A New Flexible Multi-flow LRU Cache Management Paradigm for Minimizing Misses

GUOCONG QUAN, The Ohio State University, USA
JIAN TAN, Alibaba Group, USA & The Ohio State University, USA
ATILLA ERYILMAZ, The Ohio State University, USA
NESS SHROFF, The Ohio State University, USA

The Least Recently Used (LRU) caching and its variants are used in large-scale data systems in order to provide high-speed data access for a wide class of applications. Nonetheless, a fundamental question still remains open: in order to minimize miss probabilities, how should the cache space be organized to serve multiple data flows? Commonly used strategies can be categorized into two designs: pooled LRU (PLRU) caching and separated LRU (SLRU) caching. However, neither of these designs can satisfactorily solve this problem. PLRU caching is easy to implement and self-adaptive, but does not often achieve optimal or even efficient performance because its set of feasible solutions are limited. SLRU caching can be statically configured to achieve optimal performance for stationary workload, which nevertheless could suffer in a dynamically changing environment and from a cold-start problem.

To this end, we propose a new insertion based pooled LRU paradigm, termed I-PLRU, where data flows can be inserted at different positions of a pooled cache. This new design can achieve the optimal performance of the static SLRU, and retains the adaptability of PLRU in virtue of resource sharing. Theoretically, we characterize the asymptotic miss probabilities of I-PLRU, and prove that, for any given SLRU design, there always exists an I-PLRU configuration that achieves the same asymptotic miss probability, and vice versa. We next design a policy to minimize the miss probabilities. However, the miss probability minimization problem turns out to be non-convex under the I-PLRU paradigm. Notably, we utilize an equivalence mapping between I-PLRU and SLRU to efficiently find the optimal I-PLRU configuration. We prove that I-PLRU outperforms PLRU and achieves the same miss probability as the optimal SLRU for stationary workload. Engineeringly, the flexibility of I-PLRU avoids separating the memory space, supports dynamic and refined configurations, and alleviates the cold-start problem, potentially yielding better performance than both SLRU and PLRU.

CCS Concepts: • **Theory of computation** → *Caching and paging algorithms*; • **General and reference** → *Performance*; • **Mathematics of computing** → *Stochastic processes*;

Keywords: Caching; LRU; Miss probability

ACM Reference Format:
Guocong Quan, Jian Tan, Atilla Eryilmaz, and Ness Shroff. 2019. A New Flexible Multi-flow LRU Cache Management Paradigm for Minimizing Misses. In *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 3, 2, Article 39 (June 2019). ACM, New York, NY. 30 pages. https://doi.org/10.1145/3326154

This work is supported by the DTRA grants: HDTRA-14-1-0058, HDTRA1-15-1-0003, HDTRA1-18-1-0050, the NSF grants: CMMI-SMOR-1562065, CNS-1446582, CNS-ICN-WEN-1719371, CNS-NeTS 1409336, CNS-NeTS-1514260, CNS-NeTS 1518829, CNS-NeTS-1717045, CNS-NeTS-1717060, CNS-SpecEES-1824337, CSR-NeTS 1717060, and the ONR grant: N00014-17-1-2417.

Authors' addresses: Guocong Quan, The Ohio State University, USA, quan.72@osu.edu; Jian Tan, Alibaba Group, USA & The Ohio State University, USA, j.tan@alibaba-inc.com, tan.252@osu.edu; Atilla Eryilmaz, The Ohio State University, USA, eryilmaz.2@osu.edu; Ness Shroff, The Ohio State University, USA, shroff.11@osu.edu.

2476-1249/2019/6-ART39 $15.00
https://doi.org/10.1145/3326154

1 INTRODUCTION

With increasingly deployed data-intensive applications, the critical role of caching in accelerating data access is becoming even more important. When a requested data item is found in the cache, called a *cache hit*, it can be served fast. Otherwise, a *cache miss* occurs, causing a significantly longer delay. A number of caching policies[5, 6, 11, 13, 20, 22, 23] have been proposed to update the data items in the cache. Among them, the least recently used (LRU) policy or its variants [4, 19, 24, 27, 29] are implemented as default [1, 2], owing to their simplicity and self-adaptive property [28]. For LRU, data items are listed in descending order of their last requested times. Upon a data item being requested, it is moved to the head of the list. If a miss occurs and the cache is full, the least recently used data item(s), i.e., the one(s) at the end of the list, would be evicted from the cache to accommodate the newly requested one.

A fundamental question still remains open: *in order to minimize the miss probabilities, how should the cache space be organized to serve multiple data flows?* Commonly used strategies to organize LRU caching can be categorized into two designs: pooled LRU (PLRU) caching and separated LRU (SLRU) caching. For PLRU, the entire cache space is pooled as a single LRU cache and serves multiple data flows by allowing complete cache sharing among the data flows. In contrast, for SLRU, the cache space is separated into multiple LRU cache partitions, and each flow is served by a dedicated partition.

Theoretical studies have been conducted to compare the performance of PLRU and SLRU [9, 26, 28] through characterizing the miss ratios of LRU caching [12, 13, 15–18, 25]. Remarkably, PLRU caching enjoys a nice adaptivity property [23, 28], which often yields good performance for data request flows that dynamically change over time. However, in a stationary setting, it is proven [9, 28] that the optimal SLRU caching achieves asymptotic miss probabilities at least as good as PLRU caching. In a general setting, it is reported that separating cache space is more advantageous [8]. However, due to lack of adaptivity, it is difficult for SLRU to retain the optimal performance when data statistics, e.g., data item popularities and item request rates, are time-varying. This could cause low utilization and inefficiency since the separated multiple cache partitions could be unbalanced. Importantly, dynamically resizing these separate cache partitions incurs overhead, e.g., using auto-mover for Memcached [1]. Specifically, a so-called cold-start problem [10] can deteriorate the performance during a transition period immediately after resizing the cache (see Section 3.3 on the cold-start problem). Other problems of dynamic resizing have also been reported, e.g., memory fragmentation [3, 21, 30].

To mitigate these problems, we develop a new insertion based pooled LRU (I-PLRU) caching paradigm. It achieves the optimal performance of static SLRU caching, provides more flexibility with refined control, and alleviates the cold-start problem in dynamically changing environments. For this new design, the cache space is pooled together and the data flows are inserted into the cache from different positions along the ordered data item list (see Section 4.1 for the formal definition). The miss probabilities can be optimized by configuring the insertion position of each flow. Moreover, when arrival rates or popularity distributions of the data flows dynamically change over time, the configuration can be easily adapted to retain the high efficiency.

Notably, the analysis of I-PLRU is more challenging than PLRU and SLRU. Different insertion positions significantly complicate the way that the data flows interact with each other. More concretely, 1) the maintained data item list is not completely sorted anymore; 2) the data items of different flows could be organized in the same cache space through various ways to achieve more refined control. Fortunately, we establish an equivalence mapping between I-PLRU and SLRU, based on which, we rigorously characterize the asymptotic performance of I-PLRU, and prove that I-PLRU

can achieve the same miss probabilities as the optimal SLRU. We summarize our contributions as follows.

Summary of contributions:

(1) We propose a new LRU cache management paradigm, termed I-PLRU, for multiple flows. It can be flexibly configured to optimize various performance objectives, and effectively alleviate the cold-start problem that hurts the performance of LRU caching during resizing.
(2) We rigorously characterize the asymptotic miss probability of I-PLRU caching. Under I-PLRU, the data flows are coupled together in a complicated way. Existing analytical tools for LRU caching cannot be directly applied. Instead, we overcome the difficulties by establishing an equivalence mapping between I-PLRU and SLRU. Specifically, we prove that for any given SLRU configuration, there always exists an I-PLRU configuration under which the asymptotic miss probability of each flow is the same as that under the SLRU configuration, and vice versa. Furthermore, we prove that the equivalence mapping is one-to-one.
(3) We study a class of performance optimization problems for I-PLRU caching based on miss probabilities. Such problems turn out to be non-convex. Though solving a general non-convex optimization problem is difficult, this class has a special structure to exploit. By using the equivalence mapping, we prove that the non-convex problem has only one stationary point, which is the global optimum. Interestingly, this equivalence mapping transfers the non-convex problem under I-PLRU to a convex problem under SLRU, which is analytically tractable. In a reverse direction, the optimal SLRU configuration can be mapped back to the optimal I-PLRU configuration.

The rest of the paper is organized as follows. In Section 2, we introduce notations and formulate the miss probability minimization (MPM) problem. In Section 3, we present the limitations of PLRU and SLRU. In Section 4, we propose the new caching paradigm I-PLRU and rigorously characterize its asymptotic performance. We also solve the MPM problem for I-PLRU in this section. In Section 5, we discuss the engineering issues including general or unknown popularity distributions. The theoretical results are validated by simulations in Section 6. In Section 7, we conclude our work. The proofs of main theorems are provided in Section 8.

2 PROBLEM FORMULATION

The broad objective of this paper is to systematically develop an easy-to-implement and provably efficient LRU-based cache management mechanism that allows multiple flows to flexibly share a total memory space. To that end, in this section, we introduce the basic setting and the miss probability minimization (MPM) problem that we will tackle in the subsequent sections.

Consider M data flows, where a data flow is a sequence of data requests from a data domain or an application. Let $\mathcal{D}_m = \{d_i^{(m)}, i \geq 1\}$ denote the set of data items that are requested by flow m, $1 \leq m \leq M$. Assume that $\mathcal{D}_m$'s are disjoint sets and the data items have unit sizes. Note that if $\mathcal{D}_m$'s are overlapped, we can always separate the flows into multiple subflows such that the subflows have no overlap, and the results of this paper will hold for the subflows. A similar trick can be found in [9, 28]. The requests of flow m arrive according to a Poisson process with an arrival rate λ_m, $1 \leq m \leq M$. Let $\{\tau_n, -\infty < n < \infty\}$ denote the sequence of epochs when the requests arrive. Let I_n and R_n denote the flow index and the requested data that arrive at τ_n, respectively. Note that $I_n \in \{1, 2, \cdots, M\}$ and $R_n \in \mathcal{D}_{I_n}$. After the system reaches its stationarity, it suffices to analyze the system at a given epoch, say τ_0. We assume an independent reference model (IRM) [29], i.e., for $1 \leq m \leq M$,

1) the requests of different flows arrive independently with

$$v_m \triangleq \mathbb{P}[I_0 = m] = \lambda_m \Big/ \sum_{k=1}^{M} \lambda_k;$$

2) the requests within each flow are independent and follow the popularity distribution

$$p_i^{(m)} \triangleq \mathbb{P}\left[R_0 = d_i^{(m)} \middle| I_0 = m\right], \quad i \geq 1.$$

As reported by analysis on real data traces [31], the requests typically follow a Zipf's distribution. Thus, we assume, for $i \geq 1$, $1 \leq m \leq M$

$$p_i^{(m)} \sim c_m / i^{\alpha_m}, \quad \alpha_m > 1, \tag{1}$$

where $f(x) \sim g(x)$ means $\lim_{x\to\infty} f(x)/g(x) = 1$.

Let π be the cache management paradigm (e.g., PLRU, SLRU) that organizes the cache space to serve multiple flows. We use Q_m^π to denote the miss probability of flow m under the paradigm π, i.e.,

$$Q_m^\pi \triangleq \mathbb{P}\left[R_0 \text{ is a miss} | I_0 = m; \pi\right].$$

Miss probability minimization:
Miss probability minimization (MPM) is a fundamental problem for caching systems that support data-intensive applications. For a given cache space of total size C, the objective is to minimize the miss probability. The problem is formulated as follows

$$\begin{aligned} \min_{\pi} \quad & \sum_{m=1}^{M} w_m \cdot Q_m^\pi \\ \text{subject to} \quad & \text{The total cache size is } C, \end{aligned} \tag{2}$$

where w_m's are arbitrary positive weights. The analysis in this paper for Problem (2) can be easily extended to general objective functions $\sum_{m=1}^{M} u_m(Q_m^\pi)$ where $u_m(\cdot)$'s are convex functions. More comments on the extension is provided in Section 4.3.

3 EXISTING APPROACHES

In this section, we summarize two commonly used design strategies, SLRU and PLRU, that are used to organize LRU caching for multiple data flows. We discuss their limitations, which motivate a new flexible cache management paradigm, i.e., I-PLRU.

3.1 Separated LRU (SLRU) Caching

Separated LRU (SLRU) caching is one of the most commonly used methods to organize cache space for multiple data flows [1, 2]. Under the SLRU paradigm, the total cache space is separated into multiple LRU caches and each flow is served by a dedicated partition as shown in Fig. 1. In general, consider M data flows served by SLRU with the total cache size C. Assume data flow m is only served by the m^{th} LRU cache, $1 \leq m \leq M$. Let $\theta_m C$ denote the size of the cache space allocated to the m^{th} LRU cache with $\sum_{m=1}^{M} \theta_m = 1$, $0 \leq \theta_m \leq 1$, $1 \leq m \leq M$. Although each cache can only store an integer number of data items, we assume that the cache size can take continuous real values for analytical convenience, because the discrete constraints will have a vanishing impact on the asymptotic results as $C \to \infty$. Note that the SLRU caching can be characterized by the allocation configuration $\boldsymbol{\theta}$ and the total cache size C. Therefore, we introduce the following definition.

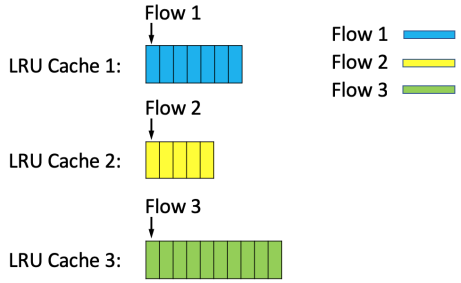


Fig. 1. Three data flows organized by SLRU caching.

