The caching problem under a point process perspective

Andres Ferragut

joint work with Matias Carrasco and Fernando Paganini

Universidad ORT Uruguay

Seminario PYE – Universidad de la República – Abril 2024

Outline

The caching problem

Point processes and stochastic intensity

The optimal caching policy

Large scale asymptotics

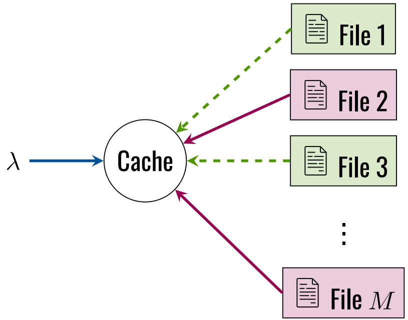
Connection with timer-based policies

Conclusions

Outline

The caching problem

- Consider a cache system with a catalog of M objects.
- Requests for objects arrive at random.
- The cache can locally store $C < M$ 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 $Z_1, Z_2, \ldots$ with values in $\{1, \ldots, M\}$.
- Consider also the set:

$$\mathcal{C} = \{\{i_1, \ldots, i_k\} \subset \{1, \ldots, M\}, k \leqslant C\}$$

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

$$\pi_n(Z_1, \ldots, Z_{n-1}) \to \mathcal{C}$$

- In probabilistic terms, let $\mathcal{F}_n = \sigma(Z_1, \ldots, Z_n)$, then π_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 $p_1 \geqslant p_2 \geqslant \ldots$.

- In this case, $Z_n \mid \mathcal{F}_{n-1} \sim p$, thus the hit probability at time n is:

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

- Taking $\pi_n \equiv \{1, \ldots, C\}$ 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 $p_1 \geqslant p_2 \geqslant \ldots$.
- In this case, $Z_n \mid \mathcal{F}_{n-1} \sim p$, thus the hit probability at time n is:

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

- Taking $\pi_n \equiv \{1, \ldots, C\}$ 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 π_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 π_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).

Beyond the IRM

- Typically, requests are correlated, and popularities evolve over time.
- For instance, requests for a file may arrive in bursts.
- LRU adapts to changes in popularity. Is good for bursts of requests. Tons of literature on this policy (also called move-to-front).
- However, performance metrics and optimality results are hard to establish.

The caching problem, take 2

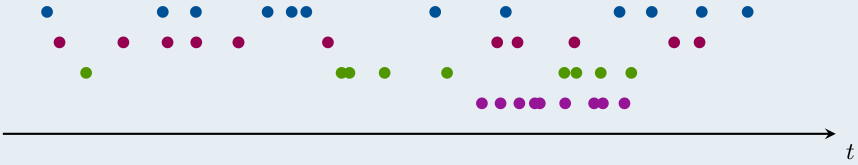
Sequential models lack time information, which may be useful!

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.

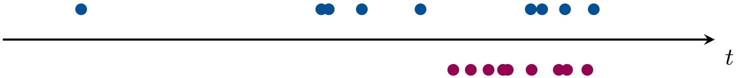
Example: Pareto arrivals

Consider two items, with equal popularity...

- Poisson arrivals:

Homogeneous

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



Bursty!

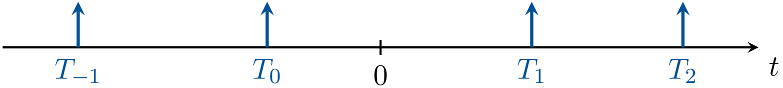
Some open questions...

- 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?

Outline

A bit of point process theory...

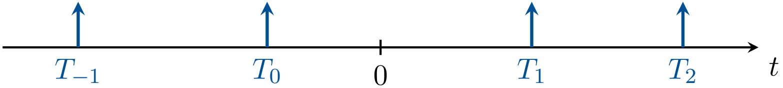
Let $N = \{T_k : k \in \mathbb{Z}\}$ be a stationary point process representing request times:



i.e. $N(B) = \sum_n \mathbf{1}_{\{T_n \in B\}}$ is a random counting measure.

A bit of point process theory...

