A General, Tractable and Accurate Model for a Cascade of LRU Caches

N. Blefari Melazzi, G. Bianchi, A. Caponi, A. Detti
Università degli Studi di Roma - Tor Vergata / CNIT, Italy
name.surname@uniroma2.it

***Abstract*—The recent evolution of the Internet towards "Information-centric" transfer modes has renewed the interest in characterizing multi-cache systems, in which requests not satisfied by a cache are forwarded to other caches. In this work, we characterize the traffic statistics of the output (miss) stream, via a simple but accurate approximate analysis for LRU caches feeded by general "renewal" traffic patterns. In turn, we exploit such output stream traffic pattern to analyze the performance of the subsequent cache stage, and so on. The computational efficiency of our model, joint with its ability to handle traffic patterns beyond the traditional independent reference model, permits simple and tractable assessment of cache hierarchies.**

***Index Terms*—information centric networking; in-network caching; analytical model; performance evaluation;**

I. Introduction

Caching is a technique used to temporarily store data, usually coming from an origin source, within a memory quickly accessible from the intended user of that data. Caches decrease access time and/or system load, as repeated requests for the same data are served by a fast/local memory rather than by a slower/remote source. Requests of a data item to a single cache system result in a cache hit when that item is found in the cache. In case of a cache miss, the request is forwarded to the origin location of the item, or to a subsequent cache along the path, when networks of caches are employed. Multi-cache systems can have different topologies, including cascade or hierarchical configurations. When caches are full, a replacement policy chooses which item is to be cancelled to make room for a new item, following a cache miss. A popular replacement policy is the Least Recently Used (LRU) algorithm, which discards the least recently used data item.

A thorough understanding of the performance and phenomena involved in multi-cache systems is beneficial not only in traditional web caching and content distribution network scenarios, but also because of the proposed evolution of the Internet towards a so called Information Centric Network (ICN) [1]–[5]. An ICN provides users with contents exposed as names, instead of providing communication channels between hosts; the network transfers individual, identifiable content chunks, instead of unidentifiable data containers (i.e., IP packets). Content chunks are explicitly addressed by-name allowing to perform content caching systematically and on-the-fly, potentially in every network node, and without the need to deploy cumbersome tasks such as HTTP header parsing [6].

This work is supported in part by the European Commission in the context of the FP7/NICT EU-JAPAN GreenICN project

The main contribution of this letter consists in characterizing the miss stream of an LRU cache loaded by a requests stream that follows a general renewal model, and exploiting such miss stream model to analyze subsequent caching stages. As shown by comparison with simulation results, our approach is extremely accurate and computationally convenient. In essence, our work provides a first step towards a *tractable* analysis of multi-cache systems using a *practical* replacement policy (LRU), and with traffic exhibiting *temporal locality*.

Indeed, most of the previous analytic work dealing with caching [7]–[13] either assume exponentially distributed inter-arrival times between requests, or employ the conceptually analogous discrete model called Independent Reference Model (IRM), which states that "requests for items occur in an infinite sequence where the item indexes required on the *i*-th request, for $i > 0$, are independent random variables on $\{1, 2, \cdots, N\}$ with a common probability distribution" [14]. As shown in [15], the IRM model fails to be realistic even when considering the traffic natively offered by clients to an edge cache (i.e. with no intermediary cache): real world traffic exhibits temporal correlation properties that cannot be captured by IRM or Poisson models. And the cache hit probability in presence of temporal locality can largely differ from the one computed with the IRM or Poisson assumption [16].

Moreover, even if we assume IRM traffic arriving at a first edge cache, it was proven in [10] that the resulting miss stream, which would be offered as input to a second stage cache in a multi-cache system, is no more IRM, thus preventing an accurate modeling of multi-cache systems. As a matter of fact, previous models for network of caches rely on IRM, but recognize it to be an approximation [8]. To the best of our knowledge, the only model for multi-cache systems (cascade and trees) that does *not* employ the IRM assumption is [17], which instead uses a renewal traffic assumption similar to ours, and to [16] for the single cache case. However, [17] resorts to an idealized cache operation based on a time-to-live (TTL) eviction policy, rather than the practical LRU policy modeled in this paper.