Definition 3.1 (SLRU). Set $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_M)$, $\theta_m > 0, 1 \le m \le M$, $\sum_{m=1}^{M} \theta_m = 1$. Define $\mathcal{S}(\boldsymbol{\theta}; C)$ as the SLRU paradigm where the total cache size is C and the size of the cache space allocated to flow m is $\theta_m C$.

Notably, under the SLRU paradigm, the flows are served independently by the corresponding LRU caches. The miss probability of each flow can be obtained using the analytical tool for standard LRU caching. Applying existing results in [16], the asymptotic miss probabilities under SLRU are provided in the following lemma.

Lemma 3.2 ([16]). *Consider M data flows organized by the SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$. Let $Q_m^{SLRU}(\boldsymbol{\theta}; C)$ denote the miss probability of flow m, $1 \le m \le M$. We have, as $C \to \infty$*

$$Q_m^{SLRU}(\boldsymbol{\theta}; C) \sim \frac{\Gamma(1 - 1/\alpha_m)^{\alpha_m}}{\alpha_m} \frac{c_m}{(\theta_m C)^{\alpha_m - 1}},$$

where $\Gamma(x) = \int_{t=0}^{\infty} t^{x-1} e^{-t} dt$ is the gamma function.

Recall that $f(x) \sim g(x)$ indicates $\lim_{x \to \infty} f(x)/g(x) = 1$. The miss probability can be minimized by optimizing the cache space allocated to each flow. We formulate the MPM problem under SLRU as

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{m=1}^{M} w_m \cdot Q_m^{\text{SLRU}}(\boldsymbol{\theta}; C) \\ \text{subject to} \quad & \theta_m \ge 0, \quad 1 \le m \le M, \\ & \sum_{m=1}^{M} \theta_m = 1, \end{aligned} \tag{3}$$

where w_m's are positive weights. Lemma 3.2 shows that the asymptotic miss probability of SLRU is a convex function with respect to $\boldsymbol{\theta}$. As a result, the MPM problem under SLRU is asymptotically a convex problem. Let $\boldsymbol{\theta}^*(C)$ denote the optimal solution given the total cache size C. According to the KKT conditions [7], we have as $C \to \infty$, for any $1 \le i, j \le M$

$$\frac{\theta_i^*(C)}{\theta_j^*(C)} \sim \frac{\Gamma(1 - 1/\alpha_i)^{\alpha_i} (1 - 1/\alpha_i) c_i w_i}{\Gamma(1 - 1/\alpha_j)^{\alpha_j} (1 - 1/\alpha_j) c_j w_j} \cdot C^{-\alpha_i + \alpha_j}. \tag{4}$$

Combining (4) and the fact that $\sum_{i=1}^{M} \theta_i^*(C)$, we can solve $\boldsymbol{\theta}^*(C)$ explicitly. Moreover, even for general objectives $\sum_{m=1}^{M} u_m(Q_m^{\text{SLRU}}(\boldsymbol{\theta}; C))$, the problem remains convex as long as $u_m(\cdot)$'s are convex functions. More details of the MPM problem under SLRU are discussed in [9, 28].

3.2 Pooled LRU (PLRU) Caching

Instead of separating the cache space into multiple LRU cache partitions that serve data flows dedicatedly, PLRU caching organizes the total cache space as a single LRU list as shown in Fig. 2. Once a request arrives, the requested data will be placed to the head of the list, no matter which flow it belongs to. In order to make room for newly requested data, the least recently used data item (i.e., the data stored at the rear) will be evicted if necessary. As we can see in Fig 2, the entire cache is shared by all data flows under LRU. Consequently, the cache space occupied by each flow is not fixed.

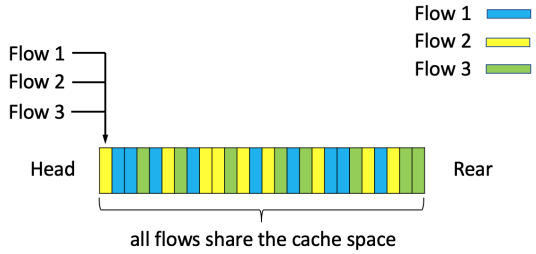


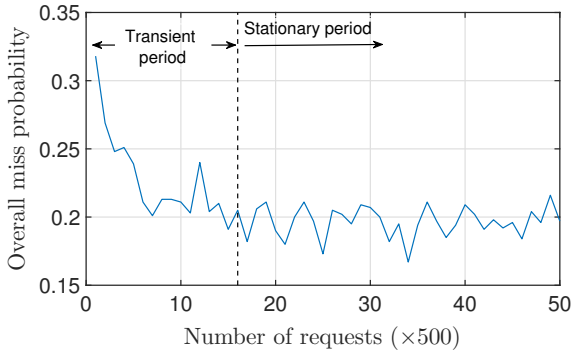
Fig. 2. Three data flows organized by PLRU caching.

The PLRU paradigm has the advantage of simplicity and resource pooling nature whereby the cache space can be used more adaptively by all flows, when data statistics (e.g., popularities, request rates) are time-varying. The asymptotic miss probability of PLRU is characterized in [9, 28]. Different from SLRU, PLRU does not support flexible configurations and consequently does not generally achieve the minimum miss probability of SLRU. However, it is proven in [28] that PLRU automatically optimizes the MPM problem with a specific objective function $\sum_{m=1}^{M} \nu_m \, Q_m^{\pi}$ for any Zipf's popularity distributions, where $\nu_m = \mathbb{P}[I_0 = m]$.

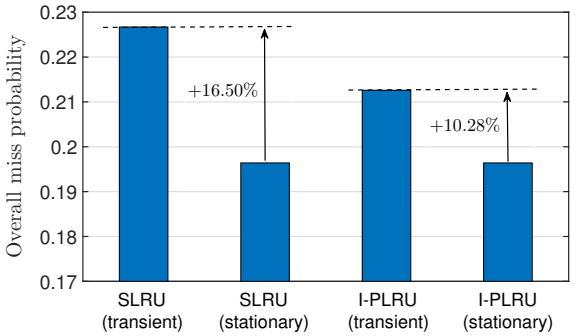
3.3 Limitations of SLRU & PLRU

Despite their appealing characteristics outlined above and successful applications, both SLRU and PLRU have limitations. In this section, we illustrate their limitations by simulation examples.

1. Limitations of SLRU:
Deteriorating performance in a dynamically changing environment: When the statistics of the workload change over time, a static cache space allocation cannot always achieve the optimal performance. On the other hand, if dynamically resizing the cache partitions reallocated among flows, a so-called cold-start problem will deteriorate the system performance [10]. When a portion of cache space is reallocated, the data stored in this portion will be invalidated, which incurs high miss probabilities before the cache becomes full again. This phenomenon is called the cold-start problem. SLRU will suffer from the cold-start problem when changing the configuration. Consider an SLRU system serving two flows with the objective to minimize the overall miss probability $\nu_1 Q_1^{\text{SLRU}} + \nu_2 Q_2^{\text{SLRU}}$. Let $\alpha_1 = \alpha_2 = 1.2$ and $C = 4000$. Assume that the data set for each flow has 10^6 data items. We have $c_1 = c_2 = 1/\sum_{i=1}^{10^6} i^{-1.2} = 0.1895$. Assume that the workload has two stages. In Stage 1, the system only serves flow 1, i.e., $\nu_1 = 1, \nu_2 = 0$. To minimize the miss probability, all cache space is allocated to flow 1. After serving 10^5 requests from flow 1, the system enters Stage 2 and starts to serve two flows with $\nu_1 = \nu_2 = 0.5$. The optimal cache space allocation in Stage 2 is $\theta_1 = \theta_2 = 0.5$ due to the symmetric setting. At the beginning of Stage 2, no valid data is stored in the cache space allocated to flow 2, and the system suffers from the cold-start problem. We define the transient period as the time period when the cache of flow 2 is not full. After that, the system is



(a) Transient period of cold-start.



(b) Increased miss probability during transient period.

Fig. 3. Deteriorating performance of SLRU. The figures show that the cold-start increases the miss probability, but I-PLRU is less impacted than SLRU.

in the stationary period. In Fig. 3a, we illustrate how the overall miss probability changes over time in Stage 2 under the SLRU paradigm. The miss probability is approximated by the miss frequency of every 500 requests. It can be observed that the overall miss probability during the transient period is much higher than that during the stationary period. In Fig. 3b, we plot the average miss probability of the transient period and the stationary period for both SLRU and I-PLRU. It can be observed that the miss probability of I-PLRU only increases by 10.28% during the transient period, compared with a 16.50% increment of SLRU. Therefore, the new I-PLRU paradigm is less impacted by the cold-start problem. Moreover, I-PLRU achieves the same stationary miss probability with the optimal SLRU.

2. Limitations of PLRU:

Lack of refined control for individual flows: Once the total cache space is given, the PLRU paradigm is fixed. It does not support flexible configurations for individual flows to optimize general performance objectives. We overcome this limitation by proposing I-PLRU paradigm, which assigns individualized insertion positions for different flows.

Consider 2 data flows with $v_1 = v_2 = 0.5, \alpha_1 = \alpha_2 = 2$. Assume that the data set for each flow has 10^6 data items with $c_1 = c_2 = 1/\sum_{i=1}^{10^6} i^{-2} = 0.6079$. The performance objective is to minimize the overall miss probabilities $w_1 Q_1^\pi + w_2 Q_2^\pi$. Let $w_1 = 0.1, w_2 = 0.9$. We plot the overall miss probabilities achieved by PLRU and the optimal I-PLRU in Fig. 4. We observe that the new I-PLRU paradigm achieves better performance than PLRU.

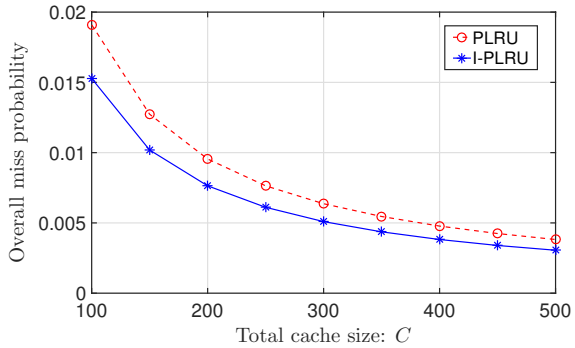


Fig. 4. Suboptimal performance of PLRU. The figure shows that I-PLRU can achieve better miss probabilities than PLRU.

4 A NEW FLEXIBLE CACHE MANAGEMENT PARADIGM

In this section, we propose an insertion based pooled LRU caching design, termed I-PLRU. It achieves the high efficiency of SLRU and retains the adaptability of PLRU at the same time. In Section 4.1, we introduce the definition of I-PLRU. In Section 4.2, we rigorously characterize the asymptotic miss probability achieved by I-PLRU, and establish an equivalence mapping between I-PLRU and SLRU. In Section 4.3, we formulate the MPM problem for I-PLRU and find the optimal I-PLRU configuration based on the equivalence mapping.

4.1 Definition of I-PLRU

Under the I-PLRU paradigm, the memory space is organized as a single list and serves multiple flows in a common shared cache as in the PLRU mechanism. However, different from the PLRU paradigm, data flows can be inserted at different positions rather than merely at the head of the list. Specifically, each data flow is assigned with an insertion position. Once a request arrives, the requested data will be inserted at the corresponding position in an LRU fashion. Note that PLRU is a special case of I-PLRU where the insertion positions of all flows are the head of the list. If a miss occurs and the cache is full, the data stored at the rear of the list will be moved out of the cache to make room for the newly requested one. Remarkably, under I-PLRU, data items are not fully sorted as in the PLRU mechanism, because the data flows can be inserted at different positions.

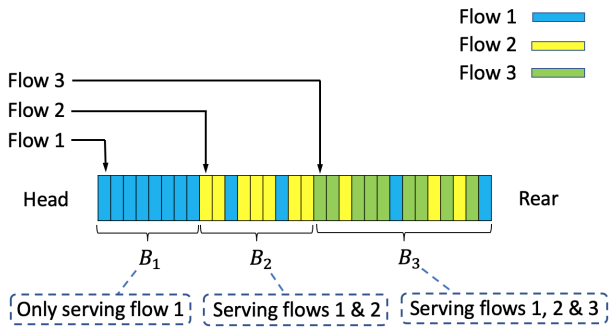


Fig. 5. Three data flows organized by I-PLRU caching.

Without loss of generality, assume the flows are sorted according to their insertion positions, such that flow 1 is inserted at the head of the list. We illustrate an I-PLRU paradigm serving 3 flows in Fig. 5. The cache can be labeled as M consecutive blocks (say B_m, $1 \leq m \leq M$) such that flow m

is inserted at the first position of B_m. According to the insertion and eviction policy, the memory block B_m is shared by flows $1, \cdots, m$, but not flows $m+1, \cdots, M$. Notably, an I-PLRU paradigm can be characterized by the total cache size C and the insertion positions (or equivalently, the size of B_m's).

Definition 4.1 (I-PLRU). Set $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_M)$, $\eta_m \geq 0$, $1 \leq m \leq M$, $\sum_{m=1}^{M} \eta_m = 1$. Define $\mathcal{I}(\boldsymbol{\eta}; C)$ as an I-PLRU paradigm where the total cache size is C and the cache size of B_m is $\eta_m C$.

We list the key advantages of I-PLRU over PLRU and SLRU as follows.

(1) *High Efficiency:* I-PLRU supports flexible configurations to optimize system performance. As one of the main results of this paper, we will show (in Theorem 4.9) that I-PLRU can achieve the same miss probabilities as the optimal SLRU paradigm, and significantly improves the performance of conventional PLRU.
(2) *High Adaptability:* When configurations require adaptive updates in dynamically changing environments, I-PLRU is less impacted by the cold-start problem compared to SLRU. By changing insertion positions rather than cache partitions, the memory space under I-PLRU is not pre-allocated to a flow until sufficient requests arrive. Consequently, the cache is never empty even when configurations are dynamically adapted. We show the benefits of I-PLRU under the cold-start through simulation results (in Experiment 3).

