Wasiur R. KhudaBukhsh, Sounak Kar, Bastian Alt, Amr Rizk, Heinz Koeppl. *Generalized Cost-Based Job Scheduling in Very Large Heterogeneous Cluster Systems.* In IEEE Transactions on Parallel and Distributed Systems, May 2020.

Generalized Cost-Based Job Scheduling in Very Large Heterogeneous Cluster Systems

Wasiur R. KhudaBukhsh, Sounak Kar, Bastian Alt, Amr Rizk, *Senior Member, IEEE,* Heinz Koeppl

Abstract—We study job assignment in large, heterogeneous resource-sharing clusters of servers with finite buffers. This load balancing problem arises naturally in today's communication and big data systems, such as Amazon Web Services, Network Service Function Chains, and Stream Processing. Arriving jobs are dispatched to a server, following a load balancing policy that optimizes a performance criterion such as job completion time. Our contribution is a randomized Cost-Based Scheduling (CBS) policy in which the job assignment is driven by general cost functions of the server queue lengths. Beyond existing schemes, such as the Join the Shortest Queue (JSQ), the power of d or the SQ(d) and the capacity-weighted JSQ, the notion of CBS yields new application-specific policies such as hybrid locally uniform JSQ.

As today's data center clusters have thousands of servers, exact analysis of CBS policies is tedious. In this work, we derive a scaling limit when the number of servers grows large, facilitating a comparison of various CBS policies with respect to their transient as well as steady state behavior. A byproduct of our derivations is the relationship between the queue filling proportions and the server buffer sizes, which cannot be obtained from infinite buffer models. Finally, we provide extensive numerical evaluations and discuss several applications including multi-stage systems.

Index Terms—Job Scheduling, performance evaluation, mean-field limit.

1 Introduction

Load balancing techniques are indispensable for workload distribution in data center clusters as they promise performance boosts such as a decrease in the average job response times [1]. Prominent examples include cloud scale load balancing [2], geographical data center load balancing to reduce energy costs [3] and Equal-cost Multi-path routing (ECMP). Randomized load balancing techniques based on the JSQ principle have shown remarkable performance in terms of average delay without the need to monitor the queue lengths of all cluster servers [4].

The randomized load balancing[1] set-up in a standard supermarket model can be described as follows: There are M possibly heterogeneous clusters of servers. We consider a stream of jobs arriving at a dispatcher. The dispatcher then routes each job to a single server in one of M clusters based on measured queue lengths from a randomly selected sample of servers in each of the M clusters. A simple example of such a load balancing policy is the randomized version of the classical JSQ, which instructs the dispatcher to inspect the queue lengths of a uniformly at random sample of size d of servers and then dispatch the job to the server with the shortest queue.

The main limitation of the state-of-the-art models of such randomized load balancing policies is that they are tied to the JSQ or variants thereof [5] that are, for example, weighted by the cluster mean service rate. *However, different application semantics impose different costs that describe the affinity or preference of the application towards different queue lengths.* For example, a database application, that achieves a processing speedup by combining multiple jobs, *i.e.*, when a certain buffer filling is given, is not well modeled through a plain JSQ [6]. Further, applications that run secondary servers to absorb peak loads also pose a difficult modeling task if restricted to a plain JSQ. Note that the term *costs* in the second example, although being used in this paper more generally, can be mapped directly to monetary costs in distributed computing infrastructures such as Amazon's AWS [7].

Our goal in this paper is to provide a model that captures many classes of Cost-Based Scheduling (CBS) policies. We provide several examples of CBS policies in Section 3. Given that today's cluster systems have tens of thousands of servers [8] with a steady growth in the number of servers, we note that standard techniques to calculate performance metrics based on individual queue lengths are tedious, if not impractical. Further, we consider the practical case of *finite* buffer systems in contrast to recent approaches focusing on idealized infinite buffer systems such as [5]. It was shown

- *Wasiur R. KhudaBukhsh is currently with the Mathematical Biosciences Institute, The Ohio State University, USA and was with the Bioinspired Communication Systems Lab (BCS), Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Germany during the major part of the project.*
  Email: khudabukhsh.2@osu.edu
- *Sounak Kar is with the Multimedia Communications Lab (KOM), Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Germany.*
  Email: sounak.kar@kom.tu-darmstadt.de, amr.rizk@kom.tu-darmstadt.de
- *Amr Rizk is with the Institute of Measurement, Control and Microtechnology, Universität Ulm, Germany.*
  Email: amr.rizk@uni-ulm.de
- *Bastian Alt and Heinz Koeppl are with the Bioinspired Communication Systems Lab (BCS), Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Germany.*
  Email: bastian.alt@bcs.tu-darmstadt.de, heinz.koeppl@bcs.tu-darmstadt.de

Manuscript received April XX, 2019; revised MAY XX, 2020.

1. We use the terms load balancing and scheduling interchangeably.

in a recent paper [9] that the queue system's behavior is fundamentally different when finite buffers are considered. In a similar vein, we demonstrate that finite buffer lengths have a non-trivial impact on certain Quality of Service (QoS) metrics.

Our contributions are as follows: *(i)* A scaling limit in the form of Laws of Large Numbers (LLNs) for the queue length proportions for increasing number of servers ($N \to \infty$) given CBS and heterogeneous clusters. We also describe extension of our results to the case of batch arrivals with varying batch sizes. The scaling limit enables the analysis and comparison of different CBS policies in transient as well as steady state regimes. *(ii)* We numerically show that certain QoS metrics decay rapidly with the buffer size; this behavior indicates that the common assumption of infinite buffer systems can be inaccurate. *(iii)* We provide several numerical results showing the accuracy of the scaling limit and comparisons of the queue length distributions for different cost functions. The techniques used in this paper to prove the scaling limit (weak convergence) of various Markov processes are well known in the probability theory literature and are straightaway borrowed from [10]. In fact, such techniques have been used to derive fluid or mean-field limits for JSQ-type scheduling strategies in queuing theory literature as well (see [5], [11], [12]). By virtue of the novel introduction of an additional cost function in this paper, we are able to provide a performance evaluation of a much wider class of load balancing strategies including ones that are not necessarily of the JSQ paradigm.

The paper is organized as follows: in Section 2, we discuss the related works. In Section 3, we introduce the queuing set-up and the CBS policy before presenting our scaling limit in Section 4. Numerical results are presented in Section 5. In Section 6, we briefly describe an extension of our results to the case of variable batch sizes. We conclude the paper with a short discussion in Section 7. Additional mathematical derivations are provided in Appendix A and Appendix B.

2 Related Work

A wide range of randomized strategies including the schemes 1 and 2 presented in [5] can be viewed as CBS strategies. The first prominent randomized strategy was the power of two strategy [4], [13], which was also generalized to the power of d strategy or the so-called SQ(d) strategy [14], [15]. A survey of SQ(d)-type load balancing strategies for large systems is provided in [16]. Mitzenmacher proved in [4] that the expected time spent in a supermarket model with N servers improves exponentially when d, the number of servers sampled for job assignment, is increased from 1 to 2 in the asymptotic regime, *i.e.*, when $N \to \infty$; the improvement is only by a constant factor when d increases from 2 to 3. This asymptotic estimate is remarkable, but it is not necessarily accurate for a finite number of servers. In this case, upper and lower bounds on the average delay are provided in [15]. Additionally, in [17] a bound for the queue lengths is estimated for the supermarket model under appropriate arrival rate scaling.

Scaling limits, such as LLNs and Central Limit Theorems (CLTs) (or their functional counterparts), for large systems provide a convenient tool for performance evaluation. In [18], the authors study the stability and convergence of various moments for a multi-class infinite-buffer queuing system based on a fluid limit. They provide sufficient conditions for the existence of long-run average queue length. In a later paper [19], the authors provide a Semi-martingale Reflecting Brownian Motion (SRBM) limit for a suitably scaled queue length process in a finite-buffer open queuing network with deterministic, feedforward routing. The paper also discusses how the established results can be extended to derive heavy traffic limit theorems for networks with finite buffers. Recently, [14] studies the power-of-d (with d depending on the total number of servers) load balancing strategy in large-scale identical server systems and provides fluid and diffusion limits for scaled buffer occupancies as $N \to \infty$. The authors show that the fluid and diffusion limit of their policy coincide with that of JSQ under certain assumptions. In [20] the authors consider a bin-packing problem. Here, the power-of-d routing policy is analyzed for homogeneous servers with job-type dependent limited resources. The authors derive an explicit upper bound for the equilibrium blocking probability using a fluid limit. Similarly, the authors in [12] focus on the stability of infinite-buffer queues under randomized JSQ strategy based on LLN-type mean field limits. They show that the uniform sampling of servers may reduce the stability region when the clusters are heterogeneous. In [21] and [22] the authors consider a parallel single-server queueing system with JSQ strategy. In their works the stationary distribution of the occupancy measure for the system is considered, where they estimate the tail-asymptotics and the bulk behavior.

The usual method of obtaining scaling limits (both LLN and CLT) in the weak sense is to use Kurtz's approximation theorems for density-dependent Markov jump processes [10]. When the service times are allowed to be non-exponentially distributed, the system loses the Markov property. However, under certain conditions such as boundedness of the hazard function for the service time distribution, it is still possible to derive fluid limits. For instance, in [23], the authors consider the SQ(d) load balancing strategy in a homogeneous parallel queuing system with infinite buffers and derive a fluid limit in the form of Partial Differential Equations (PDEs) as $N \to \infty$. Another approach is to include the age of the jobs in a queue to manufacture a Markovian descriptor of the system when the service times are non-exponential. In [24], a measure-valued process keeping track of the ages of all jobs in service at queues of various lengths was used as such a Markovian descriptor in the same queuing set-up as [23] for the SQ(d) strategy. The dynamics of a scaled version of the descriptor were further approximated by a hydrodynamic limit in [24] under boundedness or lower semicontinuity of the hazard function of the service time distribution. Authors in [25] comes up with a similar measured valued process state descriptor which keeps track of residual service times, queue length, workload and cumulative idle time in a heavily loaded processor sharing queue. They first show that, under mild conditions, a fluid model solution exists and is unique in the critical regime where arrival and service rates are equal. Further, under mild conditions and for a suitable scaling, weak convergence of state descriptors to a fluid limit is

established where the sample paths of the fluid limit are almost surely fluid model solutions. Similarly, a fluid limit for the scaled state descriptor a GI/GI/1 processor sharing queue with reneging customers is presented in [26]. In [27], authors represent the remaining processing time of customers in a M/GI/∞ queue using a point-measure valued process and establish weak convergence of the rescaled process to a fluid limit.

