Regret-Optimal Policies for Rental Caches

R Sri Prakash
IIITDM Kancheepuram
sriprakash@iiitdm.ac.in

Nikhil Karamchandani
IIT Bombay
nikhilk@ee.iitb.ac.in

Sharayu Moharir
IIT Bombay
sharayum@ee.iitb.ac.in

***Abstract*—We study the online caching problem in the setting where the service provider has access to a third-party rental cache that it can use via short-term contracts. The cost incurred to run this service is modelled as the sum of the rent cost incurred and the number of requests not served by the cache. We consider both stochastic and adversarial arrivals. Time is divided into slots, and in each time-slot, the algorithmic challenge is two-fold: *(i)* we first decide whether we want to rent the cache; and *(ii)* if we decide to rent, we determine which files to cache. We use regret to evaluate the performance of online caching policies with the optimal static caching policy as the benchmark. We propose a variant of the widely studied Follow the Perturbed Leader policy and show that it has order-optimal regret for both stochastic and adversarial arrivals. We supplement our analytical results with simulations using synthetic and trace-based request arrival sequences.**

I. Introduction

The caching problem has been studied since the 1960s. At the time, the motivation for the problem was memory management in computer systems. Since the early 2000s, there has been a renewed interest in the caching problem, fueled by the increasing popularity of Video-on-Demand (VoD) systems.

Many VoD services use cloud servers to store content and bring it to their customers on demand. This is in part necessitated by the fact that the content catalogs of most popular VoD services are huge and can only be stored on cloud servers given their large storage resources. However, serving users via the cloud leads to latency and thus negatively affects the quality of user experience. A promising solution is to use storage resources at the edge, i.e., close to the end-user to cache popular files. In this case, when a user requests a file that is stored in the cache, the file can be delivered to the user at low latency given the proximity of the cache to the user.

Most of the existing literature has studied the setting where a dedicated cache, albeit of limited capacity, is available for free [1]–[4]. In this work, we consider the setting where a third-party edge cache can be rented by the VoD service to store files via short-term contracts. This setting is motivated by the emergence of edge computing platforms, e.g., Amazon Web Services [5] and Microsoft Azure [6], which provide compute/storage resources for rent. Further, there are proposals to augment cellular base stations [7] as well WiFi access points [8] such that they can be used as storage/compute resources at the edge. It follows that, in our setting, the decision to be made is two-fold. We first decide whether to rent the cache or not, following which, if the cache is rented, we need to decide which files to store in the cache. We consider the setting where both these decisions can be dynamically changed over time.

The performance of caching policies has been evaluated for two kinds of arrival processes. The first, called the Independence Reference model is where arrivals are i.i.d. stochastic, i.e., requests are generated in an i.i.d. manner from an (unknown) distribution. The second setting, called the adversarial arrivals setting makes no structural assumptions on the arrival process. Like [4], we consider both these arrival processes and aim to design a policy that performs well for both types of arrival processes. The performance metric we use is regret [9] where we evaluate the performance of candidate policies against an appropriately defined optimal static policy. Our definition of regret takes into account the cost incurred by the system to rent the cache as well as the cost incurred by the system when the requested file is not present in the cache.

A widely studied online optimization policy is Follow the Perturbed Leader (FTPL) [10]. For the caching problem, FTPL maintains a score for each file in the catalog. This score is the sum of the number of times the file has been requested up to that point and a suitably chosen random perturbation. In each time slot, FTPL caches the files with the highest scores subject to the cache storage capacity constraint. In the classical setting with a dedicated cache, FTPL is known to have order-optimal regret performance for both adversarial [3], [4] and stochastic arrivals [4]. In this work, we build on FTPL to design a policy for our setting where instead of a dedicated cache, we have a cache that can be rented at a cost.

A. Our Contributions

The main contributions of our work are as follows:

- We propose a variant of the FTPL policy called Rent then FTPL (R-FTPL). Like the FTPL policy, R-FTPL maintains a score for each file. In addition to this, R-FTPL also maintains a score for the option of not renting the cache. This score is the sum of the rent cost per time slot and suitably chosen random perturbation. The decision of whether to rent the cache or not and what to store if the cache is rented is made based on these scores.
- We show that R-FTPL has order-optimal regret performance with respect to time for both adversarial and stochastic arrivals.
- Via simulations, we compare the performance of R-FTPL with other candidate policies for both synthetically generated stochastic request arrivals as well as trace-driven arrival sequences.

This work is supported by a SERB grant on Leveraging Edge Resources for Service Hosting. The authors thank Avhishek Chatterjee for his inputs in problem formulation.

B. Related Work

The caching problem is a special case of the widely studied online optimization of prediction with experts advice (PEA). Optimal policies for the PEA setting have prohibitively large complexity for the caching problem which has necessitated customized online algorithms for caching. The Least Frequently Used (LFU), Least Recently Used (LRU), and First-in-First-Out (FIFO) are popular caching policies with optimal performance in terms of the competitive ratio [11] but have poor regret performance for adversarial settings [1]. However, LFU, LRU, and FIFO have order-optimal regret with respect to time for stochastic arrivals [12]. There are gradient-based policies like OGA that have sub-linear regret performance for the adversarial setting [1]. More recently, the FTPL with an appropriate learning rate has been shown to have order-optimal regret performance for the caching problem for both adversarial and stochastic arrivals [4], [13].

In [14]–[18], the focus is on efficiently leveraging third-party edge resources available for rent to host Software as a Service (SaaS) applications. The decision to be made in [14]–[18] is whether or not to host the service at the edge over time. Closest to our work, [15] also aims to minimize regret. The cost incurred by the system is a function of the rental cost incurred, the cost of latency in serving customers when edge resources are not rented, and the cost of fetching and hosting the code/databases of the SaaS to host on the edge server. The policies studied in [15] include Retro-Renting [14] and a suitably adapted version of FTPL.

In our caching problem, we make two decisions in each time-slot. We first decide whether to rent the cache or not, and if we decide to rent, we determine which files to cache. Our performance metric is defined as the sum of the rental cost incurred and the number of requests that the cache cannot serve. Given these differences in the decisions to be made and the performance metric compared to [14]–[18], our results in this work include novel fundamental bounds on the cost incurred by any online algorithm. Further, we build on the analysis for FTPL-like policies in [3], [4], [15] and tailor the arguments to our setting with a rental cache to characterize the performance of our variant of FTPL.

II. System Setup

The system consists of a cache of size C that can be rented at P units per time-slot. In each time-slot, we need to decide if we want to rent the cache or not. The renting decision can be changed dynamically across time. If the cache is rented, we can store C files from a library $\mathcal{L}$ of size L in it. Exactly one file is requested in each time-slot. The sequence of events in a time slot is as follows. We decide whether to rent the cache or not. If the cache is rented, we populate the cache with C files, after which it receives a request for a file from the user. If the requested file is available in the cache, the cache is said to have a *hit* and the request is fulfilled locally by the cache, and otherwise, a *miss*, in which case the file request is fulfilled by the back-end server. If the cache is not rented, the request is served by the back-end server.

