Load Balancing in Distributed Service System: A Survey

Xingyu Zhou

The Ohio State University

zhou.2055@osu.edu

November 21, 2016

Outline

Outline

Load Balancing in Distributed Service System

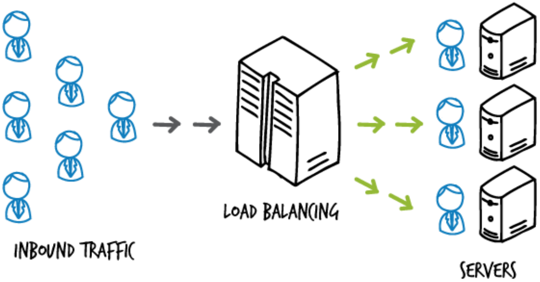


Figure: A typical model in cloud system

- **Load balancing:** Choose the right server(s) when requests coming.

Load Balancing in Distributed Service System

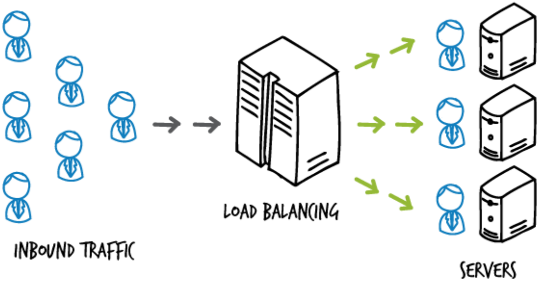


Figure: A typical model in cloud system

- **Load balancing:** Choose the right server(s) when requests coming.
  - It is the key to optimize resource use, maximize throughput, reduce response time in cloud system.

Load Balancing in Distributed Service System

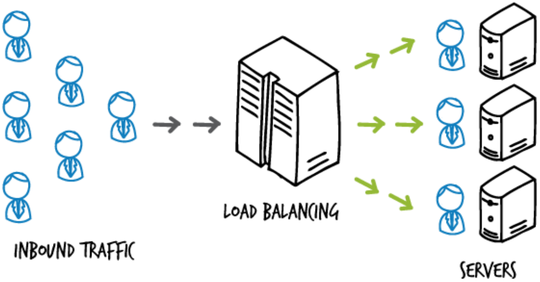


Figure: A typical model in cloud system

- **Load balancing:** Choose the right server(s) when requests coming.
  - It is the key to optimize resource use, maximize throughput, reduce response time in cloud system.
  - It becomes more and more critical due to explosive increase in the number of servers and traffic in cloud system.

Different Load Balancing Algorithms

- **Push-based:** The load balancer sends probing message to servers, and gets feedback from servers about the queue length or workload information.
  - Join-Shortest-Queue (JSQ): Upon each new arrival, the load balancer sends this new arrival to the queue with the minimum queue length.
  - Power-of-d: Upon each new arrival, the load balancer randomly selects d queues, and send the new arrival to the minimum queue among the d selected queues
- **Pull-based:** The servers send message to the load balancer when certain condition satisfied to notify the load balancer that they are ready for new arrival.
  - Join-the-Idle-Queue (JIQ): Whenever one server becomes idle, it sends a idle message to the load balancer. Upon a new arrival, if there is idle message at the load balancer, it was sent to a randomly chosen idle queue; otherwise, it was sent to a queue randomly selected among all the queues.

Outline

1. Introduction and Motivation
   - What is Load Balancing?

2. A Survey of Previous Works on Load Balancing
   - Big Picture
   - Classical Regime
   - Large System Regime
   - Heavy-traffic Regime
   - Many-server Heavy-traffic Regime

