

# Caching and pre-fetching: the role of hazard rates.

**Andres Ferragut**

joint work with Matias Carrasco and Fernando Paganini

Universidad ORT Uruguay

Laboratory for Information, Networking and Communication Sciences Seminar – May 2024

The caching problem

Point processes and stochastic intensity

The optimal caching policy

Large scale asymptotics

Connection with timer-based policies

Conclusions

## The caching problem

Point processes and stochastic intensity

The optimal caching policy

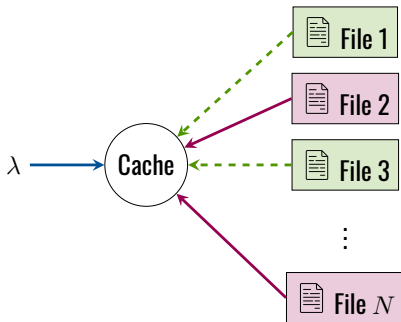
Large scale asymptotics

Connection with timer-based policies

Conclusions

# The caching problem

- Consider a **local memory system** that handles items from a catalog of  $N$  objects.
- Requests for objects arrive as a random process.
- The memory (cache) can locally store  $C < N$  of them.
- If item is in cache, we have a **hit**. Otherwise, it is a **miss**.



**Objective:** for a given arrival stream, maximize the steady-state **hit rate**.

# A sequential approach

- Consider a sequence of random variables  $Z_1, Z_2, \dots$  with values in  $\{1, \dots, N\}$ .
- Consider also the set of feasible subsets:

$$\mathcal{C} = \{\{i_1, \dots, i_k\} \subset \{1, \dots, N\}, k \leq C\}$$

- A (causal) caching policy would be a sequence of maps  $\pi_n$  deciding which contents to store:

$$\pi_n(Z_1, \dots, Z_{n-1}) \rightarrow \mathcal{C}$$

- In probabilistic terms, let  $\mathcal{F}_n = \sigma(Z_1, \dots, Z_n)$ , then  $\pi_n$  is any  $\mathcal{C}$ -valued  $\mathcal{F}_n$ -predictable process ( $\mathcal{F}_{n-1}$ -measurable).

# A simple case

## The Independent Reference Model (IRM)

- Assume now that  $Z_n$  are *iid* with distribution  $p_i = P(Z_n = i)$ , where  $p_i$  is the **popularity** of content  $i$ . Wlog, we take  $p_1 \geq p_2 \geq \dots$
- In this case,  $Z_n \mid \mathcal{F}_{n-1} \sim p$ , thus the hit probability at time  $n$  is:

$$P(Z_n \in \pi_n) = E[\mathbf{1}_{Z_n \in \pi_n}] = E[E[\mathbf{1}_{Z_n \in \pi_n} \mid \mathcal{F}_{n-1}]] = E\left[\sum_{i \in \pi_n} p_i\right] \leq \sum_{i=1}^C p_i$$

- Taking  $\pi_n \equiv \{1, \dots, C\}$  achieves the bound.

# A simple case

## The Independent Reference Model (IRM)

- Assume now that  $Z_n$  are *iid* with distribution  $p_i = P(Z_n = i)$ , where  $p_i$  is the **popularity** of content  $i$ . Wlog, we take  $p_1 \geq p_2 \geq \dots$
- In this case,  $Z_n \mid \mathcal{F}_{n-1} \sim p$ , thus the hit probability at time  $n$  is:

$$P(Z_n \in \pi_n) = E[\mathbf{1}_{Z_n \in \pi_n}] = E[E[\mathbf{1}_{Z_n \in \pi_n} \mid \mathcal{F}_{n-1}]] = E\left[\sum_{i \in \pi_n} p_i\right] \leq \sum_{i=1}^C p_i$$

- Taking  $\pi_n \equiv \{1, \dots, C\}$  achieves the bound.

**Conclusion:** under iid requests, the static “keep the most popular” policy is optimal.

## Practical policies: LFU and LRU

In practice, popularities are not known. This leads to the **least-frequently-used (LFU)** eviction policy:

- Take  $\pi_n$  as the most requested objects so far (remove the least frequently used).
- In the long range, converges to the static policy.

Another popular eviction policy is **least-recently-used (LRU)**, which treats  $\pi_n$  as a list defined recursively:

- If  $Z_n \in \pi_n$ , serve the content, move  $Z_n$  to the front of the list.
- If  $Z_n \notin \pi_n$ , fetch the content, put  $Z_n$  in the front of the list, remove the last object in the list (which is the least recently requested).

