Caches and Timelines Operate Under Heavy Traffic

Daniel S. Menasché
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil

Mark Shifrin
Ben-Gurion University
Negev, Israel

Eduardo Hargreaves
Petrobras
Rio de Janeiro, Brazil

ABSTRACT

The heavy traffic regime is a regime wherein system resources are always busy. As caches and social network timelines are intrinsically always busy, i.e., their space-shared resources are always utilized, the goal of this paper is to evaluate the implications of a simple albeit unexplored observation: *caches and timelines operate under heavy traffic.* First, we introduce the control problem of caches and timelines under heavy traffic. Then, we derive properties of the asymptotically optimal (AO) policy. In particular, we indicate that there is an AO control that threatens content diversity, as it involves maintaining contents from up to two classes in the system, leading to the so called filter bubbles.

Keywords

Heavy traffic, caches, timelines

1. INTRODUCTION

Online social network timelines are the *de facto* solution to share news in online platforms such as Facebook and Twitter. In those platforms, users are fed by posts which are organized into lists, also referred to as timelines or News Feeds. Different sources compete for the visibility over users timelines. Hence, the manager of the social network, henceforth referred to as the *admin*, faces the challenge of determining the residence time of each post in the users timelines, taking into account a complex number of factors such as revenue due to advertisements and users interests, while filtering undesired content such as fake news.

The occupation of timelines is determined based on the rate at which publishers post items and the scheduling and rejection policies adopted by the admin. Such policies, in turn, must account for the *costs of maintaining contents of each class in the users timelines.* Setting priorities for different classes, based on their costs, while ensuring first-in first-out (FIFO) placement per class, is a natural choice to determine how posts will fill timelines; *the timelines considered in this paper are therefore FIFO publisher-driven caches with priorities, which in turn are multiclass queues that by design are almost surely never empty (see Figure 1).*

Let λ_i be the rate at which new posts from class i are produced, and let $1/\mu_i$ be the reference time-to-live (TTL) of contents of type i once they reach the timeline topmost (head-of-line) position. The admin controls the occupancies of contents in the timeline by tuning the (i) scheduling and (ii) rejection policies. The policies for scheduling and rejecting posts translate into the prioritization and filtering mechanisms implemented by most of the existing social media tools. Such mechanisms are implemented by Facebook, for instance, through its News Feed algorithm.

Naturally, the topmost position of the timeline always contains a post to be retrieved by users on demand, when they access the system. Therefore, under an oblivious scheduling and rejection policy, which admits all posts and lets them remain at the topmost position for their reference TTL, it follows from Little's law that

$$\rho_i = \lambda_i/\mu_i, \qquad \sum_i \rho_i = 1. \tag{1}$$

Gaps in prior art. The above condition is known as the critical load condition. There is a vast literature analyzing systems under the critical load condition, also referred to as *heavy traffic* (see [1, 2, 5, 8] and references therein). Nonetheless, it is non-trivial to justify such assumptions for queueing systems or VoD systems, motivating their study also in the under-loaded and over-loaded regimes [5].

In the realm of space-shared resources, in contrast, the critical load condition naturally arises from the problem formulation, as by design *timelines are never empty*, opening up a number of interesting avenues for the application of heavy traffic results in a domain wherein the critical load condition is inherent to the problem [3, 6].

Contributions. Given the relationship between space shared resources and heavy traffic, our contributions are threefold.

Analytical model We propose an analytical model to study timelines and publisher-driven caches under heavy traffic. The model extends to space-shared resources, which are intrinsically always busy, motivating the critical load regime as a natural regime to work with.

Control problem formulation Leveraging the proposed model, we formulate the timeline occupancy control problem. The occupancy control involves determining the rate at which contents must be evicted from timelines, accounting for the costs to hold contents from each category.

Lack of diversity in solution We derive properties of the control problem solution. In particular, we indicate that there is a solution that threatens content diversity, as it involves maintaining contents from up to two classes in the system, leading to the so called filter bubbles.

Paper outline. The upcoming section describes the system of interest, followed by the flow of contributions outlined above, and Section 4 concludes.

