Caching or Pre-fetching? The Role of Hazard Rates.

Andres Ferragut
Universidad ORT Uruguay
Montevideo, Uruguay
ferragut@ort.edu.uy

Matías Carrasco
Universidad ORT Uruguay
Montevideo, Uruguay
carrasco_m@ort.edu.uy

Fernando Paganini
Universidad ORT Uruguay
Montevideo, Uruguay
paganini@ort.edu.uy

***Abstract*—Local memory systems play a crucial role in today's networks: keeping popular content close to users improves performance by reducing the latency of fetching an item from a more costly central location. Caching policies that retain recently requested items are effective to deal with *bursts* of requests; in particular timer-based (TTL) caching policies are of this nature, and have well understood properties. However, in some scenarios, traffic is more *regular*, reflected in the fact that the hazard rate function of inter-request times is *increasing*. For this situation we propose the strategy of *Timer-based Pre-fetching*, a dual of TTL caching. We characterize the optimal Pre-fetching timers as the solution to a convex optimization problem, showing this approach improves upon caching strategies. We also analyze the large scale behavior of the optimal policy for both cases, which amounts to threshold policy in the hazard rates, and give asymptotic performance results for a general class of arrival processes.**

***Index Terms*—Caching, Pre-Fetching, Hazard Rate Function.**

I. Introduction

Local data storage or *caching* is a pervasive feature of computer systems: local caching of instructions at processors, texture caching in graphical processing, disk caching for fast data retrieval in hard disk storage, content caching in web applications and content delivery networks, cloud storage gateways keeping readily available items stored in cloud data centers. The adequate management of such local memory is a determining factor in performance; this issue is receiving increased recent attention with the emergence of cloud and edge computing architectures.

A *local memory* may store a certain number of items locally and temporarily, out of a (typically large) catalog of size N. For simplicity, we assume that items are homogeneous in size. The main goal is to select the subset of items that are more likely to be requested next. The key performance metric to maximize is the number of *hits*, i.e. the number of times that a request can be served directly from the local memory eliminating the need of a costly retrieval from a central location at request time. All the aforementioned applications can be subsumed into this basic system.

The analysis of local memory management policies has evolved around two main lines of research: the first one centered on *eviction based* policies, where the local memory system has a fixed capacity $C < N$, and less requested items must be evicted from memory to make room for popular content. Classical policies include the Least-Frequently-Used (LFU) policy, that evicts items based on ranking empirical request frequencies, and the Least-Recently-Used (LRU) policy, that keeps in memory the more recent requests. The analysis on these policies goes back to [1], whereas subsequent interesting approaches can be found in [2]–[6], as well as network generalizations such as [7].

A second line of research, introduced in the seminal paper [8], concerns *timer based* or Time-to-live (TTL) policies, a method widely used on the Internet: each requested item is kept in local memory for a given amount of time. Such timers must be designed for an average memory occupation of C, which works now as a soft constraint. The key insight in [8] is that this approach decouples the analysis over the arrival streams, as we shall see below. This sparked a lot of attention into TTL policies, as in [9], [10]. Moreover, a connection between TTL and eviction policies was established in [11], and further justified in [12].

Building upon this work, in [13], [14] the optimal TTL caching timers were characterized under very general hypotheses for the request processes. The key result is that the optimal policy depends on the *hazard rate* function of the inter-request times. Under a decreasing hazard rate (DHR) assumption, a convex optimization problem can be formulated to compute the optimal timers. Furthermore, suitable fluid limits for large scale systems are derived, yielding explicit expressions for the hit probability. However, when hazard rates are *increasing* (IHR), the optimal timer policy degenerates in a static policy that stores the most popular items at all times [14], just as in the case of memoryless (Poisson) traffic.

The question arises whether the performance in the IHR case could be improved by exploiting traffic regularity. In this work, we develop a new policy which we initially proposed in [15], for this situation: timer-based *pre-fetching*, i.e. speculatively retrieving the content, in anticipation of future arrivals. We derive the optimal pre-fetching timers as the solution to a proper convex optimization problem, remarkably similar in form to the one used in [13], and show that we can greatly improve the hit probability for the IHR case. Our policy is also closely related to recent results in [16] for eviction policies, where the optimal replacement is linked to the *stochastic intensity* of the incoming requests.

Our analysis leads naturally to a duality result where both timer-based caching and timer-based pre-fetching can be cast as *threshold policies* for the hazard rates. This fact enables us to compute tractable asymptotic limits for the optimal hit/miss

This work was partially supported by AFOSR-US under grant #FA9550-23-1-0350

rate of both policies, for a large scale regime.

The paper is organized as follows: in Section II we formulate our model and review the main results on TTL caching. We then introduce in Section III our new timer based prefetching policy and compute its optimal timers. We explore the duality between both policies in Section IV and compute asymptotic performance limits in V. To illustrate the results, we analyze some parametric examples in VI. Conclusions are given in Section VII.

II. Timer based caching

Consider a *local memory* system, where requests from a *catalog* of N (equally sized) items are received. The cache has limited memory, and thus aims to locally keep available a subset of size $C < N$, which can then be served with lower latency. The natural objective is to maximize the *hit probability* by choosing the appropriate items to store.

Following [8], we model requests for item i as a stationary point process $\{\tau_k^{(i)}\}$ in $\mathbb{R}$ [17], with mean intensity $\lambda_i > 0$ (average requests per time unit). We follow the usual labelling convention that $\tau_0^{(i)}$ is the first point to the left of time $t = 0$.

The total intensity of requests is $\lambda^N := \sum_{i=1}^N \lambda_i$, and $p_i := \lambda_i/\lambda^N$ is the probability that a given request is for item i, i.e. its relative popularity in a mean sense. If these popularities are known, a basic local memory management strategy is the following:

Definition 1: The *static policy* is to store at all times the C most popular items, i.e. items i with the C largest λ_i's.

