Analyzing replacement policies in list-based caches with non-uniform access costs

Giuliano Casale
Department of Computing
Imperial College London, UK
g.casale@imperial.ac.uk

***Abstract*—List-based caches can offer lower miss rates than single-list caches, but their analysis is challenging due to state-space explosion. We analyze in this setting randomized replacement policies for caches with non-uniform access costs. In our model, costs can depend on the stream a request originated from, the target item, and the list that contains it.**

We first show that, similarly to the uniform-cost case, the random replacement (RR) and first-in first-out (FIFO) policies can be exactly analyzed using a product-form expression for the equilibrium state probabilities of the cache. We then tackle the state space explosion by means of the singular perturbation method, deriving limiting expressions for the equilibrium performance measures as the number of items and the cache capacity grow in a fixed ratio. Simulations indicate that our asymptotic formulas rapidly converge to the cache equilibrium distribution.

I. Introduction

Replacement policies provide a strategy to select items to replace in a cache and are an important performance factor influencing the design of web systems [27], content delivery networks [2], and peer-to-peer traffic [30]. We here focus on the theoretical analysis of randomized policies operating within a single cache, a problem that has attracted much attention over the years [3], [5], [10], [11], [16], [24], [28], [32]. We consider in particular *list-based* caches, which can deliver lower miss rates than single-list caches [14], [25], but due to state-space explosion issues are still not fully understood from a theoretical standpoint. We ask in particular the question on how to analyze the impact of access costs to items and lists.

Costs are useful in cache modeling to abstract factors such as item importance, fetch latency, optional eviction, or access price [4], [26], [27], [31]. Models that ignore these factors are said to assume uniform costs. In the literature, [31] is among the first works to analyze the least-recently used (LRU) and CLIMB policies when an item has an access cost. Later, [4] studies optimal policies for a single-list with non-uniform costs. Following the randomization approach proposed in [31], costs are here modeled as probabilities that items are promoted to a deeper list as a result of a cache hit. Other cost-based algorithms are surveyed in [27]. The goal of this work is to generalize analysis methods for list-based caches to account for access costs.

This extension is motivated by the recent development in [14] of the first general framework to analyze list-based caches with the random replacement (RR), FIFO, or CLIMB policies, under the independent reference model (IRM) [1], [20]. Within that study, it is observed that list-based caches can deliver lower miss rates than single-list caches, even when the latter are equipped with the LRU policy. The present work is to our knowledge the first one to expand the scope of the theory to non-uniform access costs. Our model is fairly general, as costs can be differentiated across multiple request streams and depend on the requested item and the list that contains it. We also consider a slightly generalized IRM model by allowing arrival rates to depend on the cache state.

Our main technical contributions are as follows. First, we define a Markov process for the list-based RR and FIFO policies with non-uniform costs and show that, similarly to the uniform case, this process admits a product-form solution that extends the one available for models with uniform costs. Next, we develop an asymptotic expression of the normalizing constant of state probabilities that allows us to derive the limiting values of the miss rates as the scale of the system grows. Our approach applies the singular perturbation method used in geometric optics and queueing theory [21]. We are not aware of this method having been applied before to caching theory.

We also derive exact recurrence relations for the cache performance measures, which are exploited to obtain a set of analytical bounds on cache performance measures. These can inexpensively approximate a cache performance in problems where speed of analysis is critical, such as within numerical optimization programs. We use examples and simulations to illustrate the validity and accuracy of our approximations.

The paper is organized as follows. The reference model is introduced in Section II and its equilibrium analyzed in Section III. An exact analysis is developed in Section IV. Section V develops the normalizing constant asymptotics. Iterative analysis and bounds are introduced in Section VI. Numerical results are reported in Section VII. Section VIII overviews related work and it is followed by conclusions. Proofs and background material are given in the Appendix.

II. Preliminaries

Throughout, we use i, k as item indexes, l, j as list indexes, and v as stream index. Our reference model consists of a cache partitioned into h lists arranged in a serial topology, each having a capacity of m_l items, $l = 1, \dots, h$. We use the convention that $l = 0$ indexes items outside the cache. We denote the capacity vector by $\boldsymbol{m} = (m_1, \dots, m_h)$ and set $m = \sum_{l=1}^{h} m_l$.

The total number of items is n and we require it to exceed the cache capacity ($n > m$). We take the usual assumption that items have identical sizes. The time required to replace an item is assumed to be negligible compared to inter-request time, and thus treated as instantaneous. The cache serves u independent streams, each issuing requests for item k in list l according to a Poisson process with rate λ_{vkl}, $v = 1, \dots, u$, $k = 1, \dots, n$, $l = 0, \dots, h$.

A. List-based replacement policies

The challenge in designing effective replacement policies is that item popularity is not known apriori, otherwise one could minimize miss rates by keeping the most frequently accessed objects in the cache [31]. The focus is therefore on policies that self-organize item storage so to optimize miss rates. List-based replacement policies under uniform access costs include RR($\boldsymbol{m}$), FIFO($\boldsymbol{m}$), CLIMB($\boldsymbol{m}$), and LRU($\boldsymbol{m}$) [14], [15]. In these policies, when an item i residing in list $l < h$ is requested, it is promoted to list $l+1$, demoting an item chosen according to a policy-dependent criterion. A detailed specification of the policies can be found in Appendix A. To simplify notation, in the following we omit the dependence on $\boldsymbol{m}$.

Our goal is in particular to investigate RR and FIFO in the context of non-uniform access costs. Since CLIMB can be reduced to a special case of these policies, it is implicitly covered by our analysis. Contrary to RR and FIFO, the Markov process of LRU is not known to admit an explicit solution in list-based caches and thus it is analyzed using approximations [15]. For this reason, and similarly to prior work on randomized policies [14], [18], [32], LRU falls beyond the scope of our investigation, which stems from an exact product-form result that holds only for RR and FIFO.

The variants of the previous policies that allow for non-uniform costs are denoted by RR-c, FIFO-c, LRU-c, and CLIMB-c. The last two variants are developed in [31] for single-list caches. The authors use a randomization approach to model the impact of costs on the cache dynamics. Compared to [31], we also adopt randomization, but focus the analysis on list-based caches.

B. Cost model

Let a_{vkl}, $l = 1, \dots, h-1$, denote the cost for item k, currently residing in list l, to access list $l+1$ after a hit from stream v. Similarly, let a_{vk0} be the cost of loading item k into the cache after a cache miss from stream v. Following [31], we define the *normalized access cost* to an item as $c_{vkl} = a_{vkl}/\max_k a_{vkl}$, which we use to define the probability that, after stream v hits item k in list l, item k is allowed to move to list $l+1$. This randomization ensures that items are promoted in proportion to their cost. Indeed, it is best to retain in the cache high-cost items, unless low-cost items are requested at a frequency that balances the gap in cost [4], [31]. In the next section, we build upon the latter observation to introduce a notion of *cost factor* that encompasses both access costs and request rates.

III. Analytical model

A. Markov process

Define the state vector $\boldsymbol{s} = [s(i,j)]$, such that $s(i,j) \in [1,n]$ is the index of the item in position i of list j, and let $\mathcal{S}$ be the space of all feasible cache states. We take the convention that $m_0 = n - m$ and that $s(i,0)$ indexes the item with the ith largest index among those outside the cache. Let $\pi(\boldsymbol{s})$ be the equilibrium probability of the continuous-time Markov chain modeling the replacement policy. We also assume that the cache is initially full. We now give a product-form result for $\pi(\boldsymbol{s})$.

Theorem 1. *In list-based caches, the RR-c and FIFO-c replacement policies admit the equilibrium distribution*

$$\pi(\boldsymbol{s}) = \frac{\prod_{j=0}^{h} \prod_{i=1}^{m_j} \gamma_{s(i,j)j}}{E(\boldsymbol{m})} \tag{1}$$

for all $\boldsymbol{s} \in \mathcal{S}$, where $E(\boldsymbol{m})$ is a normalizing constant and

$$\gamma_{ij} = \begin{cases} \prod_{l=0}^{j-1} \sum_{v=1}^{u} \lambda_{vil} c_{vil} & \text{if } j > 0 \\ 1 & \text{if } j = 0 \end{cases}$$

for all $i = 1, \dots, n$ and $j = 0, \dots, h$.