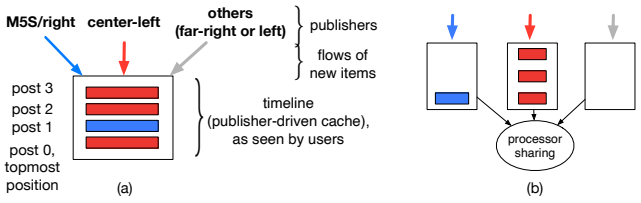

Figure 1: Timelines as publisher-driven caches: (a) system and (b) corresponding multiclass G/G/1-PS.

2. SYSTEM DESCRIPTION

Timelines. We consider a timeline, also known as a publisher-driven cache, fed by I flows. Each flow corresponds to a given content category or a given source. The flows are comprised of unique new posts. In particular, a post is never reinserted into the system [3], which is typical for user generated content.

The timeline has capacity to store up to K posts. If the timeline is full and a post arrives, it must be discarded. Posts are evicted from the timeline according to a per class first-in first-out (FIFO) policy. In particular, the least recent post in each class is a candidate to join the topmost position. The residence time of the post in the topmost position may depend on the content class, and is referred to as its time-to-live (TTL).

Cost structure. Note that the admin controls the timelines of all users. When users access the system, they retrieve and consume their timelines. Hence, the preferences of content providers (sources), users (clients) and of the owner of the social network (admin) must all be taken into account when filling up the timelines. Capturing all the aspects involved in the optimal occupation of timelines is a complex challenge, which is out of the scope of this paper. To circumvent such challenge, we assume that to each content class there is a corresponding holding cost for maintaining a post of that class in the timeline. In what follows, we briefly discuss some of the aspects involved in the setting of those costs, also referred to as shadow prices.

From the end users perspective, users incur costs if they find uninteresting content in their timelines. The interest of a user towards a content, in turn, depends on the content category as well as on its timeliness. Content from sources that typically produce fake news, for instance, must be associated to high holding costs. Nonetheless, maintaining a static post known to be authentic in the topmost position of the timeline would go counter the timeliness requirement.

Ultimately, the admin is in charge of controlling the timelines, internalizing the aforementioned costs faced by content providers and users, as well as its own costs. The problem faced by the admin is then to *minimize holding and rejection costs, accounting for shadow prices corresponding to the maintenance of posts from different content providers filling up the users timelines.*

Queues *vs* timelines. In traditional queueing systems, the higher the service rate allocated to a given class, the better the service for that class. Intuitively, the service rate for a given class should increase with respect to its holding costs. A similar relationship holds for timelines. The higher the cost for holding posts of a given class in the timeline, the shorter the period of time those posts should remain in the timeline, motivating an increase in the corresponding service rate to cause a decrease in their visibility.

3. ANALYTICAL MODEL AND SOLUTION

We model the system using a multiclass $G/G/1$ processor sharing (PS) model. News arrive into the timeline through I flows. News from flow i arrive according to a stationary process with rate λ_i. We associate shadow prices to the occupancy of the shared-resource by posts of type i. Let h_i be the shadow price (holding cost) corresponding to flow i per time unit. In addition, let r_i be the cost due to a rejection of a post from flow i. The vectors of holding costs and rejection costs are given by h and r, respectively.

We consider two versions of the problem: the shared and partitioned models, introduced in [7] and [1], respectively. Under the shared model, more natural for the analysis of timelines, a single queue of size K is shared by all flows. Under the partitioned model, I queues of size K_i, $1 \leq i \leq I$, are considered. In both cases, at any point in time the timeline comprises all the posts in all queues at that time. Note that except for the packets in the head-of-line, the multiclass G/G/1-PS model is oblivious to the order at which the jobs are stored in the queues. Correspondingly, our metric of interest, namely the relative occupancy of the timeline by each of the classes, is also insensitive to the ordering of posts.