II. Model

In this paper we assume, for modeling convenience, that the storage capacity C is expressed as number of items that may be stored therein. This assumption implies items of same

size; extensions to uneven sizes can be addressed, e.g. as discussed in [12], [14]. We assume that items are drawn from an universe size of cardinality N. Items are conveniently named using the index x, with $x \in \{1, 2, ..., N\}$. Unlike most past works (e.g. [7], [12], [14]), we characterize the traffic arrival process *without* relying on the so-called *Independent Reference Model*. Rather, we model the system under more general conditions, by assuming: i) a continuous time scale; ii) inter-arrival times between two consecutive requests for a same item being independent and identically distributed random variables T_x, with general cumulative probability distribution function $F_x(t)$ and probability density function $f_x(t)$. When needed, we denote the expected inter-arrival time with $E[T_x] = 1/\lambda_x = \int_0^\infty (1 - F_x(t))\,\mathrm{d}t$, being thus λ_x the *average arrival rate* associated to item x. We assume stationary arrivals, and consequent long-term item popularity distribution $q_x = \lambda_x / \sum_{i=1}^N \lambda_i$ which, unless otherwise specified, we quantify with a Zipf (non restrictive, as our model does not require to specify any popularity distribution).

We remark that the renewal i.i.d arrival process considered in this paper appears sufficiently descriptive to capture a wide range of *temporal locality* conditions and practical bursty-like traffic patterns, for instance by choosing a random variable T_x with relatively large coefficient of variation.

A. First-level cache with renewal input

The single-cache model presented in what follows extends, to the renewal input traffic assumption, a clever approximation originally introduced in [12] by Che et al. for IRM. Let us focus on an item x. In most generality, its cache *eviction* time, namely the time elapsing between the instant of time the item is inserted (refreshed) in the cache, and the time in which the item is evicted from the cache because other C distinct items have been therein accommodated, is a random variable with non unknown distribution. [12] suggests that, for practical (reasonably large) cache sizes and population of items (request rate for each given item being small with respect to the overall traffic), this random variable can be *approximated with a constant*, further *independent* of the specific item x considered. Despite its simplicity, such an approximation is shown to be extremely accurate, for IRM traffic, as indeed confirmed by the further analysis and discussion provided in [14].

Under Che's assumption of constant (but unknown) cache eviction time t_c, a very simple model can be devised as follows. Indeed, if t_c were known, the probability H_x that a cache hit occurs for item $x \in (1, N)$ would be trivially given by the probability that the inter-arrival time is lower than t_c,

$$H_x = P\{T_x \leq t_c\} = F_x(t_c), \tag{1}$$

resulting in a (weighted) average hit ratio for the whole cache

$$H = \frac{\sum_{x=1}^N \lambda_x H_x}{\sum_{i=1}^N \lambda_i} = \sum_{x=1}^N q_x H_x. \tag{2}$$

In order to find t_c, we remark that the arrival of a request for an item $x \in (1, N)$ is a *renewal* instant. Indeed, irrespective on whether the item was earlier evicted by the cache (and thus the new arrival is a *MISS*, and the item is reinserted) or the item was still in the cache (and thus the new arrival is a *HIT*), at the instant of arrival, under the LRU policy, the item is (logically) placed at the *top* of the cache, and its future eviction time does not depend on past events, but only on future arrivals. On top of this renewal process, we thus conveniently define a continuous-time *Indicator* process $I_x(t)$, which is equal to 1 when the item x is stored in the cache, and 0 otherwise. From the elementary renewal theorem,

$$E[I_x(t)] = \frac{E[\textit{cache time per cycle}]}{E[\textit{cycle duration}]} = \frac{\int_0^{t_c}(1 - F_x(t))\mathrm{d}t}{1/\lambda_x} \tag{3}$$

