



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

Guocong Quan

The Ohio State University, USA  
quan.72@osu.edu

Atilla Eryilmaz

The Ohio State University, USA  
eryilmaz.2@osu.edu

Jian Tan

Alibaba Group, USA & The Ohio State University, USA  
j.tan@alibaba-inc.com  
tan.252@osu.edu

Ness Shroff

The Ohio State University, USA  
shroff.11@osu.edu

## ABSTRACT

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: in order to minimize the 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, named I-PLRU, where the data flows can be inserted at different positions of a pooled cache. This new design can achieve the optimal performance of the static SLRU, but retains the adaptability of PLRU for 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 under a stationary request arrival process. From an engineering perspective, 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.

## 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 *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3309697.3331509>

## 1 MOTIVATION

LRU caching is widely used in large-scale data systems to accelerate data access. Nevertheless, a fundamental question still remains: how should the cache space be organized to serve multiple data flows? Two commonly used approaches are the separated LRU (SLRU) paradigm and the pooled LRU (PLRU) paradigm. However, neither of these two designs can satisfactorily solve this problem.

**SLRU Paradigm:** Under the SLRU paradigm, the whole cache space is separated into multiple LRU caches. Each data flow is served by a dedicated LRU cache. We use  $\mathcal{S}(\theta; C)$  to denote the SLRU paradigm serving  $M$  flows, where the total cache size is  $C$  and the size of the LRU cache serving flow  $m$  is  $\theta_m C$ ,  $0 \leq \theta_m \leq 1$ ,  $\sum_{m=1}^M \theta_m = 1$ ,  $1 \leq m \leq M$ . Although the cache space allocation (i.e.,  $\theta_m$ 's) can be statically optimized to minimize the miss probability for stationary workloads, 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. Dynamically changing the allocation may cause low utilization including cold-start problems and memory fragmentation.

**PLRU Paradigm:** Under the PLRU paradigm, the whole cache space is organized as a single LRU cache and serves all data flows in a sharing manner. Specifically, data items are maintained in a list according to the LRU policy and the requested data will always be inserted at the head of the list. Although the PLRU paradigm enjoys the self-adaptive property due to the resource pooling nature, it does not support flexible configurations to optimize general system objectives.

Since neither the SLRU or the PLRU paradigms can achieve satisfactory performance in both static and dynamic environments, we

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-SpecEES-1824337, CSR-NeTS-1717060, and the ONR grant: N00014-17-1-2417.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*SIGMETRICS '19 Abstracts*, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3309697.3331509>

are motivated to propose a new multi-flow LRU cache management paradigm to solve the problem.

## 2 A NEW FLEXIBLE CACHING PARADIGM

In this paper, we propose an insertion based pooled LRU caching design, named I-PLRU. Under the I-PLRU paradigm, the entire cache space is organized as a single list. The data items from different flows can be inserted at different positions of the list rather than merely the head of the list. In Fig. 1, we present an example of 3 flows served by the I-PLRU paradigm. Note that the PLRU paradigm

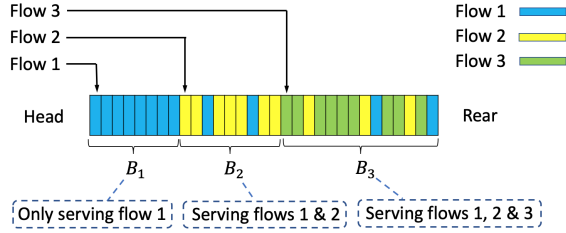


Figure 1: Three data flows served by I-PLRU caching.

is a special case of the I-PLRU paradigm where all data flows are inserted at the head of the list. For  $M$  flows served by I-PLRU, the whole list is separated into  $M$  blocks (i.e.,  $B_m, 1 \leq m \leq M$ ) by the insertion positions. Let  $\mathcal{I}(\boldsymbol{\eta}; C)$  denote the I-PLRU paradigm where the total cache size is  $C$  and the size of  $B_m$  is  $\eta_m C, 0 \leq \eta_m \leq 1, \sum_{m=1}^M \eta_m = 1, 1 \leq m \leq M$ . By adaptively changing the insertion positions, I-PLRU supports flexible and refined configurations for each individual flow. 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 show 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 dynamic 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 always fully utilized even when configurations are dynamically adapted.

