On the Performance Evaluation of Distributed Join-Idle-Queue Load Balancing

Benny Van Houdt
Department of Computer Science
University of Antwerp, Belgium

***Abstract*—Distributed Join-Idle-Queue load balancing is known to achieve vanishing waiting times in the large-scale limit provided that the number of dispatchers remains fixed, while the number of servers tends to infinity. When the number of dispatchers m scales to infinity together with the number of servers n, such that $r = n/m$ remains fixed, the large-scale performance of Join-Idle-Queue load balancing is less clear as waiting times no longer vanish.**

In this paper we first discuss some existing mean field models for distributed Join-Idle-Queue load balancing with $r = n/m$ fixed and explain why the well-known model introduced in [1] is not exact in the large-scale limit. The inexactness is caused by mixing two variants of distributed Join-Idle-Queue load balancing: a variant with and one without token withdrawals. Next we introduce mean field models for Join-Idle-Queue load balancing with token withdrawals, where an idle server places a token at a dispatcher with the shortest among d randomly chosen dispatchers.

The introduced mean field models imply that in case of phase type distributed service times and a total job arrival rate of $\lambda n < n$, the response time of a job corresponds to that in a standard M/PH/1 queue with load λq_0. The value of q_0 can be determined numerically and depends on λ, r and d, but not on the job size distribution (apart from its mean). This simple behavior is due to the token withdrawals and is lost if such withdrawals do not take place. We present simulation experiments that suggest that the unique fixed point of the introduced mean field models provides exact results in the large-scale limit.

***Index Terms*—Load balancing; join-idle-queue; mean field models; distributed computing**

I. Introduction

In traditional server farms jobs are distributed among the front end servers by a single hardware load balancer/dispatcher. While such a load balancer can support hundreds of servers, it is expensive, needs to be reconfigured when some of the servers are turned off during periods with low utilization and is not as robust as a distributed system. For this reason the use of multiple software based load balancers is preferential in a Cloud environment. While traditional load balancers often made use of the join-the-shortest-queue (JSQ) algorithm, as all the requests and responses tended to flow through the load balancer, a new class of distributed load balancers called Join-Idle-Queue (JIQ) for systems with multiple load balancers was introduced in [1]. Throughout the paper we use the terms load balancer and dispatcher interchangeably.

Distributed JIQ load balancing operates as follows: each dispatcher maintains an I-queue that contains a list of server identities. These are servers that reported that they became idle some time ago. We refer to these server identities as tokens. When a new job arrives at a dispatcher, it is immediately assigned to a server in the following manner:

- If the I-queue of the dispatcher is not empty, the job is assigned to a server, the identity of which is selected from the list in its I-queue. In such case the identity/token of the selected server is removed from the list.
- If the I-queue of the dispatcher is empty, a random server is selected.

From the server side we have that whenever a server becomes idle, it adds its identity to the I-queue of a dispatcher. Two algorithms are considered in [1]:

- JIQ-Random, meaning the dispatcher is selected at random,
- JIQ-SQ(d), meaning the server selects d dispatchers at random and adds its identity to a dispatcher with the shortest I-queue among the d selected dispatchers. For $d = 1$ this scheme coincides with JIQ-Random.

Notice that a server may hold one or several jobs even if it is listed in an I-queue of a dispatcher as other dispatchers may assign jobs to a server when their I-queue is empty. This can be avoided by demanding that an idle server *withdraws* its identity from the I-queue of the dispatcher as soon as a job is assigned by another dispatcher. As such we distinguish between the JIQ load balancing algorithm **with and without token withdrawals**. Without token withdrawals, one can make a further distinction on whether or not a server is allowed to add its token to more than one dispatcher as the server could become idle again before its outstanding token is used.

The performance of JIQ load balancing has been studied in the large-scale limit by various authors. When the number of dispatchers m remains fixed, while the number of servers n tends to infinity, JIQ is known to have vanishing waiting times in many settings [2], [3], [4] under subcritical load per server, that is, if λn denotes the total arrival rate and the mean job size equals 1, then all jobs are assigned to idle servers in the limit when $\lambda < 1$.

As any load balancer can only support a finite number of servers in any real system, it might be more appropriate to look at the limit if both the number of dispatchers m and the number of servers n tend to infinity, such that $r = n/m$ remains fixed. This limit was initially considered in [1] for JIQ-Random and JIQ-SQ(d) under two assumptions that only hold for JIQ *with* token withdrawals. However, as we explain in Section II, the limit presented in [1] is inexact as the authors mix properties of JIQ with and without withdrawals when deriving their result.

A mean field model for JIQ-Random and JIQ-Pod *with* token withdrawals and exponential job sizes is presented in [5]. JIQ-Pod operates as JIQ-Random, except that the power-of-d-choices paradigm [6], [7] is used by the dispatcher when a job is assigned to a dispatcher with an empty I-queue. While JIQ-Pod improves the performance of JIQ-Random under high loads, the downside is that some jobs are not assigned immediately. While no convergence proofs are presented in [5], simulation results suggest that the unique fixed point of the mean field model corresponds to the exact limit. The authors in [5] also illustrate using simulation that their model for JIQ-Random is more accurate than the model in [1] for finite n, but no explanation is provided.

A number of variants of JIQ, including JIQ-SQ(d), *without* token withdrawals and exponential job sizes are analyzed using mean field models in [8] for the case where servers to not add their token to more than one dispatcher. These models are significantly more complicated than the ones with token withdrawals and even proving the existence of a unique fixed point appears to be problematic, which is needed before one can even start thinking about convergence proofs. The author does present simulation results that suggest that the models provide exact results in the large-scale limit. The main insights of the models in [8] are that the system behavior of JIQ-SQ(d) without token withdrawals is quite complex as the servers experience queue length dependent arrival rates and the order in which dispatchers select tokens from their I-queue impacts performance.

In this paper we make the following contributions:

1) We explain why the large-scale analysis presented in [1] is inexact. On the upside we show that the inaccuracy of the proposed limit for JIQ-SQ(d) is small and decreases rapidly as d increases.
2) We introduce a novel mean field model for JIQ-SQ(d) with token withdrawals. We first consider exponential job sizes and then generalize to phase-type distributed job sizes. Both models are validated by simulation in Section V. Our results for JIQ-SQ(d) for exponential job sizes with $d = 1$ coincide with the results presented in [5] for JIQ-Random.

It should be relatively easy to prove that our mean field models become exact over finite time scales as the number of servers tends to infinity as our models fall within the framework of density dependent population processes of Kurtz [9].