While natural, the above policy need not be optimal in a real time setting, since the short-term behavior of the request process may deviate from the mean. These questions are naturally cast within the theory of stationary point processes in the real line. In this regard, introduce two main distributions (we drop the superscript i to ease the notation when talking about a single process): the inter-arrival distribution $F_0(t)$, i.e. the distribution of $\tau_{k+1} - \tau_k$ for a typical interval; its average is $1/\lambda$. These times are *synchronized* with the process. Instead, when the same process is viewed from a fixed reference point in time (e.g. 0 due to stationarity), the random variable measuring the time since the last request follows the *age distribution* [17]:[1]

$$F(t) := P\left(-\tau_0 \leqslant t\right) = \lambda \int_0^t 1 - F_0(s)\, ds. \tag{1}$$

Moreover, the *time to next request* τ_1 also follows the same distribution, and this is why F is also named the *residual lifetime distribution* associated to F_0. An example of this sampling effect is shown in Fig. 1.

The crucial magnitude in our upcoming analysis is the *hazard rate* function (also known as failure rate). If F_0 has density f_0, its hazard rate is defined as:

$$\eta(t) := \frac{f_0(t)}{1 - F_0(t)}, \tag{2}$$

and serves as a local measure of the likelihood that the current interval is exactly of length t, given that the elapsed time of the interval is at least t. We will focus on monotonic hazard functions $\eta(t)$. Decreasing hazard rates (DHR) correspond to *bursty* requests, whereas increasing hazard rates (IHR) indicate a more *regular* pattern of item requests. Hazard rates are constant (CHR) for a memoryless (Poisson) process.

Definition 2: A timer based (TTL) *caching* policy is specified as follows: upon arrival of a request for item i, the item is stored in memory (if not already present) and a timer of length T_i is started (or reset). When the timer expires, the content is removed (eviction).

See Figure 2 for an illustration. This method subsumes the static storage policy: just choose $T_i = \infty$ for the stored files, $T_i = 0$ for the rest. It has also been shown that the classical LRU policy may be approximated by a TTL policy with a common timer, the so-called "Che approximation" [11].

In [13], [14], the authors characterize the *optimal* choice of TTL timers from the point of view of hit probability, as a function of the inter-arrival and age distributions $F_0^{(i)}$ and $F^{(i)}$. The key observations are: on one hand, the hit probability of an item is given by $F_0^{(i)}(T_i)$, i.e. the probability that the next arrival comes before the timer expires. On the other hand, item i occupies memory at time 0 if and only if the *age* of the current interval is less than eviction time; the expected memory occupation is thus $F^{(i)}(T_i)$. Therefore, timer selection can be posed as the following optimization:

Problem 1 (Optimal TTL caching):

$$\max_{T_i \geqslant 0} \sum_{i=1}^N \lambda_i F_0^{(i)}(T_i) \tag{3a}$$

$$\text{subject to:} \quad \sum_{i=1}^N F^{(i)}(T_i) \leqslant C. \tag{3b}$$

The objective (3a) is just the total hit-rate of the system, and the constraint (3b) states that the average memory occupation is less than the allocated memory. In [13], [14] the following result is proven using tools of convex optimization:

Theorem 1 (Optimal TTL caching policy, [13]):

- For DHR, there exists a hazard rate threshold θ^* that characterizes the optimal timers T_i^*, through one of the alternatives:

$$\eta^{(i)}(T_i^*) = \theta^*, \text{ and } \; 0 < T_i^* < \infty;$$
$$\eta^{(i)}(0) \leqslant \theta^*, \text{ and } \; T_i^* = 0;$$
$$\eta^{(i)}(\infty) \geqslant \theta^*, \text{ and } \; T_i^* = \infty.$$

- For CHR o IHR, the static policy is optimal.

The stated dichotomy is consistent with intuition: if requests come in bursts, clustered in time, dynamic caching can provide benefits. If requests are purely random or even more regular, caching recent items cannot improve over the static policy.

For the bursty case, a further conclusion is that the performance of LRU (which corresponds approximately to $T_i \equiv T$ satisfying (3b) with equality) can be surpassed by using differentiated timers, characterized through the hazard rate.

[1] The preceding arguments can be formalized properly using Palm theory for Point Processes. In our case, we avoid going into details of this formalization and refer the reader to [17] for a full discussion.

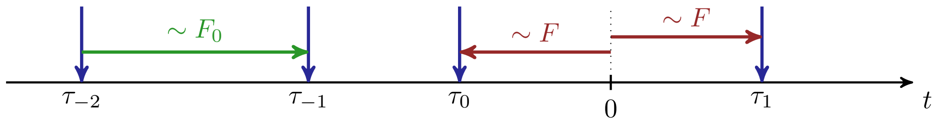

Fig. 1. Inter-arrival (F_0) and age (F) distributions showing the sampling bias.

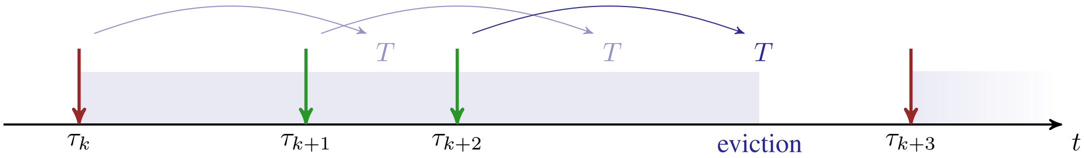

Fig. 2. TTL caching policy for a single item.

Is there an alternative method to improve over the static policy for requests with increasing hazard rates? A positive answer is given hereafter.

