Catalog Dynamics: Impact of Content Publishing and Perishing on the Performance of a LRU Cache

Felipe Olmos*, Bruno Kauffmann†, Alain Simonian† and Yannick Carlinet†
*Orange Labs and CMAP, Email: *luisfelipe.olmosmarchant@orange.com*
†Orange Labs, Email: *firstname.lastname@orange.com*

***Abstract*—The Internet heavily relies on Content Distribution Networks and transparent caches to cope with the ever-increasing traffic demand of users. Content, however, is essentially versatile: once published at a given time, its popularity vanishes over time. All requests for a given document are then concentrated between the publishing time and an effective perishing time.**

In this paper, we propose a new model for the arrival of content requests, which takes into account the dynamical nature of the content catalog. Based on two large traffic traces collected on the Orange network, we use the semi-experimental method and determine invariants of the content request process. This allows us to define a simple mathematical model for content requests; by extending the so-called "Che approximation", we then compute the performance of a LRU cache fed with such a request process, expressed by its hit ratio. We numerically validate the good accuracy of our model by comparison to trace-based simulation.

I. Introduction

Driven by video streaming, Internet data traffic is rapidly growing, up to 41% at the busy hour in 2012 according to a Cisco forecast. Content delivery networks (CDNs) are now a key component of the Internet architecture and play a central role in coping with such a demand. By means of caching and duplicating content near the end-users, CDNs provide an Internet experience with high performance and high availability. Additionally, as the cost of memory decreases faster than that of bandwidth, Internet Service Providers (ISPs) also locally resort to transparent caching to decrease the load on specific expensive links. This favorable bandwidth-memory trade-off has been confirmed by recent research [3], [9], [12]. As most practical replacement policies have a behaviour similar to that of Least-Recently-Used (LRU), we here follow [9] in using the LRU replacement policy as a representative one.

Video delivery is now the majority of traffic at the busy hour, simultaneously driven by User-Generated Content (UGC) traffic and by Video-on-Demand (VoD) services. YouTube, the best known of UGC sites today, has indeed emerged as an hyper-giant among the Content Providers, serving up to 25% of the traffic at the busy hour in ISP networks. On the other side, Video-on-Demand is growing rapidly, as examplified by the development of NetFlix. Understanding the performance of caches for video delivery becomes therefore crucial for network provisioning and operation. It allows simultaneously to decrease the network load, reduce dimensioning needs and decrease peering costs.

The versatility of content, however, raises a new challenge. As new content is continuously published, and (part of) the old content becomes outdated and non-popular, the popularity of a given document dynamically evolves with time. The dynamics of the content catalog has significant implication for caching performance. First, even with infinite memory, caches cannot manage to serve every request: in fact, the first request for any document will obviously not find the content present in the cache. Secondly, the request traffic is not stationary and caches never experience a steady state; indeed, the set of documents which are currently stored in the caches slowly evolves with time, as new content replaces the older one.

Moreover, as detailed in Section IV, the stationarity periods of the content requests process prove to be short. Consequently, due to the heavy tail of content popularity distributions, estimating the popularity of content during such a short period leads to significant variance. Additionally, the cache may not reach its steady state over short periods, and its performance will therefore depend on the recent past. Characterizing the cache performance at the busy hour is therefore a difficult task due to such inherent variance. This consequently leads us to express the cache performance as the long term average hit ratio, which estimates the average dimensioning gains and fully characterizes the peering gains.

In this paper, we provide a first answer to that dynamicity issue. Through basic manipulations, hereafter called *semi-experiments*, on two large traces of YouTube and VoD traffic collected from the Orange network, we determine the key invariants of the video request process and propose a model which captures them. This model is amenable to mathematical analysis: Using the so-called Che approximation [6], we express the hit ratio of a LRU cache fed by such a request process as a function of basic document statistics. We finally show via simulation that this approximation accurately matches the empirical hit ratio.

Our key findings are the following: ***(i)*** The document arrival process can be well represented by a Poisson point process; ***(ii)*** The document requests process can be well represented by a Poisson-Poisson cluster process; ***(iii)*** The hit ratio can be expressed in terms of the distribution of document request intensities and document lifespans only.

The remainder of the paper is organized as follows. Section II presents related work. Section III describes the dataset and the statistics drawn from traces. In Section IV, we apply the semi-experiment methodology and determine the structural invariants which are relevant for caching. Based on these observations, we then build a model (Section V) for the request process and estimate the hit ratio for a LRU cache. That estimate is applied to both YouTube and VoD traces and successfully compared to the empirical hit ratio in Section VI.

II. Related Work

We describe two areas of related work: content-level traffic characterization and cache performance analysis.

The popularity distribution of documents has been extensively discussed since the 90's. It has been shown to exhibit a light-tailed behavior (typically from a Weibull distribution) when considering the total number of requests for documents over a long period, and a heavy-tailed behavior (typically a Zipf distribution) when analyzing the viewing rate of documents (see [4], [5], [13], [19] for recent references).

The temporal pattern of document requests has also been studied. In [22], authors propose a Markov model with short term memory for document requests, but the set of available documents remains fixed. The lifespan of documents (defined as the time elapsed between the first and the last requests) has been studied in [7], [8]. Similarly, articles [19], [23] show that only a small portion of the documents are active at any moment, and that popular documents have a significant turnover. Document arrivals into the catalog are identified in [13] and their impact on the maximum achievable hit ratio is discussed. The distribution of requests for the same document over its lifetime is studied in [5], [8], but the results are aggregated over all documents, and therefore do not lead to a model for the request process. To the best of our knowledge, [24] is the single prior work aiming at a full description of the request process. Due to the short duration of the studied traces, however, the inclusion of catalog dynamics is rather simplistic in the proposed model. Finally, [12] proposes a simple model for the dynamics of documents which are requested only once, but the set of documents with several requests remains fixed. In terms of traffic characterization, the closest related work to ours is [14], although the latter focuses on packet-level traffic characterization. In particular, the authors introduce the so-called "semi-experimental methodology" that we use in section IV.

As regards cache performance literature, we here only report the recent literature focusing on the analytical characterization of the performance of caches applying to LRU policy. An asymptotic analysis of the LRU miss rate for either Zipf or Weibull requests distribution, with simple closed-form formulas, is provided in [15]. Che et al. [6] propose another approximation, which is asymptotically exact for a Zipf popularity [16] and also accurate for other types of distributions [11].