Despite all these advantages, the theoretical analysis for I-PLRU is far more challenging than that for PLRU and SLRU. We will show (in Section 4.3) that the MPM problem under I-PLRU is non-convex. To illustrate the difficulties, consider three flows served by PLRU and I-PLRU shown in Fig. 2 and Fig. 5, respectively. Under the PLRU paradigm, data items of the three flows are evenly distributed in the cache if popularity distributions are similar. Under the I-PLRU paradigm, however, data items of each flow are more concentrated around its insertion position. As a result, the flows are coupled together in a complicated way.

4.2 Equivalence Mapping Between I-PLRU and SLRU Paradigms

In this section, as the total cache size $C \to \infty$, we characterize the asymptotic behavior of the proposed I-PLRU paradigm by establishing an equivalence mapping between I-PLRU and SLRU.

Let the random variable $X_m(\boldsymbol{\eta}; C)$ denote the number of data items of flow m stored in the I-PLRU $\mathcal{I}(\boldsymbol{\eta}; C)$. Let $Q_m^{\text{I-PLRU}}(\boldsymbol{\eta}; C)$ denote the miss probability of flow m under $\mathcal{I}(\boldsymbol{\eta}; C)$. In this section, we rigorously characterize the asymptotic behavior of $X_m(\boldsymbol{\eta}; C)$ and $Q_m^{\text{I-PLRU}}(\boldsymbol{\eta}; C)$ as the total cache size $C \to \infty$.

Definition 4.2 (Equivalence). Consider M data flows and the cache space of size C. We say that the I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$ and the SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$ are equivalent, denoted by

$$\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(\boldsymbol{\theta}; C),$$

if, for any $1 \leq m \leq M$, as the total cache size $C \to \infty$

$$\frac{X_m(\boldsymbol{\eta}; C)}{\theta_m C} \xrightarrow{a.s.} 1.$$

An I-PLRU configuration is equivalent to an SLRU configuration, if for each flow, the number of items stored in the I-PLRU paradigm is almost surely concentrated around the cache space allocated to that flow in the SLRU paradigm, when the total cache space is sufficiently large. Note that $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ in Definition 4.2 can be functions of the total cache size C. Based on this definition, we show that equivalent SLRU and I-PLRU configurations achieve the same asymptotic miss probabilities in the following theorem.

Theorem 4.3. *Consider the I-PLRU configuration $\mathcal{I}(\boldsymbol{\eta}; C)$ and the SLRU configuration $\mathcal{S}(\boldsymbol{\theta}; C)$. If*

$$\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(\boldsymbol{\theta}; C),$$

then for $1 \le m \le M$, we have, as the total cache size $C \to \infty$

$$\begin{aligned} Q_m^{\text{I-PLRU}}(\boldsymbol{\eta}; C) &\sim Q_m^{\text{SLRU}}(\boldsymbol{\theta}; C) \\ &\sim \frac{\Gamma(1-1/\alpha_m)^{\alpha_m}}{\alpha_m} \frac{c_m}{(\theta_m C)^{\alpha_m - 1}}. \end{aligned}$$

The proof is presented in Section 8.1. If an I-PLRU configuration and an SLRU configuration are equivalent, then each flow will achieve the same asymptotic miss probability under these two paradigms. Therefore, we can characterize the miss probability for I-PLRU by first identifying its equivalent SLRU configuration and then applying Theorem 4.3. Next, we will show how to find the equivalent SLRU configuration $\mathcal{S}(\boldsymbol{\theta}; C)$ for a given I-PLRU configuration $\mathcal{I}(\boldsymbol{\eta}; C)$, and vice versa.

Theorem 4.4. *Consider M data flows served by an I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$. Assume that there exists $\beta \in (-1, 0]$ such that $\eta_M \gtrsim C^{\beta}$ as $C \to \infty$. Let $F_1(\boldsymbol{\eta}; C)$ denote the output of Algorithm 1 with $\boldsymbol{\eta}$ and C as its input. We have*

$$\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(F_1(\boldsymbol{\eta}; C); C).$$

Algorithm 1: Finding the equivalent SLRU for I-PLRU

Output: θ_m, $1 \le m \le M$
Input: η_m, $1 \le m \le M$, C
Initialization: set $\theta_m = 0$, $t_m = 0$, $1 \le m \le M$;
$\theta_1 \leftarrow \eta_1$;
for $m \leftarrow 2$ **to** M **do**
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $t_i \leftarrow \left(\frac{\theta_i C}{\Gamma(1-1/\alpha_i) {c_i}^{1/\alpha_i}}\right)^{\alpha_i}$;
  end
  Solve z as the unique solution of

$$\sum_{i=1}^{m} \Gamma\left(1-1/\alpha_i\right) {c_i}^{1/\alpha_i} (t_i + \nu_i z)^{1/\alpha_i} = \sum_{i=1}^{m} \eta_i C;$$

  for $i \leftarrow 1$ **to** m **do**
    $\theta_i \leftarrow \Gamma\left(1-1/\alpha_i\right) {c_i}^{1/\alpha_i} (t_i + \nu_i z)^{1/\alpha_i} / C$;
  end
end

Note that $f(x) \gtrsim g(x)$ means $\lim_{x\to\infty} f(x)/g(x) \ge 1$. The assumption in Theorem 4.4 requires that η_M cannot be too small. For example, η_M can take any constant value in $(0, 1]$. In Algorithm 1, we calculate $\boldsymbol{\theta}$ recursively. Recall that the I-PLRU cache can be labeled as M blocks (i.e., B_m, $1 \le m \le M$) by the insertion positions. In Algorithm 1, we start from the first two blocks B_1 and B_2. We calculate the equivalent SLRU configuration for the subsystem that consists of B_1 and B_2, based on which, the equivalent SLRU configurations for the subsystems that consist of $3, \cdots, M-1$ blocks are calculated recursively. And finally we derive the equivalent SLRU configuration for the system that consists of all M blocks, i.e., our original I-PLRU system. Detailed explanations and proofs of Theorem 4.4 are presented in Section 8.2. Next, we consider the inverse mapping that finds the equivalent I-PLRU configuration $\mathcal{I}(\boldsymbol{\eta}; C)$ for a given SLRU configuration $\mathcal{S}(\boldsymbol{\theta}; C)$.

Theorem 4.5. *Consider M flows served by an SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$. Assume that for $1 \leq m \leq M$, there exists $\beta_m \in (-1, 0]$ such that $\theta_m \gtrsim C^{\beta_m}$ as $C \to \infty$, and the flows are sorted such that $(\theta_m C)^{\alpha_m}/(\Gamma(1-1/\alpha_m)^{\alpha_m} c_m \nu_m)$ is decreasing with respect to m. Let $F_2(\boldsymbol{\theta}; C)$ be the output of Algorithm 2 with $\boldsymbol{\theta}$ and C as its input. We have*

$$\mathcal{S}(\boldsymbol{\theta}; C) \equiv \mathcal{I}(F_2(\boldsymbol{\theta}; C); C).$$

Algorithm 2: Finding the equivalent I-PLRU for SLRU

Output: η_m, $1 \leq m \leq M$
Input: θ_m, $1 \leq m \leq M$, C
Initialization: set $\eta_m = 1$, $1 \leq m \leq M$;
for $m \leftarrow M$ **to** 2 **do**
  $z \leftarrow \left(\frac{\theta_m C}{\Gamma(1-1/\alpha_m)(c_m \nu_m)^{1/\alpha_m}}\right)^{\alpha_m}$;
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $t_i \leftarrow \left(\frac{\theta_i C}{\Gamma(1-1/\alpha_i) c_i^{1/\alpha_i}}\right)^{\alpha_i} - \nu_i z$;
    $\theta_i \leftarrow \Gamma(1-1/\alpha_i)(c_i t_i)^{1/\alpha_i}/C$;
  end
  $\eta_{m-1} \leftarrow \sum_{i=1}^{m-1} \theta_i$;
  $\eta_m \leftarrow \eta_m - \sum_{i=1}^{m-1} \theta_i$;
end

The assumption of *Theorem* 4.5 guarantees that the flow with a smaller index should be inserted in front of the flow with a larger index under the equivalent I-PLRU paradigm. In Algorithm 2, the equivalent I-PLRU configuration is calculated recursively. We first decide the last insertion position, i.e., the size of B_M. Then, the problem is reformulated as finding the equivalent I-PLRU configuration for an SLRU paradigm serving $M-1$ data flows. Repeating the same process, we can find the insertion positions for flows $M-1, \cdots, 2$, respectively. The insertion position for flow 1 is just the head of the cache. Detailed explanations and proofs are presented in Section 8.3.

Notably, Algorithm 2 can be simplified if θ_m's are constants and the decay rates of the Zipf's popularity distributions, i.e., $\alpha_1, \alpha_2, \cdots, \alpha_M$, are all different.

Corollary 4.6. *Consider M flows served by an SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$. Assume that θ_m's are constants and the flows are sorted such that $\alpha_i > \alpha_j$, for any $1 \leq i < j \leq M$. We have*

$$\mathcal{S}(\boldsymbol{\theta}; C) \equiv \mathcal{I}(\boldsymbol{\eta}; C),$$

where $\eta_m = \theta_m$, $1 \leq m \leq M$.

The proof is presented in Section 8.4. Corollary 4.6 indicates that for M flows with $\alpha_1 > \alpha_2 > \cdots > \alpha_M$, the I-PLRU paradigm behaves as if the memory block B_m, $1 \leq m \leq M$, only serves flow m, as the total cache size goes to infinity. Note that the equivalent I-PLRU configuration found by Algorithm 2 is more accurate than Corollary 4.6 when the total cache size is relatively small.

In Theorems 4.4 and 4.5, we introduce the methods to find the equivalent SLRU configuration for a given I-PLRU, and vice versa. A remaining question is whether the mapping between equivalent I-PLRU and SLRU configurations is one-to-one or not. In the next theorem, we show that for any I-PLRU configuration $\mathcal{I}(\boldsymbol{\eta}; C)$, the configuration $\mathcal{S}(\boldsymbol{\theta}; C)$ of its equivalent SLRU paradigm is unique, and vice versa.

Theorem 4.7. *Consider the I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta};C)$ and the SLRU paradigm $\mathcal{S}(\boldsymbol{\theta};C)$ that are equivalent (i.e., $\mathcal{I}(\boldsymbol{\eta};C) \equiv \mathcal{S}(\boldsymbol{\theta};C)$). Assume that for $1 \le m \le M$, there exists $\beta_m \in (-1, 0]$ such that $\theta_m \gtrsim C^{\beta_m}$. We have, for any $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\boldsymbol{\eta}}$, as $C \to \infty$*
1) if $\mathcal{I}(\boldsymbol{\eta};C) \equiv \mathcal{S}(\widetilde{\boldsymbol{\theta}};C)$, then $\widetilde{\theta}_m \sim \theta_m$, $1 \le m \le M$;
2) if $\mathcal{S}(\boldsymbol{\theta};C) \equiv \mathcal{I}(\widetilde{\boldsymbol{\eta}};C)$, then either $\widetilde{\eta}_m \sim \eta_m$, or $\lim_{C\to\infty} \widetilde{\eta}_m/\theta_m = \lim_{C\to\infty} \eta_m/\theta_m = 0$, $1 \le m \le M$.

We give an example to help the understanding of case 2) of Theorem 4.7. Consider two flows with the same popularity distributions and request rates, and an SLRU paradigm with $\theta_1 = \theta_2 = 0.5$. Any I-PLRU paradigms with $\eta_1 = o(C)$ are equivalent to the SLRU paradigm, because $X_m(\boldsymbol{\eta};C)$, $1 \le m \le 2$ is almost surely dominated by the number of items of flow m in B_2 as $C \to \infty$. To guarantee the uniqueness, we can simply let $\eta_m = 0$ if any η_m with $\lim_{C\to\infty} \eta_m/\theta_m = 0$ is a solution for the equivalent I-PLRU. Combining Theorems 4.4, 4.5 and 4.7, we know that $F_1(\cdot)$ and $F_2(\cdot)$ define a one-to-one mapping between the equivalent I-PLRU and SLRU configurations in the asymptotic regime. By leveraging this mapping, we find the I-PLRU configuration that optimizes system performance in the following section.

4.3 Optimal I-PLRU Configuration

In this section, we consider the MPM problem under I-PLRU. Our objective is to find the insertion positions that achieve the smallest asymptotic miss probability. The problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & \sum_{m=1}^{M} w_m\, Q_m^{\text{I-PLRU}}(\boldsymbol{\eta};C) \\ \text{subject to} \quad & \eta_m \ge 0, \quad 1 \le m \le M, \\ & \sum_{m=1}^{M} \eta_m = 1. \end{aligned} \tag{5}$$

Let $\boldsymbol{\eta}^*(C)$ denote the optimal solution of Problem (5). We aim to characterize its asymptotic behavior, i.e., $\lim_{C\to\infty} \boldsymbol{\eta}^*(C)$. Before solving the problem, we first show that the problem is non-convex.

Lemma 4.8. *The miss probability $Q_m^{I\text{-}PLRU}(\boldsymbol{\eta};C)$ is a non-convex function with respect to $\boldsymbol{\eta}$. Moreover, as $C \to \infty$, we have*

$$Q_m^{I\text{-}PLRU}(\boldsymbol{\eta};C) \sim Q_m^{SLRU}(F_1(\boldsymbol{\eta};C);C),$$

where $Q_m^{SLRU}(\boldsymbol{\theta};C)$ is asymptotically a convex function with respect to $\boldsymbol{\theta}$ and $F_1(\cdot)$ is the one-to-one mapping defined by Algorithm 1.

The non-convexity can be easily verified by considering the case with $M = 2$. Remarkably, although the MPM problem for I-PLRU is non-convex, it has a special structure, i.e., each term in its objective function is asymptotically a convex function $Q_m^{\text{SLRU}}(\cdot, C)$ in conjunction with a one-to-one mapping function $F_1(\cdot)$. Thus, $\boldsymbol{\eta}^*(C)$ should be the same as the solution of the following problem in the asymptotic regime.