The proof is given in Appendix B. We refer to γ_{kl} as the *cost factor* for item k to access list l. The cost factor differs from the access cost c_{vkl} in that it accounts for both the costs and the item request rates involved in item k movement from outside the cache into list l. Note that in uniform cost models with level-independent arrivals, the cost factor reduces to $\gamma_{kl} = \lambda_k^l$ in continuous-time, where $\lambda_k = \sum_{v=1}^{u} \lambda_{vk0}$ is the total request rate for item k, and to $\gamma_{kl} = p_k^l$ in discrete-time, where p_k is the popularity of item k.

B. Performance measures

Let $\pi_{kl}(\boldsymbol{m})$ be the (marginal) probability that item k is in list l, and let $\pi_{k0}(\boldsymbol{m})$ denote the miss ratio for item k. By definition we have $\sum_{l=0}^{h} \pi_{kl}(\boldsymbol{m}) = 1$ and since at all times list l contains exactly m_l items we have

$$\sum_{k=1}^{n} \pi_{kl}(\boldsymbol{m}) = m_l \tag{2}$$

Using the product-form result (1), we can write

$$\pi_{kl}(\boldsymbol{m}) = m_l \gamma_{kl} \frac{E_k(\boldsymbol{m} - 1_l)}{E(\boldsymbol{m})} \tag{3}$$

where 1_l is the unit vector in direction l, $E_k(\boldsymbol{m})$ is the normalizing constant of a model without item k, m_l counts the possible positions of k within list l, and by (1) the remaining terms give the probability of states in which item k resides in list l.

Define now $\pi_k(\boldsymbol{m}) = \sum_{l=1}^{h} \pi_{kl}(\boldsymbol{m})$ as the probability that item k is in the cache. Noting that $E_k(\boldsymbol{m})$ encompasses all and only states where k is outside the cache, we can write the miss ratio for item k as

$$\pi_{k0}(\boldsymbol{m}) = \frac{E_k(\boldsymbol{m})}{E(\boldsymbol{m})} = 1 - \pi_k(\boldsymbol{m}) \tag{4}$$

From the definitions, the miss rate for stream v and item k is

$$M_{vk}(\boldsymbol{m}) = \pi_{k0}(\boldsymbol{m})\lambda_{vk0}c_{vk0} \tag{5}$$

The miss rate of item k is then $M_k(\boldsymbol{m}) = \sum_v M_{vk}(\boldsymbol{m})$ and the cache miss rate is $M(\boldsymbol{m}) = \sum_{v,k} M_{vk}(\boldsymbol{m})$.

IV. Exact analysis

In this section, we derive recurrence relation to calculate the performance measures. The algorithms we derive are exact and require polynomial time and space in n and m for constant h. Thus, they ease the problem that the state space $\mathcal{S}$ grows exponentially in size with n and m, preventing direct numerical analysis of RR-c and FIFO-c by (1).

A. Recurrence relation for the normalizing constant

We begin by conditioning on the list containing item k, so that we can recursively express the normalizing constant as

$$E(\boldsymbol{m}) = E_k(\boldsymbol{m}) + \sum_{j=1}^{h} m_j \gamma_{kj} E_k(\boldsymbol{m} - 1_j) \tag{6}$$

for an arbitrary $k = 1, \dots, n$. Here, m_j accounts for the permutations of the positions of item k within list j. A similar expression is obtained in [14] for uniform cost models. The recurrence relation requires the boundary conditions $E(\mathbf{0}) = 1$, and $E(\boldsymbol{m}) = 0$ if either $\sum_i m_i > n$ or $m_j < 0$ for any j.

Using (6), $E(\boldsymbol{m})$ can be computed in $\mathcal{O}(n\prod_{i=1}^{h}(m_i+1))$ time and space. Note that (6) can incur floating-point range exceptions due to the rapid growth of $E(\boldsymbol{m})$. Such exceptions can be addressed either by exact arithmetic or by dynamic scaling of the normalizing constant value [23].

B. Exact analysis of miss rates

We now consider the problem of computing the miss rate $M(\boldsymbol{m})$. Let us define

$$\xi_l(\boldsymbol{m}) = m_l \frac{E(\boldsymbol{m} - 1_l)}{E(\boldsymbol{m})} \tag{7}$$

and denote $F_l(\boldsymbol{m}) = m_l/\xi_l(\boldsymbol{m})$. Note that by (6) and by the definition of miss rate, we may write

$$M(\boldsymbol{m}) = \frac{E(\boldsymbol{m} + 1_1)}{E(\boldsymbol{m})} = F_1(\boldsymbol{m} + 1_1) = \frac{m_1 + 1}{\xi_1(\boldsymbol{m} + 1_1)} \tag{8}$$

Thus, computational methods for $\xi_l(\boldsymbol{m})$ are sufficient to determine the miss ratio $M(\boldsymbol{m})$. To this end, we observe that $\xi_l(\boldsymbol{m})$ admits the exact recurrence relation

$$\xi_l(\boldsymbol{m}) = \frac{1 + \sum_{\substack{j=1 \\ j\neq l}}^{h} \gamma_{kj}\xi_j^{-k}(\boldsymbol{m} - 1_l) + \gamma_{kl}\xi_l^{-k}(\boldsymbol{m} - 1_l)}{\frac{I\{n>m\}}{\xi_l^{-k}(\boldsymbol{m})} + \sum_{\substack{j=1 \\ j\neq l}}^{h} \gamma_{kj}\frac{\xi_j^{-k}(\boldsymbol{m}-1_l)}{\xi_l^{-k}(\boldsymbol{m}-1_j)} + \gamma_{kl}} \tag{9}$$

for arbitrary $k = 1, \dots, n$, where I is the indicator function. Relation (9) is a non-uniform cost extension of known results for $F_l(\boldsymbol{m})$ [11, Eq. 2.7] and [14, Thm. 2] in uniform cost models. We omit the details of the proof of (9) as this follows similarly to earlier work by applying (6) to (7), and then dividing numerator and denominator by $E_k(\boldsymbol{m} - 1_l)$. Using (9), we can compute the miss rate in $\mathcal{O}(hn\prod_{j=1}^{h}(m_h+1))$. The recursion has boundary conditions: i) $\xi_l(\boldsymbol{m}) = +\infty$ if $m > n$ or $\min_l m_l < 0$; ii) $\xi_l(\mathbf{0}) = 0$; iii) $\xi_l(1_l) = \gamma_{k,l}$ if $m_k = 1$ and $n = 1$; iv) $\xi_l(\boldsymbol{m}) = 0$ if $m_l = 0$.

C. Exact analysis of marginal probabilities

Note that (9) provides miss rates, but neither marginal state probabilities nor item miss rates. In small and medium-sized models such quantities satisfy the following recurrence relation.

Theorem 2. *The probability of finding item k in list l is*

$$\pi_{kl}(\boldsymbol{m}) = \gamma_{kl}\xi_l(\boldsymbol{m})(1 - \pi_k(\boldsymbol{m} - 1_l)) \tag{10}$$

for all items $k = 1, \dots, n$ and lists $l = 1, \dots, j$.

The proof is given in Appendix C. We note that (10) is similar to the arrival theorem for queueing networks [29]. It may be seen as relating the equilibrium distribution of the cache with the conditional distribution of states seen by item k while residing in list l. A similar argument holds also for the arrival theorem, which is sometimes seen as relating the equilibrium distribution of a closed product-form queueing network with the conditional distribution while a job resides at a given station [34].

