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, 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, as already hinted at [15]. In this work, we propose a new policy 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 or *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 a given intensity $\lambda_i > 0$ (average requests per time unit). Their sum $\lambda_N := \sum_{i=1}^N \lambda_i$ is the total intensity of requests, and $p_i := \lambda_i/\lambda_N$ is the probability of a given request being for item i, i.e. its relative popularity. We follow the usual labelling convention that $\tau_0^{(i)}$ is the first point to the left of time $t = 0$.

For a stationary point process in the real line there are 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.

A timer based (TTL) *caching* policy works 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 example. This method subsumes static storage policies (just choose $T_i = 0$ or ∞) such as the *keep the most popular* policy (the asymptotic limit of LFU). It also provides a good approximation for the classical LRU policy, due to the "Che approximation" [11], which corresponds to choosing a single common timer for all items.

In [13], [14], the authors formulate an optimization problem to characterize the optimal timers 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, we can formulate:

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:

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

- For *decreasing hazard rates*, the problem can be shown to have a unique non-trivial solution using convexity. The solution is such that there exists a threshold $\theta^* \geqslant 0$ such that $\eta^{(i)}(T_i^*) \geqslant \theta^*$. Moreover, inequality is strict if and only if $T_i^* = \infty$, i.e. the content is always stored.
- For *constant hazard rates* (i.e. Poisson arrivals) or *increasing hazard rates*, the best caching policy is the *static* policy of keeping the C most popular objects at the cache at all times.

The above results are expected: decreasing hazard rates correspond to *bursty* traffic, where requests are clustered in time, and thus caching can provide benefits. In particular, the performance of LRU (which corresponds approximately to $T_i \equiv T$ satisfying (3b) with equality) can be surpassed by using the appropriate timers. However, if traffic is purely random (Poisson) or more *regular*, then timer based caching cannot improve over the static policy.

This leads to the following question: can we improve performance in the case of increasing hazard rates? A positive answer is given hereafter.

III. Timer based pre-fetching

We now introduce our policy: the key insight 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

[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.

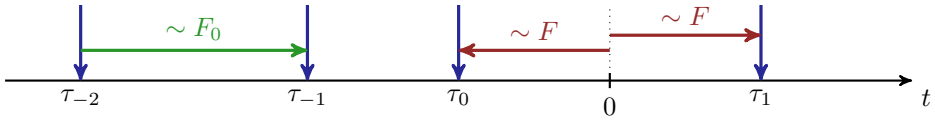

Fig. 1. Inter-arrival (F_0) and age (F) distributions showing the sampling bias.

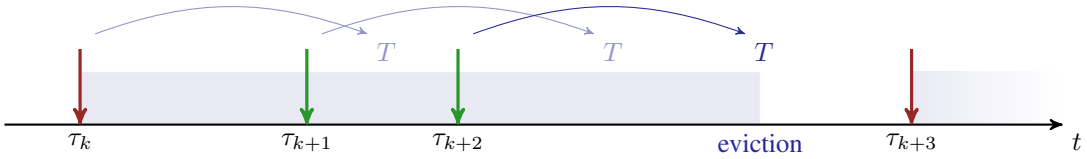

Fig. 2. TTL caching policy for a single item.

a later time may improve performance. We now make this precise.

Consider the following policy: after a request for item i, we *remove* it from memory if it was already present, and start a timer T_i. At timer expiration, we *fetch the item again* and store it on memory. If a new request arrives before this, we will have a *miss* and the timer is reset. Otherwise, the item will have been *pre-fetched* for the next arrival, and we will have a *hit*. We call our policy *timer based pre-fetching* and its behavior is depicted in Fig. 3. An important observation is that the analysis of the hit probability *decouples* among the processes, as in the case of TTL caching. The above formulation also allows static policies, where $T_i = 0$ corresponds to the item being permanently stored in the local memory.

The steady state hit-probability of item i for the aforementioned 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):

$$\max_{T_i \geqslant 0} \sum_{i=1}^{N} \lambda_i \left(1 - F_0^{(i)}(T_i)\right)$$

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

Equivalently, by getting rid of the constant terms:

$$\min_{T_i \geqslant 0} \sum_{i=1}^{N} \lambda_i F_0^{(i)}(T_i),$$

$$\text{subject to:} \quad \sum_{i=1}^{N} F^{(i)}(T_i) \geqslant N - C.$$

The above Problem is closely related to Problem 1 in [13]. We are now minimizing the *miss* rate, subject to the number of *non-stored* items being larger than $N - C$ on average. We now recast Problem 2 as a convex problem and analyze the structure of the optimal policy in different scenarios.

