Research Statement

Xingyu Zhou

Let's consider the following puzzle. Say that you are shopping at your favorite supermarket and you are going to ready to check out. In front of you, there are five counters. Now, the question is which counter would you like to choose? Maybe, you will first search for an empty counter (i.e., without any people there). But suppose you are shopping on a Sunday afternoon, when long lines are quite common. In this case, your chances of finding an empty counter are small. Then, you might turn to find a counter that has the fewest number of people waiting there. This strategy is natural if there are only a few counters, and in fact is called Join Shortest Queue (JSQ) in the literature. However, this would not be easy to do so if there were a large number of counters. This is exactly the scenario of **load balancing** in every large data center or cloud system, e.g., Goolge Cloud, Amazon AWS, and Microsoft Azure. In all of these systems, there exists a load balancer or dispatcher, which adopts a certain load balancing scheme and is responsible for dispatching user requests to one of the hundreds of servers. A good load balancing scheme can not only maximize the number requests the system can handle, but more importantly, minimize the delay each user request experience (e.g., you would not be happy if you had to spend 20 minutes in the check out line). Hence, every cloud service provider aims to design effective load balancing schemes that are able to dispatch each user request to the right server.

My Ph.D. research has focused on the critical need mentioned above in todays large data center and cloud computing systems. **To that end, I have focused on analyzing and developing load balancing strategies that aim to minimize the delay while having low cost, and can also be easily implemented in practical systems.**

Backgrounds and contributions: The analysis and design of load balancing schemes have attracted strong attention in recent years, mainly spurred by the stringent delay requirement in practical applications such as Web service, cloud computing, distributed caching system, and grid computing. To this end, I take a theoretically well-founded and asymptotic analysis approach geared towards the development of unified methods and fundamental principles that are applicable to a wide range of load balancing applications. In sum, my research activities during my Ph.D. can be classified as follows:

- *Algorithm design:* This is aimed at designing low-complexity and flexible load balancing schemes with rigorous theoretical guarantees on performance optimality. (i) To achieve the advantages of both push-based and pull-based schemes at the same time, we propose a new load balancing scheme that carries out the principles emanating from the theoretical foundations. (ii) To address the fact that existing policies are often too restrictive, we identify a class of load balancing policies that enjoy a nice trade-off between flexibility and performance guarantee.

- *Performance analysis:* This is aimed at the mathematical analysis of load balancing schemes with proper metrics to establish tight characterizations of performance and to reveal the limits of their applicability. (i) We establish the necessary and sufficient conditions for a general class of pull-based load balancing schemes to achieve an optimal delay in heavy traffic, which resolves a generalized version of the conjecture by Kelly and Laws proposed over 25 years ago. (ii) We devise a refined performance evaluation metric, which allows us not only to successfully distinguish between good and poor load balancing schemes, but to guide the design of new effective schemes in practice.

My research results have not only drawn interest from academics but also attracted cooperation from industries. Fortunately, I have been invited by Prof. Mor Harchol-Balter to give a talk at the SQUALL seminar of CMU, and invited by Prof. Adam Wierman to present my work at the RSRG Seminar of Caltech. Recently, I was also invited by Prof. Siva Theja Maguluri to give a talk at the INFORMS annual meeting. Meanwhile, we are now working with Facebook Core Systems to establish a cooperation on load balancing.

1 Ph.D. Research on Load Balancing

In this section, I will present the main progress made by my Ph.D. research in several fundamental problems on load balancing via (i) designing delay-optimal and scalable schemes, (ii) resolving a long-standing open conjecture, (iii) challenging the conventional wisdom.

1.1 Designing delay-optimal and scalable schemes

Nowadays, millions of users are shopping on websites like Amazon and Taobao, and watching videos on YouTube and Netflix. With the continuing growth in the amount of data, the number of servers in every nowadays cloud system or data center is often quite large. A major challenge for all cloud service providers is to develop load balancing solutions that can effectively scale with the number of servers (e.g., recall that the scheme in our beginning puzzle that tries to join the counter with the fewest people is only effective for a small number of counters).