III. Timer based pre-fetching

The key insight for our new policy is that, if requests follow a more regular pattern, such as having increasing hazard rates, the likelihood of a subsequent request for item i *decreases* immediately upon seeing a request. Therefore, removing this item from memory and only retrieving it at a later time may improve performance. We now make this precise.

Definition 3: The timer based *prefetching* policy is specified as follows: after a request for item i, it is *removed* from memory if already present, and a timer T_i is started. At timer expiration, the item is *fetched again* and stored in memory. If a new request arrives before this, it is a *miss* and the timer is reset. Otherwise, the item will have been *pre-fetched* for the next arrival, so there is a *hit*.

The policy is illustrated in Fig. 3. It also covers static policies, where now $T_i = 0$ corresponds to storing the item permanently in the local memory, and $T_i = \infty$ not storing it.

As in the TTL caching case, the analysis of the hit probability decouples among the processes here as well. The steady state hit-probability of item i for the pre-fetching policy can be readily computed by observing that:

$$P(\text{item } i \text{ hit}) = 1 - F_0^{(i)}(T_i),$$

that is the probability that the *next* arrival occurs *after* T_i expires. Also, the steady state average occupation can be computed by observing that item i is stored at time $t = 0$ if and only if its last request before $t = 0$ was more than T_i units of time before, i.e. the age of the current interval is longer than T_i:

$$P(\text{item } i \text{ in memory}) = 1 - F^{(i)}(T_i),$$

with $F^{(i)}$ defined in (1). Note that the average memory occupation is therefore:

$$E\left[\sum_{i=1}^{N} \mathbf{1}_{\left\{-\tau_0^{(i)} > T_i\right\}}\right] = \sum_{i=1}^{N} \left(1 - F^{(i)}(T_i)\right).$$

We can now formulate the optimal timer problem of the pre-fetching policy:

Problem 2 (Optimal timer-based pre-fetching):

$$\max_{T_i \geqslant 0} \sum_{i=1}^{N} \lambda_i \left(1 - F_0^{(i)}(T_i)\right)$$
$$\text{subject to: } \sum_{i=1}^{N} \left(1 - F^{(i)}(T_i)\right) \leqslant C.$$

Equivalently, by getting rid of constant terms:

$$\min_{T_i \geqslant 0} \sum_{i=1}^{N} \lambda_i F_0^{(i)}(T_i), \tag{4a}$$
$$\text{subject to: } \sum_{i=1}^{N} F^{(i)}(T_i) \geqslant N - C. \tag{4b}$$

The above is closely related to Problem 1 above. We are now minimizing the *miss* rate, subject to the number of *non-stored* items being larger than $N - C$ on average. Using the fact that the $F^{(i)}$ are increasing, consider the change of variables $u_i = F^{(i)}(T_i)$; here $u_i \in [0, 1]$ is the probability of *not* being stored. Problem 2 can be rewritten as:

$$\min_{u_i \in [0,1]} \sum_{i=1}^{N} \lambda_i F_0^{(i)} \circ \left(F^{(i)}\right)^{-1} (u_i), \tag{5a}$$
$$\text{subject to: } \sum_{i=1}^{N} u_i \geqslant N - C. \tag{5b}$$

We will use this version to analyze the problem under different scenarios for the hazard rate of the request processes.

A. Increasing hazard rates

Theorem 2: Suppose that the distributions $F_0^{(i)}$ satisfy the IHR property. Then, there exists a threshold $\theta^* \geqslant 0$ such that the optimal timers T_i^* satisfy one of the alternatives:

$$\eta^{(i)}(T_i^*) = \theta^*, \text{ and } 0 \leqslant T_i^* < \infty;$$
$$\eta^{(i)}(0) > \theta^*, \text{ and } T_i^* = 0;$$
$$\eta^{(i)}(\infty) \leqslant \theta^*, \text{ and } T_i^* = \infty.$$

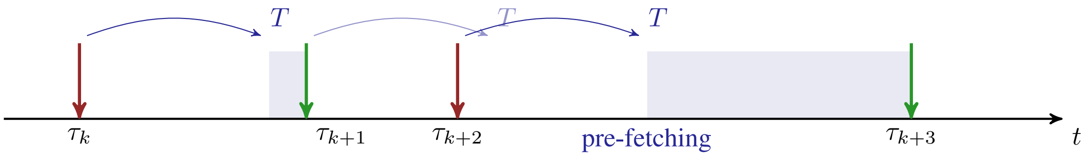

Fig. 3. Timer pre-fetching policy for a single item.

Proof: Let us compute the gradient of the objective function using eq. (1) and the inverse function theorem:

$$\begin{aligned}\frac{\partial}{\partial u_i}\lambda_i F_0^{(i)} \circ \left(F^{(i)}\right)^{-1}(u_i) &= \frac{\lambda_i f_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right)}{\lambda_i\left(1-F_0^{(i)}((F^{(i)})^{-1}(u_i))\right)} \\ &= \eta^{(i)}\left(\left(F^{(i)}\right)^{-1}(u_i)\right)\end{aligned} \tag{6}$$

with $\eta^{(i)}$ as in (2).

