Caching or pre-fetching? The role of hazard rates

Andres Ferragut

joint work with Matias Carrasco and Fernando Paganini

Universidad ORT Uruguay

SPOR Seminar – Eindhoven University of Technology – December 2023

Outline

Outline

Problem formulation

Deriving the optimal timer policy

Optimal causal policy

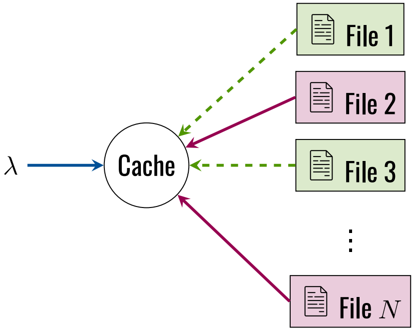
Asymptotic equivalence and optimality

Timer based pre-fetching

Conclusions

The caching problem

- Consider a cache system with a catalog of N objects.
- Requests for objects arrive at random at rate λ.
- The cache can locally store $C < N$ of them.
- If item is in cache, we have a hit.

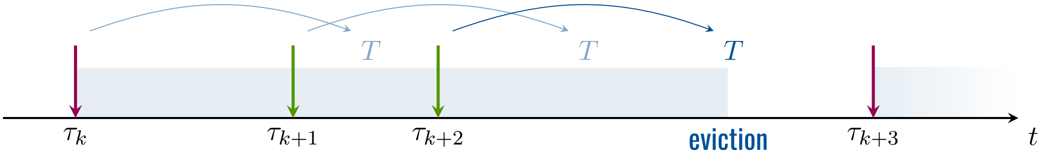


Objective: for a given arrival process, maximize the steady-state hit probability.

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.



Request arrival model

The classical arrival model is the independent reference model:

- Requests arrive as a Poisson process of intensity λ.
- Request is for item i with probability p_i (popularity).
- Poisson thinning: each request process is Poisson λp_i.
- Succesive requests are independent with distribution $(p_i : i = 1, \ldots, N)$.

- Problem: caches work best when requests are bursty, i.e. successive requests are correlated.
- However, under the IRM we have purely random requests.

Point process approach [Fofack et al. 2014]:

- Assume requests for item i come from a point process of intensity $\lambda_i := \lambda p_i$.
- If inter-request times are heavy tailed, this can model burstiness.

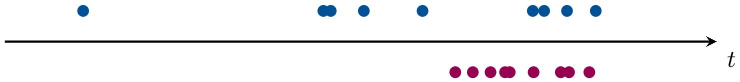
Example: Pareto arrivals

Consider two items, with equal popularity...

- Poisson arrivals:

Homogeneous

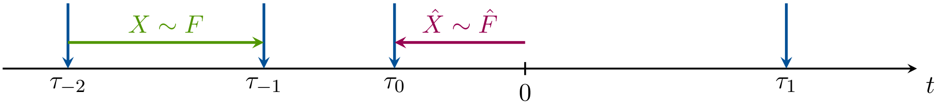
- Heavy tailed arrivals (Pareto $\alpha = 2$):



Bursty!

A bit of point process theory...

Let $N = \{\tau_k : k \in \mathbb{Z}\}$ be a stationary point process representing requests from an item:



Inter-arrival distribution:

$$F(t) := P_N^0(\tau_1 - \tau_0 \leqslant t)$$
$$E_N^0[\tau_1] = 1/\lambda.$$

Age distribution:

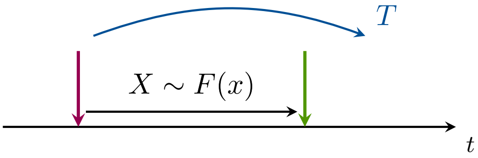
$$\hat{F}(t) := P(-\tau_0 \leqslant t) = \lambda \int_0^t (1 - F(s))ds,$$

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

Application: TTL policies

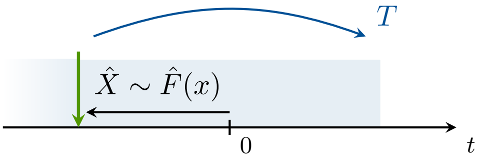
Consider a single item with a timer T and its request process:

Hit probability: next arrival occurs before timer expires.