We are now ready to obtain a recurrence relation for the marginal probabilities. Using (2) and (10), and solving for $\xi_l(\boldsymbol{m})$ we find

$$\xi_l(\boldsymbol{m}) = \frac{m_l}{\sum_{k=1}^{n} \gamma_{kl}(1 - \pi_k(\boldsymbol{m} - 1_l))} \tag{11}$$

Finally, plugging (11) into (10), and using that $\pi_k(\boldsymbol{m}) = \sum_{l=1}^{h} \pi_{kl}(\boldsymbol{m})$, we get the exact recurrence relation

$$\pi_{il}(\boldsymbol{m}) = \frac{m_l\gamma_{il}(1 - \sum_{j=1}^{h} \pi_{ij}(\boldsymbol{m} - 1_l))}{\sum_{k=1}^{n} \gamma_{kl}(1 - \sum_{j=1}^{h} \pi_{kj}(\boldsymbol{m} - 1_l))} \tag{12}$$

for all $i = 1, \dots, n$ and $l = 1, \dots, h$. The recursion has termination conditions $\pi_{il}(\boldsymbol{m}) = 0$ if either $\boldsymbol{m} = \mathbf{0}$ or $\min_j m_j < 0$. Solving (12) recursively requires $\mathcal{O}(nh\prod_{i=1}^{h}(m_i+1))$ time and $\mathcal{O}(n\prod_{i=1}^{h}(m_i+1))$ space. Note that from the knowledge of the marginal probabilities, we can compute item miss rates by the expressions in Section III-B.

V. Normalizing constant asymptotics

Exact recurrence relations are too expensive to apply for medium and large sized caches. To address this issue, we obtain an asymptotic limit for $E(\boldsymbol{m})$ that can be used to approximate performance measures under non-uniform costs. The main contribution is to develop an asymptotic expansion for $E(\boldsymbol{m})$ in the limit $n \to \infty$ and $m \to \infty$, with $n/m \sim O(1)$. To this aim, we use the singular perturbation method [21], which reformulates multivariate recurrence relations as partial differential equations (PDEs) that can be explicitly solved.

A. Singular perturbation method

Without loss of generality, we consider (6) with $k = n$ and set $H(\boldsymbol{m}) = E(\boldsymbol{m})/\prod_{j=1}^{h} m_j!$ and $H_k(\boldsymbol{m}) = E_k(\boldsymbol{m})/\prod_{j=1}^{h} m_j!$, so that (6) may be rewritten as

$$H(\boldsymbol{m}) = H_n(\boldsymbol{m}) + \sum_j \gamma_{nj} H_n(\boldsymbol{m} - 1_j) \tag{13}$$

Let us now scale the capacity vector $\boldsymbol{m}$ and the number of items n by a parameter S and rewrite (13) as

$$h(\boldsymbol{x}, y) = h(\boldsymbol{x}, y - \varepsilon) + \sum_{j=1}^{h} g_j(y) h(\boldsymbol{x} - \varepsilon 1_j, y - \varepsilon) \tag{14}$$

where we define $\varepsilon = 1/S$, $y = n\varepsilon$, $x_j = m_j\varepsilon$, $\boldsymbol{x} = \boldsymbol{m}\varepsilon$, $x = \sum_j x_j$, and $h(\boldsymbol{x}, y)$ is a scaled $H(\boldsymbol{m})$ in which $\boldsymbol{x}$ correspond to list capacities and y to the number of item. Here $g_j(y)$ is assumed to be a smooth function defined such that $g_j(k\varepsilon) = \gamma_{kj}$ and $g_j(0\varepsilon) = g_j(n\varepsilon)$, $\forall j$. Note that by the above definitions, (14) reduces to (13) for $S = 1$.

The recurrence relation (14) can be solved asymptotically for $S \to \infty$, i.e., $\varepsilon \to 0$, using the singular perturbation method [21], which is based on a real-domain variant of the WKB approximation popular in physics. This method solves a recurrence relation by assuming solutions to be in the form

$$h(\boldsymbol{x}, y) \sim D(\varepsilon) B(\boldsymbol{x}, y) e^{\phi(\boldsymbol{x}, y)/\varepsilon} \tag{15}$$

After plugging (15) into the recurrence relation, the latter is expanded in powers of ϵ on both sides. By matching left and right hand sides at different orders of ϵ, one can determine the functions appearing in (15). In the case of multivariate recurrences this often involves solving PDEs related to the eikonal and transport equations. We point to [22] for a tutorial.

The formal derivation to obtain $\phi(\boldsymbol{x}, y)$, $B(\boldsymbol{x}, y)$, and $D(\varepsilon)$ from (14) is given in Appendix D, and leads to

$$\phi(\boldsymbol{x}, y) = \int_0^y \log\left(1 + \sum_{j=1}^{h} g_j(v)\xi_j\right) dv - \sum_{j=1}^{h} x_j \log \xi_j \tag{16}$$

where the terms ξ_j, $j = 1, \ldots, h$ satisfy

$$x_j = \int_0^y \frac{g_j(v)\xi_j}{1 + \sum_{l=1}^{h} g_l(v)\xi_l} dv \tag{17}$$

We show later in Section VI-A that the terms ξ_l may be seen as the asymptotic limits of the values $\xi_l(\boldsymbol{m})$ defined in (11). The remaining coefficients are $D(\varepsilon) = \varepsilon^{h/2}$ and

$$B(\boldsymbol{x}, y) = \frac{(2\pi)^{-h/2}}{\sqrt{\det \boldsymbol{J}}\sqrt{\xi_1 \cdots \xi_h}} \tag{18}$$

where the matrix $\boldsymbol{J} = [J_{jl}]$, $j, l = 1, \ldots, h$, is given by

$$J_{jl} = \int_0^y \frac{\delta_{jl} g_j(v)}{1 + \sum_r g_r(v)\xi_r} dv - \int_0^y \frac{\xi_j g_j(v) g_l(v)}{(1 + \sum_r g_r(v)\xi_r)^2} dv \tag{19}$$

with $\delta_{jl} = 1$ if $j = l$, and $\delta_{jl} = 0$ otherwise.

B. Approximating a finite system

Having determined the asymptotics of $h(\boldsymbol{x}, y)$, we can now use the Euler-Maclaurin formula [33] to rewrite (15) in terms of the variables $\boldsymbol{m}$ and n. For a function $f(x)$ defined in $[0, n]$, the Euler-Maclaurin formula may be written as

$$\sum_{k=1}^{n} f(k) = \int_0^n f(x)dx + \frac{f(n) - f(0)}{2} + O(n^{-1})$$

Scaling variables in (17) so that the integral ranges in $[0, n]$, the error of the Euler-Maclaurin formula is $O(\varepsilon)$, and thus the resulting finite formula is asymptotically exact. Since we have chosen $g_j(0\varepsilon) \equiv g_j(n\varepsilon)$, this is given by

$$m_j = \sum_{k=1}^{n} \frac{\gamma_{kj}\xi_j}{1 + \sum_{l=1}^{h} \gamma_{kl}\xi_l} \tag{20}$$

for $j = 1, \ldots, h$. Under uniform costs, [14] obtains an expression similar to (20) as the steady-state limit of mean field ordinary differential equations. Our derivation shows that a similar result holds under non-uniform costs.

Using Euler-Maclaurin we can also write

$$\phi(\boldsymbol{m}) = \sum_{k=1}^{n} \log\left(1 + \sum_{j=1}^{h} \gamma_{kj}\xi_j\right) - \sum_{j=1}^{h} m_j \log \xi_j$$

and replace $\boldsymbol{J}$ with matrix $\boldsymbol{C} = [C_{jl}]$ where

$$C_{jl} = \sum_{k=1}^{n} \frac{\delta_{jl}\gamma_{kj}}{1 + \sum_{r=1}^{h} \gamma_{kr}\xi_r} - \sum_{k=1}^{n} \frac{\xi_j \gamma_{kj}\gamma_{kl}}{(1 + \sum_{r=1}^{h} \gamma_{kr}\xi_r)^2}$$

The above expressions provide by (15) the asymptotic expansion of $H(\boldsymbol{m})$, which implies by the definitions

$$E(\boldsymbol{m}) \sim (2\pi)^{-h/2} \frac{\prod_{k=1}^{n}\left(1 + \sum_{j=1}^{h} \gamma_{kj}\xi_j\right)}{\prod_{j=1}^{h} \xi_j^{m_j + 1/2}} \frac{\prod_{j=1}^{h} m_j!}{\sqrt{\det \boldsymbol{C}}} \tag{21}$$

where the ξ_j terms are obtained from (20).