Outline

Big Picture

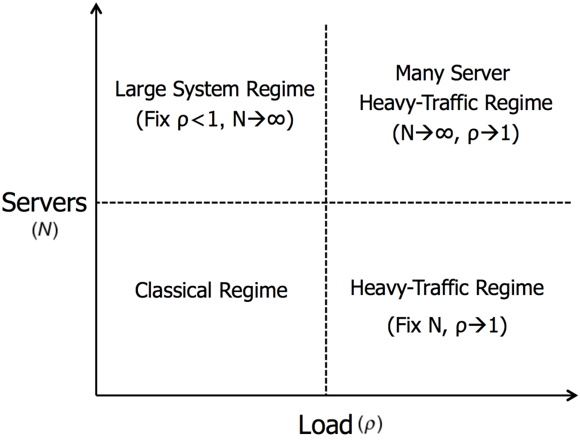
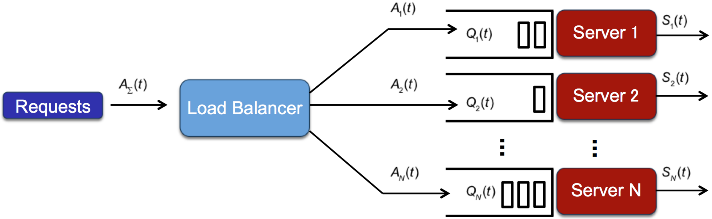


Figure: Different Regimes in Distributed Service System

Outline

Classical Regime



- N is fixed, and $\rho = \frac{\lambda_\Sigma}{N} < 1$ is fixed.

Typical Results

Join-Shortest-Queue (JSQ) is optimal in **stochastic order** sense:

- Winston'77 [17] first showed that JSQ is optimal in stochastic order for Poisson arrival and exponential service.

Typical Results

Join-Shortest-Queue (JSQ) is optimal in **stochastic order** sense:

- Winston'77 [17] first showed that JSQ is optimal in stochastic order for Poisson arrival and exponential service.
- Weber'78 [15] extended the result to arbitrary renewal arrival and non-decreasing failure rate service process.

Typical Results

Join-Shortest-Queue (JSQ) is optimal in **stochastic order** sense:

- Winston'77 [17] first showed that JSQ is optimal in stochastic order for Poisson arrival and exponential service.
- Weber'78 [15] extended the result to arbitrary renewal arrival and non-decreasing failure rate service process.
- However, Whitt'86 [16] has shown that JSQ is not optimal for general service process, even when the arrival is Poisson.

Typical Results

Join-Shortest-Queue (JSQ) is optimal in **stochastic order** sense:

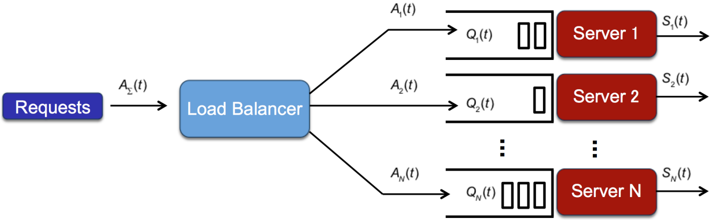
- Winston'77 [17] first showed that JSQ is optimal in stochastic order for Poisson arrival and exponential service.
- Weber'78 [15] extended the result to arbitrary renewal arrival and non-decreasing failure rate service process.
- However, Whitt'86 [16] has shown that JSQ is not optimal for general service process, even when the arrival is Poisson.
- Towsley et al,'95 [6] used sample path method, i.e., coupling and majorization to show the similar results. (*This method is the one I like most*).

Methodology

- First, apply coupling to the two queuing systems $Q^o(t)$ and $Q^\pi(t)$ under the optimal and any other policy in a certain class, respectively.
- Second, guess the proper initial relation between the two systems. For example, $Q^o(0) \prec_w Q^\pi(t)$.
- Third, verify that the initial relation hold for all t under any given sample path $\omega \in \Omega$.
- Finally, we can conclude the stochastic order relation under any functions that preserve the order. For example, the increasing and Schur convex functions.

Outline

Large System Regime



- N goes to ∞, and $\rho = \frac{\lambda_\Sigma}{N} < 1$ is fixed.
- For example, each server is with exponential service time with rate 1. The arrival process is Poisson arrival with rate $\lambda_\Sigma = \lambda N$, where $\lambda < 1$.

Typical Results

Push-based:

- The two authors Vvedenskaya et al,'96 [14] and Mitzenmacher'96 [9] independently proposed the power-of-d algorithm. They derived that in steady-state the probability for a queue to have at least k tasks is of the form $p_k = \rho^{(d^k-1)/(d-1)}$, which has a huge improvement over random selection which is $p_k = \rho^k$. It was generalized to batch-sampling in Ying, Srikant, Kang' 15 [18].