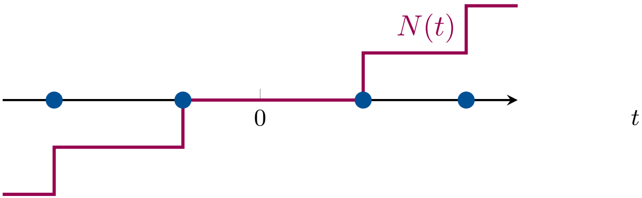
Let $N = \{T_k : k \in \mathbb{Z}\}$ be a stationary point process representing request times:



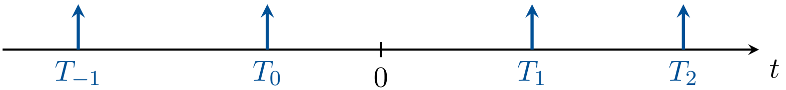
i.e. $N(B) = \sum_n \mathbf{1}_{\{T_n \in B\}}$ is a random counting measure.

Counting process:

$$N(t) = \begin{cases} N([0,t]) & t \geqslant 0 \\ -N((t,0]) & t < 0 \end{cases}$$



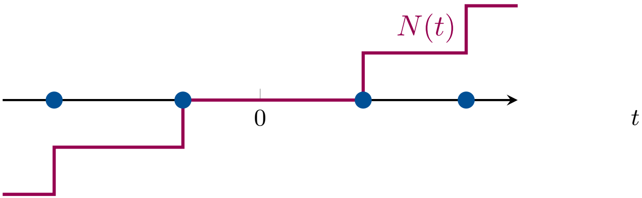
Let $N = \{T_k : k \in \mathbb{Z}\}$ be a stationary point process representing request times:



i.e. $N(B) = \sum_n \mathbf{1}_{\{T_n \in B\}}$ is a random counting measure.

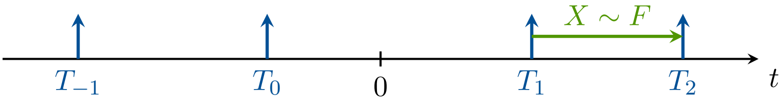
Counting process:

$$N(t) = \begin{cases} N([0,t]) & t \geqslant 0 \\ -N((t,0]) & t < 0 \end{cases}$$



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

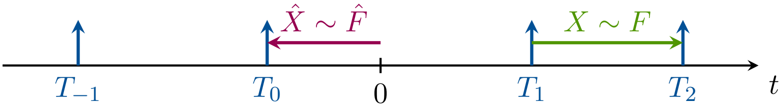
Two important distributions:



Inter-arrival distribution: $F(t) := P_N^0(T_1 - T_0 \leqslant t), \quad E_N^0[T_1] = 1/\lambda.$

Note: here P_N^0 is the Palm probability of the point process (conditioning on $T_0 = 0$).

Two important distributions:



Inter-arrival distribution: $$F(t) := P_N^0(T_1 - T_0 \leqslant t), \quad E_N^0[T_1] = 1/\lambda.$$

Age distribution: $$\hat{F}(t) := P(-T_0 \leqslant t) = \lambda \int_0^t 1 - F(s) ds,$$

Note: here P_N^0 is the Palm probability of the point process (conditioning on $T_0 = 0$).

Stochastic intensity

Consider a simple stationary point process N with intensity λ, 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 $\lambda(t) \geqslant 0$ is a stochastic intensity for the history $\mathcal{F}_t$ iff it is a.s. locally integrable, $\mathcal{F}_t-$adapted and:

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

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

Local interpretation:

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

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

Local interpretation:

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

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

Martingale interpretation:

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

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

Namely, $A(t) = N(a) + \int_a^t \lambda(s)ds$ is the compensator of the counting process.