A. Cache Configuration

Let $\rho_t \in \{0,1\}$ represent the renting decision in slot t. We define

$$\rho_t = \begin{cases} 1 & \text{if the cache is rented in time-slot } t \\ 0 & \text{otherwise.} \end{cases}$$

The file requested by the user at time t is denoted by x_t and is represented also in the form of the one-hot encoded vector $\boldsymbol{x}_t \in \{0,1\}^L$. For $\tau \geq 2$, we denote by $\boldsymbol{X}_\tau = \sum_{t=1}^{\tau-1} \boldsymbol{x}_t$ the L-length vector storing the cumulative sum of requests for each file till time slot $\tau - 1$. $\boldsymbol{X}_1$ is initialized to be the zero vector.

If the cache is rented in slot t, let $C(t)$ denote the set of files cached in slot t. Let $\mathcal{Y}$ be the set of all possible cache configurations, i.e., the set $\{y \in \{0,1\}^L : \|y\|_1 \leq C\}$. Let $\boldsymbol{y}_t \in \mathcal{Y} \subset \{0,1\}^L$ be a binary vector denoting the state of the cache at time t, such that $\boldsymbol{y}_t = (y_t^1, y_t^2, \ldots, y_t^L)$ with $y_t^i = 1$ for $i \in C(t)$ and 0 otherwise. Let $\mathcal{S}$ be the collection of indices of all possible cache configurations. For example consider $L = 3$, $C = 2$, then $\mathcal{S} = \{\{1,2\},\{2,3\}\{1,3\}\}$ and $\mathcal{Y} = \{[1,1,0]^{\mathrm{t}}, [0,1,1]^{\mathrm{t}}, [1,0,1]^{\mathrm{t}}\}$, where $[1,1,0]^{\mathrm{t}}$ represents the transpose of $[1,1,0]$. We define

$$\boldsymbol{y}'_t = \begin{cases} \begin{bmatrix} \boldsymbol{y}_t \\ 1 \end{bmatrix} & \text{if } \rho_t = 1, \\ \boldsymbol{0} & \text{if } \rho_t = 0. \end{cases}$$

Note that $\boldsymbol{y}'_t$ is a $L+1-$dimensional binary vector denoting the state of the system at time t. Let $\mathcal{Y}' \subset \{0,1\}^{L+1}$ represent the collection of all possible $\boldsymbol{y}'_t$ vectors.

B. File Requests and Caching Policy

We consider two types of file requests: adversarial and stochastic. The file requests are said to be *adversarial* if no assumptions are made regarding the statistical properties of the file requests. We assume that the adversary is oblivious, i.e., the entire file request sequence is fixed before the first request is received by the cache. The file requests are said to be *stochastic* if, in each time-slot, the request is generated independently according to a popularity distribution $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_L)^{\mathrm{t}}$, where $\mathbb{P}(x_t = i) = \mu_i$ and $\sum_i \mu_i = 1$. Without loss of generality, we assume that $\mu_1 \geq \ldots \geq \mu_L$.

At the beginning of any given time slot $t \geq 2$, a caching policy $\pi'(\cdot)$ maps the history of observations it has seen so far (denoted by $h'(t)$) to a valid $\boldsymbol{y}'_t$, i.e., $\boldsymbol{y}'_t = \pi'(h'(t))$.

C. Reward and Rent Cost

If the policy rents the cache in a slot, it obtains a reward of 1 unit when the requested file is available in the cache, i.e., a hit occurs, and a reward of 0 units otherwise. A rent cost of P units is incurred. We focus on the setting when $P < 1$ because if $P \geq 1$ then the optimal strategy is to never rent the cache. Further, if the cache is not rented in a slot, then the reward obtained is 0 irrespective of the requested file, and no rent cost is incurred. The net utility in a slot is defined as the difference between the reward obtained and the rent cost incurred.

Let $\boldsymbol{x}_t' = \begin{bmatrix} \boldsymbol{x}_t \\ -P \end{bmatrix}$, and for $\tau \geq 2$ we denote $\boldsymbol{X}_\tau' = \sum_{t=1}^{\tau-1} \boldsymbol{x}_t'$, the $(L+1)-$dimensional vector of the cumulative sum of requests for each file till time slot $\tau - 1$ along with rent cost till $\tau - 1$. Then, $\langle \boldsymbol{y}_t', \boldsymbol{x}_t' \rangle$ represents the net utility obtained in slot t, and

$$\langle \boldsymbol{y}_t', \boldsymbol{x}_t' \rangle = \begin{cases} \langle \boldsymbol{y}_t, \boldsymbol{x_t} \rangle - P & \text{for } \rho_t = 1, \\ 0 & \text{for } \rho_t = 0. \end{cases}$$

D. Performance Metric

Policies are evaluated on the basis of the regret that they incur. We define T to be the time horizon of interest. Informally, the regret of a policy till time T is the difference between the net utility of the optimal static policy and the net utility the policy under consideration.

Adversarial Arrivals: When following a policy π' on a request sequence $\{x_t\}_{t=1}^T$, the regret till time T for $\{x_t\}_{t=1}^T$ when the file requests are adversarial is defined as:

$$\mathcal{R}_A^{\pi'}(\{x_t\}_{t=1}^T, T) = \sup_{\boldsymbol{y} \in \mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{X}_{T+1}' \rangle - \sum_{t=1}^{T} \mathbb{E}[\langle \boldsymbol{y}_t', \boldsymbol{x}_t' \rangle],$$

where the expectation is with respect to any randomness introduced by the policy. The regret of a policy π' till time T is defined as the worst-case regret over all possible request sequences, i.e.,

$$\mathcal{R}_A^{\pi'}(T) = \sup_{\{x_t\}_{t=1}^T} \mathcal{R}_A^{\pi'}(\{x_t\}_{t=1}^T, T).$$

Stochastic Arrivals: Let $\boldsymbol{\mu}' = \mathbb{E}[\boldsymbol{x}_t'] = \begin{bmatrix} \boldsymbol{\mu} \\ -P \end{bmatrix}$ for stochastic file requests. When the file requests are stochastic, the regret till time T is defined as:

$$\begin{aligned} \mathcal{R}_S^{\pi'}(T) &= \sum_{t=1}^{T} \sup_{\boldsymbol{y} \in \mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{\mu}' \rangle - \mathbb{E}\left[\sum_{t=1}^{T} \langle \boldsymbol{y}_t', \boldsymbol{\mu}' \rangle\right] \\ &= \sum_{t=1}^{T} \left(\sum_{i \in \mathcal{C}} \mu_i - P\right) - \mathbb{E}\left[\sum_{t=1}^{T} \langle \boldsymbol{y}_t', \boldsymbol{\mu}' \rangle\right], \end{aligned}$$

where $\mathcal{C}$ denotes the set of files having the top C popularities. In the above expression, the expectation is taken with respect to any randomness introduced by the policy.

Our goal is to design caching policies with order-optimal regret performance for both adversarial and stochastic arrivals.

III. Policies

Any policy in our setting first needs to decide whether it wants to rent the cache in a time-slot. Following this, if the cache is rented, the next decision to be made is to select C files to store in the cache. We present two policies.