C. Performance measures

For large $\boldsymbol{m}$, we can expand $E_k(\boldsymbol{m} - 1_l)$ in a neighborhood of $\boldsymbol{m}$ to obtain by (21) the leading terms

$$\pi_{kl}(\boldsymbol{m}) \sim \begin{cases} \left(1 + \sum_{j=1}^{h} \gamma_{kj}\xi_j\right)^{-1} & \text{if } l = 0 \\ \gamma_{kl}\xi_l \left(1 + \sum_{j=1}^{h} \gamma_{kj}\xi_j\right)^{-1} & \text{if } l \geq 1 \end{cases} \tag{22}$$

for all $k = 1, \ldots, n$ and $l = 0, \ldots, j$. Additional terms in the expansions may also be used to generate higher-order approximations, which involve the derivatives of $\sqrt{\det \boldsymbol{C}}$. However, these are seldom needed since the cost of computing these terms is comparable to computing performance measures by direct application of (21) to the definitions given in Section III-B. We shall indicate the latter approach as the *singular perturbation approximation* (SPA), to distinguish it from the approach that numerically solves (20) for the ξ_l terms and then calculates performance measures by (22). The latter is further developed in the next section and referred to as the *fixed-point iteration* (FPI).

TABLE I
EXAMPLE: NORMALIZING CONSTANT ASYMPTOTICS

n	m_1	m_2	$E(\boldsymbol{m})$	SPA - eq. (21)	relative error
4	2	0	$1.2969 \cdot 10^1$	$1.3691 \cdot 10^1$	0.056
8	4	0	$3.5950 \cdot 10^2$	$3.6940 \cdot 10^2$	0.028
16	8	0	$6.7136 \cdot 10^5$	$6.8063 \cdot 10^5$	0.014
20	10	0	$3.8500 \cdot 10^7$	$3.8926 \cdot 10^7$	0.011
4	1	1	$1.6173 \cdot 10^1$	$1.8919 \cdot 10^1$	0.170
8	2	2	$2.5697 \cdot 10^2$	$2.7810 \cdot 10^2$	0.082
16	4	4	$6.2439 \cdot 10^4$	$6.4990 \cdot 10^4$	0.041
20	5	5	$9.7236 \cdot 10^5$	$1.0042 \cdot 10^6$	0.033

1) Example: Table I illustrates the singular perturbation method. We consider a small-scale model where $E(\boldsymbol{m})$ can be computed numerically using (6). The model has $u = 2$ users, $\lambda_{1kl} = k^{-0.6}$, $\lambda_{2kl} = k^{-1.4}$. We set access costs to $a_{vkl} = 1$. We fix $n = 2S$ and $m = S$ and use scalings $S = 2, 4, 8, 10$. As shown in the table, (21) converges to the exact value $E(\boldsymbol{m})$.

VI. FURTHER APPROXIMATIONS

A. Fixed-point iteration (FPI)

We now introduce a fixed-point method to obtain the ξ_l variables. Let us first consider a regular perturbation expansion [22] of (12), which for large m and n, with $m/n \sim O(1)$, assumes the form $\pi_{il}(\boldsymbol{m}) \sim \hat{\pi}^0_{il} + \varepsilon\hat{\pi}^1_{il} + ...$, for small $\varepsilon > 0$. Plugging the last expression in (12), we get at the lowest order

$$\hat{\pi}^0_{il} = \frac{m_l \gamma_{il}(1 - \sum_{j=1}^{h} \hat{\pi}^0_{ij})}{\sum_{k=1}^{n} \gamma_{kl}(1 - \sum_{j=1}^{h} \hat{\pi}^0_{kj})} \tag{23}$$

for $i = 1, \dots, n$, $l = 1, \dots, h$. This is a non-linear system of equations with nh equations and nh unknowns. It is possible to verify that the system is solved by setting

$$\hat{\pi}^0_{il} = \frac{\gamma_{il}\xi^0_l}{1 + \sum_{j=1}^{h} \gamma_{ij}\xi^0_j} \tag{24}$$

where $\xi^0_l = m_l(\sum_{k=1}^{n} \gamma_{kl}(1 - \sum_{l=1}^{h} \hat{\pi}^0_{il}))^{-1}$. Applying the regular perturbation expansion to the terms in (2) and plugging (24) in the resulting expression, we see that as $\varepsilon \to 0$ the ξ^0_l terms become solutions of (20). Since ξ^0_l is the value of ξ_l corresponding to the leading terms of the expansion of $\pi_{il}(\boldsymbol{m})$, this provides support to our earlier statement that the variables ξ_l appearing in the expansion of $E(\boldsymbol{m})$ are the limits of $\xi_l(\boldsymbol{m})$ for $\varepsilon \to 0$. A similar conclusion follows by calculating the leading term of (7) as in Section V-C.

The explicit expression of ξ^0_l enables us to solve (20) by introducing a fixed point iteration over the marginal probabilities. This is given by

$$\xi^{(t+1)}_l = \frac{m_l}{\sum_{k=1}^{n} \gamma_{kl}(1 - \sum_{l=1}^{h} \pi^{(t)}_{il})} \tag{25a}$$

$$\pi^{(t+1)}_{il} = \frac{\gamma_{il}\xi^{(t+1)}_l}{1 + \sum_{j=1}^{h} \gamma_{ij}\xi^{(t+1)}_j} \tag{25b}$$

for all $i = 1, \dots, n$, $l = 1, \dots, h$ and iterations $t \geq 0$. As mentioned before, the FPI method solves (20), which can be done by (25), and then plugs the obtained ξ_l values into (22) to obtain performance measures. The iteration can be initialized with a guess of the marginal probabilities, e.g., $\pi^{(0)}_{il} = 1/(h+1)$. We stop it when the maximum relative change of the miss ratios $\pi^{(t)}_{i0} = 1 - \sum_l \pi^{(t)}_{il}$ across iterations does not exceed a user-specified tolerance (e.g., $\tau = 10^{-6}$). We did not observe situations in which the iteration did not converge or was sensitive to the initial conditions.

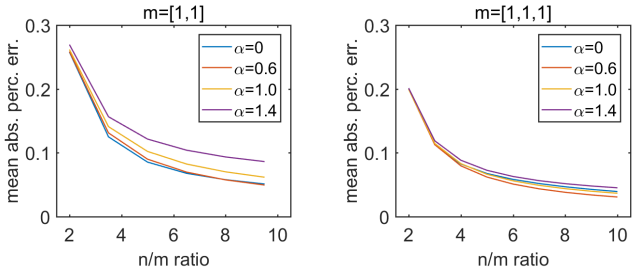

Fig. 1. Approximation error in $[0, 1]$ of (25) for the miss ratio $M(\boldsymbol{m})$ in two-stream caches with $h = 2$ and $h = 3$ lists, with convergence tolerance $\tau = 10^{-6}$. The cache has two users with request rates $\lambda_{1kl} = p_k$, $p_k = k^{-\alpha}$, $k = 1, \dots, n/3$, $\lambda_{2kl} = q_k$, $q_k = k^{-\alpha}$, $k = (n/3 + 1), \dots, n$, and $\lambda_{1kl} = \lambda_{2kl} = 0$ otherwise. Costs are set to $a_{vkl} = 1$.

1) Example: Figure 1 illustrates the miss ratio computed by fixed-point iteration against the exact solution obtained using (9) for two caches with $h = 2$ and $h = 3$ lists and $u = 2$ streams. The streams request two independent sets of objects, both according to a Zipf-like popularity distribution with parameter α.

As the ratio n/m grows, we see that the cache approaches the asymptotic behavior and the error visibly decreases. It is clear from the example that considering models with a larger number of lists h also reduces error, for fixed ratio n/m, and makes the results less sensitive to the exponent α. The maximum number of iterations of (25a)-(25b) is observed to be $t_{max} = 23$ for $\alpha = 0$, and $t_{max} = 36$ for $\alpha = 1.4$. This means in practice that the iteration converges within the tolerance in just a few milliseconds. We also observe the number of iterations to slightly decrease as n/m grows. These results are representative also of other parameterizations. Experiments with larger models are reported in Section VII.