Under the assumption (supported by simulation in Section V) that the mean field models presented in this paper are asymptotically exact, the following insights are obtained for JIQ-SQ(d):

1) As n tends to infinity with $r = n/m$ fixed, the response time distribution of a job becomes identical to that in an M/PH/1 queue with load λq_0.
2) The value of q_0 depends on λ, d and r, but is independent of the job size distribution (with mean 1).

n	q_0	$E[R]$
50	0.3738	1.4878
500	0.3757	1.4708
5000	0.3752	1.4687
model in [1]	0.4000	1.5152

TABLE I
SIMULATION RESULTS FOR JIQ-RANDOM WITH TOKEN WITHDRAWALS AND EXPONENTIAL JOB SIZES, $\lambda = 0.85$ AND $r = n/m = 10$ FOR $n = 50, 500$ AND 5000 SERVERS. $E[R]$ IS THE MEAN RESPONSE TIME AND q_0 THE PROBABILITY THAT A DISPATCHER HOLDS ZERO TOKENS. THERE IS NO CONVERGENCE TO THE MODEL AS n TENDS TO INFINITY.

3) The method used by the dispatcher to select a token from its I-queue when an arrival occurs, has no impact on the performance.

We end this introduction with a short discussion of some other JIQ related work. In [10] it was shown that JIQ is not heavy traffic optimal, which is not surprising due to the random assignments used when a job arrives at a dispatcher with an empty I-queue. The authors therefore propose and study Join-Below-Threshold load balancing which is in the same spirit as JIQ-Threshold [8]. Load balancers for homogeneous and heterogeneous systems using outdated queue length information were considered both in the case of a single [11] or multiple dispatchers [12], [13]. For heterogeneous systems that use JIQ load balancing with a fixed number of dispatchers, vanishing waiting times were achieved in the limit by exchanging tokens or using non-uniform token allotment in [14]. Finally, JIQ was also studied in a setting with service elasticity in [15] and [16].

The paper starts with a discussion of the large-scale limit analysis in [1] in Section II. Sections III to V are devoted to the JIQ-SQ(d) models and their validation. The paper ends with some conclusions in Section VI.

II. INEXACTNESS OF AN EXISTING LARGE-SCALE ANALYSIS

In this section we first demonstrate that the model in [1] for the large-scale system is inexact and identify the reason for this inexactness. In Table I we present an arbitrary simulation experiment for JIQ-Random with an increasing number of servers n and compare these results with the model in [1]. This model depends on two assumptions: (1) there is exactly one copy of each idle server in the I-queues and (2) there are only idle servers in the I-queues. As these assumptions are valid for JIQ-Random with token withdrawals, we simulated the system with token withdrawals. The fact that the model in [1] is not asymptotically exact for the system without token withdrawals was already demonstrated in [8, see Table I], which is not surprising as assumption (2) does not hold without withdrawals. Table I strongly suggests that the model in [1] is not asymptotically exact even with token withdrawals. Nevertheless, we will demonstrate that it may still be regarded as a good to excellent approximation.

To understand the cause of the inexactness, we look at Theorem 1 in [1]. Denote $\hat{q}_1$ as the probability that an I-queue contains at least one token in equilibrium when the number of

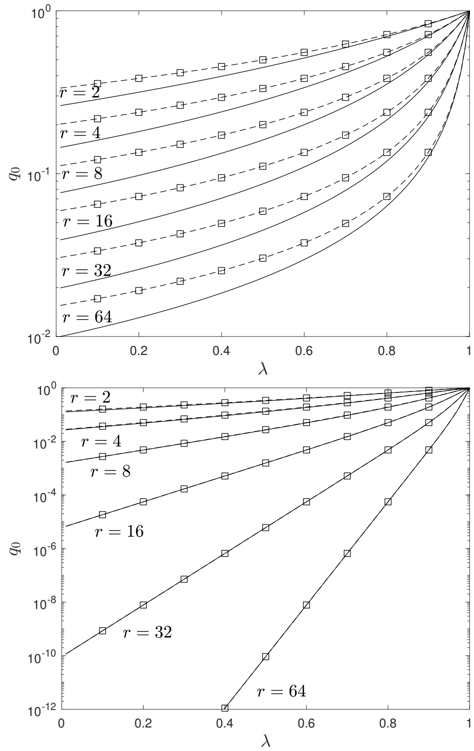

Fig. 1. Impact of λ and r on q_0 for $d = 1$ (top) and $d = 2$ (bottom). Full lines are obtained using Algorithm 1, dashed lines are for the model in [1].

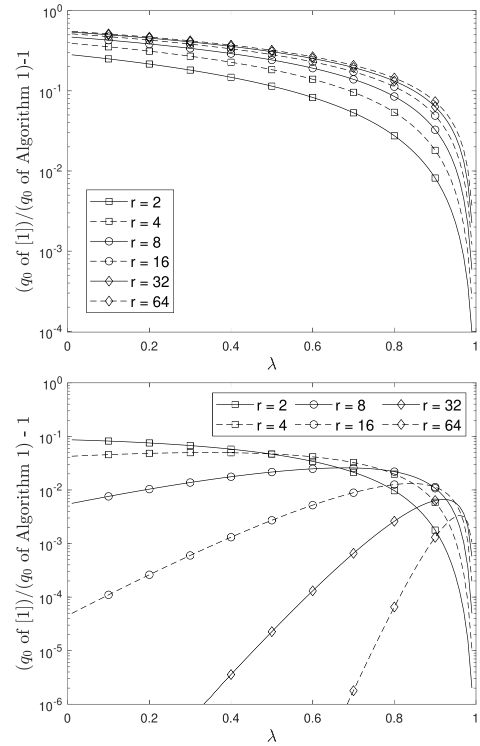

Fig. 2. Relative deviation on q_0 of model in [1] compared to Algorithm 1 as a function of λ and r for $d = 1$ (top) and $d = 2$ (bottom.

servers n tends to infinity. Theorem 1 in [1] then states that for JIQ-Random, we have

$$\frac{\hat{q}_1}{1-\hat{q}_1} = (1-\lambda)r,$$

and for JIQ-SQ(d), we have

$$\sum_{i\geq 1} \hat{q}_1^{(d^i-1)/(d-1)} = (1-\lambda)r.$$

We explain below that the left-hand sides are the mean I-queue lengths for JIQ-Random and JIQ-SQ(d) *without* token withdrawals, while the right-hand side is the mean queue length of an I-queue in a system *with* token withdrawals. Hence Theorem 1 mixes two different JIQ systems and the analysis in [1] is therefore asymptotically inexact for both systems.