A range of literature in this space discuss about the maximum load in a multi-server queueing system under SQ(d) routing. Azar et al. show in [28] that SQ(2) strategy provides a reasonable trade-off between JSQ and SQ(1) policy as the load on n servers varies by a $O(\log\log n)$ factor compared to $O(\log n)$ in case of SQ(1). In [29], the authors discuss maximum load under SQ(d) policy in a balls-and-bins model where balls have independent unit exponential lifetime. In particular, they consider a system of n servers with Poisson arrivals with rate λn and show that the system converges quickly to the equilibrium distribution under which the maximum load is concentrated on two values with probability that tend to 1 as $n \to \infty$. It is also shown that the maximum deviation of maximum load from $\log\log n/\log d$ is constant. A similar result about the maximum queue length is achieved for the supermarket model with same parameter setting in [30] where the arrival parameter λ is restricted to $(0, 1)$ due to stability reasons.

Although infinite-buffer queues are often assumed for analytical treatment and are meant to approximate large buffer sizes, the buffer sizes in practice are not always large. Therefore, one needs dedicated finite-buffer results. We refer the readers to [31], [32] for a survey of developments in this direction. There has been many attempts to derive performance bounds for finite-buffer systems by suitably approximating infinite-buffer systems. For instance, the authors in [33] approximate the total loss probability in a finite-buffer constant rate server and fluid input system by the tail probability of the queue length distribution in an infinite-buffer system. In [34], the authors present an analytic queuing model for finite-buffer systems capturing the correlations among the queues by means of certain structural parameters. The SRBM limit mentioned earlier was obtained for a suitably scaled queue length process in a finite-buffer open queuing network in [19]. Such limits can be used to compute the stationary queue length distributions. For instance, the authors of the paper [35] apply a finite element method to compute the stationary distribution in a finite buffer queuing network. Another approach has been to use decomposition methods to evaluate various performance metrics approximately (see [36], [37]).

3 Scheduling in large resource-sharing pools with finite buffers

In this section, we first describe the queuing set-up and then, explain the proposed CBS policy. We consider the standard supermarket model (*e.g.*, see [5]) and propose the following modifications: (i) we consider finite-buffers instead of infinite buffers; and (ii) we generalize the scheduling algorithm to allow user-defined cost functions associated with the queue-lengths of a randomly selected subset of servers. By virtue of (ii), a wide range of innovative load balancing algorithms can be accommodated into the model and naturally, the schemes 1 and 2 in [5] are obtained as special cases. Later in Section 6, we provide an extension to the case of batch arrivals with varying batch sizes.

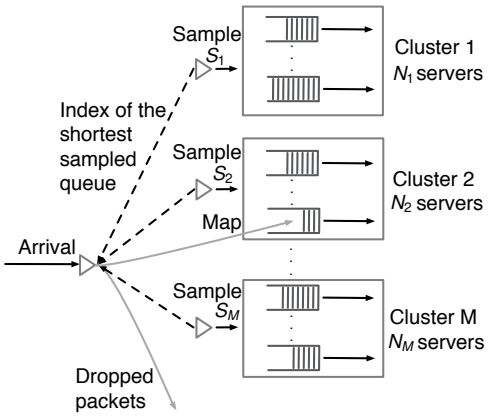

Fig. 1: Schematic description of the CBS policy: There are M heterogeneous server clusters where cluster i contains N_i homogeneous servers each with exponential service rate μ_i. Upon a job arrival, the local routers in each of the clusters randomly sample (with replacement) S_i servers and return to the dispatcher the index of the shortest queue among the sample. The dispatcher then compares the costs associated with these servers before assigning the job to the server with the minimum cost. If the assigned buffer is full the job is lost.

3.1 Description of the queuing set-up

We consider an N-server parallel processor sharing queuing system where the N servers are partitioned into M ($\ll N$) heterogeneous clusters of servers within each of which the servers are identical (see Figure 1). The service discipline is First In First Out (FIFO). Let $[N] := \{1, 2, \ldots, N\}$. We assume the i-th cluster contains N_i identical servers with exponentially distributed service times with rate μ_i. Naturally, $N_1 + N_2 + \cdots + N_M = N$ and M is assumed fixed throughout. Let I_i contain the indices of the servers in the i-th cluster. Then, $|\ I_i\ | = N_i$ and $\{I_1, I_2, \ldots, I_M\}$ is a partition of $[N]$, *i.e.*, $[N] = \bigcup_{i\in[M]} I_i$ and I_i's are disjoint. Let the buffer size of *all* N servers be K. Extension to the case of unequal buffer sizes is not difficult. In that case, we can take K to be the maximum of the different buffer sizes and modify the definitions of our stochastic processes and the transition rates of the underlying Markov process accordingly.

The cluster-structure could arise because of geographic location or otherwise. Since the servers within each cluster are assumed to be identical or homogeneous with respect to service time distributions, the different clusters represent different types of servers (fast, slow etc). Therefore, the clustering of servers provides a natural way of extending a homogeneous server setting to a heterogeneous one, which is arguably better representative of real-world situations.

Jobs arrive at the system according to a Poisson process with rate λ_N. The inter-arrival times, and the service times of each job are *all* assumed to be independent of each other.

Upon the arrival of each job, the dispatcher assigns the job to exactly one server following a CBS policy, which we describe next.

3.2 Cost-Based Scheduling

A local router is placed in each of the clusters. As seen in Figure 1, at the arrival of each job, the main dispatcher sends a request to each of the M local routers. The local router in the i-th cluster samples S_i servers uniformly at random with replacement. Suppose the queue lengths at the sampled servers at the i-th cluster are $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_l}$ where $l = S_i$. The local router returns to the dispatcher the index and the queue length of the server with the shortest queue length. That is, the local router computes

$$\iota_i := \underset{k\in\{i_1,i_2,\ldots,i_l\}}{\arg\min} \{Q_k\}, \forall\, i \in [M].$$

In case of a tie within a cluster, we break tie by choosing one of the tied indices uniformly at random. After receiving the indices $\iota_1, \iota_2, \ldots \iota_M$ from all M clusters, the dispatcher assigns a cost to each of the corresponding queue lengths. Finally, comparing the costs, the dispatcher assigns the job to the server with the minimum cost. Therefore, the index ι of the server to which the job is finally assigned is given by

$$\iota := \arg\min_{\iota_k} \{\phi_k(Q_{\iota_k}) \mid k \in [M]\},$$

where ϕ_k is the cost function associated with the k-th cluster. We assume the cost functions are user-defined and continuous[2]. In case of a tie, we break tie by choosing one of the tied indices uniformly at random. Jobs leave the system as soon as their services have been provided. We assume the scheduling task is instantaneous for modeling purposes. Having described the CBS policy formally, we provide some concrete examples next.

3.3 Choice of the cost functions

The CBS is a generalized load balancing policy. Setting $M = 1, S_1 = N, \phi_1(x) = x$ corresponds to the usual JSQ policy. Similarly, $M = 1, S_1 = 2, \phi_1(x) = x$ corresponds to the power of 2 type JSQ, and to SQ(d) policy with $S_1 = d$. Choosing $\phi_k(x) = x$ corresponds to the randomized JSQ policy over clusters. Other variants of JSQ, or a mixture of them, can be incorporated by choosing the cost functions appropriately. It is worth noting that any strictly monotonically increasing ϕ_k would give rise to a JSQ-type policy (or a randomized version of it) provided ϕ_k's are identical across the M clusters. To give one more example, $\phi_k(x) = x/\mu_k$ corresponds to scheme 2 in [5] (see F2 in Figure 2).

In addition to the different variants of JSQ, we can design a wide range of cost functions to achieve other performance objectives as depicted in Figure 2. For instance, a constant $\phi_k(x) = c_k$ for all x yields a preference policy or a uniform load balancing across all clusters for $c_k = c$ for all k (see F3 in Figure 2). In fact, one can design hybrid policies that are conceptually similar to adaptive queue management such as [38]. The function F4 is a simple policy that ensures uniform load balancing when the queue length is small; but changes to JSQ when the queue length grows large. The policy corresponding to the cost function F5, which can be seen as a locally uniform hybrid JSQ policy, ensures uniform load balancing for neighboring queue lengths but ensures a JSQ-type behavior when the queue lengths are far apart.

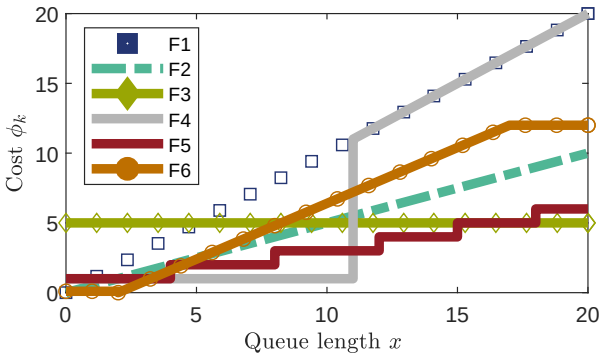

Fig. 2: Interpretations of different cost functions: The function F1 ($x \mapsto x$) corresponds to a JSQ-type policy. On the contrary, F2 given by $x \mapsto x/\mu$ (depicted for $\mu > 1$) takes into account the cluster specific service rate. The policy of F3, given by $x \mapsto 5$ here, ensures uniform load balancing. The function F4 yields uniform load balancing when the queue length is small and a JSQ-type behavior when the queue length is large. F5 yields a locally uniform hybrid JSQ policy in that it enforces a uniform load balancing when the queue lengths being compared are close to each other, and a JSQ-type behavior when they are far apart. Finally, F6 reflects a cost structure that exhibits JSQ-type behavior between two threshold queue lengths and does uniform load balancing otherwise.