We are aware of a few works which do not assume i.i.d. request sequences. In particular, [17] studies the performance of a LRU cache fed by correlated requests, where the instantaneous request distribution depends on a stationary modulating Markov process; the asymptotic performance is identical to that under IRM, showing that such a short-term correlation does not fully capture the content dynamics. [21] also estimates the performance of a LRU cache when the requests form a Markov chain, but no closed-form formula is provided. [10] and [18] provide a theoretical analysis of a network of LRU cache, when requests for a given document form an arbitrary renewal process; correlation among requests can thus be incorporated, but the catalog of document remains static.

Reference [12] is the first published paper to address, though in a limited way, the dynamics of content catalog. It provides an asymptotic performance formula when requests for popular documents follow a Zipf law under IRM, and an exogenous stream of unique requests for an infinite set of "noise" documents is added. Finally, recent developments [1], [2], though not yet refereed, share the core intuitions of our paper; they also model the document request arrivals using a shot-noise (or Poisson cluster) process, where the document popularity profile is parametrized by both its average popularity and its lifespan. We differ, however, from these papers by the semi-experiment methodology that we use in section IV and which justifies our proposed model. We also validate our results on two different traces, corresponding to the traffic profiles of YouTube and Video-on-Demand traffic. Finally, the long duration of our traces (respectively, 3 months and 3.5 years) enables us to better emphasize the impact of temporal locality on the request process.

III. Dataset

A. Data Collection

We have gathered two datasets from two services, which have different traffic profiles. The first dataset, hereafter named **YT**, captures YouTube traffic of Orange customers located in Tunisia. We have access to the logs of a transparent caching system set up in order to offload the country's international connection. This system is a commercial product from a large company specialized in the design and management of CDNs. In the observation period of January–March 2013, we collected around 420 millions of chunk requests from about $40\,000$ IP addresses to more than 120 million chunks. For each chunk request in this trace, the logs contain the user (anonymous) IP address, a video identifier, the timestamp of the end of session, the number of transmitted bytes, the duration of the HTTP connection and the beginning and ending position of the specific *chunk* requested, the latter information being available for 96% of the data.

The second dataset, hereafter called **VoD**, comes from the Orange Video-on-Demand service in France. This service proposes to Orange customers both free catch-up TV programs, pay-per-view films and series episodes. Probes deployed at the access of the service platforms recorded video requests from June 2008 to November 2011. The data amounts to more than $3\,400\,000$ requests from $60\,000$ users to $120\,000$ videos. The records in this trace consist of the request timestamp, an internal client (anonymous) identifier and a video identifier.

B. Processing

For simplicity and mathematical tractability, we focus our analysis at the document level rather than chunk level.

Since the YT trace consist of chunk requests, we consolidate them to identify the user video sessions. To this aim, we first identify the requests from a single user to a single object. Then, when the chunking information is available, we simply concatenate the requests corresponding to a chunk chain. For the requests without chunk identification, we aggregate all the requests made by the same user for the video, and with inter-arrival time smaller than 8 minutes. This threshold corresponds to the 95% percentile of the length session distribution of

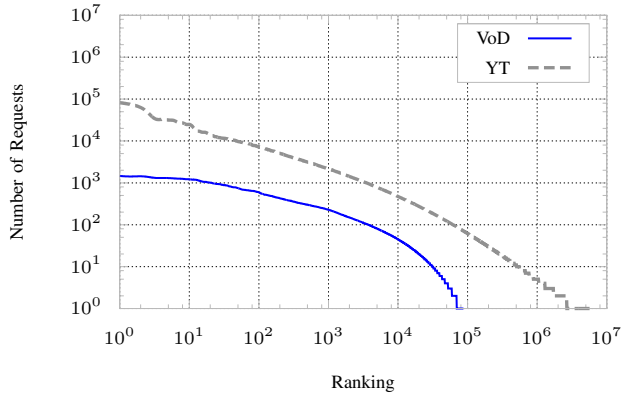


Fig. 1: *Number of requests as a function of their rank.*

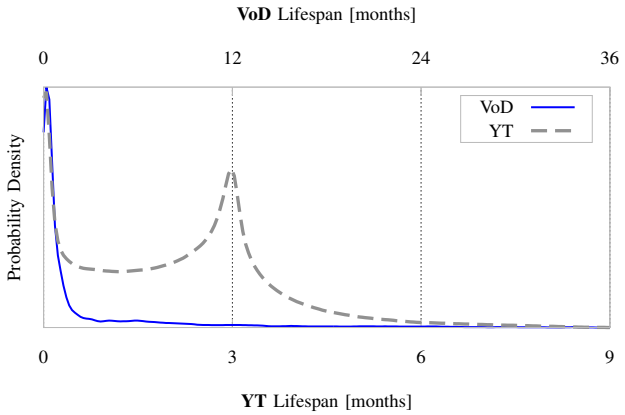


Fig. 2: *Kernel density estimate for lifespan τ.*

requests with chunk data.[1] The result of this procedure is our working YT dataset consisting of more than 46 000 000 requests to around 6 300 000 unique documents.

In the case of the VoD trace there was no need of the above consolidation procedure. Nonetheless, the trace contained requests to movie trailers or to full content but with short duration. We consider these content "surfing" requests not relevant in terms of caching performance and thus we discarded them from the VoD dataset. The working VoD dataset contains around 1 800 000 requests to more than 87 000 different objects.

C. Distribution of the Number of Requests

The logarithmic rank-frequency chart in Figure 1 shows two different popularity behavior for the traces. As expected, in the "short" YT trace, the 10 000 most popular documents follow a Zipf distribution with exponent 0.61, while the tail has an exponent of 1.03. As for the VoD trace, the popularity does not follow a power law, but is best fitted by a Weibull (also called Stretched-Exponential) distribution.

D. Distribution of Lifespan and Request Rate

We here provide a finer characterization that considers not only the number of requests to a given document, but also the period of time during which the document is active. For a given document, let τ denote its lifespan, that is, the period where the users can address requests to it; let then λ be the corresponding average request rate to that document. We estimate these two quantities that form the basis of our analysis.

Specifically, consider a given document with $n \geq 2$ requests, and let $\Theta_{\mathrm{I}} < \Theta_{\mathrm{F}}$ be the Initial and Final request times of that document in the observation window. We then estimate the catalog lifespan τ by the unbiased estimator

$$\widehat{\tau} = (\Theta_{\mathrm{F}} - \Theta_{\mathrm{I}}) \times \frac{n+1}{n-1} \tag{1}$$

(in fact, assuming that the n request times Θ_1, ..., Θ_n are uniformly distributed on the interval with length τ, we easily calculate $\mathbb{E}\left[\Theta_F - \Theta_I\right] = \frac{n-1}{n+1} \times \tau$).

[1]The reason to select a percentile instead of the maximum is that there are 185 chunk chains with extreme duration (in the order of days or even months).