The left-hand sides of the above equations can be understood by looking at the mean field models in [8]. More specifically, looking at the ODEs in [8, Section IV.C] implies that the distribution of the number of tokens at an I-queue for JIQ-Random is geometric with parameter $\hat{q}_1 = s_1/\lambda$, while for JIQ-SQ(d) the probability that an I-queue contains at least i tokens equals $\hat{q}_1^{(d^i-1)/(d-1)} = (s_1/\lambda)^{(d^i-1)/(d-1)}$. Therefore the average number of tokens in an I-queue is indeed given by $\frac{\hat{q}_1}{1-\hat{q}_1}$ for JIQ-Random and $\sum_{i\geq 1} \hat{q}_1^{(d^i-1)/(d-1)}$ for JIQ-SQ(d). The analysis in [8] is for the case *without* token withdrawals. Thus the expressions on the left-hand side in Theorem 1 in [1] are the mean I-queue lengths for the system *without* withdrawals. In such case the total number of tokens residing in the I-queues *does not* match the number of idle servers.

However the right-hand side equals $(1-\lambda)r$ for both equations in Theorem 1 in [1] and this is the mean queue length of an I-queue in a system *with* token withdrawals. Indeed, when servers withdraw their token, the total number of tokens residing in the I-queues perfectly matches the number of idle servers. As $(1-\lambda)$ should be the limiting fraction of idle servers and there are r times as many servers as dispatchers, the I-queue of a dispatcher contains on average $(1-\lambda)r$ tokens.

In Figure 1 we compare the value of q_0 computed by the model in [1] with the result of Algorithm 1 presented in Section III. Simulation experiments presented in Section V suggest that Algorithm 1 yields asymptotically exact results for JIQ-SQ(d) with token withdrawals. Figure 1 confirms the inexactness for varying r and λ values. It also clearly shows that the accuracy improves as d increases from 1 to 2. In fact, additional experiments (not presented) showed that the accuracy improved even further when considering larger d values. Figure 2 further illustrates the relative deviation between the model in [1] and Algorithm 1. It is fair to state that while the model in [1] is not asymptotically exact, its accuracy is outstanding for $d > 1$ whenever λ is not too close to one and r is sufficiently large.

III. Mean Field Model for JIQ-SQ(d) with exponential job sizes

In this section we present a new mean field model for JIQ-SQ(d) with servers that withdraw their token when a job is assigned by another dispatcher. We consider a system with m dispatchers, that each have an I-queue to hold tokens, and n servers. As before let $r = n/m$. Poisson arrivals occur at rate λn and are spread uniformly over the m dispatchers. For now the job size is exponentially distributed with mean 1.

Let $Q_i(t)$ be the number of I-queues with exactly i tokens at time t, $S_i(t)$ the number of servers with i jobs at time t. Hence, $\sum_i Q_i(t) = m$ and $\sum_i S_i(t) = n$. Let $\hat{Q}_k(t) = \sum_{i\geq k} Q_i(t)$ be the number of I-queues holding k or more tokens. Define the fractions $q_i(t) = Q_i(t)/m, \hat{q}_i(t) = \hat{Q}_i(t)/m$ and $s_i(t) = S_i(t)/n$. Let $\Delta\hat{Q}_i(t)$ denote the expected change in $\hat{Q}_i(t)$ over a small interval dt, that is, $\Delta\hat{Q}_i(t) = E[\hat{Q}_i(t+dt) - \hat{Q}_i(t)]$. As explained below, we have for $i > 0$

$$\begin{aligned}\Delta\hat{Q}_i(t) &= -(\lambda n)dt\left(\frac{\hat{Q}_i(t)}{m} - \frac{\hat{Q}_{i+1}(t))}{m}\right)\\ &+ S_1(t)dt\left(\frac{\hat{Q}_{i-1}(t)^d}{m^d} - \frac{\hat{Q}_i(t)^d}{m^d}\right)\\ &- (\lambda n)dtq_0(t)i\left(\frac{\hat{Q}_i(t)}{n} - \frac{\hat{Q}_{i+1}(t)}{n}\right).\end{aligned}$$

The first term is due to arrivals were $(\hat{Q}_i(t) - \hat{Q}_{i+1}(t))/m$ is the probability that a random arrival occurs at a dispatcher with exactly i tokens in its I-queue. The second term is due to service completions as jobs are assumed to be exponential in size with mean one and a server that becomes idle uses the power-of-d-choices rule to select a dispatcher. The last term is due to servers withdrawing tokens and can be understood as follows. First note that $i(\hat{Q}_i(t) - \hat{Q}_{i+1}(t))$ is the total number of tokens residing at the I-queues that hold exactly i tokens. Therefore $i(\hat{Q}_i(t) - \hat{Q}_{i+1}(t))/n$ is the probability that an arrival at an empty I-queue is assigned to a server that has a token at an I-queue with length i.

Dividing left and right by mdt yields

$$\begin{aligned}\frac{d\hat{q}_i(t)}{dt} &= -\lambda r(\hat{q}_i(t) - \hat{q}_{i+1}(t)) + s_1(t)r(\hat{q}_{i-1}(t)^d -\\ &\hat{q}_i(t)^d) - \lambda(1-\hat{q}_1(t))i(\hat{q}_i(t) - \hat{q}_{i+1}(t)),\end{aligned} \tag{1}$$

where $\hat{q}_0(t) = 1$ for all t.

We now proceed with the servers for $i > 0$:

$$\begin{aligned}\Delta S_i(t) &= (\lambda n)dtq_0(t)\left(\frac{S_{i-1}(t)}{n} - \frac{S_i(t))}{n}\right)\\ &- dt\,(S_i(t) - S_{i+1}(t)) + 1[i=1](\lambda n)(1-q_0(t))dt.\end{aligned}$$

The first term corresponds to arrivals in empty I-queues, the second to service completions and the last term to arrivals in a non-empty I-queue. Dividing by ndt yields

$$\begin{aligned}\frac{ds_i(t)}{dt} &= \lambda q_0(t)(s_{i-1}(t) - s_i(t)) - (s_i(t) - s_{i+1}(t))\\ &+ 1[i=1]\lambda(1-q_0(t)).\end{aligned} \tag{2}$$

Similarly, we find

$$\frac{ds_0(t)}{dt} = -\lambda q_0(t)s_0(t) + s_1(t) - \lambda(1-q_0(t)). \tag{3}$$

Notice that all the above equations apply irrespective of the manner in which a dispatcher selects a token from its I-queue, which is in contrast to JIQ-SQ(d) without token withdrawals [8].

