Stationary Distribution of a Generalized LRU-MRU Content Cache*

George Kesidis

School of EECS

Pennsylvania State University

University Park, PA, 16802, USA

Email: gik2@psu.edu

October 23, 2019

Abstract

Many different caching mechanisms have been previously proposed, exploring different insertion and eviction policies and their performance individually and as part of caching networks. We obtain a novel closed-form stationary invariant distribution for a generalization of Least Recently Used (LRU) and Most Recently Used (MRU) eviction for single caching nodes under a reference Markov model. Numerical comparisons are made with an "Incremental Rank Progress" (IRP a.k.a. CLIMB) and random eviction

*This research supported in part by a Cisco Systems URP gift and NSF CNS grant 1526133.

(RE a.k.a. random replacement, RANDOM) methods under a steady-state Zipf popularity distribution. The range of cache hit probabilities is smaller under MRU and larger under IRP compared to LRU. We conclude with the invariant distribution for a special case of a RE caching tree-network.

1 Introduction

Caching is a ubiquitous mechanism in communication and computer systems. The role of a content caching network is to reduce the load on the origin servers of requested data objects, reduce the required network bandwidth to transmit content[1], and reduce the response times to the queries. Caching in computational settings reduces delays associated with disk IO (page caches). Data actively being, or likely soon to be, accessed by a CPU is stored in lower-level caches, *i.e.*, memories closer (with less access time) to the CPU.

The invariant distribution of the widely deployed Least Recently Used (LRU) eviction mechanism for a caching node was found in [1]. LRU has lower average miss rate compared to FIFO caching[2] [2, 3, 4]. Numerically useful approximations for LRU caching nodes are found in [5, 6, 7, 8, 9, 10]; in particular the *expected working set miss ratio* (WS) approximation of [5, 6] and that of [9] are equivalent [11]. In [12], LRU caching was studied for dependent (semi-Markov) object demand processes in a limiting regime for certain object popularity profiles. In [13, 14], time-to-live (TTL) caching *networks* are studied. Approximations for

[1] That is, content that is not encrypted for particular end-users.

[2] Under FIFO caching, the oldest item in the cache is evicted upon a cache miss.

networks of "capacity driven" caches are studied in [15, 16] (the latter adapting the approximation of [5, 6, 9] including under non-LRU cache eviction policies).

Under Most Recently Used (MRU) eviction, the *youngest* object in the cache is evicted upon cache miss. More specifically, an object is evicted under MRU when it is the subject of a cache hit or miss (so becomes youngest) and then a cache miss (query for an uncached object) immediately follows. MRU is used in cases where the older the object is in the cache, the more likely it is to be accessed [17]. That is, MRU is used when demand for hot (most popular) objects is such that they are not likely to be needed again soon after they are queried for, *e.g.*, the inter-query times of hot objects are a.s. lower bounded by a strictly positive amount, *cf.*, Section 5.

In this paper, we focus on single caching nodes and present a closed-form invariant distribution for a standard Markov model of a generalization of LRU and MRU eviction under the IRM. To this end, we provide a proof for LRU which we will subsequently adapt. For a Zipf popularity distribution, numerical comparisons are made with the simple Incremental Rank Progress (IRP)[3] and and Random Eviction (RE a.k.a. random replacement, RANDOM [3]) methods. Our numerical examples focus on the range of cache-hit probabilities for steady-state Zipf popularity distributions. We numerically show that the range of cache hit probabilities is smaller under MRU and larger under IRP compared to LRU, and conjecture that this is true in general. We next give a result for a special case of

[3] Called CLIMB in [3], IRP is somewhat related to the insertion scheme based on tandem virtual caches of "k-LRU" [16].

an RE caching tree-network. The paper concludes with a summary.

2 Background

The generalized LRU/MRU problem we consider in the following is similar to permutation-valued Markov chains studied in [18, 19], where all all objects are ranked, not just those cached.

2.1 Markov model of Least Recently Used (LRU) eviction policy

The stationary state-space $\mathcal{R}$ of a LRU cache is the set of B-permutations of $\{1, 2, ..., N\}$ where N is the number of objects that could be cached and B objects is the capacity of the cache with $N > B > 0$ (typically $N \gg B$) and the objects assumed identically sized (but *cf.*, (8)). For $r \in \mathcal{R}$, define $r(k)$ as the element of r in the k^{th} position. The entries of r are *ranked* in order of their position in r:

- the most recently accessed (LRU) object being $r(1)$,
- the oldest object in the cache being $r(B)$, and
- uncached objects n are denoted $n \notin r$.

Note that in a transient regime, the cache may be in a state $\notin \mathcal{R}$ with fewer than B objects cached.

For a single node, we assume that demand process for object $n \in \{1, 2, ..., N\}$ is Poisson with intensity λ_n. The Poisson demands are assumed independent. Let the total demand intensity be $\Lambda = \sum_{n=1}^{N} \lambda_n$. So, this is the classical "Independent Reference Model" (IRM) with query probabilities $p_n = \lambda_n/\Lambda$ [3, 4].

For LRU, a cache miss of object $r(1)$ at state $M_n^{-1}(r)$ resulting in a transition to state $r \in \mathcal{R}$ occurs at rate $\lambda_{r(1)}$, where $n \notin r$ and

$$(M_n^{-1}(r))(k) = \begin{cases} n & \text{if } k = B \\ r(k+1) & \text{if } k < B \end{cases}$$

i.e., $n \notin r$ is the oldest object in the cache in state $M_n^{-1}(r)$.

