TTL Approximations of the Cache Replacement Algorithms LRU(m) and h-LRU

Nicolas Gast, Benny van Houdt

TTL Approximations of the Cache Replacement Algorithms LRU(m) and h-LRU

Nicolas Gast[a,*], Benny Van Houdt[b]

[a] *Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble, France*
[b] *Univ. of Antwerp, Depart. of Math. and Computer Science, B2020-Antwerp, Belgium*

Abstract

Computer system and network performance can be significantly improved by caching frequently used information. When the cache size is limited, the cache replacement algorithm has an important impact on the effectiveness of caching. In this paper we introduce time-to-live (TTL) approximations to determine the cache hit probability of two classes of cache replacement algorithms: h-LRU and LRU($\mathbf{m}$). These approximations only require the requests to be generated according to a general Markovian arrival process (MAP). This includes phase-type renewal processes and the IRM model as special cases. We provide both numerical and theoretical support for the claim that the proposed TTL approximations are asymptotically exact. In particular, we show that the transient hit probability converges to the solution of a set of ODEs (under the IRM model), where the fixed point of the set of ODEs corresponds to the TTL approximation.

We use this approximation and trace-based simulation to compare the performance of h-LRU and LRU($\mathbf{m}$). First, we show that they perform alike, while the latter requires less work when a hit/miss occurs. Second, we show that as opposed to LRU, h-LRU and LRU($\mathbf{m}$) are sensitive to the correlation between consecutive inter-request times. Last, we study cache partitioning. In all tested cases, the hit probability improved by partitioning the cache into different parts – each being dedicated to a particular content provider. However, the gain is limited and the optimal partition sizes are very sensitive to the problem's parameters.

Keywords: Caching, TTL approximations, LRU

1. Introduction

Caches form a key component of many computer networks and systems. A large variety of cache replacement algorithms has been introduced and analyzed over the last few decades. A lot of the initial work was focused on deriving explicit expressions for the cache content distribution by using a Markov chain analysis [1]. This approach, however, is not always feasible: Even if explicit expressions can be obtained, they are often only applicable to analyze small caches, because of the time it takes to evaluate them. This gave rise to various approximation algorithms to compute cache hit probabilities and most notably to time-to-live (TTL) approximations.

The first TTL approximation was introduced for the least recently used (LRU) policy under the Independent reference model (IRM) in [8] and more recently and independently in [6]. The main idea behind this approximation is that a LRU cache behaves similarly to a TTL cache. In a TTL cache, when an item enters the cache, it sets a deterministic timer with initial value T. When this timer expires the item is removed from the cache. If an item is requested before its timer expires, its timer is reset to T. When T is fixed, an item with popularity p_k is present in the cache at a random point in time with probability $1 - e^{-p_k T}$

*Corresponding author
Email addresses: nicolas.gast@inria.fr (Nicolas Gast), benny.vanhoudt@uantwerpen.be (Benny Van Houdt)

and $\sum_{k=1}^{N}[1-e^{-p_k T}]$ is the average number of items in the cache. The TTL approximation [8, 6] consists in approximating a LRU cache of size m by a TTL cache with characteristic time $T(m)$, where $T(m)$ is the unique solution of the fixed point equation

$$m = \sum_{k=1}^{N}(1-e^{-p_k T}). \tag{1}$$

The above TTL approximation for LRU can easily be generalized to renewal requests as well as to other simple variations of LRU and RANDOM under both IRM and renewal requests, as well as to certain network setups [3, 9, 17, 18]. All of these TTL approximations have been shown to be (very) accurate by means of numerical examples, but except for LRU in [8, 10, 14], no theoretical support was provided thus far.

In this paper we introduce TTL approximations for two classes of cache replacement algorithms that are variants of LRU. The first class, called LRU($\mathbf{m}$), dates back to the 1980s [1], while the second, called h-LRU, was introduced in [15, 17]. In fact, a TTL approximation for h-LRU was also introduced in [17], but this approximation relies on an additional approximation of independence between the different lists when $h > 2$. As we will show in the paper, this implies that the approximation error does not reduce to zero as the cache becomes large.

In this paper we make the following contributions:

- We present a TTL approximation for LRU($\mathbf{m}$) and h-LRU that is applicable when the request process of an item is a Markovian arrival process (MAP). This includes any phase-type renewal process and the IRM model. In the special case of the IRM model, we derive simple closed-form expressions for the fixed point equations.

- Our TTL approximation for h-LRU can be computed in linear time in h and appears to be asymptotically exact as the cache size and the number of items grow, in contrast to the TTL approximation in [17] for $h > 2$. Numerical results for the TTL approximation for LRU($\mathbf{m}$) also suggest that it is asymptotically exact.

- We prove that, under the IRM model, the transient behavior of both h-LRU and LRU($\mathbf{m}$) converges to the unique solution of a system of ODEs as the cache size and the number of items go to infinity. Our TTL approximations correspond to the unique fixed point of the associated systems of ODEs. This provides additional support for the claim that our TTL approximations are asymptotically exact and is the main technical contribution of the paper.

- We validate the accuracy of the TTL approximation. We show that h-LRU and LRU($\mathbf{m}$) perform alike in terms of the hit probability under both synthetic and trace-based workloads, while less work is required for LRU($\mathbf{m}$) when a hit/miss occurs.

- We indicate that both h-LRU and LRU($\mathbf{m}$) can exploit correlation in consecutive inter-request times of an item, while the hit probability of LRU is insensitive to this type of correlation.

- We show how partitioning the cache into parts – each being dedicated to a particular content provider – can improve the hit probability. It is shown in [7] that when using LRU and under an IRM request process, there exists an optimal partition of the cache that does not decrease the hit rate compared to a shared cache. Our numerical observations suggest that this is also the case for MAP arrivals and h-LRU. The gain, however, appears to be limited when the cache size is large and the optimal splitting size is very sensitive to the parameters.

The paper is structured as follows. We recall the definitions of LRU($\mathbf{m}$) and h-LRU in Section 2. We show how to build and solve the TTL approximation for LRU($\mathbf{m}$) in Section 3.1, and for h-LRU in Section 3.2. We demonstrate the accuracy of the TTL approximation for any finite time period in Section 4. We compare LRU($\mathbf{m}$) and h-LRU in Section 5, by using synthetic data and real traces. We study cache partitioning in Section 6. We conclude in Section 7.

2. Replacement Algorithms

We consider two families of cache replacement algorithms: h-LRU, introduced in [15, 17], and LRU($\mathbf{m}$), introduced in [1, 11]. Both operate on a cache that can store up to m items and both are variants of LRU, which replaces the least-recently-used item in the cache. One way to regard LRU is to think of the cache as an ordered list of m items, where the i-th position is occupied by the i-th most-recently-used item. When a miss occurs, the item in the last position of the list is removed and the requested item is inserted at the front of the list. If a hit occurs on the item in position i, item i moves to the front of the list, meaning the items in position 1 to $i-1$ move back one position.

The h-LRU replacement algorithm. h-LRU manages a cache of size m by making use of $h-1$ additional virtual lists of size m (called list 1 to list $h-1$) in which only meta-data is stored and one list of size m that corresponds to the actual cache (called list h). Each list is ordered, and the item in the ith position of list ℓ is the ith most-recently-used item among the items in list ℓ. When item k is requested, two operations are performed:

- For each list ℓ in which item k appears (say in a position i), the item k moves to the first position of list ℓ and the items in positions 1 to $i-1$ move back one position.
- For each list ℓ in which item k does not appear *but appears in list* $\ell-1$, item k is inserted in the first position of list ℓ, all other items of list ℓ are moved back one position and the item that was in position m of list ℓ is discarded from list ℓ.

List 1 of h-LRU behaves exactly as LRU, except that only the meta-data of the items is stored. Also, an item can appear in any subset of the h lists at the same time. This implies that a request can lead to as many as h list updates. Note that while there is no need for all of the h lists to have the same size m, we restrict ourselves to this setting (as in [17]).

The LRU($\mathbf{m}$) replacement algorithm. LRU($\mathbf{m}$) makes use of h lists of sizes $m_1, \ldots, m_h$, where the first few lists may be virtual, i.e., contain meta-data only. If the first v lists are virtual we have $m_{v+1} + \cdots + m_h = m$ (that is, only the items in lists $v+1$ to h are stored in the cache). With LRU($\mathbf{m}$) each item appears in at most one of the h lists at any given time. Upon each request of an item:

- If this item is not in any of the h lists, it moves to the first position of list 1 and all other items of list 1 move back one position. The item that was in position m_1 of list 1 is discarded.
- If this item is in position i of a list $\ell < h$, it is removed from list ℓ and inserted in the first position of list $\ell+1$. All other items of list $\ell+1$ move back one position and the item in the last position of list $\ell+1$ is removed from list $\ell+1$ and inserted in the first position of list ℓ. All previous items from position 1 to $i-1$ of list ℓ move back one position.
- If this item is in position i of list h, then this item moves to the first position of list h. All items that are in position 1 to $i-1$ of list h move back one position.

When using only one list, LRU($\mathbf{m}$) coincides with LRU, and therefore with 1-LRU.

3. TTL approximations

3.1. TTL approximation for LRU(m)

3.1.1. IRM setting

Under the IRM model the string of requested items is a set of i.i.d. random variables, where item k is requested with probability p_k. As far as the hit probability is concerned this corresponds to assuming that item k is requested according to a Poisson process with rate p_k.

The TTL approximation for LRU($\mathbf{m}$) consists in assuming that, when an item is not requested, the time it spends in list ℓ is deterministic and independent of the item. We denote this characteristic time by T_ℓ. Let t_n be the n-th time that item k is either requested or moves from one list to another list (where we state

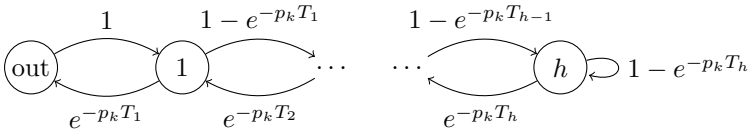


Figure 1: Discrete-time Markov models that represents how Item k moves between lists in the TTL approximation of LRU(m).

that an item is part of list 0 when not in the cache). Using the above assumption, we define an $h+1$ states discrete-time Markov chain $(X_n)_{n\geq 0}$, where X_n is equal to the list id of the list containing item k at time t_n.

With probability $e^{-p_kT_\ell}$ the time between two requests for item k exceeds T_ℓ. Therefore $e^{-p_kT_\ell}$ is the probability that an item part of list $\ell > 0$ moves to list $\ell - 1$, while with probability $1 - e^{-p_kT_\ell}$ a hit occurs and the item moves to list $\ell + 1$ if $\ell < h$. In other words, the transition matrix of $(X_n)_n$ is

$$P_k = \begin{bmatrix} 0 & 1 & & & \\ e^{-p_kT_1} & 0 & 1-e^{-p_kT_1} & & \\ & \ddots & \ddots & \ddots & \\ & & e^{-p_kT_{h-1}} & 0 & 1-e^{-p_kT_{h-1}} \\ & & & e^{-p_kT_h} & 1-e^{-p_kT_h} \end{bmatrix}.$$

The Markov chain X_n is a discrete-time birth-death process, represented in Figure 1. Hence, its steady state vector $(\pi_{k,0}, \pi_{k,1}, \ldots, \pi_{k,h})$ obeys

$$\pi_{k,\ell} = \pi_{k,0} \frac{\prod_{s=1}^{\ell-1}(1-e^{-p_kT_s})}{\prod_{s=1}^{\ell} e^{-p_kT_s}} = \pi_{k,0} e^{p_kT_\ell} \prod_{s=1}^{\ell-1}(e^{p_kT_s}-1), \tag{2}$$

for $\ell = 1, \ldots, h$.

Further for $\ell \in \{1, \ldots, h\}$, the average time spent in list ℓ is the expectation of the minimum between an exponential variable of parameter p_k and T_ℓ. Hence:

$$\begin{aligned} E[t_{n+1} - t_n | X_n = \ell] &= \int_{t=0}^{\infty} \mathbf{P}\left[t_{n+1} - t_n \geq t | X_n = \ell\right] dt \\ &= \int_{t=0}^{T_\ell} e^{-p_kt} dt \\ &= \frac{1 - e^{-p_kT_\ell}}{p_k}, \end{aligned}$$

and $E[t_{n+1} - t_n | X_n = 0] = 1/p_k$. Combined with (2), this implies that when observing the system at a random point in time, item k is in list $\ell \geq 1$ with probability

$$\frac{\pi_{k,\ell} E[t_{n+1} - t_n | X_n = \ell]}{\sum_{j=0}^{h} \pi_{k,j} E[t_{n+1} - t_n | X_n = j]} = \frac{(e^{p_kT_1}-1)\ldots(e^{p_kT_\ell}-1)}{1 + \sum_{j=1}^{h}(e^{p_kT_1}-1)\ldots(e^{p_kT_j}-1)}.$$

The expected number of items part of list ℓ is the sum of the previous expression over all items k. As for the TTL approximation, setting this sum equal to m_ℓ leads to the following set of fixed point equations for T_1 to T_h:

$$m_\ell = \sum_{k=1}^{n} \frac{(e^{p_kT_1}-1)\ldots(e^{p_kT_\ell}-1)}{1 + \sum_{j=1}^{h}(e^{p_kT_1}-1)\ldots(e^{p_kT_j}-1)}. \tag{3}$$

An iterative algorithm used to determine a solution of this set of fixed point equations is presented in Section 4.1.1. In the next section we generalize this approximation to MAP arrivals.

Remark. It is interesting to note the similarity of the above set of fixed point equations with the set of fixed point equations presented in [11] for the RAND(m) replacement algorithm. The RAND(m) algorithm works in the same manner as the LRU(m) algorithm, except that when a hit occurs on item i in a list $\ell < h$, item i is switched with a random item in list $\ell + 1$; when a miss occurs the missed item replaces a random item in list 1 and when a hit occurs in list h nothing changes. In this case the set of fixed point equations for the mean field model presented in [11] can be written as

$$m_\ell = \sum_{k=1}^{n} \frac{(p_k T_1) \dots (p_k T_\ell)}{1 + \sum_{j=1}^{h} (p_k T_1) \dots (p_k T_j)}, \tag{4}$$

which is identical to (3) if we replace the factors of the form $p_k T_i$ by $e^{p_k T_i} - 1$. Equation (4) can also be derived in a manner similar to (3) if we replace the assumption that an item spends a deterministic time T_ℓ in list ℓ by the assumption that the time spend in list ℓ is exponential with mean T_ℓ. More specifically, the rate matrix for the continuous-time Markov chain that keeps track of the position of item k for the RAND(m) replacement algorithm is given by

$$Q_k = \begin{bmatrix} -p_k & p_k & & & & \\ \mu_1 & -\mu_1 - p_k & p_k & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mu_{h-1} & -\mu_{h-1} - p_k & p_k & \\ & & & \mu_h & -\mu_h \end{bmatrix},$$

where $\mu_\ell = 1/T_\ell$. The fixed point equations (4) now readily follow from the steady state probabilities of this birth-death process.