**LRU** adapts best to **bursty** traffic.



# The caching problem, take 2

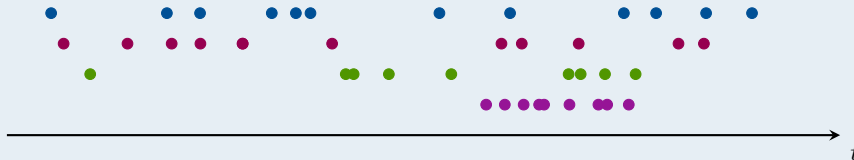
Sequential models lack **time information**, which may be useful!

# The caching problem, take 2

Sequential models lack **time information**, which may be useful!

Point process approach [Fofack et al. 2014]:

- Assume requests for item  $i$  come from a **point process** of intensity  $\lambda_i := \lambda p_i$ .



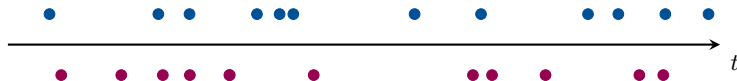
- At each point in time we must decide which items must be stored locally.

If inter-request times are **heavy tailed**, this can model burstiness.

# Example: Pareto arrivals

Consider two items, with equal popularity...

## ■ Poisson arrivals:



Homogeneous

## ■ Heavy tailed arrivals (Pareto $\alpha = 2$ ):



Bursty!

## Some open questions...

- What is the optimal causal policy in this framework?
- Can we compute the optimal hit rate/hit probability?
- What is its large scale behavior?
- How typical policies compare to the optimal one?

The caching problem

**Point processes and stochastic intensity**

The optimal caching policy

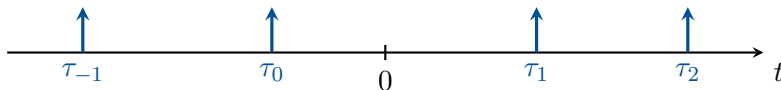
Large scale asymptotics

Connection with timer-based policies

Conclusions

## A bit of point process theory...

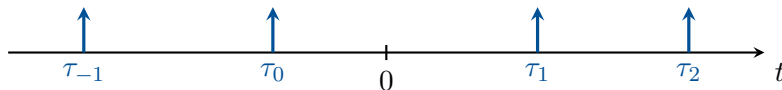
Let  $\Phi = \{\tau_k : k \in \mathbb{Z}\}$  be a **stationary point process** representing request times:



i.e.  $\Phi(B) = \sum_k \mathbf{1}_{\{\tau_k \in B\}}$  is a random counting measure.

# A bit of point process theory...

Let  $\Phi = \{\tau_k : k \in \mathbb{Z}\}$  be a **stationary point process** representing request times:

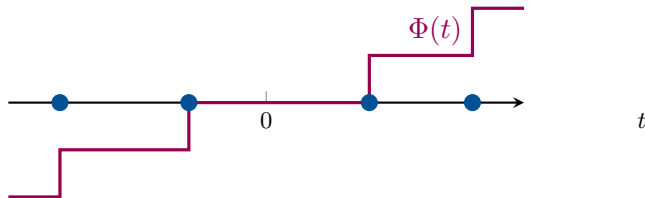


i.e.  $\Phi(B) = \sum_k \mathbf{1}_{\{\tau_k \in B\}}$  is a random counting measure.

**Counting process:**

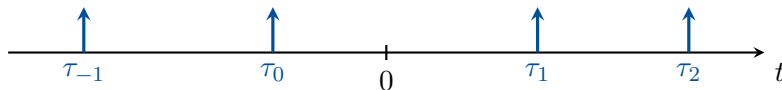
$$\Phi(t) = \begin{cases} \Phi((0, t]) & t > 0 \\ -\Phi((t, 0]) & t \leq 0 \end{cases}$$

**Note:**  $\Phi(\tau_k) = k$



# A bit of point process theory...

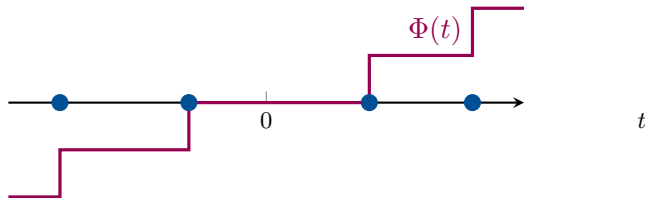
Let  $\Phi = \{\tau_k : k \in \mathbb{Z}\}$  be a **stationary point process** representing request times:



i.e.  $\Phi(B) = \sum_k \mathbf{1}_{\{\tau_k \in B\}}$  is a random counting measure.

**Counting process:**

$$\Phi(t) = \begin{cases} \Phi((0, t]) & t > 0 \\ -\Phi((t, 0]) & t \leq 0 \end{cases}$$

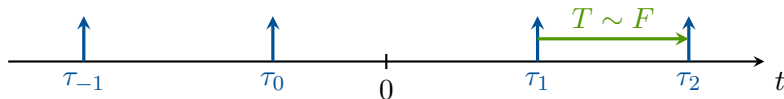


**Note:**  $\Phi(\tau_k) = k$

Let  $\mathcal{F}_t = \sigma(N(s), s \leq t)$  be its **internal history**.