Regarding intensity λ, our sample is biased by the fact that we collect only documents with at least one request. To take this bias into account, we assume that n is a Poisson random variable with mean $\lambda\tau$, given $n \geq 1$. We thus estimate the request rate λ by

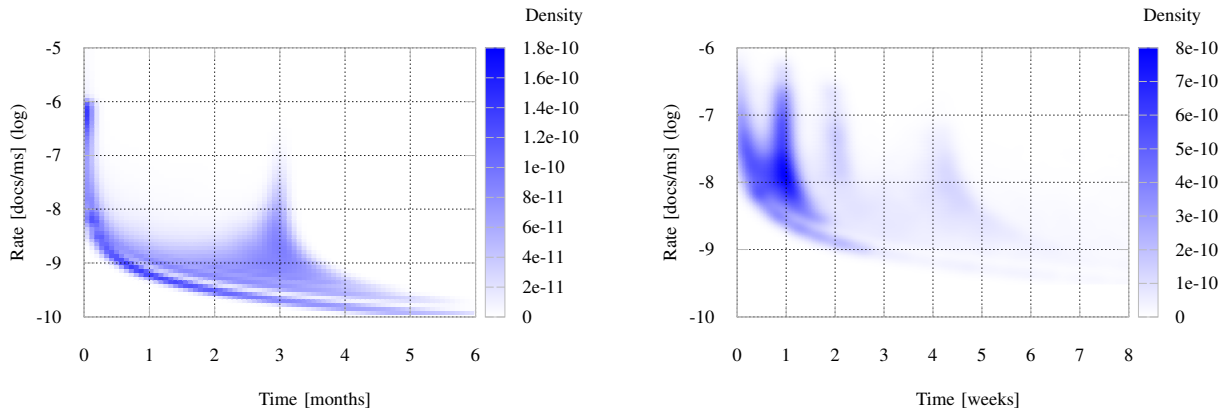
$$\widehat{\lambda} = n'/\widehat{\tau} \tag{2}$$

where n' verifies equation $n'/(1 - e^{-n'}) = n$ (the latter has a unique positive solution n', and verify $n' \approx n$ for n greater than 10). Both estimators $\widehat{\tau}$ and $\widehat{\lambda}$ are valid only for documents for which we have at least two requests. Consequently, in the remaining of this section, we will make the analysis over the set of documents that verify the latter condition.

Figure 2 shows a kernel density approximation of $\widehat{\tau}$ for each dataset. Note that the formula of $\widehat{\tau}$ allows a positive density for values larger thant the observation window, especially for documents with a small number of requests. Also, in the YT data, we observe a probability mass accumulation effect near the mark of three months, which is precisely the size of the observation window. This is a truncation effect and it is a sign that the lifespan of a video may be far longer than our current observation window in this dataset. As regards the VoD data, most documents have a lifespan shorter than one month. This corresponds to the numerous catch-up TV programs. The remaining documents have a different distribution, with lifespans varying on the range of a few weeks to the observation period (3.5 years). Due to the large observation period, the truncation effect is not visible.

As regards $\widehat{\lambda}$ (figures are here omitted for spatial constraints), the distribution of $\log\widehat{\lambda}$ shows a behavior similar to that of a Gamma distribution in the YT dataset and a Weibull distribution in the VoD dataset, but still differs numerically from this family. This suggests that, in our traces, the random variable λ has a heavy tailed distribution.

The density estimation for the pair $(\log\widehat{\lambda}, \widehat{\tau})$ is shown in Figure 3, with a focus on small values of $\widehat{\tau}$ for the VoD data (the probability mass is concentrated on the darkest areas). In both cases, we conclude from the empirical densities that τ and λ are not independent random variables. In fact, in YT dataset for example, marginal distributions for $\widehat{\tau}$ and $\log(\lambda)$ have significant masses at $\widehat{\tau} = 0.1$ and $\log(\widehat{\lambda}) = -9.0$; no mass,



(a) *YT trace.*

(b) *VoD trace.*

Fig. 3: *Density kernel estimate for the pair* $(\tau, \log \lambda)$.

however, is observed in Figure 3a for the pair $(0.1, -9.0)$. Besides, the presence of managed catch-up TV documents in the VoD data is visible in Figure 3b, $\widehat{\tau}$ showing density peaks at values 1, 2 and 4 weeks, corresponding to the availability durations of programs. Finally, as the joint distribution is not easy to fit in both YT and VoD cases, we will use the empirical joint distribution in the following.

IV. Semi-Experiments

In this section, we identify the structural properties of the request process that are relevant to LRU caching, namely:

(i) Overall correlation between requests.
(ii) Correlation in the catalog publications.
(iii) Correlation between the requests of a document.

Additionally, in the case of the first property, we look for the timescale where it starts to influence the performance of LRU.

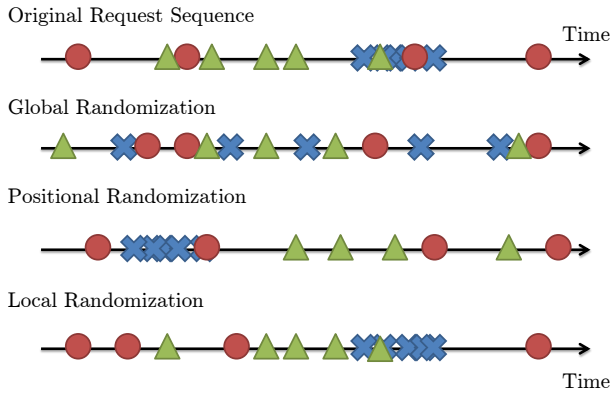


Fig. 4: *A schematic view of all three randomizations.*

To this aim, we use the semi-experimental methodology [14]. Each semi-experiment is based on two procedures: The first one is to randomly rearrange the original request sequence in a way that destroys a specific correlation structure; the second one is to use this new trace as an input for a simulation of a LRU cache and compute the corresponding hit ratio curve. We then look at the discrepancies from the hit ratio curve of the original trace; if they differ significantly we infer that the broken structure is relevant to LRU caching. In the following, we explain in detail each semi-experiment and its findings.

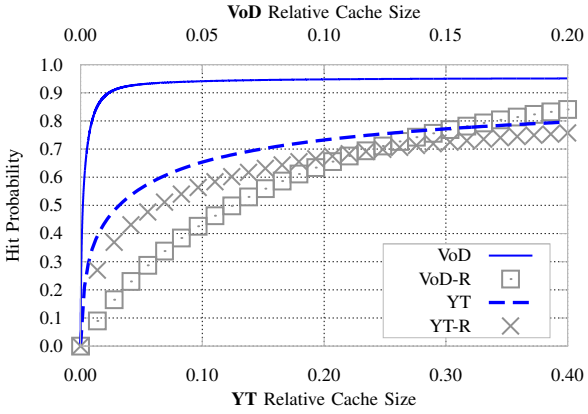
(i) Overall Correlation Between Requests: In this semi-experiment, we completely break the correlation structure of the request sequence by placing each request at an i.i.d. uniform time in the interval $[0; W]$, where W is the size of the observation window. Any request sequence shuffled in this manner leads to a IRM sequence, since the process destroys any dependence structure. We call this procedure *global randomization* and show an example in Figure 4.