B. Bound analysis

Here we develop some simple analytical bounds on $\xi_l(\boldsymbol{m})$ and π_{i0}. Bounds on $\xi_l(\boldsymbol{m})$ allow us to bound by (8) the miss rate $M(\boldsymbol{m})$, whereas bounds on the probabilities π_{i0} can be used to bound the miss rates using (5).

1) Bounds on $\xi_l(\boldsymbol{m})$: It is possible to verify that

$$m_l \left(\gamma^+_l(n - m + 1)\right)^{-1} \leq \xi_l(\boldsymbol{m}) \leq m_l \left(\gamma^-_l(n - m + 1)\right)^{-1} \tag{26}$$

where $\gamma^+_l = \max_k \gamma_{kl}$ and $\gamma^-_l = \min_k \gamma_{kl}$. These follow from (11) by bounding γ_{kl} with γ^+_l or γ^-_l and noting that $\sum_k (1 - \pi_k(\boldsymbol{m} - 1_l)) = n - m + 1$. For an important class of models, we then state a tighter lower bound. Lower bounds on $\xi_l(\boldsymbol{m})$ are the most important in practice, as they lead to pessimistic estimates of the miss rates. The bound assumptions

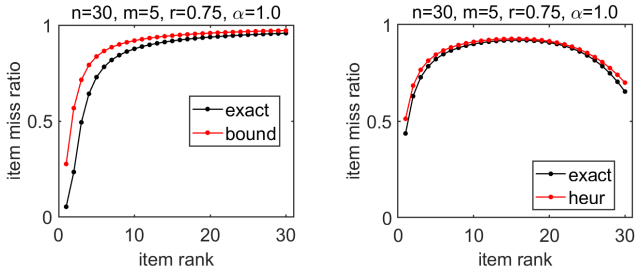

Fig. 2. Single-stream cache with capacity $\boldsymbol{m} = (3, 2)$ and request rates $\lambda_{vkl} = p_k\lambda$, with $\lambda = 1$ and Zipf-like popularity $p_k = k^{-\alpha}$. Items are ranked according to p_k. The left figure shows a monotone cache with $a_{vkl} = (1-r)r^k$, the right cache is non-monotone with access costs c_{vkl}^{-1}.

are satisfied by a rather broad class of models including, but not limited to, models with uniform costs.

Theorem 3. *Consider a cache where for each pair of items i and $k > i$, we have $\gamma_{il} \geq \gamma_{kl}$, $\forall l$. Then*

$$m_l \left(\bar{\gamma}_l(n - m + 1)\right)^{-1} \leq \xi_l(\boldsymbol{m}) \tag{27}$$

where $\bar{\gamma}_l = \sum_j \gamma_{jl}/n$ is the average cost factor for list l.

The proof is given in Appendix E and shows that the argument generalizes for any list where $\gamma_{kl} \geq \gamma_{il}$ implies $\pi_k(\boldsymbol{m} - 1_l) \geq \pi_i(\boldsymbol{m} - 1_l)$. If the property holds for all lists, we say that the model is *monotone*.

2) Bounds on $\pi_{k0}(\boldsymbol{m})$: We now develop upper bounds on the miss rate $\pi_{k0}(\boldsymbol{m})$. We begin by establishing two general monotonicity properties.

Lemma 1. $\xi_j(\boldsymbol{m} - 1_l) \leq \xi_j(\boldsymbol{m}) \leq \xi_j^{-k}(\boldsymbol{m})$, $\forall j, l, k$.

Lemma 2. $\pi_{k0}(\boldsymbol{m}) \leq \pi_{k0}(\boldsymbol{m} - 1_l)$, $\forall k, l$.

The proofs are given in Appendix F and G. We are now ready to give an upper bound on the miss rates.

Theorem 4. *The item miss ratios satisfy the bound $\pi_{k0}(\boldsymbol{m}) \leq \left(1 + \sum_{j=1}^{h} \gamma_{kj}\xi_j(\boldsymbol{m})\right)^{-1}$ for all items $k = 1, \dots, n$ and lists $l = 1, \dots, h$.*

The proof is given in Appendix H. Note that bound is asymptotically exact for fixed h, as it converges to (22) when n and m increase in a fixed ratio. A closed-form expression for this upper bound can be readily obtained by replacing $\xi_l(\boldsymbol{m})$ with the lower bound in either (26) or (27). Although (27) is not guaranteed to be a bound in non-monotone models, combining it with Theorem 4 still provides a useful non-iterative heuristic approximation of π_{k0}, as illustrated in the next example.

3) Example: Figure 2 illustrates Theorem 4 for a monotone and a non-monotone cache. Estimates are computed by replacing $\xi_j(\boldsymbol{m})$ in the bound of Theorem 4 with (27). As shown in the figure, the heuristic can return accurate values.

VII. Numerical results

A. Simulation-based validation

Since exact numerical solution is applicable only to small models, we have performed a simulation study of RR-c and FIFO-c to assess the accuracy of the proposed approximations. The set of parameters used in the study is given in Table II. Whenever $n = m$, we set $n = m + 2$ to avoid degeneracies.

TABLE II
Simulation parameters for RR-c, FIFO-c, LRU-c

n	Number of items	10, 1000
n/m	Items-to-capacity ratio	2, 4, 10
h	Number of lists	1, 2, 5
u	Number of streams	1, 4
α	Zipf parameter	0.6, 1.0, 1.4
c	Normalized access cost	0.1, 0.5, 1.0

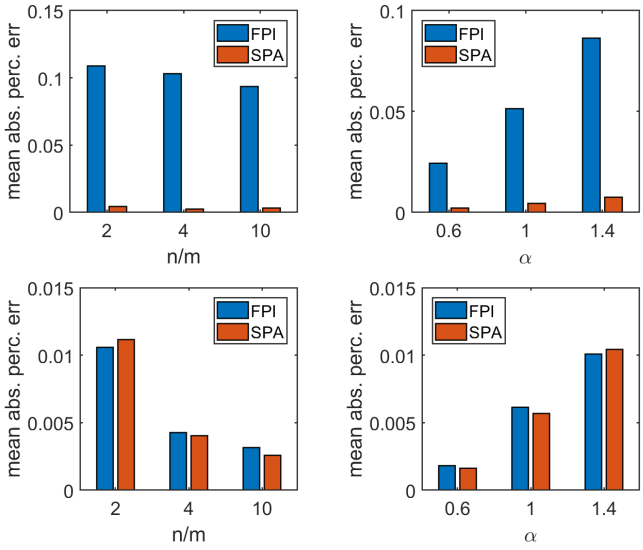

Fig. 3. Sensitivity of FPI and SPA to changes in the item-to-capacity ratio n/m and in the parameter α for caches with $n = 10$ items (top row) and $n = 1000$ items (bottom row).

Access costs are set identically for each list and equal to c. Each stream sends Poisson requests at rate $\lambda = 1$, scaled by a Zipf-like distribution with parameter α, with the object ranks assigned uniformly at random within each stream. Each simulation collects 10^9 state samples from the cache. From the results, we observe as expected that RR-c and FIFO-c have statistically indistinguishable performance, thus we discuss only RR-c and compare the FPI and SPA methods of Section V-C, based on the mean absolute percentage error on the item miss ratios $\pi_{i0}(\boldsymbol{m})$.

In small models with $n = 10$ items, SPA incurs a mean absolute percentage error lower than FPI, 0.4% instead of 10.2%. For such instances, we have observed the maximum error of FPI to reach 35.1%, whereas for SPA it is just 0.6%. We have observed the largest errors of FPI to be in the estimation of miss probabilities for high popularity items. Conversely, with $n = 1000$ items the cache dynamics reaches the asymptotic regime and as expected both methods converge with a mean absolute percentage error of 0.6% for both methods, and a maximum just below 0.7% error for both methods.

We have also investigated the sensitivity of the results to the simulation parameters. We have noticed the results to be

nearly insensitive to changes in the access cost c and the number of streams u. Figure 3 illustrates the dependence of the error on the ratio n/m and α. Both approximations tend to be more accurate in caches with a higher n/m ratio, thus we expect our methods to perform accurately in models where a small fraction of the items can be cached. Some dependence is observed with respect to increasing value of α, however the implied error becomes rather small in all cases if $n = 1000$. We have observed a similar behavior also under changes of the number of lists h, where accuracy is generally better for smaller values of h, but errors become negligible if $n = 1000$.