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & \sum_{m=1}^{M} w_m\, Q_m^{\text{SLRU}}(F_1(\boldsymbol{\eta};C);C) \\ \text{subject to} \quad & \eta_m \ge 0, \quad 1 \le m \le M, \\ & \sum_{m=1}^{M} \eta_m = 1. \end{aligned} \tag{6}$$

Furthermore, let $\boldsymbol{\theta} = F_1(\boldsymbol{\eta}; C)$. We have $\boldsymbol{\eta} = F_2(\boldsymbol{\theta}; C)$ by Theorem 4.7. Let $\boldsymbol{\theta}^*(C)$ denote the solution of the following problem,

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{m=1}^{M} w_m \, Q_m^{\text{SLRU}}(\boldsymbol{\theta}; C) \\ \text{subject to} \quad & \theta_m \geq 0, \quad 1 \leq m \leq M, \\ & \sum_{m=1}^{M} \theta_m = 1. \end{aligned} \tag{7}$$

Since $F_1(\cdot)$, $F_2(\cdot)$ are one-to-one mappings, we have $\boldsymbol{\eta}^*(C) \sim F_2\left(\boldsymbol{\theta}^*(C); C\right)$ as $C \to \infty$. Note that Problem (7) is actually the MPM problem under SLRU and is strictly convex (see Section 3.1). Therefore, the asymptotic optimal solution $\lim_{C\to\infty} \boldsymbol{\theta}^*(C)$ is unique. Since $F_2(\cdot)$ is a one-to-one mapping, the asymptotic optimal solution $\lim_{C\to\infty} \boldsymbol{\eta}^*(C)$ of the non-convex problem (5) is also unique. We formally state the relationship between $\boldsymbol{\theta}^*(C)$ and $\boldsymbol{\eta}^*(C)$ in the following theorem.

Theorem 4.9. *Recall that $\boldsymbol{\eta}^*(C)$ is the optimal I-PLRU configuration of Problem (5). We have, as the total cache size $C \to \infty$*

$$\boldsymbol{\eta}^*(C) \sim F_2\left(\boldsymbol{\theta}^*(C); C\right)$$

and for $1 \leq m \leq M$

$$Q_m^{I-PLRU}(\boldsymbol{\eta}^*(C), C) \sim Q_m^{SLRU}(\boldsymbol{\theta}^*(C), C),$$

where $\boldsymbol{\theta}^*(C)$ is the optimal SLRU configuration of Problem (7) and $F_2(\cdot)$ is the one-to-one mapping defined by Algorithm 2.

The poof is a direct application of Theorems 4.3, 4.5 and 4.7. By leveraging the special structure of $Q_m^{\text{I-PLRU}}(\boldsymbol{\eta}; C)$ characterized in Lemma 4.8, we transfer the non-convex problem to a convex problem, and are able to find the optimal I-PLRU configuration based on Equation (4) and Algorithm 2. Notably, although the result is only rigorous in the asymptotic regime, it is still very accurate when the total cache size C is small as shown by Experiment 2 in Section 6. In addition, Theorem 4.9 can be easily extended to general objective functions $\sum_{m=1}^{M} u_m(Q_m^{\pi})$ where $u_m(\cdot)$'s are convex, because the MPM problem under SLRU (i.e., Problem (7)) retains the convexity for such objective functions.

5 DISCUSSIONS ON ENGINEERING ISSUES

Based on our theoretical analysis, in this section, we present heuristic algorithms that build over our analytical investigations to deal with general and unknown popularity distributions in real applications.

5.1 General popularity distributions and non-identical data sizes

Our investigations have focused on the case of the commonly used Zipf's distribution for the popularity profile. However, in general the popularities may not follow a Zipf's distribution. In this section, we address the question of whether we can still identify an I-PLRU configuration that is equivalent to a given SLRU configuration, and vice versa. In particular, by leveraging the characteristic time approximation [12, 14], we propose Algorithms 3, 4 that generalize Algorithms 1, 2 for popularity distributions beyond Zipf's. Let $p_i^{(m)}$, $s_i^{(m)}$ denote the popularity and the size of data item $d_i^{(m)}$, respectively, $i \geq 1$, $1 \leq m \leq M$.

Conjecture 5.1. *Consider M data flows served by an I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$. Let $F_3(\boldsymbol{\eta}; C)$ denote the output of Algorithm 3 with $\boldsymbol{\eta}$ and C as its input. We have*

$$\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(F_3(\boldsymbol{\eta}; C); C).$$

Algorithm 3: Finding the equivalent SLRU for I-PLRU

Output: θ_m, $1 \le m \le M$
Input: η_m, $1 \le m \le M$, C
Initialization: set $\theta_m = 0$, $t_m = 0$, $1 \le m \le M$;
$\theta_1 \leftarrow \eta_1$;
for $m \leftarrow 2$ **to** M **do**
  for $i \leftarrow 1$ **to** $m-1$ **do**
    Solve t_i as the unique solution of

$$\sum_{j\ge 1} s_j^{(i)} \left(1 - \exp\left(-q_j^{(i)} t_i\right)\right) = \theta_i C;$$

  end
  Solve z as the unique solution of

$$\sum_{i=1}^{m} \sum_{j\ge 1} s_j^{(i)} \left(1 - \exp\left(-q_j^{(i)}(t_i + \nu_i z)\right)\right) = \sum_{i=1}^{m} \eta_i C;$$

  for $i \leftarrow 1$ **to** m **do**
    $\theta_i \leftarrow \sum_{j\ge 1} s_j^{(i)} \left(1 - \exp\left(-q_j^{(i)}(t_i + \nu_i z)\right)\right) / C$;
  end
end

Conjecture 5.2. *Consider M flows served by an SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$. Assume without loss of generality that the flows are sorted such that t_m is decreasing with respect to m, where t_m is the unique solution of*

$$\sum_{i\ge 1} s_i^{(m)} \left(1 - \exp\left(-p_i^{(m)} \nu_m t_m\right)\right) = \theta_m C.$$

Let $F_4(\boldsymbol{\theta}; C)$ be the output of Algorithm 4 with $\boldsymbol{\theta}$ and C as its input. We have

$$\mathcal{S}(\boldsymbol{\theta}; C) \equiv \mathcal{I}(F_4(\boldsymbol{\theta}; C); C).$$

Similar to the procedures in Algorithms 1 and 2, we use recursive arguments to find the equivalent SLRU and I-PLRU configurations in Algorithms 3 and 4, respectively. However, the parameters t_m and $t_m + \nu_m z$, $1 \le m \le M$ in Algorithms 3 and 4 are computed using the characteristic time approximation [12, 14] without rigorous accuracy guarantees for general distributions. We validate the accuracy of Algorithms 3 and 4 using real-world traces in Experiment 4. It is observed that the generalized algorithms are not only accurate for popularity distributions beyond Zipf's, but also robust to time correlations among the requests. Note that when the popularities satisfy the Zipf's assumption (i.e., Equation (1)) and the data sizes are 1, Algorithms 3, 4 degenerate to Algorithms 1, 2, respectively.

Algorithm 4: Finding the equivalent I-PLRU for SLRU

Output: η_m, $1 \le m \le M$
Input: θ_m, $1 \le m \le M$, C
Initialization: set $\eta_m = 1$, $1 \le m \le M$;
for $m \leftarrow M$ **to** 2 **do**
  Solve z as the unique solution of

$$\sum_{j\ge 1} s_j^{(m)} \left(1 - \exp\left(-q_j^{(m)} \nu_m z\right)\right) = \theta_m C;$$

  for $i \leftarrow 1$ **to** $m-1$ **do**
    Solve t_i as the unique solution of

$$\sum_{j\ge 1} s_j^{(i)} \left(1 - \exp\left(-q_j^{(i)} t_i\right)\right) = \theta_i C;$$

    $t_i \leftarrow t_i - \nu_i z$;
    $\theta_i \leftarrow \sum_{j\ge 1} s_j^{(i)} \left(1 - \exp\left(-q_j^{(i)} t_i\right)\right)/C$;
  end
  $\eta_{m-1} \leftarrow \sum_{i=1}^{m-1} \theta_i$;
  $\eta_m \leftarrow \eta_m - \sum_{i=1}^{m-1} \theta_i$;
end

5.2 Unknown popularity distributions

In our investigations so far, we have assumed that the popularity distributions of items are known, while the popularities of individual content are unknown. Although this assumption is acceptable in many scenarios, in other real-world applications the popularities of the data could be unknown and time-varying. This motivates us in this section to address the question of how to efficiently find the optimal insertion positions for I-PLRU under unknown popularities. To this end, we next present zeroth and first order methods to incorporate learning of popularity distributions into our design.

Zeroth-order method: A heuristic method is to update the insertion positions along the direction that can potentially reduce the miss probability. Specifically, we can first randomly initialize the insertion positions $\boldsymbol{\eta}^{(0)}$, and evaluate the miss probability of each flow denoted by $Q_m^{(0)}$, $1 \le m \le M$. Next, we randomly update the insertion positions as $\boldsymbol{\eta}^{(1)} = \boldsymbol{\eta}^{(0)} + \Delta\boldsymbol{\eta}^{(0)}$, and evaluate the miss probabilities $Q_m^{(1)}$, $1 \le m \le M$ achieved by the new insertion positions. Then, for $t = 1, 2, 3, \cdots$, let

$$\Delta\boldsymbol{\eta}^{(t)} = \left(\frac{w_1\left(Q_1^{(t)} - Q_1^{(t-1)}\right)}{\Delta\eta_1}, \frac{w_2\left(Q_2^{(t)} - Q_2^{(t-1)}\right)}{\Delta\eta_2}, \cdots, \frac{w_M\left(Q_M^{(t)} - Q_M^{(t-1)}\right)}{\Delta\eta_M}\right),$$

and update

$$\boldsymbol{\eta}^{(t+1)} = \boldsymbol{\eta}^{(t)} - \gamma \cdot \frac{\Delta\boldsymbol{\eta}^{(t)}}{||\Delta\boldsymbol{\eta}^{(t)}||},$$

where γ is the step size and $Q_m^{(t)}$ is the miss probability of flow m achieved by the insertion positions $\boldsymbol{\eta}^{(t)}$. Note that updating $\boldsymbol{\eta}^{(t)}$ along the opposite direction of $\Delta\boldsymbol{\eta}^{(t)}$ can potentially decrease the overall miss probability and get closer to the optimum. In [9], a similar approach is applied to find the optimal SLRU configurations for unknown popularities.

First-order method: Assume that the popularities follow Zipf's distributions with the same decay rate, i.e., $p_i^{(m)} \sim c_m/i^{\alpha}$, $\alpha > 1$, $i \geq 1$, $1 \leq m \leq M$, and the parameters c_m, α are unknown. We are able to estimate the gradient of the objective function and then apply the gradient descent algorithm to find the optimum. Define the gradient for MPM problems under I-PLRU as

$$\nabla_I = \left(\sum_{m=1}^{M} w_m \frac{\partial Q_m^{\text{I-PLRU}}}{\partial \eta_1}, \sum_{m=1}^{M} w_m \frac{\partial Q_m^{\text{I-PLRU}}}{\partial \eta_2}, \cdots, \sum_{m=1}^{M} w_m \frac{\partial Q_m^{\text{I-PLRU}}}{\partial \eta_M} \right).$$

For $t = 0, 1, 2, \cdots$, and the initial insertion position $\boldsymbol{\eta}^{(0)}$, we can update the insertion position as

$$\boldsymbol{\eta}^{(t+1)} = \boldsymbol{\eta}^{(t)} - \gamma \cdot \frac{\nabla_I \big|_{\boldsymbol{\eta}=\boldsymbol{\eta}^{(t)}}}{\left\| \nabla_I \big|_{\boldsymbol{\eta}=\boldsymbol{\eta}^{(t)}} \right\|}, \tag{8}$$

where γ is the step size. The remaining problem is how to estimate the direction of the gradient. Define the gradient for MPM problems under SLRU

$$\nabla_S = \left(w_1 \frac{\partial Q_1^{\text{SLRU}}}{\partial \theta_1}, w_2 \frac{\partial Q_2^{\text{SLRU}}}{\partial \theta_2}, \cdots, w_M \frac{\partial Q_M^{\text{SLRU}}}{\partial \theta_M} \right),$$

and

$$\mathbf{J}_S = \begin{bmatrix} \frac{\partial \theta_1}{\partial \eta_1} & \frac{\partial \theta_1}{\partial \eta_2} & \cdots & \frac{\partial \theta_1}{\partial \eta_M} \\ \frac{\partial \theta_2}{\partial \eta_1} & \frac{\partial \theta_2}{\partial \eta_2} & \cdots & \frac{\partial \theta_2}{\partial \eta_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \theta_M}{\partial \eta_1} & \frac{\partial \theta_M}{\partial \eta_2} & \cdots & \frac{\partial \theta_M}{\partial \eta_M} \end{bmatrix}.$$

The gradient for MPM problems under I-PLRU can be expressed as $\nabla_I = \nabla_S \, \mathbf{J}_S$, if the SLRU paradigm and the I-PLRU paradigm are equivalent. Moreover, recalling Definition 4.2 and Theorem 4.3, we have

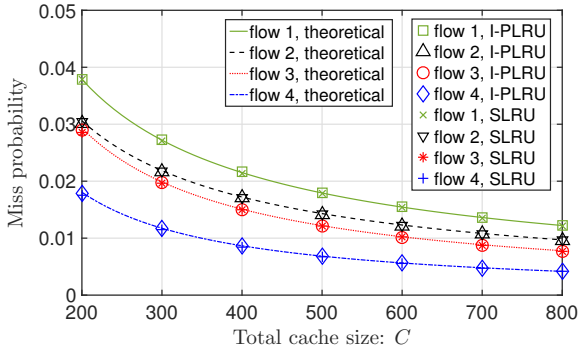
$$\theta_m \approx X_m(\boldsymbol{\eta}; C)/C, \qquad \text{for } 1 \leq m \leq M,$$
$$\nabla_S \approx -(\alpha - 1) \cdot \left(\frac{w_1 Q_1^{\text{I-PLRU}}}{\theta_1}, \frac{w_2 Q_2^{\text{I-PLRU}}}{\theta_2}, \cdots, \frac{w_M Q_M^{\text{I-PLRU}}}{\theta_M} \right).$$

The direction of ∇_S can be approximated by estimating $Q_m^{\text{I-PLRU}}$ (i.e., the miss ratio of flow m), and $X_m(\boldsymbol{\eta}; C)$ (i.e., the cache space occupied by flow m), $1 \leq m \leq M$, even when the parameters α, c_m's are unknown. In addition, the matrix $\mathbf{J}_S$ can be also approximated using such information. Combining the estimation of ∇_S and $\mathbf{J}_S$, we can approximate the direction of ∇_I and adaptively update the insertion position for I-PLRU based on (8).