Theorem 1. *Let* $(s,\hat{q})$ *be a fixed point of* (1)-(3) *such that* $\sum_{i\geq 0} s_i = 1$, $\sum_{i\geq 0} is_i < \infty$, $\hat{q}_0 = 1$ *and* $\sum_{i\geq 0}\hat{q}_i < \infty$, *then* $\lambda = \sum_{i\geq 1} s_i$,

$$s_1 = \lambda(1-\lambda q_0), \tag{4}$$

$$s_k = s_1(\lambda q_0)^{k-1}, \tag{5}$$

for $k > 1$, *with* $q_0 = 1 - \hat{q}_1$. *Further,*

$$\sum_{i\geq 1}\hat{q}_i = (1-\lambda)r. \tag{6}$$

Proof. The equality $\lambda = \sum_{i\geq 1} s_i$ follows from (2) as

$$0 = \sum_{i\geq 1} i\frac{ds_i(t)}{dt} = \lambda q_0(t)\sum_{i\geq 0} s_i(t) - \sum_{i\geq 1} s_i(t) + \lambda(1-q_0(t)).$$

Using (2) and the fact that $\sum_{i\geq k}\frac{ds_i(t)}{dt} = 0$ yields

$$s_k = s_{k-1}\lambda q_0,$$

for $k \geq 2$, which implies (5). The expression for s_1 in (4) is now immediate by combining $\lambda = \sum_{i\geq 1} s_i$ with (5). To prove (6) we note that (1) implies

$$0 = \sum_{i\geq 1}\frac{d\hat{q}_i(t)}{dt} = -\hat{q}_1(t)\lambda r + s_1(t)r - \lambda(1-\hat{q}_1(t))\sum_{i\geq 1}\hat{q}_i(t).$$

When combined with (4), we find that $\sum_{i\geq 1}\hat{q}_i = (1-\lambda)r$. □

Remarks: 1) Looking at the expression for s_k in (5), we see that the queue length distribution is identical to an M/M/1 queue with arrival rate λq_0 when the server is busy and with an increased arrival rate when the queue is idle (such that the probability that the queue is idle is $1-\lambda$ instead of $1-\lambda q_0$). As increasing the arrival rate in an idle queue does not impact the response time distribution, jobs have the same response time distribution as in an ordinary M/M/1 queue with arrival rate λq_0. Therefore the response time is exponential with parameter $1-\lambda q_0$ (as the service rate equals 1). We indicate how to compute q_0 further on.

2) The equality $\lambda = \sum_{i\geq 1} s_i$ is natural as λ should be the probability that a server is busy. The equality in (6) is also expected as every idle server has exactly one token at one of the dispatchers and the fraction of idle servers is $(1-\lambda)$ while there are r times as many servers as dispatchers. Therefore the mean number of tokens per dispatcher should be $(1-\lambda)r$.

Theorem 2. *Let* $(s,\hat{q})$ *be a fixed point of* (1)-(3) *such that* $\sum_{i\geq 0} s_i = 1$, $\sum_{i\geq 0} is_i < \infty$, $\hat{q}_0 = 1$ *and* $\sum_{i\geq 0}\hat{q}_i < \infty$, *then*

$$\hat{q}_{k+1} = \hat{q}_k - (\hat{q}_{k-1}^d - \hat{q}_k^d)\frac{1-\lambda q_0}{1+\frac{kq_0}{r}}, \tag{7}$$

for $k \geq 1$ with $q_0 = 1 - \hat{q}_1$.

Proof. For any fixed point we have $\sum_{i \geq k} \frac{d\hat{q}_i(t)}{dt} = 0$, which implies that

$$0 = -\hat{q}_k \lambda r + s_1 r \hat{q}_{k-1}^d - \lambda(1-\hat{q}_1)\left(k\hat{q}_k + \sum_{s>k} \hat{q}_s\right),$$

for $k \geq 1$. We can rewrite this as

$$\hat{q}_k = \frac{s_1 r \hat{q}_{k-1}^d - \lambda q_0 \left((1-\lambda)r - \sum_{s=1}^{k-1} \hat{q}_s\right)}{\lambda r + \lambda q_0 (k-1)}, \tag{8}$$

due to (6). Although this expression can be used to compute $\hat{q}_k$, we derive a more elegant recursion that is equivalent. We first note that by (8)

$$\hat{q}_k(\lambda r + \lambda q_0 k) = \left(s_1 r \hat{q}_{k-1}^d - \lambda q_0 \left((1-\lambda)r - \sum_{s=1}^{k-1} \hat{q}_s\right)\right) \cdot \left(1 + \frac{q_0}{r + q_0(k-1)}\right),$$

and

$$\hat{q}_{k+1}(\lambda r + \lambda q_0 k) = s_1 r \hat{q}_k^d - \lambda q_0 \left((1-\lambda)r - \sum_{s=1}^{k} \hat{q}_s\right).$$

Hence,

$$\begin{aligned}(\hat{q}_k - \hat{q}_{k+1})(\lambda r + \lambda q_0 k) &= s_1 r(\hat{q}_{k-1}^d - \hat{q}_k^d) - \lambda q_0 \hat{q}_k + \\ &\left(s_1 r \hat{q}_{k-1}^d - \lambda q_0 \left((1-\lambda)r - \sum_{s=1}^{k-1} \hat{q}_s\right)\right) \frac{q_0}{r + q_0(k-1)} \\ &= s_1 r(\hat{q}_{k-1}^d - \hat{q}_k^d) - \lambda q_0 \hat{q}_k + \lambda q_0 \hat{q}_k.\end{aligned}$$

When combined with (4) this proves (7). $\square$

Remarks: 1) When $d = 1$, (7) implies that

$$q_k = q_0 \prod_{\ell=1}^{k} \frac{1 - \lambda q_0}{1 + \frac{\ell q_0}{r}},$$

where $q_k = \hat{q}_k - \hat{q}_{k+1}$. Further, as $\sum_{i \geq 0} q_i = \hat{q}_0 = 1$ this shows that q_0 is a solution of

$$q_0 \sum_{k \geq 0} \prod_{\ell=1}^{k} \frac{1 - \lambda q_0}{1 + \frac{\ell q_0}{r}} = 1,$$

which coincides with Theorem 2 in [5], where it is shown that q_0 is the unique solution of this equation on $(0, 1)$.
2) Given q_0 we can compute $\hat{q}_k$ for $k \geq 2$ using (7) (or (8)) as $\hat{q}_1 = 1 - q_0$.

