Smoothed Online Resource Allocation in Multi-Tier Distributed Cloud Networks

Lei Jiao, *Member, IEEE*, Antonia Maria Tulino, *Fellow, IEEE*, Jaime Llorca, Yue Jin, and Alessandra Sala, *Member, IEEE*

***Abstract*— The problem of dynamic resource allocation for service provisioning in multi-tier distributed clouds is particularly challenging due to the coexistence of several factors: the need for joint allocation of cloud and network resources, the need for online decision-making under time-varying service demands and resource prices, and the reconfiguration cost associated with changing resource allocation decisions. We study this problem from an online optimization perspective to address all these challenges. We design an online algorithm that decouples the original offline problem over time by constructing a series of regularized subproblems, solvable at each corresponding time slot using the output of the previous time slot. We prove that, without prediction beyond the current time slot, our algorithm achieves a parameterized competitive ratio for arbitrarily dynamic workloads and resource prices. If prediction is available, we demonstrate that existing prediction-based control algorithms lack worst case performance guarantees for our problem, and we design two novel predictive control algorithms that inherit the theoretical guarantees of our online algorithm, while exhibiting improved practical performance. We conduct evaluations in a variety of settings based on real-world dynamic inputs and show that, without prediction, our online algorithm achieves up to nine times total cost reduction compared with the sequence of greedy one-shot optimizations and at most three times the offline optimum; with moderate predictions, our control algorithms can achieve two times total cost reduction compared with existing prediction-based algorithms.**

***Index Terms*— Cloud networks, resource allocation, resource reconfiguration, online optimization, regularization.**

I. Introduction

CLOUD resources are moving closer toward end users [6], [22], [23], enabling 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 and processing). Small-scale highly distributed edge clouds can be built at network operators' existing points of presence or implemented separately at metro, branch, and 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 computing infrastructure, as shown in Fig. 1. Exploiting this hierarchical and distributed infrastructure for service provisioning entails the joint allocation of cloud and network resources across tiers and locations. The main challenges manifest as follows:

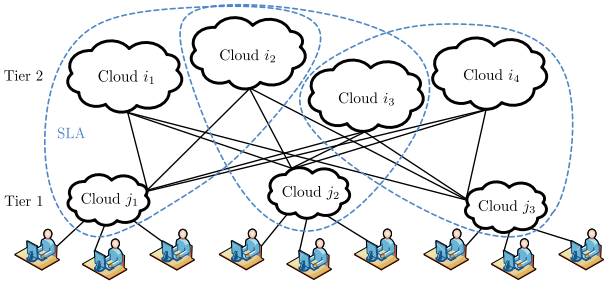


Fig. 1. The multi-tier cloud network model.

First, resource allocation and reconfiguration need to be balanced. Resource allocation incurs operating cost, *e.g.*, the cost of using physical and virtual resources such as servers, Virtual Machines (VMs), bandwidth, and energy. Resource reconfiguration, which refers to changing resource allocation decisions, incurs a different type of cost that can capture service interruption [19], hardware wear and tear [12], system instability [29], as well as resource lead time (*e.g.*, booting and initializing a VM and the services running on it) [15]. While it is desirable to allocate just enough resources to process the current workload to avoid over-provisioning and minimize operating cost, it is also beneficial that resource allocation decisions are smooth, without sharp changes over time to incur excessive reconfiguration cost. Striking the right balance under time-varying workloads and resource prices is not an easy task.

Further, dynamic resource allocation is particularly hard in an online setting, where decisions need to be made on the fly to achieve the long-term objective of optimizing allocation and reconfiguration costs over time. In cases where prediction is not possible, such as for workload flash crowds [5], a resource allocation decision for the current time slot must be made without knowing the workload and the resource price in the

Manuscript received April 23, 2016; revised December 12, 2016; accepted April 21, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Wierman. Date of publication June 8, 2017; date of current version August 16, 2017. A preliminary version of this work appeared in the Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium (IPDPS), Chicago, IL, USA, May 23–27, 2016. *(Corresponding author: Lei Jiao.)*

L. Jiao is with the Department of Computer and Information Science, University of Oregon, Eugene, OR 97403 USA (e-mail: jiao@cs.uoregon.edu).

A. M. Tulino is with Nokia Bell Labs, Holmdel, NJ 07733 USA, and also with the DIETI Department, University of Naples Federico II, 80138 Naples, Italy (e-mail: a.tulino@nokia-bell-labs.com).

J. Llorca is with Nokia Bell Labs, Holmdel, NJ 07733 USA (e-mail: jaime.llorca@nokia-bell-labs.com).

Y. Jin is with Nokia Bell Labs, 91620 Nozay, France (e-mail: yue.1.jin@nokia-bell-labs.com).

A. Sala is with Nokia Bell Labs, Dublin 15, Ireland (e-mail: alessandra.sala@nokia-bell-labs.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This consists of a PDF (343 KB) containing a three-part Appendix.

Digital Object Identifier 10.1109/TNET.2017.2707142

future. A decision for the current time slot will influence the reconfiguration cost between the current time slot and the next time slot; with zero knowledge about the next time slot, it is thus challenging to make a good decision for the current time slot. In cases where prediction about the workload and the resource price in the near future is available [20], [26], the issue becomes how to leverage such prediction to make better decisions, compared to the decisions made without prediction. Naively optimizing the cost over the current prediction window could still result in suboptimal reconfiguration between the last time slot of the current prediction window and the first time slot of the next prediction window.

Finally, resource allocation decisions must accommodate cloud and network resource heterogeneity across multiple tiers and locations, while respecting capacity limits [16], [25] and meeting Service Level Agreements (SLAs) [9], [17]. 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, only a particular subset of the upper-tier clouds may satisfy the specified SLA in terms of latency, security risk, reliability, *etc.* 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 studies fall short in addressing the aforementioned challenges. Most of them do not consider joint cloud and network resource allocation in the *multi-tier* distributed cloud infrastructure. In addition, they either ignore the reconfiguration cost [7], [8], [13], [27], or purely use prediction-based approaches [11], [20], [28], [29], not known to provide competitive guarantees when applied to the multi-tier scenario. In this paper, we make the following contributions:

We build models that can capture a range of real-world resource costs and formulate the smoothed online multi-tier resource allocation problem. The allocation cost is modeled as an affine function of the active cloud and network resources, which can capture load-proportional usage costs and pay-as-you-go business models. The reconfiguration cost is modeled assuming the cost is only incurred when increasing the amount of allocated resources from one time slot to the next, which can capture, for instance, server and VM booting and lead time. SLA is modeled by subsets, *i.e.*, for each lower-tier cloud, only a cloud in a specified subset of the upper-tier clouds may 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 resource price dynamics.

Given that prediction is unavailable, we design an online algorithm based on the technique of regularization [4], which provides a solution with a parameterized competitive ratio independent of workload and resource price dynamics. Fundamentally different from existing work, our approach decouples the original problem over time by constructing a series of subproblems where the optimal decision of a subproblem at a given time slot depends on the workload and the resource price 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 the original problem. Our algorithm, when the workload increases, allocates just enough resources to cover the current workload, and when the workload decreases, takes a controlled exponential-decay reduction in the amount of allocated resources to avoid excessive reconfiguration cost upon a future workload increase. 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.

For the case in which prediction is available, we further propose to incorporate regularization into standard prediction-based control algorithms, FHC (Fixed Horizon Control) and RHC (Receding Horizon Control) [12], [26], and design the regularized versions of the two algorithms, RFHC (Regularized Fixed Horizon Control) and RRHC (Regularized Receding Horizon Control). Via formal analysis, we first show that FHC and RHC, when applied to our resource allocation problem, can have arbitrarily bad performance, and then show the advantage of RFHC and RRHC, as they inherit the worst-case performance guarantee of our prediction-free online algorithm, while also providing improved practical performance.

We conduct 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 realistic dynamic electricity and estimated bandwidth prices, we run the sequence of greedy one-shot optimizations, our online algorithm, and the offline optimization to allocate and reconfigure resources for the 2007 Wikipedia workload [21] of 500 hours with regular dynamics and for the 1998 World Cup workload [3] of 600 hours with large spikes, respectively. Through a number of different settings, we exhibit that our online algorithm performs consistently well in practice, achieving up to $9\times$ total cost reduction compared to the greedy one-shot optimizations and at most $3\times$ the offline optimum. We also run the prediction-aware control algorithms and find that, with very moderate predictions, RFHC and RRHC can achieve $2\times$ total cost reduction compared to FHC and RHC.

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 or resource 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 bandwidth and energy 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 servers or VMs incurs considerable time while shutting them down is fast.

Workload: We target web services workload and alike, and use λ_{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 $t \in \{1, 2, 3, \ldots, T\}$ 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 also to reach one of the clouds at the top tier via different clouds in the intermediate tiers.

B. Problem Formulation

Let x_{ijt} denote the amount of resources allocated at cloud i to process the incoming workload from cloud j at time t, y_{ijt} 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} the amount of resources allocated at cloud j to process the workload that is sent to cloud i for further processing at time t. The total cost of a two-tier cloud network can be computed in terms of the following three components:

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

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

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

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

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

The problem is minimizing the total cost of cloud and network resources allocation and reconfiguration over time, while allocating sufficient resources along the service path, as in (1a), and satisfying capacity constraints, as in (1b), (1c), and (1d).

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

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

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 the paper, we focus on the following problem that we name $\mathbf{P_1}$:

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

Remarks: The affine models for allocation costs and the models based on the function $[\cdot]^+$ for reconfiguration costs seem simple yet are powerful to capture the costs incurred by a variety of different cloud resources. Allocation costs can capture VM and server usage, electricity usage, carbon emission, bandwidth usage, *etc.*, all possibly with dynamic prices. Reconfiguration costs can capture hardware wear-and-tear, resource lead time, network link establishment, *etc.* The SLA models based on subsets are also flexible in the concrete underlying criterion, which can be delay, security risk, *etc.* The case of two tiers of clouds, as captured by $\mathbf{P}_1$, is the smallest problem instance that has all the necessary elements in the multi-tier setting, including the costs inside each cloud, the costs between clouds across tiers, and the SLA constraints between clouds. Note the two key features of our problem: (1) besides the workload, operating prices are dynamic and can

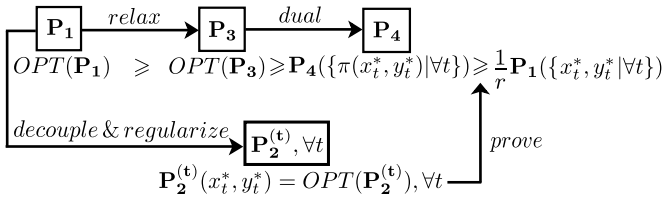


Fig. 2. Key idea.

be unbounded; (2) resource allocations are within capacity limits, and at every time slot, the sum of the resources from a specified set of clouds and networks sufficiently "cover" the corresponding workload.

