Heavy-traffic Delay Optimality in Pull-based Load Balancing Systems: Necessary and Sufficient Conditions

Xingyu Zhou
zhou.2055@osu.edu
The Ohio State University
Department of ECE
Columbus, OH, USA

Jian Tan
tan.252@osu.edu
Alibaba Group & The Ohio State
University
Machine Intelligence Technology
Sunnyvale, CA, USA

Ness Shroff
shroff.11@osu.edu
The Ohio State University
Departments of ECE and CSE
Columbus, OH, USA

ABSTRACT

In this paper, we consider a load balancing system under a general pull-based policy. In particular, each arrival is randomly dispatched to any server whose queue length is below a threshold; if no such server exists, then the arrival is randomly assigned to any server. We are interested in the fundamental relationship between the threshold and the delay performance of the system in heavy traffic. To this end, we first establish the following necessary condition to guarantee heavy-traffic delay optimality: the threshold needs to grow to infinity as the exogenous arrival rate approaches the boundary of the capacity region (i.e., the load intensity approaches one) but the growth rate should be slower than a polynomial function of the mean number of tasks in the system. As a special case of this result, we directly show that the delay performance of the popular pull-based policy Join-Idle-Queue (JIQ) is not heavy traffic optimal, but performs strictly better than random routing. We further show that a sufficient condition for heavy-traffic delay optimality is that the threshold grows logarithmically with the mean number of tasks in the system. This result directly resolves a generalized version of the conjecture by Kelly and Laws.

KEYWORDS

Heavy-traffic delay optimality; Pull-based; Load balancing; Necessary and sufficient conditions

ACM Reference Format:

Xingyu Zhou, Jian Tan, and Ness Shroff. 2019. Heavy-traffic Delay Optimality in Pull-based Load Balancing Systems: Necessary and Sufficient Conditions. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3309697.3331500>

1 INTRODUCTION

We consider a classical load balancing system that consists of a central dispatcher and N servers, each associated with an infinite

buffer queue and a service rate μ_n . The exogenous tasks arrive with rate λ_Σ , and upon arrival they must be immediately dispatched to one of the queues. A key to the performance of such a system is the load balancing policy it uses since it directly determines which queue the arriving tasks should join.

One important line of research has focused on the so-called heavy-traffic regime, where the exogenous arrival rate approaches the boundary of the capacity region, i.e., the heavy-traffic parameter $\epsilon = \Sigma \mu_n - \lambda_\Sigma$ approaches zero. It has been shown that well-known policies such as Join-Shortest-Queue (JSQ) and Power-of- d can achieve asymptotic optimal delay performance in the heavy-traffic regime [1–3, 6]. Under these two policies, an incoming task is assigned to a server with the shortest queue among $d \geq 2$ servers ($d = N$ for JSQ) sampled uniformly at random.

However, due to the sampling process, the amount of communication overhead is $2d$ per arrival (d for query and d for response), which is undesirable for a large value of d , especially in the JSQ policy when $d = N$. More importantly, since the dispatching decision can only be made after collecting the queue length feedback, there exists a non-zero dispatching delay, which contributes to an increase in the response time. To avoid these drawbacks, an alternative approach, often called pull-based load balancing, has received significant recent attention. Instead of actively sending queries to servers and waiting for responses, the dispatcher under a pull-based load balancing scheme passively listens to the reports from the servers. In particular, each server will report its ID to the dispatcher when it satisfies a certain condition (e.g., its queue length drops below a threshold). Then, upon task arrival, the dispatcher checks its record. If it is not empty, the dispatcher randomly removes one ID and sends the arrival to the corresponding server; otherwise, it just randomly selects a queue to join. The classical pull-based policy is the Join-Idle-Queue (JIQ) policy investigated in [5, 7], under which the dispatcher maintains a record of IDs of the idle servers (i.e., the reporting threshold is one). JIQ has been shown to enjoy a low message overhead (at most one per arrival), zero dispatching delay, and better delay performance than Power-of- d under medium loads. Nevertheless, under high loads, its delay performance degrades substantially due to lack of idle servers. This directly suggests that a varying reporting threshold with respect to the load is necessary to guarantee good delay performance in heavy traffic. Motivated by this observation, in a recent work [9], the authors propose a specific way to update the reporting threshold in a pull-based policy, which is proven to be heavy-traffic delay optimal, while still enjoying many of the nice features of JIQ.

This work has been funded in part through ONR grant N00014-17-1-2417 and NSF grants CNS-1719371, 1717060, and 1518829.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS '19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3309697.3331500>

In this paper, instead of focusing on another specific way of determining the reporting threshold, we step back and work towards answering the following fundamental question: *How would different reporting thresholds affect the (heavy traffic) delay performance of a pull-based policy?* To address this question, we take a systematic approach and summarize the main contributions as follows.