Control problem formulation. Next, we pose the Queuing Control Problem (QCP) which consists of minimizing a combination of holding and rejection costs. Let $X_i(t)$ and $Z_i(t)$ denote the number of posts of class i at the systems at time t, and the number of posts of that class rejected by time t. Let $\hat{X}^n(t)$ and $\hat{Z}^n(t)$ be the corresponding vectors in the system scaled by a diffusion factor n (details in [1, 7]). The cost functional of the QCP scaled with diffusion factor n is

$$J^n(x_0, U^n) = \mathbb{E}\left(\int_0^\infty e^{-\alpha t}\left(h \cdot \hat{X}^n(t)dt + r \cdot d\hat{Z}^n(t)\right)\right) \tag{2}$$

where U^n is a joint scheduling and rejection policy, and α is the discount factor. The QCP value is given by

$$V^n(x_0) = \inf_{U^n} J^n(x_0, U^n). \tag{3}$$

Next, we consider a Brownian Control Problem (BCP) whose value equals the value of the corresponding QCP diffusion limit as $n \to \infty$. Our goal is to indicate properties of an asymptotically optimal (AO) policy whose value weakly converges to the BCP value.

Intuitively, an AO policy keeps in the timeline those contents that are *cheaper to hold.* We translate *cheaper content as high quality content, where quality is captured through holding costs and rejection costs* (see eq. (2)). Note that minimizing both rejection costs and holding costs may imply admitting low quality content to be rapidly evicted.

Lack of diversity. Our main formal result concerns the potential lack of diversity of contents in the timeline.

THEOREM 3.1 (LACK OF DIVERSITY). *In the shared timeline model, there exists an AO policy under which at most two types of contents share the timeline space at any point.*

The proof of the above theorem is found in [7] and relies on the notion of state space collapse [2]. In essence, the idea consists of defining an one-dimensional problem whose solution is the same as the original I dimensional problem. Indeed, in [1] it is established an identity between two limiting value functions, $V(x_0)$ and $\bar{V}(\bar{x}_0)$, corresponding to the

solutions of BCP and the Reduced BCP (RBCP), which is an one-dimensional problem.

Workload process. The key ingredient of the one-dimensional RBCP problem is the *workload* process, which tracks the remaining work in the system, measured in time units. We denote by $w \in \mathbb{R}$ the system workload. Then, the holding cost $\bar{h}(w)$ and the rejection cost $\bar{r}$ for the *one-dimensional* RBCP are given by

$$\bar{h}(w) = \min_{\xi} \left\{ \sum_{i=1}^{I} h_i \xi_i \Big| \sum_{i=1}^{I} \xi_i \leq K, \sum_{i=1}^{I} \xi_i/\mu_i = w \right\} \quad (4)$$

with $w \in [0, \mathbf{x}]$, and

$$\bar{r} = \min \left\{ \sum_{i=1}^{I} r_i z_i : z \in \mathbb{R}_+^I, \sum_{i=1}^{I} z_i/\mu_i = 1 \right\}. \quad (5)$$

Note that $\mathbf{x} \in \mathbb{R}$ is part of the solution to RBCP, and corresponds to the maximum workload in the one-dimensional system. Let $\bar{X}^n = \sum \hat{X}_i^n/\mu_i^n$. The RBCP cost functional is

$$\bar{J}(\bar{x}_0, \bar{U}) = \mathbb{E}\left(\int_0^\infty e^{-\alpha t} \left(\bar{h}(\bar{X}(t))dt + \bar{r}d\bar{Z}(t) \right) \right) \quad (6)$$

and

$$\bar{V}(\bar{x}_0) = \inf_{\bar{U}} \bar{J}(\bar{x}_0, \bar{U}). \quad (7)$$

The equality $V(x_0) = \bar{V}(\bar{x}_0)$ is established in [1] under general conditions which hold for the shared and partitioned timeline models.

Linear program. The holding cost $\bar{h}(w)$ is given by the solution of a linear program (LP), posed in (4). The parameters of the LP are the timeline size K, workload w, and the vectors of holding costs and TTLs, h and μ. Let ξ be the solution vector. The i-th element of ξ, ξ_i, corresponds to the queue lenght $\hat{X}_i^n$ of the original QCP. Theorem 3.1 follows from the fact that the above LP is such that its minimum value is attained at the edges of a simplex. Such minimum corresponds to the entire timeline, once full, being shared by at most two classes.