III. Online Algorithm and Competitiveness

A. Key Idea

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, makes a decision for each 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 a time. The competitive ratio of an online algorithm refers to the ratio of the over-time cost incurred by the online decisions over that incurred by the offline optimal decisions, maximized over all possible inputs.

The major difficulty in solving problem $\mathbf{P_1}$ in an online manner lies in the reconfiguration cost that couples every two consecutive time slots. A resource allocation decision at a time slot can influence the reconfiguration cost associated with changing the decision from this current time slot to the next time slot—without knowing the workload and resource price and thus the decision at the next time slot, it is hard to make a good decision for the current time slot. To conquer such difficulty, we decouple the original problem $\mathbf{P_1}$ by exploiting the regularization technique to construct a series of subproblems $\{\mathbf{P_2^{(1)}}, \mathbf{P_2^{(2)}}, \ldots, \mathbf{P_2^{(t)}}, \ldots, \mathbf{P_2^{(T)}}\}$ solvable at each corresponding time slot, which allows making a decision for the current time slot with bounded proximity to the offline optimum only based on the decision of the previous time slot and the workload and resource price at the current time slot. 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 $\mathbf{P_1}$ (while Lemma 1 in the next section will show this sequence is feasible for $\mathbf{P_1}$).

Our key idea for algorithm design and competitive analysis is illustrated in Fig. 2. We proceed via 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 π 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 $\mathbf{P_i}(x)$ denote the objective function value of problem $\mathbf{i}$ evaluated at x and let $OPT(\cdot)$ denote the offline optimal objective function value. **Step 1** provides the algorithm's decision at t. From **Steps 2** and **3**, it 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. From **Step 4**, it follows $\mathbf{P_1}(\{x_t^*, y_t^*|\forall t\}) \leqslant rOPT(\mathbf{P_1})$, where 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 to $\mathbf{P_2^{(t-1)}}$ and the workload at t as input (note that the optimal solution to $\mathbf{P_2^{(0)}}$ is set to zero). We construct the following formulation as $\mathbf{P_2^{(t)}}$:

$$\begin{aligned}\min\ F_t = &\sum\nolimits_i \sum\nolimits_{j\in\mathcal{J}_i} a_{it}x_{ijt} + \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} c_{ijt}y_{ijt} \\ &+ \sum\nolimits_i \frac{b_i}{\eta_i}\left(\left(\sum_{j\in\mathcal{J}_i} x_{ijt} + \varepsilon\right)\ln\frac{\sum_{j\in\mathcal{J}_i} x_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \sum_{j\in\mathcal{J}_i} x_{ijt}\right) \\ &+ \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} \frac{d_{ij}}{\eta'_{ij}}\left((y_{ijt}+\varepsilon')\ln\frac{y_{ijt}+\varepsilon'}{y^*_{ijt-1}+\varepsilon'} - y_{ijt}\right)\end{aligned}$$

$$\text{s.t. } x_{ijt} \geqslant s_{ijt}, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{3a}$$

$$y_{ijt} \geqslant s_{ijt}, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{3b}$$

$$\sum\nolimits_{i\in\mathcal{I}_j} s_{ijt} \geqslant \lambda_{jt}, \quad \forall j, \tag{3c}$$

$$\sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \sum\nolimits_{j\in\mathcal{J}_k} x_{kjt} \geqslant \sum\nolimits_j \lambda_{jt} - C_i, \quad \forall i, \tag{3d}$$

$$\sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} y_{kjt} \geqslant \lambda_{jt} - B_{ij}, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{3e}$$

$$s_{ijt} \geqslant 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{3f}$$

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

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

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 (3d) in $\mathbf{P_2^{(t)}}$, and analogously for (2b), (2d) and (1c) in $\mathbf{P_1}$, we introduce (3e) 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^*), the optimal solution to $\mathbf{P_2^{(t)}}$, satisfies $\mathbf{P_1}$'s constraints (2a), (2b), (2d), (2e), (1b) and (1c) at t. Note that x_t^* and y_t^*, for the ease of presentation, actually refer to x_{ijt}^* and y_{ijt}^*, $\forall i \in \mathcal{I}_j$, $\forall j$. Note that when $x_{ijt} \geqslant x^*_{ijt-1}, \forall i \in \mathcal{I}_j, \forall j$,

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

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

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

That is, F_t increases monotonically for $x_{ijt} \geqslant x^*_{ijt-1}$ and $y_{ijt} \geqslant y^*_{ijt-1}$, and drops when x_{ijt} is reduced to x^*_{ijt} from a value that is larger than x^*_{ijt}, and y_{ijt} is reduced 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 decreases when x_{ij1} is reduced until $\sum_{j\in\mathcal{J}_i} x_{ij1} = C_i$ holds, and y_{ij1} is reduced 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 for $\mathbf{P_2^{(t)}}$, $\forall t$ dictate the resource allocation decisions, we consider the following simplified version of our smoothed online resource allocation problem at a single data center. Note that this problem, where the covering constraints degenerate into a very simple version as in (4a), is only for the ease of the geometric interpretation; the power of our algorithm is in fact better reflected when used to address the multi-tier multi-cloud problem where there exist complicated covering constraints.

$$\min \sum\nolimits_t a_t x_t + \sum\nolimits_t b[x_t - x_{t-1}]^+ \tag{4}$$
$$\text{s.t. } x_t \geqslant \lambda_t, \quad \forall t, \tag{4a}$$
$$x_t \leqslant C, \quad \forall t. \tag{4b}$$

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

$$a_t x_t + \frac{b}{\eta}\left((x_t+\varepsilon)\ln\frac{x_t+\varepsilon}{x^*_{t-1}+\varepsilon} - x_t\right) \tag{5}$$

where $\eta = \ln(1+C/\varepsilon)$. The problem is further decoupled over time slots. At each time slot $t \geqslant 1$, we minimize (5) subject to (4a) and (4b) at the corresponding time slot, with $x^*_0 = 0$.

By setting the derivative of (5) to zero, we get its constraint-free minimizer $\widetilde{x}_t$ as

$$\widetilde{x}_t = \left(1+\frac{C}{\varepsilon}\right)^{-\frac{a_t}{b}}(x^*_{t-1}+\varepsilon) - \varepsilon \leqslant x^*_{t-1}. \tag{6}$$

With constraints (4a) and (4b), 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 $w+1$ consecutive time slots t, $t+1$, ..., $t+w$ with the workload $\lambda_t < \lambda_{t+1} < \ldots < \lambda_{t+w}$:

- 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 $x^*_t = \lambda_t$, which further gives $\lambda_{t+1} > \lambda_t = x^*_t \geqslant \widetilde{x}_{t+1}$. Then, $\lambda_{t+1} > \widetilde{x}_{t+1}$ gives $x^*_{t+1} = \lambda_{t+1}$. This procedure can continue for any w', where $1 < w' \leqslant w$. Hence, in this case, the resource allocation follows the workload.
- In the case of $\lambda_t \leqslant \widetilde{x}_t$, by applying the recursion in (6) 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 $1 \leqslant w' \leqslant w$. In this case, if a_t does not vary with t, the resource allocation follows the exponential decay curve; 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 above) for an arbitrarily time-varying workload. At any 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 it allocates the exponential decay calculated resources; if the latter is larger, then it allocates just enough resources for 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 at 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* $B(\varepsilon') = \max_{i\in\mathcal{I}_j, j\in\mathcal{J}}\left\{(B_{ij}+\varepsilon')\ln\left(1+\frac{B_{ij}}{\varepsilon'}\right)\right\}$.

Remarks: The competitive ratio of our algorithm depends on the capacities of the system and may seem large; however, we believe it is a reasonably good ratio, due to the following reasons. Firstly, note that the way we model the problem, *e.g.*, the workload never exceeds the capacity, always allows us to normalize the inputs, including both the workload and the capacities, so that solving a normalized problem can have a much smaller competitive ratio. The decisions made by solving the normalized problem can also be translated back into the actual amount of resources. Secondly, there may be an interesting connection between our problem and a variant of the ski-rental problem [14], where, in our case, the corresponding "rental" price is time-varying and unbounded, rather than a constant as in the classic version. Although the algorithmic idea of accumulating the rental cost to match the purchase cost in the classic ski-rental problem may still be applicable here, it can be shown that the best possible competitive ratio (for any deterministic online algorithm) for our variant of the ski-rental problem is large, related to the resource price which can be unbounded in our case, which further hints that the best possible competitive ratio for our problem can also be large. We leave finding the exact best possible competitive ratio for our problem to the future work.

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 (7d) from (2a), (2d) and (1b), and deriving (7e) from (2b), (2d) and (1c), we relax $\mathbf{P_1}$ to $\mathbf{P_3}$:

$$\min \sum\nolimits_t\sum\nolimits_i\sum\nolimits_{j\in\mathcal{J}_i} a_{it}x_{ijt} + \sum\nolimits_t\sum\nolimits_i b_i v_{it} + \sum\nolimits_t\sum\nolimits_j\sum\nolimits_{i\in\mathcal{I}_j} c_{ijt}y_{ijt} + \sum\nolimits_t\sum\nolimits_j\sum\nolimits_{i\in\mathcal{I}_j} d_{ij}w_{ijt}$$
$$\text{s.t. } v_{it} \geqslant \sum\nolimits_{j\in J_i} x_{ijt} - \sum\nolimits_{j\in\mathcal{J}_i} x_{ijt-1}, \quad \forall i,\ \forall t, \tag{7a}$$
$$w_{ijt} \geqslant y_{ijt} - y_{ijt-1}, \quad \forall i\in\mathcal{I}_j,\ \forall j, \forall t, \tag{7b}$$

$$v_{it} \geqslant 0, w_{ijt} \geqslant 0, \quad \forall i,\ \forall j,\ \forall t, \tag{7c}$$

$$\sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \sum\nolimits_{j\in\mathcal{J}_k} x_{kjt} \geqslant \left[\sum\nolimits_j \lambda_{jt} - C_i\right]^+, \quad \forall i,\ \forall t, \tag{7d}$$

$$\sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} y_{kjt} \geqslant [\lambda_{jt} - B_{ij}]^+, \quad \forall i \in \mathcal{I}_j,\ \forall j,\ \forall t, \tag{7e}$$

(2a), (2b), (2d), (2e),

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 (7d) and (7e).