Theorem 3. *There exists a unique fixed point $(s, \hat{q})$ of (1)-(3) such that $\sum_{i \geq 0} s_i = 1$, $\sum_{i \geq 0} i s_i < \infty$, $\hat{q}_0 = 1$ and $\sum_{i \geq 0} \hat{q}_i < \infty$.*

Proof. Given Theorems 1 and 2, it suffices to show that there is a unique $\hat{q}_1 \in (0, 1)$ such that (6) holds, that is, such that $\sum_{i \geq 1} \hat{q}_i = (1-\lambda)r$. It is clear from (7) that $\hat{q}_k$ is a continuous function of $\hat{q}_1$ (but not necessarily monotone, see Figure 3(left) for an example) and when $\hat{q}_1 = 1$, then $\hat{q}_k = 1$ for any k.

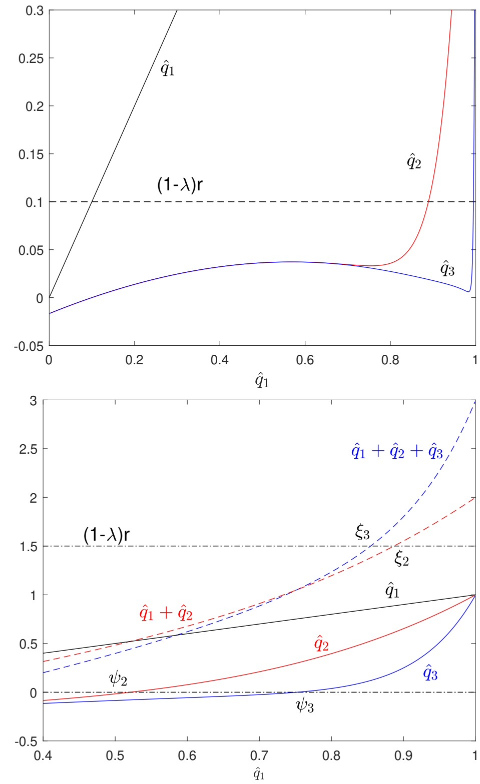

Fig. 3. Illustration of non-monotone behavior for $\lambda = 0.98$, $r = 5$ and $d = 20$ (top), ξ_k and ψ_k for $\lambda = 0.7$, $r = 5$, and $d = 3$ (bottom).

Therefore for any $k > (1-\lambda)r$ we have for $\hat{q}_1$ sufficiently close to 1 that $\sum_{s=1}^{k} \hat{q}_s > (1-\lambda)r$. Further $\hat{q}_{k+1} \leq \hat{q}_k$, meaning $\hat{q}_k \leq 0$ for any k when $\hat{q}_1 = 0$. Define $\xi_k = 1$ for $k = 1, \ldots, \lfloor (1-\lambda)r \rfloor$ as the smallest value of $\hat{q}_1 \in (0, 1)$ such that $\sum_{s=1}^{k} \hat{q}_s = (1-\lambda)r$ for $k > \lfloor (1-\lambda)r \rfloor$.

We now argue that the following two statements hold:

1) $\hat{q}_k > 0$ when $\sum_{s=1}^{k-1} \hat{q}_s \geq (1-\lambda)r$,
2) $\hat{q}_k$ has a positive derivative on $(0, \xi_{k-1}]$.

The first statement is immediate from (8) as both $s_1 r \hat{q}_{k-1}^d$ and $\lambda r + \lambda q_0(k-1)$ are positive for any $\hat{q}_1 \in (0, 1)$. Therefore $\sum_{s=1}^{k} \hat{q}_s > (1-\lambda)r$ when $\hat{q}_1$ equals ξ_{k-1}, which shows that $\xi_k \leq \xi_{k-1}$. In other words, $\xi_k \in (0, 1]$ is non-increasing in k as illustrated in Figure 3(right).

The second statement follows by induction on k as follows. The statement clearly holds for $k = 1$. For $k > 1$, we have by induction that that $\hat{q}_s$ is increasing on $(0, \xi_{s-1})$ for $s < k$, meaning $\sum_{s=1}^{k-1} \hat{q}_s$ is increasing on $(0, \xi_{k-1})$ as ξ_s is non-increasing in s. By definition of ξ_{k-1}, we also have that $\sum_{s=1}^{k-1} \hat{q}_s < (1-\lambda)r$ on $(0, \xi_{k-1})$. By (8) we now see that $\hat{q}_k$ is increasing on $(0, \xi_{k-1})$ as $s_1 r \hat{q}_{k-1}^d$ is increasing and positive on $(0, \xi_{k-1})$, $\lambda r + q_0(k-1)/r$ is decreasing and positive on $(0, \xi_{k-1})$ and $-\lambda q_0((1-\lambda)r - \sum_{s=1}^{k-1} \hat{q}_s)$ is negative and increasing on $(0, \xi_{k-1})$.

We proceed by using induction on k to argue that $\sum_{s=1}^{k} \hat{q}_s > (1-\lambda)r$ when $\hat{q}_1 \in (\xi_k, 1)$. This trivially holds for $k = 1$. For $k > 1$, we have by induction that $\sum_{s=1}^{k-1} \hat{q}_s$ exceeds $(1-\lambda)r$ on $(\xi_{k-1}, 1)$. By (8), we therefore have that $\hat{q}_k$ is positive on $(\xi_{k-1}, 1)$, so $\sum_{s=1}^{k} \hat{q}_s > \sum_{s=1}^{k-1} \hat{q}_s > (1-\lambda)r$. For $\hat{q}_1 \in (\xi_k, \xi_{k-1})$, we know that $\hat{q}_k$ is increasing and $\sum_{s=1}^{k} \hat{q}_s = (1-\lambda)r$ for $\hat{q}_1 = \xi_k$. This shows that for $k > \lfloor (1-\lambda)r \rfloor$, there is a unique solution $\hat{q}_1 = \xi_k$ on $(0,1)$ such that $\sum_{s=1}^{k-1} \hat{q}_s = (1-\lambda)r$.

The unique value of q_0 such that $\sum_{i\geq 1} \hat{q}_i = (1-\lambda)r$ is then found as $1 - \lim_{k\to\infty} \xi_k$ as $q_0 = 1 - \hat{q}_1$ and the limit of a decreasing sequence of values in $(0,1)$ exists. □

Remarks: 1) We see from the proof of Theorem 3 that for any $k > 1$ there exists a unique ψ_k such that $\hat{q}_k = 0$ if $\hat{q}_1 = \psi_k$. Further $\psi_k \geq \psi_{k-1}$ as $\hat{q}_k \leq \hat{q}_{k-1}$ (see Figure 3(right) for an illustration). This implies that if $\hat{q}_1 < \lim_{k\to\infty} \xi_k$, then $\lim_{k\to\infty} \hat{q}_k < 0$, while $\lim_{k\to\infty} \hat{q}_k > 0$ if $\hat{q}_1 > \lim_{k\to\infty} \xi_k$. Hence, $\lim_{k\to\infty} \xi_k = \lim_{k\to\infty} \psi_k$, where ξ_k is a decreasing sequence and ψ_k an increasing one.