3.1.2. MAP arrivals

We now assume that the times that item k is requested are captured by a Markovian Arrival Process (MAP). MAPs have been developed with the aim of fitting a compact Markov model to workloads with statistical correlations and non-exponential distributions [5, 19]. A MAP is characterized by two $d \times d$ matrices $(D_0^{(k)}, D_1^{(k)})$, where the entry (j, j') of $D_1^{(k)}$ is the transition rate from state j to j' that is accompanied by an arrival and the entry (j, j') of $D_0^{(k)}$ is the transition rate from state j to j' (with $j \neq j'$) without arrival. Let $\phi^{(k)}$ be the stationary distribution of this process, *i.e.*, the unique stochastic vector such that $\phi^{(k)}(D_0^{(k)} + D_1^{(k)}) = 0$. Note that the request rate λ_k of item k can be expressed as $\lambda_k = \phi^{(k)} D_1^{(k)} \mathbf{e}$, where $\mathbf{e}$ is a column vector of ones. Setting the matrices $D_0^{(k)} = -p_k$ and $D_1^{(k)} = p_k$ corresponds to the IRM case and letting $D_1^{(k)} = -D_0^{(k)} \mathbf{e} v^{(k)}$, with $v^{(k)}$ is a stochastic vector, implies that item k is requested according to a phase-type renewal process characterized by $(v^{(k)}, D_0^{(k)})$ (where $v^{(k)}$ holds the initial phase probabilities and $D_0^{(k)}$ is the subgenerator matrix of the phase-type inter-arrival time distribution).

Extending the previous section, we define a discrete-time Markov chain (X_n, S_n), where X_n is the list in which item k appears and S_n is the state of the MAP process at time t_n (recall that t_n is the n-th time that item k is requested or is moved from one list to another list). This Markov chain has $d(h+1)$ states and its transition probability matrix P_k^{MAP} is given by

$$P_k^{MAP} = \begin{bmatrix} 0 & (-D_0^{(k)})^{-1} D_1^{(k)} & & & \\ e^{D_0^{(k)} T_1} & 0 & A_1^{(k)} & & \\ & \ddots & \ddots & \ddots & \\ & & e^{D_0^{(k)} T_{h-1}} & 0 & A_{h-1}^{(k)} \\ & & & e^{D_0^{(k)} T_h} & A_h^{(k)} \end{bmatrix},$$

where

$$A_\ell^{(k)} = \int_{t=0}^{T_\ell} e^{D_0^{(k)} T_\ell} dt D_1^{(k)} = (I - e^{D_0^{(k)} T_\ell})(-D_0^{(k)})^{-1} D_1^{(k)}.$$

Indeed $(e^{D_0^{(k)}T_\ell})_{i,j}$ is the probability that there are no arrivals in an interval of length T_ℓ, the MAP state at the start of the interval equals i and the MAP state at the interval is j. Similarly, $(A_\ell^{(k)})_{i,j}$ is the probability of starting in the MAP state i, having an arrival at time $t < T_\ell$, while the MAP state equals j at time t.

Due to the block structure of P_k^{MAP}, its steady state vector $(\tilde{\pi}_{k,0}, \tilde{\pi}_{k,1}, \ldots, \tilde{\pi}_{k,h})$ obeys [16]

$$\tilde{\pi}_{k,\ell} = \tilde{\pi}_{k,0} \prod_{s=1}^{\ell} R_{k,s}, \tag{5}$$

for $\ell = 1, \ldots, h$, where the matrices $R_{k,s}$ can be computed recursively as

$$R_{k,h} = A_{h-1}^{(k)}(I - A_h^{(k)})^{-1}, \tag{6}$$

$$R_{k,\ell} = A_{\ell-1}^{(k)} \left(I - R_{k,\ell+1} e^{D_0^{(k)} T_{\ell+1}}\right)^{-1}, \tag{7}$$

for $\ell = 1, \ldots, h-1$ and $h > 1$.

We also define the average time $(N_{k,\ell})_{j,j'}$ that item k spends in state j' in (t_n, t_{n+1}) given that $X_n = (\ell, j)$, for $j, j' \in \{1, \ldots, d\}$. Let $N_{k,\ell}$ be the matrix with entry (j, j') equal to $(N_{k,\ell})_{j,j'}$, then

$$N_{k,\ell} = \int_{t=0}^{T_\ell} e^{D_0^{(k)} t} dt = (I - e^{D_0^{(k)} T_\ell})(-D_0^{(k)})^{-1},$$

for $\ell \geq 1$ and $N_{k,0} = (-D_0^{(k)})^{-1}$. The fixed point equations for T_1 to T_h given in (3) generalize to

$$m_\ell = \sum_{k=1}^{n} \frac{\tilde{\pi}_{k,\ell} N_{k,\ell} \mathbf{e}}{\sum_{j=0}^{h} \tilde{\pi}_{k,j} N_{k,j} \mathbf{e}}. \tag{8}$$

The hit probability h_ℓ in list ℓ can subsequently be computed as

$$h_\ell = \frac{1}{\sum_{s=1}^{n} \lambda_s} \sum_{k=1}^{n} \frac{\tilde{\pi}_{k,\ell} N_{k,\ell} D_1^{(k)} \mathbf{e}}{\sum_{j=0}^{h} \tilde{\pi}_{k,j} N_{k,j} \mathbf{e}}, \tag{9}$$

for $\ell = 0, \ldots, h$, where $\lambda_k / \sum_{s=1}^{n} \lambda_s$ is the probability that the requested item is item k and $\tilde{\pi}_{k,\ell} N_{k,\ell} D_1^{(k)} \mathbf{e} / \lambda_k (\sum_{j=0}^{h} \tilde{\pi}_{k,j} N_{k,j} \mathbf{e}$ is the probability that item k is in list ℓ if we observe the Markov chain of item k only at a request times (which differs from $\tilde{\pi}_{k,\ell} N_{k,\ell} \mathbf{e}$ in case of MAP arrivals).

Remark. As in the IRM case we can also derive a set of fixed point equations for the RAND(m) replacement algorithm with MAP arrivals by assuming an exponential sojourn time in list ℓ with mean T_ℓ. The continuous time Markov chain that keeps track of the list that contains item k in case of MAP arrivals is given by

$$Q_k^{MAP} = \begin{bmatrix} D_0^{(k)} & D_1^{(k)} & & & & \\ \mu_1 I & D_0^{(k)} - \mu_1 I & D_1^{(k)} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mu_{h-1} I & D_0^{(k)} - \mu_{h-1} I & D_1^{(k)} & \\ & & & \mu_h I & D^{(k)} - \mu_h I \end{bmatrix}, \tag{10}$$

and the set of fixed point equations to determine T_1 to T_h readily follow.

3.2. TTL approximation for h-LRU

3.2.1. IRM setting

As in [17], our approximation for h-LRU is obtained by assuming that an item that is not requested spends a deterministic time T_ℓ in list ℓ, independently of the identity of this item. For now we assume that $T_1 \leq T_2 \leq \ldots \leq T_h$. We will show that the fixed point solutions for T_1 to T_h always obey these inequalities.

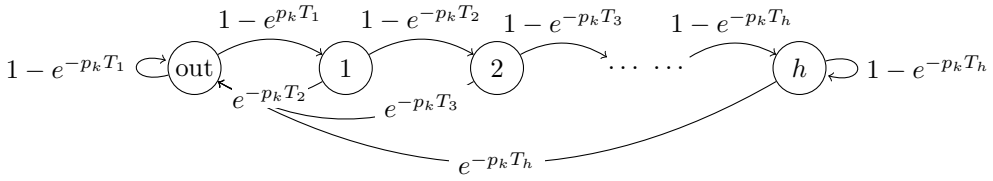


Figure 2: Discrete-time Markov models that represents the highest lits in which Item k is in the TTL approximation of h-LRU.

We start by defining a discrete-time Markov chain $(Y_n)_{n\geq 0}$ by observing the system just prior to the time epochs that item k is requested. The state space of the Markov chain is given by $\{0,\ldots,h\}$. We say that $Y_n = 0$ if item k is not in any of the lists (just prior to the nth request). Otherwise, $Y_n = \ell$ if item k is in list ℓ, but is not in any of the lists $\ell+1$ to h. In short, the state of the Markov chain is the largest id of the lists that contain item k.

If $Y_n = \ell$, then with probability $1-e^{-p_kT_\ell}$, item k is requested before time T_ℓ in which case we have $Y_{n+1} = \ell+1$. Otherwise, due to our assumption that $T_\ell \geq T_{\ell-1} \geq \ldots \geq T_1$ we have $Y_{n+1} = 0$ as in this case item k was discarded from all lists. Therefore the transition probability matrix $\bar{P}_{h,k}$ of the $h+1$ state Markov chain $(Y_n)_{n\geq 0}$ is given by

$$\bar{P}_{h,k} = \begin{bmatrix} e^{-p_kT_1} & 1-e^{-p_kT_1} & & & \\ e^{-p_kT_2} & & 1-e^{-p_kT_2} & & \\ \vdots & & & \ddots & \\ e^{-p_kT_h} & & & & 1-e^{-p_kT_h} \\ e^{-p_kT_h} & & & & 1-e^{-p_kT_h} \end{bmatrix}. \tag{11}$$

This Markov chain is represented in Figure 2.

Let $\bar{\pi}^{(h,k)} = (\bar{\pi}_0^{(h,k)},\ldots,\bar{\pi}_h^{(h,k)})$ be the stationary vector of $\bar{P}_{h,k}$, then the balance equations imply:

$$\bar{\pi}_\ell^{(h,k)} = \xi_\ell \bar{\pi}_0^{(h,k)} \prod_{s=1}^{\ell}(1-e^{-p_kT_s}), \tag{12}$$

for $\ell = 1,\ldots,h$, where $\xi_\ell = 1$ for $\ell < h$ and $\xi_h = e^{p_kT_h}$. The probability $\bar{\pi}_h^{(h,k)}$ that item k is in the cache just before a request (which by the PASTA property is also the steady-state probability for the item to be in the cache) can therefore be expressed as

$$\bar{\pi}_h^{(h,k)} = \frac{\prod_{s=1}^{h}(1-e^{-p_kT_s})}{\prod_{s=1}^{h}(1-e^{-p_kT_s}) + e^{-p_kT_h}\left(1+\sum_{\ell=1}^{h-1}\prod_{s=1}^{\ell}(1-e^{-p_kT_s})\right)}. \tag{13}$$

Due to the nature of h-LRU, T_1 can be found from analyzing LRU, T_2 from 2-LRU, etc. Thus, it suffices to define a fixed point equation for T_h. Under the IRM model this is simply $m = \sum_{k=1}^{n}\bar{\pi}_h^{(h,k)}$, due to the PASTA property. These fixed point equations can be generalized without much effort to renewal arrivals as explained in Appendix A.

Proposition 1. *The fixed point equation $m = \sum_{k=1}^{n}\bar{\pi}_h^{(h,k)}$ has a unique solution T_h which is such that $T_h \geq T_{h-1}$.*

Proof. By reordering the terms in the denominator the fixed point equation for $h \geq 2$ can be written as $m = f_h(T_h)$, where

$$f_h(x) = \sum_{k=1}^{n} \frac{(1-e^{-p_kx})\prod_{s=1}^{h-1}e_{k,s}}{\prod_{s=1}^{h-1}e_{k,s} + e^{-p_kx}\left(1+\sum_{j=1}^{h-2}\prod_{s=1}^{j}e_{k,s}\right)},$$

with $e_{k,s} = (1 - e^{-p_k T_s})$. The function $f_h(x)$ is clearly an increasing function in x and therefore $m = f(x)$ has a unique solution T_h. Further,

$$f_h(T_{h-1}) = \sum_{k=1}^{n} \frac{(1 - e^{-p_k T_{h-1}}) \prod_{s=1}^{h-1} e_{k,s}}{\prod_{s=1}^{h-1} e_{k,s} + e^{-p_k T_{h-1}} \left(1 + \sum_{j=1}^{h-2} \prod_{s=1}^{j} e_{k,s}\right)}$$
$$< \sum_{k=1}^{n} \frac{\prod_{s=1}^{h-1} e_{k,s}}{\prod_{s=1}^{h-1} e_{k,s} + e^{-p_k T_{h-1}} \left(1 + \sum_{j=1}^{h-2} \prod_{s=1}^{j} e_{k,s}\right)} = \sum_{k=1}^{n} \bar{\pi}_{h-1}^{(h-1,k)} = m,$$

meaning $T_h \geq T_{h-1}$. □

The above fixed point equations are derived from $\bar{P}_{h,k}$, which relied on the assumption that $T_1 \leq \ldots \leq T_h$. If we do not make any assumptions on the T_i values we need to consider a 2^h state Markov chain (as an item can be part of any subset of the h lists) and derive a set of m fixed point equations from its steady state. The next proposition shows that the solution of this set of fixed equations is such that $T_1 \leq \ldots \leq T_h$, which shows that we can compute the T_i values from the $h+1$ state Markov chain without loss of generality.

Proposition 2. *Any solution to the fixed point equations for the 2^h state Markov chain is such that $T_1 \leq T_2 \leq \ldots \leq T_h$.*

Proof. Using induction we prove that the fixed point solutions obey $T_1 \leq \ldots \leq T_h$. We assume that $T_1 \leq \ldots \leq T_{h-1}$ (which trivially holds for $h = 2$) and show that the fixed point equation for T_h does not have a solution for $T_h \in (0, T_{h-1})$. When $T_1 \leq \ldots \leq T_{h-1}$ and $T_h < T_{h-1}$ we still obtain a $h+1$ state discrete-time Markov chain by observing the largest id of the list that contains item k just prior to the time epochs that item k is requested. The transition probability matrix is identical to $\bar{P}_{h,k}$ except that the last two rows need to be modified. The key thing to note is that when $T_h < T_{h-1}$ item k is part of list $h-1$ whenever it is part of list h. Therefore, if item k enters (or remains in) list h upon arrival it is still in list h when the next request for item k occurs with probability $1 - e^{-p_k T_h}$, while with probability $e^{-p_k T_h}(1 - e^{-p_k (T_{h-1} - T_h)})$ it is removed from list h, but still in list $h-1$. Finally with probability $e^{-p_k T_{h-1}}$ the item is also removed from list $h-1$ in which case it is no longer part of any list as $T_1 \leq \ldots \leq T_{h-1}$ (hence, the relative order of T_{h-1} and T_i for $i < h-1$ is irrelevant). As such the last two rows of the transition probability matrix are both equal to

$$(e^{-p_k T_{h-1}}, 0, \ldots, 0, e^{-p_k T_h} - e^{-p_k T_{h-1}}, 1 - e^{-p_k T_h}).$$

Let $(\hat{\pi}_0^{(h,k)}, \ldots, \hat{\pi}_h^{(h,k)})$ be the invariant vector of this modified Markov chain, then it is easy to see that

$$\hat{\pi}_{h-1}^{(h,k)} + \hat{\pi}_h^{(h,k)} = \bar{\pi}_{h-1}^{(h-1,k)},$$

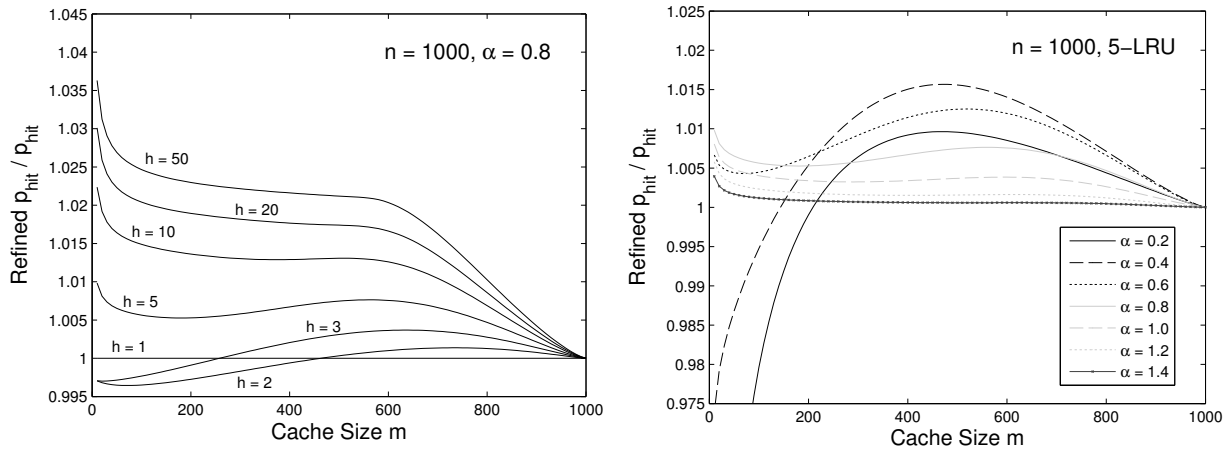
as lumping the last two states into a single state results in the matrix $\bar{P}_{h-1,k}$. Hence the fixed point equation $\sum_{k=1}^{n} \hat{\pi}_h^{(h,k)} = m$ cannot have a solution as

$$\sum_{k=1}^{n} \hat{\pi}_h^{(h,k)} < \sum_{k=1}^{n} (\hat{\pi}_{h-1}^{(h,k)} + \hat{\pi}_h^{(h,k)}) = \sum_{k=1}^{n} \bar{\pi}_{h-1}^{(h-1,k)} = m.$$

□

When $h = 2$ Equation (13) simplifies to $(1 - e^{-p_k T_1})(1 - e^{-p_k T_2})/(1 - e^{-p_k T_1} + e^{-p_k T_2})$ which coincides with the hit probability of the so-called *refined* model for 2-LRU presented in [17, Eqn (9)]. For $h > 2$ only an approximation that relied on an additional approximation of independence between the h lists was presented in [17, see Eqn (10)]. In Figure 3 we plotted the ratio between our approximation and the one based on (10) of [17]. The results indicate that the difference grows with increasing h and decreasing the Zipf parameter α. In other words, the difference decreases as the popular items gain in popularity.