where the numerator is the expected value of the random variable defined by $\min(T_x, t_c)$. Indeed, the time spent in the cache in a considered cycle is either the inter-arrival time of the next request for x, if this comes before the eviction time t_c, or it is bounded by t_c. The unknown constant t_c can now be computed by imposing the condition that, at each time instant, the cache must contain exactly C distinct items, i.e.,

$$\sum_{x=1}^N I_x(t) = C \quad \rightarrow \quad \sum_{x=1}^N E[I_x(t)] = C \tag{4}$$

B. Characterizing the cache output stream

An interesting remark in [12] is that a cache can be viewed as a low-pass filter with a cutoff frequency equal to the inverse of the eviction time of the cache, t_c. Here, filtering must be understood in the sense that requests of an item occurring with a frequency lower than $1/t_c$ will result in a cache miss and thus contribute to the *miss stream*. Higher frequency requests will find the item in the cache and will not be forwarded to the next cache. Since our model (1) is fully described by the inter-arrival distribution of requests for each item, we can push further such a filtering analogy. Indeed, we can look at this filtering process on the time axis: if an arrival of a request occurs later than t_c from the previous one, it will "pass through" the cache and contribute to the miss stream; otherwise it will be filtered out.

More specifically, let $\bar{T}_x$ be the r.v. describing the inter-arrival time between two consecutive *cache misses* for a same item x, and recall that $F_x(t)$ and $f_x(t)$ are the CDF and PDF of the original inter-arrival process T_x offered to the first level cache. By construction, a cache miss is caused by a (last) inter-arrival $T_x > t_c$, possibly preceeded by 0 or more inter-arrival times shorter than t_c (hence filtered out as first level cache hits). Hence, the PDF $f_{\bar{x}}(t)$ of the r.v. $\bar{T}_x$ can be expressed as (for $t > t_c$, otherwise zero):

$$f_{\bar{x}}(t) = u_1(t - t_c) f_x(t) * \sum_{k=0}^{\infty} \{(1 - u_1(t - t_c))\, f_x(t)\}^{*k} \tag{5}$$

where $u_1(t)$ is the unit-step function, the operator * denotes convolution, $\{g(t)\}^{*k}$ is the n-fold convolution of the (generic) function $g(t)$ with itself, with the usual convention that $\{g(t)\}^{*0}$ yields the Dirac $\delta(t)$.

It is also useful to derive compact expressions for mean and variance of $\bar{T}_x$. The mean value is trivially given by the inverse of the miss stream frequency, i.e.,

$$E[\bar{T}_x] = \frac{1}{\lambda_x(1 - H_x)} = \frac{E[T_x]}{1 - H_x}. \tag{6}$$

The variance instead requires some more algebra and is expressed in terms of the statistics of the original inter-arrival time T_x by

$$Var[\bar{T}_x] = \frac{Var[T_x]}{1-H_x} - \frac{H_x(E[T_x]^2 - 2E[T_x]E[T_x|T_x \leq t_c])}{(1-H_x)^2} \quad (7)$$

Finally, dividing (7) by the the square of (6) yields the square of the Coefficient of Variation

$$C_{\bar{x}}^2 = \frac{Var[\bar{T}_x]}{E[\bar{T}_x]^2} = C_x^2(1-H_x) + H_x\left(\frac{2E[T_x|T_x \leq t_c]}{E[T_x]} - 1\right) \quad (8)$$

Note that this value depends on the cache filtering effect; in other words, caching not only affects popularity, but also individual flow statistics (e.g. the coefficient of variation). For instance, if we assume exponentially distributed inter-arrival of requests as input, (8) simplifies to $1 - 2\lambda_x t_c e^{-\lambda_x t_c}$, showing that the cache has a varying smoothing effect on the CV, depending on the product $\lambda_x t_c$, with smoothing maximum at $\lambda_x t_c = 1$, with a resulting $CV = \sqrt{1-2/e}$.