To resolve this challenge, **I developed a class of flexible load balancing schemes that enjoys a smooth trade-off between scalability and delay performance** [4, 7]. This is achieved by relaxing a restrictive condition in previous works. Almost all the previous literature rely on a restrictive condition called *single-dimensional state-space collapse* to establish delay optimality in heavy traffic. However, we show that even under a more relaxed condition called *multi-dimensional state-space collapse*, asymptotic delay optimality can still be achieved. Then, I successfully explored the flexibility and scalability enabled by this relaxation, which results in a class of new load balancing schemes that enjoy good scalability and delay performance. In another line of works [10, 11], I went even further and developed a unified analytical framework for designing load balancing algorithms that can simultaneously achieve low latency, low complexity, and low communication overhead. The key idea behind this framework is to attain the both benefits of push-based and pull-based load balancing at the same time. This framework has been cited in a recent survey paper with a separate paragraph highlighting our contributions [3]. Meanwhile, the Facebook Core Systems team, a division building foundation for large scale distributed systems in Facebook, is planning to apply my proposed framework to build a private cloud where all Facebook products (Facebook, Messenger, Instagram, WhatsApp, Workplace), plus internal services (AI/ML, Big data, Messaging, Search, Video, Stream processing, Payment, Ranking etc) depend on to scale.

1.2 Resolving a Long-standing Open Conjecture

Let us say that you are watching the newest season of *House of Cards* on Netflix, which releases all the episodes at once. As a result, many fans binge-watch in order to finish the entire season within a day or a two. This often causes a very heavy load on the data management system that Netflix is using during a short time. If this scenario is mis-handled you might experience a large delay or lag during the most exciting moments of your favorite episode. Of course, this scenario would not be desirable to either you (as the customer) or Netflix (as the company). In order to avoid a large delay, especially in the case of this **heavy-traffic scenario**, the engineers at Netflix have recently started to consider replacing the previous static load balancing scheme with a new dynamic load balancing scheme, where there exists a threshold for making load balancing decisions that could adaptively change with the current traffic load [2]. However, they are struggling on how to adapt the threshold and yet guarantee good delay performance. In fact, the question of how to choose the correct threshold-type load balancing schemes has been a **long-standing open problem**. Over 25 years ago, Prof. Frank Kelly and C.N. Laws conjectured that in order to obtain the optimal delay performance in heavy traffic, the threshold should increase logarithmically with respect to the average number of user requests in the system. From then on, this open problem has drawn the attention of many preeminent researchers around the world from various disciplines Engineering, Mathematics, Operation Research, Computer Science and Statistics. **I am the first researcher to resolve this open problem in my recent ACM Sigmetrics 2019 paper**. In fact, my result is much more general and hence applicable to

a much wider range of problems that are not considered in the original conjecture. Almost all the previous attempts on this problem adopt a method called diffusion approximations, which is a common choice for analyzing load balancing systems. However, I made entirely novel extensions on a completely different approach that allowed me not only to resolve the original conjecture, but to offer many useful insights from the first principle that can be utilized to solve a wide range of load balancing problems [5, 8].

1.3 Challenging the Conventional Wisdom

In the performance analysis community, a metric called *heavy-traffic delay optimality* is often regarded as the key analytical criterion to evaluate the delay performance of various systems. The traditional wisdom is that if a load balancing scheme is delay optimal under very high loads, it would also perform well on the low or moderate loads scenario. However, I rigorously showed that this is not the case for load balancing systems. Specifically, **I showed that heavy-traffic delay optimality in fact is a very coarse metric that does not necessarily imply good delay performance in practical scenarios of interest for load balancing schemes** [6, 9]. Therefore, even if a load balancing scheme were heavy-traffic delay optimal, its empirical delay performance could be quite poor. To overcome this gap, I proposed a refined metric called *degree of queue imbalance*, which measures the queue lengths difference among all the servers in steady-state. I then rigorously showed that this new metric is a good indicator of the empirical delay performance of load balancing schemes. In particular, this new metric is able to explicitly differentiate between good and poor load balancing schemes that are all optimal under the old metric. Although this work uses deep mathematical theories, the end result provides a clear methodology of how one can design simple load balancing schemes that result in optimal delay performance. That is, a practically good load balancing scheme should not only be heavy-traffic delay optimal, but also enjoys a low degree of queue imbalance.

2 Future Research

Moving forward, I will continue working on the fundamental and open problems on load balancing and also branch out to explore general decision problems with learning.