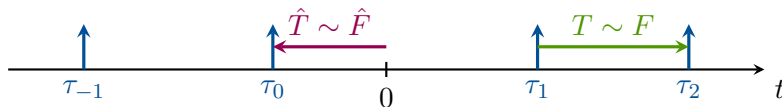
## Two important distributions:



**Inter-arrival distribution:**  $F(t) := P_N^0(\tau_1 - \tau_0 \leq t), \quad E_\Phi^0[\tau_1] = 1/\lambda.$

**Note:** here  $P_\Phi^0$  is the **Palm probability** of the point process (conditioning on  $\tau_0 = 0$ ).

## Two important distributions:



**Inter-arrival distribution:**  $F(t) := P_N^0(\tau_1 - \tau_0 \leq t), \quad E_\Phi^0[\tau_1] = 1/\lambda.$

**Age distribution:**  $\hat{F}(t) := P(-\tau_0 \leq t) = \lambda \int_0^t 1 - F(s) ds,$

**Note:** here  $P_\Phi^0$  is the **Palm probability** of the point process (conditioning on  $\tau_0 = 0$ ).

Consider a simple stationary point process  $\Phi$  with intensity  $\lambda$ , defined in some probability space  $(\Omega, \mathcal{F}, P)$ . Let some filtration  $\{\mathcal{F}_t\}_{t \in \mathbb{R}}$  be a **history** of the process.

## Definition:

The random process  $\lambda(t) \geq 0$  is a **stochastic intensity** for the history  $\mathcal{F}_t$  iff it is a.s. locally integrable,  $\mathcal{F}_t$ -adapted and:

$$E [\Phi((a, b]) \mid \mathcal{F}_a] = E \left[ \int_a^b \lambda(t) dt \mid \mathcal{F}_a \right]$$

for all  $a, b \in \mathbb{R}$ .

Local interpretation:

$$E[\Phi((t, t + h]) \mid \mathcal{F}_t] = \lambda(t)h + o(h) \quad P - a.s.,$$

So  $\lambda(t)$  acts as a **local** notion of intensity based on previous history.

### Local interpretation:

$$E[\Phi((t, t + h]) \mid \mathcal{F}_t] = \lambda(t)h + o(h) \quad P - a.s.,$$

So  $\lambda(t)$  acts as a **local** notion of intensity based on previous history.

### Martingale interpretation:

$$M_a(t) = \Phi(t) - \Phi(a) - \int_a^t \lambda(s)ds$$

is a local  $(P, \mathcal{F}_t)$  martingale for any  $a \in \mathbb{R}$ .

- If  $\Phi(t)$  is a Poisson process, then we know that

$$M(t) = \Phi(t) - \lambda t = \Phi(t) - \int_0^t \lambda dt$$

is a martingale, so the stochastic intensity of a Poisson process is just  $\lambda(t) \equiv \lambda$ .

- If  $\Phi(t)$  is a Poisson process, then we know that

$$M(t) = \Phi(t) - \lambda t = \Phi(t) - \int_0^t \lambda dt$$

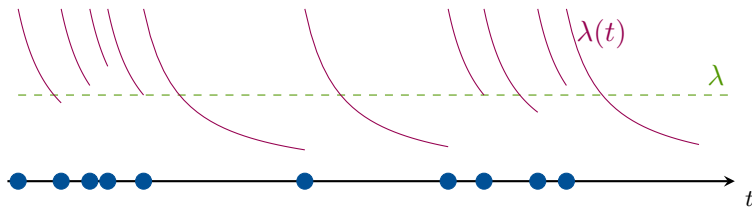
is a martingale, so the stochastic intensity of a Poisson process is just  $\lambda(t) \equiv \lambda$ .

The poisson process is the “white noise” of point processes.

# Stochastic intensity

A local notion of intensity...

However, if traffic is **bursty**, the stochastic intensity **rises** after arrivals:



Note: for stationary processes,  $E[\lambda(t)] = E[\lambda(0)] = \lambda$ , the average intensity.