In addition, one could use combinations of cost functions to express application semantics on top of the server clusters. For example, using clusters with cost function F1 in combination with clusters with F3, one can differentiate primary load-bearing server clusters from secondary peak-load absorbing ones. Some typical examples in practice for the cost functions discussed above include: multipath TCP default scheduler for F2, elastic load balancer in AWS for F3 and RED algorithm used in active queue management for F6. The RED algorithm, defined in [39], chooses packet to drop from incoming queues according to a dropping probability that varies linearly between two thresholds and remains constant otherwise. *One of the major advantages of our formulation is that we can choose different cost functions for different clusters.* Therefore, we can design a wide range of novel mixed and hybrid policies based on the characteristics of the clusters. With these examples in mind, we proceed to derive a scaling limit in the next section.

4 A Scaling Limit

In this section, we provide a scaling limit of the system as the number of servers goes to infinity for the CBS policies.

2. Note that continuity is vacuously satisfied if we restrict ourselves to integer-valued queues only, as we do in this work. However, continuity should be additionally assumed if we generalize the cost function to other domains.

4.1 Why do we need a scaling limit?

Performance evaluation of CBS policies, *e.g.*, the variants of JSQ mentioned above, can be done in various ways. The most straightforward of them is to compute the marginal probabilities of the Markov chain by solving the Kolmogorov forward equations. However, the number of equations increases exponentially with the number of servers N. Therefore, this approach is practically infeasible. Alternatively, one can estimate the probabilities using Monte Carlo simulations. However, Monte Carlo simulations are also time consuming, especially for large N.

Our approach to performance evaluation of CBS policies for increasingly large systems is to derive a scaling limit that essentially captures the limiting mean of the proportions of servers with specific numbers of unfinished jobs. After showing the accuracy of this scaling limit, we use it to study the performance and suitability of different CBS policies.

The scaling limits are particularly beneficial in terms of computational cost because we only need to solve a system of $(K+1)M$ autonomous Ordinary Differential Equations (ODEs). This is in sharp contrast to a computational cost that increases with N for the Kolmogorov forward equations or the Monte Carlo approach. Note that the number of equations, *i.e.*, $(K+1)M$, is independent of the total number of servers N but is only dependent on the buffer size K and the number of clusters M.

4.2 Technical assumptions

We first define the key stochastic processes in the system. Since CBS policies belong to the class of Cost-Based Queue-Aware (CBQA) randomized policies, it is analytically convenient to work with stochastic processes that keep track of the queue lengths at different servers or some suitably designed summary statistics of those. The latter is often desirable from a computational perspective. Therefore, in order to be parsimonious with respect to dimensionality and computational resources, we shall work with empirical tail distributions of the queue lengths in different clusters.

We keep track of the proportions of servers (within each cluster) that have at least a fixed number of unfinished jobs. Define the stochastic process

$$Z_N(t) \coloneqq \{Z_{n,i}^{(N)}(t) \mid i \in [M], n = 0, 1, 2 \ldots, K\}, \quad (1)$$

where $Z_{n,i}^{(N)}(t)$ is the fraction of servers at the i-th cluster having at least n unfinished jobs (the queue length) at time t. That is,

$$Z_{n,i}^{(N)}(t) \coloneqq \frac{1}{N_i} \sum_{k \in I_i} \mathbb{1}(Q_k(t) \geq n), \quad (2)$$

where $\mathbb{1}(F)$ is the indicator function taking value 1 if F is true and zero otherwise. We have deliberately sub-/superscripted the processes with N to emphasize their dependence on N. The process Z_N is a Markov process on the state space $\times_{i \in [M]} \mathcal{Z}_i$, the Cartesian product of $\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_M$, where

$$\mathcal{Z}_i \coloneqq \{\{a_n\}_{n=0,1,2\ldots,K} \mid a_0 = 1, a_n \geq a_{n+1}, N_i a_n \in \mathbb{N}\},$$

for all $i \in [M]$. The symbol $\mathbb{N}$ denotes the set of natural numbers. Note that the first coordinate $Z_{0,i}$ is the proportion of servers within the i-th cluster with at least zero unfinished jobs, and hence, is unity for all $i \in [M]$. Within the i-th cluster, these proportions are non-increasing in n. The last condition $N_i a_n \in \mathbb{N}$ ensures a_n's are valid proportions.

Next, we need to make precise how the arrival process and the number of servers within each cluster scale with N. Therefore, we make the following technical assumptions.

A1 **(Arrival rate)** We assume the arrival rate grows linearly with N, *i.e.*,

$$\lim_{N \to \infty} \frac{\lambda_N}{N} = \lambda \in \mathbb{R}_+.$$

The rationale behind this assumption is to avoid trivialities. If the limit λ is allowed to be infinity, then all servers will be full all the time. On the other hand, if it is zero, the servers will remain idle.

A2 **(Non-vanishing proportion of servers in each cluster)** Each cluster contains a non-vanishing proportion of the total pool of servers in the limit. That is, the cluster sizes grow linearly with N, *i.e.*

$$\lim_{N \to \infty} \frac{N_i}{N} = \nu_i \in (0, 1), \quad \forall i \in [M].$$

We make this assumption to ensure *all* clusters are large.

A3 **(Decidability)** This allows us to compare different servers across the clusters in the light of the particular CBS policy employed. If the policy assigns a job to a server with n unfinished jobs in the i-th cluster, we should be able to tell what the minimum of the sampled queue lengths in different clusters must have been, *e.g.*, n in case of a simple randomized JSQ. Therefore, given the index ι (and the corresponding queue length) of the chosen server in accordance with the CBS procedure described in Section 3.2, we assume the cost functions ϕ_i's allow us to calculate the minimum of sampled queue lengths across the clusters. Suppose $\iota \in I_i$, *i.e.*, the chosen server is in the i-th cluster. Define

$$\theta_j(i, x) \coloneqq \underset{y \in \{0,1,2,\ldots,K\}}{\arg\min} \{\phi_j(y) \geq \phi_i(x)\},$$

for all $i, j \in [M]$, $x \in \{0, 1, 2, \ldots, K\}$. The interpretation is as follows: if the CBS strategy assigns an incoming job to a server with x unfinished jobs in the i-th cluster, the minimum of the sampled queue lengths at the j-th clusters must have been at least $\theta_j(i, x)$. Our assumption of decidability amounts to demanding $\theta_j(i, x) \neq 0$ for *at least* one $j \neq i$, for all $i \in [M]$ and for all $x > 0$.

We emphasize that the assumption A3 is desirable and not crucial for the mathematical derivations. For instance, F3 in Figure 2 violates A3. The only purpose behind A3 is to avoid trivialities such as $\theta_j(i, x) = 0$ for all $x \in \{0, 1, 2, \ldots, K\}$.

4.3 Main result

Having laid down the technical assumptions, we begin our investigation of the scaling limit. As the components of the stochastic process Z_N are all proportions, we expect, at least intuitively, $Z_N(t)$ to converge to a deterministic quantity for each t as N goes to infinity, by virtue of the LLN. Therefore, we expect the process Z_N to converge to a deterministic

function as N goes to infinity, again at least intuitively. In the following, we shall show that our intuition is indeed right. Let $\mathcal{T} \coloneqq [0, T]$ be our time interval of interest, for some $T > 0$.

In this paper, we shall adopt the operator semigroup approach to prove convergence of the Markov process Z_N. Therefore, define a sequence of one-parameter families of operators $\{T_N(t)\}_{t\in\mathcal{T}}$ as follows

$$T_N(t)f(z_0) \coloneqq \mathsf{E}[f(Z_N(t)) \mid Z_N(0) = z_0],$$

for all $t \in \mathcal{T}, z_0 \in \times_{i\in[M]}\mathcal{Z}_i$ and for continuous functions $f : \times_{i\in[M]}\mathcal{Z}_i \to \mathbb{R}$. The family $\{T_N(t)\}_{t\in\mathcal{T}}$ defines a strongly continuous (contraction) semigroup [40], [41] by the Chapman-Kolmogorov property of Markov processes [10, Chapter 4].

The idea is to first prove that $T_N(t)$ converges to some limiting operator semigroup T as N goes to infinity. This can be achieved by showing convergence of the corresponding sequence of (infinitesimal) generators A_N of T_N to the generator A of T. Having shown the convergence of the operator semigroup T_N to T, the convergence of the Markov process Z_N follows immediately in the light of [10, Chapter 4, Theorem 2.11, p. 176]. This is also noted in [5] for the infinite-buffer case. In essence, we need to carry out three steps: 1) prove convergence of the generators A_N to A (Lemma 3 in Appendix A); 2) establish convergence of T_N to T (Theorem 3 in Appendix A); and then finally 3) establish convergence of Z_N to its scaling limit z by invoking [10, Chapter 4, Theorem 2.11, p. 176]. For the sake of simplicity, we defer the involved calculations needed to carry out the above three steps to the Appendix A, and only state the final convergence result in Theorem 1 below.

Please note that space $\times_{i\in[M]}\mathcal{Z}_i$ in which Z_N lies is finite for each finite N (because the proportions can only take finitely many values). However, as N goes to infinity, the proportions can eventually approximate any real value in $[0, 1]$. Therefore, define

$$\mathcal{Z} \coloneqq \{\{a_n\}_{n=0,1,\ldots,K} \mid a_0 = 1, a_n \geq a_{n+1}, a_n \in [0, 1]\}.$$

In the limit, we expect Z_N to lie in $\mathcal{Z}^M$, the $(M-1)$-fold Cartesian product of $\mathcal{Z}$ with itself. Now, we state the convergence result.