Hit probability $= F(T)$

Occupation probability: probability that timer hasn't expired by 0 since last arrival.



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

Outline

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.

Choosing the optimal timers

Idea: Change of variables $u_i = \hat{F}_i(T_i)$ (occupation).

Problem (Optimal TTL policy)

Choose timers $T_i = \hat{F}_i^{-1}(u_i)$ such that:

$$\max_{u_i \in [0,1]} \sum_i \lambda_i F_i(\hat{F}_i^{-1}(u_i))$$

subject to:

$$\sum_i u_i \leqslant C$$

The hazard rate function

Define $G_i(u) := \lambda_i F_i(\hat{F}_i^{-1}(u))$, then:

$$\frac{\partial G_i}{\partial u} = \lambda_i f_i(\hat{F}_i^{-1}(u)) \frac{\partial}{\partial u} \hat{F}_i^{-1}(u) = \frac{\lambda_i f_i(\hat{F}_i^{-1}(u))}{\lambda_i(1 - F_i(\hat{F}_i^{-1}(u)))} = \eta_i(T_i)$$

where $\eta_i(t)$ is the hazard rate function of the inter-arrival distribution:

$$\eta_i(t) := \frac{f_i(t)}{1 - F_i(t)}$$

Idea: the hazard rate measures the probability that we have a request at time t, given that the current interval is larger than t.

Increasing hazard rates

Poisson arrivals: constant hazard rate (memoryless property), $\eta_i(t) \equiv \lambda_i \to$ objective is **linear**.

Increasing hazard rates: $\eta_i(t)$ increasing (more regular traffic) $\to$ objective is **convex**!

Optimal TTL policy, constant or IHR, [F',Rodriguez, Paganini 18].

In both cases, the optimal TTL policy is static:

$$T_i^* = \infty, \quad (u_i^* = 1) \quad \text{for the } C \text{ contents with higher } \lambda_i$$

Decreasing hazard rates

- The decreasing hazard rate case corresponds to heavy tails and thus more bursty traffic $\rightarrow$ where caching is more useful!

- If $\eta_i(t)$ is decreasing, objective is concave, we have a non-trivial optimum:

$$\mathcal{L}(u,\mu) = \sum_i \lambda_i F_i(\hat{F}^{-1}(u_i)) - \mu \left(\sum_i u_i - C \right)$$

- KKT conditions:

$$\eta_i(\hat{F}^{-1}(u_i^*)) = \eta_i(T_i^*) \geqslant \mu \quad \forall i, \quad \mu \left(\sum_i u_i^* - C \right) = 0$$

Decreasing hazard rates

Optimal TTL policy, DHR, [F',Rodriguez, Paganini 18].

The optimal TTL caching policy for DHR is such that:

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

for every stored content.

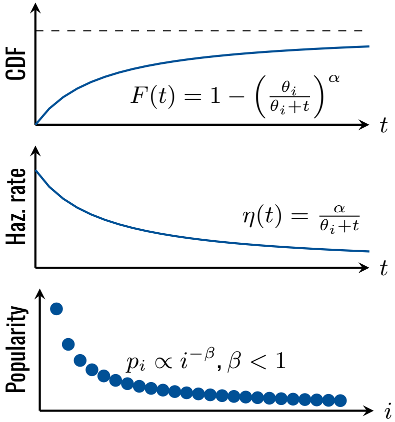
Idea: we have a fixed memory budget to allocate. $\eta_i(T_i)$ is the marginal increase in hit rate (utility) for enlarging the timer T_i.

Optimal allocation: equalize marginal utilities.