Pull-based:

Typical Results

Push-based:

- The two authors Vvedenskaya et al,'96 [14] and Mitzenmacher'96 [9] independently proposed the power-of-d algorithm. They derived that in steady-state the probability for a queue to have at least k tasks is of the form $p_k = \rho^{(d^k-1)/(d-1)}$, which has a huge improvement over random selection which is $p_k = \rho^k$. It was generalized to batch-sampling in Ying, Srikant, Kang' 15 [18].
- Mukherjee et al,'16 [11] showed that if we ensure that $\frac{1}{d(N)} \to 0$ as $N \to \infty$, then all these power-of-d algorithms will achieve the same universal steady-state distribution as $p_1 = \rho$ and $p_k = 0$, for all $k \geq 2$, which of course includes the JSQ in which case $d(N) = N$. The system behaves like a $M/M/\infty$. (Asymptotic zero delay !)

Pull-based:

Typical Results

Push-based:

- The two authors Vvedenskaya et al,'96 [14] and Mitzenmacher'96 [9] independently proposed the power-of-d algorithm. They derived that in steady-state the probability for a queue to have at least k tasks is of the form $p_k = \rho^{(d^k-1)/(d-1)}$, which has a huge improvement over random selection which is $p_k = \rho^k$. It was generalized to batch-sampling in Ying, Srikant, Kang' 15 [18].
- Mukherjee et al,'16 [11] showed that if we ensure that $\frac{1}{d(N)} \to 0$ as $N \to \infty$, then all these power-of-d algorithms will achieve the same universal steady-state distribution as $p_1 = \rho$ and $p_k = 0$, for all $k \geq 2$, which of course includes the JSQ in which case $d(N) = N$. The system behaves like a $M/M/\infty$. (Asymptotic zero delay !)

Pull-based:

- Join-the-idle-queue (JIQ) was first proposed by Lu et al,' 11 [7] with multiple load balancers.

Typical Results

Push-based:

- The two authors Vvedenskaya et al,'96 [14] and Mitzenmacher'96 [9] independently proposed the power-of-d algorithm. They derived that in steady-state the probability for a queue to have at least k tasks is of the form $p_k = \rho^{(d^k-1)/(d-1)}$, which has a huge improvement over random selection which is $p_k = \rho^k$. It was generalized to batch-sampling in Ying, Srikant, Kang' 15 [18].
- Mukherjee et al,'16 [11] showed that if we ensure that $\frac{1}{d(N)} \to 0$ as $N \to \infty$, then all these power-of-d algorithms will achieve the same universal steady-state distribution as $p_1 = \rho$ and $p_k = 0$, for all $k \geq 2$, which of course includes the JSQ in which case $d(N) = N$. The system behaves like a $M/M/\infty$. (Asymptotic zero delay !)

Pull-based:

- Join-the-idle-queue (JIQ) was first proposed by Lu et al,' 11 [7] with multiple load balancers.
- Stolyar'15 [13] has shown that $p_1 = \rho$ and $p_k = 0$, for all $k \geq 2$, which is the same as JSQ policy. The system behaves like a $M/M/\infty$. (Asymptotic zero delay !)

Typical Results (Cont'd)

- Push-based policies do not store idle messages, but it will have to sample the queue length information upon each arrival, which will incur a high communication message rate.

Typical Results (Cont'd)

- Push-based policies do not store idle messages, but it will have to sample the queue length information upon each arrival, which will incur a high communication message rate.
- Pull-based policies do not sample for each arrival, but it will have to store the idle queue message for future arrival.

Typical Results (Cont'd)

- Push-based policies do not store idle messages, but it will have to sample the queue length information upon each arrival, which will incur a high communication message rate.
- Pull-based policies do not sample for each arrival, but it will have to store the idle queue message for future arrival.

Delay, memory, and message trade-off

Typical Results (Cont'd)

- Push-based policies do not store idle messages, but it will have to sample the queue length information upon each arrival, which will incur a high communication message rate.
- Pull-based policies do not sample for each arrival, but it will have to store the idle queue message for future arrival.

