Noname manuscript No.
(will be inserted by the editor)

On the Impact of Job Size Variability on Heterogeneity-aware Load Balancing

Ignace Van Spilbeeck and Benny Van Houdt

the date of receipt and acceptance should be inserted later

Abstract Load balancing is one of the key components in many distributed systems as it heavily impacts performance and resource utilization. We consider a heterogeneous system where each server belongs to one of K classes and the speed of the server depends on its class. Two types of load balancing strategies are considered: arriving jobs are either immediately dispatched to a server class in a randomized manner, i.e., with probability p_k a job is assigned to class k, or are dispatched based on their size, i.e., jobs with a size in $[T_{k-1}, T_k)$ are assigned to class k. Within each class a power of d choices rule is used to select the server that executes the job.

For large systems and exponential job size durations the optimal probabilities p_k to minimize the mean response time can be determined easily via convex optimization. In this paper we develop a mean field model (validated by simulation) to investigate how the optimal probabilities p_k are affected by the higher moments and in particular by the variability of the job size distribution when the service discipline at each server is first-come-first-served. In addition, we make use of the cavity method to study the optimal thresholds T_k in case the dispatching is based on the job size.

1 Introduction

Consider a large distributed system consisting of N servers and a (number of) centralized dispatchers. Incoming jobs are assigned by the dispatcher(s) to the servers using a load balancing (LB) scheme. A very efficient manner to distribute the incoming jobs among the servers is to rely on a pure randomized assignment scheme or some form of round robin. While this allows very fast load balancing decisions, the resulting performance is known to be inferior to LB schemes that exploit information concerning the current system state, such as the queue lengths

Department of Mathematics and Computer Science
University of Antwerp - imec
Middelheimlaan 1, B-2020 Antwerp, Belgium
{ignace.vanspilbeeck,benny.vanhoudt}@uantwerpen.be

or server speeds. Examples of the latter include join-the-shortest-queue (JSQ) LB [16] or the power-of-d-choices (POD) LB [24,21]. Under JSQ incoming jobs are assigned to the server containing the least number of jobs, while under POD d servers are selected uniformly at random and the job is assigned to the server with the shortest queue length among the d selected servers.

When the system is heterogeneous, for instance when not all the servers have the same speed, the choice of the LB scheme becomes even more critical as LB schemes based on joining the server with the least number of jobs among a set of randomly selected servers may lead to system instability even if the total offered load is (well) below the total service rate of the system [6]. A manner to avoid instability when the servers have different speeds (and the overall load is below 1), exists in assigning jobs to servers based on the server speeds [13]. While such an assignment becomes necessary as the overall load tends to one, it is clearly suboptimal under low and medium loads as the mean response time can be reduced by assigning a larger fraction of the jobs to the faster servers. In case of Poisson arrivals, processor sharing (PS) servers and random job routing (that is, server i is selected with a fixed probability p_i) explicit expressions can be derived for the routing probabilities that minimize the mean response time [3,13]. Under first-come-first-served (FCFS) service and more complex LB schemes determining the optimal fraction of the incoming jobs that needs to be assigned to each of the servers is much harder.

In this paper we consider a system consisting of N servers where jobs arrive according to a Poisson process with rate λN with $\lambda < 1$. The servers are partitioned into K classes of homogeneous servers, process their jobs in FCFS order and have an infinite waiting room. By considering FCFS service, we are considering a setting where jobs are very expensive to preempt and are therefore typically run-to-completion without interruption (such as in supercomputing centers, see [17]). Servers belonging to class k serve jobs at rate μ_k. We consider two dispatching strategies: incoming jobs are either assigned to a class in a randomized manner or based on their size. In the randomized case a job is assigned to a class k server with probability p_k. In the size based case we define a set of thresholds T_k (with $T_0 = 0$ and $T_K = \infty$) and assign an incoming job of size x to class k if $x \in [T_{k-1}, T_k)$. This job size based dispatching rule is known as Size Interval Task Assignment (SITA) [18]. The main idea of SITA is to reduce the variability of job sizes at a server, which reduces the mean response time. Once a job is assigned to a server class, the server that executes the job within class k is selected using POD LB. In other words, a set of d servers is selected among the class k servers and the jobs is assigned to the server holding the least number of jobs among the d selected class k servers.

Note that the above randomized setting is identical to Scheme 3 presented in [22], except that our servers operate under FCFS instead of PS. For exponential job durations the queue length distribution under FCFS and PS is the same and under PS the system is believed to become insensitive to the job size distribution as the system size N tends to infinity [7,8]. Under FCFS the mean response time remains sensitive to the job size distribution as N tends to infinity. The main objective of this paper is to see how the probabilities p_k and thresholds T_k that minimize the mean response time in a large system, are effected by the variability of the job size distribution and more importantly whether these optimized values reduce the mean response time significantly compared to more more basic suboptimal manners to select p_k and T_k. To answer these questions we develop a mean field model for the randomized case, the accuracy of which is validated using simulation, and rely on the cavity method [7] for the job size based case. Some of our main insights are that neglecting the variability of the job size distribution when optimizing the probabilities p_k does not result in a substantial loss in performance and the reduction in the mean response time offered by using job size information decreases as the system becomes more heterogeneous.

The paper is structured as follows. In Section 2 we introduce the model under consideration. Related work is discussed in 3. The mean field model and numerical examples for the randomized dispatching are presented in 4 and 5. Finally, Sections 6 and 7 focus on the size based dispatching policies.

2 The model

Consider a system of N servers belonging to K classes operating under FCFS. There are N_k servers of class k and let $\gamma_k = N_k/N$ such that $\sum_{k=1}^{K} \gamma_k = 1$. All servers have an infinite waiting room and the speed of a class k server is denoted as μ_k. The server speeds are such that $\sum_{k=1}^{K} \gamma_k \mu_k = 1$, meaning the average speed of a server is equal to 1. Incoming jobs arrive at one or multiple dispatchers as a Poisson process with an overall rate λN and are immediately forwarded to one of the N servers. To select a server the dispatcher first selects a server class k and subsequently assigns the job to the server with the least number of jobs among a set of d servers selected uniformly at random among the class k servers. The server class k is either selected in a randomized manner or based on the job size x of the incoming job. In the randomized case we use a set of probabilities p_k (with $\sum_k p_k = 1$) and class k is selected with probability p_k. In the job sized based case we use a set of thresholds T_k (with $0 = T_0 < T_1 < \ldots < T_K = \infty$) and class k is selected if $x \in [T_{k-1}, T_k)$.