C. Cache cascade

At this point it is easy to evaluate the performance of a series of caches, where each cache is loaded with the output stream the previous one. Since the arrival process at a cache is the miss stream of the previous cache, to derive the probability that a cache hit occurs for item $x \in (1, N)$ it suffices to apply (1) using (5) as probability distribution function. In the case of more general cache networks, the offered traffic may lose the renewal property, as it is the superposition of exogenous inputs (e.g. miss streams of several neighboring caches). As shown in [17] for TTL caches, the renewal assumption however appears to be a reasonable approximation.

III. Numerical Results

In order to evaluate the accuracy our model we devised a trivial MATLAB simulator of an LRU cache [18] that takes as input the requests vector and outputs the per-item cache hit probability and the missed stream. This stream is used to estimate the miss sequence pdf and to feed the next cache. Results are obtained using a cache size $C = 1000$ and a total population of 10^6 content items. Although request inter-arrivals per different items may follow different probability distributions, for convenience we report results only for homogeneous distributions with frequency of requests proportional to the popularity q_x drawn from a Zipf distribution with slope coefficient $\alpha = 0.8$. We consider Exponential (Poisson) inter-arrivals, to model a request stream that follows the independent reference model (IRM) [14], and Lognormal or Hiperexponential distributions reproducing a request stream with temporal locality. In these latter cases, following [15], we change the *coefficient of variation* (CV, defined as the ratio between the standard deviation and the mean value) ranging from 1 to 8 to affect the temporal locality.

We start with the analysis of a first-level cache. Fig. 1 shows the total cache hit probability for three distributions of the request inter-arrival time versus CV ($CV = 1$ for the Exponential case). Model (2) and simulations are compared,

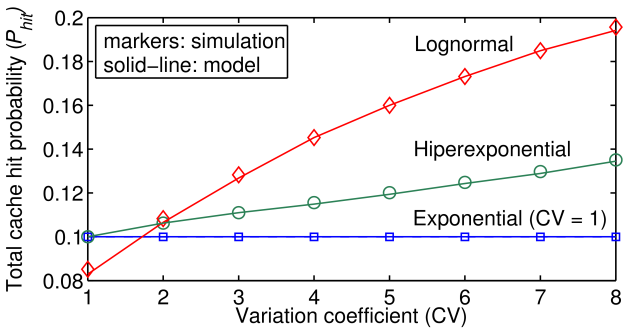


Fig. 1. Miss stream popularity distribution

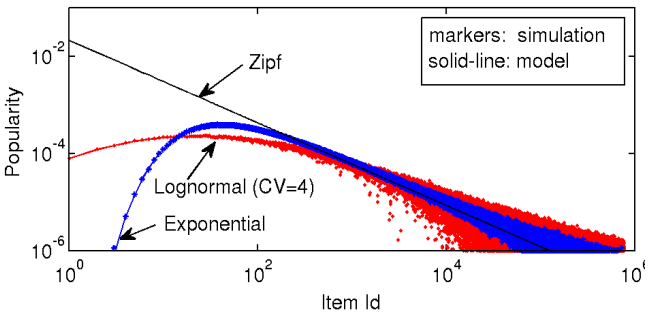


Fig. 2. Miss stream popularity distribution

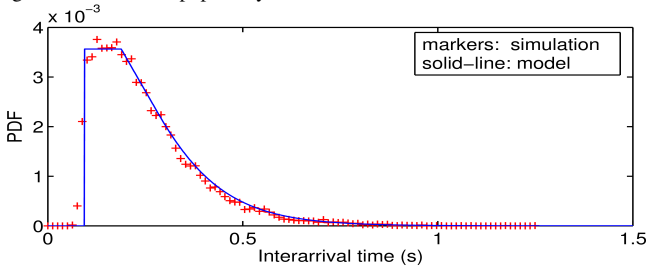
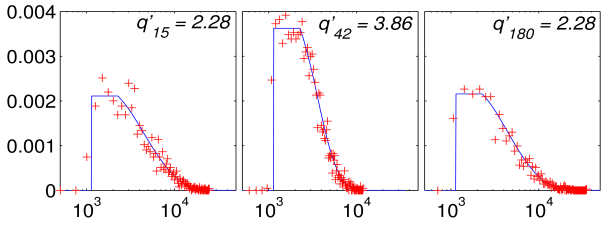