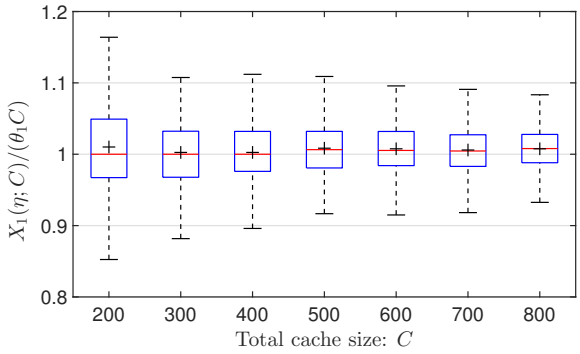
6 EXPERIMENTS

In this section, we conduct four experiments to validate our results as well as to test various metrics-of-interest under our proposed I-PLRU framework.

Experiment 1. In this experiment, we validate the mapping from I-PLRU to the equivalent SLRU by simulating 4 flows served by both paradigms. Let $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1.8, 1.8, 2.0, 2.2)$. Assume that the data set of each flow has 10^6 distinct data items. We have $c_m = 1/\sum_{i=1}^{10^6} i^{-\alpha_m}$. Let $(\nu_1, \nu_2, \nu_3, \nu_4) = (0.1, 0.3, 0.2, 0.4)$. Set the configuration $\boldsymbol{\eta}$ of the I-PLRU paradigm as $(\eta_1, \eta_2, \eta_3, \eta_4) = (0.2, 0.3, 0.2, 0.3)$. Then, we apply Algorithm 1 to calculate the equivalent SLRU configuration $\boldsymbol{\theta}$. We simulate the I-PLRU paradigm and the equivalent SLRU paradigm. The empirical miss probabilities under these two paradigms are plotted in Fig. 6a. It can be observed that the I-PLRU achieves the same miss



(a) Miss probability of I-PLRU.



(b) Cache space occupied by flow 1.

Fig. 6. Four flows served by I-PLRU. It is observed in (a) that equivalent I-PLRU and SLRU paradigms achieve the same miss probability. Moreover, the theoretical result is accurate even when the cache size is relatively small. It is observed in (b) that the ratio of the cache space occupied by flow 1 under I-PLRU to the one under SLRU is more and more concentrated around 1 when the total cache size becomes larger.

probability as its equivalent SLRU, which validates the accuracy of Algorithm 1 even for relatively small cache space (e.g., $C = 200$). We also plot the miss probability calculated by Theorem 4.3. The theoretical results match well with the empirical ones. In addition, we sample $X_1(\boldsymbol{\eta}; C)$ (i.e., the number of items of flow 1 stored in the cache) for 500 times, and plot the quantiles of the samples in Fig. 6b. The box represents the 25^{th} and 75^{th} percentiles. The whiskers extend to the most extreme data points. The red line and the symbol "+" represent the median and the mean, respectively. We can observe that $X_1(\boldsymbol{\eta}; C)/(\theta_1 C)$ is more and more concentrated around 1 as C becomes larger, which directly verifies the equivalence by Definition 4.2. Due to limited space, we omit the similar results of other flows. Note that if we apply Algorithm 2 to compute the equivalent I-PLRU for the SLRU paradigm presented in this experiment, the same result will be obtained, which validates the inverse mapping.

Experiment 2. In this experiment, we optimize I-PLRU and SLRU configurations and compare them with PLRU. Consider 3 data flows with $(\nu_1, \nu_2, \nu_3) = (0.2, 0.3, 0.5)$, $\alpha_1 = \alpha_2 = \alpha_3 = 2$. Assume that the data set of each flow has 10^6 distinct data items. $\boldsymbol{\eta}^* = (0.39, 0.37, 0.24)$, $\boldsymbol{\theta}^* = (0.47, 0.34, 0.19)$. Therefore, we have $c_1 = c_2 = c_3 = 1/\sum_{i=1}^{10^6} i^{-2} = 0.6079$. Assume the system objective is to minimize the overall miss probability $\sum_{m=1}^{M} w_m Q_m^{\pi}$ with $(w_1, w_2, w_3) = (0.6, 0.3, 0.1)$. Applying Theorem 4.9, we obtain the optimal I-PLRU and SLRU configurations. We compared the overall miss probability achieved by the optimal I-PLRU, the optimal SLRU and PLRU in Fig. 7. It can be observed that by

optimizing the insertion positions, I-PLRU significantly improves the performance of conventional PLRU, and achieves the same miss probabilities as the optimal SLRU. Moreover, all empirical results match well with the theoretical ones obtained from Theorem 4.3 and Theorem 4.9.

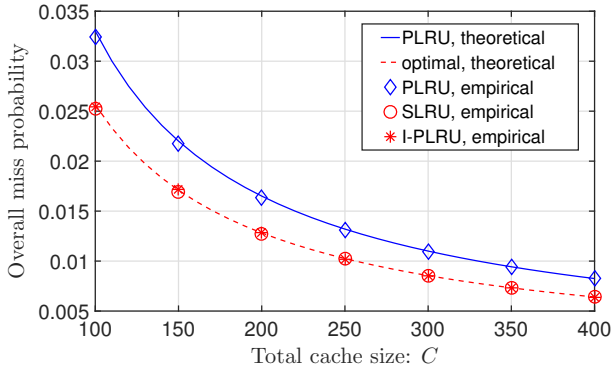


Fig. 7. Optimal performance of I-PLRU. The figure shows that the optimal I-PLRU achieves much better miss probabilities than PLRU.

Experiment 3. In this experiment, we compare I-PLRU and SLRU under the cold-start. Consider two flows with $\alpha_1 = \alpha_2 = 1.2$. Assume that the data set for each flow has 10^6 data items. We have $c_1 = c_2 = 1/\sum_{i=1}^{10^6} i^{-1.2} = 0.1895$. The system objective is to minimize the overall miss probability $\nu_1 Q_1^{\text{SLRU}} + \nu_2 Q_2^{\text{SLRU}}$. Assume the workload has two stages. In Stage 1, the system only serves flow 1, i.e., $\nu_1 = 1, \nu_2 = 0$. To minimize the miss probability, the optimal configurations in Stage 1 are $\boldsymbol{\theta}^* = (1, 0)$ for SLRU and $\boldsymbol{\eta}^* = (1, 0)$ for I-PLRU. Then, after serving 10^5 requests from flow 1, the system enters Stage 2. Assume the arrival rates of two flows are equal in Stage 2, i.e., $\nu_1 = \nu_2 = 0.5$. To retain high efficiency in stationary periods, the configurations should be updated as $\boldsymbol{\theta}^* = (0.5, 0.5)$ and $\boldsymbol{\eta}^* = (0, 1)$. In Fig. 8, we plot the average overall miss probabilities for both

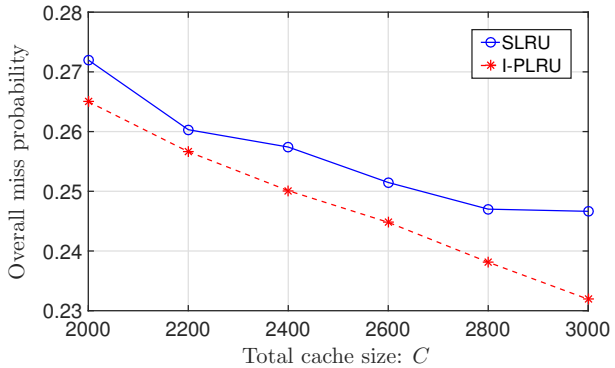


Fig. 8. Comparison of SLRU and I-PLRU performance under cold-start. The figure reveals the robustness of I-PLRU over SLRU.

paradigms during the transient period of SLRU (i.e., the time period when the cache space allocated to flow 2 is not full). Compared with SLRU, I-PLRU achieves lower overall miss probabilities and therefore alleviates the negative impact of the cold-start.

Experiment 4. In this experiment, we test the accuracy of the equivalence mapping under popularity distributions obtained from real-world traces. The trace is collected on a content delivery network and originally used for evaluation in [6]. We use a part of the trace that consists of 10^7

requests accessing 2265308 distinct data items. The data sizes are set to be 1. We randomly distribute the data items into three flows with probabilities $(0.2, 0.3, 0.5)$, and estimate the popularity of each data item by its request frequency. Setting $\boldsymbol{\eta} = (0.1, 0.4, 0.5)$ for I-PLRU, we apply Algorithm 3

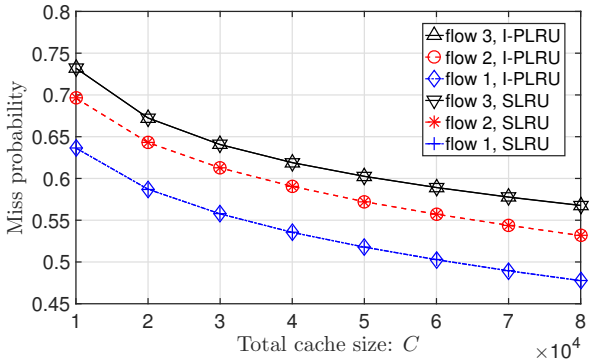


Fig. 9. Equivalent I-PLRU and SLRU evaluated by real-world traces. The figure verifies that the equivalence mapping defined by Algorithms 3 and 4 is accurate for real-world popularity distributions.

to compute the cache space allocation $\boldsymbol{\theta}$ for the equivalent SLRU. We use the same trace to evaluate the miss probability of each flow under the equivalent I-PLRU and SLRU, respectively. The results are plotted in Fig. 9. It can be observed that the miss probabilities achieved by the two paradigms are almost the same, which verifies that the equivalent SLRU configuration calculated by Algorithm 3 is accurate. The experiment can also validate Algorithm 4 since it is the inverse of Algorithm 3. Notably, the data requests in the trace do not follow an exact Zipf's distribution and have correlations over time. The experiment indicates that the equivalence mapping defined by Algorithms 3 and 4 is not only accurate under real-world popularity distributions but also robust to time correlations.

7 CONCLUSION

In this paper, we proposed a new flexible multi-flow LRU cache management paradigm, termed I-PLRU. Unlike, in the traditional SLRU paradigm, in I-PRLU, we do not separate the memory space, thus alleviating the cold-start problem. Further, I-PLRU improves the conventional PLRU by supporting dynamic and refined configurations for individual flows. We rigorously derived the asymptotic miss probability of I-PLRU by establishing an equivalence mapping between I-PLRU and SLRU. We formulated a class of miss probability minimization (MPM) problems for I-PLRU, which turn out to be non-convex. Nonetheless, by leveraging the one-to-one equivalence mapping, we were able to find the optimal I-PLRU configuration. We show that 1) for stationary workload, I-PLRU outperforms PLRU and achieves the same miss probability as the optimal SLRU; 2) for workload with dynamically changing data statistics (e.g., data popularities, request rates), I-PLRU empirically achieves lower miss probabilities than the optimal SLRU by alleviating the cold-start problem.

8 PROOFS

In this section, we provide detailed proofs for our main theorems.

Before investigating the proposed I-PLRU paradigm, we first introduce a two-level caching paradigm shown in Fig. 10 to help the analysis. Consider M flows served by the two-level caching paradigm. The total memory space is separated into M cache partitions with the first $M-1$ partitions organized as the first level and the M^{th} partition organized as the second level. Once a request from

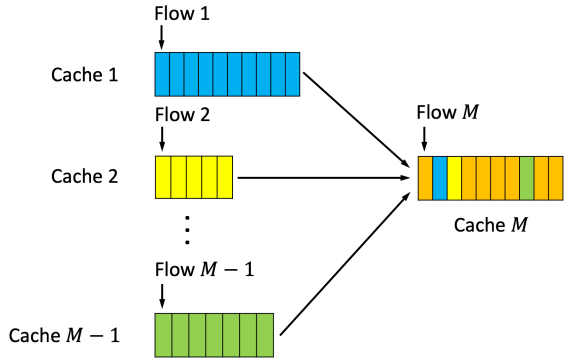


Fig. 10. A two-level caching paradigm $\mathcal{M}(\boldsymbol{\rho}; C)$.

flow m arrives, the requested data will be inserted at the head of cache m, $1 \le m \le M$. And if the cache is full, according to the LRU algorithm, the data at the rear will be evicted. However, different from the SLRU paradigm, the data items evicted from caches $1, \cdots, M-1$ will be inserted to the head of cache M immediately after the eviction. Notably, caches $1, 2, \cdots, M-1$ are LRU caches that serve flows $1, 2, \cdots, M-1$ dedicatedly. And cache M is shared by all data flows. Let C be the total cache size and $\rho_m C$ be the size of cache m, $0 < \rho_m < 1$, $\sum_{m=1}^{M} \rho_m = 1$, $1 \le m \le M$. Let $\mathcal{M}(\boldsymbol{\rho}; C)$ denote the caching paradigm shown in Fig. 10, and $Y_m(\boldsymbol{\rho}; C)$ denote the cache space occupied by flow m in the whole system. As the total cache size $C \to \infty$, we characterize the asymptotic behavior of $Y_m(\boldsymbol{\rho}; C)$ in the following lemma.