Step 2.2: We derive the Lagrange dual problem of $\mathbf{P_3}$. Let α_{it}, β_{ijt}, δ_{it}, θ_{ijt} be the dual variables associated with (7a), (7b), (7d) and (7e), respectively; let ρ_{ijt}, ϕ_{ijt}, γ_{jt} be the dual variables associated with (2a), (2b) and (2d), respectively. We have the dual problem $\mathbf{P_4}$:

$$\begin{aligned}\max\ & D = \sum\nolimits_t \sum\nolimits_j \lambda_{jt}\gamma_{jt} + \sum\nolimits_t \sum\nolimits_i \left[\sum\nolimits_j \lambda_{jt} - C_i\right]^+ \delta_{it} \\ & + \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} [\lambda_{jt} - B_{ij}]^+ \theta_{ijt} \qquad (8)\\ \text{s.t. } & a_{it} + \alpha_{it} - \alpha_{it+1} - \rho_{ijt} - \sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \delta_{kt} = 0, \\ & \qquad \forall i \in \mathcal{I}_j, \quad \forall j,\ \forall t, \qquad (8a)\\ & c_{ijt} + \beta_{ijt} - \beta_{ijt+1} - \phi_{ijt} - \sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} \theta_{kjt} = 0, \\ & \qquad \forall i \in \mathcal{I}_j,\ \forall j,\ \forall t, \qquad (8b)\\ & \rho_{ijt} + \phi_{ijt} - \gamma_{jt} \geqslant 0, \quad \forall i \in \mathcal{I}_j,\ \forall j,\ \forall t, \qquad (8c)\\ & b_i - \alpha_{it} \geqslant 0, \quad \forall i,\ \forall t, \qquad (8d)\\ & d_j - \beta_{jt} \geqslant 0, \quad \forall j,\ \forall t, \qquad (8e)\\ & \alpha_{it} \geqslant 0, \quad \delta_{it} \geqslant 0,\ \gamma_{jt} \geqslant 0, \forall i, \forall j, \forall t; \\ & \beta_{ijt} \geqslant 0, \quad \theta_{ijt} \geqslant 0,\ \rho_{ijt} \geqslant 0,\ \phi_{ijt} \geqslant 0, \\ & \qquad \forall i \in \mathcal{I}_j,\ \forall j,\ \forall t. \qquad (8f)\end{aligned}$$

Step 3.1: We write the following KKT conditions that characterize the optimal solution x^*_{ijt}, y^*_{ijt} of $\mathbf{P_2^{(t)}}$, where ρ'_{ijt}, ϕ'_{ijt}, γ'_{jt} are the dual variables associated with (3a), (3b), (3c), respectively, δ'_{it}, θ'_{ijt} are the dual variables associated with (3d), (3e), respectively, and p_{ijt} is the dual variable for (3f). Note $x_{ijt} \geqslant 0$, $y_{ijt} \geqslant 0$ due to (3a), (3b) and (3f), and thus we can apply $[\cdot]^+$ in (9g) and (9h).

$$a_{it} + \frac{b_i}{\eta_i} \ln \frac{\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \rho'_{ijt} - \sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \delta'_{kt} = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9a}$$

$$c_{ijt} + \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} - \phi'_{ijt} - \sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} \theta'_{kjt} = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9b}$$

$$\rho'_{ijt} + \phi'_{ijt} - \gamma'_{jt} - p_{ijt} = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9c}$$

$$\rho'_{ijt}(s^*_{ijt} - x^*_{ijt}) = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9d}$$

$$\phi'_{ijt}(s^*_{ijt} - y^*_{ijt}) = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9e}$$

$$\gamma'_{jt}\left(\lambda_{jt} - \sum\nolimits_{i\in\mathcal{I}_j} s^*_{ijt}\right) = 0, \quad \forall j, \tag{9f}$$

$$\delta'_{it}\left(\left[\sum\nolimits_j \lambda_{jt} - C_i\right]^+ - \sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \sum\nolimits_{j\in\mathcal{J}_k} x^*_{kjt}\right) = 0, \forall i, \tag{9g}$$

$$\theta'_{ijt}\left([\lambda_{jt} - B_{ij}]^+ - \sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} y^*_{kjt}\right) = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9h}$$

$$p_{ijt}s^*_{ijt} = 0, \quad \forall i \in \mathcal{I}_j,\ \forall j, \tag{9i}$$

$$\rho'_{ijt} \geqslant 0, \quad \phi'_{ijt} \geqslant 0,\ \theta'_{ijt} \geqslant 0,\ p_{ijt} \geqslant 0,\ \forall i \in \mathcal{I}_j,\ \forall j; \quad \gamma'_{jt} \geqslant 0, \quad \delta'_{it} \geqslant 0,\ \forall j,\ \forall i. \tag{9j}$$

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:

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

To see the feasibility, let us take constraint (8a) as an example. Putting them into the left-hand side of (8a), we get

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

The above holds due to (9a). Analogously, (8b) holds due to (9b); (8c) holds due to (9c) and (9j); (8d), (8e) hold due to $x^*_{ijt} \geqslant 0$, $y^*_{ijt} \geqslant 0$, as in (3a), (3b), (3f). In (8f), $\alpha_{it} \geqslant 0$, $\beta_{ijt} \geqslant 0$ 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 (9j).

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 α_{it}, β_{ijt}, ρ_{ijt}, ϕ_{ijt}, γ_{jt}, δ_{it}, θ_{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: First, we bound the allocation cost.

$$\sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} a_{it}x^*_{ijt} + \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} c_{ijt}y^*_{ijt} \tag{10}$$

$$= \sum_t \sum_j \sum_{i\in\mathcal{I}_j} x^*_{ijt} \left(\rho_{ijt} - \frac{b_i}{\eta_i} \ln \frac{\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} + \sum\nolimits_{\substack{k\in\mathcal{I}\\k\neq i}} \delta_{kt}\right) + \sum_t \sum_j \sum_{i\in\mathcal{I}_j} y^*_{ijt} \left(\phi_{ijt} - \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} + \sum\nolimits_{\substack{k\in\mathcal{I}_j\\k\neq i}} \theta_{kjt}\right) \tag{10a}$$

$$= \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} x^*_{ijt}\rho_{ijt} + \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} y^*_{ijt}\phi_{ijt} + \Delta - \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} x^*_{ijt} \frac{b_i}{\eta_i} \ln \frac{\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} - \sum\nolimits_t \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} y^*_{ijt} \frac{d_{ij}}{\eta'_{ij}} \ln \frac{y^*_{ijt} + \varepsilon'}{y^*_{ijt-1} + \varepsilon'} \tag{10b}$$

$$\leqslant \sum_t \sum_j \sum_{i\in\mathcal{I}_j} s^*_{ijt}(\rho_{ijt}+\phi_{ijt}) + \Delta \tag{10c}$$

$$= \sum_t \sum_j \sum_{i\in\mathcal{I}_j} s^*_{ijt}\gamma_{jt} + \Delta \tag{10d}$$

$$= D \tag{10e}$$

(10a) follows from (9a) and (9b). (10b) follows from (9g) and (9h), where

$$\Delta = \sum_t \sum_i \Big[\sum_j \lambda_{jt} - C_i\Big]^+ \delta_{it} + \sum_t \sum_j \sum_{i\in\mathcal{I}_j} [\lambda_{jt} - B_{ij}]^+ \theta_{ijt}.$$

(10c) follows from (9d), (9e), and the following 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 $\sum_t \sum_j \sum_{i\in\mathcal{I}_j} y^*_{ijt} \frac{d_j}{\eta'_{ij}} \ln \frac{y^*_{ijt}+\varepsilon'}{y^*_{ijt-1}+\varepsilon'} \geqslant 0$. (10d) follows from (9c) and (9i). (10e) follows from (9f) and (8). 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 (11a) and (11e) is no less than zero:

$$\sum_t (y^*_{ijt}+\varepsilon') \ln \frac{y^*_{ijt}+\varepsilon'}{y^*_{ijt-1}+\varepsilon'} \tag{11a}$$

$$\geqslant \Big(\sum_t (y^*_{ijt}+\varepsilon')\Big) \ln \frac{\sum_t (y^*_{ijt}+\varepsilon')}{\sum_t (y^*_{ijt-1}+\varepsilon')} \tag{11b}$$

$$\geqslant \sum_t (y^*_{ijt}+\varepsilon') - \sum_t (y^*_{ijt-1}+\varepsilon') \tag{11c}$$

$$= y^*_{ijT} - y^*_{ij0} \tag{11d}$$

$$-\sum_t \varepsilon' \ln \frac{y^*_{ijt}+\varepsilon'}{y^*_{ijt-1}+\varepsilon'} \tag{11e}$$

$$= (y^*_{ij0}+\varepsilon') \ln \frac{y^*_{ij0}+\varepsilon'}{y^*_{ijT}+\varepsilon'} \tag{11f}$$

$$\geqslant y^*_{ij0} - y^*_{ijT} \tag{11g}$$

(11b) follows from (12b) as below. (11c) and (11g) follow from (12a) as below. (11f) follows due to $y^*_{ij0} = 0$. (12a) and (12b) are two facts that we exploit.

$$m - n \leqslant m \ln \frac{m}{n}, \quad \forall m,\ n > 0, \tag{12a}$$

$$\Big(\sum_i m_i\Big) \ln \frac{\sum_i m_i}{\sum_i n_i} \leqslant \sum_i m_i \ln \frac{m_i}{n_i}, \quad \forall m, n > 0. \tag{12b}$$

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

$$\mathcal{I}_t^+ \triangleq \{i | \sum_{j\in\mathcal{J}_i} x^*_{ijt} > \sum_{j\in\mathcal{J}_i} x^*_{ijt-1}, \forall i \in \mathcal{I}\}, \tag{13a}$$

$$\{\mathcal{I}_j \times \mathcal{J}\}_t^+ \triangleq \{(i,j) | y^*_{ijt} > y^*_{ijt-1}, \forall i \in \mathcal{I}_j, \forall j \in \mathcal{J}\}. \tag{13b}$$