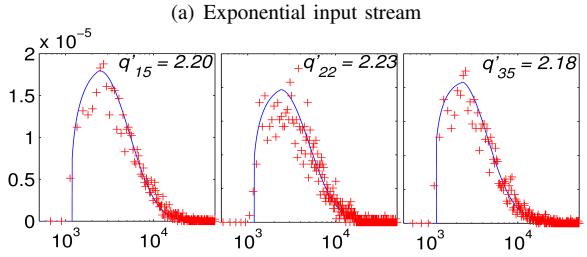
Fig. 3. Probability density function (PDF) of an item on the miss stream

showing a quite perfect fit with a mean squared error between the simulation points and those of the model in the order of 10^{-7} in the LogNormal case and 10^{-8} in the Hiperexponential one. As expected, performance significantly depends on the chosen inter-request distribution and improve for a greater CV, i.e. greater temporal locality. This is yet another confirmation of the finding in [10], that an error can be done by assuming Poisson input streams to assess the performance of cache networks fed by traffic with temporal locality [19]. Turning now to second-level cache issues, Fig. 2 shows how the Zipf popularity law of requests arriving to the first cache q_x is modified by the first-level cache: the popularity distribution of the miss stream $q'_x = q_x(1-H_x)$, shown for both exponential and lognormal distribution of the inter-arrival process of requests at the first cache, is a filtered replica of the ingress one; the miss stream popularity q'_x computed using the hit probability H_x with our model (1) closely follows simulations, with a mean squared error lower than 10^{-10}.

Fig. 3 shows the probability density function of the inter-arrival time of the most popular item (i.e., the first item of the input stream, with a first-level cache of size 100) in the miss stream, comparing simulations and model (5); the inter-arrival process of item requests to the first-level cache is exponential. The filtering effect is evident: the pdf is zero for inter-arrival



(a) Exponential input stream



(b) Lognormal input stream

Fig. 4. PDF of the output (miss) stream for different items; $q'_x * 10^{-4}$ is the (normalized) popularity at the output of the cache