A. Rent then LFU (R-LFU)

R-LFU keeps track of the number of times each file has been requested so far. In each time slot t, if the sum of requests of files with the C highest number of requests is greater than the total rent cost till t, then the cache is rented. If the cache is rented in a slot, then, under R-LFU, the C files that have been requested most frequently are stored in the cache. The formal definition of R-LFU can be found in Algorithm 1.

Algorithm 1 R-LFU algorithm

procedure R-LFU(T)
  $\boldsymbol{X}_1' \leftarrow \boldsymbol{0}$
  for $t = 1$ to T **do**
    $\boldsymbol{y}_t' \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{X}_t' \rangle$
    Receive file request x_t
    $\boldsymbol{X}_{t+1}' = \boldsymbol{X}_t' + \boldsymbol{x}_t'$
  end for
end procedure

B. Rent then FTPL (R-FTPL)

R-FTPL is a variation of the R-LFU algorithm and also keeps track of the number of times each file has been requested so far, but adds an independent Gaussian perturbation with mean 0 and standard deviation η_t (referred to as the learning rate) to the counts of each file in each time slot. The sum is referred to as the score of that file. In each time slot t, if the sum of the scores of files with the C highest scores is greater than the suitably perturbed rent cost, then the cache is rented. If the cache is rented in a slot, then, under R-FTPL, the C files with the C highest scores are stored in the cache. The formal definition of R-FTPL can be found in Algorithm 2.

Algorithm 2 R-FTPL algorithm

procedure R-FTPL(T)
  Set $\boldsymbol{X}_1' \leftarrow \boldsymbol{0}$
  Sample $\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{L \times L})$
  for $t = 1$ to T **do**
    $\boldsymbol{y}_t' \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{X}_t' + \eta_t \boldsymbol{\gamma} \rangle$
    Receive file request x_t
    $\boldsymbol{X}_{t+1}' = \boldsymbol{X}_t' + \boldsymbol{x}_t'$
  end for
end procedure

IV. Main Results

In this section, we present our analytical results.

A. Adversarial Requests

In this section, we discuss the performance of the policies introduced in Section III under adversarial file requests. The key results of this section have been summarized in the following theorem:

Theorem 1. *Under adversarial requests, we have*

(a) For any policy π' and $L \geq 2C$,

$$\mathcal{R}_A^{\pi'}(T) = \Omega(\sqrt{T}).$$

(b) The regret of R-LFU policy for $P \leq \frac{C}{C+1}$ can be characterized as:

$$\mathcal{R}_A^{\text{R-LFU}}(T) = \mathcal{O}(T).$$

(c) The regret of R-FTPL with $\eta_t = \alpha\sqrt{t/(C+1)}$ is upper bounded as:

$$\mathcal{R}_A^{\text{R-FTPL}}(T) \leq \alpha(1+\sqrt{T})\sqrt{2(C+1)\log L} + \frac{(1+P)^2}{\alpha\sqrt{\pi}}\sqrt{2T(C+1)}.$$

The proof of the theorem can be found in Section VI.

Part (a) provides a lower bound on the regret of any policy under adversarial requests.

Part (b) provides an $\mathcal{O}(T)$ upper bound on the regret of R-LFU policy under adversarial requests for $P \leq \frac{C}{C+1}$.

Part (c) provides an $\mathcal{O}(\sqrt{T})$ upper bound on the regret of the R-FTPL policy under adversarial requests, thus showing that this algorithm is order-optimal under adversarial requests.

B. Stochastic Requests

In this section, we discuss the performance of the policies under stochastic file requests. The key results of this section have been summarized in the following theorem:

Theorem 2. *The file requests are stochastic i.i.d. with the popularity distribution $\boldsymbol{\mu}$, and let $\Delta' = |\sum_{i=1}^{C}\mu_i - P|$ then,*

(a) Regret of R-LFU:

(i) The regret of R-LFU for $P < \sum_{i=1}^{C}\mu_i$ is upper bounded as:

$$\mathcal{R}_S^{\text{R-LFU}}(T) \leq R_S^{\text{LFU}}(T) + \frac{2}{\Delta'},$$

where $R_S^{\text{LFU}}(T)$ is the regret of the R-LFU policy when rent cost is zero.

(ii) The regret of R-LFU for $P > \sum_{i=1}^{C}\mu_i$ is upper bounded as:

$$\mathcal{R}_S^{\text{R-LFU}}(T) \leq P\binom{L}{C}\frac{2}{\Delta'^2}.$$

(b) Regret of R-FTPL:

(i) The regret of R-FTPL with $\eta_t = \alpha\sqrt{t/(C+1)}$ for $P < \sum_{i=1}^{C}\mu_i$ is upper bounded as:

$$\mathcal{R}_S^{\text{R-FTPL}}(T) \leq R_S^{\text{FTPL}}(T) + \frac{8\alpha^2+2}{\Delta'},$$

where $R_S^{\text{FTPL}}(T)$ is the regret of the R-FTPL policy when rent cost is zero.

(ii) The regret of R-FTPL with $\eta_t = \alpha\sqrt{t/(C+1)}$ for $P > \sum_{i=1}^{C}\mu_i$ is upper bounded as:

$$\mathcal{R}_S^{\text{R-FTPL}}(T) \leq P\binom{L}{C}\frac{8\alpha^2+2}{\Delta'^2}.$$

The proof of the theorem can be found in Section VI.

Part $(a)(i)$ shows that the regret of R-LFU is upper bounded by the sum of the regret of LFU with no rent cost and a constant which is a function of the rent cost, when $P < \sum_{i=1}^{C}\mu_i$. Therefore using Theorem 2 of [4] we get that the regret of R-LFU is $\mathcal{O}(1)$ for $P < \sum_{i=1}^{C}\mu_i$. Part $(b)(ii)$ shows that the regret of R-LFU is $\mathcal{O}(1)$ when $P > \sum_{i=1}^{C}\mu_i$. Thus the regret of R-LFU is $\mathcal{O}(1)$.

Part $(b)(i)$ shows that the regret of R-FTPL is upper bounded by the sum of the regret of FTPL with no rent cost and a constant which is a function of the rent cost, when $P < \sum_{i=1}^{C}\mu_i$. Therefore using Theorem 2 of [4] we get that the regret of R-FTPL is $\mathcal{O}(1)$ for $P < \sum_{i=1}^{C}\mu_i$. Part $(b)(ii)$ shows that the regret of R-FTPL is $\mathcal{O}(1)$ when $P > \sum_{i=1}^{C}\mu_i$. Thus the regret of R-FTPL is $\mathcal{O}(1)$. It follows that R-FTPL is order-optimal for this setting.

V. Numerical Experiments

We now present our simulation results for the policies discussed in this Section III. In addition to R-FTPL and R-LFU, we also simulate FTPL (a policy that always rents the cache and stores the file with the top C scores) [4] and LFU (a policy that always rents the cache and stores the C files that have been requested most frequently) [4].

We consider $\eta_t = P\sqrt{t/(C+1)}$ for R-FTPL, and $\eta_t = \sqrt{t/C}$ for FTPL in all our experiments.

1) Stochastic file requests: We use $L = 10$, $C = 4$, $P = 0.6$. The popularity distribution used is the Zipf distribution with parameter $\beta = 0.5$.