2) Looking at the proof of Theorem 3 and the previous remark, we have the following simple algorithm to compute $q_0 = 1-\hat{q}_1$. We start with $k = \lceil (1-\lambda)r \rceil$ and determine ξ_k and ψ_k by performing a bisection algorithm on $(0,1)$ using (7). Next we repeatedly increase k by one and determine ξ_k and ψ_k using a bisection algorithm on (ψ_{k-1}, ξ_{k-1}) until $\xi_k - \psi_k < 10^{-15}$ (see Algorithm 1). Once q_0 is found, we can compute the queue length distributions s and $\hat{q}$ using (4), (5) and (7). As noted before, the response time distribution of a job is exponential with parameter $1 - \lambda q_0$.

Input: λ, d, r
Output: q_0
$k := \lceil (1-\lambda)r \rceil$;
Compute ξ_k using bisection on $(0,1)$;
Compute ψ_k using bisection on $(0, \xi_k)$;
while $\xi_k - \psi_k > 10^{-15}$ **do**
  $k := k+1$;
  Compute ξ_k using bisection on (ψ_{k-1}, ξ_{k-1});
  Compute ψ_k using bisection on (ψ_{k-1}, ξ_k);
end
$q_0 := 1 - \xi_k$;

Algorithm 1: Computation of q_0 for JIQ-SQ(d).

IV. Mean Field Model for JIQ-SQ(d) with Phase-Type Job Sizes

We now generalize the previous mean field model to phase-type job sizes characterized by a $1 \times b$ vector α and a $b \times b$ matrix T such that $P[X > t] = \alpha \exp(Tt) e$, where X is the job size and e a vector of ones. For further use denote $t^* = (-T)e$. Phase-type (PH) distributions are distributions with a modulating finite state Markov chain (see also [17]). Any general positive-valued distribution can be approximated arbitrarily close with a PH distribution and there are various fitting tools available for PH distributions (see e.g. [18]). The main objective of this section is to show the following two properties:

1) The distribution of the number of tokens in an I-queue (given by $\hat{q}$) does not depend on the job size distribution. Hence, we have insensitivity in the mean field model for the manner in which the tokens are distributed over the dispatchers. This also implies that the probability q_0 that an I-queue is empty is insensitive to the job size distribution.

2) The response time distribution of a job under JIQ-SQ(d) with phase-type job size distribution X is identical to that of a job arriving in an M/G/1 queue with arrival rate λq_0 and job size distribution X. Note that this was already shown for exponential job sizes in the previous section.

Let $S_{i,j}(t)$ be the number of servers in service phase j with i jobs at time t and $S_0(t)$ be the number of idle servers at time t. Denote $\vec{S}_i(t) = (S_{i,1}(t), \ldots, S_{i,b}(t))$, $s_{i,j}(t) = S_{i,j}(t)/n$ and $\vec{s}_i(t) = \vec{S}_i(t)/n$. If we look at the evolution of the number of tokens in an I-queue we see that it is identical to the case of exponential job sizes, except for the term that corresponds to the service completions. In the exponential case this term was given by

$$S_1(t)dt\left(\frac{\hat{Q}_{i-1}(t)^d}{m^d} - \frac{\hat{Q}_i(t)^d}{m^d}\right),$$

while in case of phase-type job sizes this becomes

$$\left(\sum_j S_{1,j}(t)t_j^*\right) dt \left(\frac{\hat{Q}_{i-1}(t)^d}{m^d} - \frac{\hat{Q}_i(t)^d}{m^d}\right),$$

where the first sum can be written in matrix notation as $\vec{s}_1(t)t^*$. This implies that (1) becomes

$$\begin{aligned}\frac{d\hat{q}_i(t)}{dt} = &-\lambda r(\hat{q}_i(t) - \hat{q}_{i+1}(t)) + (\vec{s}_1(t)t^*)r(\hat{q}_{i-1}(t)^d - \hat{q}_i(t)^d) \\ &- \lambda(1-\hat{q}_1(t))i(\hat{q}_i(t) - \hat{q}_{i+1}(t)), \qquad (9)\end{aligned}$$

which is identical to (1) if we replace $s_1(t)$ by $\vec{s}_1(t)t^*$.

We now proceed with the expected change in $S_{i,j}(t)$:

$$\begin{aligned}\Delta S_{i,j}(t) = &(\lambda n)dtq_0(t)\left(1[i>1]\frac{S_{i-1,j}(t)}{n} + 1[i=1]\frac{S_0(t)}{n}\alpha_j\right. \\ &\left.-\frac{S_{i,j}(t))}{n}\right) - dtS_{i,j}(t)t_j^* + dt\sum_k S_{i+1,k}(t)t_k^*\alpha_j \\ &- dtS_{i,j}(t)\sum_{k\neq j} T_{j,k} + dt\sum_{k\neq j} S_{i,k}(t)T_{k,j} \\ &+ 1[i=1](\lambda n)(1-q_0(t))\alpha_j dt.\end{aligned}$$

The first and last term are very similar to (2), the other terms correspond to service completions and phase changes. As $t^* = -Te$, this can be written in matrix form as

$$\begin{aligned}\Delta \vec{S}_i(t) = &(\lambda n)dtq_0(t)\left(1[i>1]\frac{S_{i-1,j}(t)}{n} + 1[i=1]\frac{S_0(t)}{n}\alpha\right. \\ &\left.-\frac{\vec{S}_i(t))}{n}\right) - dt\vec{S}_i(t)diag(t^*) + dt\vec{S}_{i+1}(t)t^*\alpha \\ &+ dt\vec{S}_i(t)(diag(t^*) + diag(T)) + dt(\vec{S}_i(t)T \\ &- \vec{S}_i(t)diag(T)) + 1[i=1](\lambda n)(1-q_0(t))\alpha dt,\end{aligned}$$

where $diag(v)$ of a vector v is a diagonal matrix with v on its main diagonal and $diag(A)$ of a matrix A is the diagonal matrix obtained by setting all off diagonal entries of A to zero. After simplifying this implies that

$$\frac{d\vec{s}_i(t)}{dt} = \lambda q_0(t)(1[i>1]\vec{s}_{i-1}(t) + 1[i=1]s_0(t)\alpha - \vec{s}_i(t)) + \vec{s}_{i+1}(t)t^*\alpha + \vec{s}_i(t)T + 1[i=1]\lambda(1-q_0(t))\alpha. \quad (10)$$