Stochastic intensity of a Poisson process

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

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

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

Stochastic intensity of a Poisson process

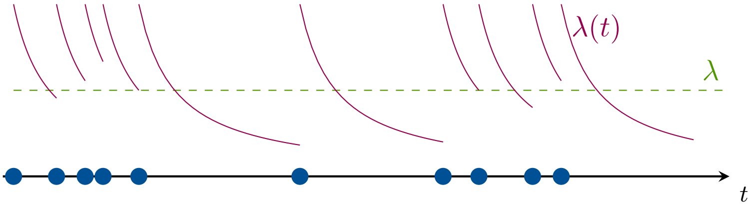
- If $N(t)$ is a Poisson process, then we know that

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

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

- In fact, this characterizes the Poisson process. The stochastic intensity $\lambda(t)$ is deterministic if and only if N is a Poisson process of (possible time-varying) intensity $\lambda(t)$.

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 N be a stationary renewal process, i.e. inter request times $T_{n+1} - T_n$ 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 N be a stationary renewal process, i.e. inter request times $T_{n+1} - T_n$ 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:

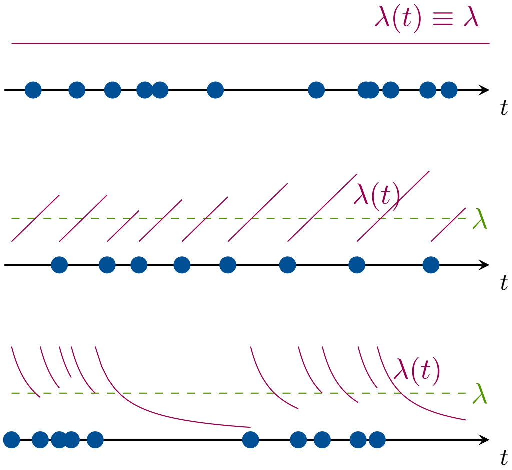
$$\lambda(t) = \eta(t - T^*(t)),$$

where

$$T^*(t) = \sup\{T_n : T_n < t\}$$

is the last point before t.

Some examples...



Constant hazard rate $\rightarrow$ Poisson process.

Increasing hazard rate $\rightarrow$ more periodic!

Decreasing hazard rate $\rightarrow$ more bursty!

Outline

Causal caching policies

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

Definition

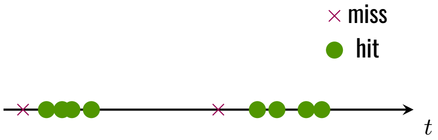
A causal caching policy is an $\mathcal{F}_t$ predictable stochastic process

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

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

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

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



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.

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

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

And its stochastic intensity is just:

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

The steady state hit rate of the policy is:

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

Maximizing the hit rate

In order to maximize λ_{hit}, consider the causal policy:

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

Then, for any causal policy π and for each realization:

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

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

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

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

Back to the Poisson case

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

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

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

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

The renewal case

- If now the N_i are renewal processes of (decreasing) intensities λ_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 N_i are renewal processes of (decreasing) intensities λ_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

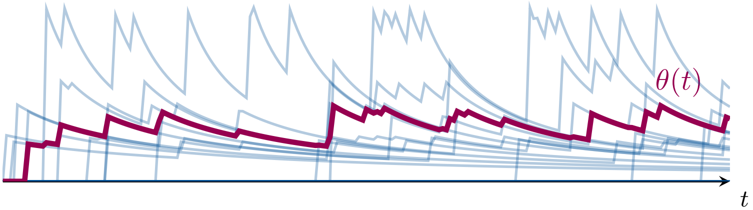
- If instead hazard rates are increasing, then when a request arrives, the item becomes less likely to be requested again!
- It may be better to remove it and make room for other ones (i.e. LRU makes no sense!).
- If we haven't seen it for a while, then we may have to fetch it anticipating the upcoming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

Outline

We can rewrite this optimal policy as a threshold policy:

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

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

$$X_i \sim \eta_i(-T_0^i), \quad -T_0 \sim \hat{F}_i(t)$$

- 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 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 $1/\lambda_i$.

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 $1/\lambda_i$.

In this case:

- The distribution of $-T_0^i$ 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 $G_0(x) = P(\eta_0(-T_0) \leqslant x)$ is the observed hazard rate distribution for the base process.