Fig. 1a shows that under stochastic request sequence, the regret of LFU, FTPL scales linearly with T, while that of R-LFU, R-FTPL is upper bounded by a constant when $P > \sum_{i=1}^{C}\mu_i \approx 0.554$. We observe R-LFU performs better than R-FTPL under stochastic request sequence. In Fig. 1b, we plot only the regret as a function of rent cost P. Note that for smaller values of P, the regret under R-FTPL and R-LFU is comparable to the regret under FTPL and LFU respectively. As the rent cost increases, R-LFU and R-FTPL do not rent the cache which leads to lower regret compared to LFU and FTPL which always rent the cache.

2) Adversarial file requests: We consider a synthetic adversarial request sequence in Fig. 2a and a real-world trace in Fig. 2b. The synthetic adversarial request sequence used is $1, 2, 1, 2, \ldots$ for $L = 2$, $C = 1$ and $P = 0.4$ i.e., a round-robin request sequence. The real-world trace used is the first 20,000 rows of the MovieLens 1M dataset [19], [20] which contains ratings for 2569 movies with timestamps that we model as requests to a CDN server of library size 2569 and a cache size of 25, $P = 0.1$.

Fig. 2a shows that under the round-robin request sequence, the regret of LFU, R-LFU scales linearly with T, while that of R-FTPL and FTPL scales sublinearly with T. Fig. 2b shows that on the MovieLens dataset, R-LFU and R-FTPL performs better than LFU and FTPL.

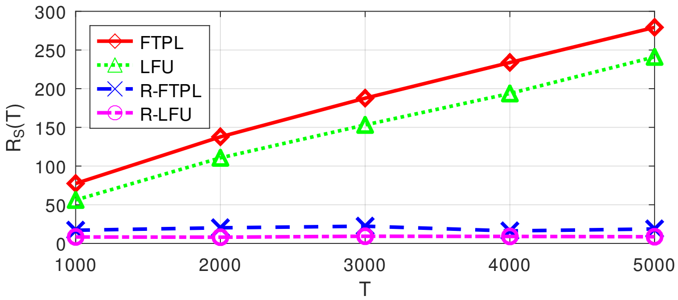
VI. Proofs

A. Proof of Theorem 1(a)

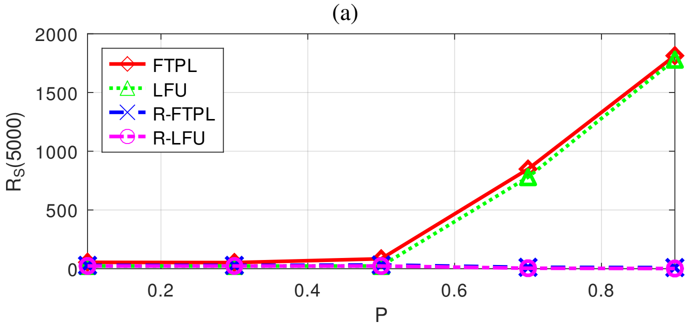
We use the following lemmas to prove Theorem $1(a)$.

Lemma 1. *For $P < \frac{C}{L}$, the regret of any policy under adversarial arrivals is given by*

$$\mathcal{R}_A^{\pi'}(T) = \Omega(\sqrt{T}).$$

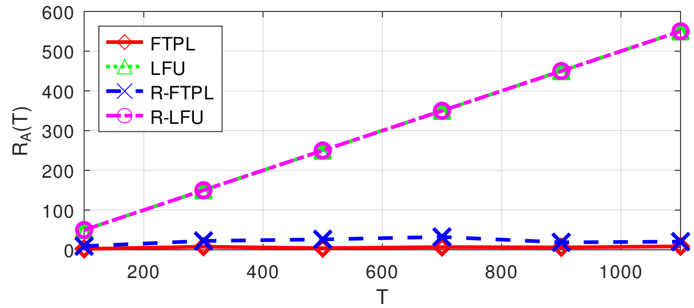


(a)

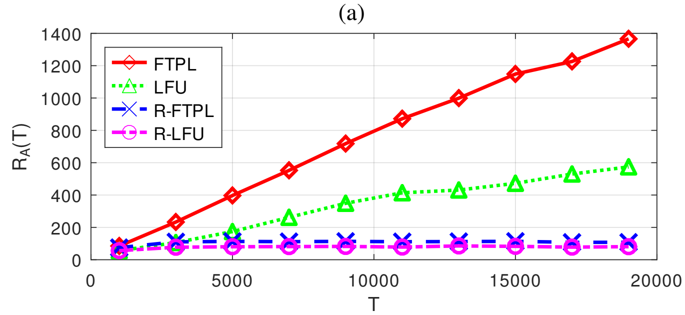


(b)

Fig. 1: (a) We plot regret as a function of T for rent cost $P = 0.6$ for stochastic file requests from a Zipf distribution with parameter 0.5. (b) We plot the regret as a function of rent cost P for stochastic file requests from a Zipf distribution with parameter 0.5 at $T = 5000$.



(a)



(b)

Fig. 2: (a) We plot regret as a function of T for rent cost $P = 0.4$ for a round-robin request sequence with $L = 2$ and $C = 1$. (b) We plot regret for rent cost $P = 0.1$ as a function of T on the MovieLens dataset. From (a), we conclude that the regret with rent cost scales linearly with T for LFU, R-LFU, while R-FTPL and FTPL show better performance. From (b), we see that for $P = 0.1$, the R-LFU and R-FTPL policies perform better than LFU and FTPL policies which always rent the cache.

Proof. Let us consider i.i.d. arrivals with $\boldsymbol{\mu} = (\frac{1}{2C}, \frac{1}{2C}, \cdots, 0, 0, \cdots)$ i.e., only $2C$ files are requested uniformly at random and the rest are not requested at all. Under this request arrival process,

$$\sum_{t=1}^{T} \langle \boldsymbol{y}_t', \mathbb{E}[\boldsymbol{x}_t'] \rangle = \sum_{t=1}^{T} \sum_{i=1}^{L} y_{t,i} \mathbb{E}[x_{t,i}] - PT \\ \stackrel{(a)}{\leq} \sum_{t=1}^{T} C.\frac{1}{2C} - PT = \frac{T}{2} - PT, \quad (1)$$

where (a) follows from the constraint on cache capacity i.e., $\sum_{i=1}^{L} y_{t,i} \leq C$ at for all t. Since $\sup_{\{x_t\}_{t=1}^T} \mathcal{R}_A^{\pi'}(\{x_t\}_{t=1}^T, T) \geq \mathbb{E}_{\{x_t\}_{t=1}^T}[\mathcal{R}_A^{\pi'}(\{x_t\}_{t=1}^T, T)]$ we have:

$$\begin{aligned}
\mathcal{R}_A^{\pi'}(T) &\geq \mathbb{E}[\max_{\boldsymbol{y}\in\mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{X}_T' \rangle\}] - \mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{y}_t', \boldsymbol{x}_t' \rangle] \\
&= \mathbb{E}[\max\{0, \max_{\boldsymbol{y}\in\mathcal{Y}'\setminus\{\boldsymbol{0}\}} \langle \boldsymbol{y}, \boldsymbol{X}_T' \rangle\}] - \sum_{t=1}^{T} \langle \boldsymbol{y}_t', \mathbb{E}[\boldsymbol{x}_t'] \rangle \\
&\geq \mathbb{E}[\max_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{y}, X_T \rangle] - PT - \sum_{t=1}^{T} \langle \boldsymbol{y}_t', \mathbb{E}[\boldsymbol{x}_t'] \rangle \\
&= \mathbb{E}[\max_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{y}, X_T \rangle] - PT - \left(\sum_{t=1}^{T} \langle \boldsymbol{y}_t, \mathbb{E}[\boldsymbol{x}_t] \rangle - PT \right) \\
&\stackrel{(a)}{\geq} \mathbb{E}[\max_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{y}, X_T \rangle] - PT - \left(\frac{T}{2} - PT \right) \\
&\stackrel{(b)}{\geq} \frac{T}{2} + \sqrt{\frac{CT}{2\pi}} - \Theta(1/\sqrt{T}) - PT - \left(\frac{T}{2} - PT \right) \\
&= \Omega(\sqrt{T}),
\end{aligned}$$

where (a) follows from (1), (b) follows from (24) of [2]. ☐

Lemma 2. *For $P \geq \frac{C}{L}$, the regret of any policy under adversarial arrivals is given by*

$$\mathcal{R}_A^{\pi'}(T) = \Omega(\sqrt{T}).$$

Proof. Let us consider i.i.d. arrivals with $\boldsymbol{\mu} = (\frac{P}{C}, \frac{P}{C}, \cdots, \frac{1-P}{L-C}, \frac{1-P}{L-C}, \cdots)$. Under this request arrival process,

$$\sum_{t=1}^{T} \langle \boldsymbol{y}_t', \mathbb{E}[\boldsymbol{x}_t'] \rangle = \sum_{t=1}^{T} \sum_{i=1}^{L} y_{t,i} \mathbb{E}[x_{t,i}] - P \stackrel{(c)}{\leq} \sum_{t=1}^{T} C.\frac{P}{C} - P = 0,$$

where (c) follows from the constraint on cache capacity i.e., $\sum_{i=1}^{L} y_{t,i} \leq C$ at for all t. Therefore,

$$\begin{aligned}
\mathcal{R}_A^{\pi'}(T) &\geq \mathbb{E}[\max\{0, \max_{\boldsymbol{y}\in\mathcal{Y}'} \langle \boldsymbol{y}, \boldsymbol{X}_T' \rangle\}] - \sum_{t=1}^{T} \langle \boldsymbol{y}_t', \mathbb{E}[\boldsymbol{x}_t'] \rangle \\
&\geq \mathbb{E}\left[\max\left\{ 0, \sum_{i=1}^{C} X_{T,i} - PT \right\} \right] - 0
\end{aligned}$$

$$= \mathbb{E}\left[\frac{1}{2}\left(\left|\sum_{i=1}^{C} X_{T,i} - PT\right| + \sum_{i=1}^{C} X_{T,i} - PT\right)\right]$$
$$= \frac{1}{2}\mathbb{E}\left[\left|\sum_{i=1}^{C} X_{T,i} - PT\right|\right] \stackrel{(d)}{=} \Omega(\sqrt{T}),$$

where (d) follows from Lemma 16 of [15]. □

Proof of Theorem 1(a). From Lemma 1 and Lemma 2 we can observe that for any value of $P(< 1)$ and for any policy π'

$$\mathcal{R}_A^{\pi'} = \Omega(\sqrt{T}).$$

□

B. Proof of Theorem 1(b)

Proof of Theorem 1(b). If $R \leq \frac{C}{C+1}$ then, for the request sequence $1, 2, \cdots, C, C+1$ in a round-robin fashion, the optimal policy is to rent the cache and host files 1 to C. R-LFU also always rents the cache and stores the same files as LFU [1]. Therefore it incurs the same regret as LFU in the setting where a dedicated cache is always available at zero cost [1]. Therefore, From Proposition 1 in [1], the result follows. □

C. Proof of Theorem 1(c)

Lemma 3. *Under adversarial arrivals for R-FTPL with learning rate η_t,*

$$\mathcal{R}_A^{\textit{R-FTPL}}(T) \leq (\eta_1 + \eta_T)(C+1)\sqrt{2\log L} + \frac{(1+P)^2}{\sqrt{2\pi}}\sum_{t=1}^{T}\frac{1}{\eta_t}.$$

Proof. The analysis follows arguments used in the proof of Proposition 6.1 of [3] and Theorem 1 of [10]. From the analysis of FTPL in [3], specifically (17) in [3], we have

$$\mathcal{R}_A^{\text{R-FTPL}}(T) \leq \boldsymbol{\Phi}_1(\mathbf{0}) + \sum_{t=2}^{T}\boldsymbol{\Phi}_t(\boldsymbol{X}_t') - \boldsymbol{\Phi}_{t-1}(\boldsymbol{X}_t') + \sum_{t=1}^{T}\frac{1}{2}\langle \boldsymbol{x}_t', \nabla^2\Phi_t(\tilde{\boldsymbol{x}'}_t)\boldsymbol{x}_t'\rangle. \quad (2)$$

We bound each term separately. First using similar analysis to Theorem 1 in [10], we get,

$$\boldsymbol{\Phi}_1(\mathbf{0}) \leq \eta_1\sqrt{2(c+1)\log|\mathcal{Y}'|}.$$

Using analysis similar to [3], [10], the second term can be bounded as follows:

$$\sum_{t=2}^{T}\boldsymbol{\Phi}_t(\boldsymbol{X}_t') - \boldsymbol{\Phi}_{t-1}(\boldsymbol{X}_t') \leq \eta_T\sqrt{2(c+1)\log|\mathcal{Y}'|}.$$

To bound the third term, we define $H(t) = \nabla^2\Phi_t(\tilde{\boldsymbol{x}}_t)$. From [21], we have that $H_{i,j}(t) = \frac{1}{\eta_t}\mathbb{E}[\hat{y}(\tilde{\boldsymbol{x}}_t + \eta_t\gamma)_i\gamma_j]$, where $\hat{y} \in \arg\max_{y\in\mathcal{Y}'}\langle y, z\rangle$. Using a similar analysis to Proposition 4.1 in [3], we get

$$|H_{i,j}(t)| \leq \frac{1}{\eta_t}\sqrt{\frac{2}{\pi}}. \quad (3)$$

If file i is requested at time t i.e., $x_t = i$ then,

$$\begin{aligned}\langle \boldsymbol{x}_t', H(t)\boldsymbol{x}_t'\rangle =& H_{i,i}(t) - PH_{i,N+1}(t) \\ &- PH_{N+1,i}(t) + P^2H_{N+1,N+1}(t) \\ \leq& |H_{i,i}(t)| + P|H_{i,N+1}(t)| \\ &+ P|H_{N+1,i}(t)| + P^2|H_{N+1,N+1}(t)| \\ \stackrel{(e)}{\leq}& (1+P)^2\frac{1}{\eta_t}\sqrt{\frac{2}{\pi}},\end{aligned}$$

where (e) follows from (3). Therefore, the overall regret can be bounded as follows:

$$\begin{aligned}\mathcal{R}_A^{\text{R-FTPL}}(T) \leq& \eta_1\sqrt{2(c+1)\log|\mathcal{Y}'|} + \eta_T\sqrt{2(c+1)\log|\mathcal{Y}'|} \\ &+ \frac{1}{\sqrt{2\pi}}(1+P)^2\sum_{t=1}^{T}\frac{1}{\eta_t} \\ \stackrel{(f)}{\leq}& (\eta_1+\eta_T)(C+1)\sqrt{2\log L} \\ &+ \frac{1}{\sqrt{2\pi}}(1+P)^2\sum_{t=1}^{T}\frac{1}{\eta_t},\end{aligned}$$

where (f) follows from the fact that $|\mathcal{Y}'| = \binom{L}{C} + 1 \leq L^{(C+1)}$. □