Delay, memory, and message trade-off

- Gamarnik, Tsitsiklis, Zubeldia,'16 [5] has shown the fundamental trade-off between message rate and memory to ensure the asymptotic zero delay. (*I like this paper the most.*)

Typical Results (Cont'd)

- Push-based policies do not store idle messages, but it will have to sample the queue length information upon each arrival, which will incur a high communication message rate.
- Pull-based policies do not sample for each arrival, but it will have to store the idle queue message for future arrival.

Delay, memory, and message trade-off

- Gamarnik, Tsitsiklis, Zubeldia,'16 [5] has shown the fundamental trade-off between message rate and memory to ensure the asymptotic zero delay. (*I like this paper the most.*)
- In particular, they has shown that the necessary condition for asymptotic zero delay under any symmetry policy is either (i) or (ii)
  - (i) the message rate grows super-linearly with N.
  - (ii) the memory grows super-logarithmically with N.

Summary for Large System Regime

Policy	Memory (bits)	Message rate	Delay
RRobin [13]	$\log_2(N)$	0	> 0
JSQ	0	$2\lambda N^2$	0
JSQ(d) [11]	0	$2d\lambda N$	> 0
JSQ(d, b) [12]	$\Omega(b \log_2(N))$	$2d\lambda N$	> 0
Pull-based [14]	N	λN	0
High Memory	$\omega(\log_2(N))$	λN	0
High Message	$C \log_2(N)$	$\omega(N)$	0
Constrained	$C \log_2(N)$	$\mu' \lambda N$	> 0

Source: *Gamarnik, Tsitsiklis, Zubeldia'16 [5]*

Methodology

- First, derive the fluid model and then show the unique and existence of fluid solution.

Methodology

- First, derive the fluid model and then show the unique and existence of fluid solution.
- Second, show that the process is almost surely close to the previous fluid solution.

Methodology

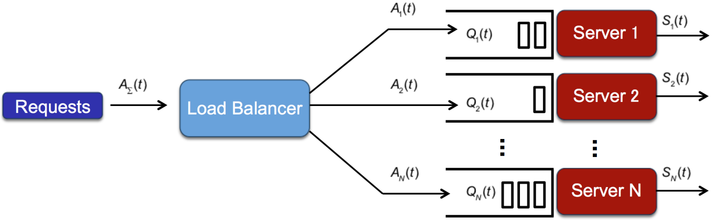
- First, derive the fluid model and then show the unique and existence of fluid solution.
- Second, show that the process is almost surely close to the previous fluid solution.
- Third, show that the process is positive recurrent for finite N.

Methodology

- First, derive the fluid model and then show the unique and existence of fluid solution.
- Second, show that the process is almost surely close to the previous fluid solution.
- Third, show that the process is positive recurrent for finite N.
- Finally, inter-change of the limits, i.e., $N \to \infty$ and $t \to \infty$.

Outline

Heavy-traffic Regime



- N is fixed, and $\rho = \frac{\lambda_\Sigma}{N} \to 1$.

Typical Results

Diffusion approximation approach:

- Diffusion approximations for JSQ under two stations was first investigated by Foschini'78 [4] for Poisson arrival λ and exponential service μ. Reiman'84 [12] extended to renewal arrivals and general service.

Lyapunov drift condition approach:

Typical Results

Diffusion approximation approach:

- Diffusion approximations for JSQ under two stations was first investigated by Foschini'78 [4] for Poisson arrival λ and exponential service μ. Reiman'84 [12] extended to renewal arrivals and general service.
- In particular, they has shown that under heavy traffic, the diffusion limit of the system is a reflected Brownian motion, which is the same as the diffusion limit of a single queue with arrival rate λ and service rate 2μ.

Lyapunov drift condition approach:

Typical Results

Diffusion approximation approach:

- Diffusion approximations for JSQ under two stations was first investigated by Foschini'78 [4] for Poisson arrival λ and exponential service μ. Reiman'84 [12] extended to renewal arrivals and general service.
- In particular, they has shown that under heavy traffic, the diffusion limit of the system is a reflected Brownian motion, which is the same as the diffusion limit of a single queue with arrival rate λ and service rate 2μ.
- Therefore, under JSQ the system behave the same as a M/M/2 queue, hence has half the delay compared to random routing.