The job size distribution is assumed to follow a phase-type distribution [20] with mean 1 characterized by (α, S), where α is a stochastic vector and S a subgenerator matrix such that $\alpha e^{Sx} e$ is the probability that the job size exceeds x, where e is a column vector of ones. The time to execute a job on a class k server is therefore phase-type distributed with parameters $(\alpha, \mu_k S)$ as it suffices to scale time by a factor μ_k. We note that the class of phase-type distributions is dense in the field of all positive-valued distributions. As such any positive-valued distribution can be approximated arbitrarily close by a phase-type distribution. Various fitting tools for phase-type distributions are also available online (e.g., jPhase [23], ProFiDo [5] or BuTools).

Note that due to the Poisson arrivals, the system under consideration behaves as a set of K independent homogeneous LB systems. In the randomized case the k-th system has load $\rho_k = \lambda p_k / (\gamma_k \mu_k)$ (as the total arrival rate is λN and with probability p_k the job is assigned to one of the $\gamma_k N$ class k servers). In the job size based case the k-th system has load

$$\rho_k = \lambda P[T_{k-1} \leq X < T_k] E[X | T_{k-1} \leq X < T_k] / (\gamma_k \mu_k),$$

where X is the job size distribution. Due to the phase-type assumption for the job sizes we have $P[T_{k-1} \leq X < T_k] = \alpha(e^{ST_{k-1}} - e^{ST_k})e$ and

$$E[X|T_{k-1} \leq X < T_k] = \\ T_{k-1} + \frac{\alpha(-S)^{-1}(e^{ST_{k-1}} - e^{ST_k})e - (T_k - T_{k-1})\alpha e^{ST_k} e}{\alpha(e^{ST_{k-1}} - e^{ST_k})e}.$$

For exponential job sizes the probabilities p_k for large N can be optimized by relying on the explicit formula for the mean response time in a homogeneous system derived in [24,21], that is, the probability that a server contains i or more jobs converges to $\rho_k^{\frac{d^i-1}{d-1}}$ as N tends to infinity under POD LB with exponential job sizes and load ρ_k. This results (by applying Little's law to the complete N-th system and taking limits) in the following convex optimization problem that can be solved numerically without much effort:

$$\begin{aligned} \underset{p_k}{\text{minimize}} \quad & f(p_1, \ldots, p_K) = \frac{1}{\lambda} \sum_k \gamma_k \sum_{i \geq 1} \rho_k^{\frac{d^i-1}{d-1}}. \\ \text{subject to} \quad & 0 \leq \rho_k < 1; k = 1, \ldots, K, \\ & \sum_k \gamma_k \rho_k = \lambda. \end{aligned} \tag{1}$$

Note that the first set of constraints demands that each of the K subsystems is stable, while the second constraint demands that the total assigned workload matches the incoming workload. For $K = 2$ the first set of constraints can be restated as $1 - \frac{\gamma_2 \mu_2}{\lambda} < p_1 < \frac{\gamma_1 \mu_1}{\lambda}$ (as $p_2 = 1 - p_1$). One of the main objectives of this paper is to study the equivalent optimization problem for phase-type distributed job lengths.

Finding the optimal thresholds T_k for a SITA policy is very challenging and different heuristics to do so have been proposed in [9,19].

3 Related work

A closely related paper for the randomized case is [22] which proposes mean field models for three LB schemes: the optimal randomized, $SQ(d)$ and hybrid $SQ(d)$ LB. The hybrid $SQ(d)$ LB scheme, which was shown to outperform the other two, is identical to the LB scheme considered in this paper except that [22] considers PS servers and exponential job sizes. Evidence that the $SQ(d)$ scheme becomes insensitive to the job size distribution was provided using simulation experiments, while evidence[1] for the asymptotic insensitivity for the hybrid $SQ(d)$ LB under PS was presented in [7,8].

Two other LB schemes for heterogeneous networks were proposed in [1]. In both LB schemes a server is chosen by first selecting d_k servers of type k at random for all k and then by selecting one of the servers among the selected $\sum_k d_k$ servers based on the queue length information only (scheme 1) or on the queue length and server speeds (scheme 2). While Figures 3 and 4 in [1] suggest that these schemes

[1] The asymptotic insensitivity under PS was proven given the ansatz of asymptotic independence of the queue length for any finite subset of queues.

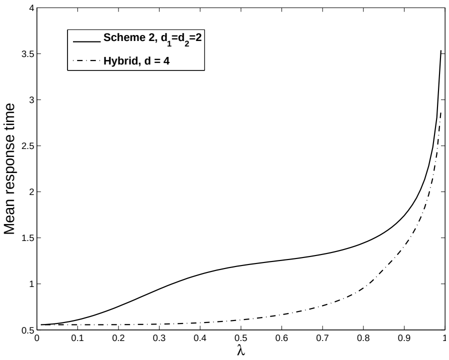


Fig. 1: Mean response time (for exponential job sizes) as a function of λ for Scheme 2 of [1] with $d_1 = d_2 = 2$ and for the hybrid $SQ(4)$ scheme when $\gamma_1 = \gamma_2 = 1/2$ and $\mu_1 = 9\mu_2$.

may outperform the hybrid $SQ(d)$ scheme in some cases, the hybrid $SQ(2)$ scheme uses the queue length information of 2 servers per incoming job, while the other two LB schemes use the queue length information of 4 queues per job. Figure 1 indicates that if we also allow 4 choices for the hybrid $SQ(d)$ scheme, the optimal hybrid $SQ(d)$ scheme outperforms scheme 2 of [1] for all arrival rates λ. Another LB scheme, called HALO_POD, that uses a POD rule in a heterogeneous PS network was proposed in [13]. In this scheme a job is assigned to the shortest of d selected servers, where a class k server is selected based on the optimal routing probabilities of a pure randomized LB scheme (first derived in [3]).

In this paper we assess the mean response time in a heterogeneous FCFS LB network with phase-type distributed job sizes under randomized and size based dispatching. Another approach to analyze such a network exists in numerically determining a fixed point of a so-called hydrodynamical PDE presented in [2] as our network is equivalent to a set of K independent homogeneous FCFS networks. In fact, this is the approach that we initially used, but finding the optimal probabilities p_k by repeatedly solving a hydrodynamical PDE turns out to be much more time and memory consuming than the approach taken in this paper (due to the required size of the mesh used by the numerical scheme).

Considerable work has also been done on SITA policies (e.g., [18,19,4,12]). The main different between our work and these prior studies is that we only use SITA to select the server class k and use POD LB within each class, while prior work selected individual servers using SITA. If SITA is used to select individual servers, each server corresponds to an M/G/1 queue. This is no longer the case if SITA is only used to select the server class k (unless $d = 1$, that is, we assign the job to a random class k server), which further complicates the analysis.