Proof of Theorem 1(c). For $\eta_t = \alpha\sqrt{t/(C+1)}$, by Lemma 3 we have,

$$\begin{aligned}\mathcal{R}_A^{\text{R-FTPL}}(T) \leq& \alpha(1+\sqrt{T})\sqrt{2(C+1)\log L} \\ &+ \frac{(1+P)^2}{\alpha\sqrt{\pi}}\sqrt{2T(C+1)}.\end{aligned}$$

□

D. Proof of Theorem 2(a)

Lemma 4. *For any policy π' the expected reward in a slot under stochastic arrivals is given by*

$$\mathbb{E}_{\pi'}[\langle \boldsymbol{y}_t', \boldsymbol{\mu}'\rangle] = (\mathbb{E}_{\pi}[\langle \boldsymbol{y}_t, \boldsymbol{\mu}\rangle] - P)\mathbb{P}(\rho_t = 1),$$

where π represents the policy when $\rho_t = 1$.

Proof. By definition,

$$\begin{aligned}\mathbb{E}_{\pi'}[\langle \boldsymbol{y}_t', \boldsymbol{\mu}'\rangle] =& (\mathbb{E}_{\pi}[\langle \boldsymbol{y}_t, \boldsymbol{\mu}\rangle] - P)\mathbb{P}(\rho_t = 1) \\ &+ 0 \times \mathbb{P}(\rho_t = 0) \\ =& (\mathbb{E}_{\pi}[\langle \boldsymbol{y}_t, \boldsymbol{\mu}\rangle] - P)\mathbb{P}(\rho_t = 1).\end{aligned}$$

□

Lemma 5. *If $P < \sum_{i=1}^{C}\mu_i$ then the regret of any policy π' can be bounded as follows:*

$$\mathcal{R}_S^{\pi'}(T) \leq R_S^{\pi}(T) + \Delta'\sum_{t=1}^{T}\mathbb{P}(\rho_t = 0).$$

Proof. If $P < \sum_{i=1}^{C}\mu_i$ then the optimal policy is to rent the cache and cache top C files in the cache. Then using definition of regret and Lemma 4 we have,

$$\mathcal{R}_S^{\pi}(T) = \sum_{t=1}^{T}\left(\sum_{i=1}^{C}\mu_i - P\right)$$

$$
\begin{aligned}
&- \sum_{t=1}^{T} \left(\mathbb{E}_\pi[\langle y_t, \boldsymbol{\mu}\rangle] - P\right) \mathbb{P}\left(\rho_t = 1\right) \\
=& \sum_{t=1}^{T} \left(\sum_{i=1}^{C} \mu_i - \mathbb{E}_\pi[\langle y_t, \boldsymbol{\mu}\rangle]\right) \mathbb{P}(\rho_t = 1) \\
&+ \sum_{t=1}^{T} \left(\sum_{i=1}^{C} \mu_i - P\right) \mathbb{P}(\rho_t = 0) \\
\leq& \sum_{t=1}^{T} \left(\sum_{i=1}^{C} \mu_i - \mathbb{E}_\pi[\langle y_t, \boldsymbol{\mu}\rangle]\right) + \Delta' \sum_{t=1}^{T} \mathbb{P}(\rho_t = 0) \\
=& R_S^\pi(T) + \Delta' \sum_{t=1}^{T} \mathbb{P}(\rho_t = 0).
\end{aligned}
$$

□

Lemma 6. *If $P > \sum_{i=1}^{C} \mu_i$ then the regret of any policy π' can be bounded as follows:*

$$
\mathcal{R}_S^\pi(T) \leq P \sum_{t=1}^{T} \mathbb{P}(\rho_t = 1).
$$

Proof. If $P > \sum_{i=1}^{C} \mu_i$ then the optimal policy is to not rent the cache and the reward obtained by the optimal policy is 0.

$$
\begin{aligned}
\mathcal{R}_S^\pi(T) =& 0 - \sum_{t=1}^{T} (\mathbb{E}_\pi[\langle y_t, \boldsymbol{\mu}\rangle] - P) \mathbb{P}(\rho_t = 1) \\
=& \sum_{t=1}^{T} (P - \mathbb{E}_\pi[\langle y_t, \boldsymbol{\mu}\rangle]) \mathbb{P}(\rho_t = 1) \leq P \sum_{t=1}^{T} \mathbb{P}(\rho_t = 1).
\end{aligned}
$$

□

Lemma 7. *Let $\mathcal{E}_L$ be the event that $\sum_{i=1}^{C} \hat{\mu}_i(t) > \sum_{i=1}^{C} \mu_i - \Delta'/2$, then we have*

$$
\mathbb{P}(\mathcal{E}_L^c) \leq \exp\left(-\Delta'^2 t/2\right),
$$

where $\hat{\mu}_i(t) = X_t^i/t$.

Proof. By definition,

$$
\begin{aligned}
\mathbb{P}(\mathcal{E}_L^c) =& \mathbb{P}\left(\sum_{i=1}^{C} \hat{\mu}_i(t) < \sum_{i=1}^{C} \mu_i - \Delta'/2\right) \\
=& \mathbb{P}\left(\sum_{i=1}^{C} \mu_i - \hat{\mu}_i(t) > \Delta'/2\right) \overset{(g)}{\leq} \exp\left(-\Delta'^2 t/2\right),
\end{aligned}
$$

where (g) follows from Hoeffding's inequality [22]. □

Lemma 8. *Let $\mathcal{E}_U$ be the event that $\max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t) < \sum_{i=1}^{C} \mu_i + \Delta'/2$, then we have*

$$
\mathbb{P}(\mathcal{E}_U^c) \leq \binom{L}{C} \exp\left(-\Delta'^2 t/2\right),
$$

where $\hat{\mu}_i(t) = X_t^i/t$.