In Figure 5a, we compare the resulting hit ratio to that obtained with the original trace and observe that the hit ratio of the latter is lower for any cache size in both datasets, but notoriously in the VoD case. More precisely, we compute the *mean absolute relative error* or MARE[2] between hit ratio curves of the original and randomized sequence. In the YT case, the MARE has a value of 5.0%; this value might seem low, but it comes mostly from the left of the curve. Since the left part of the curve is where practical cache sizes lie, this discrepancy is still important. As for the VoD trace, the MARE amounts to 17.3% which confirms the huge difference observed in Figure 5a. We thus conclude that the correlation between requests is a meaningful factor for the performance of LRU caching and that the IRM assumption leads to an underestimation of the hit ratio, which can be very significant.

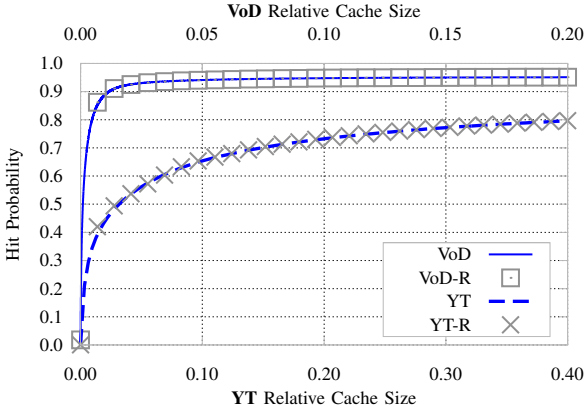
(ii) Correlation in Catalog Publications: We now examine how sensitive is our data with respect to the publication of new

[2]The MARE between a model sequence $(y_i)_{1 \le i \le N}$ and empirical data $(x_i)_{1 \le i \le N}$ is defined as $\frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - y_i|}{|x_i|}$.

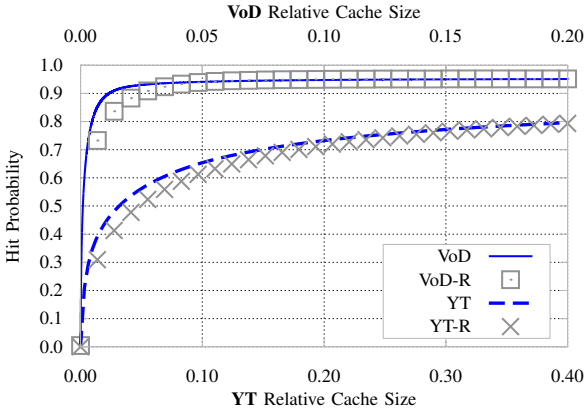
documents to the catalog. To this aim, we perform a *positional randomization*, which breaks the correlation structure between the first requests of documents, which we use as an estimate of the publication time. The procedure consists, for a given document, in leaving the inter-arrival times of its request sequence unchanged and jointly shift all of them by a random quantity, as shown in Figure 4. More precisely, let $\Theta_1, \Theta_2, \ldots, \Theta_k$ the request times for a document; first, we draw a uniform random number U from the interval $[0, W - (\Theta_k - \Theta_1)]$, then we define the new request sequence $\Theta_1^*, \Theta_2^*, \ldots, \Theta_k^*$ by $\Theta_i^* = U + \Theta_i - \Theta_1$ for $1 \leq i \leq k$.



(a) *Global Randomization*



(b) *Positional Randomization*



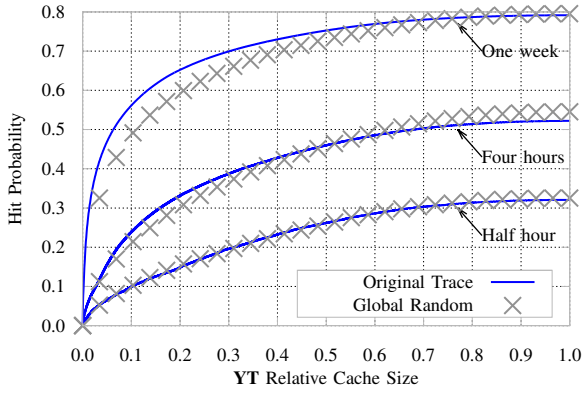
(c) *Local Randomization*

Fig. 5: *Comparison of the hit ratio of the original request sequence versus the results of each randomization.*

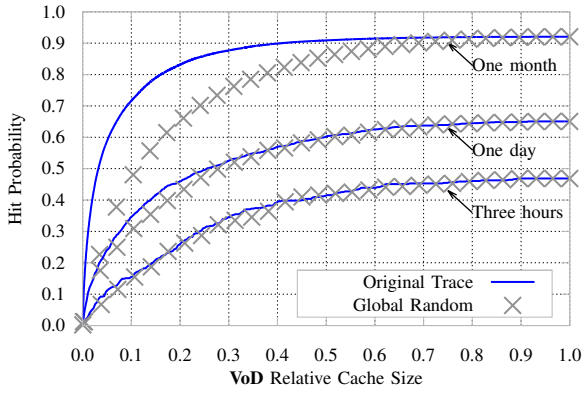
In both traces, the resulting hit ratio obtained shows no difference from the original, as observed in Figure 5b. Indeed, the MAREs in this semi-experiment are merely 0.3% in the YT case and 0.1% in the VoD case. We therefore conclude that document arrivals have no correlation structure with significant impact on caching.

(iii) Correlation between Requests of a Document: In this semi-experiment, we aim to break the request dependence structure for each document. To achieve this, we perform a *local randomization*: For a given document, we keep its first and the last request times fixed and only shuffle the ones in between at i.i.d. times following a Uniform $[\Theta_1, \Theta_k]$-distribution. Note that this procedure preserves the lifespan and intensity statistics discussed in Section III-D, but breaks any other correlation structure inherent to the request process of the document.

Figure 5c shows that, although the resulting hit ratio is slightly below (resp. over) the original for small (resp. large)



(a) *YT Trace*



(b) *VoD Trace*

Fig. 6: *Comparison between the hit ratio of the original trace and the global randomization at different scales.*

cache sizes, the MARE is just 1.6% in the YT trace and 0.7% in the VoD trace. We thus conclude that the correlation among requests of a given document has little impact on the LRU performance and we can safely neglect it for modeling purpose.