For LRU, a cache hit of object $r(1)$ at state $H_k^{-1}(r)$ resulting in a transition to state r occurs at rate $\lambda_{r(1)}$ where $1 \leq k \leq B$ and

$$(H_k^{-1}(r))(\ell) = \begin{cases} r(1) & \text{if } \ell = k \\ r(\ell+1) & \text{if } \ell < k \\ r(\ell) & \text{if } k < \ell \leq B \end{cases}$$

i.e., $r(1)$ is the k^{th} youngest object in the cache in state $H_k^{-1}(r)$ and $H_1^{-1}(r) = r$.

As commonly assumed with the IRM [15], we also assume (i) that cache misses cause the query to be forwarded, possibly to a server holding the requested object, and once resolved, the object is reverse-path forwarded so that caches that missed it can be updated; and (ii) the required time for this query resolution process is negligible compared to the inter-querying times of the caching network.

2.2 LRU stationary invariant distribution

The following invariant of LRU found by W.F. King in [1].

Theorem 2.1 *The unique invariant distribution of the LRU Markov chain is*

$$\pi(r) = \prod_{k=1}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}} \tag{1}$$

for $r \in \mathcal{R}$*, where* $\forall k$*,* $\sum_{i=k}^{k-1}(...) \equiv 0$.

proof The full balance equations are: $\forall r \in \mathcal{R}$,

$$\begin{aligned}(\Lambda - \lambda_{r(1)})\pi(r) &= \\ &\sum_{n \notin r} \lambda_{r(1)}\pi(M_n^{-1}(r)) + \sum_{j=2}^{B} \lambda_{r(1)}\pi(H_j^{-1}(r)).\end{aligned} \tag{2}$$

Under (1), for all $n \notin r$,

$$\pi(M_n^{-1}(r)) = \frac{\lambda_n}{\Lambda - \sum_{i=2}^{B} \lambda_{r(i)}} \prod_{k=2}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}}$$

Also under (1), for all $j \in \{2, 3, ..., B\}$,

$$\begin{aligned}\pi(H_j^{-1}(r)) &= \prod_{k=2}^{j} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}} \cdot \frac{\lambda_{r(1)}}{\Lambda - \sum_{i=2}^{j} \lambda_{r(i)}} \\ &\quad \cdot \prod_{k=j+1}^{B} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}}\end{aligned}$$

Substituting into (2) and after some term cancellation, we see that (1) satisfies (2) if and only if

$$\begin{aligned}1 &= \prod_{k=3}^{B+1} \frac{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}} \\ &\quad + \sum_{j=2}^{B} \prod_{k=3}^{j} \frac{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)}}{\Lambda - \sum_{i=2}^{k-1} \lambda_{r(i)}} \cdot \frac{\lambda_{r(1)}}{\Lambda - \sum_{i=2}^{j} \lambda_{r(i)}}\end{aligned} \tag{3}$$

where $\prod_{k=3}^{2}(...) \equiv 1$.

Regarding (3), consider the following sequence of independent random experiments to fill the cache. Suppose we're given initially that the first cache entry is $r(2)$. Now *sequentially*, according to the distribution (1), object $r(1)$ attempts to enter the cache after $r(2)$. If it fails to enter in the k^{th} attempt, then $r(k+2)$ is placed in the cache instead and $r(1)$ tries again. The summand of (3) with

$j = 2$ is the probability that $r(1)$ enters in the second position right after $r(2)$: $\lambda_{r(1)}/(\Lambda - \lambda_{r(2)})$. Generally, the summand for $j \in \{2, 3, ..., B\}$ is the probability $r(1)$ enters in the j^{th} position (after having failed to enter in one of the more highly ranked ones). The first term of the right-hand-side of (3) is the probability $r(1)$ fails to enter the cache. So, (3) must generally hold by the law of total probability.

Finally, since the stationary LRU Markov chain is irreducible on $\mathcal{R}$, there is a unique invariant.

This result was generalized in [20] to add object-dependent insertion probabilities interpreted as access costs. Also note that, generally, the LRU Markov chain is neither time-reversible nor quasi-reversible [21]. Obviously, more popular objects (larger λ) are more likely stored, and the LRU invariant is uniform in the special case that all the mean querying rates λ_n are the same. Finally, by PASTA, the stationary hit probability of object n in a LRU cache is

$$h_n = \sum_{r \,:\, n \in r} \pi(r),$$

where the approximations of hit probabilities in [9, 10] are obviously substantially simpler to compute.

2.3 Incremental Rank Progress (IRP or "CLIMB" [3]) upon query

Under LRU, a query for any object n results in it being ranked first in the cache. One can also consider slowing the "progress through the ranks" of objects as they are queried, leading to some obvious trade-offs with LRU: Slowing progress would mean less popular content does not enter the cache at first rank, but also more popular content will take longer to reach the cache. Such issues are important when there are dynamic changes/churn in objects cached and their popularity.

Under an Incremental Rank Progress (IRP) caching mechanism, a query for object n results in its rank improved by just one (or zero if the object is already ranked first), *i.e.*, for $1 \leq k \leq B-1$, $r \in \mathcal{R}$,

$$(T_k(r))(\ell) = \begin{cases} r(k) & \text{if } \ell = k+1 \\ r(k+1) & \text{if } \ell = k \\ r(\ell) & \text{else} \end{cases}$$

where the transition $T_k(r) \to r$ with rate $\lambda_{r(k)}$. Missed objects enter the cache at lowest rank, *i.e.*, for $n \notin r$, define

$$(S_n(r))(\ell) = \begin{cases} r(k) & \text{if } \ell < B \\ n & \text{if } \ell = B \end{cases}$$

where the transition $S_n(r) \to r$ occurs with rate $\lambda_{r(B)}$. The invariant for IRP is found in [3] and can be immediately shown using detailed balance.