Theorem 1 (Convergence of the server cluster proportions)**.** *If $\lim_{N\to\infty} Z_N(0) = z_0$, for some non-random $z_0 \in \mathcal{Z}^M$, then*

$$Z_N \overset{\mathcal{D}}{\Longrightarrow} z, \text{ as } N \to \infty, \tag{3}$$

where the limiting process z satisfies the integral equation

$$z(t) = z(0) + \int_0^t \mathbb{F}(z(s))\mathrm{d}s, \tag{4}$$

with $z(0) = z_0$, and the function $\mathbb{F}(u) \coloneqq \{\mathbb{F}_{n,i}(u) \mid i \in [M], n = 0, 1, 2, \ldots, K\}$ is given by

$$\mathbb{F}_{0,i}(u) \coloneqq 0, \quad \forall i \in [M],$$

$$\mathbb{F}_{n,i}(u) \coloneqq \frac{\lambda}{\nu_i}\left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}\right) \prod_{j\in[M]\setminus\{i\}} (u_{\theta_j(i,n-1),j})^{S_j} - \mu_i\left(u_{n,i} - u_{n+1,i}\right),$$

for $i \in [M]$, $n = 1, 2, \ldots, K$, and $u \in \mathcal{Z}^M$. Weak convergence is understood in the sense of [10, Chapter 3, p. 107], [42].

For the sake of completeness, proof of Theorem 1 is provided in Appendix A. From a practical perspective, the operator $\mathbb{F}(z)$ can be thought of as the derivative of the deterministic process z. Note that questions concerning stability of the queueing system do not arise because the system is stable regardless of the arrival and service rates by virtue of finiteness of the buffers. However, the accumulated loss process is increasing because it has only positive jumps.

4.4 Properties of the limit

Let us now discuss some properties of the proposed limit. Since the limit is an autonomous system of ODEs, the first natural question that comes up is regarding the uniqueness of the solution to (4). As stated in the next lemma, the solutions are indeed unique.

Lemma 1. *For any starting point $u \in \mathcal{Z}^M$, the solution to the integral equation* (4) *is unique on $\mathcal{T}$.*

The proof of Lemma 1 is provided in Appendix A. The next lemma is regarding the smoothness of the solution with respect to the initial conditions. As the limiting process $z(t)$ depends on the initial value $z(0)$, we introduce the notation $z(t, u)$ to denote the solution of the integral equation (4) with $z(0) = u$. We require certain smoothness of the solution $z(t, u)$ and bounded partial derivatives with respect to the initial point u. Therefore, we have the following lemma.

Lemma 2. *The partial derivatives*

$$\frac{\partial}{\partial u_{n,i}}z(t,u), \ \frac{\partial^2}{\partial u_{n,i}^2}z(t,u), \ \text{ and } \ \frac{\partial^2}{\partial u_{n,j}u_{n,i}}z(t,u)$$

exist for all $u \in \mathcal{Z}^M$, and are uniformly bounded above

$$\left|\frac{\partial}{\partial u_{n,i}}z(t,u)\right| \leq \exp(at), \tag{5}$$

$$\left|\frac{\partial^2}{\partial u_{n,i}^2}z(t,u)\right|, \left|\frac{\partial^2}{\partial u_{n,j}u_{n,i}}z(t,u)\right| \leq \exp(bt), \tag{6}$$

for some constants $a, b \in \mathbb{R}_+$.

The proof of Lemma 2 is the same as [5, Lemma B.2] and [11, Lemma 3.2] except for some minor changes in the constants. However, for the sake of completeness, a brief sketch is provided in Appendix A. Next, we numerically evaluate the theoretical results presented in the current section.

5 Numerical Evaluations

In this section, we seek to gain insights into CBS-driven systems in terms of its evolution with time, steady state and also the indicative effect of buffer length on the job loss. Using Theorem 1, we can circumvent simulating the cluster system and still derive meaningful results about its performance such as the queue proportions when the number of servers in each cluster grows large. To validate our scaling limit, we compare the simulated system with the scaling limit. We use the generative model with CBS scheduling as described in Section 3.1. We consider five simulation scenarios with growing number of servers and use 10^2 Monte Carlo simulations for each setting to finally perform averaging. If not stated otherwise, the numerical

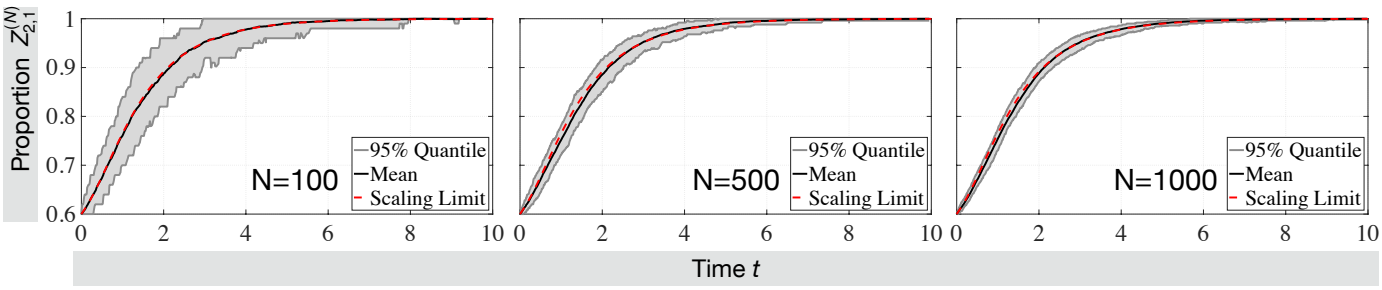

Fig. 3: Accuracy of the scaling limit in Theorem 1: Comparison of the time evolution of $Z_{2,1}^{(N)}$ from (2) for increasing number of servers N against the scaling limit $z_{2,1}$ denoting the proportion of servers with queue length greater than or equal to 2 in cluster number 1. The fluctuation of $Z_{2,1}^{(N)}$ around $z_{2,1}$ decrease as N increases validating the convergence.

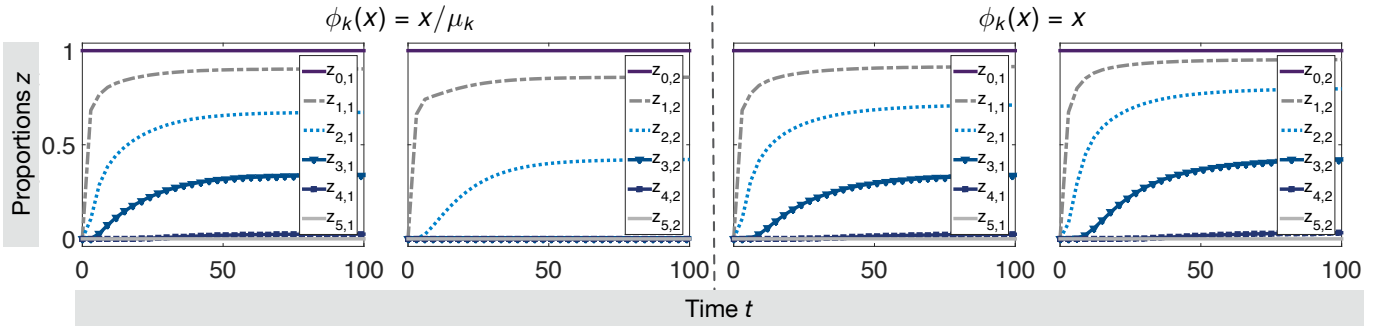

Fig. 4: Time evolution of cluster queue length proportions for two choices of cost function ϕ: $\phi_k(x) = x/\mu_k$ and $\phi_k(x) = x$. Simulation settings: $M = 2$, $K = 5$, $\lambda = 1/4$, $(\nu_1, \nu_2) = (0.4, 0.6)$, $(\mu_1, \mu_2) = (1/2, 1/4)$, time horizon $= [0, 100)$. At a fixed point, $z_{i,j}$ gives proportion of servers with queue length greater than or equal to i in cluster j at that instant. The subplot on the left shows cluster queue length proportions for $\phi_k(x) = x/\mu_k$ while the right one is for $\phi_k(x) = x$. As expected, the former picks servers from the faster cluster ($\mu_1 = 1/2$) more frequently than the latter.

settings for the figures are given by number of clusters $M = 2$, buffer size $K = 5$, arrival rate $\lambda = 1/4$, proportion of servers in each cluster $(\nu_1, \nu_2) = (0.4, 0.6)$, service rates within the cluster $(\mu_1, \mu_2) = (1/2, 1/4)$, and time horizon $= [0, 100)$.

First, we depict the time evolution of the queuing system in Figure 3. The scaling limit matches the empirical mean of the Monte Carlo simulations well and the variance of the simulations reduces as the number of servers N grows. It is worth noting that the scaling limit can be used to describe the behavior of the system not only in the steady state, but also during the transient phase.

Furthermore, the time evolution of the proportions of servers corresponding to possible queue lengths for each cluster are shown in Figure 4. Here, the cost functions mirror randomized JSQ ($\phi_k(x) = x$) and service-rate-weighted randomized JSQ ($\phi_k(x) = x/\mu_k$) scheduling. In comparison to randomized JSQ, service-rate-weighted randomized JSQ takes advantage of the faster server as reflected through the larger proportions of longer queue lengths.

We further explore the time evolution in heterogeneous environment in Figure 5 where the cluster service rates vary. For both clusters, we consider cost functions F1 (randomized JSQ), F3 (uniform load balancing) and F4 (initially uniform and switching to randomized JSQ beyond a threshold) from Figure 2. We change the simulation settings to $K = 8$, $\lambda = 1/8$, and $(\mu_1, \mu_2) = (1, 1/16)$ to underline heterogeneity. We observe that the cost function F3 and F4 eventually enforces almost all servers in the slower cluster to be busy whereas, under F1, few servers remain idle (left subplot). Further, under F4 certain servers have more workload as seen in the right subplot of Figure 5. This is due to the fact that until a certain queue length (3 in case of Figure 5), there is no preference for servers with lower workload. Also, compared to F3, F4 has positive proportion of servers with intermediate queue length.

We also look at the steady-state proportions of queue lengths for each cluster under different arrival intensities. This is shown in Figure 6 under *high*, *medium* and *low* arrival rates, *i.e.*, $\lambda \in \{1/4, 1/2, 1\}$, respectively.

Finally, we show in Figure 7 the impact of buffer sizing on the time averaged proportion of full servers in the whole system. This QoS metric expresses how full the system is and it may act as an approximate indicator of job loss. We fix a long time horizon $= [0, 400)$ to calculate the time averaged proportion. It is seen that the buffer length has a quickly diminishing impact on this metric, which is a strong argument for small sized buffers when designing such load-balanced cluster systems. Note that our result provides this relationship between the queue proportions and the server buffer sizes, which cannot be obtained from state-of-the-art infinite buffer results such as [5].