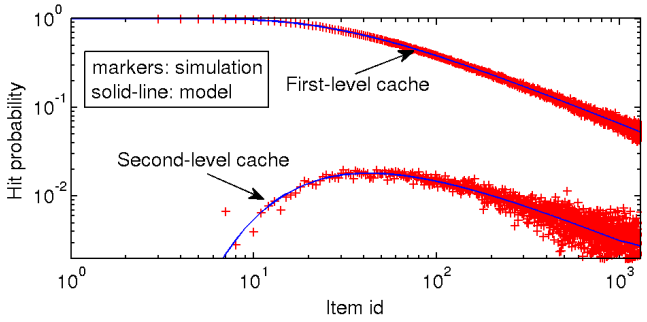


Fig. 5. Per item hit-rate on 2nd cache for exponential distrib. on 1st cache

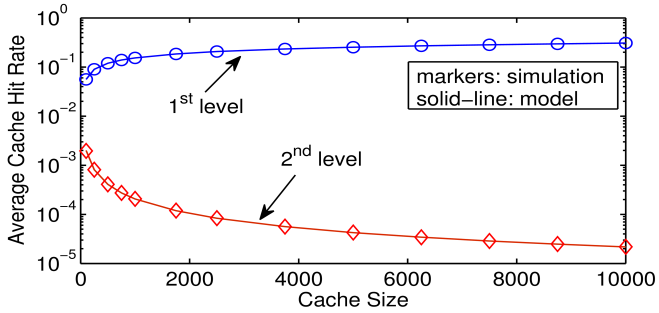


Fig. 6. Average cache hit rate on 1st and 2nd level caches vs. the cache size

times less than t_c ($93.2ms$). Once again, fitting is remarkable, with a mean squared error of $11.7 * 10^{-7}$. Fig. 4 shows the same performance measure for other items and for both exponential and lognormal inter-arrivals at the first cache. Fig. 5 shows the items hit-rate on the first and second cache, with an exponential distribution of the inter-arrival process at the first cache. The difference between the two is noticeable and confirms the need of suitable models for multi-cache systems. Finally, Fig. 6 shows how our model perfectly succeeds in evaluating average cache hit rate for different cache sizes.

IV. Conclusion and Future Work

We conclude with three remarks: i) as noted in [14], the approximation [12] works very well even beyond the applicability scenarios stated by its own authors; in addition to the results presented above, we followed a suggestion of [14] and used the approximation [12] instead of the model [7] for the single cache approximation used in [8] to evaluate cache networks; the result was a 30th fold decrease of the computing time in some exemplary cases, with a remarkable accuracy; ii) the summation (5) converges very rapidly: few iterations are enough to reach accuracy in the order of 10^{-6} in some exemplary cases; iii) our model is general, as it allows using any renewal distribution, tractable as it requires simple algebra with low computing time, and accurate, as our results make evident. A more thorough analysis of results, ensuing design principles and applications, and explicit extension of the model to cache networks, possibly considering the approximation approach proposed in [17], is left for further work.

References

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *5th ACM CoNext*, 2009.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of Information-Centric Networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, July 2012.

[3] Xylomenos, G. et al., "A survey of information-centric networking research," iEEE Communications Surveys and Tutorials, 2013, Early Access Article, DOI: 10.1109/SURV.2013.070813.00063.

[4] A. Detti, M. Pomposini, N. Blefari-Melazzi, and S. Salsano, "Supporting the web with an information centric network that routes by name," *Computer Networks*, vol. 56, no. 17, pp. 3705 – 3722, 2012.

[5] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *2nd SIGCOMM ICN workshop*, 2012, pp. 55–60.

[6] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-Centric Networking: seeing the forest for the trees," in *10th ACM SIGCOMM HotNets*, 2011.

[7] A. Dan and D. F. Towsley, "An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes," in *ACM SIGMETRICS*, 1990, pp. 143–152.

[8] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *IEEE INFOCOM*, 2010, pp. 1–9.

[9] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *10th IFIP NETWORKING'11*. Springer, 2011, pp. 78–91.

[10] P. R. Jelenković and X. Kang, "Characterizing the miss sequence of the lru cache," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 2, pp. 119–121, 2008.

[11] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 758 – 770, 2013.

[12] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: modeling, design and experimental results," *IEEE J. on Sel. Areas in Commun.*, vol. 20, no. 7, pp. 1305–1314, 2002.

[13] J. Ardelius, B. Grönvall, L. Westberg, and Å. Arvidsson, "On the effects of caching in access aggregation networks," in *2nd ACM SIGCOMM ICN workshop*. ACM, 2012, pp. 67–72.

[14] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *24th international teletraffic conference, ITC24*, 2012, pp. 1–8.

[15] R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao, "On the intrinsic locality properties of web reference streams," in *IEEE INFOCOM*, 2003.

[16] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi, "Check before storing: what is the performance price of content integrity verification in lru caching?" *ACM SIGCOMM Comp. Commun. Rev.*, vol. 43(3), pp. 59–67, 2013.

[17] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley, "Analysis of ttl-based cache networks," in *IEEE Perf. Eval. Methodologies and Tools (VALUETOOLS)*, 2012, pp. 1–10.

[18] http://netgroup.uniroma2.it/research/icn/code/.

[19] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, 1999, pp. 126–134.