Theorem 2.2 *IRP is time-reversible with unique stationary invariant*

$$\pi(r) = \frac{\prod_{k=1}^{B} \lambda_{r(k)}^{B+1-k}}{\sum_{r' \in \mathcal{R}} \prod_{k=1}^{B} \lambda_{r'(k)}^{B+1-k}}. \quad (4)$$

2.4 Random Eviction (RE or "RANDOM" [3]) upon cache miss without cache rankings

Suppose that a cache miss of object n at state $M_{\ell,n}^{-1}(r)$ results in a transition to state $r \in \mathcal{R}$ at rate $B^{-1}\lambda_n$, where $n \in r$, $n \notin M_{\ell,n}^{-1}(r)$, $\ell \in M_{\ell,n}^{-1}(r)$, and $\ell \notin r$.

That is, a cache miss for object n results in n inserted into the cache and evicting of an object ℓ selected uniformly at random from the cache. The cache state r does not change if a cache hit occurs. The stationary state-space $\mathcal{R}$ is the set of B-*combinations* of N different objects. The following invariant for RE is also found in [3] and can also be immediately shown by detailed balance.

Theorem 2.3 *The RE Markov chain is time-reversible with unique stationary invariant distribution*

$$\pi(r) \quad = \quad \frac{\prod_{n\in r} \lambda_n}{\sum_{r'\in\mathcal{R}} \prod_{n\in r'} \lambda_n}. \tag{5}$$

2.5 Aggregate cache-hit rates

Define the aggregate hit rate for a caching discipline as

$$H \quad := \quad \sum_{n=1}^{N} h_n p_n = \sum_{n=1}^{N} h_n \frac{\lambda_n}{\Lambda}, \tag{6}$$

i.e., the probability that a query is a cache hit. This is a single criterion that can be used to compare different caching disciplines. Typically H is largest for LRU eviction under the IRM. Note that under the IRM, by PASTA and Fubini's theorem the following holds for all of the above capacity-driven caching disciplines,

$$\sum_{n=1}^{N} h_n = \sum_{n=1}^{N} \sum_{r\in\mathcal{R}:n\in r} \pi(r) = \sum_{r\in\mathcal{R}} \pi(r) B = B. \tag{7}$$

2.6 Considering objects with different lengths

To account for objects of different lengths for capacity-driven caches (with ranked objects) like LRU, simply consider a "complete-rankings" LRU variation, where the ranking of *all* objects is maintained whether the objects are cached or not. That is, the state-space $\mathcal{R}$ is now the set of permutations of *all* N objects.

Corollary 2.1 *The unique stationary invariant π of complete-rankings LRU is (1) with B replaced by N.*

Additionally consider the different sizes ℓ_n bytes of objects n, where the cache capacity B is in bytes. The number of objects in the cache is given by

$$K(r) \quad = \quad \max\{K \mid \sum_{k=1}^{K} \ell_{r(k)} \leq B, \ 1 \leq K \leq N\}.$$

So, the hit probability of object n when the objects are of variable length is

$$h_n \quad = \quad \sum_{r \,:\, r(n) \leq K(r)} \pi(r). \tag{8}$$

See the byte-hit performance metric of [22].

3 Most Recently Used (MRU) eviction

Again define the state-space $\mathcal{R}$ as the set of B-permutations of $\{1, 2, ..., N\}$. Under MRU [17, 22], a cache hit of object $r(1)$ at state $H_k^{-1}(r)$ resulting in a transition to state r occurs at rate $\lambda_{r(1)}$ where $1 \leq k \leq B$ and $(H_k^{-1}(r))(\ell)$ is given by (1) as LRU. But for MRU, a cache miss of object $r(1)$ at state $M_n^{-1}(r)$ resulting in a transition to state $r \in \mathcal{R}$ occurs at rate $\lambda_{r(1)}$, where $n \notin r$ and

$$(M_n^{-1}(r))(k) \quad = \quad \begin{cases} n & \text{if } k = 1 \\ r(k) & \text{if } k > 1 \end{cases}$$

i.e., $n \notin r$ is the *youngest* object in the cache in state $M_n^{-1}(r)$.

Theorem 3.1 *The unique invariant distribution of the MRU Markov chain is, for $r \in \mathcal{R}$,*

$$\begin{aligned}\pi(r) &= \\ &\frac{\lambda_{r(1)}}{\Lambda} \cdot \frac{1}{\binom{N-1}{B-1}} \prod_{k=2}^{B-1} \frac{\lambda_{r(k)}}{\Lambda - \sum_{i=1}^{k-1} \lambda_{r(i)} - \sum_{n \notin r} \lambda_n}.\end{aligned} \tag{9}$$

proof The full balance equations are as for LRU but with a different definition for M_n^{-1}.