Parametric heavy tailed case

- For Pareto arrivals and Zipf popularities you can obtain a nice fluid limit.
- Let N go to ∞ and $C = cN$, then u_i^* has a functional limit.
- The hit probability is given by [FRP '18]:

$$H^* = (1-\beta)\int_0^1 x^{-\beta}\left[1-(1-u^*(x))^{\frac{\alpha}{\alpha-1}}\right]dx,$$



Conclusions so far...

- The hazard rate function of F plays a crucial role in determining the optimal TTL policy!
- For IHR: just store the most popular content.
- For DHR: proper optimization problem, equalize hazard rates.
- Asymptotic analysis has explicit expressions.

Outline

Replacement policies

- Assume now that you have a fixed capacity C. We have to decide which contents to store.
- Naïve idea: just keep the C most popular ones (higher λ_i). Can we do better?
- Another idea: Least-recently-used (discard from the cache the oldest request).

Replacement policies

- Assume now that you have a fixed capacity C. We have to decide which contents to store.
- Naïve idea: just keep the C most popular ones (higher λ_i). Can we do better?
- Another idea: Least-recently-used (discard from the cache the oldest request).

Problem

Given some independent stationary request processes with intensities λ_i, what is the optimal causal policy?

Idea: we should keep track of some local notion of intensity!

Stochastic intensity

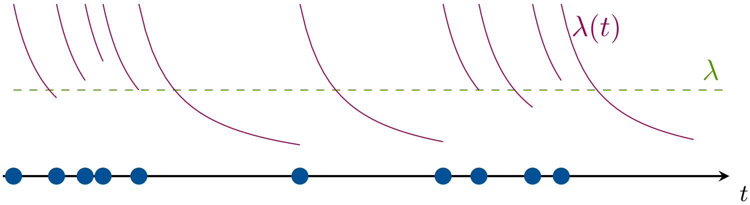
Consider a simple stationary point process N with intensity λ, defined in some probability space $(\Omega, \mathcal{F}, P)$. Let some filtration $\{\mathcal{F}_t\}_{t\in\mathbb{R}}$ be a history of the process.

Define the stochastic intensity $\lambda(t)$ of N as:

$$\lim_{h\to 0} \frac{1}{h} E[N((t, t+h]) \mid \mathcal{F}_t] = \lambda(t) \quad P - a.s.,$$

Idea: If the process is simple (isolated points), $E[N((t, t+h]) = \lambda h + o(h)$, so the average stochastic intensity is λ. But given the history, the value of $\lambda(t)$ may change.

If traffic is bursty, the stochastic intensity rises near arrivals:



Stochastic intensity of a renewal process

- Let now N be a renewal process $\to$ inter-request times are $iid \sim F$.
- Let $\mathcal{F}_t$ be the natural history of the process (i.e. the information of points up to t).

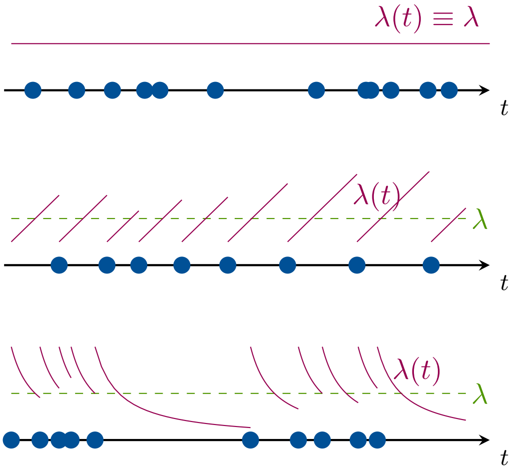
Theorem (cf. Brémaud 21)

Let $\eta(t) := f(t)/(1 - F(t))$ be the hazard rate function of F. Define:

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

where τ_t^* is the last point before t. Then $\lambda(t)$ is a stochastic intensity for $(N, \mathcal{F}_t)$.

Some examples...



Constant hazard rate $\rightarrow$ Poisson process.

Increasing hazard rate $\rightarrow$ more periodic!

Decreasing hazard rate $\rightarrow$ more bursty!

