Exact Analysis of TTL Cache Networks –

The Case of Caching Policies Driven by Stopping Times

Daniel S. Berger
Distributed Computer Systems (DISCO) Lab
University of Kaiserslautern
berger@cs.uni-kl.de

Philipp Gland
Institute for Mathematics
Technical University Berlin
gland@mailbox.tu-berlin.de

Sahil Singla
School of Computer Science
Carnegie Mellon University
ssingla@cmu.edu

Florin Ciucu
Computer Science Department
University of Warwick
f.ciucu@warwick.ac.uk

ABSTRACT

TTL caching models have recently regained significant research interest, largely due to their ability to fit popular caching policies such as LRU. In this extended abstract we briefly describe our recent work on two *exact* methods to analyze TTL cache networks. The first method generalizes existing results for line networks under renewal requests to the broad class of caching policies whereby evictions are driven by stopping times. The obtained results are further generalized, using the second method, to feedforward networks with Markov arrival processes (MAP) requests. MAPs are particularly suitable for non-line networks because they are closed not only under superposition and splitting, as known, but also under input-output caching operations as proven herein for phase-type TTL distributions. The crucial benefit of the two closure properties is that they jointly enable the first exact analysis of feedforward networks of TTL caches in great generality.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques

Keywords

Cache networks, Analytical model, cache performance

1. INTRODUCTION

Time-to-Live (TTL) caches decouple the eviction mechanisms amongst objects by associating each object with a timer. When a timer expires, the corresponding object is evicted from the cache. This seemingly simple scheme has been widely deployed in DNS, web caching, and flow tables of software-defined networking switches. What has recently additionally popularized TTL analytical models is a subtle mapping between *capacity-driven* (e.g., Least-Recently-Used (LRU)) and *TTL-based* caching policies. This mapping was firstly established through a remarkably accurate approximation by Che *et al.* [3] for the popular LRU policy, and has received theoretical justifications and extensions to a variety of cache policies [4, 6]. While the analysis of TTL caches is arguably simpler and more general than the analysis of capacity-driven policies, the exact analysis of TTL networks in general has remained an open problem.

When considering a network of caches, the analysis has to address two inherent network operations. Given a caching node and some object, the first operation relates the request point process (the *input*) to the corresponding miss process (the *output*), which is a sample of the request process at those points when the object is absent from the cache. The *superposition* operation occurs when merging miss processes from upstream caches into a new input process. The exact characterization of these processes is challenging; for instance, the input-output relation does not retain the memorylessness property of request processes and renewals are not closed under the superposition operation.

In this extended abstract we summarize our recent results [2] on the first exact analysis of caching networks. We address lines of caches under renewal request processes and a general TTL distribution, and feedforward caching networks under Markov arrival processes (MAPs) and a TTL phase-type (PH) renewal process. In our abstract model for TTL caching policies, cache evictions are driven by stopping times, which capture three popular policies (amongst others). The '$\mathcal{R}$' policy regenerates the TTLs at every object's request and maps to the LRU policy. The 'Σ' policy regenerates the TTLs only at those requests resulting in cache misses and maps to FIFO and RND policies. Besides unifying the analysis of the '$\mathcal{R}$' and 'Σ', which have already been studied (see, e.g., an overview in [2]), our stopping time model covers an emerging new policy, called '$\min(\Sigma, \mathcal{R})$', which combines the key features of Σ and $\mathcal{R}$, i.e., weak consistency guarantee and efficient utilization of cache space. In '$\min(\Sigma, \mathcal{R})$' caches, two TTLs run in parallel and an object is evicted upon the expiration of *either* of them. This policy has been recently implemented in both SDN switches (e.g., in the OpenFlow Switch Specification 1.4.0) and Amazon web services.

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.

http://dx.doi.org/10.1145/2591971.2592038.

2. OUR RESULTS

We formally define the relation of the input (request) process and the output (miss) process as a stopped sum. We state the definition in the renewal case as the corresponding definition for the non-renewal case can be stated similarly.

DEFINITION 1 (MISS/OUTPUT PROCESS).
Let a caching node with inter-request times and TTL's be given by two independent renewal processes: $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$. The corresponding miss process is also a renewal process with the same distribution as the stopped sum

$$S_\tau := X_1 + \cdots + X_\tau \ , \tag{1}$$

where τ is a stopping time[1] defined separately for each caching (TTL) policy:

$$\textit{Policy } \mathcal{R}: \quad \tau := \min\{t : X_t > T_t\} \tag{2}$$

$$\textit{Policy } \Sigma: \quad \tau := \min\{t : \sum_{s=1}^{t} X_s > T_1\} \tag{3}$$

$$\textit{Policy } \min(\Sigma,\mathcal{R}): \quad \tau := \min\{\min\{t : \sum_{s=1}^{t} X_s > T_1^{\Sigma}\}, \min\{t : X_t > T_t^{\mathcal{R}}\}\} \ . \tag{4}$$

For $\min(\Sigma,\mathcal{R})$, T_1^{Σ} and $T_t^{\mathcal{R}}$ are independent renewal processes.