Lemma 8.1. *Consider M flows served by the caching paradigm $\mathcal{M}(\boldsymbol{\rho}; C)$. Assume that there exists $\beta \in (-1, 0]$ such that $\rho_M \gtrsim C^{\beta}$ as $C \to \infty$. We have*

$$\frac{Y_m(\boldsymbol{\rho}; C)}{y_m} \xrightarrow{a.s.} 1, \qquad as \;\; C \to \infty, \tag{9}$$

where

$$y_m = \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m + \nu_m z)^{1/\alpha_m} \; for \; 1 \le m \le M, \tag{10}$$

$$t_m = \begin{cases} \left(\frac{\rho_m C}{\Gamma(1-1/\alpha_m) {c_m}^{1/\alpha_m}} \right)^{\alpha_m} & for \; 1 \le m \le M-1, \\ 0 & for \; m = M, \end{cases}$$

and z is the unique solution of

$$\sum_{m=1}^{M} \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m + \nu_m z)^{1/\alpha_m} = C.$$

The proof is presented in Appendix A. Lemma 8.1 shows that, as the total cache size C goes to infinity, $Y_m(\boldsymbol{\rho}; C)$ will be concentrated around y_m almost surely. We will apply Lemma 8.1 to prove the main theorems.

8.1 Proof of Theorem 4.3

Proof. Consider a data flow organized by an LRU cache with a total cache size C. Assume all data items (including the data items that are not stored in the cache) are maintained as a list and sorted according to the last request time. The most recently requested data item is listed at the first position. Note that only the first C data items are stored in the cache.

Consider M data flows organized by the I-PLRU caches. We also maintain a list for each flow, where the data items are sorted according to the last request time. Note that the order of the data items in the list is only determined by the requests and independent with the cache size. A hit of flow m occurs at time τ_0 under the I-PLRU architecture $\mathcal{I}(\boldsymbol{\eta}; C)$, if and only if the requested data is placed at the first $X_m(\boldsymbol{\eta}; C)$ positions in the list of flow m, where $X_m(\boldsymbol{\eta}; C)$ is the number of data items of flow m stored in the I-PLRU cache at time τ_0. Notably, the same request result (hit or miss) will occur at time τ_0 if the flow m is organized by the LRU cache with a cache size $X_m(\boldsymbol{\eta}; C)$, since the list of the data items remains the same.

Assume $\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(\boldsymbol{\theta}; C)$. For any $\epsilon \in (0, 1)$, there exists $C_0(\epsilon)$, such that for all $1 \le m \le M$ and any $C > C_0(\epsilon)$

$$\mathbb{P}\left[(1-\epsilon)\theta_m C \le X_m(\boldsymbol{\eta}; C) \le (1+\epsilon)\theta_m C\right] = 1.$$

Therefore, letting $Q_m^{\text{LRU}}(x)$ denote the miss probability of data flow m organized by a LRU cache with cache space x, we have, for $C > C_0(\epsilon)$

$$Q_m^{\text{LRU}}\left((1-\epsilon)\theta_m C\right) \ge Q_m^{\text{I-PLRU}}(\boldsymbol{\eta}; C) \ge Q_m^{\text{LRU}}\left((1+\epsilon)\theta_m C\right). \tag{11}$$

In addition, according to the result in [16], we have, as $C \to \infty$

$$\begin{aligned} Q_m^{\text{SLRU}}(\boldsymbol{\theta}; C) &= Q_m^{\text{LRU}}(\theta_m C) \\ &\sim \frac{\Gamma(1-1/\alpha_m)^{\alpha_m}}{\alpha_m} \frac{c_m}{(\theta_m C)^{\alpha_m - 1}}. \end{aligned} \tag{12}$$

Combining (11) and (12) finishes the proof. □

8.2 Proof of Theorem 4.4

Proof. First, we prove the theorem under the assumption that for $1 \le m \le M$, there exists $\beta_m \in (-1, 0]$ such that $\eta_m \gtrsim C_m^{\beta}$. Then we will show that the assumption only need to hold for η_M. We use an induction argument to prove the theorem. First, for one flow, the I-PLRU paradigm is exactly the same as the SLRU paradigm with the same cache space, i.e., $\theta_1 = \eta_1$.

Then, assume that we have the equivalence mapping for $M-1$ flows, i.e.,

$$\mathcal{I}\left((\eta_1, \cdots, \eta_{M-1}); C\right) \equiv \mathcal{S}\left((\theta_1^\circ, \cdots, \theta_{M-1}^\circ); C\right), \tag{13}$$

based on which, we will investigate the equivalent SLRU paradigm for M data flows organized by the I-PLRU paradigm $\mathcal{I}((\eta_1, \cdots, \eta_{M-1}, \eta_M); C)$. Assuming

$$\mathcal{I}\left((\eta_1, \cdots, \eta_{M-1}, \eta_M); C\right) \equiv \mathcal{S}\left((\theta_1, \cdots, \theta_M - 1, \theta_M); C\right),$$

we will derive θ_m's as functions of θ_m°'s. Let $\widetilde{X}_i((\eta_1, \cdots, \eta_{M-1}); C)$, $1 \le i \le M-1$ denote the number of data items of flow i stored in the I-PLRU cache $\mathcal{I}((\eta_1, \cdots, \eta_{M-1}); C)$, and $X_j((\eta_1, \cdots, \eta_M); C)$, $1 \le j \le M$, denote the number of data items of flow j in the I-PLRU cache $\mathcal{I}((\eta_1, \cdots, \eta_M); C)$.

The behavior of the I-PLRU paradigm $\mathcal{I}((\eta_1, \cdots, \eta_M); C)$ is the same as the behavior of the caching paradigm $\mathcal{M}((\rho_1, \cdots, \rho_M); C)$ introduced in Fig. 10, where $\rho_i C = \widetilde{X}_i((\eta_1, \cdots, \eta_{M-1}); C)$, $1 \le i \le M-1$, and $\rho_M C = \eta_M C$. Recalling the assumption (13) and Definition 4.2, we have, as $C \to \infty$

$$\frac{\widetilde{X}_i\left((\eta_1, \cdots, \eta_{M-1}); C\right)}{\theta_i^\circ C} \xrightarrow{a.s.} 1. \tag{14}$$

Therefore, for any $\epsilon \in (0, 1)$, there exists $C_0(\epsilon)$ such that for all $C > C_0(\epsilon)$ and $1 \le i \le M-1$

$$(1-\epsilon)\theta_i^\circ C \le \widetilde{X}_i\left((\eta_1, \cdots, \eta_{M-1}); C\right) \le (1+\epsilon)\theta_i^\circ C.$$

Therefore, applying Lemma 8.1, we have for $C > C_0(\epsilon)$,

$$y_m^- \le \theta_m C \le y_m^+, \tag{15}$$

where

$$y_m^+ = \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m^+ + \nu_m z^+)^{1/\alpha_m} \text{ for } 1 \le m \le M,$$

$$y_m^- = \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m^- + \nu_m z^-)^{1/\alpha_m} \text{ for } 1 \le m \le M,$$

$$t_m^+ = \begin{cases} \left(\frac{(1+\epsilon)\theta_m^\circ C}{\Gamma(1-1/\alpha_m){c_m}^{1/\alpha_m}}\right)^{\alpha_m} & \text{for } 1 \le m \le M-1, \\ 0 & \text{for } m = M, \end{cases}$$

$$t_m^- = \begin{cases} \left(\frac{(1-\epsilon)\theta_m^\circ C}{\Gamma(1-1/\alpha_m){c_m}^{1/\alpha_m}}\right)^{\alpha_m} & \text{for } 1 \le m \le M-1, \\ 0 & \text{for } m = M, \end{cases}$$

z^+ is the unique solution of

$$\sum_{m=1}^{M} \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m^+ + \nu_m z^+)^{1/\alpha_m} = (1+\epsilon)C,$$

and z^- is the unique solution of

$$\sum_{m=1}^{M} \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m^- + \nu_m z^-)^{1/\alpha_m} = (1-\epsilon)C.$$

Define

$$y_m = \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m + \nu_m z)^{1/\alpha_m} \text{ for } 1 \le m \le M,$$

$$t_m = \begin{cases} \left(\frac{\theta_m^\circ C}{\Gamma(1-1/\alpha_m){c_m}^{1/\alpha_m}}\right)^{\alpha_m} & \text{for } 1 \le m \le M-1, \\ 0 & \text{for } m = M, \end{cases}$$

and z is the unique solution of

$$\sum_{m=1}^{M} \Gamma\left(1 - 1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m + \nu_m z)^{1/\alpha_m} = C.$$

We have, for $1 \le m \le M$

$$\lim_{C\to\infty} \frac{y_m^-}{y_m} = \lim_{C\to\infty} \frac{y_m^+}{y_m} = 1. \tag{16}$$

Recalling (15), we have, for $1 \le m \le M$

$$\lim_{C\to\infty} \frac{\theta_m C}{y_m} = 1.$$

So far, we derive the equivalent I-PLRU paradigm for M data flows.

Using the induction argument, we can calculate the equivalent I-PLRU paradigm for any SLRU paradigm. This induction argument is summarized as Algorithm 1.

Notably, if for $1 \le m \le M-1$, $\eta_m C \sim l_m(C)$, where $l_m(\cdot)$'s are slowly varying functions that satisfy $\lim_{x\to\infty} l_m(bx)/l(x) = 1$ for any positive constant b, then the cache space occupied by each

flow in B_m's can be ignored compared with $\theta_m C$ as $C \to \infty$. Therefore, the result still holds for such η_m's. □

8.3 Proof of Theorem 4.5

Proof. We assume that the flow indices are sorted such that

$$\frac{(\theta_{m_1} C)^{\alpha_{m_1}}}{\Gamma(1-1/\alpha_{m_1})^{\alpha_{m_1}} c_{m_1} \nu_{m_1}} < \frac{(\theta_{m_2} C)^{\alpha_{m_2}}}{\Gamma(1-1/\alpha_{m_2})^{\alpha_{m_2}} c_{m_2} \nu_{m_2}}, \tag{17}$$

for any $1 \le m_1 < m_2 \le M$. We will first prove that the insertion position of flow m_1 is in front of the insertion position of flow m_2 for $1 \le m_1 < m_2 \le M$.

Consider 2 flows (i.e., flow 1 and flow 2) organized by an I-PLRU cache $\mathcal{I}((\tilde{\eta}_1, \tilde{\eta}_2); C)$, where flow 1 is inserted at the head of the cache and flow 2 is inserted at $\tilde{\eta}_1 C + 1$. Theorem 4.4 implies that the I-PLRU paradigm is equivalent to the SLRU paradigm $\mathcal{S}((\tilde{\theta}_1, \tilde{\theta}_2); C)$ where

$$\begin{aligned}
\tilde{\theta}_m C &= \Gamma\left(1-1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_m + \nu_m z)^{1/\alpha_m}, \\
t_1 &= \left(\frac{\tilde{\eta}_1 C}{\Gamma(1-1/\alpha_1) {c_1}^{1/\alpha_1}} \right)^{\alpha_1}, \\
t_2 &= 0,
\end{aligned}$$

and z is the unique solution of

$$\Gamma\left(1-1/\alpha_m\right) {c_m}^{1/\alpha_m} (\nu_1 z)^{1/\alpha_m} + \Gamma\left(1-1/\alpha_m\right) {c_m}^{1/\alpha_m} (t_2 + \nu_2 z)^{1/\alpha_m} = C.$$

Thus, we have

$$\begin{aligned}
\frac{(\tilde{\theta}_1 C)^{\alpha_1}}{\Gamma(1-1/\alpha_1)^{\alpha_1} c_1 \nu_1} &= t_1/\nu_1 + z \\
&> t_2/\nu_2 + z = \frac{(\tilde{\theta}_2 C)^{\alpha_2}}{\Gamma(1-1/\alpha_2)^{\alpha_2} c_2 \nu_2}.
\end{aligned} \tag{18}$$

Note that the inequality (18) is sufficient to guarantee that flow 1 is inserted in front of flow 2. For M flows organized by the I-PLRU cache, the inequality (18) still holds. Therefore, under the assumption (17), the flows are sorted such that the insertion position of flow m_1 is in front of the insertion position of flow m_2 for $1 \le m_1 < m_2 \le M$.

Next, we will use an induction argument to find the equivalent I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$. In each iteration, there are two steps. In iteration 1, we will decide the insertion position for flow M. Given the SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$, using Lemma 8.1, we can find the equivalent $\mathcal{M}(\boldsymbol{\rho}^{(1)}; C)$ paradigm, where

$$\begin{aligned}
\rho_m^{(1)} C &= \Gamma\left(1-1/\alpha_m\right) {c_m}^{1/\alpha_m} {t_m}^{1/\alpha_m} \quad \text{for } 1 \le m \le M-1, \\
\rho_M^{(1)} C &= C - \sum_{m=1}^{M-1} \rho_m^{(1)} C, \\
t_m &= \begin{cases} \left(\frac{\theta_m C}{\Gamma(1-1/\alpha_m) {c_m}^{1/\alpha_m}} \right)^{\alpha_m} - \nu_m z & \text{for } 1 \le m \le M-1, \\ 0 & \text{for } m = M, \end{cases} \\
z &= \frac{(\theta_M C)^{\alpha_M}}{\Gamma(1-1/\alpha_M)^{\alpha_M} c_M \nu_M}.
\end{aligned}$$

This is the first step. The second step for iteration 1 is simply letting $\eta_M = \rho_M$, i.e., the size of B_M is $\rho_M C$.