A. Increasing hazard rates

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. The above problem can be rewritten as:

$$\min_{u_i \in [0,1]} \sum_{i=1}^{N} \lambda_i F_0^{(i)} \circ \left(F^{(i)}\right)^{-1}(u_i), \tag{4a}$$

$$\text{subject to:} \quad \sum_{i=1}^{N} u_i \geqslant N - C. \tag{4b}$$

We are now ready to prove:

Theorem 2: Provided that the distributions $F_0^{(i)}$ satisfy the IHR property, there exists a threshold $\theta^* \geqslant 0$ and timers T_i^* such that the optimal timer based pre-fetching policy defined by Problem 2 is given by:

$$\eta^{(i)}(T_i^*) \geqslant \theta^*,$$

whenever $T_i^* < \infty$ (pre-fetching). The inequality is strict if and only if $T_i^* = 0$, i.e. the content is always stored.

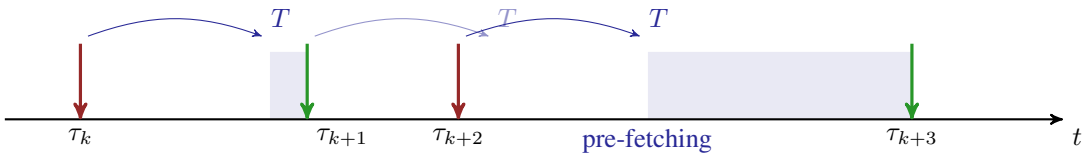

Fig. 3. Timer pre-fetching policy for a single item.

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

$$\frac{\partial}{\partial u_i}\lambda_i F_0^{(i)} \circ \left(F^{(i)}\right)^{-1}(u_i) = \frac{\lambda_i f_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right)}{\lambda_i\left(1 - F_0^{(i)}((F^{(i)})^{-1}(u_i))\right)} = \eta^{(i)}\left(\left(F^{(i)}\right)^{-1}(u_i)\right) \tag{5}$$

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

From (5), if the $\eta^{(i)}$ are increasing, the objective function in eq. (4) is convex, and thus eqs. (4) define a proper convex optimization program. In this case, we can introduce a multiplier θ for the constraint and write its Lagrangian as:

$$\mathcal{L}(u,\theta) = \sum_{i=1}^{N}\lambda_i F_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right) + \theta\left(N - C - \sum_{i=1}^{N} u_i\right) = \sum_{i=1}^{N}\left[\lambda_i F_0^{(i)}\left((F^{(i)})^{-1}(u_i)\right) - \theta u_i\right] + \theta(N-C). \tag{6}$$

Let us impose the KKT conditions for optimality. We minimize first over $u_i \in [0,1]$, where the problem decouples over i:

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

By using (5), the gradient of the objective is computed as $\eta^{(i)}((F^{(i)})^{-1}(u_i)) - \theta^*$, and it is increasing by hypothesis. Therefore, if $\eta^{(i)}((F^{(i)})^{-1}(0)) = \eta^{(i)}(0) \geqslant \theta^*$, the above optimum is attained for $u_i^* = T_i^* = 0$ and the content is always stored. If instead, $\eta^{(i)}((F^{(i)})^{-1}(1)) = \lim_{t\to\infty}\eta^{(i)}(t) < \theta^*$, the optimum is attained for $u_i^* = 1$ ($T_i^* = \infty$) and the item is never stored. In the remaining cases, the optimum is interior and satisfies:

$$\eta^{(i)}((F^{(i)})^{-1}(u_i^*)) = \eta^{(i)}(T_i^*) = \theta^*. \tag{7}$$

Finally, the optimal threshold θ^* must satisfy the *complementary slackness* condition:

$$\theta^*\left(N - C - \sum_{i=1}^{N} u_i^*\right) = 0. \tag{8}$$

Note that in the optimum of (4), we cannot have strict inequality in the constraint, because in that case the objective can be improved by lowering some u_i^*. Therefore, the second term in (8) must be 0 and θ^* must satisfy:

$$\sum_{i=1}^{N} u_i^* = \sum_{i=1}^{N} F^{(i)}(T_i^*(\theta^*)) = N - C, \tag{9}$$

with $T_i^*(\theta^*)$ constructed as before. ∎

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 $\eta^{(i)}(\infty) < \theta^*$ 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.

Note moreover that, in the case of increasing hazard rates, pre-fetching *strictly* improves upon caching, since the optimal caching policy is just the static policy, as given in Theorem 1.