Figure 2 illustrates trajectories of the occupancies of timelines, corresponding to a timeline shared across 3 classes of contents (3D plots, in the left) and 2 classes of contents (2D plots, in the right). The simplexes illustrate the sets of feasible solutions of (4). When there are 3 classes, at most 2 share the timeline space at any point in time.

Structure of the derived policy. In essence, the considered policy favors the removal (service) of posts whose eviction should be prioritized (high priority). Note that whereas in standard queuing systems high priority corresponds to high quality of service (QoS), in the realm of timelines higher priority yields lower residence times and lower visibility. The scheduling priority of class i is determined as a function of the current state and h_i and the rejection policy as a function of current state, r_i and $\mathbf{x}$. The structure of the policy for the partitioned and shared models are presented in [1] and [7], respectively.

Diversity in Facebook. The occupancy of users timelines using real Facebook data has been considered in related studies, such as [3]. In [3], the authors illustrate the feasibility of tracking the occupancy of timelines and indicate that the lack of diversity predicted by Theorem 3.1 may hold in practice. Such lack of diversity has been reported in a number of other studies, and has been credited to the so called echo chambers and filter bubble effects of social networks.

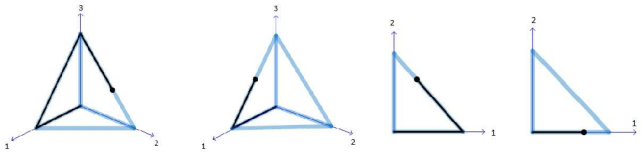

Figure 2: Lack of diversity: when there are 3 classes, at most 2 share the timeline space at any point.

Studying conditions under which the occupation predicted by the asymptotically optimal policy introduced in this paper resembles the occupancy found at real Facebook timelines, and reverse engineering the costs to match those, as well as identifying the minimum timeline size for which asymptotic behavior holds in practice (see Figure 2 in [3]) are envisioned as interesting topics for future work.

4. CONCLUSION

In the past few decades, there has been significant progress in the understanding of queueing systems under the heavy traffic regime, wherein systems are almost surely always busy [2, 4]. In this paper, we posit that even though queueing systems cannot operate under heavy traffic, space-shared resources typically work under such conditions. Leveraging a relationship between social network timelines and publisher driven caches, we have shown that the study of those systems using tools from heavy traffic literature allows us to provide novel insights into their properties. In particular, we have indicated that a lack of diversity in online social networks timelines, which may be attributed to networking effects such as filter bubbles, may also naturally emerge as an outcome of a heavy traffic model. We envision that this finding is a first step towards a broader understanding of how heavy traffic tools can be instrumental to analyze publisher driven caches and online social network timelines.

5. REFERENCES

[1] Atar, R., and Shifrin, M. An asymptotic optimality result for the multiclass queue with finite buffers in heavy traffic. *Stochastic Systems 4*, 2 (2015), 556–603.

[2] Bramson, M. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems 30*, 1-2 (1998), 89–140.

[3] Hargreaves, E., et al. Fairness in online social network timelines. *Performance Evaluation 129* (2019).

[4] Harrison, J. M., and Taksar, M. I. Instantaneous control of Brownian motion. *Mathematics of Operations research 8*, 3 (1983), 439–453.

[5] Hsieh, P.-C., Liu, X., and Hou, I. Fresher content or smoother playback? A Brownian-approximation framework for real-time video delivery in wireless networks. *arXiv:1911.00902* (2019).

[6] Shi, L., Wang, X., Ma, R. T., and Tay, Y. Weighted fair caching: Occupancy-centric allocation for space-shared resources. *Performance Evaluation 127* (2018), 194–211.

[7] Shifrin, M. An asymptotically optimal policy and state-space collapse for the multi-class shared queue. In *Proceedings of YEQT [arXiv:1503.02603]* (2015).

[8] Verloop, M., and Borst, S. Heavy-traffic delay minimization in bandwidth-sharing networks. In *INFOCOM* (2007), IEEE, pp. 1586–1594.