As (13) does not rely on the additional independence approximation, we expect that its approximation error is smaller and even tends to zero as m tends to infinity. This is confirmed by simulation and we list a small set of randomly chosen examples in Table 1 to illustrate.



(a) $\alpha = 0.8$ (b) $h = 5$

Figure 3: Ratio of the approximation of the hit rate for h-LRU under the IRM model based on (13) and (10) of [17] as a function of the cache size with $n = 1000$ items with a Zipf-like popularity distribution with parameter α.

3.2.2. MAP arrivals

For order d MAP arrivals, characterized by $(D_0^{(k)}, D_1^{(k)})$ for item k, we obtain a $(h+1)d$ state MC by additionally keeping track of the MAP state immediately after the requests (this construction is done by assuming that, as for IRM arrivals, $T_1 \leq \ldots \leq T_h$ for the solutions to the fixed point equations. This can be proven using a monotonicity argument similar to the one used in Proposition 2). The transition probability matrix has the same form as $\bar{P}_{h,k}$, we only need to replace the probabilities of the form $e^{-p_k T_\ell}$ by $e^{D_0^{(k)} T_\ell}(-D_0^{(k)})^{-1} D_1^{(k)}$ and $1-e^{-p_k T_\ell}$ by $(I-e^{D_0^{(k)} T_\ell})(-D_0^{(k)})^{-1} D_1^{(k)}$. Note that $(e^{D_0^{(k)} T_\ell}(-D_0^{(k)})^{-1} D_1^{(k)})_{i,j}$ is the probability that we start in MAP state i, the next request for item k occurs after time T_ℓ and the MAP state when item k is requested next is j. In order to express the fixed point equations we need to determine the probability that item k is in the cache at a random point in time as the PASTA property does not hold in case of MAP arrivals. Using a standard argument we have that the probability that item k is in the cache at a random point in time equals

$$\frac{(\bar{\pi}_{h-1}^{(h,k)} + \bar{\pi}_h^{(h,k)}) \left(\int_0^{T_h} e^{D_0^{(k)} u} du \right) \mathbf{e}}{1/\lambda_k},$$

where λ_k is the request rate of item k and entry j of $\bar{\pi}_\ell^{(h,k)}$ is the probability that item k is in list ℓ (but not in lists $\ell+1, \ldots, h$) just prior to a request of item k and the MAP state immediately after the request is j. The fixed point equation for determining T_h can therefore be expressed as

$$m = \sum_{k=1}^{n} \frac{(\bar{\pi}_{h-1}^{(h,k)} + \bar{\pi}_h^{(h,k)})(I - e^{D_0^{(k)} T_h})(-D_0^{(k)})^{-1}\mathbf{e}}{1/\lambda_k}, \tag{14}$$

where λ_k is the request rate of item k. Due to the structure of the transition probability matrix of the $(h+1)d$ state Markov chain, the vectors $\bar{\pi}_\ell^{(h,k)}$ obey

$$\bar{\pi}_\ell^{(h,k)} = \bar{\pi}_0^{(h,k)} \left(\prod_{s=1}^{\ell} (I - e^{D_0^{(k)} T_s})(-D_0^{(k)})^{-1} D_1^{(k)} \right) \Xi_\ell,$$

for $\ell = 1, \ldots, h$, where $\Xi_\ell = I$ for $\ell < h$ and $\Xi_h = (I - (I - e^{D_0^{(k)} T_h})(-D_0^{(k)})^{-1} D_1^{(k)})^{-1}$. Finally, let $\nu^{(k)}$ be the stochastic invariant vector of $(-D_0^{(k)})^{-1} D_1^{(k)}$, that is, its d entries contain the probabilities to be in state 1 to d immediately after an arrival. Hence, $\bar{\pi}_0^{(h,k)}$ can be computed by noting that $\sum_{\ell=0}^{h} \bar{\pi}_\ell^{(h,k)} = \nu^{(k)}$.

h	Simul.	Eq. (10) of [17] (err)	Eq. (13) (err)
		$n = 1000$, $m = 10$	
2	0.19826	0.20139 (+1.576%)	0.20080 (+1.277%)
3	0.21139	0.21399 (+1.230%)	0.21336 (+0.932%)
5	0.21863	0.21780 (−0.381%)	0.21994 (+0.598%)
10	0.22357	0.21912 (−1.991%)	0.22402 (+0.201%)
		$n = 1000$, $m = 100$	
2	0.47610	0.47808 (+0.415%)	0.47641 (+0.064%)
3	0.49535	0.49695 (+0.322%)	0.49579 (+0.089%)
5	0.50777	0.50521 (−0.504%)	0.50806 (+0.056%)
10	0.51506	0.50796 (−1.380%)	0.51552 (+0.088%)
		$n = 10000$, $m = 100$	
2	0.27322	0.27404 (+0.302%)	0.27352 (+0.109%)
3	0.28453	0.28533 (+0.281%)	0.28477 (+0.085%)
5	0.29048	0.28873 (−0.602%)	0.29065 (+0.061%)
10	0.29427	0.28991 (−1.483%)	0.29430 (+0.011%)
		$n = 10000$, $m = 1000$	
2	0.52589	0.52746 (+0.300%)	0.52596 (+0.013%)
3	0.54340	0.54453 (+0.207%)	0.54348 (+0.015%)
5	0.55452	0.55199 (−0.455%)	0.55457 (+0.009%)
10	0.56124	0.55447 (−1.206%)	0.56130 (+0.012%)

Table 1: Accuracy of the two approximations for the hit probability of h-LRU under the IRM model with a Zipf-like popularity distribution with $\alpha = 0.8$. Simulation is based on 10 runs of $10^3 n$ requests with a warm-up period of 33%.

4. Asymptotic Exactness of the approximations

In this section, we provide evidence that the TTL approximations presented in the previous section are asymptotically exact as cache size and the number of items tends to infinity. We first provide numerical evidence. We then show that the transient behavior of LRU($\mathbf{m}$) and h-LRU converges to a system of ODEs. By using a change of variable, these ODE can be transformed into PDEs whose fixed points are our TTL approximations.

4.1. Numerical validation

4.1.1. Numerical procedure to solve the fixed-point equations

The only costly operation when evaluating the performance of h-LRU and LRU(m) is to solve the fixed point equations (14) and (8). As we explain below, for h-LRU computing $T_1, \ldots, T_h$ corresponds to solving h one dimensional problems whereas for LRU($\mathbf{m}$), computing $T_1 \ldots T_h$ corresponds to solving a single h-dimensional one.

The computation time for h-LRU scales linearly with the number of lists: by construction, the first $h-1$ lists of a h-LRU cache behave like an $(h-1)$-LRU cache. Once T_{h-1} has been computed, the right-hand side of the fixed point equation (14) is increasing in T_h and can therefore be easily solved. For LRU($\mathbf{m}$) solving the fixed point equations is more costly. In our experiments the fixed point of Equation (8) is computed by an iterative procedure that updates the values T_ℓ in a round-robin fashion. This iterative procedure is detailed in Algorithm 1. It works well for up to $h \approx 5$ lists, but becomes very slow for a large number of lists. At this stage we do not have a proof that this algorithm converges, but it appears to do so in practice.

4.1.2. Synthetic data-set

We assume that the inter-request times of item k follow a hyperexponential distribution with rate zp_k in state one and p_k/z in state two, while the popularity distribution is a Zipf-like distribution with parameter

Input: $D_0, D_1, m_1, \ldots, m_h, \epsilon$
Output: fixed point solution $\hat{T}_1, \ldots, \hat{T}_m$

for $\ell = 1$ *to* h **do**
  $\hat{T}_\ell = n$;
end
$\hat{T}_{h+1} = \infty$, $x = 1$;
while $x > \epsilon$ **do**
  for $\ell = 1$ *to* h **do**
    Find $x \in (-\hat{T}_\ell, \hat{T}_{\ell+1})$ such that $(T_1, \ldots, T_h)$ equal to $(\hat{T}_1, \ldots, \hat{T}_\ell + x, \hat{T}_{\ell+1} - x, \ldots, \hat{T}_h)$ minimizes $|m_\ell -$ rhs of (8)$|$;
    $\hat{T}_\ell = \hat{T}_\ell + x$; $\hat{T}_{\ell+1} = \hat{T}_{\ell+1} - x$;
  end
end

Algorithm 1: Iterative algorithm to compute the fixed point of (8).

α, i.e., $p_k = (1/k^\alpha)/\sum_{i=1}^{n} 1/i^\alpha$. Correlation between consecutive inter-request times is introduced using the parameter $q \in (0, 1]$. More precisely, let

$$D_0^{(k)} = p_k \begin{bmatrix} -z & 0 \\ 0 & -1/z \end{bmatrix},$$

and

$$D_1^{(k)} = p_k \left(q \begin{bmatrix} z \\ 1/z \end{bmatrix} \begin{bmatrix} \frac{z}{1+z} & \frac{1}{1+z} \end{bmatrix} - (1-q) D_0^{(k)} \right).$$

The squared coefficient of variation (SCV) of the inter-request times of item k is given by $2(z^2 - z + 1)/z - 1$ and the lag-1 autocorrelation of inter-request times of item k is

$$\rho_1 = (1-q)\frac{(1-z)^2}{2(1-z)^2 + z}.$$

In other words the lag-1 autocorrelation decreases linearly in q and setting $q = 1$ implies that the arrival process is a renewal process with hyperexponential inter-request times. Setting $z = 1$ reduces the model to the IRM model.

4.1.3. Accuracy of the approximation for LRU(m) and h-LRU

To test the accuracy of our approximations, we implemented a stochastic simulator of h-LRU and LRU(m). We use the hyperexponential distribution described in the previous section.

In Table 2, we compare the accuracy of the model with time consuming simulations (based on 5 runs of $2 \cdot 10^6$ requests) for LRU($\mathbf{m}$). We observe a good agreement between the TTL approximation and simulation that tends to improve with the size of the system (i.e., when n increases from 100 to 1000).

For h-LRU, the TTL approximation for the IRM model was already validated by simulation in Table 1. Using the same numerical examples as for LRU($\mathbf{m}$) we now demonstrate the accuracy of the TTL approximation under MAP arrivals in Table 3. Simulation results are based on 5 runs containing $2 \cdot 10^6$ requests each. As for $LRU(\mathbf{m})$, the TTL approximation is in good agreement with the simulation and tend to be more accurate as the number of items grows.

4.2. Asymptotic behavior and TTL approximation

In this subsection, we construct two systems of ODEs: Equation (19) for h-LRU and Equation (26) for LRU($\mathbf{m}$). We prove that the solutions of these ODEs are approximations of the transient behavior

n	q	z	h_0 model (simu.)	h_1 model (simu.)	h_2 model (simu.)
100	1	2	0.26898 (0.27021)	0.19304 (0.19340)	0.53798 (0.53639)
		10	0.03712 (0.03723)	0.05889 (0.06106)	0.90399 (0.90171)
1000	1	2	0.22580 (0.22599)	0.16262 (0.16256)	0.61158 (0.61145)
		10	0.03112 (0.0310)	0.04963 (0.04969)	0.91925 (0.91923)
1000	0.1	2	0.21609 (0.21603)	0.14510 (0.14526)	0.63881 (0.63870)
		10	0.03006 (0.02984)	0.02044 (0.02032)	0.94950 (0.9498)

Table 2: Accuracy of probability h_ℓ of finding a requested item in list ℓ for LRU($\mathbf{m}$). In this example $\alpha = 0.8$, $h = 2$ and $m_1 = m_2 = n/5$ (*i.e.,* 20 or 200).

n	q	z	$h = 2$ model (simu.)	$h = 3$ model (simu.)
100	1	2	0.53619 (0.53449)	0.54292 (0.54150)
		10	0.88249 (0.87936)	0.83718 (0.83449)
1000	1	2	0.61028 (0.61016)	0.61605 (0.61587)
		10	0.90103 (0.90071)	0.86300 (0.86262)
1000	0.1	2	0.64744 (0.64807)	0.65841 (0.65899)
		10	0.94935 (0.94924)	0.94646 (0.94632)

Table 3: Accuracy of hit probability for h-LRU with MAP arrivals. In this example $\alpha = 0.8$ and $m = n/5$.

of LRU($\mathbf{m}$) and h-LRU that become exact as the popularity of the most popular item decreases to zero (regardless of the cache size). To ease the presentation, we present the convergence result when the arrivals follow a discrete-time IRM model: time is slotted and at each time-step item k has a probability p_k of being requested.

Theorem 1. *Consider the IRM model. Let $H_\ell(t)$ be the sum of the popularity of the items of list ℓ and $h_\ell(t)$ be the corresponding ODE approximation (Equation (19) for h-LRU and Equation (26) for LRU($\mathbf{m}$)). Then: for any time T, there exists a constant C such that*

$$\mathbf{E}\left[\sup_{t\leq T/\sqrt{\max_k p_k}} |H_\ell(t) - h_\ell(t)|\right] \leq C\sqrt{\max_k p_k},$$

where C does not depend on the probabilities $p_1 \dots p_n$, the cache size m or the number of items n.

Remarks:

- The above result concerns the transient regime of the hit rate. In each case, we will show that the ODE can be transformed into a PDE that has the same fixed point as the TTL approximation developed in Section 3. This does not fully guarantee the asymptotic exactness of the TTL approximation. To show that, one would in addition need to show that all trajectories of the PDE converge to their fixed point. We believe that this is the case but we have no proof of this result so far.

- Our proof of this result is to use an alternative representation of the state space that allows us to use techniques from stochastic approximation. We associate to each item k a variable $\tau_k(t)$ that we call the *request time* of item k and that is the time of the most recent request of item k before time t and an additional variable that tracks if an item appears in a list. Our approximation is given by an ordinary differential equation (ODE) on $x_{k,\ell,b}(t)$ that is an approximation of the probability that $\tau_k(t)$ is greater than b while appearing in a list ℓ. In each case, we show that the fixed point of the PDE corresponds to the TTL approximation of LRU($\mathbf{m}$) and h-LRU presented in Sections 3.1 and 3.2.