In iteration 2, we will decide the insertion position for flow $M-1$. Notably, in the caching system $\mathcal{M}(\boldsymbol{\rho}^{(1)}; C)$, the caches $1, 2, \cdots, M-1$ can be viewed as a new SLRU system $\mathcal{S}(\boldsymbol{\theta}^{(2)}; C^{(2)})$, where for $1 \le m \le M-1$,

$$\theta_m^{(2)} = \frac{\rho_m^{(2)}}{\sum_{i=1}^{M-1} \rho_i^{(2)}}, \qquad C^{(2)} = C - \rho_M C.$$

The first step is to find the equivalent $\mathcal{M}(\boldsymbol{\rho}^{(2)}; C^{(2)})$ paradigm for $\mathcal{S}(\boldsymbol{\theta}^{(2)}; C^{(2)})$ based on Lemma 8.1. The second step is to construct the system shown in Fig. 11, where the first $M-2$ flows are served by $M-2$ separated LRU caches, and flows $M-1$, M are served by an I-PLRU paradigm. Let the

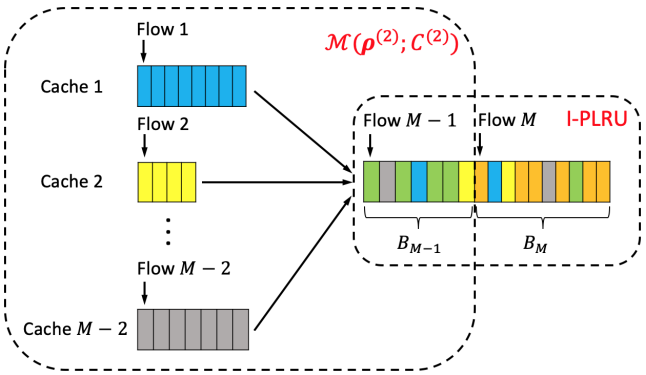


Fig. 11. Identifying the insertion position for flow $M-1$.

cache space of cache m be $\rho_m^{(2)} C^{(2)}$, $1 \le m \le M-2$. Let the cache space of the blocks B_{M-1}, B_M in the I-PLRU paradigm be $\rho_{M-1}^{(2)} C^{(2)}$ and $\rho_M^{(1)} C$, respectively. Applying Lemma 8.1, we can prove that the caching system shown in Fig. 11 is equivalent to the original SLRU system $\mathcal{S}(\boldsymbol{\theta}; C)$.

So on so forth, repeating these two steps for $M-2$ more iterations, we eventually find the I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$ that is equivalent to the original SLRU architecture $\mathcal{S}(\boldsymbol{\theta}; C)$, where

$$\eta_m = \rho_m^{(M-m+1)} C^{(M-m+1)}/C, \qquad C^{(1)} = C.$$

This induction process is summarized as Algorithm 2. □

8.4 Proof of Corollary 4.6

Proof. In Algorithm 2, a critical step is to update t_i, $1 \le i \le m-1$, as

$$t_i = \left(\frac{\theta_i C}{\Gamma(1-1/\alpha_i) c_i{}^{1/\alpha_i}} \right)^{\alpha_i} - \nu_i z,$$

where

$$z = \left(\frac{\theta_m C}{\Gamma(1-1/\alpha_m)(c_m \nu_m)^{1/\alpha_m}} \right)^{\alpha_m}.$$

If we have $\alpha_m < \alpha_i$, then, as $C \to \infty$

$$t_i \sim \left(\frac{\theta_i C}{\Gamma(1-1/\alpha_i) c_i{}^{1/\alpha_i}} \right)^{\alpha_i}. \tag{19}$$

Combining (19) with the remaining steps of Algorithm 2, we prove Corollary 4.6. □

8.5 Proof of Theorem 4.7

Proof. Consider equivalent I-PLRU paradigm $\mathcal{I}(\boldsymbol{\eta}; C)$ and SLRU paradigm $\mathcal{S}(\boldsymbol{\theta}; C)$. First, assume towards contradiction that $\mathcal{I}(\boldsymbol{\eta}; C) \equiv \mathcal{S}(\widetilde{\boldsymbol{\theta}}; C)$, with $\lim_{C\to\infty} \widetilde{\theta}_m/\theta_m \neq 1$ for some m. Then, applying Theorem 4.3, we know that the asymptotic miss probability achieved by $\mathcal{S}(\boldsymbol{\theta}; C)$ and $\mathcal{S}(\widetilde{\boldsymbol{\theta}}; C)$ are different. Since I-PLRU achieves the same asymptotic miss probability as its equivalent SLRU paradigm, $\mathcal{S}(\boldsymbol{\theta}; C)$ and $\mathcal{S}(\widetilde{\boldsymbol{\theta}}; C)$ cannot be both equivalent to $\mathcal{I}(\boldsymbol{\eta}; C)$. We have a contradiction and therefore prove $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

Then, assume $\mathcal{S}(\boldsymbol{\theta}; C) \equiv \mathcal{I}(\boldsymbol{\eta}; C)$ and $\mathcal{S}(\boldsymbol{\theta}; C) \equiv \mathcal{I}(\widetilde{\boldsymbol{\eta}}; C)$. Recall the proof of Theorem 4.5. The equivalent I-PLRU configuration is obtained by constructing a two-level caching paradigm $\mathcal{M}(\boldsymbol{\rho}, C)$ (shown in Fig. 10) that is equivalent to $\mathcal{S}(\boldsymbol{\theta}; C)$. Moreover, according to Lemma 8.1, if there exist two caching paradigms $\mathcal{M}(\boldsymbol{\rho}, C)$ and $\mathcal{M}(\widetilde{\boldsymbol{\rho}}, C)$ that are both equivalent to $\mathcal{S}(\boldsymbol{\theta}; C)$. Then we must have either $\lim_{C\to\infty} \widetilde{\rho}_m/\rho_m = 1$, or $\lim_{C\to\infty} \widetilde{\rho}_m/\theta_m = \lim_{C\to\infty} \rho_m/\theta_m = 0$, $1 \leq m \leq M$. Note that ρ_M indicates the last insertion position of the equivalent I-PLRU. Thus, we have either $\lim_{C\to\infty} \widetilde{\eta}_M/\eta_M = 1$. Applying the recursive argument used in Section 8.3, we can prove the theorem. □

REFERENCES

[1] Memcached. http://memcached.org/.

[2] Berk Atikoglu, Yuehai Xu, Eitan Frachtenberg, Song Jiang, and Mike Paleczny. 2012. Workload analysis of a large-scale key-value store. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40. ACM, 53–64.

[3] Muhammad Abdullah Awais. 2016. Memory management: Challenges and techniques for traditional memory allocation algorithms in relation with today's real time needs. *Advances in Computer Science: an International Journal* 5, 2 (2016), 22–27.

[4] Sorav Bansal and Dharmendra S Modha. 2004. CAR: Clock with adaptive replacement.. In *FAST*, Vol. 4. 187–200.

[5] Nathan Beckmann, Haoxian Chen, and Asaf Cidon. 2018. LHD: Improving cache hit rate by maximizing hit density. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. USENIX Association.

[6] Daniel S Berger, Ramesh K Sitaraman, and Mor Harchol-Balter. 2017. AdaptSize: Orchestrating the hot object memory cache in a content delivery network.. In *NSDI*. 483–498.

[7] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[8] Jacob Brock, Chencheng Ye, Chen Ding, Yechen Li, Xiaolin Wang, and Yingwei Luo. 2015. Optimal cache partition-sharing. In *2015 44th International Conference on Parallel Processing (ICPP)*. IEEE, 749–758.

[9] Weibo Chu, Mostafa Dehghan, Don Towsley, and Zhi-Li Zhang. 2016. On allocating cache resources to content providers. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. ACM, 154–159.

[10] Malcolm C Easton and Ronald Fagin. 1978. Cold-start vs. warm-start miss ratios. *Commun. ACM* 21, 10 (1978), 866–872.

[11] Gil Einziger, Roy Friedman, and Ben Manes. 2017. TinyLFU: A highly efficient cache admission policy. *ACM Transactions on Storage (TOS)* 13, 4 (2017), 35.

[12] Ronald Fagin. 1977. Asymptotic miss ratios over independent references. *J. Comput. System Sci.* 14, 2 (1977), 222–250.

[13] Nicolas Gast and Benny Van Houdt. 2015. Transient and steady-state regime of a family of list-based cache replacement algorithms. *ACM SIGMETRICS Performance Evaluation Review* 43, 1 (2015), 123–136.

[14] Nicolas Gast and Benny Van Houdt. 2017. TTL approximations of the cache replacement algorithms LRU(m) and h-LRU. *Performance Evaluation* 117 (2017), 33–57.

[15] Ryo Hirade and Takayuki Osogami. 2010. Analysis of page replacement policies in the fluid limit. *Operations research* 58, 4-part-1 (2010), 971–984.

[16] Predrag R. Jelenković. 1999. Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities. *The Annals of Applied Probability* 2 (1999), 430–464.

[17] Predrag R Jelenković and Xiaozhu Kang. 2007. LRU caching with moderately heavy request distributions. In *2007 Proceedings of the Fourth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*. SIAM, 212–222.

[18] Kaiyi Ji, Guocong Quan, and Jian Tan. 2018. Asymptotic miss ratio of LRU caching with consistent hashing. In *IEEE Conference on Computer Communications (INFOCOM 2018)*. Honolulu, USA.

[19] Song Jiang, Feng Chen, and Xiaodong Zhang. 2005. CLOCK-Pro: An effective improvement of the CLOCK replacement. In *USENIX Annual Technical Conference, General Track*. 323–336.

[20] Song Jiang and Xiaodong Zhang. 2002. LIRS: An efficient low inter-reference recency set replacement policy to improve buffer cache performance. *ACM SIGMETRICS Performance Evaluation Review* 30, 1 (2002), 31–42.

[21] Mark S Johnstone and Paul R Wilson. 1998. The memory fragmentation problem: Solved?. In *ACM Sigplan Notices*, Vol. 34. ACM, 26–36.
[22] Conglong Li and Alan L Cox. 2015. GD-Wheel: A cost-aware replacement policy for key-value stores. In *Tenth European Conference on Computer Systems*. ACM, 5.
[23] Nimrod Megiddo and Dharmendra S Modha. 2003. ARC: A self-tuning, low overhead replacement cache. In *FAST*, Vol. 3. 115–130.
[24] Elizabeth J O'neil, Patrick E O'neil, and Gerhard Weikum. 1993. The LRU-K page replacement algorithm for database disk buffering. *ACM Sigmod Record* 22, 2 (1993), 297–306.
[25] Guocong Quan, Kaiyi Ji, and Jian Tan. 2018. LRU caching with dependent competing requests. In *IEEE Conference on Computer Communications (INFOCOM 2018)*. Honolulu, USA.
[26] Guocong Quan, Jian Tan, and Atilla Eryilmaz. 2019. Counterintuitive characteristics of optimal distributed LRU caching over unreliable channels. In *IEEE Conference on Computer Communications (INFOCOM 2019)*. Paris, France.
[27] Yannis Smaragdakis, Scott Kaplan, and Paul Wilson. 1999. EELRU: Simple and effective adaptive page replacement. In *ACM SIGMETRICS Conference on Measuring and Modeling of Computer Systems*. ACM, 122–133.
[28] Jian Tan, Guocong Quan, Kaiyi Ji, and Ness Shroff. 2018. On resource pooling and separation for LRU caching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 5.
[29] Andrew S. Tanenbaum. 2001. *Modern operating systems* (2rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.
[30] Paul R Wilson, Mark S Johnstone, Michael Neely, and David Boles. 1995. Dynamic storage allocation: A survey and critical review. In *Memory Management*. Springer, 1–116.
[31] Yue Yang and Jianwen Zhu. 2016. Write skew and Zipf distribution: evidence and implications. *ACM Transactions on Storage (TOS)* 12, 4 (2016), 21.

A PROOF OF LEMMA 8.1

Before presenting the proof, we introduce some additional concepts and notations. Since the requests are all independent, it is sufficient to prove the result for a given time (saying τ_0) after the system reaches its stationarity. Consider the two-level caching framework $\mathcal{M}(\boldsymbol{\rho}; C)$ introduced in Section 8. For $1 \le m \le M$, $n \in \mathbb{N}$, define

$$V_i^{(m)}(n) = \begin{cases} 1 & \text{if data } d_i^{(m)} \text{ is requested during } [\tau_{-n}, \tau_{-1}], \\ 0 & \text{otherwise.} \end{cases}$$

Define, for $0 \le m \le M-1$, $n \in \mathbb{N}$,

$$W_i^{(m)}(n) = \begin{cases} 1 & \text{if data } d_i^{(m)} \text{ is not stored in cache } m \text{ during } [\tau_{-(n+1)}, \tau_{-n}), \\ 0 & \text{otherwise.} \end{cases}$$

and $W_i^{(M)}(n) = 1$. Let

$$\omega = \min\left\{ n : \sum_{m=1}^{M} \sum_{i \ge 1} V_i^{(m)}(n) W_i^{(m)}(n) = \rho_M C \right\}. \tag{20}$$

Note that the data items stored in cache M at time τ_0 are determined by the requests during $[\tau_{-\omega}, \tau_{-1}]$ and independent with the requests before $\tau_{-\omega}$. We have, for $1 \le m \le M$

$$Y_m(\boldsymbol{\rho}; C) = \rho_m C + \sum_{i \ge 1} V_i^{(m)}(\omega) W_i^{(m)}(\omega).$$

Notably, $\sum_{i \ge 1} V_i^{(m)}(\omega) W_i^{(m)}(\omega)$ is the total size of distinct data items that are requested by flow m during $[\tau_{-\omega}, \tau_{-1}]$ and not stored in cache m right before $\tau_{-\omega}$.

Define $S_m(n)$ as the total size of distinct data items of n requests from flow m for $1 \le m \le M$. Define $T_m = \min\{n : S_m(n) = \rho_m C\}$ for $1 \le m \le M-1$ and $T_M = 0$. Let $n_m = \sum_{i=-\omega}^{-1} \mathbf{1}_{\{I_i = m\}}$ denote the number of requests that are from flow m during $[\tau_{-\omega}, \tau_{-1}]$, $1 \le m \le M$. Since the requests are all independent, we have

$$Y_m(\boldsymbol{\rho}; C) = \sum_{i \ge 1} V_i^{(m)}(\omega) W_i^{(m)}(\omega) + \rho_m C \overset{d}{=} S_m(T_m + n_m), \tag{21}$$

given the condition that $\sum_{m=1}^{M} S_m(T_m + n_m) = C$, where $X \overset{d}{=} Y$ denotes that the random variables X and Y have the same probability distribution.