We bound the first part of the reconfiguration cost:

$$\sum_t \sum_{i\in\mathcal{I}} b_i \Big[\sum_{j\in\mathcal{J}_i} x^*_{ijt} - \sum_{j\in\mathcal{J}_i} x^*_{ijt-1}\Big]^+ \tag{14}$$

$$= \sum_t \sum_{i\in\mathcal{I}_t^+} b_i \Big(\sum_{j\in\mathcal{J}_i} x^*_{ijt} - \sum_{j\in\mathcal{J}_i} x^*_{ijt-1}\Big) \tag{14a}$$

$$\leqslant \sum_t \sum_{i\in\mathcal{I}_t^+} b_i \Big(\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon\Big) \ln \frac{\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \tag{14b}$$

$$\leqslant \max_i\{(C_i+\varepsilon)\eta_i\} \sum_t \sum_{i\in\mathcal{I}_t^+} \frac{b_i}{\eta_i} \ln \frac{\sum_{j\in\mathcal{J}_i} x^*_{ijt} + \varepsilon}{\sum_{j\in\mathcal{J}_i} x^*_{ijt-1} + \varepsilon} \tag{14c}$$

$$\leqslant C(\varepsilon) \sum_t \sum_{\substack{i\in\mathcal{I}_t^+ \\ x^*_{ijt}>0}} \Big(\rho_{ijt} + \sum_{k\in\mathcal{I}\atop k\neq i} \delta_{kt}\Big) \tag{14d}$$

$$\leqslant C(\varepsilon) \sum_t \left(\sum_{\substack{i\in\mathcal{I}_t^+ \\ x^*_{ijt}>0 \\ \rho_{ijt}>0}} (\gamma_{jt} + p_{ijt} - \phi_{ijt}) + |\mathcal{I}| \sum_i \delta_{it}\right) \tag{14e}$$

$$\leqslant C(\varepsilon)|\mathcal{I}| \sum_t (\gamma_{jt} + \sum_i \delta_{it}) \tag{14f}$$

$$\leqslant C(\varepsilon)|\mathcal{I}| D \tag{14g}$$

(14a) follows from (13a). (14b) follows from (12a). (14c) follows, due to $\sum_{j\in\mathcal{J}_i} x^*_{ijt} \leqslant C_i$. (14d) follows from (9a). Note that in (14d), for any given $i \in \mathcal{I}_t^+$, we can choose to use any ρ_{ijt}, $j \in \mathcal{J}_i$; however, we choose the particular ρ_{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 (14e) only for those i where $\rho_{ijt} > 0$; if $\rho_{ijt} = 0$, $\forall i \in \mathcal{I}_t^+$, we can directly reach (14g) from (14d). (14e) follows from (9c). (14f) follows, because of $p_{ijt} = 0$. Applying $x^*_{ijt} > 0$, $\rho_{ijt} > 0$ to (9d), we have $s^*_{ijt} > 0$; applying $s^*_{ijt} > 0$ to (9i), we have $p_{ijt} = 0$. (14g) follows, because of (8), $\gamma_{jt} > 0$ and $\lambda_{jt} \geqslant 1$. $\gamma_{jt} > 0$ is due to (9c), $\rho_{ijt} > 0$ and $p_{ijt} = 0$; $\lambda_{jt} \geqslant 1$ holds because λ_{jt} is an integer, and $\lambda_{jt} > 0$ due to (9f), $\gamma_{jt} > 0$ and $s^*_{ijt} > 0$. We also require C_i to be an integer. Note that if $\sum_j \lambda_{jt} - C_i \leqslant 0$, then we will have no (7d) and no dual variable δ_{it}, and so (8) changes accordingly and (14) $\leqslant$ (14g) still holds.

We bound the second part of the reconfiguration cost:

$$\sum_t \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}_j} d_{ij} [y^*_{ijt} - y^*_{ijt-1}]^+ \tag{15}$$

$$= \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} d_{ij}(y^*_{ijt} - y^*_{ijt-1}) \tag{15a}$$

$$\leqslant \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} d_{ij}(y^*_{ijt}+\varepsilon') \ln \frac{y^*_{ijt}+\varepsilon}{y^*_{ijt-1}+\varepsilon'} \tag{15b}$$

$$\leqslant \max_{i,j}\{(B_{ij}+\varepsilon')\eta'_{ij}\} \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} \frac{d_{ij}}{\eta'_{ij}} \times \ln \frac{y^*_{ijt}+\varepsilon}{y^*_{ijt-1}+\varepsilon'} \tag{15c}$$

$$\leqslant B(\varepsilon') \sum_t \sum_{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+} \Big(\phi_{ijt} + \sum_{k\in\mathcal{I}_j\atop k\neq i} \theta_{kjt}\Big) \tag{15d}$$

$$\leqslant B(\varepsilon') \sum_t \left(\sum_{\substack{(i,j)\in\{\mathcal{I}_j\times\mathcal{J}\}_t^+ \\ \phi_{ijt}>0}} (\gamma_{jt} + p_{ijt} - \rho_{ijt}) + |\mathcal{I}| \sum_j \sum_{i\in\mathcal{I}_j} \theta_{ijt}\right) \tag{15e}$$

$$\leqslant B(\varepsilon')|\mathcal{I}|\sum\nolimits_t(\sum\nolimits_j \gamma_{jt} + \sum\nolimits_j \sum\nolimits_{i\in\mathcal{I}_j} \theta_{ijt}) \quad (15f)$$

$$\leqslant B(\varepsilon')|\mathcal{I}|D \quad (15g)$$

(15a) follows from (13b). (15b) follows from (12a). (15c) follows, due to $y^*_{ijt} \leqslant B_{ij}$. (15d) follows from (9b). We continue to (15e) 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 (15g) from (15d). (15e) follows from (9c). (15f) follows, because of $p_{ijt} = 0$. Applying $y^*_{ijt} > y^*_{ijt-1} \geqslant 0$, $\phi_{ijt} > 0$ to (9e), we have $s^*_{ijt} > 0$; applying $s^*_{ijt} > 0$ to (9i), we have $p_{ijt} = 0$. Finally, reaching (15g) is analogous to reaching (14g).

E. Generalization

Our models, online algorithm, and competitive analysis can be generalized to arbitrary $N \geqslant 2$ tiers of clouds [10]. Due to the page limit, we put the theorem on the competitive ratio for N-tier clouds and its proof sketch in a supplementary file which is published accompanying this paper.

IV. Online Algorithms Using Predictions

In this section, we introduce existing standard control algorithms that use predictions, prove their lack of worst-case performance guarantees for our problem, and afterwards we explore the regularization technique to design novel online algorithms that leverage predictions to further enhance the performance of our prediction-oblivious online algorithm, while providing worst-case performance guarantees.

A. Standard Control Algorithms

Notations: Throughout this section, we use x_t to generally denote a feasible solution to $\mathbf{P_1}$ at t. For instance, x_t refers to $(x_{ijt}, y_{ijt}, z_{ijt})$ in the two-tier cloud scenario. We use $\mathbf{P_1}(x_{m-1}; x_m \dots x_{m+n})$ to denote the objective function value of $\mathbf{P_1}$ evaluated with the solution $\{x_m, x_{m+1}, \dots, x_{m+n}\}$ over the time slots $\{m, m+1, \dots, m+n\}$, given the solution x_{m-1} at the time slot $m-1$. We use $\mathbf{P_1}^{(x_{m-1};m\dots m+n;)}$ to denote the problem of minimizing $\mathbf{P_1}$ over the time slots $\{m, m+1, \dots, m+n\}$ given the solution x_{m-1} at the time slot $m-1$, and use $\mathbf{P_1}^{(x_{m-1};m\dots m+n;x_{m+n})}$ to denote the problem of minimizing $\mathbf{P_1}$ over the same time slots, given x_{m-1} at the time slot $m-1$ and x_{m+n} at the time slot $m+n$. Based on these definitions, we have $\mathbf{P_1} \triangleq \mathbf{P_1}^{(x_0;1\dots T;)} = \mathbf{P_1}^{(x_0;1\dots T+1;x_{T+1})}$, where $x_0 = x_{T+1} = 0$.

Standard Algorithms: The two standard online control algorithms that use predictions are FHC (Fixed Horizon Control) and RHC (Receding Horizon Control). Assuming that at any $t \geqslant 1$ we have the exact prediction of all the operating prices and the workloads for the w time slots $\{t, t+1, \dots, t+w-1\}$, where $w \geqslant 1$ is the length of the prediction window, the two control algorithms are described as follows. In particular, when $w = 1$, both FHC and RHC fall back to the sequence of one-shot optimizations which we also call greedy control.

- FHC: At the time slot t, where $t = 1, w+1, 2w+1, \dots$, we solve $\mathbf{P_1}^{(x_{t-1};t\dots t+w-1;)}$ and apply the solution $\{x_t, \dots, x_{t+w-1}\}$ to the time slots $\{t, \dots, t+w-1\}$.
- RHC: At the time slot t, where $t = 1, 2, 3, \dots$, we solve $\mathbf{P_1}^{(x_{t-1};t\dots t+w-1;)}$ and acquire the solution $\{x_t, \dots, x_{t+w-1}\}$, but only apply x_t to the time slot t.

B. Limitation of Standard Algorithms

We demonstrate that FHC and RHC, when used to solve our problem, can have arbitrarily bad performance. We first characterize the shape of the geometric curve of the offline optimal resource allocation for a simple workload (Lemma 2), and then prove that, for this workload, the total cost over time incurred by greedy control (Theorem 2) as well as by FHC and RHC (Theorem 3) can be arbitrarily larger than the offline optimum. We consider a simplified problem as in (4), (4a) and (4b), where $a_t > 0$, $0 < \lambda_t \leqslant C$, $\forall t$ and $b > 0$, for the ease of presentation.

Lemma 2: Given a workload $\{\lambda_t\}$ which strictly decreases monotonically from t_0 to t_2 and strictly increases monotonically from t_2 to t_4, and given $a_t > 0$, $\forall t$ and $b > 0$, it is always possible to find an offline optimal resource allocation which follows the workload from t_0 to t_1, stays constant from t_1 to t_3, and follows the workload from t_3 to t_4, with t_1 and t_3 as two proper time slots satisfying $t_0 \leqslant t_1 \leqslant t_2 \leqslant t_3 \leqslant t_4$. Specifically, if $\sum_{t=t_2-1}^{t_2+1} a_t \leqslant b$, with $t_2 - 1 \geqslant t_1$ and $t_2 + 1 \leqslant t_4$, then t_1 and t_3 satisfy $t_0 \leqslant t_1 < t_2 < t_3 \leqslant t_4$.

Proof: To prove this lemma, we show that, for any given feasible resource allocation, $\{\lambda^*_t\}$, where $\lambda^*_t \geqslant \lambda_t$, $\forall t \in [t_0, t_4]$, there always exists another feasible resource allocation $\{\widetilde{\lambda}^*_t\}$, which has the curve of the shape described in the lemma and has the total cost of allocation and reconfiguration *no larger* than that of the given feasible resource allocation. The offline optimal solution is also a feasible solution, and thus there always exists a corresponding optimal solution with the shape described in the lemma and the *same* total cost.