# Renewal processes

- Let now  $\Phi$  be a **stationary renewal process**, i.e. inter request times  $\tau_{k+1} - \tau_k$  are *iid*  $\sim F$ .
- Assume that  $F$  has a density, and define the **hazard rate** of  $F$  as:

$$\eta(t) = \frac{f(t)}{1 - F(t)}$$

# Renewal processes

- Let now  $\Phi$  be a **stationary renewal process**, i.e. inter request times  $\tau_{k+1} - \tau_k$  are  $iid \sim F$ .
- Assume that  $F$  has a density, and define the **hazard rate** of  $F$  as:

$$\eta(t) = \frac{f(t)}{1 - F(t)}$$

## Theorem (Daley-Vere Jones, Chapter 7)

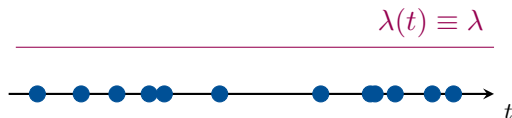
For a renewal process and its natural history, the stochastic intensity is:

$$\lambda(t) = \eta(t - \tau^-(t)),$$

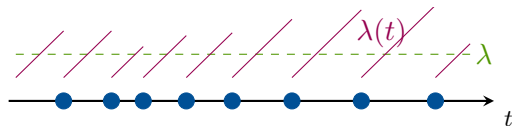
where  $\tau^-(t)$  is the last point before  $t$ :

$$\tau^-(t) = \sup\{\tau_k : \tau_k < t\}$$

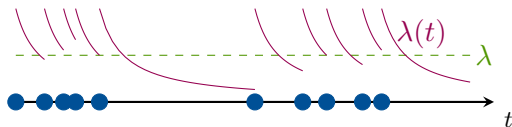
## Some examples...



Constant hazard rate  $\rightarrow$  Poisson process.



Increasing hazard rate  $\rightarrow$  more periodic!



Decreasing hazard rate  $\rightarrow$  more bursty!

The caching problem

Point processes and stochastic intensity

**The optimal caching policy**

Large scale asymptotics

Connection with timer-based policies

Conclusions

# Causal caching policies

- Consider again a cache system fed by  $N$  **independent** request processes  $\Phi_i(t)$  with stochastic intensities  $\lambda_i(t)$ .
- Let  $\mathcal{F}_t = \sigma(\{\mathcal{F}_t^{(i)} : i = 1, \dots, M\})$  their aggregate history.

## Definition

A **causal** caching policy is an  $\mathcal{F}_t$  **predictable** stochastic process

$$\pi(t) : \Omega \times \mathbb{R} \rightarrow \mathcal{C}$$

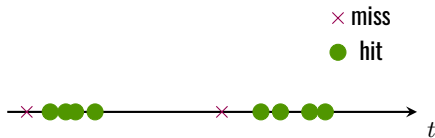
i.e.  $\pi(t) = \{i_1, \dots, i_k\}$  (with  $k \leq C$ ) is the subset kept at time  $t$ , and only depends on the past history of item requests.

# The hit process

## Stochastic intensity

Focus now on a particular content  $i$ , its **hit process** is the point process given by:

$$H_i(B) = \sum_{k \in \mathbb{Z}} \mathbf{1}_{\{\tau_k^i \in B\}} \mathbf{1}_{\{i \in \pi(\tau_k^i)\}}$$



Now  $\mathbf{1}_{\{i \in \pi(t)\}}$  is  $\mathcal{F}_t$ -predictable, so the stochastic intensity of  $H_i$  is:

$$h_i(t) = \lambda_i(t) \mathbf{1}_{\{i \in \pi(t)\}}$$

i.e.,  $h_i(t) = \lambda_i(t)$  while  $i$  is cached and otherwise 0.

# The hit process

## The hit rate

If we now consider the aggregate of requests, the **total hit process** is given by:

$$H = \sum_{i=1}^N H_i$$

And its stochastic intensity is just:

$$h(t) = \sum_{i=1}^N h_i(t) = \sum_{i=1}^N \lambda_i(t) \mathbf{1}_{\{i \in \pi(t)\}}$$

The steady state **hit rate** of the policy is:

$$\text{hit rate} = \lambda_{\text{hit}} := E[h(0)]$$

# Maximizing the hit rate

In order to maximize  $\lambda_{\text{hit}}$ , consider the causal policy:

$$\pi^*(t) = \{i_1, \dots, i_C\} \quad \text{such that} \quad \sum_{i \in \{i_1, \dots, i_C\}} \lambda_i(t) \text{ is maximized.}$$

Then, for any causal policy  $\pi$  and for each realization:

$$h(t) = \sum_{i \in \pi(t)} \lambda_i(t) \leq \sum_{i \in \pi^*(t)} \lambda_i(t) = h^*(t).$$

## Theorem

The **optimal causal policy** is to keep in the cache the  $C$  objects with the **highest stochastic intensity** at any time.



## Back to the Poisson case

- Assume the  $\Phi_i$  are Poisson processes of intensities  $\lambda_i$ .
- We take  $\lambda_1 > \lambda_2 > \dots$  as the popularities.
- The total request process is also Poisson of intensity  $\sum_i \lambda_i$ .
- In that case, the optimal policy is:

$$\pi^*(t) \equiv \{1, \dots, C\}$$

since  $\lambda_i(t) \equiv \lambda_i$  and these are is decreasing.