Relation between Correlations and Timescales: We now determine at which timescale the correlation between requests has an impact in the LRU performance. With this in mind, we design a slightly different semi-experiment where we first extract sub-traces of different timescales, choosing high load periods. Then we apply the global randomization semi-experiment to each of these shorter traces. For each dataset, we distinguish three timescales and the results for each one are shown in Figure 6; other timescales lead to results that are just intermediate to the three presented here.

Near the first timescale (one week for YT and one month for VoD) and beyond, all timescales have a request correlation structure that approaches that observed in the full trace, and thus its hit ratio differs significantly from that of the global randomization. Indeed, already at this timescale, the MAREs are of 5.3% and 11.6% in the YT and VoD datasets, respectively; around the second timescale (four hours for YT and one day for VoD), we observe a decrease in the discrepancies as the MAREs are 5% and 2.3% in the YT and VoD case, respectively. Though we see that the correlation structure does not influence strongly the hit ratio, we remark again that the underestimation happens in the left side of the curves which corresponds to practical cache size. Finally for traces around the last timescale (half hour for YT and three hours for VoD), the MARE are 1.4% and 2.3% for YT and VoD, respectively, and we thus conclude that there are no significant structures between requests at this timescale.

Insights Gained: The latter results of the semi-experiments lead us to conclude that:

- I1: The correlation structure of the whole request process is not negligible, in terms of the hit ratio, at large timescales. We also infer that most of the correlation comes from the fact that all requests for the same document are grouped within its lifespan.
- I2: The document publications exhibit a correlation structure that does not have a significant impact on the hit ratio. In particular, we deduce that document arrivals to the catalog can be modeled by a Poisson process without losing accuracy on the estimation of the hit ratio curve.
- I3: For a given document, the request process within its lifespan exhibits some structure, but with little impact of the hit ratio. Thus, for a given document, we can approximate the requests sequence by a Poisson process defined on the lifespan of the document while still preserving the hit ratio.

V. Mathematical Analysis

In this section, we use the previous insights to build a mathematical model for the whole request process and detail the estimation of the corresponding hit rate in a LRU cache (throughout, the caching granularity is that corresponding to a document). The proofs of all propositions are in [20].

A. Catalog Arrival and Request Processes

We build our model for the document request process by following a top-down approach:

- on the top level, we consider the ground process Γ, hereafter called **catalog arrival process**; this point process dictates the consecutive arrivals of documents to the catalog. In our model, Γ is assumed to be a homogeneous Poisson process with constant intensity γ, according to Insight I2.

- let then d be the index of a document generated by process Γ, whose arrival time to the catalog is denoted by a_d. Document d then generates a **document request process** $\mathcal{R}_d$ determined by two random variables Λ_d and τ_d. Specifically, given Λ_d and τ_d, we assume the document request process to be Poisson with intensity function Λ_d on interval $[a_d, a_d + \tau_d]$ (cf. Insight I3); the duration τ_d is the lifespan of document d, intensity function Λ_d being zero outside interval $[a_d, a_d + \tau_d]$. In the following, we assume that

$$\overline{n}_d = \int_{a_d}^{+\infty} \Lambda_d(u)\mathrm{d}u \tag{3}$$

is almost surely finite;

- finally, the superposition of all processes $\mathcal{R}_d$ for all d generates the **total request process** $\mathcal{R} = \Sigma_d \mathcal{R}_d$ that contains the requests to all documents. In the following, we will also denote by $\mathcal{R}'_d = \mathcal{R} \setminus \mathcal{R}_d$ the request process resulting from the removal from total process $\mathcal{R}$ of points pertaining to request process $\mathcal{R}_d$ associated with given document d.

We can regard the point process $\mathcal{R}$ either as a doubly-stochastic Poisson process (also called *Cox* in the literature), that is a Poisson Process with random intensity (in our case the shot-noise process generated by the popularity functions Λ_d). Additionally, we can regard $\mathcal{R}$ as a *cluster* point process. Figure 7 gives a schematic view of our request model.

As detailed below (Section V-C), we will specialize to "box shaped" intensities, that is, $\Lambda_d(s) = \lambda_d \cdot \mathbb{1}_{\{a_d \le s \le a_d + \tau_d\}}$ where we independently choose the pair of (possibly dependent) random variables λ_d and τ_d. This setting will be referred to as the *Box Model*. Note that in this case, (3) reduces to $\overline{n}_d = \lambda_d \tau_d$.

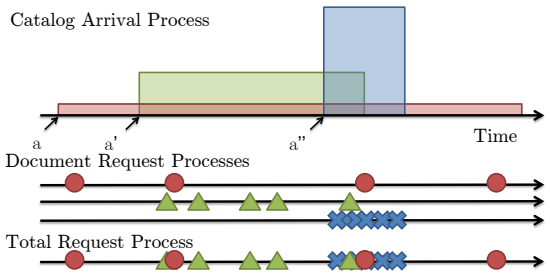


Fig. 7: *Sample of the document arrival and request processes.* ***Top:*** *Boxes represent the lifespan and popularity by their width and height, resp. (e.g., the document arriving at a is less popular than that arriving at a'' but it has a longer lifespan).* ***Bottom:*** *Sample of document request processes. Their superposition generates the total request process.*

B. General Hit Ratio Estimation

Given the dynamical request model presented in Section V-A, we now discuss the adaptation of the so-called "Che approximation" [6], [11] to calculate the hit ratio for requests addressed to a cache ruled by the LRU policy.

Assume that a given document d arrives to the catalog at time a_d. As the request process to document d is a Poisson process with intensity function Λ_d, the sequence $\Theta_1, \Theta_2, \ldots, \Theta_{n_d}$ of request times to d has n_d elements, where n_d follows a Poisson distribution with parameter $\overline{n}_d$ introduced in (3). The expected number of hits to the given document d then reads

$$\begin{aligned}\overline{H}_d &= \mathbb{E}_d\left[H_d\right] = \mathbb{E}_d\left[\sum_{r=2}^{n_d} \mathbb{1}_{\{\text{Request at } \Theta_r \text{ is a hit}\}}\right] \\ &= \sum_{k=2}^{+\infty}\left(\sum_{r=2}^{k} \mathbb{E}_d\left[\mathbb{1}_{\{\text{Request at } \Theta_r \text{ is a hit}\}} \,|\, n_d = k\right]\right)\frac{e^{-\overline{n}_d}\overline{n}_d^k}{k!}\end{aligned} \tag{4}$$

where $\mathbb{E}_d$ denotes the expectation, given Λ_d and τ_d. To proceed further with the calculation of $\overline{H}_d$, we need to incorporate the caching management policy. Specifically, we consider a cache of size C ruled under the LRU policy; the request at time Θ_r will then be a hit if and only if less than C different documents have been requested since the request arrival at time Θ_{r-1}. Formally, let X_t^s denote the number of *different* documents requested in time interval $[s,t]$, that is, $X_t^s = \#\{\text{Different documents requested in } [s,t]\}$ for $t > s$. From the stationarity of the ground process Γ, we first deduce the following.