Similarly, we find

$$\frac{ds_0(t)}{dt} = -\lambda q_0(t)s_0(t) + \vec{s}_1(t)t^* - \lambda(1-q_0(t)). \quad (11)$$

Theorem 4. *Let $(s,\hat{q})$ be a fixed point of (9-11) such that $\sum_{i\geq 0}\vec{s}_i e = 1$, $\sum_{i\geq 0} i\vec{s}_i e < \infty$, $\hat{q}_0 = 1$ and $\sum_{i\geq 1}\hat{q}_i < \infty$ then $\lambda = \sum_{i\geq 1}\vec{s}_i e$,*

$$\vec{s}_1 = (1-\lambda q_0)\alpha R/q_0, \quad (12)$$
$$\vec{s}_k = \vec{s}_1 R^{k-1}, \quad (13)$$

for $k>1$, with $q_0 = 1-\hat{q}_1$ and $R = \lambda q_0(\lambda q_0 I - \lambda q_0 e\alpha - T)^{-1}$. Further, $\vec{s}_1 t^* = \lambda(1-\lambda q_0)$.

Proof. By demanding that $\sum_{i\geq 1}\frac{d\vec{s}_i(t)}{dt} + s_0(t)\alpha = 0$ and using the definition of t^*, one finds that

$$\sum_{i\geq 1}\vec{s}_i(T+t^*\alpha) = 0.$$

Hence, $\sum_{i\geq 1}\vec{s}_i$ is a multiple of the unique invariant vector β for which $\beta(T+t^*\alpha)=0$ and $\beta e = 1$ holds. By considering the equality $\sum_{i\geq 1} i\frac{d\vec{s}_i(t)}{dt}e = 0$, one also finds that

$$\sum_{i\geq 1}\vec{s}_i t^* = \lambda.$$

As $\beta t^* = 1$, this allows us to conclude that $\sum_{i\geq 1}\vec{s}_i = \lambda\beta$ and thus that $s_0 = 1-\lambda$. We may therefore write

$$\lambda q_0 s_0 + \lambda(1-q_0) = \lambda\frac{1-\lambda q_0}{1-\lambda}s_0,$$

which means that the fixed point equations associated with (10) and (11) can be stated as

$$0 = 1[i>1]\lambda q_0\vec{s}_{i-1} - \lambda q_0\vec{s}_i + 1[i=1]\lambda\frac{1-\lambda q_0}{1-\lambda}s_0\alpha + \vec{s}_{i+1}t^*\alpha + \vec{s}_i T,$$

and

$$0 = -\lambda\frac{1-\lambda q_0}{1-\lambda}s_0 + \vec{s}_1 t^*.$$

These fixed point equations are identical to the balance equations of an M/PH/1 queue with arrival rate $\lambda_0 = \lambda(1-\lambda q_0)/(1-\lambda)$ when the queue is empty and arrival rate λq_0 otherwise. Therefore (12) and (13) hold due to Theorem 2 in [19]. To verify that $\vec{s}_1 t^* = \lambda(1-\lambda q_0)$, we note that

$$(\lambda q_0 I - \lambda q_0 e\alpha - T)e = t^*,$$

which shows that $Rt^* = \lambda q_0 e$ and therefore by (12) we find $\vec{s}_1 t^* = \lambda(1-\lambda q_0)$ as required. □

Theorem 5. *There exists a unique fixed point $(\vec{s},\hat{q})$ of (9)-(11) such that $\sum_{i\geq 0}s_i = 1$, $\sum_{i\geq 0} is_i < \infty$, $\hat{q}_0 = 1$ and $\sum_{i\geq 0}\hat{q}_i < \infty$. Further, $\hat{q}$ is insensitive to the phase-type distribution (α,T).*

Proof. By Theorem 4 we have that $\vec{s}_1 t = \lambda(1-q_0\lambda)$, therefore the fixed point equations associated with (1) and (9) are identical and have a unique solution $\hat{q}$ due to Theorem 3. As $q_0 = 1-\hat{q}_1$, the $\vec{s}$ part of the fixed point $(\vec{s},\hat{q})$ is fully determined by (12) and (13). □

Remarks: 1) The unique fixed point $(\vec{s},\hat{q})$ can be computed by first computing q_0 in the same manner as in the exponential case. The vectors $\vec{s}_k$, for $k\geq 1$ are then computed using (12) and (13).

2) The fixed point $(\vec{s},\hat{q})$ is such that the distribution $\vec{s}$ at the servers is identical to an M/PH/1 queue with arrival rate λq_0 and an increased arrival rate λ_0 when the queue is empty (such that the probability to have an empty queue is $1-\lambda$). As such the response time distribution is the same as in an M/G/1-queue with arrival rate λq_0, therefore the mean response time $E[R]$ is given by the Pollaczek–Khinchin mean value formula:

$$E[R] = \frac{\lambda q_0 E[X^2]}{2(1-\lambda q_0)} + 1,$$

as the mean job size $E[X]=1$.

settings	Job sizes	n	q_0	$E[R]$	rel.err. %
$\lambda=0.7$, $r=5$, $d=3$	Exponential	50	0.1682	1.1457	0.0341
		500	0.1482	1.1173	0.0030
		5000	0.1463	1.1141	0.0003
		∞	0.1460	1.1138	
	HypExp $SCV=10$ $f=1/2$	50	0.1712	1.8476	0.1364
		500	0.1477	1.6428	0.0104
		5000	0.1458	1.6271	0.0007
		∞	0.1460	1.6259	-
$\lambda=0.85$, $r=10$, $d=1$	Exponential	50	0.3738	1.4878	0.0132
		500	0.3757	1.4708	0.0017
		5000	0.3752	1.4687	0.0002
		∞	0.3753	1.4684	
	HypExp $SCV=10$ $f=1/10$	50	0.3790	3.8480	0.0707
		500	0.3765	3.6012	0.0070
		5000	0.3757	3.5825	0.0018
		∞	0.3753	3.5761	

TABLE II
JIQ-SQ(d) APPROXIMATION ERROR OF THE SIMULATED q_0 AND $E[R]$ AND RELATIVE ERROR OF $E[R]$.