Overall, the results indicate that FPI suffices in practice on realistic models, but SPA provides more robust estimates for small models far from the asymptotic regime.

B. Youtube trace

Lastly, we use a trace-driven simulation to illustrate the applicability of the proposed methods to real workloads. We have simulated the 2008 Youtube trace collected in [35] and also used to validate the list-based cache models in [14], [18]. The trace consists of $611,968$ requests, spanning $n = 303,332$ items and originating from $u = 16,337$ client IPs. Each client IP is mapped to a single stream v.

We run trace-driven simulations, using caches with capacity $m = 5000$, $m_1 = 2900$, and $m_2 = \cdots = m_h = (m - m_1)/(h - 1)$ and varying the number of lists $h = 2, 3, 5$, the replacement policy (FIFO-c, RR-c), and the normalized access costs to the next list ($c = 0.1, 0.5, 1.0$). For each simulation, we compute the item-miss rates M_k, $k = 1 \dots, n$, and compare them with the estimates returned by FPI. In the stochastic model, each stream v is modeled as a Poisson process with rate $\lambda_{vk} = n_{vk}/T$, where n_{vk} is the number of times item k is requested by stream v and T is the timespan of the trace.

Despite the large number of items, FPI produces an estimate of all item miss rates in just 8.2 seconds on a laptop (Intel Core i5). Even though the Youtube trace is temporally-correlated, thus departing from IRM and Poisson assumptions, and about half of the items are accessed just a single time each, the mean absolute percentage error of FPI on the M_k values is only 1.74%, with a maximum error of just 4.68%. Both values are compatible with the uncertainties of the simulation, in which the trace is simulated only once. Overall, the experiments indicate that our model can reliably reproduce simulation results in the presence of access costs and multiple request streams.

VIII. Related work

Recently, most of the research literature has focused on time-to-live (TTL) caches, which can be analyzed either by characteristic time approximation [7] or by exact methods [6]. The former has also been shown to be valid for randomized policies [12]. However, there is still a limited body of work on using this method to study list-based caches [15].

This paper is related to works that consider the independent reference model (IRM), which include [20], [16], [10], [11], [9], and references therein. Although real systems deviate from the IRM assumptions, IRM theoretical results provide useful insights, such as the characterization of the miss ratio of LRU [5], and formulas for the approximate analysis of cache networks [13] and hybrid memory systems [18].

Recently, some frameworks that simplify the mathematical analysis of caches have been investigated [25], [14]. This work extends the work in [14], which provides a mathematical framework to study list-based caches. [18] offers another example of generalization of [14], but to list-based caches with flat and layered list topologies.

IX. Conclusion

We have generalized models for list-based caches to include non-uniform access costs. After solving the Markov process for the RR and FIFO policies, we have developed exact and approximate formulas to efficiently analyze performance measures such as miss rates. We have then applied singular perturbation methods to determine the asymptotic behavior of the cache. Simulation results indicate that our approximations typically incur less than 1% error on large models.

In the future, it would be interesting to apply our results to develop policies for cost optimization. A number of studies have investigated optimal cost structures, e.g. [4], [26], but optimality results are not yet available for the class of models studied in this present paper.

X. Acknowledgement

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644869 (DICE) and by a UK Engineering and Physical Sciences Research Council grant (EP/L00738X/1, iBids). Research data is available at http://doi.org/10.5281/zenodo.1134886 (CC BY 4.0 license).

References

[1] A. Aho, P. Denning, and J. Ullman. Principles of optimal page replacement. *JACM*, 18, 1971.

[2] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying. Content-aware caching and traffic management in content distribution networks. *Proc. INFOCOM*, 2858–2866, 2011.

[3] O. I. Aven, E. G. Coffman, and Y. A. Kogan. *Stochastic analysis of computer storage*. Reidel publishing, 1987.

[4] O. Bahat and A. M. Makowski. Optimal replacement policies for non-uniform cache objects with optional eviction. *Proc. INFOCOM*, 427–437. IEEE, 2003.

[5] J. V. D. Berg and A. Gandolfi. LRU is better than FIFO under the independent reference model. *J. App. Prob.*, 29(1):239–243, 1992.

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

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

[8] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Volume II*. Interscience, 1962.

[9] A. Dan and D. F. Towsley. An approximate analysis of the LRU and FIFO buffer replacement schemes. *Proc. SIGMETRICS*, 143–152. ACM, 1990.

[10] R. Fagin. Asymptotic miss ratios over independent references. *J. Comp. and Sys. Sci.*, 14(2):222–250, Apr. 1977.

[11] R. Fagin and T. G. Price. Efficient calculation of expected miss ratios in the independent reference model. *SIAM Journal on Computing*, 7, 1978.

[12] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. *Proc. ITC*, 1–8, 2012.

[13] M. Gallo, B. Kauffmann, L. Muscariello, A. Simonian, and C. Tanguy. Performance evaluation of the random replacement policy for networks of caches. *Perform. Eval.*, 72:16–36, 2014.

[14] N. Gast and B. V. Houdt. Transient and steady-state regime of a family of list-based cache replacement algorithms. *Proc. SIGMETRICS*, 123–136. ACM, 2015.
[15] N. Gast and B. V. Houdt. Asymptotically exact TTL-approximations of the cache replacement algorithms LRU(m) and h-LRU. *Proc. ITC*, 157–165. IEEE, 2016.
[16] E. Gelenbe. A unified approach to the evaluation of a class of replacement algorithms. *IEEE Trans. Computers*, 22(6):611–618, 1973.
[17] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, second edition, 1952.
[18] G. Ju, Y. Li, Y. Xu, J. Chen, and J. C. S. Lui. Stochastic modeling of hybrid cache systems. *Proc. MASCOTS*, 69–78. IEEE, 2016.
[19] F. P. Kelly. *Reversibility and Stochastic Networks*. John Wiley & Sons, New York, 1979.
[20] W. F. King. Analysis of demand paging algorithms. In *IFIP Congress (1)*, 485–490, 1971.
[21] C. Knessl and C. Tier. Asymptotic expansions for large closed queueing networks with multiple job classes. *IEEE Trans. Computers*, 41(4):480–488, 1992.
[22] C. Knessl and C. Tier. Applications of singular perturbation methods in queueing. In *Frontiers in Queueing: Models and Applications in Science and Engineering*, 331–336. CRC Press, 1996.
[23] S. Lam. Dynamic scaling and growth behavior of queueing network normalization constants. *JACM*, 29(2):492–513, 1982.
[24] J. Li, S. Shakkottai, J. C. S. Lui, and V. Subramanian. Accurate learning or fast mixing? Dynamic adaptability of caching algorithms. *CoRR*, abs/1701.02214, 2017.
[25] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. *Proc. of INFOCOM*, 2040–2048, 2014.
[26] G. Neglia, D. Carra, M. Feng, V. Janardhan, P. Michiardi, and D. Tsigkari. Access-time aware cache algorithms. *Proc. of ITC*, 148–156. 2016.
[27] S. Podlipnig and L. Boszormenyi. A survey of web cache replacement strategies. *ACM Computing Surveys*, 35, 2003.
[28] K. Psounis and B. Prabhakar. Efficient randomized web-cache replacement schemes using samples from past eviction times. *IEEE/ACM Trans. Netw*, 10(4):441–455, 2002.
[29] M. Reiser and S. S. Lavenberg. Mean-value analysis of closed multichain queueing networks. *JACM*, 27(2):312–322, 1980.
[30] O. Saleh and M. Hefeeda. Modeling and caching of peer-to-peer traffic. *Proc. ICNP*, 249–258. IEEE, 2006.
[31] D. Starobinski and D. Tse. Probabilistic methods in web caching. *Perform. Eval.*, 46(2-3):125–137, Oct. 2001.
[32] N. Tsukada, R. Hirade, and N. Miyoshi. Fluid limit analysis of FIFO and RR caching for independent reference models. *Perform. Eval.*, 69(9):403–412, 2012.
[33] E. W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman and Hall, 2003.
[34] J. Zahorjan. The distribution of network states during residence times in product form queueing networks. *Perform. Eval*, 4(2):99–104, 1984.
[35] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of YouTube network traffic at a campus network - Measurements, models, and implications. *Computer Networks*, 53(4):501–514, 2009.