Proposition 1: For any $s > 0$, processes $(X_t^s)_{t\geq s}$ and $(X_{t-s}^0)_{t\geq 0}$ have identical distributions. Furthermore, $\overline{X}^0$ is a Poisson process with associated mean function $\mathbb{E}\left[X_t^0\right] = \Xi(t)$ given by

$$\Xi(t) = \gamma \int_{-\infty}^{t} \mathbb{E}\left[1 - \exp\left\{-\int_0^t \Lambda_d(v)\,dv\right\}\right] \mathrm{d}a$$

for all $t \geq 0$ (in the latter integral, variable a stands for a_d).

The latter formula for $\mathbb{E}\left[X_t^0\right] = \Xi(t)$ can be easily interpreted by noting that the mean number of different documents arriving in interval $[a, a+\mathrm{d}a[$ being $\gamma \mathrm{d}a$ (for any $-\infty < a = a_d < t$), each of those documents is requested in interval $[0,t]$ with probability $1 - \exp(-\int_{[0,t]} \Lambda_d(v)\mathrm{d}v)$.

Now, an immediate consequence of Proposition 1 is that the first passage time $T_C^s = \inf\{t \geq s : X_t^s = C\}$ of process $(X_t^s)_{t\geq s}$ to level C has the same distribution than $T_C + s$, where $T_C = T_C^0$. We can now proceed with the calculation of $\overline{H}_d$ expressed in (4). From the previous discussion, a hit event at time Θ_r can be equivalently written as

$$\{\text{Request at } \Theta_r \text{ is a hit}\} = \left\{\Theta_r - \Theta_{r-1} < T_C^{\Theta_{r-1}} - \Theta_{r-1}\right\}$$

in terms of T_C. Recall that, given the arrival of document d at time a_d, the remaining process $\mathcal{R}'_d = \mathcal{R} \setminus \mathcal{R}_d$ has the same distribution than $\mathcal{R}$. It follows that the distribution of T_C for process $\mathcal{R}$ is identical to that associated with process $\mathcal{R}'_d$. As $T_C^{\Theta_{r-1}} = T_C + \Theta_{r-1}$ in distribution, we can write (4) as

$$\overline{H}_d = \sum_{k=2}^{+\infty}\left(\sum_{r=2}^{k} \mathbb{E}_d\left[\mathbb{1}_{\{\Theta_r - \Theta_{r-1} < T_C\}} \,|\, n_d = k\right]\right)\frac{e^{-\overline{n}_d}\overline{n}_d^k}{k!}. \tag{5}$$

The distribution of T_C intervening in (5) is usually unknown or hard to calculate. To overcome this difficulty, we now invoke the so-called **Che approximation**: we assume that the distribution of T_C is very concentrated so that it can be approximated by a constant t_C, hereafter called the *characteristic time*. The calculation of that characteristic time then proceeds as follows; using Proposition 1.B together with the approximation $T_C \approx t_C$, we can write $C = \mathbb{E}\left[X_{T_C}\right] \approx \mathbb{E}\left[X_{t_C}\right] = \Xi(t_C)$. We therefore define the characteristic time t_C by the inverse relation

$$t_C = \Xi^{-1}(C); \tag{6}$$

replacing T_C by t_C in (5), we then obtain the approximation

$$\overline{H}_d \approx \sum_{k=2}^{+\infty}\left(\sum_{r=2}^{k} \mathbb{E}_d\left[\mathbb{1}_{\{\Theta_r - \Theta_{r-1} < t_C\}} \,|\, n_d = k\right]\right)\frac{e^{-\overline{n}_d}\overline{n}_d^k}{k!} \tag{7}$$

for the expected number of hits.

C. Application to the Box Model

The general expressions for the average $\Xi(t) = \mathbb{E}\left[X_t^0\right]$ and the expected number of hits derived in Section V-B are now specified for the Box Model in order to obtain explicit formulas. In fact, the choice of a piecewise intensity function is consistent with Insight I3 gained from Section IV.

Proposition 2: For the Box Model, the mean of the process X^0 is given by

$$\begin{aligned}\Xi(t) = \gamma\,\mathbb{E}\left[2t + \left(1 - e^{-\lambda t}\right)\left(\tau - t - \frac{2}{\lambda}\right)\mathbb{1}_{\{\tau \geq t\}}\right] \\ + \gamma\,\mathbb{E}\left[2\tau + \left(1 - e^{-\lambda \tau}\right)\left(t - \tau - \frac{2}{\lambda}\right)\mathbb{1}_{\{\tau < t\}}\right]\end{aligned}$$

for all $t \geq 0$, where the pair of positive variables (λ, τ) is distributed as any pair (λ_d, τ_d).

To interpret the latter expression of $\Xi(t)$, assume $\tau_d = \tau_0$ and $\Lambda_d = \lambda_0$ are constants; then $\Xi(t)$ grows non-linearly in t if $t < \tau_0$ and linearly otherwise; in the latter case, we can write the mean function as $\Xi(t) = \Xi(\tau_0) + \gamma(t - \tau_0)(1 - e^{-\lambda_0 \tau_0})$ which is just the mean number of new objects up to time τ_0, plus the mean number of arrivals to the catalog in interval $[\tau_0, t]$ penalized by the probability that the document has at least one request.

To finally specify the Che approximation for the Box Model, we now state the following.

Proposition 3: Under the Che approximation and for the Box Model, the conditional expectation of the expected number of hits to document d is given by

$$\overline{H}_d = \begin{cases} \lambda_d \tau_d - 1 + e^{-\lambda_d \tau_d} & \text{if } \tau_d < t_C, \\ (\lambda_d \tau_d - 1)(1 - e^{-\lambda_d t_C}) + \lambda_d t_C e^{-\lambda_d t_C} & \text{if } \tau_d \geq t_C. \end{cases}$$

We then conclude from Proposition 3 that the expected number of hits to all documents is given by

$$\begin{aligned}\mathbb{E}\left[H\right] = \int\int_{\lambda>0, \tau<t_c} \left[\lambda\tau - 1 + e^{-\lambda\tau}\right] f(\lambda,\tau)\mathrm{d}\lambda\mathrm{d}\tau + \\ \int\int_{\lambda>0, \tau\geq t_c} \left[(\lambda\tau - 1)(1 - e^{-\lambda t_C}) + \lambda t_C e^{-\lambda t_C}\right] f(\lambda,\tau)\mathrm{d}\lambda\mathrm{d}\tau\end{aligned} \tag{8}$$

where f denotes the joint probability density of the pair (λ, τ), and with t_C derived by (6) via the expression of $\Xi(t)$ obtained in Proposition 2.