Load balancing using delayed information: The existing load balancing schemes mainly rely on up-to-date and accurate information about the sate of the system to make the decision. Nevertheless, this condition is often not satisfied in a lot of practical cases. For instance, because of the physical separation between the load balancer and servers, communication delays, processing effects, or periodic updates from the servers, the load balancer may only have access to the information about the delayed and inaccurate states of the system. In this setting, as stated in [1], a lot of interesting problems remain open. I am highly interested in exploring this direction in the future, and in fact, one of my on-going work is trying to solve exactly one of the open problems listed in [1]. More specifically, I am looking forward to investigating (1) the impact of message delays in pull-based load balancing schemes; (2) delay performance of load balancing schemes that adopt local memory; (3) the analysis and design of load balancing schemes that are applicable in the distributed case. Moreover, I also plan to apply our proposed metric *degree of queue imbalance* to give a sharp characterization of system performance beyond the average delay when using delayed information.

Load balancing in big-data analytics: Nowadays, big data analytics offers a nearly endless source of business and informational insight, which can lead to operational improvement and valuable opportunity for revenue increase. The advance of big data analytics relies heavily on large-scale distributed computing and storage systems. In this setting, data locality (which refers to the fact that request often needs to be executed on the node or server that stores the data) becomes a key consideration in design of good load balancing schemes. In most of previous schemes, they often assume that any request can be equally well handled by any server. However, with data locality, some servers are better equipped to process certain requests because of affinity or compatibility relations. Moreover, for distributed storage systems (e.g., key-values stores), a certain key should only be assigned to a certain node by partitioning. Faced with these constraints, I am

interested in (1) how to design efficient load balancing schemes for distributed computing systems with data locality; (2) how to design reduce the load imbalance across nodes in storage systems by combining load balancing and caching schemes.

Load balancing and machine learning: Machine learning, especially reinforcement learning has been applied in a lot of modern applications in data center resource managements. It is also possible to be integrated with load balancing. One interesting problem would be how to adopt learning to efficiently handle load balancing in heterogeneous systems. However, I will avoid just plugging learning directly with load balancing without considering any useful insights. On the other hand, load balancing also plays a big role in the computing of training tasks deep learning tasks. Thus, one future problem I hope to explore is how can we design load balancing schemes that can provide good performance for training tasks.

References

[1] David Lipshutz. Open problem—load balancing using delayed information. *Stochastic Systems*, 2019.

[2] Mike Smith. Rethinking netflixs edge load balancing. https://medium.com/netflix-techblog/netflix-edge-load-balancing-695308b5548c, Netflix Technology Blog, Sep. 28, 2018.

[3] Mark van der Boor, Sem C Borst, Johan SH van Leeuwaarden, and Debankur Mukherjee. Scalable load balancing in networked systems: A survey of recent advances. *arXiv preprint arXiv:1806.05444*, 2018.

[4] Xingyu Zhou, Jian Tan, and Ness Shroff. Flexible load balancing with multi-dimensional state-space collapse: Throughput and heavy-traffic delay optimality. *Performance Evaluation*, 127:176–193, 2018.

[5] Xingyu Zhou, Jian Tan, and Ness Shroff. 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):41, 2018.

[6] Xingyu Zhou, Fei Wu, Jian Tan, Kannan Srinivasan, and Ness Shroff. Degree of queue imbalance: Overcoming the limitation of heavy-traffic delay optimality in load balancing systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):21, 2018.

[7] Xingyu Zhou, Jian Tan, and Ness Shroff. Flexible load balancing with multi-dimensional state-space collapse: Throughput and heavy-traffic delay optimality. In *Proc. of IFIP Performance*. Toulouse, France, Dec. 2018.

[8] Xingyu Zhou, Jian Tan, and Ness Shroff. Heavy-traffic delay optimality in pull-based load balancing systems: Necessary and sufficient conditions. In *Proc. of ACM Sigmetrics/IFIP Performance*. Phoenix, Arizona, USA, to appear, June 2019.

[9] Xingyu Zhou, Fei Wu, Jian Tan, Kannan Srinivasan, and Ness Shroff. Degree of queue imbalance: Overcoming the limitation of heavy-traffic delay optimality in load balancing systems. In *Proc. of ACM SIGMETRICS*. Irvine, California, USA, June 2018.

[10] Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. Designing low-complexity heavy-traffic delay-optimal load balancing schemes: Theory to algorithms. In *Proc. of ACM Sigmetrics*. Irvine, California, USA, June 2018.

[11] Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. 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):39, 2017.