Let $\Lambda_r = \Lambda - \sum_{n \notin r} \lambda_n$. By substituting (9) into the full balance equations (and moving the cache-miss terms to the left-hand side), we get that (9) satisfies the full balance equations if and only if

$$\begin{aligned} 1 &= \frac{1}{\Lambda_r - \lambda_{r(1)}} \left(\lambda_{r(1)} \sum_{j=2}^{B-1} \prod_{k=2}^{j-1} \frac{\Lambda_r - \sum_{i=1}^{k-1} \lambda_{r(i)}}{\Lambda_r - \sum_{i=2}^{k} \lambda_{r(i)}} \right. \\ &\quad \left. + \lambda_{r(B)} \prod_{k=2}^{B-1} \frac{\Lambda_r - \sum_{i=1}^{k-1} \lambda_{r(i)}}{\Lambda_r - \sum_{i=2}^{k} \lambda_{r(i)}} \right) \\ &= \sum_{j=2}^{B-1} \left(\prod_{k=2}^{j-1} \frac{\Lambda_r - \sum_{i=1}^{k} \lambda_{r(i)}}{\Lambda_r - \sum_{i=2}^{k} \lambda_{r(i)}} \right) \frac{\lambda_{r(1)}}{\Lambda_r - \sum_{i=2}^{j} \lambda_{r(i)}} \\ &\quad + \prod_{k=2}^{B-1} \frac{\Lambda_r - \sum_{i=1}^{k} \lambda_{r(i)}}{\Lambda_r - \sum_{i=2}^{k} \lambda_{r(i)}} \end{aligned} \tag{10}$$

where $\prod_{k=2}^{1}(...) \equiv 1$.

Regarding (10), consider the following sequence of independent random experiments to determine the position of object $\lambda_{r(1)}$ when filling the cache, *given* that only objects $\in r$ will be chosen and that $\lambda_{r(2)}$ has already been chosen first. $\lambda_{r(1)}$ is chosen on the first try with probability $\lambda_{r(1)}/(\Lambda_r - \lambda_{r(2)})$, otherwise $\lambda_{r(3)}$ enters the cache - this is the summand of (10) with $j = 2$. Generally, the j^{th} summand is the probability that $\lambda_{r(1)}$ enters the cache on the $(j-1)^{\text{th}}$ try, otherwise object $\lambda_{r(j+1)}$ is placed in the cache. The final term of (10) is the probability $r(1)$ fails to enter the cache before the last (B^{th}) position, because in the penultimate choice only objects $r(B)$ and $r(1)$ remain, *i.e.*, $\lambda_{r(B)} = \Lambda_r - \sum_{i=1}^{B-1} \lambda_{r(i)}$. So, (3) must generally hold by the law of total probability.

Finally, since the stationary LRU Markov chain is irreducible on $\mathcal{R}$, there is a unique invariant.

Note that it's easily directly verified that (9) satisfies (2) for the cases $B = 2$ and $B = 3$, *e.g.*, for $B = 3$ and $N = 4$,

$$\pi(r) = \lambda_{r(1)}\lambda_{r(2)}/(3\Lambda(\lambda_{r(2)} + \lambda_{r(3)})).$$

To interpret (9): $\lambda_{r(1)}$ is chosen with probability $\lambda_{r(1)}/\Lambda$; then the remaining $B-1$ objects in r are chosen from the remaining $N-1$ objects uniformly at random with probability $\binom{N-1}{B-1}^{-1}$; finally, the order of the remaining items $\lambda_{r(2)}, \lambda_{r(3)}, ...$ are determined as the LRU invariant distribution (1).

Finally, we make an observation about cache-hit probabilities under MRU eviction. Consider a MRU cache under the IRM that is "synchronized" so that a query for object n occurs at time 0. Thus, immediately thereafter, n is the MRU object in the cache. The next query for object n will be at time $T_n \sim \exp(\lambda_n)$. Again, under MRU eviction, the only way an object n is evicted is when a cache miss occurs immediately after a query for n, *i.e.*, a cache miss when n is the MRU object. So, the stationary hit probability h_n of object n equals the probability that a hit occurs at time T_n, which is

- the probability that no other queries occurred in the interval $(0, T_n)$ plus
- the probability that a query does occur in $(0, T_n)$ and the first such query is

a hit.

Thus, we can write $\forall n$,

$$h_n = \mathsf{E}\left(\mathrm{e}^{-T_n \sum_{j\neq n} \lambda_j} + (1 - \mathrm{e}^{-T_n \sum_{j\neq n} \lambda_j}) \sum_{j\neq n} \frac{\lambda_j h_{j|n}}{\sum_{i\neq n} \lambda_i} \right),$$

where $h_{j|n}$ is the probability that a query is a hit on j given that object n is MRU. We have therefore shown the following.

Proposition 3.1 *For a MRU-eviction cache under the stationary IRM:* $\forall n$, $h_n = p_n + \sum_{j\neq n} p_j h_{j|n} = \sum_j p_j h_{j|n}$, *where* $p_j = \lambda_j / \sum_i \lambda_i$ *and* $h_{j|j} = 1$*; equivalently, a kind of balance equation:* $\forall n$,

$$\sum_j p_j h_{n|j} = \sum_j p_j h_{j|n}.$$

4 Generalization of LRU and MRU

"k^{th} Recently Used" (kRU) is a simple generalization of LRU and MRU wherein object $r(k)$, for some fixed $k \in \{1, 2, ..., B\}$, is evicted upon cache miss; otherwise cache insertion (at rank 1) upon misses and promotion (to rank 1) and demotions (by 1) upon hits are the same as both MRU and LRU. That is, BRU is LRU and 1RU is MRU.