6 Extension to Batch Arrivals

In this section, we discuss how the scaling limit can be extended to the case when the jobs arrive in batches of

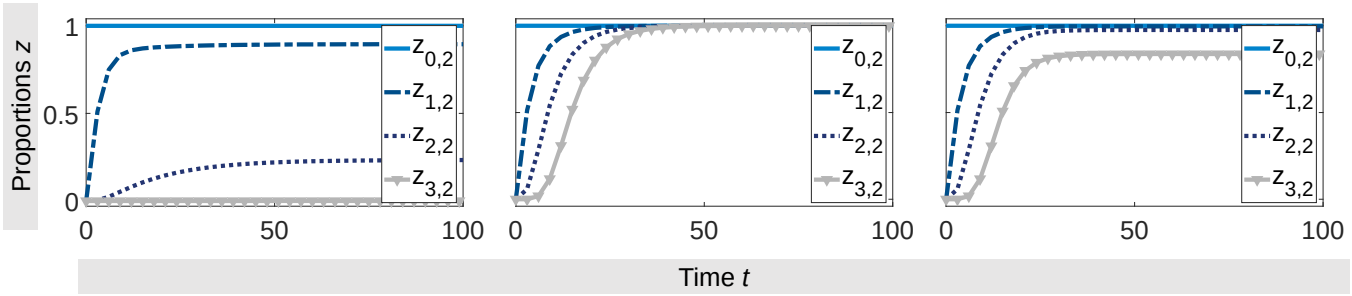

Fig. 5: Queue length proportions at the slower of two heterogeneous clusters. Cost functions chosen are: F1 (randomized JSQ), F3 (uniform load balancing) and F4 (initially uniform and switching to randomized JSQ at a threshold of 4), respectively from Figure 2. We plot $z_{i,2}$, the proportion of servers with queue length greater than or equal to i in the slower cluster. While F1 has generally shorter queues (left subplot), F3 does not allow servers to idle (middle subplot). In contrast to F3, F4 has positive proportion of servers with intermediate queue length.

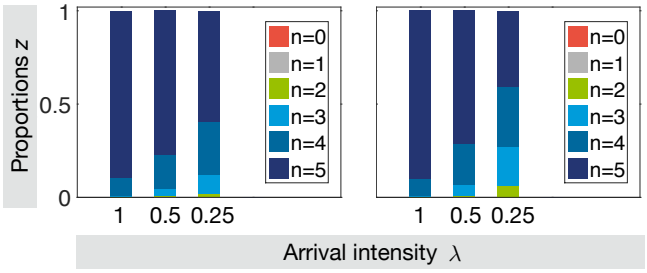

Fig. 6: Steady state queue length (n) proportions for cluster 1 (left) and cluster 2 (right) for different arrival intensities under CBS scheduling with $\phi_k(x) = x/\mu_k$.

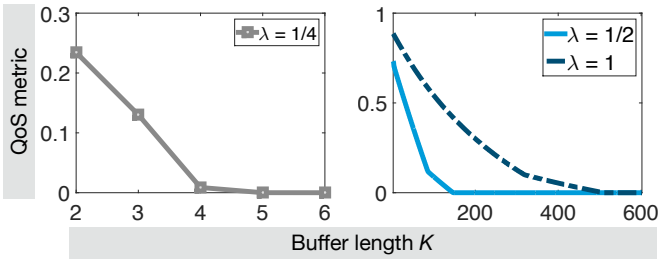

Fig. 7: Impact of the buffer size on time-averaged proportion of full servers (QoS metric) in a CBS load-balanced cluster system for different arrival intensities. The buffer length has a diminishing impact on the QoS metric.

variable size. We assume the batch sizes follow a probability distribution π, *i.e.*, the probability that the batch size of an incoming job is d is $\pi(d)$. Since we have a finite-buffer system, we can assume the probability distribution π has a bounded support. Let D denote the largest batch size. We also assume the batch sizes of different incoming job-batches are independently distributed and are independent of the arrival times. Upon arrival of an incoming batch of size d, the entire batch is allocated to a single server. In case there is no server (among the ramdomly sampled servers at each cluster) that can accommodate the entire job, *i.e.*, if the number of unfinished jobs is larger than $K - d$, the part of the incoming batch that could not be accommodated is dropped and hence lost. The rest of the scheduling procedure remains the same as before. Jobs leave the system one by one as they are serviced. That is, even if a job arrives in a batch of size d, each of the d jobs leaves separately as soon as it receives service.

We retain the notations and stochastic processes defined earlier. In particular, we consider the same stochastic process $Z_N(t)$ defined in (1). For such a queuing system with variable batch sizes, the scaling limit for Z_N as $N \to \infty$ can be proved using similar machinery. The statement of the scaling limit is presented below.

Theorem 2 (Convergence of the server cluster proportions: batch arrival case)**.** *If* $\lim_{N\to\infty} Z_N(0) = z_0$*, for some non-random* $z_0 \in \mathcal{Z}^M$*, then*

$$Z_N \overset{\mathcal{D}}{\Longrightarrow} z, \text{ as } N \to \infty, \tag{7}$$

where the limiting process z *satisfies the integral equation*

$$z(t) = z(0) + \int_0^t \mathbb{G}(z(s))\mathrm{d}s, \tag{8}$$

with $z(0) = z_0$*, and the function* $\mathbb{G}(u) := \{\mathbb{G}_{n,i}(u) \mid i \in [M], n = 0, 1, 2, \ldots, K\}$ *is given by*

$$\mathbb{G}_{0,i}(u) := 0, \quad \forall i \in [M],$$

$$\mathbb{G}_{n,i}(u) := \sum_{d=1}^{\min\{n,D\}} \pi(d) \frac{\lambda}{\nu_i} \left((u_{n-d,i})^{S_i} - (u_{n-d+1,i})^{S_i} \right) \times \prod_{j \in [M]\setminus\{i\}} (u_{\theta_j(i,n-d),j})^{S_j} - \mu_i \left(u_{n,i} - u_{n+1,i} \right),$$

for $i \in [M]$, $n = 1, 2, \ldots, K$, *and* $u \in \mathcal{Z}^M$*. The batch sizes follow the probability distribution* π*.*

The proof of Theorem 2 follows similar chain of argument as Theorem 1. For the sake of completeness, a brief sketch highlighting the main difference is outlined in Appendix B.

7 Discussion & Conclusions

In this work, we considered the problem of scheduling in large resource-sharing heterogeneous clusters with finite-buffer servers. We introduced the notion of a randomized Cost-Based Scheduling policy, where job assignment

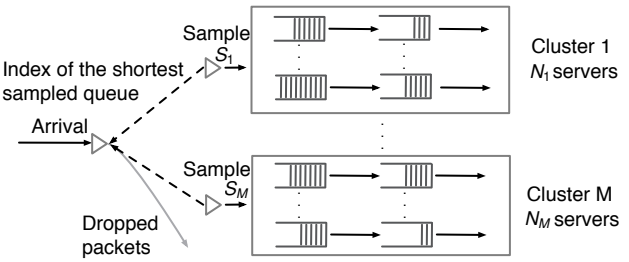

Fig. 8: The multi-stage system. Jobs receive multiple services sequentially at different non-interacting service stations.

is driven by general application-defined cost functions of the queue lengths rather than the queue lengths themselves as in JSQ. Traditional randomized scheduling policies, such as randomized JSQ, power of two or SQ(d) policies, are obtained as special cases of CBS by suitably choosing the cost functions.

For performance evaluation of CBS policies under a Markovian set-up, we can either compute exact marginal probabilities by solving the Kolmogorov equations or estimate them using Monte Carlo simulations. Unfortunately, both approaches are computationally expensive, and virtually infeasible when the number of servers is large. To this end, we proved a scaling limit, an autonomous set of ODEs, for the proportions of servers at each cluster with at least a fixed number of unfinished jobs. The scaling limit can be used in two ways: 1) it provides ready estimates of QoS, such as buffer filling proportions, given a CBS policy; and 2) it can be used as a tool to compare different CBS policies. In Section 5, we showed extensive numerical evaluations. In particular, we showed how queue length proportions and full server ratios can be computed from the scaling limit under different settings. In Section 6, we provided an extension of our results to the case of variable batch sizes.

One of our assumptions in the main model is that each of the M clusters contains a non-vanishing proportion of servers in the limit. We assume that M remains fixed as the sizes of the clusters grow. In some situations, the cluster system could be grown by adding new clusters of servers rather than increasing individual ones. In such a situation, $M \to \infty$ as the cluster system grows. We believe the mathematical tools exploited in this paper (limit theory of operator semigroups) could be used in a similar fashion for the purpose of performance evaluation, provided we define the stochastic processes in an appropriate way. Because we assume the servers within a cluster are homogeneous, the number M of clusters can be interpreted as a measure of the degree of heterogeneity of the N servers. By keeping M fixed, we are essentially maintaining the overall heterogeneity of the cluster system.

We note that our model can be further extended to the case where the jobs receive multiple services sequentially at different service stations as depicted in Figure 8. This occurs, for example, when employing load balancing between different service function chains in a network. There, arriving jobs are mapped to separate chains of servers carrying out consecutive tasks. These chains are assumed non-interacting for the sake of simplicity. From the perspective of an incoming job, the amount of time it spends in the cluster after it starts getting serviced in the first stage can be thought of as the service time of a single hypothetical server. This service time is precisely the difference between time of the start of service in the first stage and the time of departure from the cluster after completion of service in the last stage. Naturally, the probability distribution of the service time of this single, hypothetical server would not be exponential. Therefore, even if we draw the analogy between the multi-stage system and the hypothetical single server system, our results would not be immediately applicable because the hypothetical single-server system could be potentially non-Markovian. However, by approximating the cumulative service times by the closest exponential distribution, we can turn the system into a Markov system as desired. Therefore, we can then utilize the derived scaling limit for the performance evaluation of the system.