Proof. By definition,

$$
\begin{aligned}
\mathbb{P}(\mathcal{E}_U^c) =& \mathbb{P}\left(\max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t) > \sum_{i=1}^{C} \mu_i + \Delta'/2\right) \\
&\leq \sum_{S \in \mathcal{S}} \mathbb{P}\left(\sum_{i \in S} \hat{\mu}_i(t) > \sum_{i=1}^{C} \mu_i + \Delta'/2\right) \\
&\leq \sum_{S \in \mathcal{S}} \mathbb{P}\left(\sum_{i \in S} \hat{\mu}_i(t) - \mu_i > \sum_{i=1}^{C} \mu_i - \sum_{i \in S} \mu_i + \Delta'/2\right) \\
&\leq \sum_{S \in \mathcal{S}} \mathbb{P}\left(\sum_{i \in S} \hat{\mu}_i(t) - \mu_i > \Delta'/2\right) \\
&\overset{(h)}{\leq} \sum_{S \in \mathcal{S}} \exp\left(-\Delta'^2 t/2\right) = \binom{L}{C} \exp\left(-\Delta'^2 t/2\right),
\end{aligned}
$$

where (h) follows from Hoeffding's inequality [22]. □

Lemma 9. *If $P < \sum_{i=1}^{C} \mu_i$ then under R-LFU,*

$$
\mathbb{P}(\rho_t = 0) \leq \exp\left(-\Delta'^2 t/2\right).
$$

Proof. Let $\mathcal{E}_L$ be the event that $\sum_{i=1}^{C} \hat{\mu}_i(t) > \sum_{i=1}^{C} \mu_i - \epsilon$, where $\epsilon = (\sum_{i=1}^{C} \mu_i - P)/2 = \Delta'/2$. Then,

$$
\begin{aligned}
\mathbb{P}(\rho_t = 0) =& \mathbb{P}\left(P > \max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t)\right) \\
\leq& \mathbb{P}\left(P > \sum_{i=1}^{C} \hat{\mu}_i(t)\right) \\
\leq& \mathbb{P}\left(P > \sum_{i=1}^{C} \hat{\mu}_i(t), \mathcal{E}_L\right) + \mathbb{P}(\mathcal{E}_L^c) \\
\leq& \mathbb{P}\left(P > \frac{\sum_{i=1}^{C} \mu_i + P}{2}\right) + \mathbb{P}(\mathcal{E}_L^c) \\
=& 0 + \mathbb{P}(\mathcal{E}_L^c) \overset{(i)}{\leq} \exp\left(-\Delta'^2 t/2\right),
\end{aligned}
$$

where (i) follows from Lemma 7. □

Lemma 10. *If $P > \sum_{i=1}^{C} \mu_i$, then under R-LFU,*

$$
\mathbb{P}(\rho_t = 1) \leq \binom{L}{C} \exp\left(-\Delta'^2 t/2\right).
$$

Proof. Let $\mathcal{E}_U$ be the event that $\max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t) < \sum_{i=1}^{C} \mu_i + \epsilon$, where $\epsilon = (P - \sum_{i=1}^{C} \mu_i)/2 = \Delta'/2 > 0$. Then,

$$
\begin{aligned}
\mathbb{P}(\rho_t = 1) =& \mathbb{P}\left(P < \max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t)\right) \\
\leq& \mathbb{P}\left(P < \max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i, \mathcal{E}_U\right) + \mathbb{P}(\mathcal{E}_U^c) \\
\leq& \mathbb{P}\left(P < \sum_{i=1}^{C} \mu_i + \epsilon\right) + \mathbb{P}(\mathcal{E}_U^c) \\
=& 0 + \mathbb{P}(\mathcal{E}_U^c) \overset{(j)}{\leq} \binom{L}{C} \exp\left(-\Delta'^2 t/2\right).
\end{aligned}
$$

where (j) follows from Lemma 8. □

Proof of Theorem 2(a). We consider two cases:

(i) If $P < \sum_{i=1}^C \mu_i$, then from Lemma 5 and Lemma 9, we have that

$$\mathcal{R}_S^{\text{R-LFU}} \leq R_S^{\text{LFU}}(T) + \Delta' \sum_{t=1}^{T} \exp\left(-\Delta'^2 t/2\right) \leq R_S^{\text{LFU}}(T) + \frac{2}{\Delta'}.$$

(ii) If $P > \sum_{i=1}^C \mu_i$, then from Lemma 6 and Lemma 10 we have,

$$\mathcal{R}_S^{\text{R-LFU}} \leq P \sum_{t=1}^{T} \binom{L}{C} \exp\left(-\Delta'^2 t/2\right) \leq P \binom{L}{C} \frac{2}{\Delta'^2}.$$

This completes the proof. □

E. Proof of Theorem 2(b)

Lemma 11. *If $P < \sum_{i=1}^C \mu_i$, then under R-FTPL,*

$$\mathbb{P}(\rho_t = 0) \leq \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) + \exp\left(-\Delta'^2 t/2\right).$$

Proof. Let $\mathcal{E}_L$ be the event that $\sum_{i=1}^C \hat{\mu}_i(t) > \sum_{i=1}^C \mu_i - \epsilon$, where $\epsilon = (\sum_{i=1}^C \mu_i - P)/2 = \Delta'/2 > 0$. Then,

$$\begin{aligned}
&\mathbb{P}(\rho_t = 0) \\
=&\mathbb{P}\left(Pt + \eta_t \gamma_{N+1} > \max_{S \in \mathcal{S}} \sum_{i \in S} (t\hat{\mu}_i(t) + \eta_t \gamma_i)\right) \\
\leq&\mathbb{P}\left(Pt + \eta_t \gamma_{N+1} > \sum_{i=1}^{C} (t\hat{\mu}_i(t) + \eta_t \gamma_i)\right) \\
=&\mathbb{P}\left((\eta_t(\gamma_{N+1} - \sum_{i=1}^{C} \gamma_i) > (\sum_{i=1}^{C} \hat{\mu}_i(t) - P)t\right) \\
=&\mathbb{P}\left(\frac{\gamma_{N+1} - \sum_{i=1}^C \gamma_i}{\sqrt{C+1}} > \frac{(\sum_{i=1}^C \hat{\mu}_i(t) - P)t}{\eta_t \sqrt{C+1}}\right) \\
\leq&\mathbb{P}\left(\frac{\gamma_{N+1} - \sum_{i=1}^C \gamma_i}{\sqrt{C+1}} > \frac{(\sum_{i=1}^C \hat{\mu}_i(t) - P)t}{\eta_t \sqrt{C+1}}, \mathcal{E}_L\right) + \mathbb{P}(\mathcal{E}_L^c) \\
\leq&\mathbb{P}\left(\frac{\gamma_{N+1} - \sum_{i=1}^C \gamma_i}{\sqrt{C+1}} > \frac{(\sum_{i=1}^C \mu_i(t) - P)t}{2\eta_t \sqrt{C+1}}\right) + \mathbb{P}(\mathcal{E}_L^c) \\
\overset{(k)}{\leq}& \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) + \exp\left(-\Delta'^2 t/2\right),
\end{aligned}$$

where (k) follows from Lemma 7 and the fact that for standard Gaussian $Q(x) \leq e^{-x^2/2}$. □

Lemma 12. *If $P > \sum_{i=1}^C \mu_i$, then under R-FTPL,*

$$\begin{aligned}
&\mathbb{P}(\rho_t = 1) \\
\leq&\binom{L}{C} \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) + \binom{L}{C} \exp\left(-\Delta'^2 t/2\right).
\end{aligned}$$