- This proof can be adapted to the case of MAP arrivals but at the price of more complex notations. Indeed, for IRM, our system of ODEs is given by the variables $x_{k,\ell,b}(t)$ (or $x_{k,b}(t)$ for LRU(m)) which are essentially an approximation of the probability for item k to be in a list ℓ while having been requested between time b and t. If the arrival process of an item is modeled by a MAP with d states, then our approximation would need to consider $x_{k,\ell,b,j}(t)$ which would approximate the probabilities for the MAP of item k to be in state j, for item k to be in list ℓ and having being requested between b and t.

4.2.1. Proof of Theorem 1: the case of LRU

Before presenting the complex cases of h-LRU and LRU($\mathbf{m}$), we first construct the ODE approximation for LRU. The main purpose of this section is to serve as a basis for the more complex cases of h-LRU and LRU($\mathbf{m}$). Note that in the simpler case of LRU the proof of the validity of the TTL approximation could rely on a more direct argument that uses a simple property of the steady state distribution: the items in the cache are the m most recently requested. This argument, used in [10, 14], makes an easy connection between the LRU cache and the TTL approximation cache: the TTL of a LRU-cache is the mth order statistics of n non-identically distributed, but independent random variables. For LRU(m) and h-LRU, there are strong dependencies between items that makes the approach of [10, 14] impossible.

The cache contains m items. We denote[1] by $\Theta(t) = \sup\{b : \sum_{k=1}^{n} \mathbf{1}_{\{\tau_k(t)\geq b\}} \geq m\}$ the request time of the mth most recently requested item at time t. When using LRU, an item k having a request time $\tau_k(t)$ greater or equal to $\Theta(t)$ is in the cache at time t. Let $H(t)$ be the sum of the popularities of items in the cache:

$$H(t) = \sum_{k=1}^{n} p_k \mathbf{1}_{\{\tau_k(t)\geq\Theta(t)\}}.$$

Our approximation of the probability for item k to have a request time after b, is given by the following ODE (for $b < t$):

$$\dot{x}_{k,b}(t) = p_k(1 - x_{k,b}(t)). \tag{15}$$

with the initial conditions: $x_{k,b}(0) = \mathbf{1}_{\{\tau_k(t)\geq b\}}$ and $x_{k,b}(b) = 0$ for $b > 0$. The initial condition $x_{k,b}(0) = \mathbf{1}_{\{\tau_k(t)\geq b\}}$ corresponds to the state of the cache at time 0. The initial condition $x_{k,b}(b) = 0$ for $b > 0$ indicate that at time b, no items have a request time higher than b.

By analogy with the stochastic system, let $\theta(t) = \sup\{b : \sum_{k=1}^{n} x_{k,b}(t) \geq m\}$, be the time at which the sum of $x_{k,b}(t)$ equals to m. The approximation of the hit ratio for LRU is then given by

$$h(t) = \sum_{k=1}^{n} p_k x_{k,\theta(t)}(t).$$

The key difficulty when comparing $H(t)$ and $h(t)$ is that the quantities $\mathbf{1}_{\{\tau_k(t)\geq\Theta(t)\}}$ and $x_{k,\theta(t)}(t)$ are not easily comparable. The key ingredient of our proof is then to use the same change of variables as in the proof of Theorem 6 of [11], which is to consider the variables $P_{\delta,b}(t)$ and $\rho_{\delta,b}(t)$:

$$P_{\delta,b}(t) = a^{1-\delta} \sum_{k=1}^{n} (p_k)^{\delta} \mathbf{1}_{\{\tau_k(t)\geq b\}} \qquad \text{and} \qquad \rho_{\delta,b}(t) = a^{1-\delta} \sum_{k=1}^{n} (p_k)^{\delta} x_{k,b}(t)$$

where $a := \max_{k=1}^{n} p_k$. These variables are defined for $\delta \in \{0, 1, \dots\}$ and $b \in \mathbf{Z}$. They live in a set of infinite dimension $\mathcal{P}$:

$$\mathcal{P} = \Big\{(P_{\delta,b})_{\delta,b} : \exists (x_{k,b}) \text{ non-increasing in } b, \text{ bounded by } 1$$
$$\text{such that for all } \delta, b\text{: } P_{\delta,b} = a^{1-\delta} \sum_{k=1}^{n} (p_k)^{\delta} x_{k,b} \Big\}.$$

[1] Throughout the paper $\mathbf{1}_{\{A\}}$ is the indicator function of an event A. It is equal to 1 if A is true and 0 otherwise.

We equip $\mathcal{P}$ with the L_∞ norm and denote $\|\rho\|_\infty = \sup_{\delta,b} |\rho_{\delta,b}|$ the norm of a vector $\rho \in \mathcal{P}$.

The proof of the theorem relies on the following result of stochastic approximation. For completeness, we provide a proof of Lemma 1 in Appendix B.

Lemma 1. *Let $f : \mathcal{P} \to \mathrm{span}(\mathcal{P})$ be a Lipchitz continuous function with constant aL such that $\sup_{x\in\mathcal{P}} \|f(x)\|_\infty \leq a \leq 1$ and $f(x) - x \in \mathcal{P}$. Let X be a $\mathcal{P}$-valued stochastic process adapted to a filtration $\mathcal{F}$ such that $\mathbf{E}\left[X(t+1) - X(t) \mid \mathcal{F}_t\right] = f(X(t))$ and $\mathbf{E}\left[\|X(t+1) - X(t)\|_\infty^2\right] \leq a^2$. Then, the ODE $\dot{x} = f(x)$ has a unique solution $x_{X(0)}$ that starts in $X(0)$ and for any $T > 0$,*

$$\mathbf{E}\left[\sup_{t\leq T/a} \left\|X(t) - x_{X(0)}(t)\right\|_\infty^2\right] \leq T(2L+1)\exp(2TL)a.$$

To apply this result, we use the fact that:

- The functions $\rho_{\delta,b}$ are solutions of the system of ODEs $d/dt\rho_{\delta,b}(t) = f_{\delta,b}(\rho)$, where:

$$f_{\delta,b}(\rho) = a^{1-\delta}(\sum_k (p_k)^{\delta+1}) - a\rho_{\delta+1,a}(t).$$

  where $f : \mathcal{P} \to \mathrm{span}(\mathcal{P})$ is a Lipschitz-continuous function.

- f is the drift of the stochastic system: indeed, $P_{\delta,b}(t)$ changes if the requested item has a request time prior to b. If this item is k, then $P_{\delta,b}(t+1) = P_{\delta,b}(t) + a^{1-\delta}(p_k)^\delta$. This shows that

$$\mathbf{E}\left[P_{\delta,b}(t+1) - P_{\delta,b}(t) \mid \mathcal{F}_t\right] = \sum_{k=1}^n p_k a^{1-\delta}(p_k)^\delta \mathbf{1}_{\{\tau_k(t)<b\}} = f_{\delta,b}(P(t)),$$

  where $(\mathcal{F}_t)$ denotes the natural filtration associated to the stochastic process P.

- The variance of $P(t)$ is bounded:

$$\begin{aligned}
\mathbf{E}\left[\|P(t+1) - P(t)\|_\infty^2 \,|\mathcal{P}\right] &= \mathbf{E}\left[\sup_{\delta,b} |P_{\delta,b}(t+1) - P_{\delta,b}(t)|^2 \,|\mathcal{F}_t\right] \\
&= \mathbf{E}\left[|P_{0,t}(t+1) - P_{0,t}(t)|^2 \,|\mathcal{F}_t\right] \\
&= a^2.
\end{aligned}$$

By using Lemma 1, this implies that for each $T > 0$, there exists a constant C such that $\mathbf{E}\left[\sup_{t\leq T/a} \|P(t) - \rho(t)\|_\infty^2\right] \leq Ca^2$. Lemma 2, whose is given in Appendix B, concludes the proof for LRU.

Lemma 2. *Let $g_m : \mathcal{P} \to [0,1]$ be the function defined by $g_m(\rho) = \rho_{1,\theta}$, where $\theta = \sup\{b : \rho_{0,b} \geq m\}$. The function $g_m(\rho)$ is Lipschitz-continuous on $\mathcal{P}$ with the constant 2.*

Note that Equation (15) can be transformed into a PDE by considering the change of variable $y_{k,s}(t) = x_{k,t-s}(t)$. The quantity $y_{k,s}(t)$ is an approximation of the probability for an item k to have been requested between $t-s$ and t. The set of ordinary differential Equations (15) can then be naturally transformed in the following PDE:

$$\frac{\partial}{\partial t} y_{k,s}(t) = p_k(1 - y_{k,s}(t)) - \frac{\partial}{\partial s} y_{k,s}(t). \tag{16}$$

The fixed point y of the PDE can be obtained by solving the equation $\frac{\partial}{\partial t} y = 0$. This fixed point satisfies $y_{k,s} = 1 - e^{-p_k s}$. For this fixed point, the quantity $T = t - \theta$ satisfies $m = \sum_{k=1}^n (1 - e^{-p_k T})$. This equation is the same as the TTL approximation, given by Equation (1).

4.2.2. h-LRU

The construction for LRU can be extended to the case of h-LRU by adding to each item h variables $L_{k,\ell}(t) \in \{\text{true}, \text{false}\}$. For item k and a list ℓ, $L_{k,\ell}(t)$ equals true if item k was present in list ℓ just after the last request[2] of item k and false otherwise. Similarly to the case of LRU, we define the quantity $\Theta_\ell(t)$ to be the request time of the least recently requested item that belongs to list ℓ at time t, that is,

$$\Theta_\ell(t) = \sup\{b : \sum_{k=1}^{n} \mathbf{1}_{\{\tau_k(t) \geq b \wedge L_{k,\ell}(t)\}} \geq m\}.$$

We then define $x_{k,\ell,b}(t)$ that is an approximation of the probability for item k to have $\tau_k(t) \geq b$ and $L_\ell(t) = \text{true}$.

As $L_1(t)$ is always equal to true, the ODE approximation for $x_{k,1,b}(t)$ is the same as (15). Moreover, this implies that $\Theta_1(t) \geq \Theta_\ell(t)$ for $\ell \geq 2$. For the list $\ell = 2$, the approximation is obtained by considering the evolution of $L_2(t)$. After a request, $L_2(t+1)$ is true if $\tau_k(t) \geq \Theta_1(t)$ or if ($\tau_k(t) \geq \Theta_2(t)$ and $L_2(t) = \text{true}$). Both these events occur if ($\tau_k(t) \geq \Theta_1(t)$ and $L_2(t) = \text{true}$) as $\Theta_1(t) \geq \Theta_2(t)$. This suggests that, if the item k is requested, then, in average $L_{k,2}(t+1)$ is approximately $x_{k,1,\theta_1(t)} + x_{k,2,\theta_2(t)} - x_{k,2,\theta_1(t)}$, which leads to the following ODE approximation for $x_{k,2,b}$:

$$\dot{x}_{k,2,b} = p_k(x_{k,2,\theta_2(t)} + x_{k,1,\theta_1(t)} - x_{k,2,\theta_1(t)} - x_{k,2,b}), \tag{17}$$

where $\theta_\ell(t) = \sup\{b : \sum_{k=1}^{n} x_{k,\ell,b}(t) \geq m\}$ for $\ell \in \{1, 2\}$.

The formulation for the third list and above is more complex. In Section 3.2, we showed that the computation of the fixed point is simple because the quantities T_ℓ of the fixed point satisfy $T_1 \leq T_2 \cdots \leq T_h$. However, for the stochastic system, we do not necessarily have[3] $\Theta_\ell(t) \geq \Theta_{\ell+1}(t)$ when $\ell \geq 2$, which implies that the ODE approximation for h-LRU has 2^{h-1} terms.

Applying the reasoning of $L_{k,2}$ to compute $L_{k,\ell}$ ($\ell \geq 3$) involves computing the probability of ($\tau_k(t) \geq \Theta_{\ell-1}(t)$ and $L_{k,\ell-1}(t) = \text{true}$) or ($\tau_k(t) \geq \Theta_\ell(t)$ and $L_{k,\ell}(t) = \text{true}$). When $\Theta_\ell(t) \leq \Theta_{\ell-1}(t)$, both these events occur if ($\tau_k(t) \geq \Theta_{\ell-1}(t)$ and $L_{k,\ell}(t) = L_{k,\ell-1}(t) = \text{true}$). This suggests that the ODE for $x_{k,\ell,b}(t)$ has to involve a term $x_{k,\{\ell-1,\ell\},\theta_{\ell-1}(t)}(t)$, that is an approximation for the item k to have a request time after $\theta_{\ell-1}(t)$ and such that $L_{k,\ell-1}(t) = L_{k,\ell}(t) = \text{true}$. Note, for $\ell = 2$ we have $x_{k,\{\ell-1,\ell\},b}(t) = x_{k,\ell,b}(t)$ as $L_{k,1}(t)$ is always true, but this does not hold for $\ell > 2$. This leads to:

$$\dot{x}_{k,\ell,b} = p_k(x_{k,\ell,\theta_\ell(t)} + x_{k,\ell-1,\theta_{\ell-1}(t)} - x_{k,\{\ell-1,\ell\},\max\{\theta_{\ell-1}(t),\theta_\ell(t)\}} - x_{k,\ell,b}), \tag{18}$$

A similar reasoning can be applied to obtain an ODE for $x_{k,\{\ell-1,\ell\},b}(t)$ as a function of $x_{k,\{\ell-1,\ell\},b}(t)$, $x_{k,\{\ell-2,\ell-1,\ell\},b}(t)$ and $x_{k,\{\ell-2,\ell\},b}(t)$. For example, for $\ell = 3$ the changes of $x_{k,\{2,3\},b}(t)$ are caused by items that were only in lists 2 or in list 3 and that are now in both lists $\{2, 3\}$, or by items that leave list $\{2, 3\}$. Hence, for $\{2, 3\}$, Equation (17) becomes

$$\dot{x}_{k,\{2,3\},b}(t) = x_{k,2,\theta_2(t)} + x_{k,3,\theta_1(t)} - x_{k,\{2,3\},\theta_1(t)} - x_{k,\{2,3\},b}$$

as $L_{k,1}(t)$ is always true.

The hit probability of list ℓ used in Theorem 1 is then

$$h_\ell(t) = \sum_{k=1}^{n} x_{k,\ell,\theta_\ell(t)}(t), \tag{19}$$

where the variables $x_{k,\ell,b}$ satisfy the above ODE.

[2]Note that, after a request, an item is always inserted in list 1. This implies $L_{k,1}(t) = \text{true}$.

[3]When $h = 3$ lists, the variables $\Theta_\ell(t)$ are not always ordered. For example, consider the case of four items $\{1, 2, 3, 4\}$ and $m_1 = m_2 = m_3 = 3$. If initially the three caches contain the three items $1, 2, 3$. Then, after a stream of requests: $4, 4, 3, 2, 1$, the cache 1 and 3 will contain the items $\{1, 2, 3\}$ while the cache 2 will contain $\{1, 2, 4\}$. This implies that $t - 3 = \Theta_2(t) < \Theta_3(t) = \Theta_1(t) = t - 2$.