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 policy degenerates to the *static* policy where $T_i^* = 0$ for the C most-popular objects, which remain in memory forever. If instead the hazard rates are *decreasing*, we have the following result, which can be proved using similar arguments to [14, Theorem 1], i.e. that we are now minimizing a *concave* function over a convex domain, 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 and pre-fetching does not help, i.e. *caching* makes more sense for DHR and *pre-fetching* for IHR traffic.

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. However, from the formulation of Problems 1 and 2 it is clear that a strong connection exists between both policies. This connection is better understood from the depiction in Fig. 4. When hazard rates are monotone, either decreasing or increasing, the optimal policy is just a *threshold policy* on the current hazard rates, denoted by $\lambda_i(t)$ and defined by:

$$\lambda_i(t) := \eta^{(i)}(t - \tau_i^*(t)), \tag{10}$$

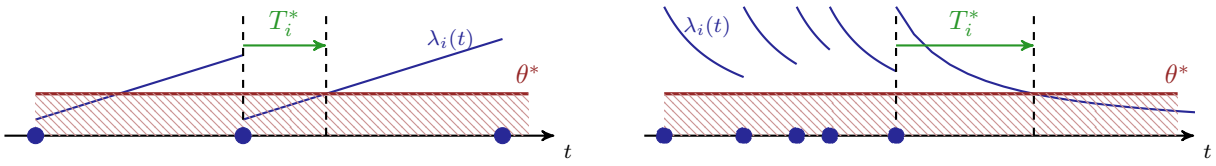

Fig. 4. Pre-fetching and caching as threshold policies for the hazard rates.

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 increasing 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 decreasing case, this translates to 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:

$$F^{(i)}(t) = P\left(-\tau_0^{(i)} \leqslant t\right) = F(\lambda_i t) \tag{11a}$$

$$\eta^{(i)}(t) = \frac{f^{(i)}(t)}{1 - F_0^{(i)}(t)} = \lambda_i \eta(\lambda_i t) \tag{11b}$$

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:

$$X^{(i)} = \eta^{(i)}(-\tau_0^{(i)}). \tag{12}$$

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

$$\begin{aligned} G(x) &:= P(\eta(-\tau_0) \leqslant x) = P\left(-\tau_0 \in \eta^{-1}([0,x])\right) \\ &= \int_{\eta^{-1}([0,x])} F(dx), \end{aligned} \tag{13}$$

since $-\tau_0 \sim F$.

By resorting to eqs. (11), 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:

$$G^{(i)}(x) = G(x/\lambda_i). \tag{14}$$

A. Scaling the 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 $\lambda_1^{(N)} \geqslant \ldots \lambda_N^{(N)} > 0$. Define the following *empirical* distribution function:

$$L_N(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\left\{\lambda_i^{(N)} \leqslant \lambda\right\}}. \tag{15}$$

Then L_N is a distribution function in $\lambda \geqslant 0$. In order to have a proper limit theorem, we will need the following:

Assumption 2: As $N \to \infty$, the distribution $L_N \Rightarrow^w L$, a fixed distribution, where $\Rightarrow^w$ denotes usual weak convergence.

This assumption is very general; we explore some parametric examples that satisfy this 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, characterized by eqs. (7, 8). For each N, we assume that the inter-arrival distributions satisfy Assumption 1 and that the arrival rates of the N-th system are characterized by their empirical distribution $L_N(\lambda)$, satisfying Assumption 2.

We need a final assumption on the hazard rate function of the base distribution:

Assumption 3: The hazard rate function η associated to F_0 is continuous and strictly monotone.

Remark that in this case, the set $\eta^{-1}([0,x])$ in (13) will be an interval. Since F is a continuous distribution, we have that G is also continuous. We are now ready to prove the main result of the paper.

Theorem 4: Consider a family of local memory systems, indexed by N, with request processes satisfying Assumptions 1–3. 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 following function:

$$G_\infty(x) := \int_0^\infty G(x/\lambda)L(d\lambda), \tag{16}$$

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

$$G_\infty(\theta^*) = 1 - c, \tag{17}$$

then the hazard rate threshold θ_N^* defined by (9) for the N-th system verifies:

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

Proof: We prove the theorem for monotonically increasing hazard rates; the proof for decreasing hazard rates follows along similar lines. We begin by rewriting the memory constraint equation (9) 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:

$$\frac{1}{N}\sum_{i=1}^N F^{(i)}(T_i^*(\theta_N^*)) = 1 - c. \tag{18}$$

Now $F^{(i)}(T_i^*(\theta_N^*)) = P(-\tau_0^{(i)} \leqslant T_i^*(\theta_N^*))$. Using the KKT 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 $G^{(i)}$ (13) we get:

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

Substituting in the left-hand side of (18), and resorting to the scaling property (14) we get:

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

Using the above equality, we can rewrite (18) using the empirical distribution L_N as:

$$\int_0^\infty G(\theta_N^*/\lambda)L_N(d\lambda) = 1 - c$$

Observe that the left-hand side is similar to (13), but with L_N instead of L. Since by the Assumptions, G is a continuous distribution function, and thus bounded, from the weak convergence of L_N we have that:

$$\int_0^\infty G(\theta/\lambda)L_N(d\lambda) \to_{N\to\infty} G_\infty(\theta)$$

for any θ continuity point of G_∞. Therefore, θ_N^* is a quantile of a distribution function with limit G_∞. Using Lemma [18, Lemma 21.2], which guarantees the pointwise convergence of quantiles in this case, we have that in the limit $\theta_N^* \to \theta^*$, the quantile of the G_∞ distribution:

$$G_\infty(\theta^*) = 1 - c,$$

thus concluding the proof. ∎

C. Asymptotic performance

Theorem 4 shows that in the limit, the optimal policy behaves as a fixed threshold policy satisfying (17): for large N, a given item i will be stored in the cache if and only its hazard rate is higher than $\theta_N^* =\approx \theta^*$. However, the request will be a hit or a miss depending on whether the hazard rate of item i is above the threshold *just prior* to a new request. This magnitude is *synchronized* with requests and thus must be computed in terms of the inter-arrival times. We now exploit this remark to obtain an asymptotic performance result for the miss probability of the system.

Since we are observing the process on a request, we are conditioning on $\tau_0^{(i)} = 0$.[2] We now introduce the *observed hazard rate upon arrival* distribution. Let $X_0^{(i)} = \lambda_i((\tau_0^{(i)})^-)$, with $\lambda_i(t)$ as in (10). This is the hazard rate at the moment when an arrival for process i occurs. Then the request will be a *miss* if and only if $X_0^{(i)} \leqslant \theta_N^*$, where θ_N^* is the previous threshold.

For the base distribution F_0, and increasing hazard rates, we can compute the distribution of the random variable X_0 by recalling eq. (10) as:

$$\begin{aligned} G_0(x) := P(X_0 \leqslant x) &= P(\lambda(\tau_0^-) \leqslant x) = P(\eta(\tau_{-1}) \leqslant x) \\ &= P\left(\tau_{-1} \in \eta^{-1}([0,x])\right) \end{aligned}$$

Where we have used that $\tau_0 = 0$ because we are synchronized with process i. In this case τ_{-1} follows the inter-arrival distribution F_0, and thus:

$$G_0(x) = \int_{\eta^{-1}([0,x])} F_0(dx). \tag{19}$$

Using the scaling properties (11), we get the corresponding versions for the i-th process:

$$G_0^{(i)}(x) = G_0(x/\lambda_i). \tag{20}$$

We need a final Assumption for our asymptotic result:

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. Then the miss probability for system N, M_N, satisfies:

$$M_N \underset{N\to\infty}{\longrightarrow} \frac{\displaystyle\int_0^\infty \lambda G_0(\theta^*/\lambda)L(d\lambda)}{\displaystyle\int_0^\infty \lambda L(d\lambda)}, \tag{21}$$

where θ^* is defined by eq. (17).

Proof: The miss probability in the $N-$th system is given by the optimum of 2:

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

[2]More formally, we are using the Palm probability of process i to condition on the event $\tau_0^{(i)} = 0\}$. Due to stationarity, this probability is independent of the point sampled.

By substituting T_i^* and using the KKT conditions, we can work similarly to (18) to express the above in terms of G_0:

$$M_N = \sum_{i=1}^{N} \lambda_i^{(N)} G_0\left(\frac{\theta_N^*}{\lambda_i^{(N)}}\right) = N \int_0^\infty \lambda G_0\left(\frac{\theta_N^*}{\lambda}\right) L_N(d\lambda).$$

Moreover, the total arrival rate for the N-th system is:

$$\lambda_N := \sum_{i=1}^{N} \lambda_i^{(N)} = N \int_0^\infty \lambda L_N(d\lambda).$$

Therefore, the miss probability for system N is:

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

By using Assumptions 2 and 4 and taking limit as $N \to \infty$, we can substitute L_N by its limit L and the result follows. ■

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 2.