A potential limitation of our proposed extension to the batch system is that if adequate space is not available, a part of the batch could be lost. An alternative approach could be to assign the remaining part of the job to another available server. While this would entail a communication overhead, it is perhaps more profitable than our proposed strategy. We, however, do not have any analytic result quantifying performance boost. Another important subtlety is that we assume the job scheduling task is instantaneous. However, it is hardly so in reality. The rationale behind this assumption is that the delay in scheduling is usually so small compared to the order of magnitude of the inter-arrival times and the service times that it can be neglected without sacrificing much accuracy.

The explicit dependence of the limiting process in Theorem 1 on the function θ_j corresponding to the CBS policy is worth noting. These dependencies can be exploited in designing application-specific CBS policies. The cost functions themselves can then be tuned for performance in an optimal control setting. Thus, the scaling limits could potentially pave the way for design of near-optimal policies even for finite N.

APPENDIX A

In order to metrize the space $\mathcal{Z}^M$, define the metric

$$\rho_{\mathcal{Z}^M}(u,v) := \sup_{i\in[M]} \sup_{n=0,1,2,\ldots,K} \frac{|u_{n,i} - v_{n,i}|}{n+1}, \qquad (9)$$

for $u = (u^{(1)}, u^{(2)}, \ldots, u^{(M)}), v = (v^{(1)}, v^{(2)}, \ldots, v^{(M)}) \in \mathcal{Z}^M$, with $u^{(i)} = (u_{n,i} \mid n = 0, 1, 2, \ldots, K)$ and $v^{(i)} = (v_{n,i} \mid n = 0, 1, 2, \ldots, K)$ for each $i \in [M]$. Under ρ, the space $\mathcal{Z}^M$ turns complete, separable and compact [5], [11].

Now, let us define the generator A_N of the Markov process Z_N as

$$\begin{aligned} A_N f(u) :=& \lambda_N \sum_{i=1}^{M}\sum_{n=1}^{K} \left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \\ & \times \prod_{j\in[M]\setminus\{i\}} (u_{\theta_j(i,n-1),j})^{S_j} \\ & \times \left(f(u + \frac{1}{N_i} e_{n,i}) - f(u) \right) \\ & + \sum_{i=1}^{M}\sum_{n=1}^{K} N_i \mu_i \left(u_{n,i} - u_{n+1,i} \right) \\ & \times \left(f(u - \frac{1}{N_i} e_{n,i}) - f(u) \right), \end{aligned} \tag{10}$$

where $f : \times_{i\in[M]} \mathcal{Z}_i \to \mathbb{R}$, $u = \{(u^{(1)}, u^{(2)}, \ldots, u^{(M)}) \mid u^{(i)} = \{u_{n,i}\}_{n\in\{0,1,2,\ldots,K\}} \in \mathcal{Z}_i,\ i \in [M]\} \in \times_{i\in[M]} \mathcal{Z}_i$ represents a possible state of Z_N and $e_{n,i} := \{(a_1, a_2, \ldots, a_M) \mid a_j = \{\mathbb{1}(j = i, k = n)\}_{k\in\{0,1,2,\ldots,K\}},\ \forall j \in [M]\}$ tells us which component of Z_N changes. We set $u_{K+1,i} = 0$ for all $i \in [M]$. The generator defined in (10) is constructed by considering all the jumps of Z_N. The first part of (10) is due to admittance of a customer and the second part corresponds to departure of a customer after service. Consider the assignment of an arriving job to a server with exactly $n-1$ unfinished jobs in the i-th cluster when Z_N is in state $u \in \times_{i\in[M]} \mathcal{Z}_i$. This entails a jump from u to the state $u + e_{n,i}/N_i$. The term $\left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}\right) \prod_{j\in[M]\setminus\{i\}} (u_{\theta_j(i,n-1),j})^{S_j}$ gives the probability under CBS of an arriving job to be assigned to a server with exactly $n-1$ unfinished jobs in the ith-cluster. This happens only when the following two events take place.

1) At least one of the S_i sampled servers in the i-th cluster has exactly $n-1$ unfinished jobs and the others have at least n unfinished jobs.
2) In the light of A3, the fact that the dispatcher selects a server from the ith-cluster implies that *all* the sampled servers in the j-th cluster must have at least $\theta_j(i, n-1)$ unfinished jobs, for all $j \neq i$.

Now, the rate at which customers depart from servers having exactly n unfinished jobs in the i-th cluster is $N_i\mu_i\left(u_{n,i} - u_{n+1,i}\right)$, which explains the second part of (10).

Proof of Lemma 1. The proof follows by Picard's iterative technique. Following [5], [11], define the operators

$$\begin{aligned} &\mathbb{H}_{0,i}(u) := 0, \quad \forall i \in [M], \\ &\mathbb{H}_{n,i}(u) := \frac{\lambda}{\nu_i} \left(\zeta(u_{n-1,i})^{S_i} - \zeta(u_{n,i})^{S_i} \right) \\ &\quad \times \prod_{j\in[M]\setminus\{i\}} \zeta(u_{\theta_j(i,n-1),j})^{S_j} - \mu_i \left(\zeta(u_{n,i}) - \zeta(u_{n+1,i}) \right), \end{aligned} \tag{11}$$

where $\zeta(x) = \max(0, \min(x, 1))$. Consider the solutions of the integral equation

$$w(t) = w(0) + \int_0^t \mathbb{H}(w(s))\mathrm{d}s. \tag{12}$$

Note that the $\mathbb{H}(u)$ is defined for $u \in (\mathbb{R}^K)^M$. The operators $\mathbb{H}(u)$ and $\mathbb{F}(u)$ agree if $u \in \mathcal{Z}^M$. Therefore, the two systems (4) and (12) yield identical solutions in $\mathcal{Z}^M$. Moreover, if $w(0) = u \in \mathcal{Z}^M$, then the solution of the modified system (12) remains within $\mathcal{Z}^M$ (see [5] for similar arguments). In order to show uniqueness of solutions to (4) in $\mathcal{Z}^M$, it suffices to show that solutions to (12) are unique in $(\mathbb{R}^K)^M$. Therefore, we extend the norm ρ defined in (9) to $(\mathbb{R}^K)^M$. Following the same line of argument as in [5], we can find constants $a, b \in \mathbb{R}_+$ such that

$$\begin{aligned} \rho_{(\mathbb{R}^K)^M}\left(\mathbb{H}(u), \mathbb{H}(u)\right) \leq& a, \\ \rho_{(\mathbb{R}^K)^M}\left(\mathbb{H}(u), \mathbb{H}(v)\right) \leq& b\rho_{(\mathbb{R}^K)^M}\left(u, v\right). \end{aligned}$$

The uniqueness of the solution follows by virtue of the above, and using Picard's iterative approximation method, because the space $(\mathbb{R}^K)^M$ is complete under the metric defined in (9) (extended to $(\mathbb{R}^K)^M$). □

Proof of Lemma 2. The proof follows along the same line of argument as in [5, Lemma B.2] and [11, Lemma 3.2] if we set

$$a := \frac{2\lambda \sum_{i=1\in[M]} S_i}{\min_{i\in[M]} \nu_i} + 2 \max_{i\in[M]} \mu_i, \tag{13}$$

$$b := \sum_{i\in[M]} S_i + 2a. \tag{14}$$

Please note that the above bounds can be made tighter, but for our purposes, they suffice. □

Finally, we find the limit of the generators.

Lemma 3 (Convergence of the generators)**.** *Let $\mathcal{C} := \mathcal{C}(\mathcal{Z}^M)$ denote the space of all real-valued continuous functions defined on $\mathcal{Z}^M$. Consider the subspace $\mathcal{C}_D \subseteq \mathcal{C}$ of functions for which the partial derivatives*

$$\frac{\partial}{\partial u_{n,i}} z(t,u), \ \frac{\partial^2}{\partial u_{n,i}^2} z(t,u), \ \text{and} \ \frac{\partial^2}{\partial u_{n,j} u_{n,i}} z(t,u)$$

exist for all $u \in \mathcal{Z}^M$ and are uniformly bounded by some constant. Then, for all $f \in \mathcal{C}_D$,

$$\lim_{N\to\infty} A_N f(u) = \frac{\mathrm{d}}{\mathrm{d}t} f(z(t,u))\Big|_{t=0}, \tag{15}$$

where z is the solution of the integral equation (4)*.*

The proof of Lemma 3 is similar to the proof of [11, Theorem 2]. For the sake of completeness, it is provided here.

Proof of Lemma 3. Let $f \in \mathcal{C}_D$. Then, for each $i \in [M]$,

$$\begin{aligned} &\lim_{N_i\to\infty} N_i \left(f(u + \frac{1}{N_i} e_{n,i}) - f(u) \right) = \frac{\partial}{\partial u_{n,i}} f(u), \\ \text{and} \ &\lim_{N_i\to\infty} N_i \left(f(u - \frac{1}{N_i} e_{n,i}) - f(u) \right) = \frac{\partial}{\partial u_{n,i}} f(u), \end{aligned}$$

uniformly in $u \in \mathcal{Z}^M$. Thus, from (10), we have

$$\begin{aligned} A_N f(u) \to \sum_{i=1}^{M}\sum_{n=1}^{K} \Bigg[& \frac{\lambda}{\nu_i} \left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \\ & \times \prod_{j\in[M]\setminus\{i\}} (u_{\theta_j(i,n-1),j})^{S_j} \\ & - \mu_i \left(u_{n,i} - u_{n+1,i} \right) \Bigg] \frac{\partial}{\partial u_{n,i}} f(u), \end{aligned}$$

as $N \to \infty$ in the light of A1 and A2. The right hand side of the above equation is identical with $\frac{\mathrm{d}}{\mathrm{d}t} f(z(t,u))\Big|_{t=0}$, where z satisfies the integral equation (4) with $z(0) = u$. □

With the convergence of the generators in Lemma 3, we are now ready to prove convergence of the operator semigroup T_N.