Corollary 4.1 *The invariant distribution of kRU is*

$$\pi(r) = \prod_{j=1}^{k} \frac{\lambda_{r(j)}}{\Lambda - \sum_{i=2}^{j} \lambda_{r(i)}} \times \frac{1}{\binom{N-k}{B-k}} \prod_{j=k+1}^{B-1} \frac{\lambda_{r(j)}}{\Lambda - \sum_{i=1}^{j-1} \lambda_{r(i)} - \sum_{n\notin r} \lambda_n}. \tag{11}$$

5 Numerical results for small N, B

In this numerical study, we directly computed the invariants π by generating all possible object permutations representing cache state by the Steinhaus-Johnson-Trotter algorithm. So, we considered only small values for the number of objects and the cache size. Figure 1 is representative of our numerical study on cache-hit probabilities using a Zipf popularity model $\lambda_n = n^{-\alpha}$ for with $\alpha = 0.75$ (see Table 1 of [23]) and most popular object indexed 1 with normalized rate $\lambda_1 = 1$.

kRU with $1 < k < B$ gives hit-probability performance between MRU ($k = 1$) and LRU ($k = B$). That is, one can see that the range of hit probabilities for LRU is larger than that of MRU.

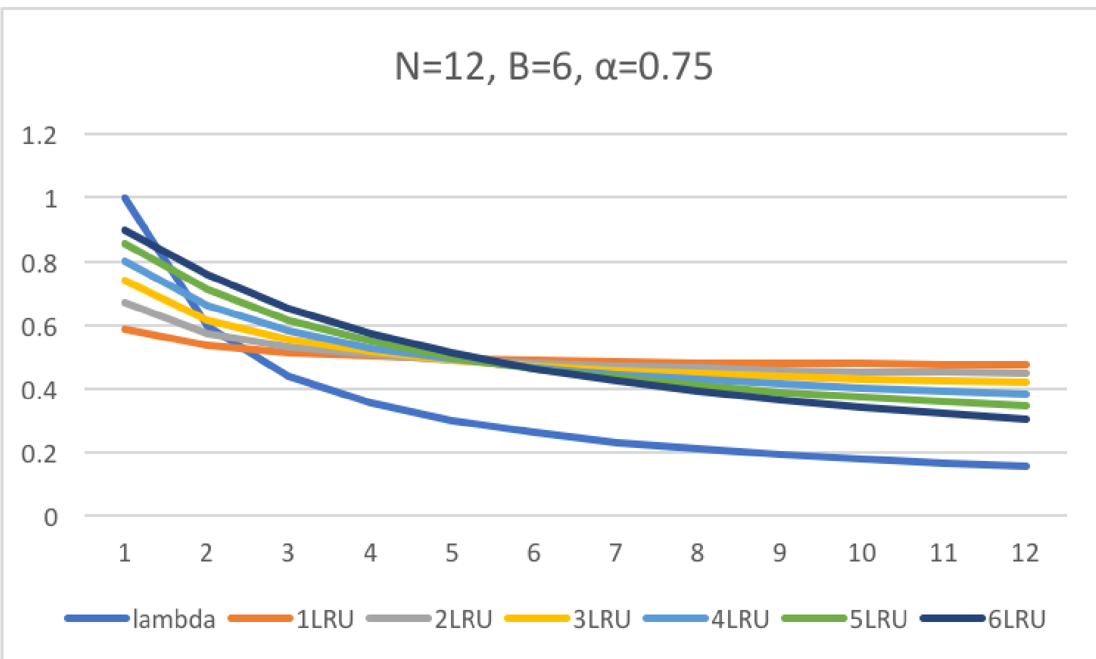

Figure 1: kRU cache hit probabilities h_n and popularity λ_n versus object index n for a cache of size $B = 6$, $N = 12$ objects, and Zipf popularity parameter $\alpha = 0.75$, where LRU$=$ 6RU and MRU=1RU

Figure 2 shows the results of a typical simulation study of kRP with cache entry at lowest rank B upon cache miss compared to LRU. Note that kRP has greater range of hit probability values than LRU. We postulate that generally for Zipf popularity distributions, the range of hit probabilities of IRP is larger than those of LRU which is larger than those of MRU.

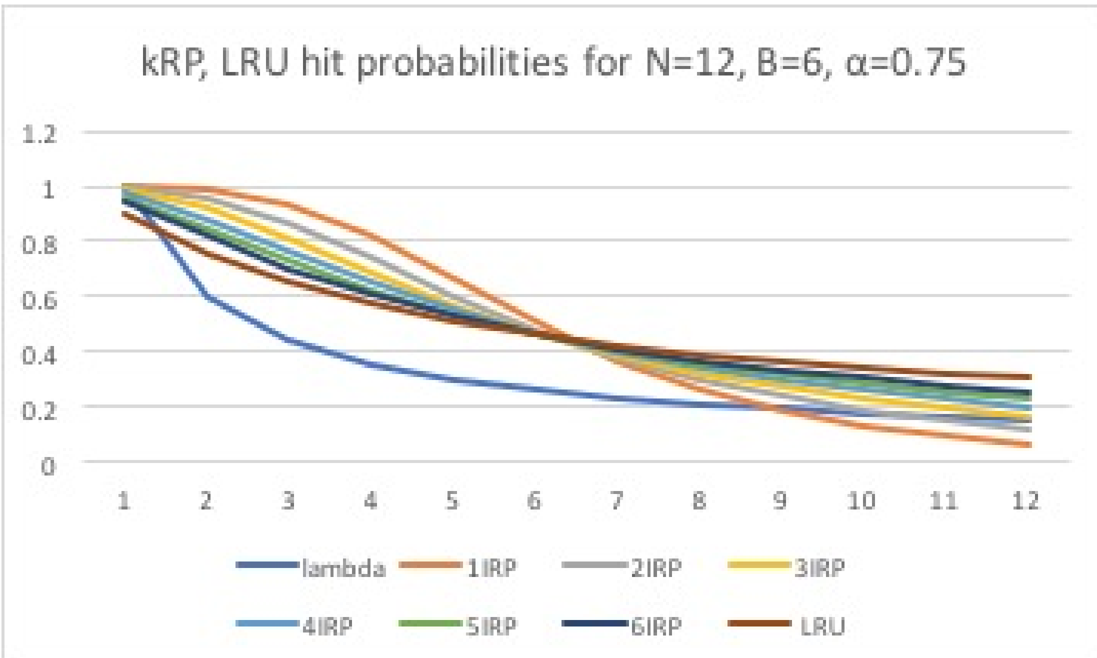

Figure 2: kRP (with cache entry upon cache miss) and LRU cache hit probabilities h_n and popularity λ_n versus object index n for a cache of size $B = 6$, $N = 12$ objects, and Zipf popularity parameter $\alpha = 0.75$