2.1 Lines of Caches

The key problem to obtain caching metrics at the downstream nodes in a line of cache is to recursively characterize the inter-miss time S_τ. We aim to derive explicit solution for the Laplace transform of the stopped sum, i.e., $\mathcal{L}_\omega(S_\tau)$. One may remark that, due to the intrinsic dependencies amongst X_t's, T_t's, and the stopping time τ, the analysis of the stopped sum is conceivably quite involved. Even for stopping times, which depend only on the X_t's, higher moments are typically only known in terms of bounds (cf. Gut [5]).

Our approach is motivated by martingale theory, in which stopping times preserve certain martingale results, e.g., if L_t is a martingale and τ is a bounded stopping time then $E[L_\tau] = E[L_1]$. In particular, we can adopt the general change of measure theory based on martingales [1] to our stopping time τ and the Wald's martingale $L_t := e^{-\omega S_t}/\mathcal{L}_\omega(X)^t$. This allows to properly define a new probability measure $\tilde{\mathbb{P}}_t(F) := E[L_t 1_F]$, which incorporates the dependencies of S_τ into the measure, and allows to derive an explicit solution.

THEOREM 1 (PROOF IN [2]).
For an (a.s.) finite stopping time τ, the Laplace transform of the stopped sum S_τ from Eq. (1) is given by

$$\mathcal{L}_\omega(S_\tau) = \tilde{E}[\mathcal{L}_\omega(X)^\tau] \ . \tag{5}$$

This result fundamentally simplifies the analysis because $\tilde{E}[\mathcal{L}_\omega(X)^\tau]$ only requires to derive the pmf $\tilde{\mathbb{P}}(\tau = k)$, which is straightforward to obtain in the new space (cf. [2]).

2.2 Feedforward Cache Networks

We use the canonical definition of a MAP as a tuple of matrices $(\mathbf{D}_0, \mathbf{D}_1)$, where $\mathbf{D}_0$ contains the rates of the so-called *hidden transitions*, and $\mathbf{D}_1$ contains the (positive) rates of the so-called *active transitions*. We further assume that TTLs follow a PH distribution, which is defined by the tuple $(\mathbf{S}, \pi)$, where $\mathbf{S}$ is a $m \times m$ subintensity, $\mathbf{S}_0 := -\mathbf{S}\mathbf{1}$, and π a stochastic m-vector. We use the less standard notation of a PH generator of the form $\mathbf{P} := \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{S}_0 & \mathbf{S} \end{pmatrix}$, because this choice permits convenient expressions for the caching operation.

Using this notation, the output of any $\mathcal{R}$, Σ, and $\min(\Sigma,\mathcal{R})$ cache can be explicitly stated using the Kronecker product $\otimes$ and sum $\oplus$. We give an example for the Σ cache.

THEOREM 2 (PROOF IN [2]).
Consider a Σ-cache under requests given by $(\mathbf{D}_0, \mathbf{D}_1)$ and TTLs given by $(\mathbf{S}, \pi)$ with generator $\mathbf{P}$. Then the miss process is a MAP $(\mathbf{D}_0', \mathbf{D}_1')$ given by

$$\mathbf{D}_0' = (\mathbf{P} \oplus \mathbf{D}_0) + \mathbf{1}' \otimes D_1 \quad \textit{and} \quad \mathbf{D}_1' = \begin{pmatrix} \mathbf{0}_{n\times n} & \pi\mathbf{D}_1 \\ \mathbf{0}_{m\times n} & \mathbf{0}_{n\times m} \end{pmatrix} \ ,$$

where $\mathbf{1}'$ is a modified $n(m+1) \times n(m+1)$ identity matrix with $\mathbf{1}'_{1,1} = 0$, if M has n states and T has m transient and one absorbing states, and the $\mathbf{0}_{x\times y}$ are $x \times y$ zero matrices.

Together with corresponding theorems for $\mathcal{R}$ and $\min(\Sigma,\mathcal{R})$ caches and the known superposition property of MAPs, this method allows the characterization of miss processes at each node in a feedforward cache network. The key drawback of using MAPs is, however, that the state-spaces of the involved MAPs increase multiplicatively in the number of caches and the number of states of the TTLs' PH distribution. Nevertheless, because the underlying MAPs' matrices are sparse, the numerical complexity can be significantly reduced by using appropriate numerical methods or by formulating lower and upper bound stochastic models.

3. CONCLUSION

Our two methods jointly enable the exact analysis of caching networks and offer an interesting tradeoff: the first method is computationally fast but restricted to line networks, whereas the second has a much wider applicability but suffers from a high computational complexity.

4. REFERENCES

[1] S. Asmussen. *Applied probability and queues*, volume 2. Springer, 2003.

[2] D. S. Berger, P. Gland, S. Singla, and F. Ciucu. Exact analysis of TTL cache networks: The case of caching policies driven by stopping times. *CoRR*, abs/1402.5987, 2014.

[3] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314, 2002.

[4] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. In *Proceedings of ITC*, pages 1–8, 2012.

[5] A. Gut. *Stopped Random Walks: Limit Theorems and Applications*. Springer, 2009.

[6] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *Proceedings of IEEE INFOCOM (preprint)*, 2014. `http://arxiv.org/abs/1307.6702`.

[1] We here consider the filtration generated by $\sigma((X_1,T_1),\ldots,(X_t,T_t))$ for the $\mathcal{R}$ model; results for Σ and $\min(\Sigma,\mathcal{R})$ follow analogously.