4 A mean field model for randomized dispatching

Let $X^{(N)}_{k,j,i}(t)$ be the number of type $k \in \{1, \dots, K\}$ servers with $i > 0$ or more jobs that are in service phase $j \in \{1, \dots, J\}$ at time t. Define $Z^{(N)}_{k,j,i}(t) = X^{(N)}_{k,j,i}(t)/N_k$

as its scaled version. We would like to study $\lim_{t\to\infty} Z_{k,j,i}^{(N)}(t)$ for large N. For this purpose we introduce a mean field model in Section 4.1 for which we provide theoretical and numerical support in Sections 4.2 and 4.3

4.1 System dynamics

Assume (α, S) is an order J phase-type distribution. The mean field model uses the variables $s_{k,j,i}(t)$ with $i > 0$, $1 \leq j \leq J$ and $1 \leq k \leq K$, that represent the fraction of servers that are of type k, contain i or more jobs and are in service phase j at time t. Let $z_{k,j,i}(t) = s_{k,j,i}(t)/\gamma_k$, denote $\sigma_{i,j}$ as the (i,j)-th entry of the matrix S and let $\nu_j = (-Se)_j$. The evolution of $z_{k,j,i}(t)$ is described by the following set of ODEs, the intuition behind this set of ODEs is presented below:

$$\begin{aligned}\frac{dz_{k,j,1}(t)}{dt} &= \frac{\lambda p_k}{\gamma_k}(1 - z_{k,1}^d(t))\alpha_j - \mu_k\nu_j(z_{k,j,1}(t) - z_{k,j,2}(t)) \\ &\quad - \mu_k\nu_j(1-\alpha_j)z_{k,j,2}(t) + \sum_{j'\neq j} \mu_k\nu_{j'} z_{k,j',2}(t)\alpha_j \\ &\quad + \sum_{j'\neq j} z_{k,j',1}(t)\mu_k\sigma_{j',j} - z_{k,j,1}(t)\sum_{j'\neq j}\mu_k\sigma_{j,j'} \\ &= \frac{\lambda p_k}{\gamma_k}(1 - z_{k,1}^d(t))\alpha_j - \mu_k z_{k,j,1}(t)\nu_j + \mu_k\sum_{j'=1}^{J} z_{k,j',2}(t)\nu_{j'}\alpha_j \\ &\quad + \mu_k\sum_{j'=1}^{J} z_{k,j',1}(t)\sigma_{j',j} - \mu_k z_{k,j,1}(t)\sum_{j'=1}^{J}\sigma_{j,j'}, \end{aligned} \tag{2}$$

where $z_{k,i}(t) = \sum_{j=1}^{J} z_{k,j,i}(t)$ and

$$\begin{aligned}\frac{dz_{k,j,i}(t)}{dt} &= \frac{\lambda p_k}{\gamma_k}\frac{z_{k,j,i-1}(t) - z_{k,j,i}(t)}{z_{k,i-1}(t) - z_{k,i}(t)}(z_{k,i-1}^d(t) - z_{k,i}^d(t)) \\ &\quad - \mu_k\nu_j(z_{k,j,i}(t) - z_{k,j,i+1}(t)) - \mu_k\nu_j(1-\alpha_j)z_{k,j,i+1}(t) \\ &\quad + \sum_{j'\neq j}\mu_k\nu_{j'}z_{k,j',i+1}(t)\alpha_j + \sum_{j'\neq j} z_{k,j',i}(t)\mu_k\sigma_{j',j} - z_{k,j,i}(t)\sum_{j'\neq j}\mu_k\sigma_{j,j'}, \end{aligned} \tag{3}$$

for $i > 1$. For $i = 1$ the intuition is as follows. The arrival rate in a class k server is $\lambda p_k/\gamma_k$ and $z_{k,j,1}(t)$ increases when not all of the d selected servers are busy (probability $(1 - z_{k,1}^d(t))$) and service starts in phase j (probability α_j). For $i > 1$, $z_{k,j,i}(t)$ increases when all d selected servers have at least $i-1$ jobs and not all have i jobs or more, this is represented by the probability $(z_{k,i-1}^d(t) - z_{k,i}^d(t))$. The server that gets the job has to be in phase j, which is represented by the probability $\frac{z_{k,j,i-1}(t)-z_{k,j,i}(t)}{z_{k,i-1}(t)-z_{k,i}(t)}$.

For $i \geq 1$, $z_{k,j,i}(t)$ decreases when a job completion occurs in a class k server with exactly i jobs that is in phase j (with rate $\mu_k\nu_j$). It also decreases when a server in phase j with at least $i+1$ jobs has a job completion and starts processing the next job in phase $j' \neq j$ (with rate $\mu_k\nu_j(1-\alpha_j)$) or a server with at least i jobs changes its phase from j to $j' \neq j$ (with rate $\mu_k\sigma_{j,j'}$). Finally, $z_{k,j,i}(t)$ increases when a server in phase $j' \neq j$ with $i+1$ or more jobs completes a job and start processing the next job in phase j (with rate $\mu_k \nu_{j'} \alpha_j$) or a server with at least i jobs changes its phase from $j' \neq j$ to j (with rate $\mu_k \sigma_{j',j}$).

Numerical evaluation: The queue length distribution of the mean field model, characterized by (2-3), is determined via a forward Euler iteration. More specifically, we start with an empty system at time $t=0$, i.e., set $z_{k,j,i}(0)=0$ for all k, j and $i>0$, and compute

$$z_{k,j,i}(t+\delta t) = z_{k,j,i}(t) + \delta t \frac{dz_{k,j,i}(t)}{dt},$$

with a step size δt that is sufficiently small. This iteration is repeated until a fixed point π is found, i.e., until $dz_{k,j,i}(t)/dt \leq \epsilon$ for ϵ small (e.g., $\epsilon = 10^{-9}$). The mean response time is subsequently determined via Little's law. We did not encounter any numerical issues when computing a fixed point using this simple Euler iteration, as such there was no need to rely on more advanced Runge-Kutta methods.

