Last, this figure shows that a large $\varepsilon$ allocates resources no fewer than a small $\varepsilon$ (*cf.* Fig. 6a), but the corresponding reconfiguration is no more than the small $\varepsilon$ case (*cf.* Fig. 6b). The allocation and the reconfiguration compensate for each other and the total cost is approximately the same, about $1.3\times$ the offline optimum (*cf.* Fig. 5b) for the reconfiguration price of $10^2$.

Fig. 7 verifies the performance of our online algorithm for different SLAs. In this figure we set $\varepsilon = 10^{-2}$, and the reconfiguration price is $10^3$. When every tier-1 cloud uses more tier-2 clouds to satisfy the SLA, there is also more room for optimization, both online and offline. The trend is that the total cost achieved by our online algorithm gets closer to the offline optimum as the SLA involves more tier-2 clouds.

## VI. Related Work

***Reconfiguration-oblivious Resource Allocation***: Hao *et al.* [7] designed an online optimization algorithm to allocate VMs at distributed clouds for revenue maximization while satisfying

the dynamic demands for VMs and a diversity of resource constraints. Hu *et al.* [8] made online decisions of buying cloud contracts of different prices, resource rates, and durations to accommodate the unpredictably varying demand, based on a multi-dimensional version of a classic parking permit problem. Liu *et al.* [12] optimized the energy cost and the end-to-end user delay over time with consideration of energy price and network delay diversity by allocating capacities across data centers via distributed algorithms. Zhang *et al.* [25] treated the cloud provider as the auctioneer who leased resources and users as bidders who bade for VMs of different types, and designed an online, randomized combinatorial auction to maximize the economical efficiency upon bid arrivals.

***Reconfiguration-aware Resource Allocation***: Lin *et al.* [10], [11] might be among the first few to study the dynamic resource allocation in the cloud context with smoothing the reconfiguration cost as part of the objective, and proposed the "lazy" capacity provisioning for the single cloud case [11] and the averaged fixed horizon control for the multi-cloud case [10]. Zhang *et al.* [27] investigated a similar problem in the geo-distributed scenario where server number changes incurred the reconfiguration cost and applied model predictive control to reduce system dynamics. Zhang *et al.* [26] developed the randomized fixed horizon control to route big data from sources to selected data centers for aggregation and processing and Wu *et al.* [23] exploited Lyapunov optimization to distribute social media to clouds to satisfy time-varying demands, where in both cases the reconfiguration cost was caused by data movement across locations over time.

Our work differs from both categories of existing work. Firstly, the previous dynamic optimization research does not capture the joint, smoothed resource allocation in a multi-tier cloud and network infrastructure. Besides, the first category of work has no reconfiguration cost considered, *i.e.*, switching from one decision to another across time slots is free. With the reconfiguration cost, it is unclear whether it is possible and how to adapt such reconfiguration-oblivious approaches, which also motivates this paper. The second category of work often assumes the lookahead into future time slots and their results largely depend on the future information, while our work does not assume any lookahead beyond the current time slot and is different in that it is derived via the primal-dual approach based on regularization.

## VII. Conclusion

The problem of jointly allocating and reconfiguring cloud and network resources in an online setting is increasingly important as the cloud computing paradigm shifts to a multi-tier hierarchic structure. In this paper, we take a regularization-based method to design an online algorithm. We overcome the major challenge stemming from reconfiguration-induced, coupled decisions by constructing a series of subproblems, each of which is solvable at the corresponding time slot. We formally prove that this algorithm can produce a solution with a parameterized competitive ratio for any arbitrary workload and operating price. Evaluations based on real-world data also confirm that our algorithm performs well in practice.

## References


[1] "Amazon EC2 Pricing," http://aws.amazon.com/ec2/pricing/.

[2] "AT&T's 38 Global Internet Data Centers," http://www.business.att.com/content/productbrochures/eb_idcmap.pdf.

[3] M. Arlitt and T. Jin, "1998 World Cup Web Site Access Logs," http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[4] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive Analysis via Regularization," in *ACM-SIAM SODA*, 2014.

[5] B. Chandramouli, J. Claessens, S. Nath, I. Santos, and W. Zhou, "RACE: Real-Time Applications over Cloud-Edge," in *ACM SIGMOD*, 2012.

[6] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming (2nd Edition)*. Duxbury Press, 2002.

[7] F. Hao, M. Kodialam, T. Lakshman, and S. Mukherjee, "Online Allocation of Virtual Machines in a Distributed Cloud," in *IEEE INFOCOM*, 2014.

[8] X. Hu, A. Ludwig, A. Richa, and S. Schmid, "Competitive Strategies for Online Cloud Resource Allocation with Discounts," in *IEEE ICDCS*, 2015.

[9] L. Jiao, J. Li, W. Du, and X. Fu, "Multi-Objective Data Placement for Multi-Cloud Socially Aware Services," in *IEEE INFOCOM*, 2014.

[10] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online Algorithms for Geographical Load Balancing," in *IEEE IGCC*, 2012.

[11] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic Right-Sizing for Power-Proportional Data Centers," in *IEEE INFOCOM*, 2011.

[12] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening Geographical Load Balancing," in *ACM SIGMETRICS*, 2011.

[13] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud," in *IEEE CLOUD*, 2012.

[14] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, "To Coordinate or Not to Coordinate? Wide-Area Traffic Management for Data Centers," *Tech. Rept., Princeton University*, 2012.

[15] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the Electric Bill for Internet-Scale Systems," in *ACM SIGCOMM*, 2009.

[16] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment," in *IEEE INFOCOM*, 2010.

[17] A. Shraer, B. Reed, D. Malkhi, and F. P. Junqueira, "Dynamic Reconfiguration of Primary/Backup Clusters," in *USENIX ATC*, 2012.

[18] G. Urdaneta, G. Pierre, and M. van Steen, "Wikipedia Workload Analysis for Decentralized Hosting," *Elsevier Computer Networks*, vol. 53, no. 11, pp. 1830–1845, 2009.

[19] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic Service Migration and Workload Scheduling in Edge-Clouds," *Elsevier Performance Evaluation*, vol. 91, pp. 205–228, 2015.

[20] L. M. Vaquero and L. Rodero-Merino, "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM CCR*, vol. 44, no. 5, pp. 27–32, 2014.

[21] A. Wächter and L. T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[22] M. Weldon, *The Future X Network: A Bell Labs Perspective*. CRC Press, 2015.

[23] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, "Scaling Social Media Applications into Geo-Distributed Clouds," in *IEEE INFOCOM*, 2012.

[24] H. Xu and B. Li, "Joint Request Mapping and Response Routing for Geo-Distributed Cloud Services," in *IEEE INFOCOM*, 2013.

[25] L. Zhang, Z. Li, and C. Wu, "Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach," in *IEEE INFOCOM*, 2014.

[26] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. Lau, "Moving Big Data to the Cloud: An Online Cost-Minimizing Approach," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2710–2721, 2013.

[27] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic Service Placement in Geographically Distributed Clouds," in *IEEE ICDCS*, 2012.