APPENDIX

A. List-based replacement policies

• ***RR***$(\boldsymbol{m})$: a cache miss for item i is handled by evicting a random item from list 1 and replacing it with item i. Conversely, a cache hit swaps the positions of item i in list l and of an item k chosen at random in list $l+1$, if $l < h$, or otherwise if $l = h$ item i does not move.

• ***FIFO***$(\boldsymbol{m})$: this policy is similar to RR$(\boldsymbol{m})$, but the demoted item k is picked from the end of list $l+1$ and moved into the former position of i in list l. Then, for $l < h$, all other items in list $l+1$ move back by one position, and i is placed at the front of list $l+1$. For $l = h$, item i does not move.

• ***LRU***$(\boldsymbol{m})$: this policy is similar to FIFO$(\boldsymbol{m})$, but the victim item k is moved to the front of list l, shifting items in l back by one position up to filling the position held by the requested item i. A hit on item i in list h moves i to the list front, shifting back by one position all items in between.

• ***CLIMB***$(\boldsymbol{m})$ [3]: this is simply the limiting case of FIFO$(\boldsymbol{m})$ and RR$(\boldsymbol{m})$ when $\boldsymbol{m} = (1, \dots, 1)$ and $h = m$.

B. Proof of product-form solution

Under RR-c, consider a state $\boldsymbol{s}_t \in \mathcal{S}$, and let $\boldsymbol{s}_{t+1} \in \mathcal{S}$ be obtained from $\boldsymbol{s}_t$ after a request to item k promotes it from list j to list $j+1$, demoting i from list $j+1$ to list j. The transition rates are $q(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}) = \sum_{v=1}^{u} \lambda_{vkj} c_{vkj} / m_{j+1}$ and $q(\boldsymbol{s}_{t+1}, \boldsymbol{s}_t) = \sum_{v=1}^{u} \lambda_{vij} c_{vij} / m_{j+1}$. This holds also for cache misses, if we see items outside the cache as residing in list $j = 0$. Using the definition of the transition rates, it is then possible to check that (1) satisfies for any path $\boldsymbol{s}_0 \to \boldsymbol{s}_1 \to \cdots \to \boldsymbol{s}_S$ Kolmogorov's criterion [19] $\pi(\boldsymbol{s}_0) \prod_{t=1}^{S} q(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t) = \pi(\boldsymbol{s}_S) \prod_{t=1}^{S} q(\boldsymbol{s}_t, \boldsymbol{s}_{t-1})$. The latter is necessary and sufficient for the Markov process to be reversible and for (1) to be its equilibrium distribution [19].

In the case of FIFO-c, the process is not reversible. However, it is sufficient to plug (1) in the global balance equations to verify the statement. The global balance equations are

$$\left(\sum_{j=0}^{h-1} \sum_{k=1}^{m_j} \sum_{v=1}^{u} \lambda_{vs(k,j)j} c_{vs(k,j)j} \right) \pi(\boldsymbol{s}) = \left(\sum_{j=0}^{h-1} \sum_{k=1}^{m_j} \sum_{v=1}^{u} \lambda_{vs(1,j+1)j} c_{vs(1,j+1)j} \right) \pi(\boldsymbol{z}_{kj}) \quad (28)$$

where $s \equiv s(i,j)$ and $\boldsymbol{s}$ is obtained from $\boldsymbol{z}_{kj} \in \mathcal{S}$ by promoting item $s(1, j+1)$ at position k of list j to the front of list $j+1$. From the definitions, according to (1)

$$\frac{\pi(\boldsymbol{z}_{kj})}{\pi(\boldsymbol{s})} = \frac{\gamma_{s(1,j+1)j} \gamma_{s(k,j)(j+1)}}{\gamma_{s(1,j+1)(j+1)} \gamma_{s(k,j)j}} \quad (29)$$

Noting that (28) may be rewritten as

$$\sum_{j=0}^{h-1} \sum_{k=1}^{m_j} \frac{\gamma_{s(k,j)(j+1)} \pi(\boldsymbol{s})}{\gamma_{s(k,j)j}} = \sum_{j=0}^{h-1} \sum_{k=1}^{m_j} \frac{\gamma_{s(1,j+1)(j+1)} \pi(\boldsymbol{z}_{kj})}{\gamma_{s(1,j+1)j}}$$

the result follows after plugging (29).

C. Proof of normalizing constant asymptotic expansion

It is possible to verify with a little algebra that

$$\frac{\partial E(\boldsymbol{m})}{\partial \gamma_{kl}} = m_l E_k(\boldsymbol{m} - 1_l) = \frac{\pi_{kl}(\boldsymbol{m})}{\gamma_{kl}} E(\boldsymbol{m}). \quad (30)$$

Applying this relationship to (6) and using (7) we find

$$\pi_{kl}(\boldsymbol{m}) = m_l \gamma_{kl} \frac{E_k(\boldsymbol{m} - 1_l)}{E(\boldsymbol{m})} = \gamma_{kl} \xi_l(\boldsymbol{m}) \frac{E_k(\boldsymbol{m} - 1_l)}{E(\boldsymbol{m} - 1_l)}$$

and the result follows by (4).

D. Proof of normalizing constant asymptotic expansion

1) Derivation of $\phi(\boldsymbol{x}, y)$: Plugging (15) into (14) we get

$$B(\boldsymbol{x}, y) e^{\phi(\boldsymbol{x}, y)/\varepsilon} = B(\boldsymbol{x}, y - \varepsilon) e^{\phi(\boldsymbol{x}, y-\varepsilon)/\varepsilon} + \textstyle\sum_{j=1}^{h} g_j(y) B(\boldsymbol{x} - \varepsilon 1_j, y - \varepsilon) e^{\phi(\boldsymbol{x} - \varepsilon 1_j, y-\varepsilon)/\varepsilon} \quad (31)$$

Expanding (31) in powers of ε, we obtain at the lowest order $1 = e^{-\phi_y} + \sum_j g_j(y) e^{-\phi_y - \phi_j}$, where ϕ_j (resp. ϕ_y) denotes partial differentiation with respect to x_j (resp. y). Multiplying both sides by e^{ϕ_y} and simplifying yields

$$1 - e^{\phi_y} + \textstyle\sum_{j=1}^{h} g_j(y) e^{-\phi_j} = 0 \tag{32}$$

The method of characteristics for non-linear PDEs [8] yields

$$dy = -e^{\phi_y} dt \tag{33a}$$
$$dx_j = -g_j(y) e^{-\phi_j} dt \tag{33b}$$
$$d\phi = \phi_y dy + \textstyle\sum_j \phi_j dx_j \tag{33c}$$
$$d\phi_y = -\textstyle\sum_j g'(y) e^{-\phi_j} dt \tag{33d}$$
$$d\phi_j = 0 \tag{33e}$$

where all variables are intended as functions of $(t, \boldsymbol{\xi})$, with t being the integration variable over the characteristic curves and let $\boldsymbol{\xi} = (\xi_1, \dots, \xi_h)$ parameterize the family of curves. To solve this system, we follow a strategy similar to the one used in [21] to analyze queueing networks. We first solve (33e) as $\phi_j = \log \xi_j$, where we have set the integration constant to $\log \xi_j$, $j = 1, \dots, h$. Multiplying (33d) by (33a) after dividing both by dt, and integrating we find under the initial conditions $\phi = 0$, $\boldsymbol{x} = 0$, and $y = 0$ if $t = 0$, that $\phi_y = \log(1 + \sum_j g_j(y)\xi_j)$, where intermediate integration constants need to be set to unity for consistency with (32). Plugging the last result in (33a) and integrating we obtain t, after which (33b) yields (17). Using the above results, (33c) integrates to (16).