Asymptotic sensitivity: We end this subsection by showing that any fixed point π of the set of ODEs is sensitive to the higher moments of the job size distribution (as opposed to the system with PS service). Summing (2-3) over i and j yields

$$\sum_{i\geq 1} dz_{k,i}(t)/dt = \mu_k(\rho_k - \sum_{j'} z_{k,j',1}\nu_{j'}).$$

Let ν be the column vector with its j-th entry equal to ν_j and denote β_j as the j-th entry of the unique row vector β for which $\beta(S+\nu\alpha)=0$ and $\sum_j \beta_j = 1$ holds. It is easy to check that $\beta = \alpha(-S)^{-1}$ and therefore $1/(\beta\nu)$ is the mean job duration. If we now assume asymptotic insensitivity, that is, $\pi_{k,j,i}$ can be written as $\pi_{k,i}\beta_j$, where $\pi_{k,i}$ is the fixed point of the set of ODEs in case of exponential job sizes with load ρ_k, then (3) implies

$$\begin{aligned} 0 &= \frac{\lambda p_k}{\gamma_k}\beta_j(\pi^d_{k,i-1} - \pi^d_{k,i}) + \mu_k(\pi_{k,i}(\beta S)_j + \pi_{k,i+1}(\beta\nu)\alpha_j) \\ &= \frac{\lambda p_k}{\gamma_k}\beta_j(\pi^d_{k,i-1} - \pi^d_{k,i}) - \mu_k(\pi_{k,i} - \pi_{k,i+1})(\beta\nu)\alpha_j, \end{aligned} \tag{4}$$

with $\beta\nu = 1$. As $\pi_{k,i}$ is the fixed point of the set of ODEs in case of exponential job sizes with load ρ_k, we have

$$0 = \frac{\lambda p_k}{\gamma_k}(\pi^d_{k,i-1} - \pi^d_{k,i}) - \mu_k(\pi_{k,i} - \pi_{k,i+1}). \tag{5}$$

Hence, (4) holds if and only if $\beta_j = \alpha_j$ for all $j \in \{1, \ldots, J\}$. However, when $\beta = \alpha$ one finds (using $\alpha(S+\nu\alpha) = \beta(S+\nu\alpha) = 0$) that the probability $\alpha e^{Sx} e$ that the job size exceeds x can be written as

$$\alpha e^{Sx} e = \sum_{s=0}^{\infty} \alpha S^s e x^s/s! = \sum_{s=0}^{\infty} (-\beta\nu)^s x^s/s! = e^{-\beta\nu x},$$

meaning the job sizes are exponential with mean $1/(\beta\nu)$. Thus for any phase-type distribution that is not a redundant[2] representation of the exponential distribution $\pi_{k,i}\beta_j$ is not a fixed point (which would be the case if the system was asymptotically insensitive as in the PS service case).

4.2 Theoretical support

Let $\mathcal{J} = \{1,\ldots,J\}$ and denote the set of ODEs given by (2-3) as $dz_{k,j,i}(t)/dt = F_{k,j,i}(\mathbf{z}_k(t))$, where $\mathbf{z}_k(t) = (\mathbf{z}_{k,1}(t),\mathbf{z}_{k,2}(t),\ldots)$ and $\mathbf{z}_{k,i}(t) = (z_{k,1,i}(t),\ldots,z_{k,J,i}(t))$. Define the space $E^J = \{(x_{j,i})_{j\in\mathcal{J},i\geq 1} | 1 \geq x_{j,1} \geq x_{j,2} \geq \ldots \geq 0; 1 \geq \sum_{j\in\mathcal{J}} x_{j,1}\}$. Let w be the metric defined on E^J by setting

$$w(\mathbf{x},\mathbf{y}) = \sup_{j\in\mathcal{J}} \sup_{i\geq 1} \frac{|x_{j,i}-y_{j,i}|}{(i+1)^2}.$$

Proposition 1 *(E^J,w) is a compact metric space.*

Proof By Tychonoff's theorem any sequence $(\mathbf{x}_n)_n$ in E^J has a subsequence $(\mathbf{x}_{n_m})_m$ that converges pointwise to some limit $\mathbf{x}^* \in E^J$. We argue that this subsequence also converges to $\mathbf{x}^*$ under the metric w which proves compactness. For any i we can pick m' large enough such that for $m \geq m'$ we have

$$\sup_{j\in\mathcal{J}} \frac{|(x_{n_m})_{j,i'} - x^*_{j,i'}|}{(i'+1)^2} \leq 1/(i+1)^2,$$

for $1 \leq i' \leq i$ due to the pointwise convergence. Further

$$\sup_{j\in\mathcal{J}} \frac{|(x_{n_m})_{j,i'} - x^*_{j,i'}|}{(i'+1)^2} \leq 1/(i'+1)^2 < 1/(i+1)^2,$$

for any m when $i < i'$ as $|(x_{n_m})_{j,i'} - x^*_{j,i'}| \leq 1$. □

The next proposition shows that $\mathbf{F}_k(\mathbf{x}) : E^J \to E^J$ is Lipschitz, that is, there exists a constant L_k such that $w(\mathbf{F}_k(\mathbf{x}),\mathbf{F}_k(\mathbf{y})) \leq L_k w(\mathbf{x},\mathbf{y})$. As (E^J,w) is compact it is a Banach space and the Lipschitz property implies that the set of ODEs (2-3) has a unique solution $z_{k,j,i}(t)$ for any given initial state $(z_{k,j,i}(0))_{j\in\mathcal{J},i\geq 1} \in E^J$ and this solution is continuous in t and the initial state.

Proposition 2 *$F_k(\mathbf{x})$ is Lipschitz with constant $L_k = 3J\mu_k \max_j(-\sigma_{j,j}) + \lambda p_k(2 + dJ + 2Jd^2)/\gamma_k$ on (E^J,w).*

[2] Redundant representations are order J phase-type distributions (α,S) that can be represented by a phase-type distribution of a smaller order. For instance, any order $J > 1$ phase type distribution with S equal to minus the identity matrix is a redundant representation of the exponential distribution with mean one.

Proof We make repeated use of the inequality $|a_1^{m_1}a_2^{m_2} - b_1^{m_1}b_2^{m_2}| \leq m_1|a_1 - b_1| + m_2|a_2 - b_2|$ for $0 \leq a_1, a_2, b_1, b_2 \leq 1$ and $m_1, m_2 \in \{1, 2, \ldots\}$. Due to (2-3) one finds

$$w(\mathbf{F}_k(\mathbf{x}),\mathbf{F}_k(\mathbf{y})) \leq 3J\mu_k \max_j(-\sigma_{j,j})w(\mathbf{x},\mathbf{y}) + \frac{\lambda p_k}{\gamma_k}dJw(\mathbf{x},\mathbf{y}) + \frac{\lambda p_k}{\gamma_k}2w(\mathbf{x},\mathbf{y})$$
$$+ \frac{\lambda p_k}{\gamma_k}\sup_{i>1}\frac{1}{i+1}\left|\frac{x_{k,i-1}^d - x_{k,i}^d}{x_{k,i-1} - x_{k,i}} - \frac{y_{k,i-1}^d - y_{k,i}^d}{y_{k,i-1} - y_{k,i}}\right|$$