For the example of Figure 3, RE has a range of hit probabilities between MRU and LRU. Recall (7), *i.e.*, that the sum of the stationary hit probabilities is the same for all of these caching disciplines under the IRM

Though our derivations herein are for the IRM, MRU may out-perform LRU for non-Poisson arrivals in terms of aggregate hit rate (6). Recall mention in

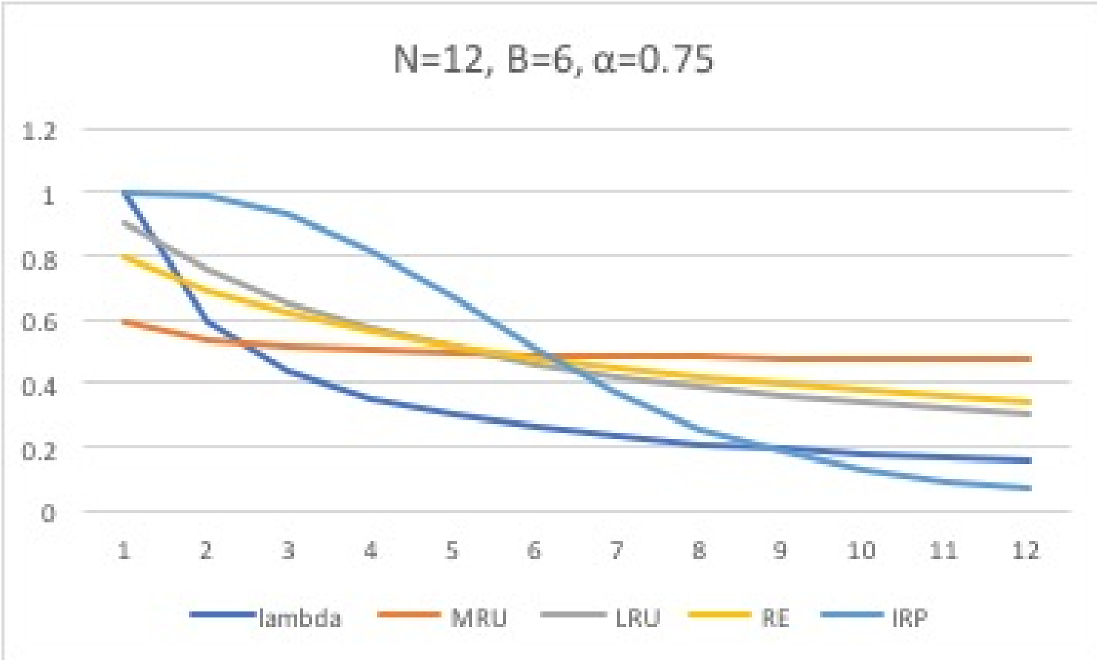

Figure 3: Cache hit probabilities h versus popularity λ for a cache of size $B = 3$, $N = 12$ objects, and Zipf popularity parameter $\alpha = 0.75$

the Section 1 that MRU is used when demand for hot (most popular) objects is such that they are not likely to be needed again soon after they are queried for. Consider the case where inter-query times are lower bounded by a constant D. Specifically, inter-query times equal D plus an exponentially distributed quantity, such that $D = 0$ corresponds to the IRM (here with intensities following a Zipf popularity distribution). In Table 1, we see that LRU has best aggregate hit rate under IRM (mean hit rate increases with k when $D = 0$), while MRU is best when $D = 1, 2$ (mean hit rate decreases with k when $D = 2$).

k	$D=0$	$D=1$	$D=2$
1 (MRU)	0.52	0.4578	0.45
2	0.54	0.4213	0.40
3	0.56	0.4014	0.35
4	0.58	0.4026	0.31
5	0.60	0.4187	0.29
6 (LRU)	0.62	0.4423	0.29

Table 1: kRU aggregate hit rate (6) for $N = 12$ objects, cache of capacity $B = 6$ objects, and Zipf popularity distribution with exponent $\alpha = 0.75$ ($D = 0$ corresponds to the IRM).

6 Discussion: Networks of RE caches

The performance of Markovian networks of such capacity-driven caches are approximated in *e.g.*, [15, 16]. To illustrate the difficulties with capacity-driven caching networks, now consider the simplest ones based on RE. Though RE caches are time-reversible, a tree of independent local caches whose collective query-misses are forwarded to an Internet cache (also running RE, see Figure 4), is not time-reversible and its non-local nodes do not operate under the IRM. To see why it's not time-reversible, consider a cache miss of object n of local cache q of size B_q in state r_q, so that object n_q is evicted, and suppose it's also a miss on the Internet cache of size b in state R, so that object n is evicted; this can be reversed with one query (so that states r_q and R are restored) only if $n = n_q$.