VI. Model Validation

The aim of this final section is to assess the validity of our Box model for the calculation of the hit ratio (as derived in Proposition 3), when compared to the values obtained by a direct simulation. To this end, we first detail the computation of the necessary statistics to use our model, namely ***(i)*** the catalog arrival intensity γ, ***(ii)*** the mean $\Xi(t)$ for all $t \geq 0$, and ***(iii)*** the hit ratio $\mathrm{HR} = (\sum_d H_d)/(\sum_d n_d) = \mathbb{E}[H_d]/\mathbb{E}[n_d]$.

(i) Let first N be the number of documents in our sample and denote by W the size of the observation window; we can then estimate the catalog arrival rate γ by $\widehat{\gamma} = N/W$;

(ii) We have noted in Section III-D that estimators $\widehat{\tau}$ and $\widehat{\lambda}$, respectively expressed in (1) and (2), are not available for documents with only one request. This sub-sample has, however, a considerable size (58% of all documents) and cannot be neglected in a direct application of Proposition 2.

To incorporate this data, we use the approximation discussed in [12] where the set of documents requested only once is represented by a "noise" process. Let Ξ_1 (resp. Ξ_2) denote the mean function of that noise process (resp. the mean function associated with the "non-noise" part of the process), with $\Xi = \Xi_1 + \Xi_2$. We can separate the noise process from the rest of the request process and, using a procedure similar to that of Proposition 2, we easily obtain an explicit formula for $\Xi_1(t)$ (omitted here for brevity); the latter together with the formula for $\Xi(t)$ in Proposition 2 then gives

$$\Xi_2(t) = \Xi(t) - \Xi_1(t) = \gamma\,\mathbb{E}[L(\lambda, \tau, t)]$$

where

$$L(\lambda,\tau,t) = \left[2t(1-e^{\lambda t}) + \left(1 - e^{\lambda t} - \lambda t e^{\lambda t}\right)\left(\tau - t - \frac{4}{\lambda}\right)\right]\mathbb{1}_{\{\tau \geq t\}} + \left[2\tau(1-e^{\lambda \tau}) + \left(1 - e^{\lambda \tau} - \lambda \tau e^{\lambda \tau}\right)\left(t - \tau - \frac{4}{\lambda}\right)\right]\mathbb{1}_{\{\tau < t\}}.$$

Recall that the documents for which we have an estimate of the pair (λ_d, τ_d) are precisely those accounted into Ξ_2. We thus estimate

$$\widehat{\Xi}_2(t) = \widehat{\gamma} \times \mathbb{E}[L(\lambda, \tau, t)]$$

with the expectation taken w.r.t. the empirical distribution of (λ, τ) in the trace.

Finally, let N_1 and N_2 be the number of documents with one, and more than one requests, respectively. We then estimate $\Xi_1(t)$ by the mean function of a homogeneous Poisson process, that is,

$$\widehat{\Xi}_1(t) = N_1 \times t/W.$$

The estimator of the characteristic time associated with the Che approximation is therefore given by $\widehat{t}_C = \widehat{\Xi}^{-1}(C)$, with $\widehat{\Xi}(t) = \widehat{\Xi}_1(t) + \widehat{\Xi}_2(t)$.

(iii) Concerning the hit ratio HR, we must similarly take the documents with just one request into account. Note that since the documents pertaining to the noise process do not produce hits, we can write

$$\mathrm{HR} = \frac{\mathbb{E}[H_d]}{\mathbb{E}[n_d]} = \frac{\mathbb{E}\left[H_d \mathbb{1}_{\{n_d \geq 2\}}\right]}{\mathbb{E}\left[n_d \mathbb{1}_{\{n_d \geq 1\}}\right]} = \frac{\mathbb{E}\left[\mathbb{E}_d\left[H_d \mathbb{1}_{\{n_d \geq 2\}}\right] \mid n_d \geq 2\right]}{\mathbb{E}[n_d \mid n_d \geq 2] + \dfrac{\mathbb{P}(n_d = 1)}{\mathbb{P}(n_d \geq 2)}}.$$

Let $\overline{H}_d = M(\tau_d, \lambda_d, t_C)$ be the conditional expectation of the number of hits given by Proposition 3; by similar arguments to those used in *(ii)*, the numerator of the hit ratio is thus estimated by

$$\mathbb{E}\left[\overline{H}_d \mid n_d \geq 2\right] \approx \mathbb{E}\left[M(\tau, \lambda, \widehat{t}_C)\right]$$

with the expectation taken w.r.t. the empirical distribution of (λ, τ) in the trace. As to the term $\mathbb{E}[n_d \mid n_d \geq 2]$, it can be computed as the average number of requests in the corresponding sub-sample. Finally, the ratio $\mathbb{P}(n_d = 1)/\mathbb{P}(n_d \geq 2)$ in the denominator is estimated by