2) Derivation of $B(\boldsymbol{x}, y)$: Expanding (31) and matching first-order terms, we write after simplifications $\frac{dB}{dt} = \frac{B}{2}(\phi_{yy} e^{\phi_y} + \sum_j g_j(y)(\phi_{jj} + 2\phi_{jy}) e^{-\phi_j})$. Plugging (16), this becomes similar to the transport equation studied in [21]. Similar passages as in [21] yield $B(\boldsymbol{x}, y) = b(\xi_1, \dots, \xi_h)(\det \boldsymbol{J})^{-1/2}(1 + \sum_j g_j(y)\xi_j)^{-1/2}$, in which $b(\xi_1, \dots, \xi_h)$ is an arbitrary function, the matrix $\boldsymbol{J}$ is given in (19), and the ξ_j's depend on $\boldsymbol{x}$ and y via (17).

3) Boundary condition: At the boundary, we choose to match the arbitrary function $b(\xi_1, \dots, \xi_h)$ to the value of $H(\boldsymbol{m})$ when $\gamma_{kj} = g_j(k) = \gamma$, $\forall k, j$, for arbitrary γ. In this case, by definition

$$H_0(\boldsymbol{m}, n) = \frac{n!}{(n-m)! m_1! \cdots m_h!} e^{\log \gamma \sum_j j m_j}$$

Using Stirling approximation $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ and scaling the variables in $H_0(\boldsymbol{m}, n)$ to obtain a scaled constant $h_0(\boldsymbol{x}, y)$, we get after simple manipulations

$$h_0(\boldsymbol{x}, y) \sim \frac{(2\pi)^{-h/2}}{\sqrt{\prod_j x_j}} \sqrt{\frac{y}{y-x}} e^{\gamma/\varepsilon}$$

where $\gamma = y \log y - (y - x)\log(y - x) - \sum_j x_j \log x_j + \log \gamma \sum_j j x_j$. For small $\boldsymbol{x}$ and y, we note that (17) gives $x_j = \xi_j \gamma^j y (1 + \sum_{l=1}^{h} \gamma^l \xi_l)^{-1}$, implying $\xi_j = x_j (y - x)^{-1} \gamma^{-j}$ upon summing over j. Substituting the expression of ξ_j in (16) it is possible to verify that ϕ matches γ. We then require

$$\frac{b(\xi_1, \dots, \xi_h)}{\sqrt{\det \boldsymbol{J}} \sqrt{1 + \sum_j \gamma^j \xi_j}} = \frac{(2\pi)^{-h/2}}{\sqrt{\prod_j x_j}} \sqrt{\frac{y}{y-x}}$$

By definition of J_{ik}, we find $J_{ik} = (y - x)(\delta_{ik}\gamma^i - \frac{x_i}{y}\gamma^k)$. Using this result we compute $\det \boldsymbol{J}$ and determine with a little algebra the required value of $b(\xi_1, \dots, \xi_h)$, which yields (18). Lastly, to match $H_0(\boldsymbol{m})$, we need to set $D(\varepsilon) = \varepsilon^{h/2}$.

E. Proof of Theorem 3

To prove the lower bound, we show that the assumption implies $\pi_i(\boldsymbol{m}) \geq \pi_k(\boldsymbol{m})$, $\forall \boldsymbol{m}$, so that the statement follows by applying Chebyshev's sum inequality [17] to the denominator of (11). Using (4), we need to show that $E_k(\boldsymbol{m}) \geq E_i(\boldsymbol{m})$ for all i, $k > i$. Note that $E_k(\boldsymbol{m})$ is obtained from $E_i(\boldsymbol{m})$ by replacing γ_{kl} in $E_i(\boldsymbol{m})$ with $\gamma_{il} \geq \gamma_{kl}$. The statement then holds since the normalizing constant is by (30) non-decreasing with the cost factors.

F. Proof of Lemma 1

The proof is by induction on the number of items n.
Case $n = m+1$. The case holds since by (7) it is $\xi_j^{-i}(\boldsymbol{m} - 1_l) = 0$, being $E(\boldsymbol{m} - 1_l - 1_j) = 0$.
Induction step. Let the model with $n-1$ items exclude item i, we focus on the hypothesis $\xi_j^{-i}(\boldsymbol{m} - 1_l) \leq \xi_j^{-i}(\boldsymbol{m})$, $\forall j, l$. We choose $k = i$ in (9) and divide both sides by $\xi_l^{-i}(\boldsymbol{m})$ so that

$$\frac{\xi_l(\boldsymbol{m})}{\xi_l^{-i}(\boldsymbol{m})} = \frac{1 + \sum_{\substack{j=1 \\ j \neq l}}^{h} \gamma_{ij} \xi_j^{-i}(\boldsymbol{m} - 1_l) + \gamma_{il} \xi_l^{-i}(\boldsymbol{m} - 1_l)}{1 + \sum_{\substack{j=1 \\ j \neq l}}^{h} \gamma_{ij} \xi_j^{-i}(\boldsymbol{m}) + \gamma_{il} \xi_l^{-i}(\boldsymbol{m})}$$

since $I\{n > m\} = 1$ and by (7) we note that $\xi_j^{-i}(\boldsymbol{m})/\xi_l^{-i}(\boldsymbol{m}) = E_j(\boldsymbol{m} - 1_l)/E_i(\boldsymbol{m} - 1_l) = \xi_j^{-i}(\boldsymbol{m} - 1_l)/\xi_l^{-i}(\boldsymbol{m} - 1_j)$. If we compare term-by-term numerator and denominator in the above expression, by the induction hypothesis we obtain the upper bound $\xi_l(\boldsymbol{m}) \leq \xi_l^{-i}(\boldsymbol{m})$. Plugging (7) in the last inequality, we get $E(\boldsymbol{m} - 1_l)E_i(\boldsymbol{m}) \leq E_i(\boldsymbol{m} - 1_l)E(\boldsymbol{m})$. Diving both sides by $E(\boldsymbol{m} - 1_l)E(\boldsymbol{m})$, by (4) we get $\pi_i(\boldsymbol{m}) \geq \pi_i(\boldsymbol{m} - 1_l)$. The last relationship implies in particular $\pi_i(\boldsymbol{m} - 1_j) \geq \pi_i(\boldsymbol{m} - 1_l - 1_j)$, thus by (11) we find the other bound $\xi_j(\boldsymbol{m} - 1_l) \leq m_j \left(\sum_{i=1}^{n} \gamma_{il}(1 - \pi_i(\boldsymbol{m} - 1_j))\right)^{-1} = \xi_j(\boldsymbol{m})$.

G. Proof of Lemma 2

We give the proof for $\pi_i(\boldsymbol{m}) = 1 - \pi_{i0}(\boldsymbol{m})$ by showing that $\pi_i(\boldsymbol{m}) \geq \pi_i(\boldsymbol{m} - 1_l)$, for any l. By (4), we need show that $E(\boldsymbol{m} - 1_l)E_i(\boldsymbol{m}) \leq E_i(\boldsymbol{m} - 1_l)E(\boldsymbol{m})$. Plugging (6) in $E(\boldsymbol{m} - 1_l)$ and $E(\boldsymbol{m})$, with the choice $k = i$, after simplifications it is sufficient to show that $E_i(\boldsymbol{m} - 1_l - 1_j)E_i(\boldsymbol{m}) \leq E_i(\boldsymbol{m} - 1_l)E_i(\boldsymbol{m} - 1_j)$, for all $j \neq l$. Dividing by $E_i(\boldsymbol{m})E_i(\boldsymbol{m} - 1_j)$ this may be seen as stating $\xi_j^{-i}(\boldsymbol{m} - 1_l) \leq \xi_j^{-i}(\boldsymbol{m})$, which holds after applying Lemma 1 to a model without item i.

H. Proof of Theorem 4

We obtain the bound by summing (12) over all l, and then by Lemma 2 replacing $\pi_i(\boldsymbol{m} - 1_l)$ with its upper bound $\pi_i(\boldsymbol{m})$. This yields $\pi_i(\boldsymbol{m}) \geq \sum_{l=1}^{h} \gamma_{il} \xi_l(\boldsymbol{m})(1 - \pi_i(\boldsymbol{m}))$. The proof follows after solving for $\pi_i(\boldsymbol{m})$ and using the result to compute $\pi_{i0}(\boldsymbol{m})$.