The following result is for the very special case that the Internet cache holds

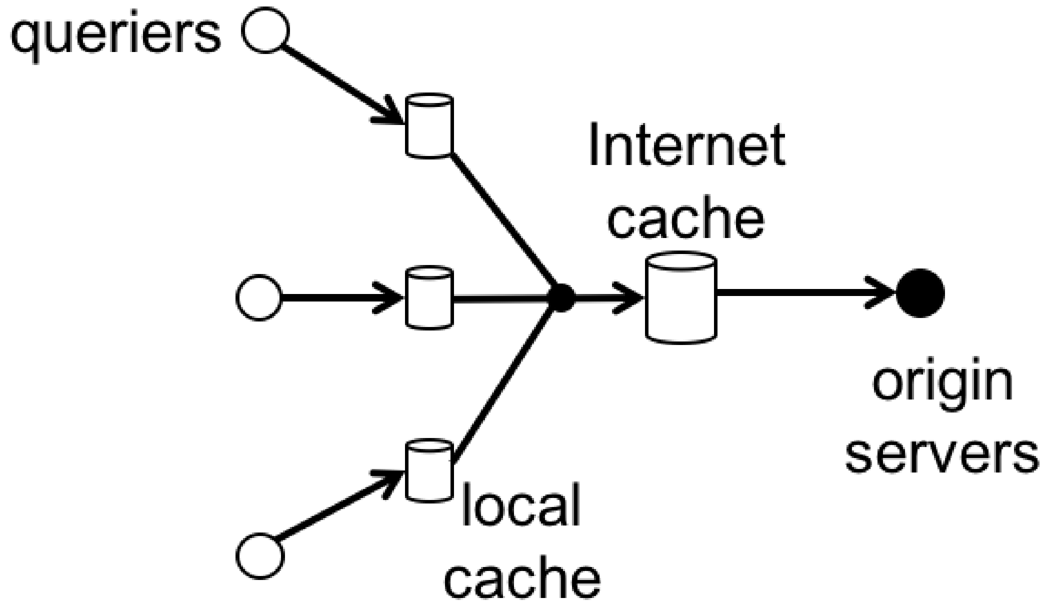

Figure 4: A tree-network of caching nodes that feeds forward cache misses with assumed independent local caches

only one object.

Proposition 6.1 *The invariant distribution* π *of the network Figure 4 with RE caching and* $b = 1$ *satisfies*

$$\pi(R|\underline{r}) \quad = \quad \frac{\sum_q \mathbf{1}\{R \in r_q\}\Lambda_{q,\overline{r}_q}/B_q}{\sum_q \Lambda_{q,\overline{r}_q}} \tag{12}$$

where

$$\Lambda_{q,\overline{x}} \quad = \quad \sum_{\ell \notin x} \lambda_{q,\ell},$$

$\sum_{\emptyset}(...) \equiv 0$, *and indicator* $\mathbf{1}X = 1$ *if* X *is true otherwise* $= 0$.

proof For $n \in r_q, m \notin r_q$, let $\delta_{q-n+m}\underline{r}$ be $\underline{r}$ but with n in r_q replaced by m.

Similarly define $\delta_{-n+\ell}R$. The full balance equations are

$$\begin{aligned}
&\pi(\underline{r}, R) \sum_{q,m:m\notin r_q} \lambda_{q,m} \\
&= \sum_{q,m,n:m\notin r_q;n\in r_q\cap R} \pi(\delta_{q-n+m}\underline{r}, R)\frac{\lambda_{q,n}}{B_q} + \\
&\quad \sum_{q,m,n,\ell:m\notin r_q;n\in r_q\cap R;\ell\notin R} \pi(\delta_{q-n+m}\underline{r}, \delta_{-n+\ell}R)\frac{\lambda_{q,n}}{B_q b}
\end{aligned}$$

Dividing by $\pi(\underline{r}, R) = \pi(R|\underline{r})\prod_q \pi(r_q)$ and then substituting the stationary joint distribution of the independent RE local caches (5) into the full balance equations gives: $\forall \underline{r}, R$,

$$\begin{aligned}
\pi(R|\underline{r}) \sum_{q,m:m\notin r_q} \lambda_{q,m} \;&=\; \sum_{q,m:m\notin r_q} \frac{\lambda_{q,m}}{B_q} \times \\
\sum_{n\in r_q\cap R} &\left(\pi(R|\delta_{q-n+m}\underline{r}) + \frac{1}{b}\sum_{\ell\notin R} \pi(\delta_{+l-n}R|\delta_{q-n+m}\underline{r})\right)
\end{aligned}$$

For the special case of $b = 1$, *i.e.*, R $(= n)$ is a single object, we get that the right-hand-side simplifies to

$$\begin{aligned}
&\sum_{q,m:m\notin r_q} \frac{\lambda_{q,m}}{B_q}\mathbf{1}\{R \in r_q\} \\
&\times \left(\pi(R|\delta_{q-R+m}\underline{r}) + \sum_{\ell\neq R} \pi(\ell|\delta_{q-R+m}\underline{r})\right) \\
&= \sum_{q,m:m\notin r_q} \frac{\lambda_{q,m}}{B_q}\mathbf{1}\{R \in r_q\} \;=\; \sum_q \frac{\Lambda_{q,\bar{r}_q}}{B_q}\mathbf{1}\{R \in r_q\}
\end{aligned}$$

The invariant is unique since $(R, \underline{r})$ is irreducible.