Firstly, we identify the locations of t_1 and t_3. To that end, we find $t_{\min}$, where $\lambda^*_t \geqslant \lambda^*_{t_{\min}}$, $\forall t \in [t_0, t_4]$. A line segment staying at $\lambda^*_{t_{\min}}$ can be drawn:

- If $\lambda^*_{t_{\min}} \geqslant \min\{\lambda_{t_0}, \lambda_{t_4}\}$ and $\lambda_{t_0} > \lambda_{t_4}$, it intersects λ_t and the vertical line of $t = t_4$. In this case, $t_0 \leqslant t_1 < t_2 < t_3 = t_4$, shown in Fig. 3a.
- If $\lambda^*_{t_{\min}} \geqslant \min\{\lambda_{t_0}, \lambda_{t_4}\}$ and $\lambda_{t_0} \leqslant \lambda_{t_4}$, it intersects λ_t and the vertical line of $t = t_0$. In this case, $t_0 = t_1 < t_2 < t_3 \leqslant t_4$, shown in Fig. 3b.
- If $\lambda_{t_2} < \lambda^*_{t_{\min}} < \min\{\lambda_{t_0}, \lambda_{t_4}\}$, it intersects λ_t. In this case, $t_0 < t_1 < t_2 < t_3 < t_4$, shown in Fig. 3c.

Note that the line segment can also degrade to a point[1]:

- If $\lambda^*_{t_{\min}} = \lambda_{t_2}$, it intersects λ_t. In this case, $t_0 < t_1 = t_2 = t_3 < t_4$, shown in Fig. 3d. Later in the proof, we show we could find another t_1 and t_3 with $t_1 < t_2 < t_3$.

Secondly, we show that for all cases above we can construct $\{\widetilde{\lambda}^*_t\}$ whose total cost of allocation and reconfiguration

[1]More strictly, a "point" in a time-slotted system should be a horizontal line segment with the length of a time slot. In this sense, in our problem, the line segment degrades to a point if $\lambda^*_{t_{\min}} < \min\{\lambda_{t_2-1}, \lambda_{t_2+1}\}$; similarly, later in the third step of our proof, we have t_1, t_3 with $t_1 < t_2 < t_3$ if the workload satisfies $\max\{\lambda_{t_1}, \lambda_{t_3}\} \leqslant \min\{\lambda_{t_1-1}, \lambda_{t_3+1}\}$. Our current proof can be adjusted to address such details if necessary, while we find it easier to present our proof without these details.

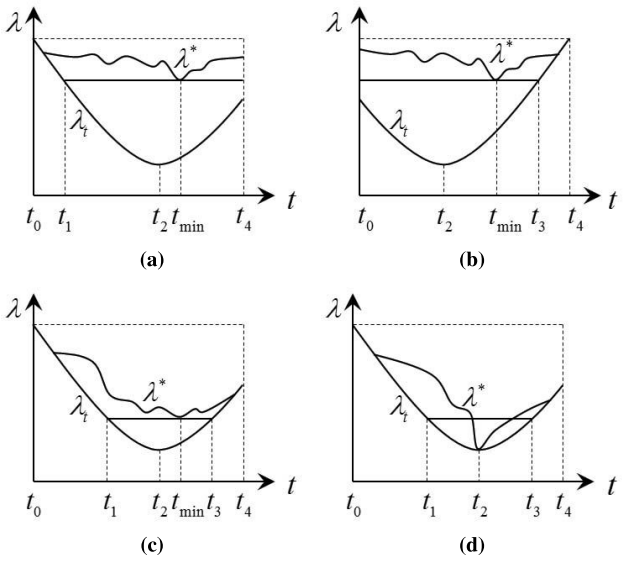


Fig. 3. Workload and resource allocation. (a) $t_0 \leqslant t_1 < t_2 < t_3 = t_4$. (b) $t_0 = t_1 < t_2 < t_3 \leqslant t_4$. (c) $t_0 < t_1 < t_2 < t_3 < t_4$. (d) $t_0 < t_1 = t_2 = t_3 < t_4$ and $t_0 < t_1 < t_2 < t_3 < t_4$.

is no larger than that of $\{\lambda_t^*\}$. We construct $\{\widetilde{\lambda}_t^*\}$ as follows: $\widetilde{\lambda}_t^* = \lambda_t$, $\forall t \in [t_0, t_1]$; $\widetilde{\lambda}_t^* = \lambda_{t_{\min}}^*$, $\forall t \in [t_1, t_3]$; $\widetilde{\lambda}_t^* = \lambda_t$, $\forall t \in [t_3, t_4]$. Note that here we permit t_1 to be equal to t_3. Consider the allocation cost:

$$COST_{\{\widetilde{\lambda}_t^*\}}^{Alloc} = \sum_{t=t_0}^{t_1} a_t\lambda_t + \lambda_{t_{\min}}^* \sum_{t=t_1}^{t_3} a_t + \sum_{t=t_3}^{t_4} a_t\lambda_t,$$
$$COST_{\{\lambda_t^*\}}^{Alloc} = \sum_{t=t_0}^{t_1} a_t\lambda_t^* + \sum_{t=t_1}^{t_3} a_t\lambda_t^* + \sum_{t=t_3}^{t_4} a_t\lambda_t^*.$$

We can see $COST_{\{\widetilde{\lambda}_t^*\}}^{Alloc} \leqslant COST_{\{\lambda_t^*\}}^{Alloc}$, because of $\lambda_t \leqslant \lambda_t^*$, $\forall t \in [t_0, t_1] \cup [t_3, t_4]$ and $\lambda_{t_{\min}}^* \leqslant \lambda_t^*$, $\forall t \in [t_1, t_3]$. Consider the reconfiguration cost:

$$\begin{aligned} &COST_{\{\widetilde{\lambda}_t^*\}}^{Reconfig} \\ &\quad= \sum_{t=t_0}^{t_4} b[\widetilde{\lambda}_t^* - \widetilde{\lambda}_{t-1}^*]^+ \\ &\quad= b(\lambda_{t_4} - \lambda_{t_{\min}}^*) \\ &\quad= b(\lambda_{t_4} - \lambda_{t_4-1}^*) + b\sum_{t=t_{\min}+1}^{t_4-1} (\lambda_t^* - \lambda_{t-1}^*), \\ &COST_{\{\lambda_t^*\}}^{Reconfig} \\ &\quad= \sum_{t=t_0}^{t_4} b[\lambda_t^* - \lambda_{t-1}^*]^+ \\ &\quad\geqslant \sum_{t=t_{\min}+1}^{t_4} b[\lambda_t^* - \lambda_{t-1}^*]^+ \\ &\quad= b[\lambda_{t_4}^* - \lambda_{t_4-1}^*]^+ + b\sum_{t=t_{\min}+1}^{t_4-1} [\lambda_t^* - \lambda_{t-1}^*]^+. \end{aligned}$$

We can also see $COST_{\{\widetilde{\lambda}_t^*\}}^{Reconfig} \leqslant COST_{\{\lambda_t^*\}}^{Reconfig}$. This is because $\lambda_{t_4} \leqslant \lambda_{t_4}^*$ leads to $\lambda_{t_4} - \lambda_{t_4-1}^* \leqslant [\lambda_{t_4}^* - \lambda_{t_4-1}^*]^+$; besides, we have $\lambda_t^* - \lambda_{t-1}^* \leqslant [\lambda_t^* - \lambda_{t-1}^*]^+$, $\forall t \in [t_0, t_4]$.

Thirdly, we investigate a special case: $\sum_{t=t_2-1}^{t_2+1} a_t \leqslant b$. We have proved in the above that for arbitrary $a_t > 0$, $\forall t$ and $b > 0$ the total cost of $\{\widetilde{\lambda}_t^*\}$, as constructed above (*i.e.*, either $t_1 < t_2 < t_3$ in the first three cases or $t_1 = t_2 = t_3$ in the last case), is no larger than that of $\{\lambda_t^*\}$. Now, we show that even in the last case, if it is the special case as described here, there still exists $\{\widetilde{\lambda}_t^*\}$, and t_1, t_3 with $t_1 < t_2 < t_3$, which has the total cost no larger than that of $\{\lambda_t^*\}$. In order to show this, we need to construct $\{\widetilde{\lambda}_t^*\}$ in a different way.

Given $\lambda_{t_{\min}}^* = \lambda_{t_2}$, we have

$$\sum_{t=t_0}^{t_4} a_t\lambda_t \leqslant COST_{\{\lambda_t^*\}}^{Alloc}; b(\lambda_{t_4} - \lambda_{t_2}) \leqslant COST_{\{\lambda_t^*\}}^{Reconfig}. \tag{16}$$

If $\sum_{t=t_2-1}^{t_2+1} a_t \leqslant b$, then there always exist t_1, t_3 with $t_1 < t_2 < t_3$ (*e.g.*, at least t_1 can be $t_2 - 1$ and t_3 can be $t_2 + 1$), which satisfy $\sum_{t=t_1}^{t_3} a_t \leqslant b$. Besides, we have $\lambda_t \geqslant \lambda_{t_2}$, $\forall t$. Consequently, we have

$$\begin{aligned} \widetilde{\lambda}_{t_1}^* \sum_{t=t_1}^{t_3} a_t + b(\lambda_{t_4} - \widetilde{\lambda}_{t_1}^*) &\leqslant \lambda_{t_2} \sum_{t=t_1}^{t_3} a_t + b(\lambda_{t_4} - \lambda_{t_2}) \\ &\leqslant \sum_{t=t_1}^{t_3} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2}), \end{aligned}$$

where $\widetilde{\lambda}_{t_1}^* = \max\{\lambda_{t_1}, \lambda_{t_3}\}$. Adding $\sum_{t=t_0}^{t_1} a_t\lambda_t + \sum_{t=t_3}^{t_4} a_t\lambda_t$ to both sides of the inequality results in

$$\begin{aligned} &\sum_{t=t_0}^{t_1} a_t\lambda_t + \widetilde{\lambda}_{t_1}^* \sum_{t=t_1}^{t_3} a_t + \sum_{t=t_3}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \widetilde{\lambda}_{t_1}^*) \\ &\quad\leqslant \sum_{t=t_0}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2}) \\ &\quad\leqslant COST_{\{\lambda_t^*\}}^{Alloc} + COST_{\{\lambda_t^*\}}^{Reconfig}. \end{aligned} \tag{17a}$$

(17a) follows from (16). The left-hand side of the inequality above also tells us how to construct $\{\widetilde{\lambda}_t^*\}$: $\widetilde{\lambda}_t^* = \lambda_t$, $\forall t \in [t_0, t_1]$; $\widetilde{\lambda}_t^* = \widetilde{\lambda}_{t_1}^*$, $\forall t \in [t_1, t_3]$; $\widetilde{\lambda}_t^* = \lambda_t$, $\forall t \in [t_3, t_4]$. □

Theorem 2: The worst-case cost of greedy control can be arbitrarily larger than the corresponding offline optimum.