The distribution of popularities

Consider now the popularities $\lambda_1 > \ldots > \lambda_M$ and define:

$$\phi_M(\lambda) = \frac{1}{M} \sum_{i=1}^{M} \mathbf{1}_{\{\lambda_i \leqslant \lambda\}}$$

their empirical (deterministic) distribution.

The distribution of popularities

Consider now the popularities $\lambda_1 > \ldots > \lambda_M$ and define:

$$\phi_M(\lambda) = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{\{\lambda_i \leqslant \lambda\}}$$

their empirical (deterministic) distribution.

Assumption:

$$\phi_M(\lambda) \to \phi(\lambda) \quad \text{as } M \to \infty$$

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:

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

- Then we can show that:

$$\phi_M(\lambda) \to \phi(\lambda) = \left[1 - \lambda^{-1/\beta}\right] \mathbf{1}_{\{\lambda \geqslant 1\}}$$

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

Main result

Theorem (Carrasco,F',Paganini)

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

$$\hat{G}_M(x) = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{\{X_i \leqslant x\}} \to_M G_\infty(x) = \int_0^\infty G_0\left(\frac{x}{\lambda}\right)\phi(d\lambda)$$

A law of large numbers for the threshold

Assume further that the cache has capacity $C = cM$ 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 $C = cM$ with $0 < c < 1$ is the fraction of the catalog that can be stored.
Then, the optimal policy threshold $\theta^*_M(0)$ is the random variable:

$$\theta^*_M : \sum_{i=1}^{M} \mathbf{1}_{\{X_i \leqslant \theta^*_M\}} = (1-c)M$$

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

A law of large numbers for the threshold

Assume further that the cache has capacity $C = cM$ with $0 < c < 1$ is the fraction of the catalog that can be stored.
Then, the optimal policy threshold $\theta^*_M(0)$ is the random variable:

$$\theta^*_M : \sum_{i=1}^{M} \mathbf{1}_{\{X_i \leqslant \theta^*_M\}} = (1-c)M$$

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

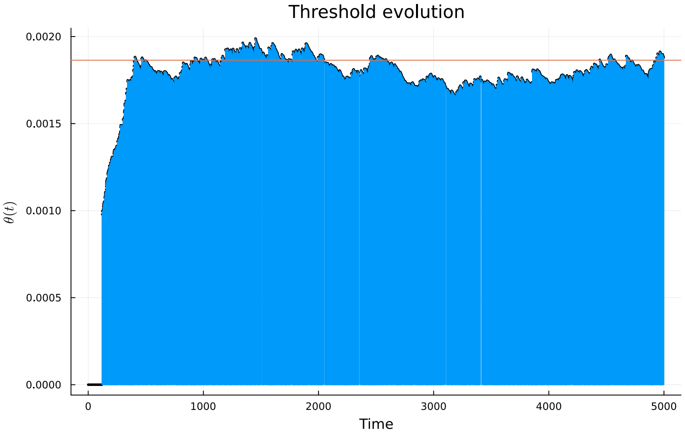
Corollary

If the cache size scales linearly with the catalog as $C_M = cM$, then:

$$\theta^*_M \to \theta^* : \ G_\infty(\theta^*) = 1 - c$$

So the optimal policy becomes a fixed threshold policy.

Simulation example



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

Asymptotic miss probability

Moreover, we can calculate the asymptotic performance:

Theorem

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

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

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

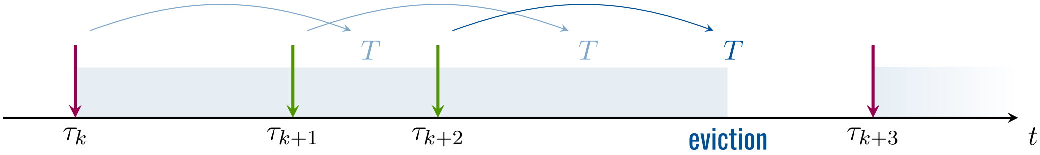
$$\tilde{G}_0(x) = \int_0^\infty \mathbf{1}_{\{\eta_0(t) \leqslant x\}} F_0(dt).$$

Outline

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 λ_i and inter-arrival distribution F_i:

Problem (Optimal TTL policy)

Choose timers $T_i \geqslant 0$ such that:

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

subject to:

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

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

The optimal timers

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:

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

for every stored content.