## Back to the Poisson case

- Assume the  $\Phi_i$  are Poisson processes of intensities  $\lambda_i$ .
- We take  $\lambda_1 > \lambda_2 > \dots$  as the popularities.
- The total request process is also Poisson of intensity  $\sum_i \lambda_i$ .
- In that case, the optimal policy is:

$$\pi^*(t) \equiv \{1, \dots, C\}$$

since  $\lambda_i(t) \equiv \lambda_i$  and these are decreasing.

**Conclusion:** under Poisson arrivals, statically keeping the most popular objects is optimal (compare to the IRM before).

# The Renewal case

- If now the  $\Phi_i$  are renewal processes of (decreasing) intensities  $\lambda_i$ .
- The total request process is no longer renewal, but its intensity is again  $\sum_i \lambda_i$ .
- Since  $\lambda_i(t) = \eta_i(t - T_i^*(t))$ , the optimal policy is:
  - Keep track of the **current hazard rate** of each content  $i$ .
  - Choose to keep in  $\pi^*(t)$  the  $C$  highest.

# The Renewal case

- If now the  $\Phi_i$  are renewal processes of (decreasing) intensities  $\lambda_i$ .
- The total request process is no longer renewal, but its intensity is again  $\sum_i \lambda_i$ .
- Since  $\lambda_i(t) = \eta_i(t - T_i^*(t))$ , the optimal policy is:
  - Keep track of the **current hazard rate** of each content  $i$ .
  - Choose to keep in  $\pi^*(t)$  the  $C$  highest.

**Conclusion:** under renewal arrivals, the optimal policy only depends on the current hazard rates since the last request.

# An interesting observation

## Decreasing hazard rates

- If hazard rates are **decreasing**, caching makes sense! After an arrival it becomes more likely to get another request.
- After some time, we will evict the content to make room for more recent ones (as in LRU).

# An interesting observation

## Decreasing hazard rates

- If hazard rates are **decreasing**, caching makes sense! After an arrival it becomes more likely to get another request.
- After some time, we will evict the content to make room for more recent ones (as in LRU).

## Increasing hazard rates

- If instead hazard rates are **increasing**, then when a request arrives, the item becomes less likely to be requested again!
- It may be better to remove it and make room for other ones (i.e. LRU makes no sense!).
- If we haven't seen it for a while, then we may have to fetch it **anticipating** the upcoming request.

The caching problem

Point processes and stochastic intensity

The optimal caching policy

**Large scale asymptotics**

Connection with timer-based policies

Conclusions

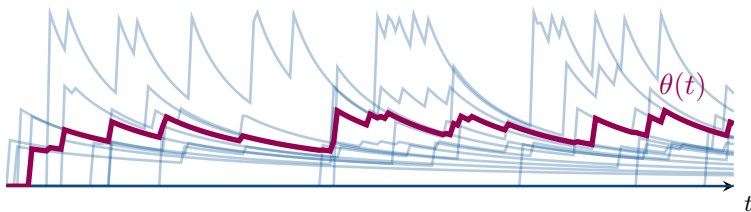
# Understanding the optimal policy

## The threshold process

We can rewrite this optimal policy as a **threshold** policy:

$$i \in \pi^*(t) \Leftrightarrow \lambda_i(t) \geq \theta(t) := \text{the } C \text{ largest stochastic intensity}$$

**Example:** Pareto requests, Zipf popularities,  $N = 20$ ,  $C = 4$ .



¿What is the large scale behavior of  $\theta(t)$  in steady state?



# The threshold value in steady state

- Now we have  $N$  independent renewal processes with intensities  $\lambda_i(t)$ .
- At time  $t = 0$ , we have a sample  $\{X_1, \dots, X_N\}$  of independent, but **not identically distributed** random variables, with distribution:

$$X_i \sim \eta_i(-\tau_0^i), \quad -\tau_0 \sim \hat{F}_i(t)$$

- The threshold  $\theta(0)$  is the  $C$ -th **order statistic** (in decreasing order) of the sample.

**Problem:** for non *iid* random variables, no closed form  $\rightarrow$  Can we say something about the large scale limit?

## A useful Theorem

Let  $\{X_i\}$  be a sequence of independent random variables with distributions  $G_i$ . Define:

$$\hat{G}_N(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{X_i \leq x\}}$$

the empirical distribution, and let:

$$\bar{G}_N(x) = \frac{1}{N} \sum_{i=1}^N G_i(x)$$

### Theorem (Shorack)

If the family  $\{G_i\}$  is tight, then:

$$\|\hat{G}_N - \bar{G}_N\|_{\infty} \rightarrow 0 \quad \text{almost surely as } N \rightarrow \infty.$$

## A little more structure

Assume now that the request processes come from a common scale family, i.e. their inter-arrival distributions satisfy:

$$F_i(t) = F_0(\lambda_i t)$$

where  $F_0$  has mean 1, so  $F_i$  has mean  $1/\lambda_i$ .