As $(x_{k,i-1}^d - x_{k,i}^d)/(x_{k,i-1} - x_{k,i}) = \sum_{m=0}^{d-1} x_{k,i-1}^m x_{k,i}^{d-1-m}$ we get

$$\sup_{i>1}\frac{1}{i+1}\left|\frac{x_{k,i-1}^d - x_{k,i}^d}{x_{k,i-1} - x_{k,i}} - \frac{y_{k,i-1}^d - y_{k,i}^d}{y_{k,i-1} - y_{k,i}}\right|$$
$$\leq \sup_{i>1}\sum_{m=0}^{d-1}\frac{|x_{k,i-1}^m x_{k,i}^{d-1-m} - y_{k,i-1}^m y_{k,i}^{d-1-m}|}{(i+1)^2}$$
$$\leq 2d^2 \sup_{i>1}\frac{|x_{k,i} - y_{k,i}|}{i+1} \leq 2Jd^2 w(\mathbf{x},\mathbf{y}).$$

□

Let $\bar{E}^J = \{(x_{j,i})_{j\in\mathcal{J},i\geq 1} \in E^J | \sum_{i>0}\sum_{j=1}^J x_{j,i} < \infty\}$, then we have the following result:

Theorem 1 *Let* $\mathbf{x}(0) \in \bar{E}^J$ *and assume* $\lim_{N\to\infty} Z_{k,j,i}^{(N)}(0) = x_{j,i}(0)$, *then*

$$\lim_{N\to\infty}\sup_{u\leq t}\sup_{j\in\mathcal{J}}\sup_{i\geq 1}\frac{|Z_{k,j,i}^{(N)}(u) - z_{k,j,i}(u)|}{(i+1)^2} = 0 \qquad a.s.,$$

for any fixed t*, where* $\mathbf{z}(u)$ *is the unique solution of the set of ODEs given by (2-3) with* $z_{k,j,i}(0) = x_{j,i}(0)$.

Proof The Markov chain $Z_{k,j,i}^{(N)}(t)$, for $N \geq 1$, is a density dependent population process as defined in [11, Chapter 11]. Theorem 2.1 in [11, Chapter 11] establishes our result provided that two conditions (being (2.6) and (2.7) in [11, Chapter 11]) apply for any $K \subset \bar{E}^J$ compact. We will argue that both conditions are valid on E^J which implies that they apply to any compact subset of $\bar{E}^J$.

The first condition demands that

$$\sum_{\boldsymbol{\ell}\in L} w(\boldsymbol{\ell},0) \sup_{\mathbf{x}\in E^J}\beta_{\boldsymbol{\ell}}(\mathbf{x}) < \infty,$$

where L is the set of all transitions and $\beta_{\boldsymbol{\ell}}(\mathbf{x})$ is the scaled transition rate of transition $\boldsymbol{\ell}$ in state $\mathbf{x}$. In our system there are three types of transitions (in a queue of length $i > 0$): arrivals, changes in the service phase and service completions. Arrivals in a queue of length i (in service phase j) increase the queue length by one and the vector $\boldsymbol{\ell}$ corresponding to an arrival therefore has two non-zero entries: being $\ell_{j,i}$ which equals -1 and $\ell_{j,i+1}$ which equals $+1$. Hence, $w(\boldsymbol{\ell},0) = 1/(i+1)^2$. Similarly for a change of service phase and a service completion in a queue of length i we find $w(\boldsymbol{\ell},0) = 1/(i+1)^2$.

The scaled rate of any of these transitions for any $\mathbf{x} \in E^J$ is bounded by $\lambda p_k/\gamma_k$ (for arrivals) and $\mu_k \max_j(-\sigma_{j,j})$ (for phase changes or service completions). Thus,

$$\sum_{\boldsymbol{\ell} \in L} w(\boldsymbol{\ell}, 0) \sup_{\mathbf{x} \in E^J} \beta_{\boldsymbol{\ell}}(\mathbf{x}) \leq \\ (J\lambda p_k/\gamma_k + J^2 \mu_k \max_j(-\sigma_{j,j})) \sum_{i \geq 0} 1/(i+1)^2 < \infty.$$

The second condition demands that $F_k(\mathbf{x})$ is Lipschitz, which was shown in Proposition 2. □

The above theorem indicates that the sample paths of the Markov chains converge to the unique solution of the set of ODEs given by (2-3) as the number of queues N tends to infinity over any finite time scale. One may wonder whether this convergence extends to the stationary regime, meaning whether the steady state measures of the Markov chains weakly converge to the Dirac measure of a fixed point of the set of ODEs. While we believe this to be the case (as indicated in next section that numerically validates this convergence), proving such a result is hard and considered to be out of scope of the current paper.

4.3 Validation

For the model validation we present only results for $K = 2$ types of servers, similar results were obtained for $K > 2$. Let $\mu_r = \frac{\mu_1}{\mu_2}$ and recall that that $\gamma_1\mu_1 + \gamma_2\mu_2 = 1$. Further assume that $\mu_1 > \mu_2$, meaning class 1 servers are the *fast* servers and class 2 the *slow* servers. As stated before the mean job size is assumed to be 1. Let C_X^2 be the squared coefficient of variation of the job size distribution. Whenever $C_X^2 = 1/k$ for some $k \in \{2, 3, \ldots\}$, we model the job size distribution as an Erlang distribution with k phases. For $C_X^2 \geq 1$, we used a hyperexponential (HEXP) distribution with parameters (α_1, ν_1, ν_2), thus with probability α_i a job is a type-i job and has an exponential duration with mean $1/\nu_i$, for $i = 1, 2$ (where $\alpha_2 = 1 - \alpha_1$). When $C_X^2 \geq 1$ we additionally match the fraction f of the workload that is contributed by the type-1 jobs (i.e., $f = \alpha_1/\nu_1$). If we assume that $\nu_1 \gg \nu_2$ this can be interpreted as stating that a fraction f of the workload is contributed by the *short* jobs. The mean (equal to 1), C_X^2 and fraction f can be matched as follows:

$$\nu_1 = \frac{C_X^2 + (4f - 1) + \sqrt{(C_X^2 - 1)(C_X^2 - 1 + 8f\bar{f})}}{2f(C_X^2 + 1)}, \tag{6}$$

$$\nu_2 = \frac{C_X^2 + (4\bar{f} - 1) - \sqrt{(C_X^2 - 1)(C_X^2 - 1 + 8f\bar{f})}}{2\bar{f}(C_X^2 + 1)}, \tag{7}$$

with $\bar{f} = 1 - f$ and $\alpha_1 = \nu_1 f$.