Causal caching policies

- Consider a cache system fed by N independent renewal processes.
- Let $\mathcal{F}_t = \sigma(\{\mathcal{F}_t^i : i = 1, \ldots, N\})$ their aggregate history.

Definition

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

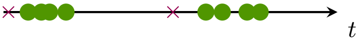
$$\mathcal{C} : \Omega \times \mathbb{R} \to 2^{\{1,\ldots,N\}}$$

i.e. $\mathcal{C}(t) = \{i_1, \ldots, i_C\}$ is the subset cached 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:

$$H_i(B) = \sum_{n \in \mathbb{Z}} \mathbf{1}_{\{\tau_n^i \in B\}} \mathbf{1}_{\{i \in \mathcal{C}(\tau_n^i)\}}$$



Since $\mathbf{1}_{\{i \in \mathcal{C}(\tau_n^i)\}}$ is $\mathcal{F}_t$ predictable, its stochastic intensity is:

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

i.e., $h_i(t) = \lambda_i(t)$ while $i \in \mathcal{C}(t)$ and otherwise 0.

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:

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

The hit rate and hit probabilities of the policies are given by:

$$\text{hit rate} = \lambda_H := E[h(t)], \quad \text{hit probability} := \frac{\lambda_H}{\lambda}.$$

Maximizing the hit rate

In order to maximize λ_H, consider the policy:

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

Then, for any non-anticipative policy and for each realization:

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

Theorem (Towsley et al. '22)

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

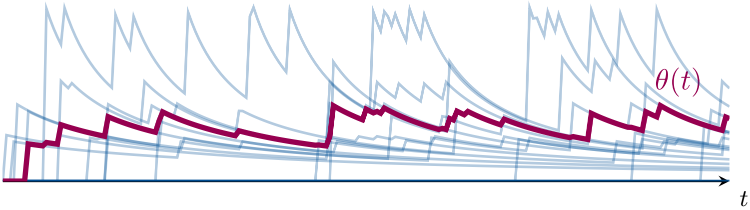
Outline

The threshold process

We can rewrite this optimal policy as a threshold policy:

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

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



We want to understand $\theta(t)$.

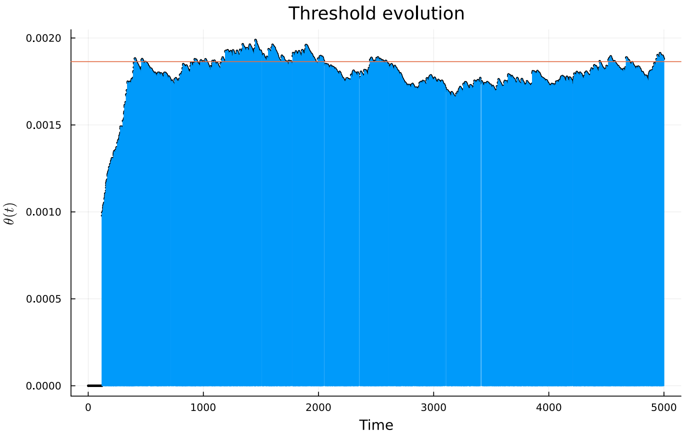
Asymptotic equivalence result

Theorem [F', Carrasco, Paganini, three weeks ago...]

Consider a cache system fed by N independent renewal processes with DHR inter-arrival times, and the optimal non-anticipative policy. Let $N \to \infty$ with $C = cN$. Then, in steady state:

- The (appropriately scaled) threshold $\theta_N(t)$ converges almost surely to a constant θ^*.
- θ^* is the dual value of the optimal TTL policy, i.e. the value that equalizes hazard rates.
- If popularities are slowly decaying (i.e. $\beta < 1$) then the hit probability of the optimal policy converges to H^*, the hit probability of the optimal TTL policy.