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:

$$L(\lambda) = 1 - \lambda^{-1/\beta} \quad \text{for } \lambda \geq 1. \tag{22}$$

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 $\lambda = 1$.

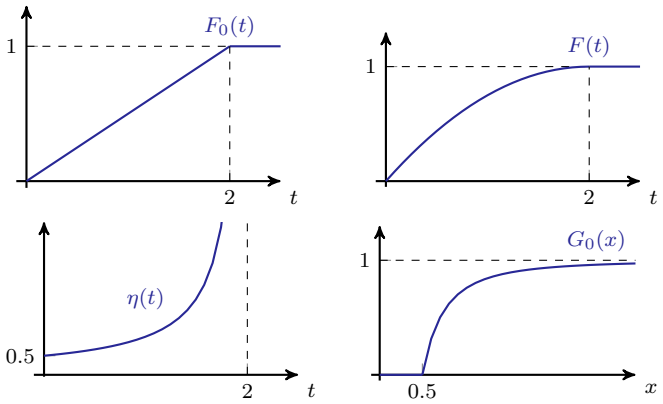

Fig. 5. Uniform inter-arrival times and associated distributions.

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 popularities, we now analyze a parametric model for F_0, namely that inter-arrival times follow a uniform distribution, which has increasing hazard rates. Under Assumption 1, the base distribution F_0 and its associated age distribution are given by:

$$F_0(t) = \begin{cases} t/2 & 0 < t < 2 \\ 1 & t \geqslant 2 \end{cases}, \; F(t) = \begin{cases} t - \frac{t^2}{4} & 0 < t < 2 \\ 1 & t \geqslant 2 \end{cases}, \tag{23}$$

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 codomain $[1/2, \infty)$. Applying eq. (13) we can compute the observed hazard rate G_0 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 compute the relevant function for the miss rate from eq. (19):

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

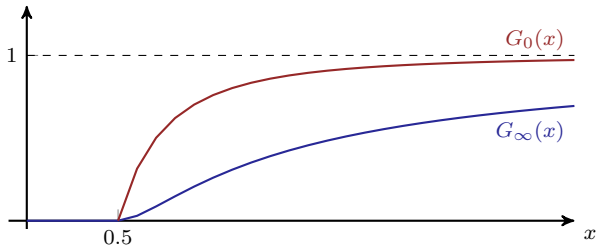

Fig. 6. Asymptotic observed hazard rate for Zipf(1/2) popularities and uniform inter-arrival times.

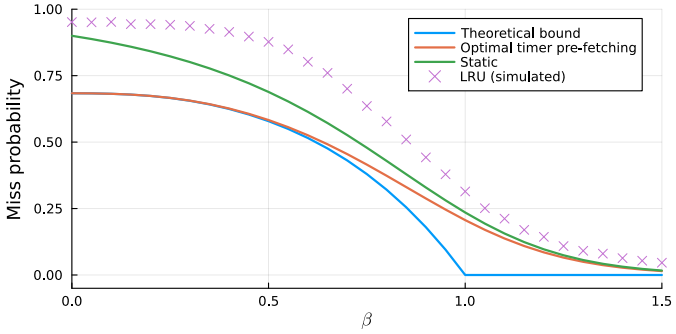

Fig. 7. Miss probability comparison for optimal timer pre-fetching, static storage and LRU caching. The theoretical bound is computed using eq. (21).

Armed with the above results, we can compute the asymptotic observed hazard rates by substituting in eq. (16):

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

This integral can be explicitly solved, for any value of β, enabling us to compute the threshold θ^* by numerically solving eq. (17). The function $G_\infty(x)$ is depicted in Fig. 6 for $\beta = 1$.

For $\beta < 1$, the family $\{L_N\}$ defined above is uniformly integrable, and thus (21) 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. It is easy to show [13] that for $\beta \geqslant 1$, $M_N \to_N 0$. Also shown are the performance of the optimal timer-based pre-fetching policy for finite $N = 10000$, $C = 1000$ computed by explicitly solving Problem 2 and the corresponding static policy (which is also the optimal TTL caching policy for this traffic). Finally, we show the miss probability obtained by using a classical caching strategy such as LRU, to highlight how bad performance can be in this scenario if we do not take into account the regularity of the traffic pattern.

VII. Conclusions

In this paper, we analyzed the role of the hazard rate function in the inter-arrival times of requests to a local memory systems, highlighting how the shape of the hazard rates crucially determines which is 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 this kind of 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 performance.

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 paper.

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.