The proof that the ODE (18) describes well the transient behavior of h-LRU is almost identical to the corresponding proof for LRU. For example, if we focus on the case of 2-LRU[4], the main idea would be to define the quantities $\rho_{\delta,\ell,b}(t)$ and $P_{\delta,\ell,b}(t)$ (for $\ell \in \{1,2\}$):

$$\rho_{\delta,\ell,b}(t) = a^{1-\delta}\sum_{k=1}^{n}(p_k)^{\delta}x_{k,\ell,b}(t); \text{ and } P_{\delta,\ell,b}(t) = a^{1-\delta}\sum_{k=1}^{n}(p_k)^{\delta}\mathbf{1}_{\{\tau_k(t)\geq b\wedge L_{k,\ell}(t)\}}.$$

Equation (17) implies that

$$\dot{\rho}_{\delta,2,b} = a(\rho_{\delta+1,2,\theta_2(t)} + \rho_{\delta+1,1,\theta_1(t)} - \rho_{\delta+1,2,\theta_1(t)} - \rho_{\delta+1,2,b}). \tag{20}$$

Lemma 2 implies that the quantity $g_{m,\ell}(\rho) = \rho_{1,\ell,\theta}$, where θ is such that $\rho_{0,\ell,\theta} = m$, is a Lipschitz function of ρ with constant 2. It follows that the right-side of the ODE Equation (20) is Lipschitz-continuous with constant $4a$. As for LRU, the right side of Equation (20) is the average variation of $P_{\delta,2,b}$ and that the second moment of the variation is bounded by a. Lemma 1 concludes the proof for 2-LRU.

As for LRU, we can transform (17) into a PDE by using the change of variables $y_{k,\ell,s}(t) = x_{k,\ell,t-s}(t)$ and $T_\ell(t) = t - \theta_\ell(t)$. For example, for $\ell = 2$, the fixed point y of this PDE satisfies

$$0 = p_k(y_{k,2,T_2} + y_{k,1,T_1} - y_{k,2,T_1} - y_{k,2,s}) - \frac{\partial}{\partial s}y_{k,2,s}.$$

The solution of this ODE in s is given by

$$y_{k,2,s} = (y_{k,2,T_2} - y_{k,2,T_1} + y_{k,1,T_1})(1 - e^{-p_k s}) \tag{21}$$

$$= \frac{y_{k,1,T_1}}{1 + e^{-p_k T_2} - e^{p_k T_1}}(1 - e^{-p_k s}), \tag{22}$$

where we use (21) for $s = T_1$ and $s = T_2$ to obtain (22).

In Section 4.2.1, we have shown that $y_{k,1,T_1} = 1 - e^{-p_k T_1}$ where T_1 is such that $\sum_{k=1}^{n} y_{k,1,T_1} = m$. One can verify that replacing $y_{k,1,T_1}$ by $1 - e^{-p_k T_1}$ in Equation (22) with $s = T_2$ leads to Equation (13).

4.2.3. LRU(m)

The construction of the approximation and the proof for the case of LRU($\mathbf{m}$) is more involved because of discontinuities in the dynamics. We replace the request time by a quantity that we call a *virtual request time* that is such that the m_h items that have the largest virtual request times are in list h. The next m_{h-1} are in list $h-1$, etc. At time 0, we initialize the virtual request times to be minus the position of the item in the cache. The virtual request time of an item changes when this item is requested. If the item was in list h or $h-1$ prior to the request, its virtual request time becomes $t+1$. If the item was in a list $\ell \in \{0 \dots h-2\}$, its virtual request time becomes the largest virtual request time of the items in list $\ell + 1$.

The approximation of the distribution of virtual request times is given by an ODE on the quantities $x_{k,b}(t)$ that are meant to be an approximation of the probability that the item k has a virtual request time after b:

$$\dot{x}_{k,b}(t) = p_k(x_{k,\theta_{\zeta_b(t)-1}(t)}(t) - x_{k,b}(t)), \tag{23}$$

where $\theta_\ell(t)$ and $\zeta_b(t)$ are defined by:

$$\theta_\ell(t) = \sup\{b : \sum_{k=1}^{n} x_{k,b}(t) \geq m_h + \cdots + m_\ell\} \tag{24}$$

$$\zeta_b(t) = \max\{\ell : \theta_\ell(t) \leq b\} \tag{25}$$

In the above equation, $\theta_\ell(t)$ is an approximation of the highest virtual request time of an object that is in list $\ell - 1$ (at time t) and $\zeta_b(t)$ is the list in which an item with a request time b is (at time t).

[4]For $h \geq 3$, the proof is similar but one would need to also consider quantities like $\rho_{\delta,\{2,3\},b}(t) = a^{1-\delta}\sum_{k=1}^{n}(p_k)^{\delta}x_{k,\{2,3\},b}(t)$.

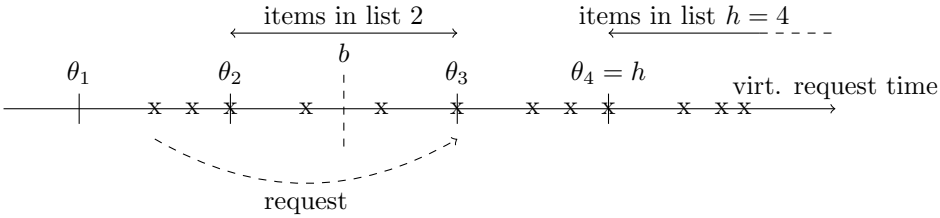


Figure 4: The evolution of virtual request times. Each "x" corresponds to the virtual request time of an object. We consider a LRU(m) cache with 4 lists. The objects that have a virtual request time betwee, θ_2 and θ_3 are in list 2. If a request item has a virtual request time between θ_1 and b, then its virtual request time will be higher than b at the next time step.

The intuition behind Equation (23) is as follows. The quantity $x_{k,b}(t)$ is meant to be an approximation of the probability that item k has a virtual request time after b. Hence, this probability evolves because there is a probability that object k had a virtual request time prior to b and that now has a virtual request time b or after. This occurs if item k had a virtual request time between $\theta_{\zeta_b(t)-1}(t)$ and b and was requested (in which case its new virtual request time is $\theta_{\zeta_b(t)+1}(t) \geq b$). Otherwise, if the item k had a virtual request time prior to $\theta_{\zeta_b(t)-1}(t)$, then upon request it jumps to a list $\ell < \zeta_b(t) - 1$ and therefore will keep a virtual request time prior to b. Figure 4 illustrate how virtual request times evolve.

The hit ratio for LRU($\mathbf{m}$) used in Theorem 1 is given by

$$h_\ell(t) = \sum_{k=1}^{n} p_k(x_{k,\theta_\ell(t)}(t) - x_{k,\theta_{\ell+1}(t)}(t)) \tag{26}$$

The main difference between the proof for LRU($\mathbf{m}$) compared to the one of h-LRU is that the right-side of the differential equation (23) is not Lipschitz-continuous in ρ because the list in which an item that has a virtual request time b belongs to depends non-continuously on ρ (the list ζ_b is a discrete quantity). Our method to overcome this difficulty is prove that the drift is partially one-sided Lipschitz-continuous functions (in a sense that will me made precise in Lemma 3).

As before, let $P_{\delta,b}(t) = a^{1-\delta} \sum_{k=1}^{n} (p_k)^\delta \mathbf{1}_{\{\tau_k(t) \geq b\}}$, where $a = \max_{k=1}^{n} p_k$. We also define $f : \mathcal{P} \to \mathrm{span}(\mathcal{P})$ by $f_{\delta,b}(\rho) = a(\rho_{\delta+1,\theta_{\zeta_b}-1} - \rho_{\delta+1,b})$, where θ_ℓ and ζ_b are two functions of ρ that are defined by

$$\rho_{0,\theta_\ell} = m_\ell + \cdots + m_h \text{ and } \theta_{\zeta_b} \leq b < \theta_{\zeta_b+1}.$$

As for the the cases of LRU and h-LRU, one can verify that $f_{\delta,b}$ is the average variation of $P_{\delta,b}(t)$ during one time step and that the second moment of the average variation is bounded by a^2. Moreover, if x is a solution of the differential equation (23), then $\rho_{\delta,b}(t) = \sum_{k=1}^{n} x_{k,b}(t)$ is a solution of the differential equation $\dot{\rho} = f(\rho)$.

The next lemma – whose proof is given in Appendix B.1 – states some key properties of the function f. In particular, *(i)* quantifies what we mean by partially one-sided Lipschitz.

Lemma 3. *For any $\rho, \rho' \in \mathcal{P}$ and $\delta \geq 1$, we have:*

(i) $(\rho_{0,b} - \rho'_{0,b})(f_{0,b}(\rho) - f_{0,b}(\rho')) \leq 2a \left\|\rho - \rho'\right\|_\infty^2$;

(ii) $\left\|f(\rho)\right\|_\infty \leq a$;

(iii) $|f_{\delta,b}(\rho) - f_{\delta,b}(\rho') - (f_{0,b}(\rho) - f_{0,b}(\rho'))| \leq 5a \left\|\rho - \rho'\right\|_\infty$.

Let $V(t) \in \mathcal{P}$ be the vector defined by $V_{\delta,b}(t) = P_{\delta,b}(t+1) - P_{\delta,b}(t) - f_{\delta,b}(P(t))$. By using the definition

$P_{\delta,b}(b) = \rho_{\delta,b}(b)$, the fact that $\rho_{\delta,b}(t) = \rho_{\delta,b}(b) + \int_b^t f_{\delta,b}(\rho(s))ds$, and Lemma 3(iii), we have

$$|P_{\delta,b}(t) - \rho_{\delta,b}(t)| = \left| \int_{s=b}^{t} (f_{\delta,b}(P(\lfloor s \rfloor)) - f_{\delta,b}(\rho(s)))ds + \sum_{s=b}^{t-1} V_{\delta,b}(s) \right| \tag{27}$$

$$\leq \left| \int_{s=b}^{t} f_{0,b}(P(\lfloor s \rfloor)) - f_{0,b}(\rho(s))ds \right| + 5a \int_{s=b}^{t} \|P(\lfloor s \rfloor) - \rho(s)\|_\infty \, ds + \sum_{s=b}^{t-1} \|V(s)\|_\infty$$

$$\leq |P_{0,b}(t) - \rho_{0,b}(s)| + 5a \int_{s=b}^{t} \|P(\lfloor s \rfloor) - \rho(s)\|_\infty \, ds + 2 \sum_{s=b}^{t-1} \|V(s)\|_\infty \,, \tag{28}$$

where the last line comes from the reverse triangle inequality applied to Equation (27) with $\delta = 0$. As at most one item change list at each time-slot, we have $\|V(s)\|_\infty \leq a$. Moreover, by using Gronwall's Lemma, Equation (28) implies that

$$\begin{aligned} \|P(t) - \rho(t)\|_\infty &\leq |P_{0,b}(t) - \rho_{0,b}(s)| + 5a \int_{s=b}^{t} \|P(\lfloor s \rfloor) - \rho(s)\|_\infty \, ds + 2at \\ &\leq (|P_{0,b}(t) - \rho_{0,b}(s)| + 2at)e^{5at}. \end{aligned} \tag{29}$$

In order to bound the previous equation, we will use Lemma 3(i) to bound $|P_{0,b}(t) - \rho_{0,b}(t)|$. As ρ is solution of the differential equation $\dot{\rho} = f(\rho)$, we have $\rho(t+1) = \rho(t) + \int_0^1 f(\rho(t+s))ds$. This implies

$$\begin{aligned} (P_{0,b}(t+1) - \rho_{0,b}(t+1))^2 &= (P_{0,b}(t) - \rho_{0,b}(t) + V_{0,b}(t) + f_{0,b}(P(t)) + \int_0^1 f_{0,b}(\rho(t+s))ds)^2 \\ &= (P_{0,b}(t) - \rho_{0,b}(t))^2 + \left[V_{0,b}(t) + f_{0,b}(P(t)) + \int_0^1 f_{0,b}(\rho(t+s))ds \right]^2 \\ &\quad + 2\big(P_{0,b}(t) - \rho_{0,b}(t)\big)V_{0,b}(t) \\ &\quad + 2\Big(P_{0,b}(t) - \rho_{0,b}(t)\Big)\Big(f_{0,b}(P(t)) + \int_0^1 f_{0,b}(\rho(t+s))ds\Big) \end{aligned} \tag{30}$$

As at most one object changes list at each time step, we have $\mathbf{E}\left[\|V(t)\|_\infty^2 \mid \mathcal{F}_t\right] \leq \max_k(p_k)^2 = a^2$. This, pus the fact that $f_{0,b}(\cdot) \leq a$, implies that the expectation of the second term is smaller than $9a^2$. Moreover, $f(P)$ is the average variation of P, and therefore $\mathbf{E}\left[V(t) \mid \mathcal{F}_t\right] = 0$ which implies that the expectation of the third term is equal to 0. Moreover, The last term equals

$$\begin{aligned} &2 \int_0^1 \Big(P_{0,b}(t) - \rho_{0,b}(t)\Big)\Big(f_{0,b}(P(t)) + f_{0,b}(\rho(t+s))\Big)ds \\ &\quad = 2 \int_0^1 \Big(P_{0,b}(t) - \rho_{0,b}(t+s)\Big)\Big(f_{0,b}(P(t)) + f_{0,b}(\rho(t+s))\Big)ds \\ &\qquad + 2 \int_0^1 \Big(\rho_{0,b}(t+s) - \rho_{0,b}(t)\Big)\Big(f_{0,b}(P(t)) + f_{0,b}(\rho(t+s))\Big)ds \qquad (31) \\ &\quad \leq 4a \int_0^1 \|P(t) - \rho(t+s)\|_\infty^2 \, ds + 2a^2, \qquad (32) \\ &\quad \leq 4a \, \|P(t) - \rho(t)\|_\infty^2 + 4a^2, \qquad (33) \end{aligned}$$

where we use Lemma 3(i) to bound the first term of the Equation (31) and Lemma 3(ii) to bound its second term. We again used Lemma 3(ii) to bound (32).

Combining Equation (30) and (33) shows that

$$\mathbf{E}\left[|P_{0,b}(t) - \rho_{0,b}(t)|^2\right] \leq 2a \sum_{s=b}^{t} \mathbf{E}\left[\|P(s) - \rho(s)\|_\infty^2\right] + 13a^2 t \tag{34}$$

Equation (29) implies that

$$\|P(t)-\rho(t)\|_\infty^2 \leq (2\left|P_{0,b}(t)-\rho_{0,b}(s)\right|^2 + 4a^2t^2)e^{10at}.$$

We can then plug the above inequality into Equation (34) to show that

$$\begin{aligned}\mathbf{E}\left[\left|P_{0,b}(t)-\rho_{0,b}(t)\right|^2\right] &\leq 2a\sum_{s=0}^{t}\left(2\mathbf{E}\left[\left|P_{0,b}(s)-\rho_{0,b}(s)\right|^2\right]+4a^2t^2\right)e^{10at}+13a^2t\\ &= 4a\sum_{s=0}^{t}\mathbf{E}\left[\left|P_{0,b}(s)-\rho_{0,b}(s)\right|^2\right]e^{10at}+a(8a^2t^2e^{10at}+13at)\end{aligned}$$

By using the discrete Gronwall's inequality, for all T, there exists a constant $C=(8T^2e^{10T}+13T)e^{2e^{10T}/3}$ such that this is less than Ca when t is less than T/a. Lemma 2 concludes the result.

5. Comparison of LRU, LRU(m) and h-LRU

In this section we start by presenting an insensitivity result for LRU, next we compare the performance of LRU, LRU($\mathbf{m}$) and h-LRU in terms of the achieved hit probability when subject to IRM, renewal, MAP requests and trace-based simulation. A good replacement algorithm should keep popular items in the cache, but needs to be sufficiently responsive to changes in the popularity. As LRU($\mathbf{m}$) and h-LRU are clearly better suited to keep popular items in the cache than LRU, they perform better under static workloads (IRM). We demonstrate that they often also outperform LRU when the workload is dynamic.