To validate the mean field model, the ODE based mean response times are compared to a discrete event simulation of the system for various parameter settings listed in Table 1. The discrete event simulation has an additional parameter N which is the size of the system. We let $N \in \{40, 80, 160, 320, 640, 1280\}$ and expect that the mean field model becomes more accurate as N increases. In fact due

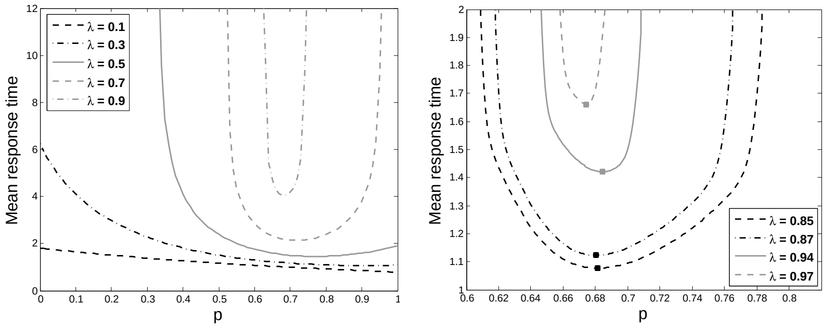
Case	λ	μ_r	γ_1	d	p_1	C_X^2
1	0.26754	1.34	0.6	2	0.1692	0.25
2	0.4116	2.8116	0.4	3	0.79378	0.25
3	0.29374	1.3922	0.7	4	0.47121	0.5
4	0.57975	1.9541	0.5	5	0.53563	0.125
5	0.18995	1.3764	0.3	3	0.43491	0.125
6	0.66992	2.2192	0.6	3	0.71812	0.5
7	0.65294	2.0177	0.4	5	0.57074	4
8	0.24765	1.7567	0.6	2	0.63567	2
9	0.75905	1.6631	0.3	3	0.38569	8
10	0.13251	2.2569	0.5	4	0.22224	4
11	0.78211	2.9592	0.6	5	0.95466	2
12	0.25638	2.8824	0.3	5	0.24434	8

Table 1: Parameter settings used to validate the accuracy of the mean field model.

Case	$N = 40$ (95% conf.)	$N = 80$ (95% conf.)	$N = 160$ (95% conf.)
1	1.321e−2 (±5.901e−5)	6.439e−3 (±3.939e−5)	3.225e−3 (±3.013e−5)
2	5.484e−3 (±3.682e−5)	2.646e−3 (±2.505e−5)	1.285e−3 (±1.546e−5)
3	2.229e−2 (±5.484e−5)	1.060e−2 (±3.433e−5)	5.277e−3 (±2.656e−5)
4	2.290e−2 (±4.309e−5)	1.078e−2 (±3.063e−5)	5.239e−3 (±2.331e−5)
5	7.873e−4 (±3.726e−5)	3.566e−4 (±2.547e−5)	1.819e−4 (±1.989e−5)
6	3.038e−2 (±6.501e−5)	1.415e−2 (±4.541e−5)	6.720e−3 (±2.900e−5)
7	5.163e−2 (±7.688e−4)	2.260e−2 (±7.466e−4)	1.114e−2 (±3.122e−4)
8	3.935e−3 (±4.969e−4)	1.535e−3 (±3.217e−4)	9.654e−4 (±3.264e−4)
9	9.580e−2 (±2.509e−3)	4.394e−2 (±1.804e−3)	1.937e−2 (±1.028e−3)
10	8.466e−3 (±1.027e−3)	3.486e−3 (±7.692e−4)	2.000e−3 (±4.353e−4)
11	1.631e−1 (±2.260e−3)	7.475e−2 (±1.102e−3)	3.594e−2 (±5.970e−4)
12	1.630e−2 (±1.228e−3)	8.366e−3 (±7.925e−4)	3.983e−3 (±6.873e−4)
Case	$N = 320$ (95% conf.)	$N = 640$ (95% conf.)	$N = 1280$ (95% conf.)
1	1.601e−3 (±1.742e−5)	8.157e−4 (±1.547e−5)	4.197e−4 (±1.081e−5)
2	6.338e−4 (±1.225e−5)	3.003e−4 (±9.128e−6)	1.477e−4 (±7.971e−6)
3	2.709e−3 (±1.710e−5)	1.429e−3 (±1.191e−5)	8.141e−4 (±1.066e−5)
4	2.587e−3 (±1.304e−5)	1.290e−3 (±9.276e−6)	6.288e−4 (±1.115e−5)
5	9.046e−5 (±1.415e−5)	3.764e−5 (±1.043e−5)	1.935e−5 (±5.237e−6)
6	3.109e−3 (±1.994e−5)	1.387e−3 (±1.329e−5)	5.081e−4 (±1.880e−5)
7	5.248e−3 (±2.872e−4)	2.691e−3 (±1.914e−4)	1.332e−3 (±1.746e−4)
8	3.784e−4 (±2.446e−4)	2.082e−4 (±1.665e−4)	1.151e−4 (±1.212e−4)
9	9.372e−3 (±6.956e−4)	5.681e−3 (±3.789e−4)	2.372e−3 (±3.195e−4)
10	9.152e−4 (±4.019e−4)	5.060e−4 (±2.877e−4)	2.400e−4 (±1.696e−4)
11	1.807e−2 (±4.032e−4)	1.012e−2 (±2.764e−4)	6.137e−3 (±2.767e−4)
12	2.280e−3 (±3.967e−4)	1.166e−3 (±2.814e−4)	4.118e−4 (±1.546e−4)

Table 2: Relative error of the mean field model wrt simulation.

to the results in [14], the expected response time predicted by the mean field model is $1/N$-accurate, which means that multiplying N by 2 should approximately reduce the relative error by a factor 2. The first six scenarios considered have Erlang distributed job sizes, the last six scenarios have hyperexponentially distributed job sizes where the fraction $f = 1/2$ (for $f \neq 1/2$ similar results were obtained). Table 2 shows the relative error of the mean field model and the associated 95% confidence interval of the simulation runs. In all cases the accuracy improves with N and the relative error is below or close to 10^{-2} for $N \geq 160$. We note that for small N the relative error can be further reduced by relying on the *refined* mean field approximation introduced in [15].



(a) $d = 2$

(b) $d = 20$

Fig. 2: Mean response time as a function of p for $\gamma_1 = \gamma_2 = 1/2$, $\mu_r = 2$, $f = 1/2$ and $C_X^2 = 4$.