Simulation example

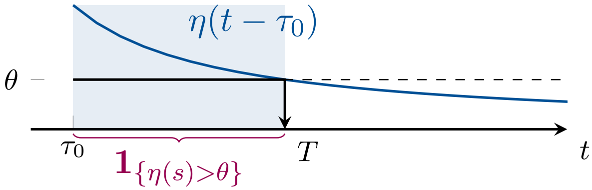


$N = 1000, C = 100$. Pareto $\alpha = 2$ requests, Zipf $\beta = 0.5$ popularities.

Why this happens?

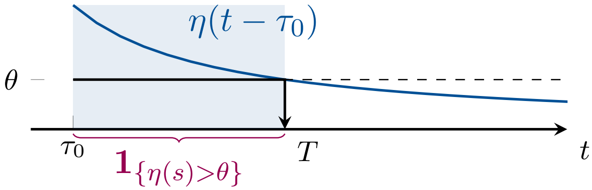
Why this happens?

Because, for decreasing hazard rates, the TTL policy is also a threshold policy!



Why this happens?

Because, for decreasing hazard rates, the TTL policy is also a threshold policy!

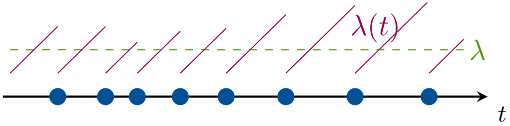


Key idea: replace the timer T_i by $\theta_i = \eta_i^{-1}(T_i)$, the corresponding hazard rate at the timer.

Outline

Back to increasing hazard rates...

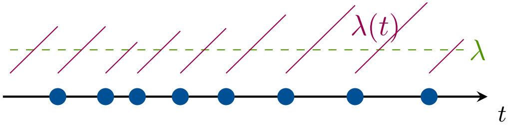
- Recall the increasing hazard rate behavior:



- 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

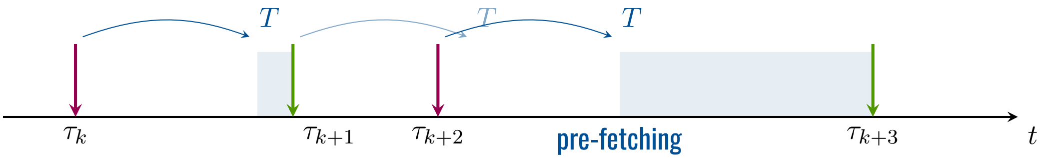
The question now is not how long we should remember something, but instead how long we should forget about it!

Timer based pre-fetching policies

Key insight

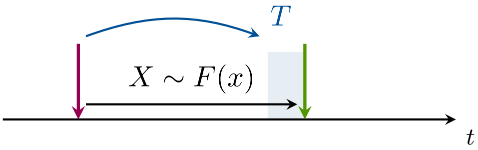
The question now is not how long we should remember something, but instead how long we should forget about it!

Timer based pre-fetching policy:



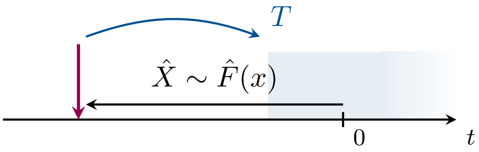
Consider a single item with a timer T and its request process:

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, last week...].

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

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

for every stored content.

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

Ongoing work: use this pre-fetching threshold policy to prove that in the fluid limit, the optimal causal policy is a timer-based pre-fetching policy.

Outline

Key takeaways...

- We analyzed two types of caching policies: TTL and replacement.
- We identified the hazard rate function as a crucial component of optimal policies.
- Using the point process framework, we can model burstiness and exactly compute asymptotics for TTL policies.
- We provide a large scale equivalence result for the optimal causal policy and the optimal TTL policy, enabling us to compute universal bounds on asymptotic performance!

Key takeaways...

- For IHR (more regular) traffic, caching is not a good idea!
- Instead, in order to use the information about arrivals, it is better to pre-fetch the content after some time.
- We derived the optimal timer based pre-fetching policy and expect to prove a similar equivalence result with the optimal causal policy!

Thank you!

Andres Ferragut
ferragut@ort.edu.uy
http://aferragu.github.io