$$\mathbb{P}(n_d = 1)/\mathbb{P}(n_d \geq 2) \approx N_1/N_2.$$

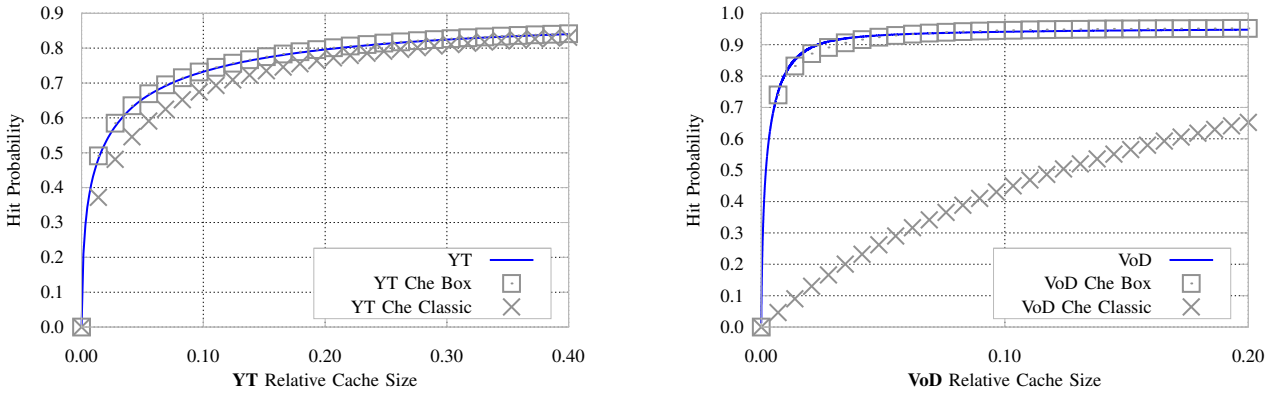
Using the above estimators, we can eventually compare the hit ratio derived from the Box Model to that obtained by simulation for each trace, as depicted in Figure 8. For comparison purpose, we provide also the estimation of the hit ratio obtained by the Che approximation when the request process is assumed to be IRM. For the YouTube traffic, the Box Model improves the accuracy by one order of magnitude compared to the estimation with an IRM process, with respective MARE of 0.5% and 4.1%. For the VoD traffic, the improvement is even more spectacular, due to the large duration of the trace. The IRM is far from estimating properly the hit ratio with a MARE of 17.2% (this value is significantly decreased by including the tail of the curve, not plotted here, and where the IRM converges towards the correct value). On the other hand, the Box Model estimates accurately the hit ratio, with a MARE of 0.6%. This validates our model.

VII. Conclusion

The current literature on the performance of caching systems ignores the fact that content is produced and becomes outdated everyday. The consequence for caches is non-negligible, as requests for a given document are concentrated within its lifetime and the request process is thus non stationary. This paper addresses the issue of catalog dynamics. Based on two traffic traces, we provide evidence for the impact of the catalog dynamics and identify the core structures of the request process. We then propose a general model for the aggregated request process and provide an estimate of the hit ratio of a LRU cache fed by such a request process.

Our results show that the document request process can be easily described, as far as caching is concerned, in terms of basic document statistics: the document lifespan and its average request intensity (within its lifespan). As expected, the hit ratio is mainly driven by the distribution of the number of requests for each document. The distribution of request intensities, however, has a secondary impact on the hit ratio: higher intensities (and thus shorter lifespan) lead to higher performance, which confirms basic intuition.

Our proposed model currently uses documents as a basic unit. In practice, however, bandwidth and cache size are



(a) *YT trace.*

(b) *VoD trace.*

Fig. 8: *Fittings for the Che estimation*

counted in bytes. Additionally, in the case of video streaming, the downloads of videos are frequently interrupted because users switch to another video. As a further study, we intend to account for both the size distribution of videos and the impact of these interruptions on caching. Our model for the catalog dynamics can also be improved as follows: (i) the catalog size in our model is stationary, while it actually increases with time; (ii) the Box Model can be generalized to a broader family of intensity functions. On the basis of these potential generalizations, we believe that the approach of the present paper can be easily extended to other traffic types.

REFERENCES

[1] M. Ahmed, S. Traverso, P. Giaccone, E. Leonardi, and S. Niccolini. Analyzing the performance of LRU caches under non-stationary traffic patterns. *CoRR*, 2013.

[2] M. Ahmed, S. Traverso, P. Giaccone, E. Leonardi, and S. Niccolini. Why temporal locality matters: Evaluating cache performance for content-on-demand distribution. *CoRR*, 2013.

[3] B. Azimdoost, C. Westphal, and H. R. Sadjadpour. On the throughput capacity of information-centric networks. In *Teletraffic Congress (ITC), 2013 25th International*, 2013.

[4] Y. Carlinet, T. D. Huynh, B. Kauffmann, F. Mathieu, L. Noirie, and S. Tixeuil. Four months in dailymotion: Dissecting user video requests. *International Workshop on TRaffic Analysis and Classification (TRAC)*, Aug. 2012.

[5] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.*, 17(5), Oct. 2009.

[6] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *Selected Areas in Communications, IEEE Journal on*, 20(7), 2002.

[7] X. Cheng, C. Dale, and J. Liu. Understanding the characteristics of internet short video sharing: YouTube as a case study. *CoRR*, 2007.

[8] L. Cherkasova and M. Gupta. Analysis of enterprise media server workloads: access patterns, locality, content evolution, and rates of change. *IEEE/ACM Transactions on Networking*, 12(5), oct. 2004.

[9] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: Incrementally deployable ICN. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 2013.

[10] N. Fofack, P. Nain, G. Neglia, and D. Towsley. Analysis of TTL-based cache networks. In *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on*, 2012.

[11] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for cache performance. In *24th International Teletraffic Congress*. IEEE Communications Society, 2012.

[12] F. Guillemin, B. Kauffmann, S. Moteau, and A. Simonian. Experimental analysis of caching efficiency for YouTube traffic in an isp network. In *Teletraffic Congress (ITC), 2013 25th International*, 2013.

[13] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang. The stretched exponential distribution of internet media access patterns. In *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*, 2008.

[14] N. Hohn, D. Veitch, and P. Abry. Cluster processes, a natural language for network traffic. *IEEE Transactions on Signal Processing, special issue "Signal Processing in Networking"*, 51(8), Aug. 2003.

[15] P. R. Jelenković. Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities. *The Annals of Applied Probability*, 9(2), 1999.

[16] P. R. Jelenković and X. Kang. Characterizing the miss sequence of the LRU cache. *ACM SIGMETRICS Performance Evaluation Review*, 36, August 2008.

[17] P. R. Jelenković and A. Radovanović. Least-recently-used caching with dependent requests. *Theoretical Computer Science*, 326(1–3), 2004.

[18] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. *arXiv preprint arXiv:1307.6702*, 2013.

[19] S. Mitra, M. Agrawal, A. Yadav, N. Carlsson, D. Eager, and A. Mahanti. Characterizing web-based video sharing workloads. *ACM Transactions on the Web*, 5, May 2011.

[20] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet. Catalog dynamics: Impact of content publishing and perishing on the performance of a LRU cache. http://perso.rd.francetelecom.fr/kauffmann/publi/dynamicityArXiv.pdf, 2014.

[21] A. Panagakis, A. Vaios, and I. Stavrakakis. Approximate analysis of lru in the case of short term correlations. *Computer Networks*, 52(6), 2008.

[22] K. Psounis, A. Zhu, B. Prabhakar, and R. Motwani. Modeling correlations in web traces and implications for designing replacement policies. *Computer Networks*, 45(4), 2004.

[23] L. Rizzo and L. Vicisano. Replacement policies for a proxy cache. *IEEE/ACM Trans. Netw.*, 8(2), Apr. 2000.

[24] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of YouTube network traffic at a campus network — measurements, models, and implications. *Computer Networks*, 53(4), 2009.