5 Numerical results for randomized dispatching

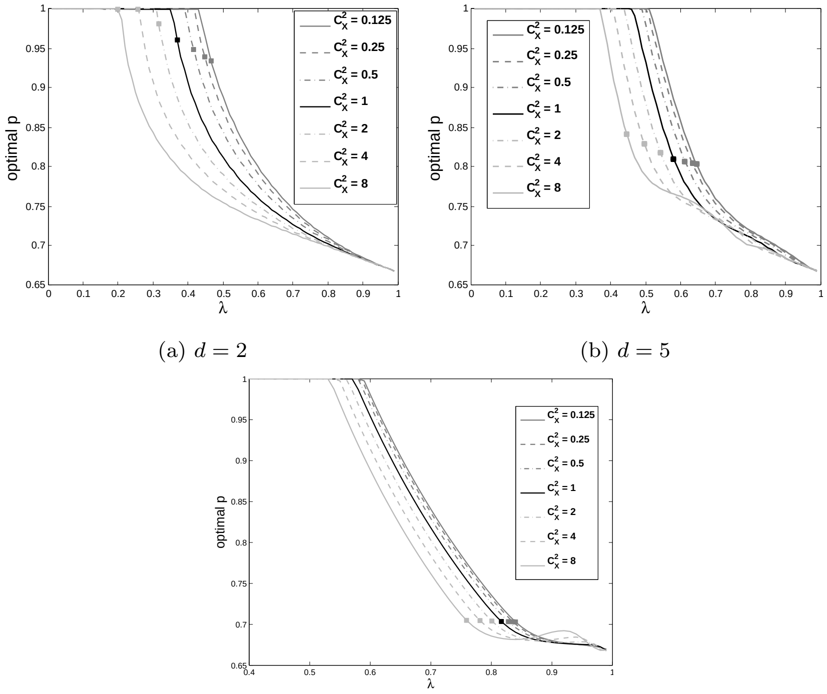
We mainly focus on the case with $K = 2$ types of servers and discuss settings with more than two types of servers in subsection 5.4.

5.1 Optimal p_1

In case of exponential job sizes we know that the mean response time is a convex function of p_1 as stated in Section 2. Various numerical experiments (see Figure 2 for one specific example) suggest that the mean response time is still convex in p_1 in case of non-exponential service times. Note that as λ approaches 1, the system is only stable in a very narrow region around $p = \gamma_1\mu_1$ (which corresponds to a simple proportional assignment). Let p_{opt} be the value of p_1 for which the resulting mean response time is minimized. We now study the impact of the various system parameters on p_{opt}. In Section 5.2 we look at the relative increase in the mean response time when a suboptimal p_1 is used.

Arrival rate λ: As illustrated in Figure 3 p_{opt} typically decreases as a function of λ (the squares mark the λ value for which the mean response time equals 1). This is expected as fewer jobs in the system implies that one can benefit from sending a larger fraction of the jobs to the fast servers. There are however exceptions, when the job sizes are highly variable and the number of choices is large (e.g., $C_X^2 = 8$ and $d = 20$) the optimal p_1 value may increase as a function of λ at high loads. For λ sufficiently small only the fast servers receive jobs and as $\lambda \to 1$ the load on both server types must be balanced to guarantee stability, i.e., p_1 and p_2 are such that $\frac{\lambda p_1}{\gamma_1\mu_1} = \frac{\lambda p_2}{\gamma_2\mu_2}$.

Job size variability C_X^2: When looking at the impact of the job size variability C_X^2 in Figure 3, we note that p_{opt} drops below 1 at lower rates λ when C_X^2 increases. This can be understood by noting that if all the jobs go to the fast servers and λ

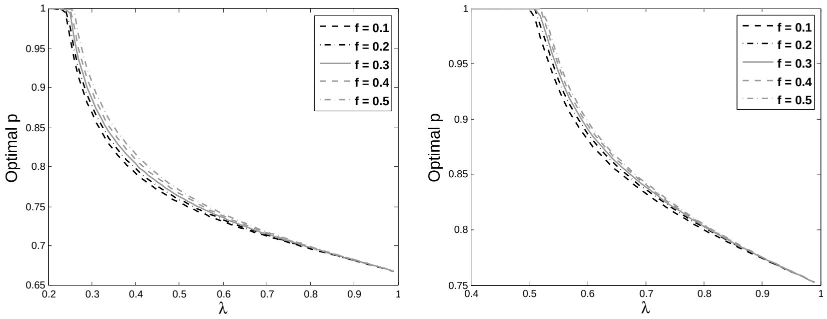


(a) $d = 2$

(b) $d = 5$

(c) $d = 20$

Fig. 3: Optimal choice of p_1 as a function of λ for $\gamma_1 = 0.5$, $\mu_r = 2$, $f = 1/2$ and different values of C_X^2



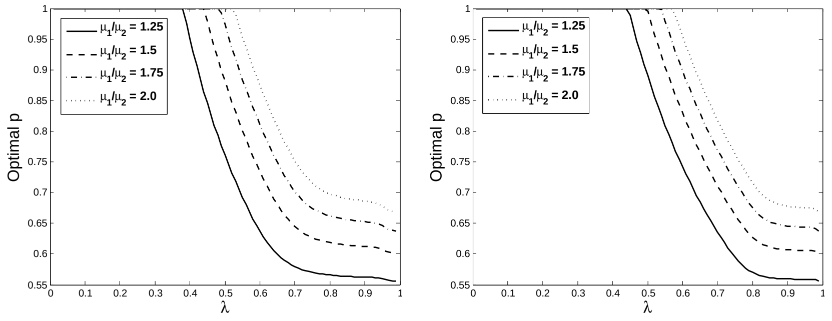
(a) $d = 2, C_X^2 = 4, \mu_r = 2$

(b) $d = 3, C_X^2 = 2, \mu_r = 3$

Fig. 4: Optimal p_{opt} as a function of λ for $\gamma_1 = \gamma_2 = 1/2$.

becomes large enough, some of the jobs start to experience queueing delays. When the job sizes are highly variable, there is a bigger risk of experiencing a long delay, thus it is advisable to start making use of the slow servers at smaller λ values.