V. Validation of the JIQ-SQ(d) mean field model

In this section we present simulation results to verify the accuracy of the mean field models presented in Sections III and IV. We performed a large number of simulation experiments and present some arbitrarily selected cases in Table II. We performed experiments with increasing values for the number of servers $n = 50, 500$ and 5000. The number of dispatchers used equals $m = n/r$. The system was simulated for a length of $10^7/m$ time units (where the mean job size equals one time unit). A warm-up period of 33% was used and the results were averaged over several runs. Apart from the simulation results Table II also contains the value of q_0 determined by Algorithm 1 and the mean response time $E[R]$ of the corresponding

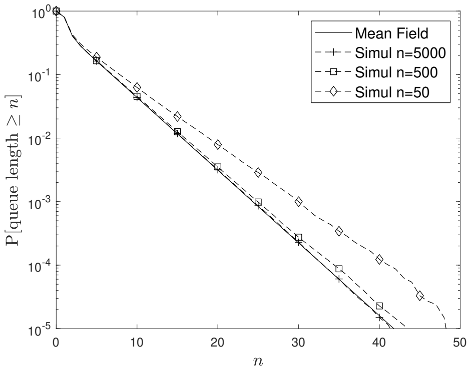

Fig. 4. Validation of the queue length distribution at server side for $\lambda = 0.8$, $d = 2$, $r = 5$ and hyper-exponential job sizes with SCV=10 and $f = 1/2$.

M/PH/1 queue with load λq_0. They are presented on the line with $n = \infty$.

We considered both exponential job sizes and hyper-exponential job sizes with mean 1. In the latter case, we set the parameters p, μ_1, μ_2, with $\alpha = (p, 1-p)$ and $T = diag(-\mu_1, -\mu_2)$, such that the squared coefficient of variation (SCV) equals 10 and $f = p/\mu_1$. Table II clearly suggests that the mean field models of Sections III and IV are asymptotically exact as the simulation results appear to converge towards the performance predicted by the unique fixed point of the corresponding mean field model.

Having validated the mean field models, it seems fair to state that for JIQ-SQ(d) with token withdrawals the response time of a job approaches the response time in an ordinary M/PH/1 queue with load λq_0 as the number of servers n becomes large (with $r = n/m$ fixed). Hence, the system behavior on the server side becomes very simple. Unfortunately determining q_0 is much harder due to the withdrawal of tokens. If we consider JIQ-SQ(d) without token withdrawals, then [8] suggests that the dynamics at the dispatcher side become very simple, e.g., for $d = 1$ the number of tokens has a geometric distribution. However, without token withdrawals the server dynamics are much more involved as the job arrival rates become queue length dependent. The mean field model in [1] can be regarded as combining the simple server side dynamics of JIQ-SQ(d) with token withdrawals with the simple dispatcher dynamics of JIQ-SQ(d) without token withdrawals. Remarkably, this combination yields a good to excellent approximation as shown in Figures 1 and 2.

We also performed several simulation experiments that demonstrated that our mean field models are also very accurate for the queue length distribution at the servers (and not just the mean), only one such example is presented in Figure 4 due to the lack of space.

VI. Conclusions

In this paper we first explained why the well-known model introduced in [1] is not asymptotically exact for JIQ-SQ(d) with or without token withdrawals. We introduced a number of performance models for JIQ-SQ(d) load balancing with token withdrawals. Simulation experiments suggest that these models provide asymptotically exact results.

For JIQ-SQ(d) with token withdrawals these models suggest that the response time distribution of a job becomes identical to that in an M/PH/1 queue with load λq_0 in the large-scale limit. The value of q_0 depends on λ, d and r, but is independent of the job size distribution (with mean 1). The token selection method used by the dispatcher does not impact the system performance.

References

[1] 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," *Perform. Eval.*, vol. 68, pp. 1056–1071, 2011.

[2] A. Stolyar, "Pull-based load distribution in large-scale heterogeneous service systems," *Queueing Systems*, vol. 80, no. 4, pp. 341–361, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11134-015-9448-8

[3] ——, "Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers," *Queueing Systems*, vol. 85, pp. 31–65, 2017.

[4] S. Foss and A. L. Stolyar, "Large-scale join-idle-queue system with general service times," *Journal of Applied Probability*, vol. 54, no. 4, p. 995–1007, 2017.

[5] C. Wang, C. Feng, and J. Cheng, "Distributed join-the-idle-queue for low latency cloud services," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2309–2319, 2018.

[6] M. Mitzenmacher, "The power of two choices in randomized load balancing," Ph.D. dissertation, University of California, Berkeley, 1996.

[7] N. Vvedenskaya, R. Dobrushin, and F. Karpelevich, "Queueing system with selection of the shortest of two queues: an asymptotic approach," *Problemy Peredachi Informatsii*, vol. 32, pp. 15–27, 1996.

[8] M. Mitzenmacher, "Analyzing distributed join-idle-queue: A fluid limit approach," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 312–318.

[9] T. Kurtz, *Approximation of population processes*. Society for Industrial and Applied Mathematics, 1981.

[10] X. Zhou, F. Wu, J. Tan, Y. Sun, and N. Shroff, "Designing low-complexity heavy-traffic delay-optimal load balancing schemes: Theory to algorithms," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, dec 2017. [Online]. Available: https://doi.org/10.1145/3154498

[11] M. van der Boor, S. Borst, and J. van Leeuwaarden, "Hyper-scalable JSQ with sparse feedback," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 1, pp. 1–37, 2019.

[12] S. Vargaftik, I. Keslassy, and A. Orda, "LSQ: Load balancing in large-scale heterogeneous systems with multiple dispatchers," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, p. 1186–1198, Jun. 2020. [Online]. Available: https://doi.org/10.1109/TNET.2020.2980061

[13] X. Zhou, N. Shroff, and A. Wierman, "Asymptotically optimal load balancing in large-scale heterogeneous systems with multiple dispatchers," *Performance Evaluation*, vol. 145, p. 102146, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166531620300663

[14] M. van der Boor, S. Borst, and J. van Leeuwaarden, "Load balancing in large-scale systems with multiple dispatchers," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[15] D. Mukherjee, S. Dhara, S. C. Borst, and J. van Leeuwaarden, "Optimal service elasticity in large-scale distributed systems," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, jun 2017. [Online]. Available: https://doi.org/10.1145/3084463

[16] D. Mukherjee and A. Stolyar, "Join idle queue with service elasticity: Large-scale asymptotics of a nonmonotone system," *Stochastic Systems*, vol. 9, no. 4, pp. 338–358, 2019.

[17] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods and stochastic modeling*. Philadelphia: SIAM, 1999.

[18] J. Kriege and P. Buchholz, *PH and MAP Fitting with Aggregated Traffic Traces*. Cham: Springer International Publishing, 2014, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-05359-2_1

[19] F. Gillent and G. Latouche, "Semi-explicit solutions for M/PH/1-like queuing systems," *European Journal of Operational Research*, vol. 13, no. 2, pp. 151–160, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377221783900772