5.1. LRU insensitivity

The theorem presented in this subsection complements the results of Jelenkovic and Radovanovic who showed in [13, 12] that for dependent request processes, the hit probability is asymptotically, for large cache sizes, the same as in the corresponding LRU system with i.i.d. requests. Our insensitivity result is valid not just asymptotically, but requires the request processes of the various items to be independent.

Theorem 2. *Assume that the items' request processes are stationary, independent of each other and that the expected number of requests per unit time is positive and finite. Then, the hit probability of LRU only depends on the inter-arrival time distribution. In particular, it does not depend on the correlation between inter-arrival times.*

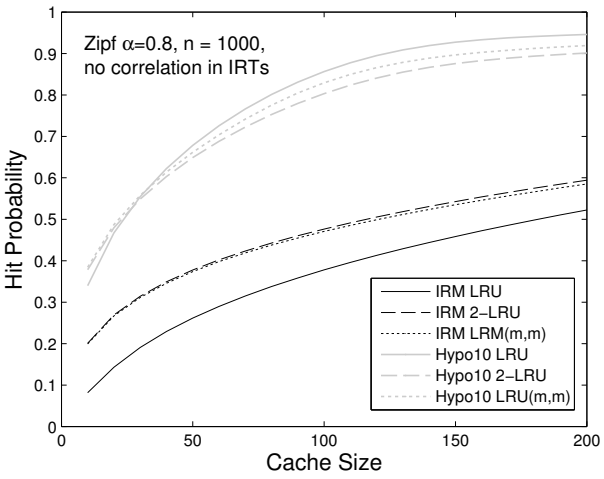
Proof of Theorem 2. For each k, the requests of k are generated according to a stationary point process R_k. For $t<s$, $R_k[t,s)$ is the number of requests of item k during a time interval $[t,s]$. Let $\vartheta_k(t)$ be the time elapsed since the last request of item k. Without loss of generality, in the rest of the proof, we assume that the request process is simple (*i.e.* that with probability 1, the time between two consecutive requests of an item is never 0). If it is not the case, one can suppress any of the two request and obtain the same behavior of the LRU cache. Hence, the process (R_k,ϑ_k) is a stationary marked point process that satisfies Hypothesis 1.1.1 of [2].

As R is stationary, the probability that the item k is requested during a time interval $[t,t+x]$ does not depend on t. Let $\tilde{F}_k(x)$ denote this quantity. We have:

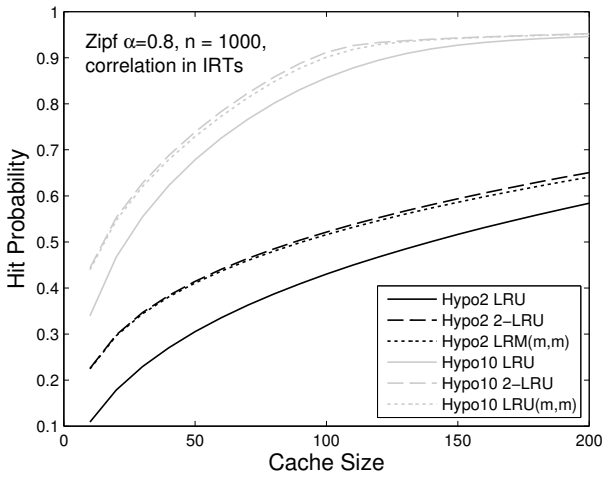
$$\tilde{F}_k(x)=\mathbf{P}\left[R_k[t,t+x]\geq 1\right]=\mathbf{P}\left[R_k[0,x]\geq 1\right].$$

We also define $F_k(x)$ that is the probability that the time between two requests from item k is smaller than x. As (R_k,ϑ_k) is a stationary marked point process, this quantity is well defined and can be expressed as

$$\begin{aligned}F_k(x) &= \mathbf{P}\left[R_k[t,t+x]\geq 1 \mid \text{ a request occurred a time } t\right]\\ &= \mathbf{P}\left[R_k[0,x]\geq 1 \mid \text{ a request occurred a time } 0\right]\end{aligned}$$



(a) IRM model or hyperexponential inter-request times (with $z = 10$).



(b) MAP arrivals (with $z = 2, 10$ and $q = 1/20$).

Figure 5: Hit probability as a function of the cache size for LRU, LRU(m, m) and 2-LRU

Note that the definition of $F_k(x)$ only requires the process R_k to be stationary. When the process is a renewal process, $F_k(x)$ is the cumulative distribution function of the inter-request time.

By the inversion formula [2, Section 1.2.4], $\tilde{F}_k$ is a function of F_k:

$$\tilde{F}_k(x) = \lambda_k \int_0^x (1 - F_k(t))dt, \tag{35}$$

where $\lambda_k = 1/\int_0^\infty (1 - F_k(t))dt$ is the request rate of item k. This quantity only depends on F_k and not on the correlation between two arrivals.

To conclude the proof, we remark that the probability that an item k is in the cache when it is requested can be expressed in terms of the functions F_k and $\tilde{F}_\ell$ for $\ell \neq k$. Indeed, Let $\mathfrak{S}_{n,-k}$ be the set of permutation of $\{1 \dots k-1\} \bigcup \{k+1 \dots n\}$ (*i.e.* all integers between 1 and n except k). An item is in the cache at time t if it is among the m items that were last requested. Hence, the probability for item k to be in the cache at time t is

$$\sum_{\sigma \in \mathfrak{S}_{n,-k}} \mathbf{P}\left[\vartheta_k(t) \leq \vartheta_{\sigma(m)}(t), \vartheta_{\sigma(1)}(t) \leq \cdots \leq \vartheta_{\sigma(n-1)}(t)\right].$$

This event conditioned on the fact that item k is requested at time t is the probability that item k is in the cache when requested. Hence, the hit rate is:

$$\sum_k \lambda_k \sum_{\sigma \in \mathfrak{S}_{n,-k}} \mathbf{P}\left(\begin{matrix}\vartheta_k(t) \leq \vartheta_{\sigma(m)}(t), \\ \vartheta_{\sigma(1)}(t) \leq \cdots \leq \vartheta_{\sigma(n-1)}(t)\end{matrix} \middle| \begin{matrix}\text{item } k \text{ is} \\ \text{requested at } t\end{matrix}\right).$$

This quantity can clearly be expressed as a function of the F_k and $\tilde{F}_k$ which by Equation (35) can be expressed solely as a function of the F_k. □

5.2. Synthetic (static) workloads

For the synthetic workloads we restrict ourselves to LRU, 2-LRU and LRU(m, m). The latter two algorithms both use a cache of size m and additionally keep track of meta-data only for the m items in list 1.

Figure 5a depicts the hit probability as a function of the cache size when $n = 1000$, items follow a Zipf-like popularity distribution with parameter $\alpha = 0.8$ under IRM and renewal requests (with $z = 10$, see

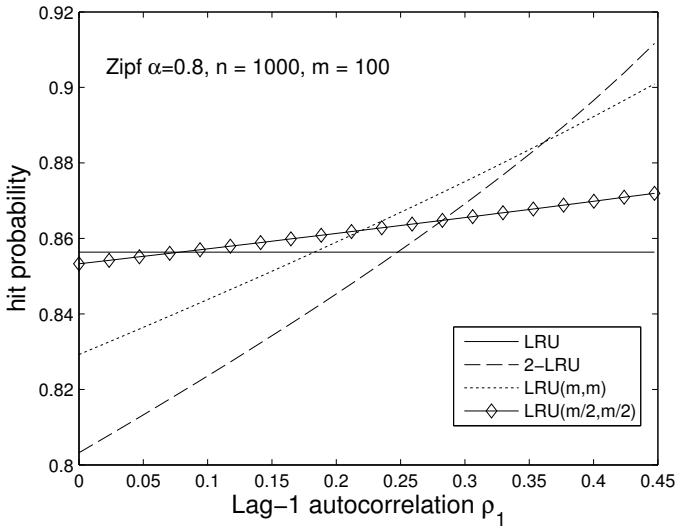


Figure 6: Hit probability as a function of the lag-1 autocorrelation ρ_1 for LRU, LRU(m, m), LRU$(m/2, m/2)$ and 2-LRU when subject to MAP arrivals (with $z = 10$).

Section 4.1.2). Figure 5b shows the impact of having correlation between consecutive inter-request times (that is, $q = 1/20$ instead of $q = 1$ for $z = 2, 10$).

One of the main observations is that LRU(m, m) performs very similar to 2-LRU under IRM, renewal and MAP requests. In fact, 2-LRU performs slightly better, unless the workload is very dynamic ($z = 10$ and $q = 1$ case). Another conclusion that can be drawn from comparing Figures 5a and 5b is that the hit rate of both 2-LRU and LRU(m, m) significantly improves in the presence of correlation between consecutive inter-request times (that is, when $q < 1$), while LRU does not. Recall that LRU$(\mathbf{m})$ needs to update at most one list per hit, as opposed to h-LRU. Thus, whenever both algorithms perform alike, LRU$(\mathbf{m})$ may be more attractive to use.

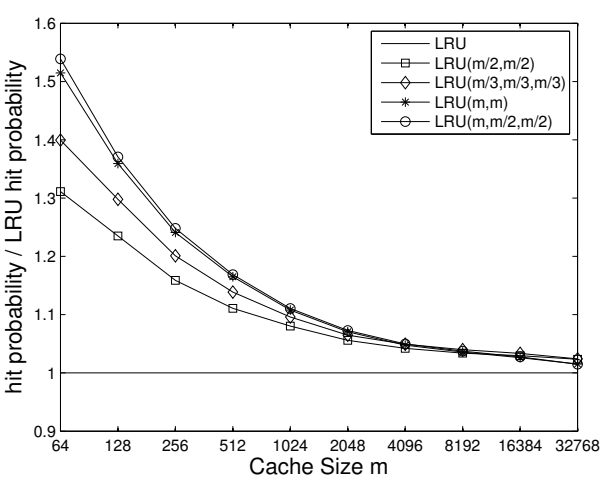
Figure 6 shows that the hit rate of 2-LRU and LRU(m, m) both increase with increasing lag-1 autocorrelation and confirms that the hit probability of LRU is completely *insensitive* to any correlation between consecutive inter-request times (as proven by Theorem 2). Figure 6 further indicates that the hit probability also increases with ρ_1 when splitting the cache in two lists of equal size (although the gain is less pronounced).

5.3. Trace-based simulation

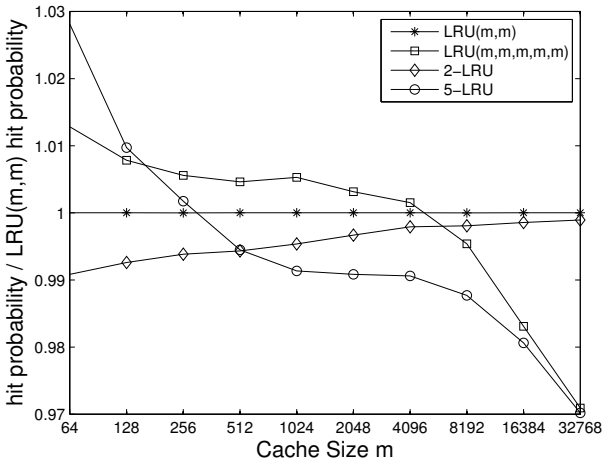
To perform the trace-based simulations we rely on the same 4 IR cache traces as in [4, Section 4]. In this section, we only report the result for the trace collected on Monday 18th Feb 2013. We also simulated the other traces and obtained very similar results.

The hit probability of LRU$(\mathbf{m})$ with a split cache and/or virtual lists normalized by the LRU hit probability is depicted in Figure 7a as a function of the cache size m. It indicates that LRU$(\mathbf{m})$ is more effective than LRU, especially when the cache is small. For small caches using a virtual list is better than splitting the cache and using both a virtual list and split cache offers only a slight additional gain. While not depicted here, we should note that using more virtual lists or splitting the cache in more parts sometimes results in a hit probability below the LRU hit probability for larger caches.

Figure 7b compares h-LRU with LRU$(\mathbf{m})$ using virtual lists, where the hit probability is now normalized by the hit probability of LRU(m, m) to better highlight the differences. We observe that 2-LRU differs by less than 1% from LRU(m, m), while 5-LRU and LRU(m, m, m, m, m) differ by less than 2%. Given that h-LRU may require an update of up to h lists, while LRU$(\mathbf{m})$ requires only one update in case of a hit, LRU$(\mathbf{m})$ seems preferential in this particular case.



(a) LRU(**m**) compared to LRU.



(b) LRU(m, m, m, m, m) and h-LRU compared to LRU(m, m).

Figure 7: Hit probability as a function of the cache size using trace-based simulation.

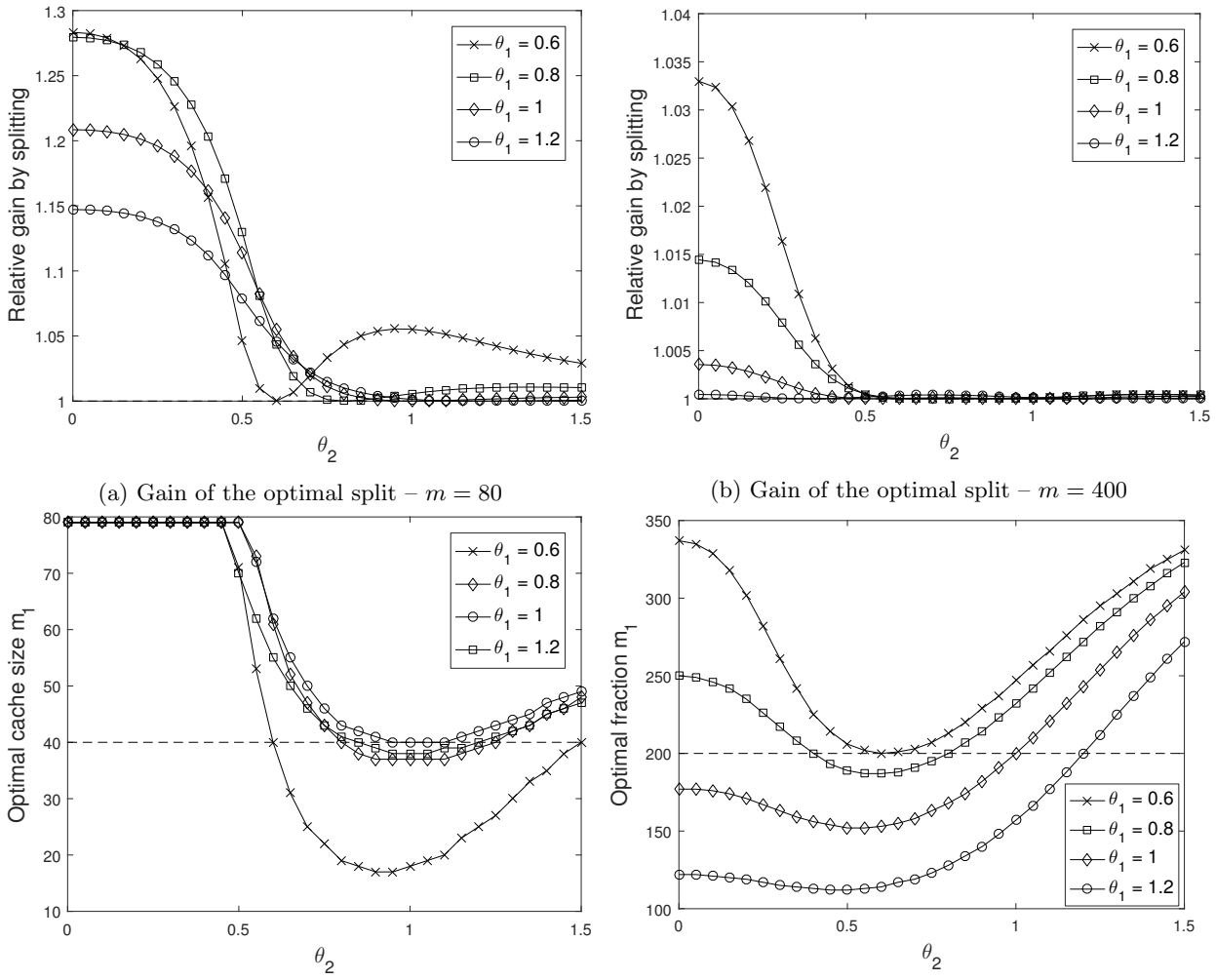
6. Cache partitioning

In this section we consider the cache partitioning scenario introduced in [7]. Consider a cache of size m that is accessed by users for content generated by 2 content providers (CPs). CP k serves a set of n_k items (i.e., files) of equal size that are distinct from the items served by the other CP. Our main objective is to compare the following two setups. In the first setup the cache of size m is shared by both CPs and a single replacement algorithm manages the entire cache. In the second setup the cache provider splits the cache into 2 parts of size m_1 and m_2, such that $m_1, m_2 > 0$ and $m_1 + m_2 = m$. The size m_k part of the cache is dedicated to CP k and therefore only stores the items of CP k. Each part is managed by its own (possibly different) replacement algorithm. The work presented in [7] focused on the LRU replacement algorithm combined with IRM requests. We start by revisiting this case and subsequently consider h-LRU as well as non-IRM request streams.