From (6), if the $\eta^{(i)}$ are increasing, the objective function in (5a) is convex, and thus (5) is a proper convex optimization problem. Introduce the Lagrangian with multiplier $\theta \geqslant 0$ applied to the constraint (5b):

$$\begin{aligned}\mathcal{L}(u,\theta) &= \sum_{i=1}^{N}\lambda_i F_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right) + \theta\left(N - C - \sum_{i=1}^{N}u_i\right) \\ &= \sum_{i=1}^{N}\left[\lambda_i F_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right) - \theta u_i\right] + \theta(N-C);\end{aligned}$$

we know from convex duality that there exists a saddle point (u^*, θ^*). In particular, for the dual optimal θ^* we have:

$$u_i^* \in \arg\min_{u_i\in[0,1]}\left[\lambda_i F_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right) - \theta^* u_i\right],$$

a *decoupled* condition over the items i. To solve for the above minimum, note that by (6), the derivative of the objective is $\eta^{(i)}((F^{(i)})^{-1}(u_i)) - \theta^*$, which increasing by hypothesis. We have the following cases:

- If $\eta^{(i)}((F^{(i)})^{-1}(0)) = \eta^{(i)}(0) > \theta^*$, the derivative is always positive, so the optimum is attained only for $u_i^* = T_i^* = 0$; the content must be always stored.
- If instead, $\eta^{(i)}((F^{(i)})^{-1}(1)) \leqslant \theta^*$, the derivative is always non-positive, the optimum is attained for $u_i^* = 1$ with the item never stored.
- In the remaining case, there exists $u_i^* \in [0,1)$ where
$$\eta^{(i)}((F^{(i)})^{-1}(u_i^*)) = \theta^*; \tag{7}$$
the item must be prefetched at $T_i^* = (F^{(i)})^{-1}(u_i^*)$.

■

Remark 1: It is in principle possible to have $\theta^* = 0$, indicating that the caching constraint is not binding at optimality. Examples of this kind appear when the inter-arrival distribution is supported at a positive distance from 0, so the hazard rate remains at zero until $T_0 > 0$. For instance, a uniform distribution in $[T_0, T_1]$, which has IHR. In such cases, pre-fetching an item at T_0 ensures a hit, with an impact on memory occupation which is less than unity.

For simplicity, we will focus henceforth on the case $\theta^* > 0$. In this case, from the *complementary slackness* condition in convex duality, the cache constraint must be at equality:

$$\sum_{i=1}^{N}u_i^* = \sum_{i=1}^{N}F^{(i)}(T_i^*(\theta^*)) = N - C. \tag{8}$$

Theorem 2 shows that, under the IHR property, the optimal policy is again a *threshold* policy: there exists a threshold θ^* such that an item is stored in the local memory if and only if its *current hazard rate* is greater than the threshold. The items with $\eta^{(i)}(0) \geqslant \theta^*$ are always stored, the items with $\eta^{(i)}(\infty) \leqslant \theta^*$ are never stored, and the remaining items are pre-fetched after a time T_i^* since the last request, when their hazard rates reach the threshold. The underlying idea being that the hazard rate is a measure of the current likelihood of getting a request, and thus the *marginal utility* of storing some object in the local memory with a fixed budget C.

Recall from Theorem 1 that in the case of IHR, the optimal *caching* policy was the static one. Since this possibility is also covered by pre-fetching, but not optimal in general, we conclude that pre-fetching improves upon caching for IHR.

B. Constant and decreasing hazard rates

For constant hazard rates, the arrivals become Poisson and the change of variables turn eqs. (4) into a linear program, since $F_0 \equiv F$. It is easy to see that in this case the optimal pre-fetching policy is just the static one.

The same conclusion holds for the DHR case. This result can be proved with analogous arguments to [14, Theorem 1]; basically we reduce the problem to minimizing a *concave* function over a simplex, and thus the optimum should be at a vertex of the feasible region:

Theorem 3: Provided that the distributions $F_0^{(i)}$ satisfy the DHR property, the optimal timer based pre-fetching policy is to statically store the C most popular contents.

The result of Theorem 3 is expected in light of the discussion of Section II: when arrivals have the DHR property, traffic is bursty, so the strategy of initially removing and later pre-fetching is not helpful. *Caching* makes more sense for DHR and *pre-fetching* for IHR requests.

IV. A Tale of Two Policies

The preceding discussion highlights that the underlying characteristics of the traffic determine which policy, caching or pre-fetching, will work best. Moreover, from the formulation of Problems 1 and 2 it is clear that a strong connection exists between both policies. This connection is better understood

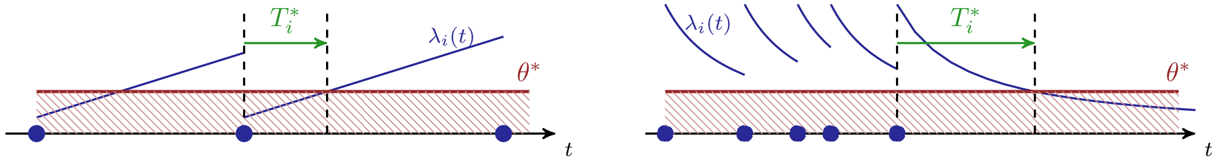

Fig. 4. Pre-fetching and caching as threshold policies for the hazard rates.

from the depiction in Fig. 4. When hazard rates are monotone, either decreasing or increasing, the optimal policy is defined by *threshold* on the *stochastic intensity* $\lambda_i(t)$ of the request process, a local measure of the likelihood of an arrival. In this renewal case stochastic intensity is given by the hazard rate function measured since the last arrival:

$$\lambda_i(t) := \eta^{(i)}(t - \tau_i^*(t)), \tag{9}$$

where $\tau_i^*(t) = \sup\{\tau_k^{(i)} : \tau_k^{(i)} < t\}$ is the last point before t of process i, i.e. $t - \tau_i^*(t)$ is the current interval age for the $i-$th process.

For the IHR case, this translates into waiting for some time *until* the likelihood of an arrival is above the threshold, and then pre-fetch the item. For the DHR case, it implies keeping the item in memory, because an arrival *increases* the likelihood of future requests, since it resets the hazard rate to its maximum value. After the hazard rate crosses below the threshold, the item can be evicted from memory.