Lyapunov drift condition approach:

Typical Results

Diffusion approximation approach:

- Diffusion approximations for JSQ under two stations was first investigated by Foschini'78 [4] for Poisson arrival λ and exponential service μ. Reiman'84 [12] extended to renewal arrivals and general service.
- In particular, they has shown that under heavy traffic, the diffusion limit of the system is a reflected Brownian motion, which is the same as the diffusion limit of a single queue with arrival rate λ and service rate 2μ.
- Therefore, under JSQ the system behave the same as a M/M/2 queue, hence has half the delay compared to random routing.

Lyapunov drift condition approach:

- This approach was first proposed by Eryilmaz and Srikant'12 [1]. They has shown that the lower bound and upper bound of the first moment under JSQ in heavy traffic coincides, hence it ensure the first moment heavy-traffic optimality of JSQ.

Typical Results

Diffusion approximation approach:

- Diffusion approximations for JSQ under two stations was first investigated by Foschini'78 [4] for Poisson arrival λ and exponential service μ. Reiman'84 [12] extended to renewal arrivals and general service.
- In particular, they has shown that under heavy traffic, the diffusion limit of the system is a reflected Brownian motion, which is the same as the diffusion limit of a single queue with arrival rate λ and service rate 2μ.
- Therefore, under JSQ the system behave the same as a M/M/2 queue, hence has half the delay compared to random routing.

Lyapunov drift condition approach:

- This approach was first proposed by Eryilmaz and Srikant'12 [1]. They has shown that the lower bound and upper bound of the first moment under JSQ in heavy traffic coincides, hence it ensure the first moment heavy-traffic optimality of JSQ.
- With the same approach, Maguluri and Srikant'14 [8] has shown the the first moment heavy-traffic optimality of power-of-d.

Methodology

Diffusion approximation approach:

- First, under diffusion scaling, show the queue process has a limit in Skorokhod space. e.g., reflected Brownian motion process. In particular, scaling time and number, try to see some formula of the limiting system. Then apply reflected mapping and the continuity to show weak convergence.

Lyapunov drift condition approach:

Methodology

Diffusion approximation approach:

- First, under diffusion scaling, show the queue process has a limit in Skorokhod space. e.g., reflected Brownian motion process. In particular, scaling time and number, try to see some formula of the limiting system. Then apply reflected mapping and the continuity to show weak convergence.
- Second, try to conjecture or prove the interchange of limit to show convergence to steady-state distribution.

Lyapunov drift condition approach:

Methodology

Diffusion approximation approach:

- First, under diffusion scaling, show the queue process has a limit in Skorokhod space. e.g., reflected Brownian motion process. In particular, scaling time and number, try to see some formula of the limiting system. Then apply reflected mapping and the continuity to show weak convergence.
- Second, try to conjecture or prove the interchange of limit to show convergence to steady-state distribution.

Lyapunov drift condition approach:

- First, show the positive recurrent of the process and bounded stationary moments.

Methodology

Diffusion approximation approach:

- First, under diffusion scaling, show the queue process has a limit in Skorokhod space. e.g., reflected Brownian motion process. In particular, scaling time and number, try to see some formula of the limiting system. Then apply reflected mapping and the continuity to show weak convergence.
- Second, try to conjecture or prove the interchange of limit to show convergence to steady-state distribution.

Lyapunov drift condition approach:

- First, show the positive recurrent of the process and bounded stationary moments.
- Second, show steady-state collapse in the sense of the first moment sense.

Methodology

Diffusion approximation approach:

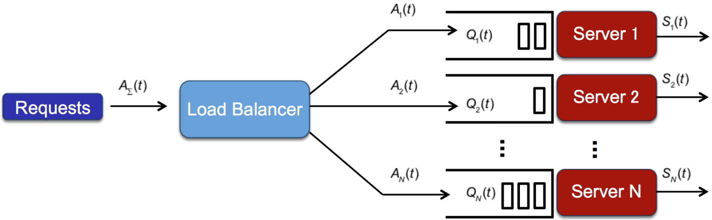
- First, under diffusion scaling, show the queue process has a limit in Skorokhod space. e.g., reflected Brownian motion process. In particular, scaling time and number, try to see some formula of the limiting system. Then apply reflected mapping and the continuity to show weak convergence.
- Second, try to conjecture or prove the interchange of limit to show convergence to steady-state distribution.