Proof: To prove this theorem, let us consider the workload in Lemma 2. The solution produced by greedy control (*i.e.*, the sequence of one-shot optimizations) is always following the workload. Thus, the cost of greedy control is

$$\begin{aligned} COST_{greedy} &= \sum_{t=t_0}^{t_4} a_t\lambda_t + \sum_{t=t_0}^{t_4} b[\lambda_t - \lambda_{t-1}]^+ \\ &= \sum_{t=t_0}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2}), \end{aligned}$$

and the offline optimum is, following Lemma 2,

$$COST_{opt} = \sum_{t=t_0}^{t_1} a_t\lambda_t + \widetilde{\lambda}_{t_1}^* \sum_{t=t_1}^{t_3} a_t + \sum_{t=t_3}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*),$$

assuming $\widetilde{\lambda}_{t_0-1}^* = 0$. Now we have

$$\begin{aligned} \frac{COST_{greedy}}{COST_{opt}} &= \frac{\sum_{t=t_0}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2})}{\sum_{t=t_0}^{t_4} a_t\widetilde{\lambda}_t^* + b(\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*)} \\ &= \frac{1 + \widetilde{b}(\lambda_{t_4} - \lambda_{t_2})}{\kappa + \widetilde{b}(\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*)}, \end{aligned}$$

where $\widetilde{b} = \frac{b}{\sum_{t=t_0}^{t_4} a_t\lambda_t}$ and $\kappa = \frac{\sum_{t=t_0}^{t_4} a_t\widetilde{\lambda}_t^*}{\sum_{t=t_0}^{t_4} a_t\lambda_t}$. When $\widetilde{b} \gg 1$, we have $\frac{COST_{greedy}}{COST_{opt}} \approx \frac{\widetilde{b}(\lambda_{t_4} - \lambda_{t_2})}{\widetilde{b}(\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*)}$, from which it follows that the ratio can be arbitrary large if the $\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*$ is arbitrary small. From Lemma 2, it follows that $\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*$ is a decreasing function of the value $\widetilde{b}$. In particular, for $\widetilde{b} \to +\infty$, $\lambda_{t_4} - \widetilde{\lambda}_{t_3}^* \to 0$. □

Theorem 3: Given that the length of the prediction window is smaller than the length of the workload, the worst-case

cost of FHC and RHC can be arbitrarily larger than the corresponding offline optimum.

Proof: Consider the workload in Lemma 2. Additionally, let us consider the case of $\max\{t_0, t_4 - 2w\} < t_2 < t_4 - w$, where w is the length of the prediction window.

FHC divides the entire workload into noninterlaced prediction windows, and the resource allocation follows the workload in every prediction window as the workload is monotonic, until it enters the prediction window that contains t_2, and afterwards it enters the last prediction window and follows the workload as it is monotonic again. Assuming the prediction window that contains t_2 is from t_2' to t_2'', where $t_2'' < t_4$ because $t_2 < t_4 - w$, we have $COST_{FHC} \geqslant \sum_{t=t_2''}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2''}) > 0$, while, again, from Lemma 2, it follows that $\lambda_{t_4} - \widetilde{\lambda}_{t_3}^*$ is a decreasing function of the value $\widetilde{b}$, and in particular, $COST_{opt} \to 0$ for $a_t \to 0$. Thus, $COST_{FHC}/COST_{opt} \to +\infty$.

RHC allocates resource at every time slot using the predicted information in the current prediction window that starts from the current time slot. However, note that no matter how RHC allocates resource before $t_2 + w$, from $t_2 + w$ until the end of the workload RHC always follows the workload. Analogous to the above analysis, $COST_{RHC} \geqslant \sum_{t=t_2+w}^{t_4} a_t\lambda_t + b(\lambda_{t_4} - \lambda_{t_2+w}) > 0$. When $a_t \to 0$, $COST_{opt} \to 0$, and $COST_{RHC}/COST_{opt} \to +\infty$. □

C. Regularized Control Algorithms

We propose to incorporate regularization to FHC and RHC, and design two novel control algorithms RFHC (Regularized Fixed Horizon Control) and RRHC (Regularized Receding Horizon Control) correspondingly. RFHC and RRHC are upper-bounded by our online algorithm that uses no prediction, and thus inherit its competitive ratio.

- RFHC: At the time slot t, where $t = 1, w+1, 2w+1, \ldots$, we solve $\{\mathbf{P_2}^{(t)}, \ldots, \mathbf{P_2}^{(t+w-1)}\}$ and obtain the solution $\{x_t^*, \ldots, x_{t+w-1}^*\}$. Then, we keep x_{t+w-1}^*, solve $\mathbf{P_1}^{(x_{t-1}; t \ldots t+w-1; x_{t+w-1}^*)}$, and apply the solution $\{\widetilde{x}_t, \ldots, \widetilde{x}_{t+w-2}, x_{t+w-1}^*\}$ to time slots $\{t, \ldots, t+w-1\}$.
- RRHC: At the time slot t, where $t = 1, 2, 3, \ldots$, we solve $\{\mathbf{P_2}^{(t)}, \ldots, \mathbf{P_2}^{(t+w-1)}\}$, and if $\mathbf{P_2}^{(t)}, \ldots, \mathbf{P_2}^{(t+w-2)}$ have been solved previously, we only solve $\mathbf{P_2}^{(t+w-1)}$ at the current time slot. We get the solution $\{x_t^*, \ldots, x_{t+w-1}^*\}$. We keep x_{t+w-1}^*, solve $\mathbf{P_1}^{(x_{t-1}; t \ldots t+w-1; x_{t+w-1}^*)}$, obtain the solution $\{\widetilde{x}_t, \ldots, \widetilde{x}_{t+w-2}, x_{t+w-1}^*\}$, and only apply $\widetilde{x}_t$ to the time slot t.

RFHC and RRHC are guaranteed to produce a solution whose cost over time is no larger than our online algorithm that does not use prediction. To exhibit it formally, we firstly show Lemma 3, based on which, we then prove Theorem 4.

Lemma 3: Given a feasible solution $\{x_1, \ldots, x_T\}$ to $\mathbf{P_1}$ and integers τ, κ, where $1 \leqslant \tau < \kappa \leqslant T$, we have this inequality hold: $\mathbf{P_1}(x_0; x_1, \ldots, x_{\tau-1}, \widetilde{x}_\tau, \widetilde{x}_{\tau+1}, \ldots, \widetilde{x}_{\kappa-1}, x_\kappa, \ldots, x_T) \leqslant \mathbf{P_1}(x_0; x_1 \ldots x_T)$, *where* $\{\widetilde{x}_\tau, \widetilde{x}_{\tau+1}, \ldots, \widetilde{x}_{\kappa-1}, x_\kappa\}$ *minimizes the problem* $\mathbf{P_1}^{(x_{\tau-1}; \tau \ldots \kappa; x_\kappa)}$.

Proof: Because $\{\widetilde{x}_\tau, \widetilde{x}_{\tau+1}, \ldots, \widetilde{x}_{\kappa-1}, x_\kappa\}$ optimally solves $\mathbf{P_1}^{(x_{\tau-1}; \tau \ldots \kappa; x_\kappa)}$, we have this: $\mathbf{P_1}(x_{\tau-1}; \widetilde{x}_\tau, \ldots, \widetilde{x}_{\kappa-1}, x_\kappa) \leqslant \mathbf{P_1}(x_{\tau-1}; x_\tau \ldots x_\kappa)$. Further, we complete the proof by adding $\mathbf{P_1}(x_0; x_1 \ldots x_{\tau-1}) + \mathbf{P_1}(x_\kappa; x_{\kappa+1} \ldots x_T)$ to both sides of this inequality. □

Theorem 4: $COST_{RFHC} \leqslant COST_{\{\mathbf{P_2}^{(t)}, \forall t\}}$, $COST_{RRHC} \leqslant COST_{\{\mathbf{P_2}^{(t)}, \forall t\}}$.

Proof: $COST_{\{\mathbf{P_2}^{(t)}, \forall t\}} = \mathbf{P_1}(x_0; x_1^* \ldots x_T^*)$. By choosing the solution $\{x_1^*, \ldots, x_T^*\}$ as the feasible solution, we iteratively apply Lemma 3 to prove this theorem.

For RFHC, by setting $\tau = (n-1)w + 1$, $\kappa = nw$, $n = 1, 2, \ldots$, we have $\mathbf{P_1}(x_0; \widetilde{x}_1, \ldots, \widetilde{x}_{w-1}, x_w^*, \widetilde{x}_{w+1}, \ldots, \widetilde{x}_{2w-1}, x_{2w}^*, \ldots) \leqslant \mathbf{P_1}(x_0; x_1^* \ldots x_T^*)$, *i.e.*, we have the first part of the theorem.

For RRHC, we use $\widetilde{x}_t^{(n)}$, $n \leqslant t < n + w - 1$ to denote the solution value for the time slot t by solving the problem $\mathbf{P_1}^{(x_{n-1}^{(n-1)}; n \ldots n+w-1; x_{n+w-1}^*)}$, where $x_0^{(0)} \triangleq x_0$. We set $\tau = n$, $\kappa = n + w - 1$, $n = 1, 2, \ldots$, and we can get a series of inequalities as follows. When $\tau = 1$, $\kappa = w$, we have $\mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \ldots, \widetilde{x}_{w-1}^{(1)}, x_w^*, \ldots, x_T^*) \leqslant \mathbf{P_1}(x_0; x_1^* \ldots x_T^*)$; when $\tau = 2$, $\kappa = w + 1$, we have $\mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \widetilde{x}_2^{(2)}, \ldots, \widetilde{x}_w^{(2)}, x_{w+1}^*, \ldots, x_T^*) \leqslant \mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \ldots, \widetilde{x}_{w-1}^{(1)}, x_w^*, \ldots, x_T^*)$; when $\tau = 3$, $\kappa = w + 2$, we have $\mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \widetilde{x}_2^{(2)}, \widetilde{x}_3^{(3)}, \ldots, \widetilde{x}_{w+1}^{(3)}, x_{w+2}^*, \ldots, x_T^*) \leqslant \mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \widetilde{x}_2^{(2)}, \ldots, \widetilde{x}_w^{(2)}, x_{w+1}^*, \ldots, x_T^*)$; and so on. Note that the right-hand side of the inequality when $\tau = n + 1$ is always the same as the left-hand side of the inequality when $\tau = n$. Through the chain of all inequalities we eventually have $\mathbf{P_1}(x_0; \widetilde{x}_1^{(1)}, \widetilde{x}_2^{(2)}, \widetilde{x}_3^{(3)}, \ldots, \widetilde{x}_T^{(T)}) \leqslant \mathbf{P_1}(x_0; x_1^* \ldots x_T^*)$, *i.e.*, we prove the second part of the theorem. □

V. Numerical Evaluations

We evaluate our algorithms using real-world data traces. Having proved the worst-case guarantees, we investigate the performances of our online and control algorithms in realistic scenarios and compare them with existing 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 the locations of the 48 continental state capitals as the locations of tier-1 clouds. Having the location of each cloud, we use the geographic distance to define the SLAs [9], [17]: 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 λ_{jt}: We select two types of real-world workloads, one with regular dynamics and the other with more bursts, for our evaluations. We use the workload of Wikipedia in October 2007 [21] and the workload of the HTTP servers from April to July 1998 during the World Cup'98 period [3], as in Fig. 4a and 4b, 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. There are 500 hours for Wikipedia. There are 2089 hours for the original World Cup workload; however,

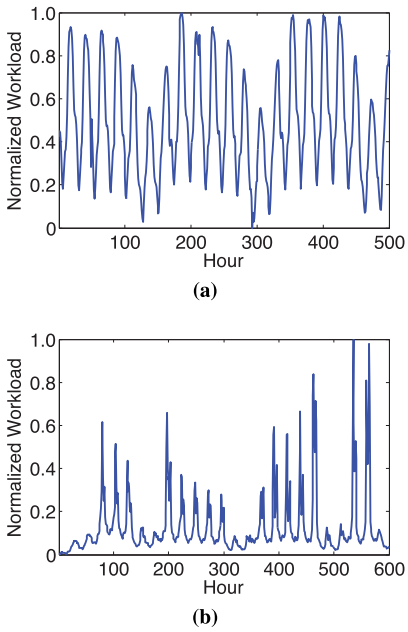


Fig. 4. The time-varying workload. (a) Wikipedia. (b) World Cup.