V. Large scale asymptotics

In order to better understand the performance of the system in a large scale limit, we now derive a suitable fluid scaling where the catalog size $N \to \infty$. In order to do so, we have to incorporate a little more structure into the problem. We begin by making the following:

Assumption 1: The request processes are independent, and their inter-arrival time distributions come from a common scale family, i.e.

$$F_0^{(i)} = F_0(\lambda_i t),$$

where the base distribution function $F_0(t)$ has density f_0 and unit mean. Without loss of generality we will assume that the intensities are in decreasing order, i.e. $\lambda_1 \geqslant \cdots \geqslant \lambda_N$.

In particular, the $i-$th process has intensity λ_i, and applying the definitions (1) and (2), it is easy to show that the following equalities hold:

$$F^{(i)}(t) = P\left(-\tau_0^{(i)} \leqslant t\right) = F(\lambda_i t); \tag{10a}$$

$$\eta^{(i)}(t) = \frac{f^{(i)}(t)}{1 - F_0^{(i)}(t)} = \lambda_i \eta(\lambda_i t). \tag{10b}$$

Let us now define the *observed hazard rate* random variable, which is the hazard rate of the current interval, sampled at time 0. More formally:

$$X^{(i)} = \eta^{(i)}(-\tau_0^{(i)}). \tag{11}$$

For the base distribution F_0, we can compute the distribution of this random variable X as:

$$\begin{aligned} G(x) &:= P(\eta(-\tau_0) \leqslant x) = P\left(-\tau_0 \in \eta^{-1}([0,x])\right) \\ &= \int_{\eta^{-1}([0,x])} F(dx), \end{aligned} \tag{12}$$

since $-\tau_0 \sim F$.

By resorting to eqs. (10), we have the corresponding scaling property for $G^{(i)}$:

$$\begin{aligned} G^{(i)}(x) &:= P(X^{(i)} \leqslant x) = P\left(\eta^{(i)}\left(-\tau_0^{(i)}\right) \leqslant x\right) \\ &= P\left(\eta\left(-\lambda_i \tau_0^{(i)}\right) \leqslant x/\lambda_i\right). \end{aligned}$$

Due to the scaling, $-\lambda_i \tau_0^{(i)} \sim F$, the age distribution of the base process, thus we get:

$$G^{(i)}(x) = G(x/\lambda_i). \tag{13}$$

To build on our analysis of timer-based policies in the preceding sections, we make a final assumption:

Assumption 2: The hazard rate function η associated to F_0 is continuous and strictly monotone.

In this case, the set $\eta^{-1}([0,x])$ in (12) will be an interval. Since F is a continuous distribution, we have that G is also continuous.

In what follows, for concreteness we will focus on the IHR case and the pre-fetching policy, but the analysis extends analogously to the DHR/caching case.

A. Scaling the family of arrival rates

We will now construct a series of systems, indexed by N, where each system has N arrival streams or, in other words, items in its catalog. Denote by $\{\lambda_i^{(N)}\}$ the arrival rates of the system of size N, with the above convention that $\lambda_1^{(N)} \geqslant \ldots \lambda_N^{(N)} > 0$. For each N, this set of intensities can be interpreted as a discrete distribution on the positive real line $\lambda > 0$, with cumulative distribution function:

$$L_N(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\left\{\lambda_i^{(N)} \leqslant \lambda\right\}}. \tag{14}$$

Our large-scale limit theorems are based on the assumption that as $N \to \infty$, the above family of discrete distributions of traffic intensity has a weak limit:

Assumption 3: As $N \to \infty$, the distribution $L_N \Rightarrow^w L$, a fixed distribution, where $\Rightarrow^w$ denotes usual weak convergence. L has no atoms at $\lambda = 0$.

This assumption is very general; we explore an important parametric example that satisfies it in Section VI.

B. Optimal policy asymptotics

We now let $N \to \infty$ and analyze the behavior of the optimal threshold θ^*_N of the N-th system, obtained in Theorem 2 for the IHR case.

Theorem 4: Consider a family of local memory systems, indexed by N, with request processes satisfying Assumptions 1–3, in the IHR case. Choose the memory size of the $N-$th system as $C_N = cN$, with $0 \leqslant c \leqslant 1$ being the fraction of the catalog that the system is able to store. Define the function:

$$G_\infty(\theta) := \int_0^\infty G(\theta/\lambda)L(d\lambda). \tag{15}$$

If there exists a unique solution to θ^* satisfying:

$$G_\infty(\theta^*) = 1 - c, \tag{16}$$

then the sequence of hazard rate thresholds θ^*_N defined by (8) for the N-th system verifies:

$$\theta_N \underset{N\to\infty}{\longrightarrow} \theta^*.$$

Proof: We begin by rewriting the memory constraint equation (8) for the optimal timers of the N-th system:

$$\sum_{i=1}^N F^{(i)}(T_i^*(\theta^*_N)) = N - C_N.$$

Equivalently, by using that $C_N = cN$ and dividing by N:

$$\frac{1}{N}\sum_{i=1}^N F^{(i)}(T_i^*(\theta^*_N)) = 1 - c. \tag{17}$$

Now $F^{(i)}(T_i^*(\theta^*_N)) = P(-\tau_0^{(i)} \leqslant T_i^*(\theta^*_N))$. Using the optimality condition (7) for the interior case $0 < T_i^* < \infty$, we know that $\eta^{(i)}(T_i^*(\theta^*_N)) = \theta^*_N$. Therefore, applying the monotonically increasing transformation $\eta^{(i)}$ to both sides and using the definition of $G^{(i)}$ (12) we obtain:

$$\begin{aligned} F^{(i)}(T_i^*(\theta^*_N)) &= P(-\tau_0^{(i)} \leqslant T_i^*(\theta^*_N)) \\ &= P(\eta^{(i)}(-\tau_0^{(i)}) \leqslant \theta^*_N) \\ &= G^{(i)}(\theta^*_N). \end{aligned}$$