Assume now that the request processes come from a common scale family, i.e. their inter-arrival distributions satisfy:

$$F_i(t) = F_0(\lambda_i t)$$

where  $F_0$  has mean 1, so  $F_i$  has mean  $1/\lambda_i$ .

In this case:

- The distribution of  $-\tau_0^i$  is  $\hat{F}_i(t) = \hat{F}_0(\lambda_i t)$ .
- The hazard-rate of  $F_i$  is  $\eta_i(t) = \lambda_i \eta_0(t/\lambda_i)$ .
- The random variable  $X_i \sim G_i(x) := G_0(x/\lambda_i)$

where  $G_0(x) = P(\eta_0(-\tau_0) \leq x)$  is the observed hazard rate distribution for the base process.

Consider now the popularities  $\lambda_1 > \dots > \lambda_N$  and define:

$$\phi_N(\lambda) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{\lambda_i \leq \lambda\}}$$

their empirical (deterministic) distribution.

# The distribution of popularities

Consider now the popularities  $\lambda_1 > \dots > \lambda_N$  and define:

$$\phi_N(\lambda) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{\lambda_i \leq \lambda\}}$$

their empirical (deterministic) distribution.

**Assumption:**

$$\phi_N(\lambda) \rightarrow \phi(\lambda) \quad \text{as } N \rightarrow \infty \text{ (weakly)}$$

where  $\phi(\lambda)$  is a probability distribution.

## Example: Zipf popularities

- A common model for popularities is the **Zipf** distribution, where  $\lambda_i \propto \frac{1}{i^\beta}$ .

- In our framework, take:

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

- Then we can show that:

$$\phi_N(\lambda) \rightarrow \phi(\lambda) = \left[1 - \lambda^{-1/\beta}\right] \mathbf{1}_{\{\lambda \geq 1\}}$$

**Remark:** note that  $\sum_i \lambda_i$  diverges, so the system is scaling up...

## Theorem (Carrasco,F',Paganini)

Consider a caching system fed by  $N$  independent and stationary renewal processes, with intensities  $\{\lambda_i\}$ , and inter-arrival distributions  $F_i(t) = F_0(\lambda_i t)$ . Let  $X_1, \dots, X_N$  denote the observed hazard-rates at time 0. Then, under the preceding assumption, the empirical distribution:

$$\hat{G}_N(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{X_i \leq x\}} \rightarrow_N G_\infty(x) = \int_0^\infty G_0\left(\frac{x}{\lambda}\right) \phi(d\lambda)$$



- By Shorack's result:

$$\hat{G}_N(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{X_i \leq x\}} \approx \bar{G}_N := \frac{1}{N} \sum_{i=1}^M G_i(x)$$

- Note that:

$$\frac{1}{N} \sum_{i=1}^N G_i(x) = \sum_{i=1}^N G_0\left(\frac{x}{\lambda_i}\right) \frac{1}{N} = \int_0^\infty G_0\left(\frac{x}{\lambda}\right) \phi_M(d\lambda)$$

- Use the assumption to show that:

$$\int_0^\infty G_0\left(\frac{x}{\lambda}\right) \phi_M(d\lambda) \rightarrow_M \int_0^\infty G_0\left(\frac{x}{\lambda}\right) \phi(d\lambda) = G_\infty(x).$$

## A law of large numbers for the threshold

Assume further that the cache has capacity  $C = cN$  with  $0 < c < 1$  is the fraction of the catalog that can be stored.

# A law of large numbers for the threshold

Assume further that the cache has capacity  $C = cN$  with  $0 < c < 1$  is the fraction of the catalog that can be stored.

Then, the optimal policy threshold  $\theta_N^*(0)$  is the random variable:

$$\theta_N^* : \sum_{i=1}^N \mathbf{1}_{\{X_i \leq \theta_N^*\}} = (1 - c)N$$

or equivalently  $\theta_N^*$  is such that  $\hat{G}_N(\theta_N^*) = 1 - c$ .

## A law of large numbers for the threshold

Assume further that the cache has capacity  $C = cN$  with  $0 < c < 1$  is the fraction of the catalog that can be stored.

Then, the optimal policy threshold  $\theta_N^*(0)$  is the random variable:

$$\theta_N^* : \sum_{i=1}^N \mathbf{1}_{\{X_i \leq \theta_N^*\}} = (1 - c)N$$

or equivalently  $\theta_N^*$  is such that  $\hat{G}_N(\theta_N^*) = 1 - c$ .