TABLE I
ELECTRICITY PRICE STATISTICS [17]

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

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.

Operating Price a_{it}, c_{ijt}: We use energy and WAN bandwidth prices respectively, which are reported among the most significant operating expenses 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 [17]. 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 [18].

Cloud WAN bandwidth price is estimated based on network capacity [16], [25]. 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 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 [13], [16]. 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 cloud to the capacity of the incident tier-2 cloud.

Algorithms: For prediction-free algorithms, we compare the following: (1) the sequence of one-shot optimizations, which solves the one-shot slice of $\mathbf{P_1}$ at every time slot; (2) our proposed online algorithm, which solves $\mathbf{P_2^{(t)}}$ at every time slot; (3) the online algorithm that we call LCP-M, which, at every time slot, solves both $\mathbf{P}_1^{(x_0;1\ldots t;)}$ and a related problem with the reconfiguration cost reverse in time and then applies the lazy capacity principle to every variable in our problem, following the design of the LCP(0) algorithm [12]; (4) the offline optimum, which solves $\mathbf{P_1}$ with accurate knowledge of the operating price and the workload in the entire future.

For predictive algorithms, we compare FHC with RFHC, and RHC with RRHC, when predictions about the operating price a_{it}, $\forall i$ and the workload λ_{jt}, $\forall j$ are available. We evaluate both accurate and inaccurate predictions.

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.

Other Parameters ε, ε', k, w: We set $\varepsilon = \varepsilon'$, where ε, $\varepsilon' > 0$ are parameters of our online algorithm, and vary ε 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. k, the number of the closest clouds chosen by every tier-1 cloud, is set as 1, 2, 3, and 4 to show how the variation of SLA may affect the results. w, the length of the prediction

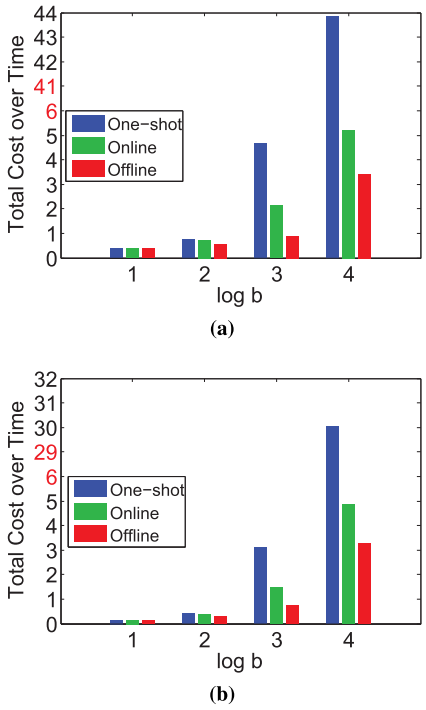


Fig. 5. Cost for different reconfiguration prices. (a) Wikipedia. (b) World Cup.

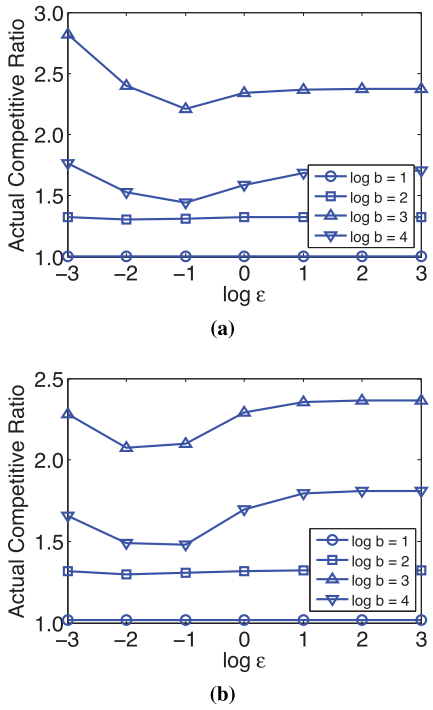


Fig. 6. Actual competitive ratio. (a) Wikipedia. (b) World Cup.

window, is set as 2, 4, 6, 8, and 10 to evaluate how the amount of future information may affect the results.

Prediction Error: To test the robustness of our predictive algorithms under noisy predictions, we inject zero-mean Gaussian noise into a_{it} and λ_{jt}, while setting the standard deviation of such noise as a percentage (*i.e.*, the prediction error) of the mean of the corresponding a_{it} and λ_{jt} over time. We vary the prediction error up to 15%.

C. Results Without Prediction

Fig. 5 visualizes the normalized total cost over time when the cloud and network resources are allocated and reconfigured by the sequence of 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 achieves a 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 our algorithm behaves consistently well for the two types of workloads.

Fig. 6 shows how the "actual" competitive ratio, *i.e.*, the ratio of the total cost incurred by our online algorithm over that incurred by the offline optimal solution in practice, varies

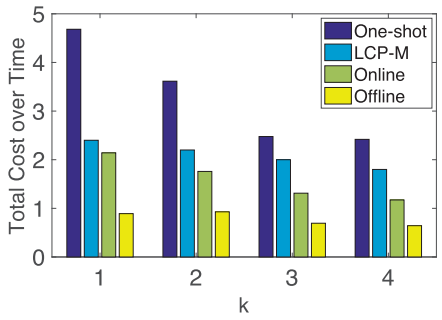


Fig. 7. Cost for different SLAs.

with the algorithmic parameter ε for the two workloads. In this figure, we set $k = 1$. First and overall, this ratio is reasonably good for both workloads, as it is always below 3. Second, 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. 5). Third, the curve of the actual competitive ratio has a valley. Note that our worst-case theoretical competitive ratio always decreases as ε grows, but this figure implies that, in practice, a lower ε may achieve a lower actual competitive ratio.

Fig. 7 investigates the performance of the algorithms for the Wikipedia workload 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. 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. Note that we also show LCP-M in this figure. The reason that it does

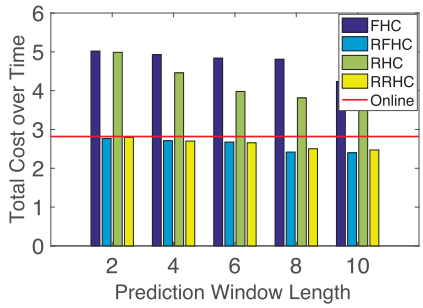


Fig. 8. Cost under accurate predictions.

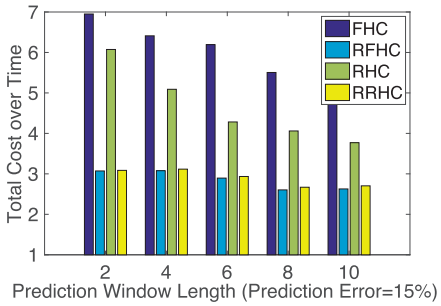


Fig. 9. Cost under inaccurate predictions.

not behave as well as our online algorithm may be partially ascribed to that the lazy capacity principle, as originally derived for the single-cloud case, may not properly hold for the multi-cloud or the multi-tier-cloud case. See Section VI for more discussion.

D. Results Under Accurate and Inaccurate Predictions

Fig. 8 compares the normalized total cost over time of different algorithms with different prediction window lengths for the Wikipedia workload, when accurate predictions are available. In this figure we set $b = 10^3$, $\varepsilon = 10^{-3}$, and $k = 1$. Our online algorithm does not use prediction and is thus drawn as a horizontal line. We observe that, as we proved previously, both regularized control algorithms RFHC and RRHC, with the help of prediction, are always better than our online algorithm; however, prediction is not able to help standard control algorithms FHC and RHC to beat our online algorithm and regularized control algorithms. The reason, as we explained before, can be ascribed to the fact that the prediction window length is smaller than the length of the ramp down phase of the workload. In fact, in our Wikipedia workload, about 40% of all the ramp down phases have a length larger than 10 time slots. Even using a prediction window of 10 time slots, as what we do in this figure, these ramp down phases still cause FHC and RHC to follow the workload and weaken their performance. This figure thus verifies that, when we only have limited predicted information compared with the ramp down phase of the workload, using such information in our regularized control algorithms results in better performance than using it in standard control algorithms.

Figs. 9 and 10 focus on the normalized total cost over time of the predictive algorithms with different prediction window

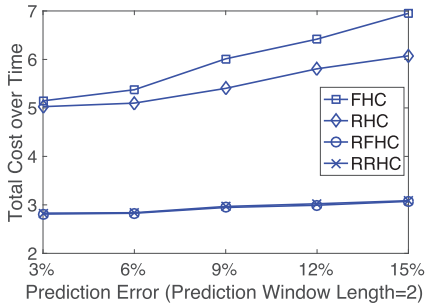


Fig. 10. Influence of prediction error.

lengths and different prediction error rates for the Wikipedia workload, when the predictions about the operating price and the workload are available but inaccurate. In this figure we set b, ε, and k the same as in Fig. 8. Fig. 9 takes the 15% prediction error rate as an example, and confirms that RFHC and RRHC are still much better than FHC and RHC, although all algorithms have worse performance than in Fig. 8, due to the noise in the predictions. We have actually observed this phenomenon consistently for all the other prediction error rates under our tests. Fig. 10 varies the prediction error rate using a prediction window of 2 time slots. RFHC and RRHC are quite robust to prediction errors, because they grow negligibly in the total cost as the prediction error increases; in contrast, FHC and RHC grow much faster, about 40% and 20% more, in the total cost as the prediction error goes up to 15%. Compared to Fig. 8, one interesting thing to note is that, when the length of the prediction window is small, RFHC and RRHC under noisy predictions are even worse than our online algorithm using no prediction.