Lyapunov drift condition approach:

- First, show the positive recurrent of the process and bounded stationary moments.
- Second, show steady-state collapse in the sense of the first moment sense.
- Finally, show the upper bound obtained via drift condition in steady-state meets the lower bound with the help of state collapse.

Outline

Many-server Heavy-traffic Regime (Halfin-Whitt Regime)



- $N \to \infty$, and $\rho_N = \frac{\lambda_\Sigma}{N} \to 1$, with the speed relation $\lim_{N\to\infty} \sqrt{N}(1 - \rho_N) = \beta$ for some fixed $\beta > 0$.

Typical Results

Push-based:

- Gamarnik et al,'15 [2] first study the JSQ in Halfin-Whitt regime. They have shown that the number of idle servers and the number of servers with exact two customers are both of order $O(\sqrt{n})$, and all the queues with longer queue lengths will decay to zero within constant time, under certain reasonable initial conditions.
- Gamarnik et al,'16 [3] first study the power-of-d in Halfin-Whitt regime. They have shown that the majority of queues have steady state length at least $\log_d(1-\rho_N)^{-1} - O(1)$ with probability approaching to 1 as $N \to \infty$

Pull-based:

- Mukherjee et al,'15 [10] first study the JIQ in Halfin-Whitt regime. They have shown that JIQ achieves the same diffusion limit as JSQ in this regime via stochastic coupling and martingales techniques.

Methodology

- First, scaling time and number, express the process in some form. e.g., integral equations.
- Second, show the integral form has nice property, e.g., uniqueness, continuity.
- Third, prove the weak convergence of some parts in the integral form.
- Finally, apply Continuous Mapping Theorem to derive the weak convergence of the scaled process.

Thank you!
Q & A

References I

[1] A. Eryilmaz and R. Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72(3-4):311–359, 2012.

[2] P. Eschenfeldt and D. Gamarnik. Join the shortest queue with many servers. the heavy traffic asymptotics. *arXiv preprint arXiv:1502.00999*, 2015.

[3] P. Eschenfeldt and D. Gamarnik. Supermarket queueing system in the heavy traffic regime. short queue dynamics. *arXiv preprint arXiv:1610.03522*, 2016.

[4] G. Foschini and J. Salz. A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications*, 26(3):320–327, 1978.

[5] D. Gamarnik, J. N. Tsitsiklis, and M. Zubeldia. Delay, memory, and messaging tradeoffs in distributed service systems. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 1–12. ACM, 2016.

[6] Z. Liu, P. Nain, and D. Towsley. Sample path methods in the control of queues. *Queueing Systems*, 21(3-4):293–335, 1995.

[7] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg. Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation*, 68(11):1056–1071, 2011.

[8] S. T. Maguluri, R. Srikant, and L. Ying. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Performance Evaluation*, 81:20–39, 2014.

[9] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

References II

[10] D. Mukherjee, S. Borst, J. van Leeuwaarden, and P. Whiting. Universality of load balancing schemes on diffusion scale. *arXiv preprint arXiv:1510.02657*, 2015.

[11] D. Mukherjee, S. Borst, J. van Leeuwaarden, and P. Whiting. Universality of power-of-d load balancing schemes. *ACM SIGMETRICS Performance Evaluation Review*, 44(2):36–38, 2016.

[12] M. I. Reiman. Some diffusion approximations with state space collapse. In *Modelling and performance evaluation methodology*, pages 207–240. Springer, 1984.

[13] A. L. Stolyar. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems*, 80(4):341–361, 2015.

[14] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii*, 32(1):20–34, 1996.

[15] R. R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, pages 406–413, 1978.

[16] W. Whitt. Deciding which queue to join: Some counterexamples. *Operations research*, 34(1):55–62, 1986.

[17] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, pages 181–189, 1977.

[18] L. Ying, R. Srikant, and X. Kang. The power of slightly more than one sample in randomized load balancing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1131–1139. IEEE, 2015.