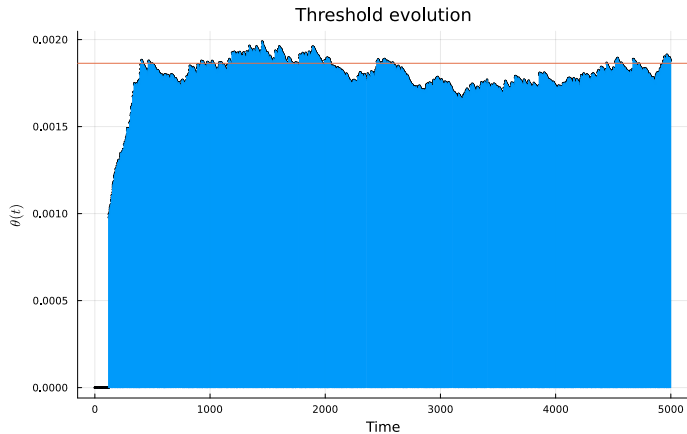
### Corollary

If the cache size scales linearly with the catalog as  $C_N = cN$ , then:

$$\theta_N^* \rightarrow \theta^* : G_\infty(\theta^*) = 1 - c$$

So the optimal policy becomes a **fixed** threshold policy.

# Simulation example



$M = 1000, C = 100$ . Pareto  $\alpha = 2$  requests, Zipf  $\beta = 0.5$  popularities.

Moreover, we can calculate the asymptotic performance:

## Theorem

Under all the above assumptions, the asymptotic **miss rate** verifies:

$$\lambda_{\text{miss},M} \rightarrow_M \int_0^\infty \lambda \tilde{G}_0 \left( \frac{\theta^*}{\lambda} \right) \phi(d\lambda) = E \left[ \Lambda \tilde{G}_0 \left( \frac{\theta^*}{\Lambda} \right) \right]$$

where  $\Lambda \sim \phi$ , and  $\tilde{G}_0$  is the distribution of the hazard-rate prior to an arrival:

$$\tilde{G}_0(x) = \int_0^\infty \mathbf{1}_{\{\eta_0(t) \leq x\}} F_0(dt).$$

The caching problem

Point processes and stochastic intensity

The optimal caching policy

Large scale asymptotics

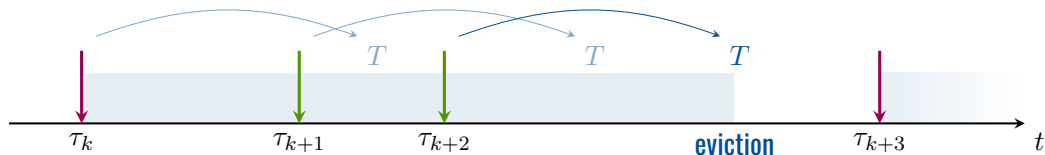
**Connection with timer-based policies**

Conclusions

# Populating a cache: timer based policies

## Timer based (TTL) policies:

- Upon request arrival for item  $i$ , check for presence.
- If new, store item and start a **timer**  $T_i$  to evict.
- If present, reset timer to  $T_i$ .
- Keep timers  $T_i$  such that **average** cache occupation is  $C$ .





# Choosing the optimal timers

Requests come from independent sources with intensities  $\lambda_i$  and inter-arrival distribution  $F_i$ :

## Problem (Optimal TTL policy)

Choose timers  $T_i \geq 0$  such that:

$$\max_{T_i \geq 0} \sum_i \lambda_i F_i(T_i)$$

subject to:

$$\sum_i \hat{F}_i(T_i) \leq C$$

**Remark:** non-convex non-linear program. But it can be solved by a change of variables!!! [Ferragut et al. 2018].

## Theorem

For the following cases, the optimal timers are:

- Constant hazard rate (Poisson) or increasing hazard rate: keep the most popular objects ( $T_i = \infty$  or 0).
- Decreasing hazard rate:

$$\eta_i(T_i^*) \geq \theta^*$$

for every stored content.

## Theorem

For the following cases, the optimal timers are:

- Constant hazard rate (Poisson) or increasing hazard rate: keep the most popular objects ( $T_i = \infty$  or 0).
- Decreasing hazard rate:

$$\eta_i(T_i^*) \geq \theta^*$$

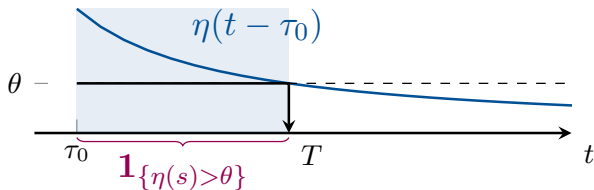
for every stored content.