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.* [13] 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.* [27] 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.* [11], [12] might be the first 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 (LCP) online algorithm for the single-cloud case [12] and the Averaging Fixed Horizon Control (AFHC) algorithm for the multi-cloud case [11]. Zhang *et al.* [29] 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.* [28] developed the randomized fixed horizon control to route big data from sources to selected data centers for aggregative processing and Wu *et al.* [24] exploited Lyapunov optimization to distribute social media to clouds to meet the dynamic demands, where in both cases the reconfiguration cost was caused by data movement across locations. Lu *et al.* [14] connected the dynamic server provisioning problem to the classic ski-rental problem and proposed both online and predictive algorithms with competitive guarantees.

We highlight the reasons why existing research falls insufficient for our problem and how our work makes a difference. The first category of existing work does not consider the reconfiguration cost, *i.e.*, switching from one decision to another is assumed free. It is often difficult to directly adapt such reconfiguration-oblivious approaches to accounting for reconfigurations while still providing performance guarantees, which also motivates us to rethink the problem and develop new algorithms from a different angle. The second category of existing work accounts for the reconfiguration cost, but is unable to address all the challenges that we have addressed in this paper. For example, LCP [12] is reported to be unable to be generalized to the multi-cloud case with a guaranteed competitive ratio, and AFHC [11], while applicable to multiple clouds, may always require predictions. As another example, the ski-rental-based online algorithm [14] may be applicable to our case, but its break-even algorithmic idea may lead to an unbounded competitive ratio as the resource price in our case translates into an arbitrarily time-varying "rental" price, unlike the constant rental price in the classic ski-rental problem; the indivisible and continuous amount of resources in our problem may also be an obstacle, as the connection to the ski-rental problem requires the resources to be discrete in the first place. Besides, even when previous work is applicable, as in the case of the prediction-based control algorithms FHC and RHC, they lack worst-case performance guarantees for our problem as proved in this paper. Our work fundamentally differs from all existing work in that it targets the scenario of multiple tiers of multiple clouds, and it takes a regularization-based approach to design a prediction-free online algorithm and two prediction-aware control algorithms, all with competitive guarantees.

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 hierarchical structure. In this paper, we take a regularization-based method to design dynamic algorithms for the scenarios with and without available prediction, respectively. We overcome the challenge stemming from reconfiguration-induced, coupled decisions by constructing a series of subproblems, each of which is solvable at the corresponding time slot by only taking the solution of the previous time slot and the workload and resource price of the current time slot as inputs. We prove that this algorithm produces a solution with a parameterized competitive ratio for arbitrarily dynamic workloads and operating prices. We also design novel control algorithms that leverage predictions, while providing the theoretical performance guarantee for the worst case. We prove the lack of such a guarantee for existing control algorithms. Evaluations based on real-world data confirm that our algorithms perform well in practice and are superior to existing algorithms.

References

[1] *Amazon EC2 Pricing*, accessed on Dec. 11, 2016. [Online]. Available: http://aws.amazon.com/ec2/pricing/

[2] *AT&T's 38 Global Internet Data Centers*, accessed on Dec. 11, 2016. [Online]. Available: http://www.business.att.com/content/productbrochures/eb_idcmap.pdf

[3] M. Arlitt and T. Jin. (1998). *World Cup Web Site Access Logs*. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[4] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive analysis via regularization," in *Proc. ACM-SIAM SODA*, Jan. 2014, pp. 436–444.

[5] M. C. Calzarossa, L. Massari, and D. Tessera, "Workload characterization: A survey revisited," *ACM Comput. Surv.*, vol. 48, no. 3, p. 48, Feb. 2016.

[6] B. Chandramouli, J. Claessens, S. Nath, I. Santos, and W. Zhou, "RACE: Real-time applications over cloud-edge," in *Proc. ACM SIGMOD*, 2012, pp. 625–628.

[7] F. Hao, M. Kodialam, T. Lakshman, and S. Mukherjee, "Online allocation of virtual machines in a distributed cloud," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 10–18.

[8] X. Hu, A. Ludwig, A. Richa, and S. Schmid, "Competitive strategies for online cloud resource allocation with discounts: The 2-dimensional parking permit problem," in *Proc. IEEE ICDCS*, Jun./Jul. 2015, pp. 93–102.

[9] L. Jiao, J. Li, W. Du, and X. Fu, "Multi-objective data placement for multi-cloud socially aware services," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 28–36.

[10] L. Jiao, A. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed online resource allocation in multi-tier distributed cloud networks," in *Proc. IEEE IPDPS*, May 2016, pp. 333–342.

[11] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *Proc. IEEE IGCC*, Jun. 2012, pp. 1–10.

[12] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[13] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS*, Jun. 2011, pp. 233–244.

[14] T. Lu, M. Chen, and L. L. H. Andrew, "Simple and effective dynamic provisioning for power-proportional data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1161–1171, Jun. 2013.

[15] M. Mao and M. Humphrey, "A performance study on the VM startup time in the cloud," in *Proc. IEEE CLOUD*, Jun. 2012, pp. 423–430.

[16] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, "To coordinate or not to coordinate? Wide-area traffic management for data centers," Dept. Comput. Sci.,Princeton Univ., Princeton, NJ, USA, Tech. Rep. TR-998-15, 2012.

[17] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009.

[18] 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 *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

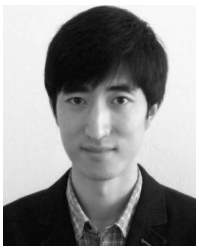
[19] A. Shraer, B. Reed, D. Malkhi, and F. P. Junqueira, "Dynamic reconfiguration of primary/backup clusters," in *Proc. USENIX ATC*, Jun. 2012, pp. 425–437.

[20] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proc. ACM e-Energy*, Jan. 2013, pp. 137–148.

[21] G. Urdaneta, G. Pierre, and M. van Steen, "Wikipedia workload analysis for decentralized hosting," *Comput. Netw.*, vol. 53, no. 11, pp. 1830–1845, 2009.

[22] R. Urgaonkar *et al.*, "Dynamic service migration and workload scheduling in edge-clouds," *Perform. Eval.*, vol. 91, pp. 205–228, Sep. 2015.

[23] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.
[24] Y. Wu *et al.*, "Scaling social media applications into geo-distributed clouds," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 684–692.
[25] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 854–862.
[26] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, Sep. 2015.
[27] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 433–441.
[28] L. Zhang *et al.*, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2710–2721, Dec. 2013.
[29] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *Proc. IEEE ICDCS*, Jun. 2012, pp. 526–535.



Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Göttingen, Germany, in 2014. He was a Researcher with IBM Research, Beijing, China, in 2010. He was a member of Technical Staff, Bell Labs, Dublin, Ireland, from 2014 to 2016. Since 2016, he has been an Assistant Professor with the Department of Computer and Information Science, University of Oregon. His research interests span broadly networking and distributed computing, with a focus on system modeling, algorithm design, and performance evaluation via optimization and control. His recent work has been on cloud data centers, edge computing, and online social networking. His research has been published in the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE INFOCOM, ICNP, IPDPS, and ICDCS. He is a recipient of the Best Paper Award of IEEE LANMAN 2013.

Antonia Maria Tulino (F'13) received the Ph.D. degree in electrical engineering from Seconda Università degli Studi di Napoli, Italy, in 1999. She held research positions with Princeton University, the Center for Wireless Communications, Oulu, Finland, and Università degli Studi del Sannio, Benevento, Italy. Since 2002, she has been an Associate Professor with Università degli Studi di Napoli Federico II. In 2009, she joined Bell Labs. She has been the Principal Investigator of several research projects sponsored by the European Union and the Italian National Council. Her research interests lay in the area of communication systems approached with the complementary tools provided by signal processing, information theory, and random matrix theory. Since 2011, she has been a member of the Editorial Board of the IEEE TRANSACTIONS ON INFORMATION THEORY. She has received several paper awards and among others the 2009 Stephen O. Rice Prize in the field of communications theory for the best paper published in IEEE TRANSACTIONS ON COMMUNICATIONS in 2008. She was selected by the National Academy of Engineering for the Frontiers of Engineering program in 2013.

Jaime Llorca received the B.E. degree in electrical engineering from Universidad Politecnica de Catalunya, Barcelona, Spain, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 2003 and 2008, respectively. He held a post-doctoral position with the Center for Networking of Infrastructure Sensors, College Park, from 2008 to 2010. He joined Nokia Bell Labs, Holmdel, NJ, USA, in 2010, where he is currently a Research Scientist with the Network Algorithms Group. His research interests are in the field of network algorithms, network optimization, distributed control, and network information theory, with special focus on communication networks, distributed cloud networking, and content distribution. He is a recipient of the 2007 Best Paper Award at the IEEE International Conference on Sensors, Sensor Networks and Information Processing, the 2016 Best Paper Award at the IEEE International Conference on Communications, and the 2015 Jimmy H.C. Lin Award for Innovation.

Yue Jin received the Ph.D. degree in industrial engineering and operations research from the University of Massachusetts Amherst. She became a Researcher with Bell Labs, Ireland, in 2008, after spending 1.5 years there as a Post-Doctoral Researcher. She is currently a Researcher with the Advanced Analytics Group, Bell Labs, France. Her main research area is optimization and coordination in service systems and her works include supplier management, service manpower planning, smart data pricing, and cloud resource management. She is constantly involved in internal and external research collaborations. She has also made major contributions in several projects that address practical problems for Nokia business units, including the reconfiguration of regional supply chains, the engineer scheduling in customer delivery units, and product structure optimization.

Alessandra Sala received the Ph.D. degree in computer science from the University of Salerno, Italy. In her prior appointment, she was the Technical Manager of the Data Analytics and Operations Research Group, Bell Labs, Ireland. She held a research associate position with the Department of Computer Science, University of California at Santa Barbara. She was a key contributor of several funded proposals from the National Science Foundation, USA. She focused her research on modeling massive graphs with an emphasis on mitigating privacy threats for online social network users. She worked for two years as a Post-Doctoral Fellow with the CurrentLab Research Group led by Prof. B. Y. Zhao. She is currently the Head of the Bell Labs Analytics Research Group. Her research focus lies on distributed algorithms and complexity analysis with an emphasis on graph algorithms and privacy issues in large networks. In her previous work, she developed efficient distributed systems to support robust and flexible application level services, such as scalable search, flexible data dissemination, and reliable anonymous communication. Her research was awarded with the Cisco Research Award in 2011.