- We first present a necessary condition on the reporting threshold for the delay optimality of a pull-based policy in heavy-traffic. In particular, we show that to achieve heavy-traffic delay optimality, the reporting threshold r should grow to infinity as the heavy-traffic parameter ϵ approaches zero, however, it cannot grow too fast (slower than a polynomial function, see Theorem 2.2). An important corollary of Theorem 2.2 is that the delay performance of the JIQ policy (i.e., constant threshold $r = 1$) in heavy traffic lies *strictly* between that of any heavy-traffic delay optimal policies (e.g., JSQ) and that of random routing. This result is somewhat counter-intuitive, since at first glance one may guess that JIQ would degenerate to random routing in heavy traffic since there are hardly any idle servers in the system. However, it turns out that it is not true, and this allows us to get a sharp characterization of the JIQ policy in heavy traffic.
- We then establish a sufficient condition on the reporting threshold for heavy-traffic delay optimality of pull-based policies. Specifically, we show that a logarithmic growth rate of the reporting threshold with respect to the mean number of tasks in the system is sufficient to guarantee the steady-state delay optimality in heavy traffic (see Theorem 2.3). This result directly resolves a conjecture by Kelly and Laws in [4]. In particular, the authors in [4] consider a two-server system with Poisson arrivals and exponential service under a varying reporting threshold. They conjecture that as long as the threshold is greater than a specified constant times the logarithm of the mean number of tasks in the system, then asymptotic delay optimality holds in heavy traffic. Thus, our result not only resolves the conjecture but generalizes it to any fixed finite number of servers with general arrival and service distributions.

2 MAIN RESULTS

In this paper, we focus on the following general pull-based policy.

Definition 2.1. Join-Below-Threshold (JBT) policy is composed of the following components:

- Each server is initialized with an empty queue, and a corresponding ID in the local memory of the dispatcher.
- Upon new arrivals at the beginning of each time-slot, the dispatcher checks the available IDs in memory. If one or more IDs exist, it removes one uniformly at random, and sends all the new arrivals to the corresponding server. Otherwise, all the new arrivals are dispatched uniformly at random to one of the servers in the system.
- Each server reports its ID to the dispatcher at the end of each time-slot if its queue length is below the threshold, and the dispatcher does not contain its ID (see the remark below on this condition).

- For the case of heterogeneous servers, in (c) each server also sends its μ_n to the dispatcher and in (b) instead of choosing the ID uniformly at random, the dispatcher selects the ID in proportion to the service rate. Specifically, if the ID of server i is in $m(t)$, the probability for server i to be chosen is $\mu_i / \sum_{j \in m(t)} \mu_j$.

We first establish the following necessary condition.

THEOREM 2.2. *Consider a load balancing system with homogeneous servers under the JBT policy.*

- Suppose the threshold r is any constant in $[1, \infty)$, then we have

$$\liminf_{\epsilon \downarrow 0} \epsilon \mathbb{E} \left[\sum_{n=1}^N \bar{Q}_n^{(\epsilon)} \right] > \frac{\zeta}{2} \quad (1)$$

and

$$\limsup_{\epsilon \downarrow 0} \epsilon \mathbb{E} \left[\sum_{n=1}^N \bar{Q}_n^{(\epsilon)} \right] < \lim_{\epsilon \downarrow 0} \epsilon \mathbb{E} \left[\sum_{n=1}^N \bar{Q}_{n, \text{Rand}}^{(\epsilon)} \right], \quad (2)$$

where $\bar{Q}_{\text{Rand}}^{(\epsilon)}$ is the steady-state vector under random routing policy.

- Suppose the threshold $r^{(\epsilon)} = (1/\epsilon)^{1+\alpha}$ for any constant $\alpha > 0$, then we have

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E} \left[\sum_{n=1}^N \bar{Q}_n^{(\epsilon)} \right] = \lim_{\epsilon \downarrow 0} \epsilon \mathbb{E} \left[\sum_{n=1}^N \bar{Q}_{n, \text{Rand}}^{(\epsilon)} \right]. \quad (3)$$

Then, we successfully prove the following sufficient condition.

THEOREM 2.3. *Consider a load balancing system under the JBT policy. Suppose that the threshold r satisfies $r^{(\epsilon)} \geq K \log(1/\epsilon)$ and $r^{(\epsilon)} = o(1/\epsilon)$, where K is some pre-defined constant, then JBT is heavy-traffic delay optimal in steady state.*

The full version is available at [8].

REFERENCES

- Hong Chen and Heng-Qing Ye. 2012. Asymptotic optimality of balanced routing. *Operations research* 60, 1 (2012), 163–179.
- Atilla Eryilmaz and R Srikant. 2012. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems* 72, 3-4 (2012), 311–359.
- G Foschini and J. Salz. 1978. A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications* 26, 3 (1978), 320–327.
- FP Kelly and CN Laws. 1993. Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing systems* 13, 1-3 (1993), 47–86.
- Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R Larus, and Albert Greenberg. 2011. Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation* 68, 11 (2011), 1056–1071.
- Siva Theja Maguluri, R Srikant, and Lei Ying. 2014. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Performance Evaluation* 81 (2014), 20–39.
- Alexander L Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems* 80, 4 (2015), 341–361.
- Xingyu Zhou, Jian Tan, and Ness Shroff. 2018. Heavy-traffic Delay Optimality in Pull-based Load Balancing Systems: Necessary and Sufficient Conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 3 (2018), 41.
- Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. 2017. Designing Low-Complexity Heavy-Traffic Delay-Optimal Load Balancing Schemes: Theory to Algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 2 (2017), 39.