# Why this happens?

**So the optimal timer policy is a threshold policy?**

# Why this happens?

So the optimal timer policy is a threshold policy?



## Theorem (F', Carrasco, Paganini)

In the scaling regime considered earlier, for renewal processes with DHR, the optimal TTL policy is also asymptotically optimal within the class of causal policies.

**Idea:** prove that the thresholds are the same in the limit.

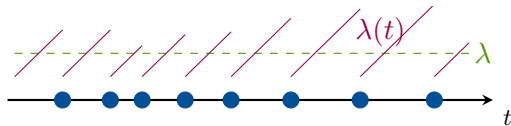
## Theorem (F', Carrasco, Paganini)

In the scaling regime considered earlier, for renewal processes with DHR, the optimal TTL policy is also asymptotically optimal within the class of causal policies.

**Idea:** prove that the thresholds are the same in the limit.  
But what about **increasing hazard rates**?

## Back to increasing hazard rates...

- Recall the increasing hazard rate behavior:

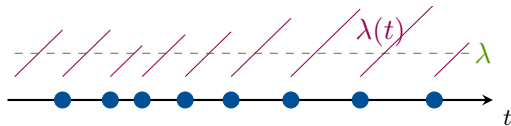


- Once you have seen a request, it's less likely to see another one for a while.



## Back to increasing hazard rates...

- Recall the increasing hazard rate behavior:



- Once you have seen a request, it's less likely to see another one for a while.

What is the timer based equivalent of this case?

# Timer based pre-fetching policies

## Key insight

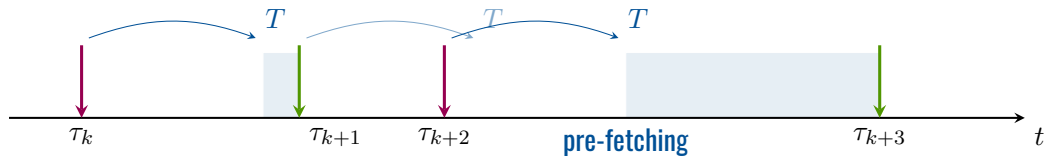
The question now is not **how long we should remember something**, but instead **how long we should forget about it!**

# Timer based pre-fetching policies

## Key insight

The question now is not **how long we should remember something**, but instead **how long we should forget about it!**

Timer based pre-fetching policy:



# Choosing the optimal timers

Requests come from independent sources with intensities  $\lambda_i$  and inter-arrival distribution  $F_i$ :

## Problem (Optimal pre-fetching policy)

Choose timers  $T_i \geq 0$  such that:

$$\max_{T_i \geq 0} \sum_i \lambda_i (1 - F_i(T_i))$$

subject to:

$$\sum_i (1 - \hat{F}_i(T_i)) \leq C$$

# Choosing the optimal timers

Requests come from independent sources with intensities  $\lambda_i$  and inter-arrival distribution  $F_i$ :

## Problem (Optimal pre-fetching policy)

Choose timers  $T_i \geq 0$  such that:

$$\min_{T_i \geq 0} \sum_i \lambda_i F_i(T_i)$$

subject to:

$$\sum_i \hat{F}_i(T_i) \geq N - C$$

**Remark:** we can use the same change of variables again!

# Pre-fetching for increasing hazard rates

Optimal pre-fetching policy, IHR, [F',Carrasco, Paganini].

The optimal timer based pre-fetching policy for IHR is such that:

$$\eta_i(T_i^*) \geq \theta^*$$

for every stored content.

**Remark:** Again we have to equalize hazard-rates. The policy is a threshold policy.

## Theorem (F', Carrasco, Paganini)

In the scaling regime considered earlier, for renewal processes with IHR, the **timer based pre-fetching policy** is asymptotically optimal.

**Idea:** as before, prove that the thresholds are the same in the limit.

The caching problem

Point processes and stochastic intensity

The optimal caching policy

Large scale asymptotics

Connection with timer-based policies

**Conclusions**



- The main result characterizes the optimal policy completely in the large-scale scenario.
- For particular distributions of interest (e.g. Pareto requests, Zipf popularities) the threshold can be computed explicitly.
- Once the threshold is computed, we can compute the asymptotic hit probability.
- Therefore, we have a computable absolute performance bound in the limit.

- The main result characterizes the optimal policy completely in the large-scale scenario.
- For particular distributions of interest (e.g. Pareto requests, Zipf popularities) the threshold can be computed explicitly.
- Once the threshold is computed, we can compute the asymptotic hit probability.
- Therefore, we have a computable absolute performance bound in the limit.
- There is much more to do!

# Merci beaucoup!

**Andres Ferragut**

`ferragut@ort.edu.uy`

`aferragu.github.io`