The optimal timers

Theorem

For the following cases, the optimal timers are:

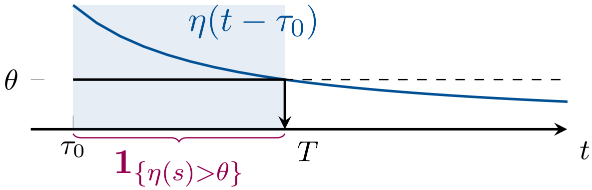
- Constant hazard rate (Poisson) or increasing hazard rate: keep the most popular objects ($T_i = \infty$ or 0).
- Decreasing hazard rate:

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

for every stored content.

So the optimal timer policy is a threshold policy?

Why this happens?



Asymptotic optimality

Theorem (F', Carrasco, Paganini)

In the scaling regime considered earlier, for renewal processes with DHR, the TTL policy is asymptotically optimal.

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

Asymptotic optimality

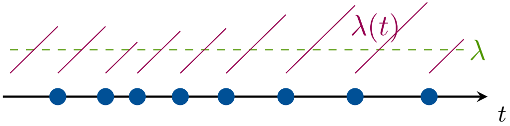
Theorem (F', Carrasco, Paganini)

In the scaling regime considered earlier, for renewal processes with DHR, the TTL policy is asymptotically optimal.

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

Back to increasing hazard rates...

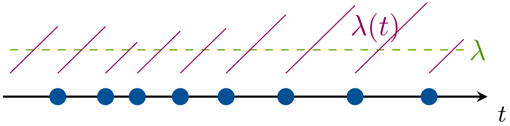
- Recall the increasing hazard rate behavior:



- 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

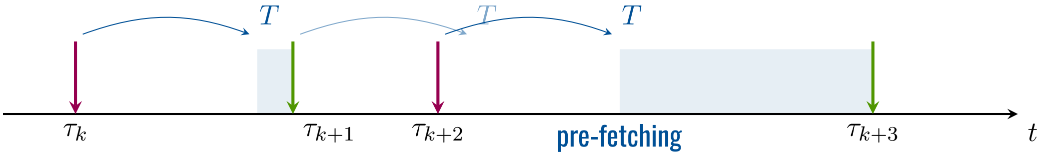
The question now is not how long we should remember something, but instead how long we should forget about it!

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!

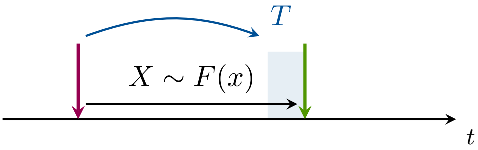
Timer based pre-fetching policy:



Timer based pre-fetching

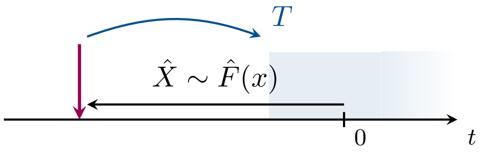
Consider a single item with a timer T and its request process:

Hit probability: next arrival occurs after timer expires.



Hit probability $= 1 - F(T)$

Occupation probability: probability that timer has expired by 0 since last arrival.



Avg. occupation $= 1 - \hat{F}(T)$

Choosing the optimal timers

Requests come from independent sources with intensities λ_i and inter-arrival distribution F_i:

Problem (Optimal pre-fetching policy)

Choose timers $T_i \geqslant 0$ such that:

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

subject to:

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

Choosing the optimal timers

Requests come from independent sources with intensities λ_i and inter-arrival distribution F_i:

Problem (Optimal pre-fetching policy)

Choose timers $T_i \geqslant 0$ such that:

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

subject to:

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

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:

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

for every stored content.

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

Asymptotic optimality

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.

Outline

Final remarks

- 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.

Final remarks

- 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! Students Welcome!.

Thanks!

Andres Ferragut
ferragut@ort.edu.uy
aferragu.github.io