6.1. IRM combined with LRU

In this subsection we assume that the popularity of the n_k items of CP k follows a Zipf distribution with parameter θ_k. We further assume that the request rate for content of each CP is the same and define the overall hit rate in case of the split cache as the sum of the hit rates in both parts of the cache (i.e., this corresponds to setting $\lambda_1 = \lambda_2$ and $w_1 = w_2 = 1$ in [7]). For the split cache we set the size m_1 of the first part of the cache such that the overall hit rate is optimized. In this case Theorem 2 of [7] shows that sharing the cache is never better than the optimal split cache. In Figure 8 we depict the gain achieved by splitting cache in the optimal manner when $n_1 = n_2 = 1000$ and the cache size m is either 80 or 400. We also plotted the optimal cache size m_1.

A first observation is that the gain decreases as the cache size increases and is negligible for large caches unless the popularity of the content of one of the CPs is close to uniform (this trend was confirmed by considering other values for m). Second, when both popularity distributions have the same shape (i.e., $\theta_1 = \theta_2$) the optimal split is to set $m_1 = m/2$ (as expected). In this case the optimal split cache achieves the same overall hit rate as the shared cache, meaning there is no gain in splitting the cache. Third, although the gain by splitting the cache may be very limited, the optimal size m_1 is quite sensitive to the shape of both distributions. Figure 9 further demonstrates that some care is required when splitting the cache in case the shape of the distribution is not known.



(a) Gain of the optimal split – $m = 80$

(b) Gain of the optimal split – $m = 400$

(c) Optimal split size – $m = 80$

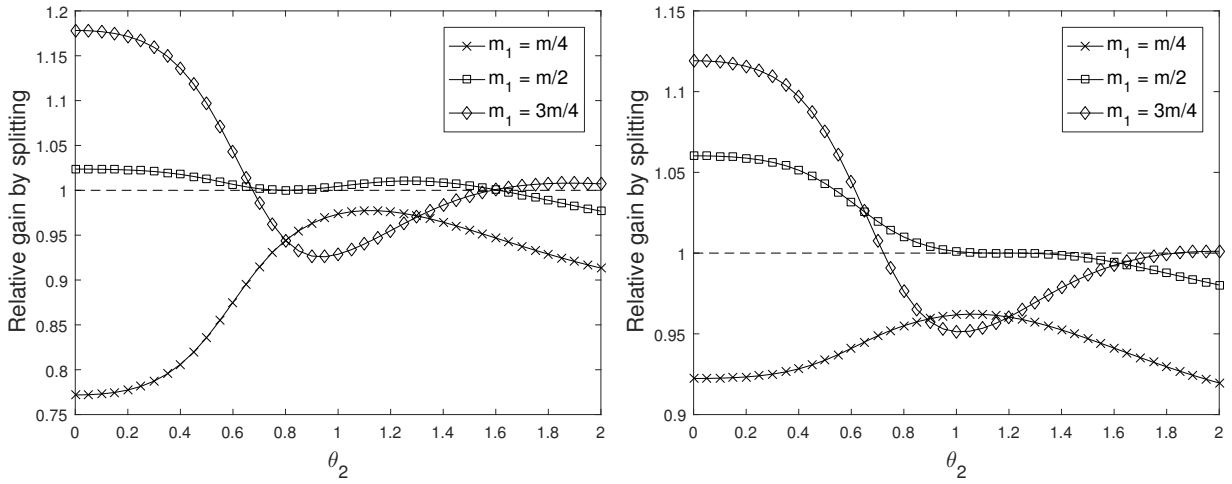
(d) Optimal split size – $m = 400$

Figure 8: Relative hit rate gain by the optimal split cache and optimal cache size dedicated to the first CP under IRM requests with $n_1 = n_2 = 1000$, and Zipf popularity for the LRU replacement algorithm.

6.2. IRM combined with h-LRU

We now consider the same scenario as in the previous subsection, except that we replace LRU by h-LRU. The main questions we wish to answer are: does the optimal split cache still outperform the shared cache and how are the possible gains achieved the optimal split cache affected by the number of lists h. When combining h-LRU with a split cache, we split all of the h lists in two parts such that the first part has size m_1. In other words CP k uses h-LRU where all the lists have size m_k. Allowing different splits in each of the h lists may further improve the hit rate, but is not considered in this section.

Figure 10 plots the relative gains achieved by splitting the cache in the optimal manner when the cache size $m = 80$ and h-LRU is used instead of LRU. We first note that all the relative gains are at least one, meaning the optimal split cache also appears to outperform the shared cache for h-LRU. When comparing Figures 8a, 10a and 10b we further note that the gain achieved by the optimal split cache diminishes as h increases. Thus, when using h-LRU (combined with IRM requests) there is less use in implementing a split cache compared to LRU. This can be understood by noting that under the IRM model increasing h improves the hit probability of the shared cache and therefore it becomes harder to achieve significant gains by splitting the cache.



(a) $\theta_1 = 0.8$ (b) $\theta_1 = 1.2$

Figure 9: Relative hit rate gain of the split cache compared to a shared cache under IRM requests with $n_1 = n_2 = 1000$, $m = 80$ and Zipf popularity for the LRU replacement algorithm for $m_1 = 20, 40$ and 60.

6.3. MAP requests

In this subsection we replace the IRM request process by the MAP process described in Section 4.1.2, where the inter request time distribution follows a hyperexponential distribution with rates z and $1/z$. We only present results for LRU, for 2-LRU similar observations were made.

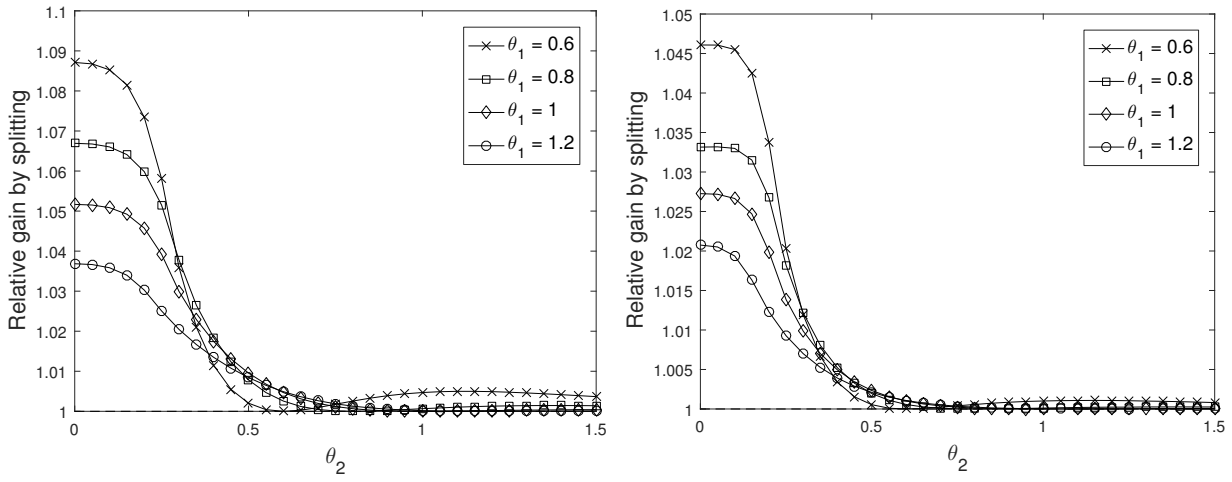
The scenario presented in Figure 11b is identical to Figure 8a except that the exponential inter request times are replaced by a hyperexponential distribution with $z = 10$ (note the value of q is irrelevant due to Theorem 2). We first note that as in the IRM case the optimal split cache achieves a higher hit rate than the shared cache. The relative gain is however much smaller. This can be attributed to the fact that higher hit rates are observed with hyperexponential inter request times, meaning there is less room for improvement by splitting the cache. If we further lower the cache size to $m = 16$ as in Figure 11a we observe larger relative gains. Comparing Figures 8 and 11 shows that the optimal manner in which the cache is split depends heavily on the inter request time distribution as well as on the overall cache size.

7. Conclusion

In this paper, we developed algorithms to approximate the hit probability of the cache replacement policies LRU($\mathbf{m}$) and h-LRU. These algorithms rely on an equivalence between LRU-based and TTL-based cache replacement algorithms. We showed numerically that the TLL approximations are very accurate for moderate cache sizes and appear asymptotically exact as the cache size and number of items tends to infinity. We also provide theoretical support for this claim, by establishing a bound between the transient dynamics of both policies and a set of ODEs whose fixed-point coincides with the proposed TTL approximation.

Using these approximations, we showed that the hit probability of h-LRU and LRU($\mathbf{m}$) are comparable in many scenarios. We also studied how splitting the cache can improve the performance. Our numerical observations confirm that for all the tested parameters, the optimal split cache outperforms a shared cache. However, the gain appears to be limited for large cache sizes and the optimal splitting size is very sensitive to the parameters.

A possible extension of our results would be to study networks of caches in which LRU, LRU($\mathbf{m}$) or h-LRU is used in each node. Further, our TTL approximation with MAP arrivals can be readily adapted to other policies such as FIFO($\mathbf{m}$) and RAND($\mathbf{m}$) introduced in [11]. In fact, a generalization to a network of caches would be fairly straightforward for the class of RAND($\mathbf{m}$) policies.



(a) $h = 2$ (b) $h = 4$

Figure 10: Relative hit rate gain by the optimal split cache compared to a shared cache under IRM requests with $n_1 = n_2 = 1000$, $m = 80$ and Zipf popularity for the h-LRU replacement algorithm.

Acknowledgements

This work is partially supported by the EU project QUANTICOL, 600708.

[1] O. I. Aven, E. G. Coffman, Jr., and Y. A. Kogan. *Stochastic Analysis of Computer Storage*. Kluwer Academic Publishers, Norwell, MA, USA, 1987.

[2] F. Baccelli and P. Brémaud. *Elements of queueing theory: Palm Martingale calculus and stochastic recurrences*, volume 26. Springer Science & Business Media, 2013.

[3] D. S. Berger, P. Gland, S. Singla, and F. Ciucu. Exact analysis of TTL cache networks. *Performance Evaluation*, 79:2–23, 2014.

[4] G. Bianchi, A. Detti, A. Caponi, and N. Blefari-Melazzi. Check before storing: What is the performance price of content integrity verification in LRU caching? *SIGCOMM Comput. Commun. Rev.*, 43(3):59–67, July 2013.

[5] G. Casale. Building accurate workload models using markovian arrival processes. In *ACM SIGMETRICS*, SIGMETRICS '11, pages 357–358, New York, NY, USA, 2011. ACM.

[6] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *IEEE J.Sel. A. Commun.*, 20(7):1305–1314, 2002.

[7] W. Chu, M. Dehghan, D. Towsley, and Z. Zhang. On allocating cache resources to content providers. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 154–159, New York, NY, USA, 2016. ACM.

[8] R. Fagin. Asymptotic miss ratios over independent references. *Journal of Computer and System Sciences*, 14(2):222 – 250, 1977.

[9] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Performance evaluation of hierarchical TTL-based cache networks. *Computer Networks*, 65:212–231, 2014.

[10] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. In *Proceedings of the 24th International Teletraffic Congress*, ITC '12, pages 8:1–8, 2012.

[11] N. Gast and B. Van Houdt. Transient and steady-state regime of a family of list-based cache replacement algorithms. In *Proceedings of ACM SIGMETRICS*. ACM, 2015.

[12] P. Jelenkovic and A. Radovanovic. Asymptotic insensitivity of least-recently-used caching to statistical dependency. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 438–447. IEEE, 2003.

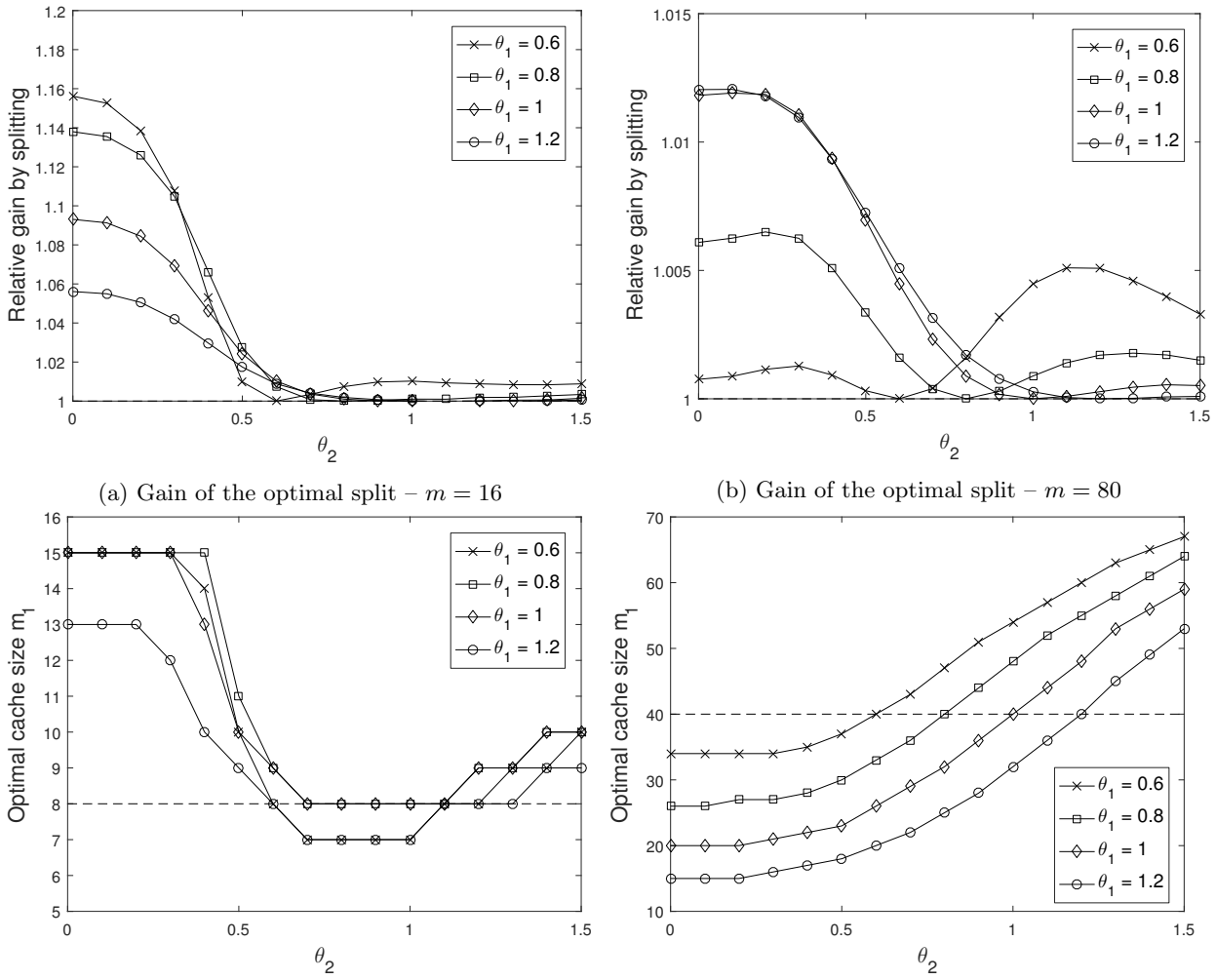
[13] P. R. Jelenković and A. Radovanović. Least-recently-used caching with dependent requests. *Theoretical computer science*, 326(1):293–327, 2004.