### 2.1 Equivalence Mapping

In order to characterize the asymptotic performance of I-PLRU, we establish the following equivalence mapping between an SLRU configuration and an I-PLRU configuration.

*Definition 2.1 (Equivalence)*. Consider  $M$  data flows and a total cache size  $C$ . Let  $X_m(\boldsymbol{\eta}; C), 1 \leq m \leq M$  denote the cache space occupied by flow  $m$  under the I-PLRU paradigm  $\mathcal{I}(\boldsymbol{\eta}; C)$ . 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  $C \rightarrow \infty$

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

According to Definition 2.1, the cache space occupied by each flow under an I-PLRU paradigm is concentrated around the cache space allocated to that flow under the equivalent SLRU paradigm. In addition, we prove that under equivalent I-PLRU and SLRU paradigms, each flow achieves the same asymptotic miss probability. Therefore, in order to characterize the asymptotic miss probabilities under an I-PLRU paradigm, it suffices to find its equivalent SLRU paradigm and apply preliminary results on SLRU caching. In this paper, we propose effective algorithms to find the equivalent SLRU configuration for a given I-PLRU configuration and vice versa.

**THEOREM 2.2.** Consider  $M$  data flows served by an I-PLRU paradigm  $\mathcal{I}(\boldsymbol{\eta}; C)$ . Let  $F_1(\boldsymbol{\eta}; C)$  denote the output of Algorithm 1 of [1] with  $\boldsymbol{\eta}$  and  $C$  as its input. If the popularity of each flow follows a Zipf's distribution, then under mild conditions, we have

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

**THEOREM 2.3.** Consider  $M$  flows served by an SLRU paradigm  $\mathcal{S}(\boldsymbol{\theta}; C)$ . Let  $F_2(\boldsymbol{\theta}; C)$  be the output of Algorithm 2 of [1] with  $\boldsymbol{\theta}$  and  $C$  as its input. If the popularity of each flow follows a Zipf's distribution, then under mild conditions, we have

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

Moreover, we prove that the equivalence mapping defined by Algorithms 1 and 2 is one-to-one in the asymptotic regime.

### 2.2 Optimal I-PLRU for Minimizing Misses

Let  $Q_m^\pi$  denote the miss probability of flow  $m$  under the caching paradigm  $\pi$  (e.g., SLRU, I-PLRU). Let  $w_m \geq 0, 1 \leq m \leq M$ . Define the miss probability minimization (MPM) problem as

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

Notably, the MPM problem under I-PLRU is non-convex. However, we are able to find the optimal I-PLRU configuration  $\boldsymbol{\eta}^*$  by first solving the MPM problem under SLRU which is convex and then applying Algorithm 2 to find the equivalent I-PLRU configuration.

**THEOREM 2.4.** Let  $\boldsymbol{\eta}^*$  denote the optimal I-PLRU configuration and  $\boldsymbol{\theta}^*$  denote the optimal SLRU configuration. Under mild conditions, we have, for  $1 \leq m \leq M$ , as  $C \rightarrow \infty$

$$\begin{aligned} \boldsymbol{\eta}^* & \sim F_2(\boldsymbol{\theta}^*; C), \\ Q_m^{I-PLRU}(\boldsymbol{\eta}^*; C) & \sim Q_m^{SLRU}(\boldsymbol{\theta}^*; C). \end{aligned}$$

For  $f(x) = (f_1(x), \dots, f_M(x)) \in \mathbb{R}^M$ , and  $g(x) = (g_1(x), \dots, g_M(x)) \in \mathbb{R}^M$ ,  $f(x) \sim g(x)$  means  $\lim_{x \rightarrow \infty} f_m(x)/g_m(x) = 1, 1 \leq m \leq M$ . Applying Theorem 2.4, we solve the MPM problem under I-PLRU in the asymptotic regime, and prove that the optimal I-PLRU achieves the same asymptotic miss probability as the optimal SLRU.

In real-world applications, the popularity may not follow a Zipf's distribution, and could be unknown and time-varying. We propose heuristic algorithms in the full paper [1] to address these problems.

## REFERENCES

- [1] Guocong Quan, Jian Tan, Atilla Eryilmaz, and Ness Shroff. 2019. A New Flexible Multi-flow LRU Cache Management Paradigm for Minimizing Misses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 2 (2019).