Substituting in the left-hand side of (17), and resorting to the scaling property (13) we get:

$$\frac{1}{N}\sum_{i=1}^N F^{(i)}(T_i^*(\theta^*_N)) = \frac{1}{N}\sum_{i=1}^N G\left(\frac{\theta^*_N}{\lambda_i^{(N)}}\right).$$

We can now rewrite (17) in terms of the distribution L_N as:

$$\int_0^\infty G(\theta^*_N/\lambda)L_N(d\lambda) = 1 - c. \tag{18}$$

Consider the function

$$G_N(\theta) := \int_0^\infty G(\theta/\lambda)L_N(d\lambda);$$

since by the Assumptions, $G(\cdot)$ is a continuous and also bounded as a distribution function, the weak convergence $L_N \Rightarrow^w L$ implies that $G_N(\theta)$ converges to $G_\infty(\theta)$, pointwise at each θ.

Observe that $G_N(\theta)$ and $G_\infty(\theta)$ are themselves distribution functions in θ, so we can say that $G_N \Rightarrow^w G_\infty$, which implies (see e.g. Lemma [18, Lemma 21.2]) the pointwise convergence of quantiles of G_N to those of G_∞, at points where the latter are well defined.

By hypothesis G_∞ has well-defined $(1-c)$-quantile θ^*. Noting from (18) that θ^*_N is the $(1-c)$-quantile of G_N, we have the desired result. ■

C. Asymptotic performance

Theorem 4 shows that in the limit, the optimal policy behaves as a fixed threshold policy satisfying (16): for large N, a given item i will be stored in the cache if and only its hazard rate is higher than $\theta^*_N \approx \theta^*$.

Whether a certain request for an item i is a hit or a miss will thus be determined by the comparison of the threshold with the value of the hazard rate *just prior* to the request. This magnitude is *synchronized* with requests and must be computed in terms of the inter-arrival distribution. We now exploit this remark to obtain an asymptotic performance result for the miss probability of the system.

Introduce the *observed hazard rate upon arrival* random variable, $X_0^{(i)} = \eta_i(\tau_1^{(i)} - \tau_0^{(i)})$, i.e. the composition of the hazard rate function with the inter-arrival times. Its distribution for the base process can be computed as follows:

$$\begin{aligned} G_0(x) &:= P(X_0 \leqslant x) = P(\eta(\tau_1 - \tau_0) \leqslant x) \\ &= P((\tau_1 - \tau_0) \in \eta^{-1}([0,x]) = \int_{\eta^{-1}([0,x])} F_0(dt). \end{aligned} \tag{19}$$

In the IHR case we can further write $G_0(x) = F_0(\eta^{-1}(x))$.

A basic inequality for this distribution follows from the definition of the hazard rate:

$$\begin{aligned} G_0(x) &= \int_{\eta^{-1}([0,x])} f_0(t)dt = \int_{\{t:\eta(t)\leqslant x\}} \eta(t)(1-F_0(t))dt \\ &\leq x\int_{\mathbb{R}}(1-F_0(t))dt = xE[\tau_1 - \tau_0] = x. \end{aligned} \tag{20}$$

Using the scaling properties (10), we can derive the following properties for the i-th process, in the IHR case:

$$G_0^{(i)}(x) = F_0^{(i)}\left((\eta^{(i)})^{-1}(x)\right) = G_0(x/\lambda_i). \tag{21}$$

Our asymptotic performance result is now stated. It requires a stronger condition on the scaling.

Assumption 4: The family of measures L_N is uniformly integrable.

Theorem 5: Consider a family of local memory systems as before, under Assumptions 1–4. Let $\mathcal{P}_N$ denote the miss probability for system N. Then:

$$\mathcal{P}_N \underset{N\to\infty}{\longrightarrow} \frac{\int_0^\infty \lambda G_0(\theta^*/\lambda)L(d\lambda)}{\int_0^\infty \lambda L(d\lambda)}, \tag{22}$$

where θ^* is defined by eq. (16).

Proof: Denote by M_N the miss *rate* in the $N-$th system. It is given by the optimum cost in (4a):

$$M_N = \sum_{i=1}^{N} \lambda_i^{(N)} F_0^{(i)}(T_i^*).$$

Substituting the optimal timers T_i^* from Theorem 2 we have

$$M_N = \sum_{i=1}^{N} \lambda_i^{(N)} F_0^{(i)} \left((\eta^{(i)})^{-1}(\theta_N^*) \right) = \sum_{i=1}^{N} \lambda_i^{(N)} G_0 \left(\frac{\theta_N^*}{\lambda_i^{(N)}} \right),$$

where we have used (21). Invoking the distribution L_N we write:

$$\frac{M_N}{N} = \int_0^\infty \lambda G_0(\theta_N^*/\lambda) L_N(d\lambda).$$

Due to the bound (20), and the convergence of the sequence θ_N^*, the integrand above is uniformly bounded for all N. This, together with weak convergence yields the limit

$$\frac{M_N}{N} \underset{N\to\infty}{\longrightarrow} \int_0^\infty \lambda G_0(\theta^*/\lambda) L(d\lambda). \tag{23}$$

Also, note that the total rate $\lambda^N := \sum_{i=1}^N \lambda_i^{(N)}$ satisfies

$$\frac{\lambda^N}{N} := \int_0^\infty \lambda L_N(d\lambda) \underset{N\to\infty}{\longrightarrow} \int_0^\infty \lambda L(d\lambda); \tag{24}$$

here (only) we have invoked uniform integrability.