[14] B. Jiang, P. Nain, and D. Towsley. LRU cache under stationary requests. *arXiv preprint arXiv:1707.06204*, 2017.

[15] T. Johnson and D. Shasha. 2Q: A low overhead high performance buffer management replacement algorithm. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 439–450, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[16] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods and stochastic modeling*. SIAM, Philadelphia, 1999.

[17] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *INFOCOM 2014*, pages 2040–2048, 2014.



(a) Gain of the optimal split – $m = 16$

(b) Gain of the optimal split – $m = 80$

(c) Optimal split size –y $m = 16$

(d) Optimal split size – $m = 80$

Figure 11: Relative hit rate gain compared to a shared cache and optimal cache size dedicated to the first CP under hyperexponential requests ($z = 10$) with $n_1 = n_2 = 1000$, $m = 80$ and Zipf popularity for the LRU replacement algorithm.

[18] E. J. Rosensweig, J. Kurose, and D. Towsley. Approximate models for general cache networks. In *INFOCOM'10*, pages 1100–1108, Piscataway, NJ, USA, 2010. IEEE Press.

[19] M. Telek and G. Horváth. A minimal representation of markov arrival processes and a moments matching method. *Perform. Eval.*, 64(9-12):1153–1168, Oct. 2007.

Appendix A. h-LRU with renewal arrivals

The same approach as for the IRM model can be used to obtain a TTL approximation when the requests for item k follow a renewal process, characterized by a distribution with cumulative distribution function $F_k(x)$. Let $\bar{F}_k(x) = 1 - F_k(x)$. In this case we get $(\bar{P}_{h,k})_{j,0} = \bar{F}_k(T_{\min(h,j+1)})$ and $(\bar{P}_{h,k})_{j,\min(h,j+1)} = F_k(T_{\min(h,j+1)})$. The hit probability for item k can therefore be expressed as

$$\bar{\pi}_h^{(h,k)} = \frac{\prod_{s=1}^{h} F_k(T_s)}{\prod_{s=1}^{h} F_k(T_s) + \bar{F}_k(T_h)\left(1 + \sum_{j=1}^{h-1}\prod_{s=1}^{j} F_k(T_s)\right)},$$

while for $j = 1, \dots, h-1$ we have

$$\bar{\pi}_j^{(h,k)} = \frac{\bar{F}_k(T_h) \prod_{s=1}^{j} F_k(T_s)}{\prod_{s=1}^{h} F_k(T_s) + \bar{F}_k(T_h) \left(1 + \sum_{j=1}^{h-1} \prod_{s=1}^{j} F_k(T_s)\right)}.$$

The fixed point equation for determining T_h is found as

$$m = \sum_{k=1}^{n} \frac{(\bar{\pi}_{h-1}^{(h,k)} + \bar{\pi}_h^{(h,k)}) \int_{x=0}^{T_h} x dF_k(x)}{\int_{x=0}^{\infty} \bar{F}_k(x) dx} = \sum_{k=1}^{n} \frac{(\bar{\pi}_{h-1}^{(h,k)} + \bar{\pi}_h^{(h,k)}) \left(T_h - \int_{x=0}^{T_h} F_k(x) dx\right)}{\int_{x=0}^{\infty} \bar{F}_k(x) dx},$$

as $(\bar{\pi}_{h-1}^{(h,k)} + \bar{\pi}_h^{(h,k)}) \int_{x=0}^{T_h} x dF_k(x)$ is the mean time that item k spends in the cache between two requests for item k and $\int_{x=0}^{\infty} \bar{F}_k(x) dx$ is simply the mean time between two requests.

Appendix B. Proofs of the lemmas used in the proof of Theorem 1

Proof of Lemma 1. The only difficulty of Lemma 1 is that the variables $\{P_{\delta,b}\}_{\delta,b}$ live in a set of infinite dimension $\mathcal{P}$:

$$\mathcal{P} = \Big\{ (P_{\delta,b})_{\delta,b} : \exists (x_{k,b}) \text{ non-increasing in } b, \text{ bounded by } 1 \\ \text{such that for all } \delta, b\text{: } P_{\delta,b} = a^{1-\delta} \sum_{k=1}^{n} (p_k)^{\delta} x_{k,b} \Big\}.$$

We equip $\mathcal{P}$ with the L_∞ norm and denote $\|\rho\|_\infty = \sup_{\delta,b} |\rho_{\delta,b}|$ the norm of a vector $\rho \in \mathcal{P}$.

The solution of the ODE $\dot{x} = f(x)$ that starts in $X(0)$ satisfies $x(t) = X(0) + \int_{s=0}^{t} f(x(s)) ds$. Let $E(t)$ be such that

$$\begin{aligned} X(t) &= X(0) + \sum_{s=0}^{t-1} f(X(s)) + E(t) \\ &= X(0) + \int_0^{t-1} f(X(\lfloor s \rfloor)) ds + E(t). \end{aligned}$$

We have:

$$\begin{aligned} \|X(t) - x(t)\|_\infty &\le \int_{s=0}^{t} \|f(X(\lfloor s \rfloor)) - f(x(s))\|_\infty + \|E(t)\|_\infty \\ &\le aL \int_{s=0}^{t} \|X(\lfloor s \rfloor) - x(s)\|_\infty + \|E(t)\|_\infty , \end{aligned}$$

where we used that f is Lipschitz-continuous of constant La.

Let $\bar{X}(t)$ be the the piecewise-linear interpolation of X such that $\bar{X}(t) = X(t)$ when $t \in \mathbf{Z}^+$. We have:

$$\begin{aligned} \|X(\lfloor s \rfloor) - x(s)\|_\infty &\le \left\|X(\lfloor s \rfloor) - \bar{X}(s)\right\|_\infty + \left\|\bar{X}(s) - x(s)\right\|_\infty \\ &\le a + \left\|\bar{X}(s) - x(s)\right\|_\infty , \end{aligned}$$

where we used that $\|f(x)\|_\infty \le a$.

This shows that for any $t \le T/a$ (with $t \in \mathbf{Z}^+$):

$$\begin{aligned} \left\|\bar{X}(t) - x(t)\right\|_\infty &\le aL \int_{s=0}^{t} \left\|\bar{X}(s) - x(s)\right\|_\infty + a^2 Lt + \|E(t)\|_\infty \\ &\le \exp(aLt)(a^2 Lt + \sup_{s \le t} \|E(s)\|_\infty) \\ &\le \exp(LT)(aLT + \sup_{t \le T/a} \|E(t)\|_\infty), \end{aligned}$$

using Gronwall's inequality.
By assumption,

$$\mathbf{E}\left[\|E(t+1)-E(t)\|_\infty^2 \mid \mathcal{F}_t\right] = \operatorname{var}\left[\|X(t+1)-X(t) \mid \mathcal{F}_t\|_\infty\right] \le a^2.$$

As $E(t)$ is a martingale, this implies that

$$\mathbf{E}\left[\sup_{t\le T/a}\|E(t)\|_\infty^2\right] \le \mathbf{E}\left[\|E(T)\|_\infty^2\right] \le aT.$$

□

Proof of Lemma 2. Let $\rho, \rho' \in \mathcal{P}$. By definition of $\mathcal{P}$, there exist x and x' such that $\rho_{\delta,b} = a^{1-\delta}\sum_{k=1}^n (p_k)^\delta x_{k,b}$ and $\rho'_{\delta,b} = a^{1-\delta}\sum_{k=1}^n (p_k)^\delta x'_{k,b}$. Let θ, θ' be such that $\rho_{0,\theta} = \rho'_{0,\theta'} = m$ and assume without loss of generality that $\theta' \le \theta$. As $x_{k,b}$ is non-increasing in b, this implies that $x_{k,\theta} \ge x_{k,\theta'}$. Hence, we have:

$$\begin{aligned}
|\rho_{1,\theta} - \rho_{1,\theta'}| &= \sum_{k=1}^n p_k(x_{k,\theta} - x_{k,\theta'}) \le \sum_{k=1}^n a(x_{k,\theta} - x_{k,\theta'}) \\
&= |\rho_{0,\theta} - \rho_{0,\theta'}| \le \left|\rho_{0,\theta} - \rho'_{0,\theta'}\right| + \left|\rho'_{0,\theta'} - \rho_{0,\theta'}\right| \\
&= |\rho_{0,\theta'} - \rho_{0,\theta'}| \le \|\rho - \rho'\|_\infty . \qquad \text{(B.1)}
\end{aligned}$$

Therefore:

$$|g_m(\rho) - g_m(\rho')| = \left|\rho_{1,\theta} - \rho'_{1,\theta'}\right| \le |\rho_{1,\theta} - \rho_{1,\theta'}| + \left|\rho_{1,\theta'} - \rho'_{1,\theta'}\right| \le 2\|\rho - \rho'\|_\infty ,$$

where the last inequality comes from (B.1). □

Appendix B.1. Proof of Lemma 3

Proof of Lemma 3. The function $f : \mathcal{P} \to \operatorname{span}(\mathcal{P})$ is given by

$$f_{\delta,b}(\rho) = a(\rho_{\delta+1,\theta_{\zeta_b}-1} - \rho_{\delta+1,b}), \qquad \text{(B.2)}$$

where θ_ℓ and ζ_b are two functions of ρ that are defined by

$$\rho_{0,\theta_\ell} = m_\ell + \cdots + m_h \text{ and } \theta_{\zeta_b} \le b < \theta_{\zeta_b+1}. \qquad \text{(B.3)}$$

We begin by the proof of *(i)* which states that $(\rho_{0,b} - \rho'_{0,b})(f_{0,b}(\rho) - f_{0,b}(\rho')) \le 2a\|\rho - \rho'\|_\infty^2$. Let $\rho, \rho' \in \mathcal{P}$ and let ζ_b and ζ'_b be defined as in Equation (B.3). We have

$$\begin{aligned}
&(\rho_{0,b} - \rho'_{0,b})(f_{0,b}(\rho) - f_{0,b}(\rho')) \\
&= a(\rho_{0,b} - \rho'_{0,b})(\rho'_{\delta+1,b} - \rho_{\delta+1,b} + \rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta'_b-1}}) \\
&\le a\|\rho - \rho'\|_\infty^2 + a(\rho_{0,b} - \rho'_{0,b})(\rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta'_b-1}}).
\end{aligned}$$

We then distinguish three cases:

- If $\zeta_b = \zeta'_b$, then we can use Lemma 2 to show that we have $\left|\rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta'_b-1}}\right| \le \|\rho - \rho'\|_\infty$, which implies that $(\rho_{0,b} - \rho'_{0,b})(\rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta'_b-1}}) \le \|\rho - \rho'\|_\infty^2$.

- If $\zeta_b > \zeta'_b$, then Equation (B.3) implies that $\rho_{0,b} \geq m_{\zeta_b+1} + \cdots + m_h > \rho'_{0,b}$. $\zeta_b > \zeta'_b$ also implies that $\rho'_{\delta+1,\theta'_{\zeta'_b-1}} > \rho'_{\delta+1,\theta'_{\zeta_b-1}}$. Hence,

$$\begin{aligned}&(\rho_{0,b} - \rho'_{0,b})(\rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta'_b-1}})\\ &\leq (\rho_{0,b} - \rho'_{0,b})(\rho_{\delta+1,\theta_{\zeta_b-1}} - \rho'_{\delta+1,\theta'_{\zeta_b-1}}) \leq \|\rho - \rho'\|_\infty^2,\end{aligned}$$

  where the last inequality comes from Lemma 2.

- The case $\zeta_b < \zeta'_b$ is symmetric.

This concludes the proof of *(i)*. Point *(ii)* follows directly from Equation (B.2).

For point *(iii)*, we can mimic the proof of Equation (B.1). Let $\rho, \rho' \in \mathcal{P}$ and assume without loss of generality that $\theta_{\zeta_b} \leq \theta'_{\zeta'_b}$ which implies that $x_{k,\theta_{\zeta_b}} \geq x_{k,\theta'_{\zeta'_b}}$ for all $k \in \{1 \dots n\}$. Equation (B.2) implies that for $\delta \geq 1$, we have

$$\frac{1}{a}(f_{\delta-1,b}(\rho') - f_{\delta-1,b}(\rho)) = \rho'_{\delta,\theta'_{\zeta_b}-1} - \rho_{\delta,\theta_{\zeta_b}-1} + \rho_{\delta,b} - \rho'_{\delta,b}.$$

This shows that

$$\begin{aligned}\frac{1}{a}(f_{\delta-1,b}(\rho') - f_{\delta-1,b}(\rho) - (f_{0,b}(\rho') - f_{0,b}(\rho))) \leq &\left|\rho'_{\delta,\theta'_{\zeta_b}-1} - \rho_{\delta,\theta_{\zeta_b}-1} - (\rho'_{0,\theta'_{\zeta_b}-1} - \rho_{0,\theta_{\zeta_b}-1})\right|\\ &+ 2\|\rho - \rho'\|_\infty.\end{aligned}$$

$$\begin{aligned}\left|\rho_{\delta,\theta_{\zeta_b-1}} - \rho'_{\delta,\theta'_{\zeta_b-1}} - (\rho_{0,\theta_{\zeta_b-1}} - \rho'_{0,\theta'_{\zeta_b-1}})\right| &= \left|\sum_{k=1}^n (a^{1-\delta}(p_k)^\delta - a)(x_{k,\theta_{\zeta_b}} - x'_{k,\theta'_{\zeta'_b}})\right|\\ &\leq \left|\sum_{k=1}^n (a^{1-\delta}(p_k)^\delta - a)(x_{k,\theta_{\zeta_b}} - x_{k,\theta'_{\zeta'_b}})\right| + \|\rho - \rho'\|_\infty\\ &= \sum_{k=1}^n (a - a^{1-\delta}(p_k)^\delta)(x_{k,\theta_{\zeta_b}} - x_{k,\theta'_{\zeta'_b}}) + \|\rho - \rho'\|_\infty\\ &\leq \sum_{k=1}^n a(x_{k,\theta_{\zeta_b}} - x_{k,\theta'_{\zeta'_b}}) + \|\rho - \rho'\|_\infty\\ &\leq \left|\sum_{k=1}^n a(x_{k,\theta_{\zeta_b}} - x'_{k,\theta'_{\zeta'_b}})\right| + 2\|\rho - \rho'\|_\infty\\ &\leq 3\|\rho - \rho'\|_\infty\end{aligned}$$

where we used the assumption $x_{k,\theta_{\zeta_b}} \geq x_{k,\theta'_{\zeta'_b}}$.

□