In steady state, $R \subset \cup_q r_q$ a.s., *i.e.*, if $\forall q,\ R \notin r_q$ then $\pi(R|\underline{r}) = 0$. Note that (12) is the eviction probability of object R upon local cache miss in local cache state $\underline{r}$. An individual RE cache r is not quasi-reversible since the miss rates ("departures"), $\frac{1}{\pi(r)}\sum_{m\notin r,n\in r} \lambda_n \pi(\delta_{-n+m}(r))$ depend on the state r. Though quasi-reversibility is not a necessary condition [21], Proposition 6.1 shows that RE networks generally do not have product-form invariants. More specifically, one can identify the incident mean rate of queries for object n to the Internet cache, $\widehat{\lambda}_n := \sum_q \lambda_{q,n}(1-h_{q,n}) = \sum_q \lambda_{q,n} \sum_{r_q : n \notin r_q} \pi(r_q)$, where $1-h_{q,n}$ is the stationary miss probability of local cache q for object n under RE[4]. According to this proposition, $\pi(R)$ does not depend on the $\widehat{\lambda}_n$ in the way the IRM invariant $\pi(r_q)$ depends on the $\lambda_{q,n}$ in (5), *i.e.*, $\pi(R) = \sum_{\underline{r}} \pi(R|\underline{r})\pi(\underline{r}) = \sum_{\underline{r}} \pi(R|\underline{r}) \prod_q \pi(r_q) \neq \widehat{\lambda}_R / \sum_n \widehat{\lambda}_n$. Finally note that, since the capacity of the Internet cache is one object ($b=1$), it could obviously be operating any eviction policy.

7 Summary

In this paper, under the IRM, a closed-form expression for the invariant distribution was derived for a caching node using kRU eviction. Numerically, it was shown that under IRM and Zipf popularity distributions for the data objects, the range of cache-hit probabilities of the data objects under IRP caching is larger than LRU, which is larger than RE, which is larger than MRU (also, a non-IRM example was given where MRU had higher aggregate hit rate than LRU). Finally, the invariant distribution of a special case of a Markovian RE caching tree-network was also derived.

[4]In this way, one can easily identify the "flow-balance equations" for more general caching networks [15].

References

[1] W. King, “Analysis of paging algorithms,” in *Proc. IFIP Congress*, Lyublyana, Yugoslavia, Aug. 1971.

[2] L. Belady, R. Nelson, and G. Shedler, “An Anomaly in Space-time Characteristics of Certain Programs Running in a Paging Machine,” *Commun. ACM*, vol. 12, no. 6, June 1969.

[3] O. Aven, E. Coffman, and Y. Kogan, *Stochastic analysis of computer storage.* D. Reidel Publishing Co., 1987.

[4] J. V. D. Berg and A. Gandolfi, “LRU is better than FIFO under the independent reference model,” *J. Appl. Prob.*, vol. 29, 1992.

[5] P. Denning and S. Schwartz, “Properties of the working-set model,” *Commun. ACM*, vol. 15, no. 3, p. 191 198, March 1972.

[6] R. Fagin, “Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities,” p. 222250, 1977.

[7] A. Dan and D. Towsley, “An approximate analysis of the LRU and FIFO buffer replacement schemes,” *SIGMETRICS Perform. Eval. Rev.*, vol. 18, p. 143 152, April 1990.

[8] P. Jelenkovic, “Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities,” *Ann. Appl. Probab.*, vol. 9, no. 2, p. 430464, 1999.

[9] H. Che, Y. Tung, and Z. Wang, “Hierarchical Web Caching Systems: Modeling, Design and Experimental Results,” *IEEE JSAC*, vol. 20, no. 7, Sept. 2002.

[10] C. Fricker, P. Robert, and J. Roberts, “A Versatile and Accurate Approximation for LRU Cache Performance,” in *Proc. International Teletraffic Congress*, 2012.

[11] P. Jelenkovic, “Private communication,” Dec. 2017.

[12] P. Jelenkovic and A. Radovanovic, “Least-recently-used caching with dependent requests,” *Theoretical Computer Science*, vol. 326, pp. 293–327, Oct. 2004.

[13] D. Berger, S. S. School, P. Gland, and F. Ciucu, “Exact Analysis of TTL Cache Networks The Case of Caching Policies Driven by Stopping Times,” in *Proc. ACM SIGMETRICS*, Austin, Texas, June 2014.

[14] F. Cavallin, A. Marin, and S. Rossi, “A product-form model for the analysis of systems with aging objects,” in *Proc. IEEE MASCOTS*, Atlanta, Sept. 2015.

[15] E. Rosensweig, J. Kurose, and D. Towsley, “Approximate models for general cache networks,” in *Proc. IEEE INFOCOM*, March 2010.

[16] M. Garetto, E. Leonardi, and V. Martina, “A Unified Approach to the Performance Analysis of Caching Systems,” *ACM TOMPECS*, vol. 1, no. 3, May 2016.

[17] S. Dar, M. Franklin, B. Jonsson, D. Srivastava, and M. Tan, "Semantic data caching and replacement," in *Proc. Conf. on Very Large Databases (VLDB)*, 1996.

[18] W. Hendricks, "An extension of a theorem concerning an interesting Markov chain," *J. Appl. Prob.*, vol. 10, p. 886890, 1973.

[19] ——, "An account of self-organizing systems," *SIAM J. Comput.*, vol. 5, no. 4, pp. 715–723, 1976.

[20] D. Starobinski and D. Tse, "Probabilistic methods for web caching," *Performance Evaluation*, 2001.

[21] X. Chao, M. Miyazawa, R. Serfozo, and H. Takada, "Markov network processes with product form stationary distributions," *Queueing Systems*, vol. 28, p. 377401, 1998.

[22] A. Balamash and M. Krunz, "An overview of web caching replacement algorithms," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, 2004.

[23] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," in *Proc. IEEE INFOCOM*, 1999.