The miss probability of system N is given by $\mathcal{P}_N = M_N/\lambda^N$; its limit follows from (23) and (24). ■

VI. Parametric examples and simulations

In this Section we describe some examples using the above results. We begin with an important parametric family for the popularity distribution: the generalized Zipf distribution, commonly used in this setting [11]. In this case, the popularity of item i is proportional to $i^{-\beta}$ where $\beta \geqslant 0$ is known as the *tail* parameter of the Zipf random variable. Values of $\beta \in [0, 1]$ correspond to heavy tailed popularities. We now show how to incorporate this model in a way that satisfies Assumption 3.

Example 1 (Zipf popularities scaling): Assume that the $N-$th system has the following arrival rates:

$$\lambda_i^{(N)} = \left(\frac{N}{i} \right)^\beta$$

with $\beta \geq 0$ the tail parameter of the Zipf law. Note that, under this scaling, the less popular object has intensity 1 for all N. Now, for any $\lambda > 1$, we have:

$$\begin{aligned} 1 - L_N(\lambda) &= \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\left\{ \left(\frac{N}{i} \right)^\beta > \lambda \right\}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\left\{ i < \frac{N}{\lambda^{1/\beta}} \right\}} \\ &= \frac{1}{N} \left\lfloor \frac{N}{\lambda^{1/\beta}} \right\rfloor \underset{N\to\infty}{\longrightarrow} \lambda^{-1/\beta}. \end{aligned}$$

Therefore, since the above convergence is pointwise and the limit is continuous, $L_N(\lambda) \Rightarrow^w L(\lambda)$ given by:

$$L(\lambda) = 1 - \lambda^{-1/\beta} \quad \text{for } \lambda \geq 1. \tag{25}$$

In the limit the popularities follow a standard Pareto distribution with tail parameter $1/\beta$. If $\beta \geqslant 1$, i.e. the popularities decay fast, L does not have finite mean, resulting from some objects being extremely most popular than others. If instead $0 < \beta < 1$, where popularities are more homogeneous, L has finite mean $1/(1-\beta)$. For $\beta = 0$, the system degenerates into every object having the same popularity, and thus L is the step function at $\lambda = 1$.

The total arrival rate into the $N-$th system satisfies:

$$\lambda^N = \sum_{i=1}^{N} \lambda_i^{(N)} = N^\beta \sum_{i=1}^{N} \frac{1}{i^\beta} =: N^\beta S_N(\beta),$$

where $S_N(\beta)$ is the generalized harmonic series partial sum. Using the well known equivalents for this series, we have that:

$$\lambda^N = \begin{cases} O(N^\beta) & \text{if } \beta > 1, \\ O(N \log N) & \text{if } \beta = 1, \\ O(N) & \text{if } \beta < 1. \end{cases}$$

In particular, with our scaling, the total arrival rate $\lambda^N \to \infty$ as $N \to \infty$, albeit at different rates depending on the tail parameter β.

Example 2 (Uniform arrivals): Using the above model for traffic intensities, we now analyze a parametric model for the inter-arrival time distributions, namely a uniform distribution which has IHR. Under Assumption 1, the base distribution F_0 (of mean 1) and its associated age distribution are given by:

$$F_0(t) = \begin{cases} t/2 & 0 < t < 2 \\ 1 & t \geqslant 2 \end{cases}, \quad F(t) = \begin{cases} t - \frac{t^2}{4} & 0 < t < 2 \\ 1 & t \geqslant 2 \end{cases}, \tag{26}$$

in the positive half line. The associated hazard rate function is given by:

$$\eta(t) = \frac{1}{2-t} \quad 0 \leqslant t < 2.$$

In particular, it is continuous and strictly monotone, with range $[1/2, \infty)$. Applying eq. (12) we can compute the observed hazard rate distribution G as:

$$G(x) = 1 - \frac{1}{4x^2}, \quad x \geqslant \frac{1}{2}.$$

The above functions are depicted in Fig. 5 for reference. Finally, we can also find the distribution of observed hazard rates upon arrival, using eq. (19):

$$G_0(x) = F_0(\eta^{-1}(x)) = 1 - \frac{1}{2x}, \quad x \geqslant \frac{1}{2}.$$

Armed with the above tools, we now compute the asymptotic global observed hazard rate distribution from eq. (15):

$$\begin{aligned} G_\infty(x) &= \int_1^\infty G_0\left(\frac{x}{\lambda}\right) L(d\lambda) \\ &= \int_1^\infty G_0\left(\frac{x}{\lambda}\right) \frac{1}{\beta} \lambda^{-\frac{1}{\beta}-1} d\lambda. \end{aligned}$$

This integral can be explicitly solved, for any value of β; a representative case is depicted in Fig. 6. This function enables us to compute the threshold θ^* for any desired quantile.

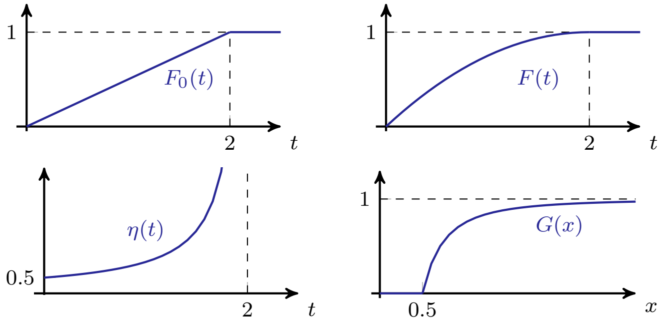

Fig. 5. Uniform inter-arrival times and associated distributions.

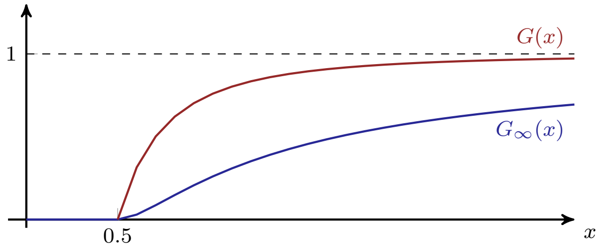

Fig. 6. Asymptotic observed hazard rate for Zipf(1/2) popularities and uniform inter-arrival times.