Define, for $1 \le m \le M$,

$$s_m(n) = \Gamma(1 - 1/\alpha_m)\, {c_m}^{1/\alpha_m} n^{1/\alpha_m}. \tag{22}$$

Recalling (10), we have $y_m = s_m(t_m + \nu_m z)$ and $\sum_{m=1}^{M} s_m(t_m + \nu_m z) = C$. Before proving Lemma 8.1, we first establish the following lemma showing that $S_m(n)$ is concentrated around $s_m(n)$ with high probability when n is large.

Lemma A.1. *For $\epsilon \in (0,1)$, there exists a constant $N_m(\epsilon)$ that for all $n > N_m(\epsilon)$ and $1 \le m \le M$,*

$$\mathbb{P}[S_m(n) \ge (1+\epsilon) s_m(n)] \le \exp(-\epsilon^2 s_m(n)/36).$$

Proof. First, we will show that as $n \to \infty$, $\mathbb{E}[S_m(n)] \sim s_m(n)$. Recalling the definition of $S_m(n)$, we have

$$\mathbb{E}[S_m(n)] = \sum_{i=1}^{\infty} \left(1 - \left(1 - p_i^{(m)}\right)^n\right).$$

As $n \to \infty$, we have

$$\begin{aligned}\sum_{i=1}^{\infty}\left(1-\left(1-p_i^{(m)}\right)^n\right) &\sim \sum_{i=1}^{\infty}\left(1-\left(1-\frac{c_m}{i^{\alpha_m}}\right)^n\right)\\ &\sim \sum_{i=1}^{\infty}\left(1-\exp\left(\frac{c_m n}{i^{\alpha_m}}\right)\right)\\ &\sim \int_1^{\infty}\left(1-\exp\left(\frac{c_m n}{t^{\alpha_m}}\right)\right)dt\\ &\sim \Gamma\left(1-1/\alpha_m\right){c_m}^{1/\alpha_m}n^{1/\alpha_m}\\ &= s_m(n),\end{aligned}$$

which implies $\mathbb{E}[S_m(n)] \sim s_m(n)$, i.e., $\lim_{n\to\infty}\mathbb{E}[S_m(n)]/s_m(n) = 1$.

Therefore, for any $\epsilon \in (0,1)$, there always exists $N_m(\epsilon)$ such that for all $n > N_m(\epsilon)$ $\mathbb{E}[S_m(n)] \le (1+\epsilon/2)s_m(n)$. Therefore, for $n > N_m(\epsilon)$, we have

$$\begin{aligned}\mathbb{P}\left[S_m(n) \ge (1+\epsilon)s_m(n)\right] &\le \mathbb{P}\left[S_m(n) \ge \frac{1+\epsilon}{1+\epsilon/2}\mathbb{E}[S_m(n)]\right]\\ &\le \mathbb{P}\left[S_m(n) \ge (1+\epsilon/3)\mathbb{E}[S_m(n)]\right]\end{aligned}$$

Then, applying Lemma 7.1 in [28], we complete the proof. □

Now we are ready to prove Lemma 8.1.

Proof. Recalling (10), (21) and (22), in order to prove (9), it is sufficient to show

$$\frac{S_m(T_m+n_m)}{s_m(t_m+\nu_m z)} \xrightarrow{a.s.} 1, \quad \text{as } C \to \infty. \tag{23}$$

To prove this, we will first show

$$\frac{S_m(T_m+n_m)}{s_m(t_m+\nu_m \omega)} \xrightarrow{a.s.} 1, \quad \text{as } C \to \infty. \tag{24}$$

We need prove that for any $\epsilon \in (0,1)$, the events

$$\{S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega) \mid \text{the total cache space is } C\}_{C=1}^{\infty}$$

and

$$\{S_m(T_m+n_m) < (1-\epsilon)s_m(t_m+\nu_m\omega) \mid \text{the total cache space is } C\}_{C=1}^{\infty}$$

are not infinitely often (i.o.) almost surely, i.e.,

$$\mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega) \text{ i.o. } \mid \text{the total cache space is } C\right] = 0, \tag{25}$$

$$\mathbb{P}\left[S_m(T_m+n_m) < (1-\epsilon)s_m(t_m+\nu_m\omega) \text{ i.o. } \mid \text{the total cache space is } C\right] = 0. \tag{26}$$

In order to prove $\mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega) \text{ i.o. } \mid \text{the total cache space is } C\right] = 0$, we will show

$$\sum_{C=1}^{\infty}\mathbb{P}\left[A_C\right] < \infty \tag{27}$$

and then apply the Borel-Cantelli lemma, where

$$A_C \triangleq \{S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega) \mid \text{the total cache size is } C\}.$$

In the rest of the proof, we always assume that the total cache size is C and do not write it as the condition for simplicity. We first prove the lemma with the assumption that for $1 \le m \le M$, there

exists $\beta_m \in (-1, 0]$ such that $\rho_m \gtrsim C^{\beta_m}$. Then we will show that the result is still correct if the first $M-1$ flows do not satisfy this assumption.

Define $\mathcal{E}_1^+ = \{n_m > (1+\epsilon/2)\nu_m\omega\}$ and $\mathcal{E}_1^- = \{n_m < (1-\epsilon/2)\nu_m\omega\}$. We can bound $\mathbb{P}[A_C]$ by

$$\begin{aligned}\mathbb{P}[A_C] &= \mathbb{P}[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega)] \\ &\le \mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\quad + \mathbb{P}\left[\mathcal{E}_1^+\right] + \mathbb{P}\left[\mathcal{E}_1^-\right], \end{aligned} \tag{28}$$

where $\mathcal{E}_1^{+c}$ and $\mathcal{E}_1^{-c}$ denote the complements of $\mathcal{E}_1^+$ and $\mathcal{E}_1^-$, respectively.

The remaining proof for (27) consists of two steps. We will first derive an upper bound for $\mathbb{P}[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega)|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}]$ in Step 1, and then derive upper bounds for $\mathbb{P}[\mathcal{E}_1^+]$ and $\mathbb{P}[\mathcal{E}_1^-]$ in Step 2.

Step 1: $\mathbb{P}[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega)|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}]$ can be upper bounded as

$$\begin{aligned}&\mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+\nu_m\omega)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\le \mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon)s_m(t_m+n_m/(1+\epsilon/2))\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &= \mathbb{P}\left[S_m(T_m+n_m) > \frac{1+\epsilon}{(1+\epsilon/2)^{1/\alpha_m}}s_m((1+\epsilon/2)t_m+n_m)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\le \mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon/3)s_m((1+\epsilon/2)t_m+n_m)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\le \mathbb{P}\left[S_m(T_m+n_m) > (1+\epsilon/3)s_m((1+\epsilon/2)t_m+n_m)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}, T_m < (1+\epsilon/2)t_m\right] \\ &\quad + \mathbb{P}\left[T_m \ge (1+\epsilon/2)t_m\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &= \mathbb{P}\left[S_m((1+\epsilon/2)t_m+n_m) > (1+\epsilon/3)s_m((1+\epsilon/2)t_m+n_m)\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\quad + \mathbb{P}\left[T_m \ge (1+\epsilon/2)t_m\middle|\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\right] \\ &\triangleq I_1 + I_2.\end{aligned}$$

Note that since n_m is a random variable, we cannot directly use Lemma A.1 to bound I_1. Given $\{\mathcal{E}_1^{+c}\cap\mathcal{E}_1^{-c}\}$, we have

$$\begin{aligned}s_m((1+\epsilon/2)t_m+n_m) &> s_m((1+\epsilon/2)t_m+(1-\epsilon/2)n_m) \\ &> s_m((1-\epsilon)(t_m+\nu_m\omega)).\end{aligned}$$

Recalling (20), we have $\omega > \rho_M C$. Lemma A.1 implies that there always exists $C_{1,m}(\epsilon)$ such that for all $C > C_{1,m}(\epsilon)$,

$$\begin{aligned}I_1 &\le \exp(-\epsilon^2 s_m((1-\epsilon)(t_m+\nu_m\rho_M C))/324) \\ &= \exp\left(-\epsilon^2(1-\epsilon)^{1/\alpha_m}s_m(t_m+\nu_m\rho_M C)/324\right) \\ &\le \exp\left(-\epsilon^2(1-\epsilon)^{1/\alpha_m}s_m(\nu_m\rho_M C)/324\right) \\ &= \exp\left(-\frac{\epsilon^2(1-\epsilon)^{1/\alpha_m}}{324}\Gamma\left(1-\frac{1}{\alpha_m}\right)(c_m\nu_m\rho_M C)^{1/\alpha_m}\right).\end{aligned} \tag{29}$$

Similarly, I_2 can be upper bounded by Lemma A.1. Recall $S_m(T_m) = s_m(t_m) = \rho_m C$. Lemma A.1 yields that there exists $C_{2,m}(\epsilon)$ such that for all $C > C_{2,m}(\epsilon)$

$$\begin{aligned} I_2 &= \mathbb{P}\left[S_m((1+\epsilon/2)t_m) \le S_m(T_m) \middle| \mathcal{E}_1^{+c} \cap \mathcal{E}_1^{-c}\right] \\ &= \mathbb{P}\left[S_m((1+\epsilon/2)t_m) \le \rho_m C\right] \\ &= \mathbb{P}\left[S_m((1+\epsilon/2)t_m) \le (1+\epsilon/2)^{-1/\alpha_m} s_m((1+\epsilon/2)t_m)\right] \\ &\le \exp\left(-((1+\epsilon/2)^{1/\alpha_m} - 1)^2 s_m((1+\epsilon/2)t_m)/36\right) \\ &\le \exp\left(-((1+\epsilon/2)^{1/\alpha_m} - 1)^2 s_m(t_m)/36\right) \\ &\le \exp\left(-\frac{\epsilon^2 \rho_m C}{324{\alpha_m}^2}\right). \end{aligned} \tag{30}$$

Combining (29) and (30) implies that, for any $\epsilon \in (0,1)$ and $C > \max\{C_{k,m}(\epsilon) : 1 \le k \le 2\}$,

$$\begin{aligned} &\mathbb{P}\left[S_m(T_m + n_m) > (1+\epsilon) s_m(t_m + \nu_m z) \middle| \mathcal{E}_1^{+c} \cap \mathcal{E}_1^{-c}\right] \\ &\le \exp\left(-\frac{\epsilon^2(1-\epsilon)^{1/\alpha_m}}{324}\Gamma\left(1 - \frac{1}{\alpha_m}\right)(c_m \nu_m \rho_M C)^{1/\alpha_m}\right) + \exp\left(-\frac{\epsilon^2 \rho_m C}{324{\alpha_m}^2}\right). \end{aligned} \tag{31}$$

Up to now, we finish Step 1.

Step 2: To complete the proof, we will derive upper bounds for $\mathbb{P}[\mathcal{E}_1^-]$ and $\mathbb{P}[\mathcal{E}_1^+]$ in Step 2. Note that $\mathbb{E}[n_m|\omega] = \nu_m \omega$. Applying the Chernoff bound and the fact that $\omega > \rho_M C$, we have

$$\mathbb{P}\left[\mathcal{E}_1^-\right] = \mathbb{P}\left[n_m < (1-\epsilon/2)\nu_m \omega\right] \le \exp\left(-\frac{\epsilon^2 \nu_m \rho_M C}{8}\right) \tag{32}$$

and

$$\mathbb{P}\left[\mathcal{E}_1^+\right] = \mathbb{P}\left[n_m > (1+\epsilon/2)\nu_m \omega\right] \le \exp\left(-\frac{\epsilon^2 \nu_m \rho_M C}{8}\right). \tag{33}$$

Combining (28), (31), (32) and (33) implies that, for any $\epsilon \in (0,1)$ and $C > \max\{C_{k,m}(\epsilon) : 1 \le k \le 2\}$,

$$\begin{aligned} \mathbb{P}[A_C] \le &\exp\left(-\frac{\epsilon^2(1-\epsilon)^{1/\alpha_m}}{324}\Gamma\left(1 - \frac{1}{\alpha_m}\right)(c_m \nu_m \rho_M C)^{1/\alpha_m}\right) \\ &+ \exp\left(-\frac{\epsilon^2 \rho_m C}{324{\alpha_m}^2}\right) + 2\exp\left(-\frac{\epsilon^2 \nu_m \rho_M C}{8}\right). \end{aligned}$$

Recall the assumption that for $1 \le m \le M$, there exists $\beta_m \in (-1, 0]$ such that $\rho_m \gtrsim C^{\beta_m}$. We have $\sum_{C=1}^{\infty} \mathbb{P}[A_C] < \infty$, which implies (25) by applying the Borel-Cantelli lemma. Using a similar approach we can prove (26). Combining (25) and (26) yields (24). Combining (24) and the fact that $\sum_{m=1}^{M} s_m(t_m + \nu_m z) = \sum_{m=1}^{M} S_m(T_m + n_m) = C$, we prove (23).

Notably, the result still holds when $\rho_m C \sim l_m(C)$ for $1 \le m \le M-1$, where $l_m(\cdot)$'s are slowly varying functions that satisfy $\lim_{x\to\infty} l_m(bx)/l_m(x) = 1$ for any positive constant b, because in this case, $t_m + \nu_m z$ is dominated by $\nu_m z$ and $Y_m(\boldsymbol{\rho}; C)$ is almost surely dominated by the cache space occupied by flow m in cache M, $1 \le m \le M$, as the total cache size $C \to \infty$. □

Received February 2019; revised March 2019; accepted April 2019