Theorem 3 (Convergence of the operator semigroup)**.** *Under the CBS policy, for any $f \in \mathcal{C}_D$, and $t \in \mathcal{T}$, we have the following convergence of the operator semigroup $\{T_N(t)\}_{t\in\mathcal{T}}$,*

$$\lim_{N\to\infty} \sup_{u\in\times_{i\in[M]}\mathcal{Z}_i} |T_N(t)f(u) - T(t)f(u)| = 0, \tag{16}$$

where the limiting operator semigroup $\{T(t)\}_{t\in\mathcal{T}}$ is defined by $T(t)f(u) \coloneqq f(z(t,u))$, and is generated by the generator

$$Af(u) \coloneqq \lim_{h\to 0+} \frac{T(t+h)f(u) - T(t)f(u)}{h},$$

which is equivalent to $\frac{\mathrm{d}}{\mathrm{d}t} f(z(t,u))\Big|_{t=0}$ and where z is the solution of the integral equation (4)*.*

Proof of Theorem 3. First note that the space $\mathcal{C}_D$ is dense in $\mathcal{C}$. Also, both the semigroups $\{T_N(t)\}_{t\in\mathcal{T}}$ and $\{T(t)\}_{t\in\mathcal{T}}$ are strongly continuous and contracting [10]. Moreover, $\mathcal{C}_D$ is also a core of A . Therefore, following the same approach as [11, Theorem 2], [5], and by virtue of Lemma 3, we get the asserted convergence of the semigroups with the application of [10, Chapter 1, Theorem 6.1]. □

The proof of Theorem 1 now follows immediately.

Proof of Theorem 1. By virtue of Lemma 2, please note that continuity of $z(t,u)$ with respect to the initial condition u is ensured. Therefore, the limiting operator semigroup T is Feller. Now, having shown the convergence of the operator semigroup $\{T_N(t)\}_{t\in\mathcal{T}}$ to $\{T(t)\}_{t\in\mathcal{T}}$ in Theorem 3, the convergence of the Markov process Z_N follows immediately in the light of [10, Chapter 4, Theorem 2.11], as also noted in [5] also for the infinite-buffer case. □

APPENDIX B

We provide a brief sketch of the proof as it follows a similar line of argument as Theorem 1.

We use the same metric defined in (9) as before. In order to define the generator of the Markov process Z_N, we introduce the following notation: $e^{(d)}_{n,i} \coloneqq \{(a_1, a_2, \ldots, a_M) \mid a_j = \{\mathbb{1}(j = i, k \in \{n, n+1, \ldots, n+d-1\})\}_{k\in\{0,1,2,\ldots,K\}},\ \forall j \in [M]\}$, for $d = 1, 2, \ldots, D$. The quantity $e^{(d)}_{n,i}$ is a matrix the i-th column vector of which contains exactly d ones in rows $n, n+1$ to $n+d-1$. All other elements of $e^{(d)}_{n,i}$ are zeroes. Now, let us define the generator B_N of the Markov process Z_N as

$$\begin{aligned} B_N f(u) \coloneqq & \lambda_N \sum_{d=1}^{D}\sum_{i=1}^{M}\sum_{n=1}^{K} \pi(d) \left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}\right) \\ & \times \prod_{j\in[M]\setminus\{i\}} (u_{\theta_j(i,n-1),j})^{S_j} \\ & \times \left(f(u + \frac{1}{N_i} e^{(d)}_{n,i}) - f(u)\right) \\ & + \sum_{i=1}^{M}\sum_{n=1}^{K} N_i\mu_i \left(u_{n,i} - u_{n+1,i}\right) \\ & \times \left(f(u - \frac{1}{N_i} e_{n,i}) - f(u)\right), \end{aligned} \tag{17}$$

where, as before, $f : \times_{i\in[M]}\mathcal{Z}_i \to \mathbb{R}$, and $u = \{(u^{(1)}, u^{(2)}, \ldots, u^{(M)}) \mid u^{(i)} = \{u_{n,i}\}_{n\in\{0,1,2,\ldots,K\}} \in \mathcal{Z}_i,\ i \in [M]\} \in \times_{i\in[M]}\mathcal{Z}_i$ represents a possible state of Z_N. As before, we set $u_{K+1,i} = 0$ for all $i \in [M]$.

The generator defined in (17) is constructed by considering all the jumps of Z_N. The first part of (17) is due to admittance of a customer and the second part corresponds to departure of a customer after service. Consider the assignment of an arriving job of batch size d to a server with exactly $n-1$ unfinished jobs in the i-th cluster when Z_N is in state $u \in \times_{i\in[M]}\mathcal{Z}_i$. This entails a jump from u to the state $u + e^{(d)}_{n,i}/N_i$. The term $\left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}\right)\prod_{j\in[M]\setminus\{i\}}(u_{\theta_j(i,n-1),j})^{S_j}$ gives the probability under CBS of an arriving job to be assigned to a server with exactly $n-1$ unfinished jobs in the ith-cluster. The θ functions can be modified to depend on the batch size to reflect real-life application situations. Then, following the discussion in Appendix A, the rate of a transition from u to $u + e^{(d)}_{n,i}/N_i$ due to an admittance of an incoming job of buffer length d to a server with exactly $n-1$ unfinished jobs in the i-th cluster is given by $\lambda_N\pi(d)\left((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}\right)\prod_{j\in[M]\setminus\{i\}}(u_{\theta_j(i,n-1),j})^{S_j}$. The transition rates corresponding to departure of jobs remain the same as before.

Note that the uniqueness of the solution of the limiting differential equations (8) follows in practically the same way as before. Also, the solutions of the limiting differential equations (8) are smooth with respect to the initial conditions. In fact, by slightly modifying the constants, we can show that the partial derivatives

$$\frac{\partial}{\partial u_{n,i}} z(t,u),\ \frac{\partial^2}{\partial u^2_{n,i}} z(t,u),\ \text{ and } \frac{\partial^2}{\partial u_{n,j}u_{n,i}} z(t,u)$$

exist for all $u \in \mathcal{Z}^M$, and are uniformly bounded above by exponential bounds in a similar fashion as done in [5, Lemma B.2] and [11, Lemma 3.2].

We are now ready to give a brief sketch of the proof of Theorem 2.

Proof of Theorem 2. Similar to the proof of Theorem 1, we first show convergence of the generator B_N as $N \to \infty$. To this end, let the subspace $\mathcal{C}_D \subseteq \mathcal{C}$ of functions for which the partial derivatives

$$\frac{\partial}{\partial u_{n,i}} z(t,u),\ \frac{\partial^2}{\partial u^2_{n,i}} z(t,u),\ \text{ and } \frac{\partial^2}{\partial u_{n,j}u_{n,i}} z(t,u)$$

exist for all $u \in \mathcal{Z}^M$ and are uniformly bounded by some constant. Then, note that, for $f \in \mathcal{C}_D$ and for each $i \in [M]$,

$$\lim_{N_i\to\infty} N_i\left(f(u+\frac{1}{N_i}e^{(d)}_{n,i})-f(u)\right)=\sum_{l=1}^{d}\frac{\partial}{\partial u_{n,i+l-1}}f(u),$$

$$\text{and } \lim_{N_i\to\infty} N_i\left(f(u-\frac{1}{N_i}e_{n,i})-f(u)\right)=\frac{\partial}{\partial u_{n,i}}f(u),$$

uniformly in $u \in \mathcal{Z}^M$. Thus, from (17), we have

$$\begin{aligned}B_N f(u) \to \sum_{i=1}^{M}\sum_{n=1}^{K}\Bigg[&\sum_{d=1}^{\min\{n,D\}}\pi(d)\frac{\lambda}{\nu_i}\\ &\times\left((u_{n-d,i})^{S_i}-(u_{n-d+1,i})^{S_i}\right)\\ &\times\prod_{j\in[M]\setminus\{i\}}(u_{\theta_j(i,n-d),j})^{S_j}\\ &-\mu_i\left(u_{n,i}-u_{n+1,i}\right)\Bigg]\frac{\partial}{\partial u_{n,i}}f(u),\end{aligned}$$

as $N \to \infty$ in the light of A1 and A2. The right hand side of the above equation coincides with $Bf(u) := \frac{\mathrm{d}}{\mathrm{d}t}f(z(t,u))\Big|_{t=0}$, where z satisfies the integral equation (8) with $z(0) = u$.

Convergence of the generators B_N to the limiting generator B ensures the convergence of the corresponding operator semigroups T_N generated by B_N to the limiting operator semigroup T generated by B because $\mathcal{C}_D$ is a core. Moreover, the limiting operator semigroup T is Feller.

The weak convergence of the corresponding Markov process Z_N to the solution of the system of ODEs follows from the convergence of the convergence of the corresponding operator semigroups T_N to the limiting operator semigroup T generated by B in the light of [10, Chapter 4, Theorem 2.11]. This completes the proof. □

APPENDIX C

CBQA	Cost-Based Queue-Aware
CBS	Cost-Based Scheduling
CLT	Central Limit Theorem
ECMP	Equal-cost Multi-path routing
FIFO	First In First Out
JSQ	Join the Shortest Queue
LLN	Law of Large Numbers
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
QoS	Quality of Service
SRBM	Semi-martingale Reflecting Brownian Motion

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers. Their careful reading and constructive feedback improved the exposition of the paper significantly. This work has been funded by the German Research Foundation (DFG) as part of projects C3 and B4 within the Collaborative Research Centre (CRC) 1053 – MAKI. WKB is currently supported by the President's Postdoctoral Scholars Program (PPSP) of the Ohio State University. Computational facilities provided by the Lichtenberg - High Performance Computer at TU Darmstadt are gratefully acknowledged. We authors also acknowledge Florin Ciucu for helpful discussions.

REFERENCES

[1] W. Winston, "Optimality of the shortest line discipline," *Journal of Applied Probability*, vol. 14, no. 1, pp. 181–189, 1977.

[2] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," in *Proc. of IEEE INFOCOM*, 2013, pp. 872–880.

[3] G. Neglia, M. Sereno, and G. Bianchi, "Geographical load balancing across green datacenters: A mean field analysis," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 2, pp. 64–69, Sep. 2016.

[4] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 10 2001.

[5] A. Mukhopadhyay, A. Karthik, and R. R. Mazumdar, "Randomized assignment of jobs to servers in heterogeneous clusters of shared servers for low delay," *Stochastic Systems*, vol. 6, no. 1, pp. 90–131, 2016.

[6] G. Giannikis, G. Alonso, and D. Kossmann, "SharedDB: Killing One Thousand Queries with One Stone," *Proc. VLDB Endow.*, vol. 5, no. 6, pp. 526–537, Feb. 2012.