Proof. Let $\mathcal{E}_U$ be the event that $\max_{S \in \mathcal{S}} \sum_{i \in S} \hat{\mu}_i(t) < \sum_{i=1}^C \mu_i + \epsilon$, where $\epsilon = P - \sum_{i=1}^C \mu_i/2 = \Delta'/2 > 0$. Then,

$$\begin{aligned}
&\mathbb{P}(\rho_t = 1) \\
=&\mathbb{P}\left(Pt + \eta_t \gamma_{N+1} < \max_{S \in \mathcal{S}} \sum_{i \in S} (t\hat{\mu}_i(t) + \eta_t \gamma_i)\right) \\
\leq&\mathbb{P}\left(Pt + \eta_t \gamma_{N+1} < \max_{S \in \mathcal{S}} \sum_{i \in S} (t\hat{\mu}_i(t) + \eta_t \gamma_i), \mathcal{E}_U\right) + \mathbb{P}(\mathcal{E}_U^c) \\
\leq&\mathbb{P}\left(Pt + \eta_t \gamma_{N+1} < \sum_{i=1}^{C} (t(\mu_i + \epsilon) + \max_{S \in \mathcal{S}} \sum_{i \in S} \eta_t \gamma_i)\right) \\
&+ \mathbb{P}(\mathcal{E}_U^c) \\
=&\mathbb{P}\left((\eta_t(\gamma_{N+1} - \max_{S \in \mathcal{S}} \sum_{i \in S} \gamma_i) > (P - \sum_{i=1}^{C} \mu_i)t/2\right) + \mathbb{P}(\mathcal{E}_U^c) \\
\leq&\binom{L}{C} \mathbb{P}\left(\frac{\gamma_{N+1} - \sum_{i \in S} \gamma_i}{\sqrt{C+1}} > \frac{(P - \sum_{i=1}^C \mu_i)t}{2\eta_t \sqrt{C+1}}\right) + \mathbb{P}(\mathcal{E}_U^c) \\
\overset{(l)}{\leq}&\binom{L}{C} \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) + \binom{L}{C} \exp\left(-\Delta'^2 t/2\right),
\end{aligned}$$

where (l) follows from Lemma 8 and the fact that for standard Gaussian $Q(x) \leq e^{-x^2/2}$. □

Proof of Theorem 2(b). We consider two cases:

(i) If $P < \sum_{i=1}^C \mu_i$, then from Lemma 5 and Lemma 11 we have,

$$\begin{aligned}
\mathcal{R}_S^{\text{R-FTPL}} \leq& R_S^{\text{FTPL}}(T) + \Delta' \sum_{t=1}^{T} \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) \\
&+ \Delta' \sum_{t=1}^{T} \exp\left(-\Delta'^2 t/2\right).
\end{aligned}$$

For $\eta_t = \alpha\sqrt{t/(C+1)}$ we have,

$$\begin{aligned}
\mathcal{R}_S^{\text{R-FTPL}} \leq& R_S^{\text{FTPL}}(T) + \Delta' \sum_{t=1}^{T} \exp\left(\frac{-\Delta'^2 t}{8\alpha^2}\right) \\
&+ \exp\left(-\Delta'^2 t/2\right) \\
\leq& R_S^{\text{FTPL}}(T) + \frac{8\alpha^2 + 2}{\Delta'}.
\end{aligned}$$

(ii) If $P > \sum_{i=1}^C \mu_i$, then from Lemma 6 and Lemma 12 we have,

$$\begin{aligned}
\mathcal{R}_S^{\text{R-FTPL}} \leq& P \binom{L}{C} \sum_{t=1}^{T} \exp\left(\frac{-\Delta'^2 t^2}{8\eta_t^2 (C+1)}\right) \\
&+ \exp\left(-\Delta'^2 t/2\right).
\end{aligned}$$

For $\eta_t = \alpha\sqrt{t/(C+1)}$ we have,

$$\begin{aligned}
\mathcal{R}_S^{\text{R-FTPL}} \leq& P \binom{L}{C} \sum_{t=1}^{T} \exp\left(\frac{-\Delta'^2 t}{8\alpha^2}\right) + \exp\left(-\Delta'^2 t/2\right) \\
\leq& P \binom{L}{C} \frac{8\alpha^2 + 2}{\Delta'^2}.
\end{aligned}$$

This completes the proof. □

References

[1] G. S. Paschos, A. Destounis, L. Vigneri, and G. Iosifidis, "Learning to cache with no regrets," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. IEEE Press, 2019, p. 235–243.

[2] R. Bhattacharjee, S. Banerjee, and A. Sinha, "Fundamental limits on the regret of online network-caching," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 2, pp. 1–31, 2020.

[3] S. Mukhopadhyay and A. Sinha, "Online caching with optimal switching regret," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1546–1551.

[4] F. Z. Faizal, P. Singh, N. Karamchandani, and S. Moharir, "Regret-optimal online caching for adversarial and stochastic arrivals," in *EAI International Conference on Performance Evaluation Methodologies and Tools*. Springer, 2022, pp. 147–163.

[5] "Aws," 2024. [Online]. Available: https://aws.amazon.com

[6] "Azure," 2024. [Online]. Available: https://azure.microsoft.com/

[7] L. Chen and J. Xu, "Collaborative service caching for edge computing in dense small cell networks," *arXiv preprint arXiv:1709.08662*, 2017.

[8] D. Willis, A. Dasgupta, and S. Banerjee, "Paradrop: a multi-tenant platform to dynamically install third party services on wireless gateways," in *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, 2014, pp. 43–48.

[9] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[10] A. Cohen and T. Hazan, "Following the perturbed leader for online structured learning," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1034–1042.

[11] S. Albers, *Competitive online algorithms*. BRICS, 1996.

[12] A. Bura, D. Rengarajan, D. Kalathil, S. Shakkottai, and J.-F. Chamberland, "Learning to cache and caching to learn: Regret analysis of caching algorithms," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 18–31, 2022.

[13] R. Bhattacharjee, S. Banerjee, and A. Sinha, "Fundamental limits of online network-caching," *CoRR*, vol. abs/2003.14085, 2020.

[14] V. C. L. N. BORUSU and M. AGARWALA, "Online partial service hosting at the edge," *Modeling and Performance Evaluation of Computing Systems*, vol. 9, no. 1, 2024.

[15] R. S. Prakash, N. Karamchandani, and S. Moharir, "On the regret of online edge service hosting," *Performance Evaluation*, vol. 162, p. 102367, 2023.

[16] R. S. Prakash, N. Karamchandani, V. Kavitha, and S. Moharir, "Partial service caching at the edge," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2020, pp. 1–8.

[17] V. C. L. Narayana, S. Moharir, and N. Karamchandani, "On renting edge resources for service hosting," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 6, no. 2, pp. 1–30, 2021.

[18] ——, "Retrorenting: An online policy for service caching at the edge," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2020, pp. 1–8.

[19] "MovieLens 1M dataset," https://grouplens.org/datasets/movielens/, [accessed 8-August-2022].

[20] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, dec 2015.

[21] J. Abernethy, C. Lee, A. Sinha, and A. Tewari, "Online linear optimization via smoothing," in *Conference on Learning Theory*. PMLR, 2014, pp. 807–823.

[22] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.