For $\beta < 1$, the family $\{L_N\}$ defined above is uniformly integrable, and thus (22) holds. In Fig. 7, we plot the numerically computed asymptotic behavior for the miss probability as a function of β, for a memory size $c = 0.1$ or 10% of the catalog. If $\beta \geqslant 1$ it is easy to show that $\mathcal{P}_N \to_N 0$, indeed this happens for the suboptimal static policy [13].

Also shown is the miss probability for: (i) the optimal timer-based pre-fetching policy for finite $N = 10000$, $C = 1000$ computed by explicitly solving Problem 2; (ii) the static policy (which is also the optimal TTL caching policy for this traffic); (iii) the classical LRU caching strategy. We highlight the bad performance of the latter in the case of a regular traffic pattern.

VII. Conclusions

In this paper, we analyzed the role of the hazard rate function of the inter-arrival times between requests to a local memory systems, showing how the shape of the HR crucially determines the best strategy for memory management. In particular, we extended the notion of TTL caching to timer based pre-fetching, which improves performance over well-known caching policies for more regular traffic patterns. As we can see from the example we analyzed, for these regular processes, classical caching can underperform and our new policy can drastically improve the hit probability.

Several lines of future work remain open: in particular how to estimate the timers based on previous data, and obtaining analogues of the classical caching policies that can be applied for pre-fetching.

Acknowledgments

The authors would like to thank Prof. B. Hajek for his insightful inputs in the early stages of this work.

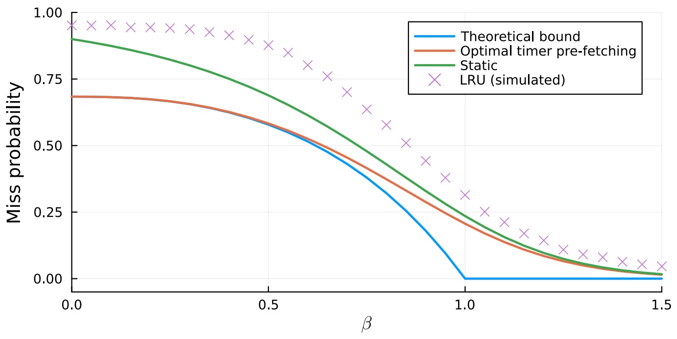

Fig. 7. Miss probability comparison for optimal timer pre-fetching, static storage and LRU caching. The theoretical bound is computed using eq. (22).

References

[1] A. Dan and D. Towsley, "An approximate analysis of the LRU and FIFO buffer replacement schemes," in *Proc. of ACM/SIGMETRICS 1990*, June 1990, pp. 143–152.

[2] P. Jelenković and A. Radovanović, "Asymptotic insensitivity of least recently used caching to statistical dependency," in *Proc. of IEEE/Infocom 2003*, Apr. 2003, pp. 438–447.

[3] P. R. Jelenković and A. Radovanović, "Least-recently-used caching with dependent requests," *Theoretical computer science*, vol. 326, no. 1, pp. 293–327, 2004.

[4] P. R. Jelenković, A. Radovanović, and M. S. Squillante, "Critical sizing of LRU caches with dependent requests," *Journal of Applied Probability*, vol. 43, no. 4, pp. 1013–1027, 2006.

[5] P. R. Jelenković and A. Radovanović, "The persistent-access-caching algorithm," *Random Structures & Algorithms*, vol. 33, no. 2, pp. 219–251, 2008.

[6] N. Gast and B. V. Houdt, "Transient and steady-state regime of a family of list-based cache replacement algorithms," in *Proc. of ACM/SIGMETRICS 2015*, Jun. 2015, pp. 123–136.

[7] S. Ioannidis and E. Yeh, "Adaptive caching networks with optimality guarantees," *IEEE/ACM transactions on networking*, vol. 26, no. 2, pp. 737–750, 2018.

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

[9] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. C. Tay, "A utility optimization approach to network cache design," in *Proc. of IEEE/Infocom 2016*, Apr. 2016, pp. 1–9.

[10] M. Dehghan, L. Massoulie, D. Towsley, D. S. Menasche, and Y. C. Tay, "A utility optimization approach to network cache design," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1013–1027, 2019.

[11] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2002.

[12] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *Proc. of the 24th International Teletraffic Congress*, 2012, pp. 57–64.

[13] A. Ferragut, I. Rodriguez, and F. Paganini, "Optimizing TTL caches under heavy tailed demands," in *Proc. of ACM/SIGMETRICS 2016*, Jun. 2016, pp. 101–112.

[14] A. Ferragut, I. Rodríguez, and F. Paganini, "Optimal timer-based caching policies for general arrival processes," *Queueing Systems*, vol. 88, no. 3–4, pp. 207–241, 2018.

[15] A. Ferragut, M. Carrasco, and F. Paganini, "Timer-based pre-fetching for increasing hazard rates," *SIGMETRICS Perform. Eval. Rev.*, vol. 52, no. 2, pp. 9–11, Sep. 2024.

[16] N. K. Panigrahy, P. Nain, G. Neglia, and D. Towsley, "A new upper bound on cache hit probability for non-anticipative caching policies," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 7, no. 2–4, November 2022.

[17] P. Brémaud, *Point process calculus in time and space*. Springer, 2020.

[18] A. W. van der Vaart, *Asymptotic statistics*, ser. Camb. Ser. Stat. Probab. Math. Cambridge: Cambridge Univ. Press, 1998, vol. 3.