[7] Amazon AWS. Accessed: March 30, 2020 2:15 AM (EDT). [Online]. Available: https://aws.amazon.com/

[8] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. H'olzle, S. Stuart, and A. Vahdat, "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," in *Proc. of Sigcomm*, 2015.

[9] F. Ciucu, F. Poloczek, and A. Rizk, "Queue and loss distributions in finite-buffer queues," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 2, pp. 31:1–31:29, Jun. 2019. [Online]. Available: http://doi.acm.org/10.1145/3341617.3326146

[10] S. N. Ethier and T. G. Kurtz, *Markov Processes: Characterization and Convergence*, ser. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York, 1986, characterization and convergence.

[11] J. B. Martin and Y. M. Suhov, "Fast jackson networks," *Ann. Appl. Probab.*, vol. 9, no. 3, pp. 854–870, 08 1999.

[12] A. Mukhopadhyay and R. R. Mazumdar, "Analysis of randomized join-the-shortest-queue (jsq) schemes in large heterogeneous processor-sharing systems," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 116–126, 6 2016.

[13] 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*, vol. 32, no. 1, pp. 20–34, 1996.

[14] D. Mukherjee, S. C. Borst, J. S. van Leeuwaarden, and P. A. Whiting, "Asymptotic optimality of power-of-*d* load balancing in large-scale systems," *arXiv preprint arXiv:1612.00722*, 2016.

[15] A. S. Godtschalk and F. Ciucu, "Randomized load balancing in finite regimes," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 6 2016, pp. 580–589.

[16] M. van der Boor, S. C. Borst, J. S. van Leeuwaarden, and D. Mukherjee, "Scalable load balancing in networked systems: A survey of recent advances," *arXiv preprint arXiv:1806.05444*, 2018.

[17] G. Brightwell, M. Fairthorne, and M. J. Luczak, "The supermarket model with bounded queue lengths in equilibrium," *Journal of Statistical Physics*, vol. 173, no. 3, pp. 1149–1194, 2018. [Online]. Available: https://doi.org/10.1007/s10955-018-2044-7

[18] J. G. Dai and S. P. Meyn, "Stability and convergence of moments for multiclass queueing networks via fluid limit models," *IEEE Transactions on Automatic Control*, vol. 40, no. 11, pp. 1889–1904, Nov 1995.

[19] J. Dai and W. Dai, "A heavy traffic limit theorem for a class of open queueing networks with finite buffers," *Queueing Systems*, vol. 32, no. 1, pp. 5–40, Sep 1999.

[20] Q. Xie, X. Dong, Y. Lu, and R. Srikant, "Power of d choices for large-scale bin packing: A loss model," *SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 1, p. 321?334, Jun. 2015. [Online]. Available: https://doi.org/10.1145/2796314.2745849

[21] S. Banerjee and D. Mukherjee, "Join-the-shortest queue diffusion limit in halfin?whitt regime: Tail asymptotics and scaling of extrema," *The Annals of Applied Probability*, vol. 29, no. 2, pp. 1262–1309, 04 2019. [Online]. Available: https://doi.org/10.1214/18-AAP1436

[22] ——, "Join-the-shortest queue diffusion limit in halfin?whitt regime: Sensitivity on the heavy-traffic parameter," *The Annals of*

Applied Probability, vol. 30, no. 1, pp. 80–144, 02 2020. [Online]. Available: https://doi.org/10.1214/19-AAP1496

[23] R. Aghajani, X. Li, and K. Ramanan, "The pde method for the analysis of randomized load balancing networks," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 38:1–38:28, Dec. 2017. [Online]. Available: http://doi.acm.org/10.1145/3154497

[24] R. Aghajani and K. Ramanan, "The hydrodynamic limit of a randomized load balancing network," *arXiv preprint arXiv:1707.02005*, 2017.

[25] H. C. Gromoll, A. L. Puha, and R. J. Williams, "The fluid limit of a heavily loaded processor sharing queue," *The Annals of Applied Probability*, vol. 12, no. 3, pp. 797–859, 08 2002. [Online]. Available: https://doi.org/10.1214/aoap/1031863171

[26] H. C. Gromoll, P. Robert, and B. Zwart, "Fluid limits for processor-sharing queues with impatience," *Mathematics of Operations Research*, vol. 33, no. 2, pp. 375–402, 2008. [Online]. Available: https://doi.org/10.1287/moor.1070.0298

[27] L. Decreusefond and P. Moyal, "A functional central limit theorem for the m/gi/? queue," *The Annals of Applied Probability*, vol. 18, no. 6, pp. 2156–2178, 12 2008. [Online]. Available: https://doi.org/10.1214/08-AAP518

[28] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, no. 1, p. 180?200, Sep. 1999. [Online]. Available: https://doi.org/10.1137/S0097539795288490

[29] M. J. Luczak and C. McDiarmid, "On the power of two choices: Balls and bins in continuous time," *The Annals of Applied Probability*, vol. 15, no. 3, pp. 1733–1764, 08 2005. [Online]. Available: https://doi.org/10.1214/105051605000000205

[30] ——, "On the maximum queue length in the supermarket model," *The Annals of Probability*, vol. 34, no. 2, pp. 493–527, 03 2006. [Online]. Available: https://doi.org/10.1214/00911790500000710

[31] H. Perros, "A bibliography of papers on queueing networks with finite capacity queues," *Performance Evaluation*, vol. 10, no. 3, pp. 255 – 260, 1989, queueing Networks with Finite Capacity Queues.

[32] S. Balsamo, V. D. N. Personé, and P. Inverardi, "A review on queueing network models with finite capacity queues for software architectures performance prediction," *Performance Evaluation*, vol. 51, no. 2, pp. 269 – 288, 2003, queueing Networks with Blocking.

[33] H. S. Kim and N. B. Shroff, "Loss probability calculations and asymptotic analysis for finite buffer multiplexers," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 755–768, 12 2001.

[34] C. Osorio and M. Bierlaire, "An analytic finite capacity queueing network model capturing the propagation of congestion and blocking," *European Journal of Operational Research*, vol. 196, no. 3, pp. 996 – 1007, 2009.

[35] X. Shen, H. Chen, J. Dai, and W. Dai, "The finite element method for computing the stationary distribution of an srbm in a hypercube with applications to finite buffer queueing networks," *Queueing Systems*, vol. 42, no. 1, pp. 33–62, 9 2002.

[36] S. B. Gershwin, "An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking," *Operations Research*, vol. 35, no. 2, pp. 291–305, 1987.

[37] R. Sadre, B. R. Haverkort, and A. Ost, "An efficient and accurate decomposition method for open finite- and infinite-buffer queueing networks," in *Proceedings of the Third International Workshop on Numerical Solution of Markov Chains*, 1999, pp. 1–20.

[38] S. S. Kunniyur and R. Srikant, "An adaptive virtual queue (avq) algorithm for active queue management," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 286–299, April 2004.

[39] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on networking*, vol. 1, no. 4, pp. 397–413, 1993.

[40] K. Ito and F. Kappel, *Evolution Equations And Approximations*. WORLD SCIENTIFIC, 2002. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/4990

[41] K.-J. Engel and R. Nagel, *One-parameter Semigroups for Linear Evolution Equations*. Springer Science & Business Media, 1999, vol. 194.

[42] P. Billingsley, *Convergence of Probability Measures*, 2nd ed., ser. Wiley Series in Probability and Statistics: Probability and Statistics. John Wiley & Sons, Inc., New York, 1999, a Wiley-Interscience Publication.

Wasiur R. KhudaBukhsh is currently a Postdoctoral Research Fellow at the Mathematical Biosciences Institute, the Ohio State University. He obtained his PhD from the Technische Universität Darmstadt in Germany under the supervision of Heinz Koeppl. Prior to joining the PhD programme, he received his Master of Statistics (M.Stat.) degree from the Indian Statistical Institute, Kolkata (India). He is interested in applications of probability theory and statistics to various problems arising from computer science, biology and other domains of science.

Sounak Kar received his Master of Statistics (M.Stat.) as well as Bachelor of Statistics (B.Stat.) degrees from the Indian Statistical Institute, Kolkata (India) in 2010 and 2008 respectively. He is now a PhD student in the department of electrical engineering and information technology, Technische Universität Darmstadt in Germany since April 2017. He is interested in queuing theory and applications of probability theory in various domains of computer science.

Bastian Alt received his B.Sc. and M.Sc. degrees in Electrical Engineering and Information Technology from Technische Universität Darmstadt in November 2013 and December 2016, respectively. He joined the Bioinspired Communication Systems Lab as a PhD student in January 2017. He is interested in the modelling of networking systems, in order to infer and predict system behaviour to do control.

PLACE PHOTO HERE

Amr Rizk received the doctoral degree (Dr.-Ing.) from the Leibniz Universität Hannover, Germany, in 2013. He has been a Post-Doctoral Research Fellow with the Department of Computer Science, University of Warwick, U.K., in 2014, and the University of Massachusetts, Amherst, in 2015, where he received a stipend from the German Academic Exchange Service for his research stay. He is currently a Post-Doctoral Research Group Head with the Multimedia Communications Laboratory, TU Darmstadt, Germany. He was a recipient of the Excellence in DASH Award at ACM MMSyst 2016. His research interests are in the areas of performance evaluation of communication networks, stochastic modeling, software-defined networks, and teletraffic theory.

Heinz Koeppl received his M.Sc. in physics (2001) from Karl-Franzens University Graz and his Ph.D. in electrical engineering (2004) from Graz University of Technology, Austria. After that he held postdoctoral positions at UC Berkeley and Ecole Polytechnique Federale de Lausanne (EPFL). From 2010 to 2014 he was an assistant professor at the ETH Zurich and part-time group leader at IBM Research Zurich, Switzerland. Since 2014 he is a full professor at the department of electrical engineering and information technology at Technische Universität Darmstadt, Germany. He received two awards for his Ph.D. thesis, the Erwin Schrödinger fellowship, the SNSF professorship award and currently holds an ERC consolidator grant. He is interested in stochastic models and their inference in applications ranging from communication networks to cell biology.