For some parameter settings we see that more variable job sizes result in a lower p_{opt} for any arrival rate λ. This means that when job sizes become more



(a) $d = 10$

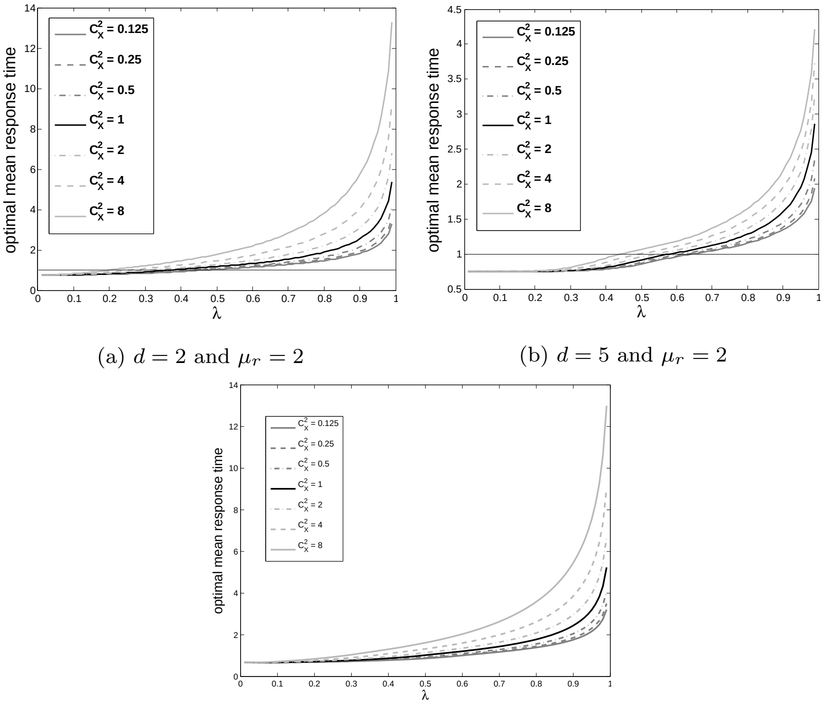
(b) $d = 20$

Fig. 5: Optimal choice of p_1 as a function of λ for $\gamma_1 = 0.5$, $f = 1/2$, $C_X^2 = 1$ and different values of $\mu_r = \mu_1/\mu_2$.

variable, it is beneficial to reduce the fraction of the jobs assigned to the faster servers when minimizing mean response times. This rule is however not valid in all cases: in Figure 3b, where $d = 5$ and $\mu_r = 2$, we see that p_{opt} for $C_X^2 = 8$ is larger than the corresponding value for $C_X^2 = 4$ for some λ ranges. The cause lies in the fact that the curves of p_{opt} start to oscillate notably for larger d values. These oscillations (that are also visible in Figure 2b) are probably caused by the fact that for larger d the tail probabilities of the queue length distribution decay very rapidly and depending on the precise value of λ a minor change in λ may cause a more significant change in the tail probabilities of either the fast or slow servers.

Number of choices d: Another observation from Figure 3 is that higher choices for d tend to increase the optimal value of p_1. When d increases the rate λ at which p_1 drops below 1 always seems to increase. This can be understood by noting that increasing d implies that the likeliness of finding an idle fast server when all the jobs are assigned to the fast servers increases. Thus the risk of experiencing a queueing delay decreases with d and therefore assigning all the jobs to the fast servers remains optimal for larger λ values. The fact that p_{opt} increases for increasing d is generally valid for small to medium loads, but does not remain valid for higher loads. For instance it is easily verified that when $\lambda = 0.8$ the p_{opt} for $d = 2$ equals 0.7024, while for $d = 10$ it equals 0.6982 when the job sizes are exponentially distributed.

Higher moments f: Figure 4 illustrates that the first two moments of the job size distribution do not suffice to determine the optimal split probability p_{opt}, meaning there is no insensitivity with respect to the moments beyond the second moment and optimizing p_{opt} in practice is therefore hard to achieve. The figure also indicates that the optimal fraction of jobs assigned to the fast servers is lower when a larger fraction of the workload consists of *long* jobs. This is intuitively clear: if a larger fraction $1 - f$ of the load is contributed by the long jobs, there is a bigger risk for short jobs to be stuck behind a long job and therefore it is better to make more use of the slow servers.



(a) $d = 2$ and $\mu_r = 2$

(b) $d = 5$ and $\mu_r = 2$

(c) $d = 2$ and $\mu_r = 3$

Fig. 6: Optimal mean response times as a function of λ for $\gamma_1 = 0.5$ and $f = 1/2$ for different values of C_X^2.

System heterogeneity μ_r: We expect that p_{opt} tends to increase as the system heterogeneity $\mu_r = \mu_1/\mu_2$ increases. Figure 5 confirms this intuition for the case with $d = 10$ and 20 choices when $\gamma = f = 1/2$ and $C_X^2 = 1$.

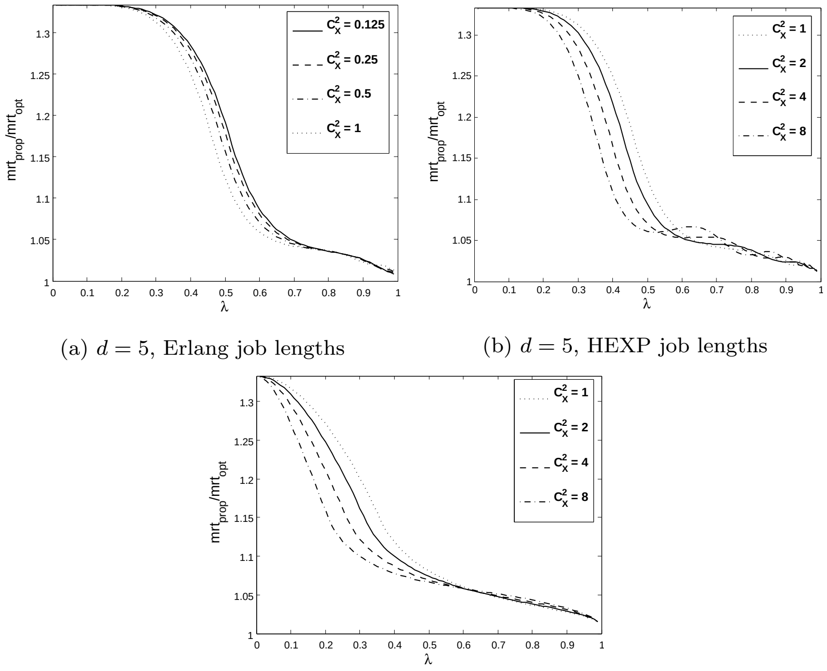
5.2 Accuracy of simple suboptimal policies

We start by depicting the mean response time for various settings of d, μ_r and C_X^2 in Figure 6 when using the optimal splitting probability p_{opt}. As expected the mean response time increases with the job size variability, decreases as a function of d and μ_r, and drops below 1 for sufficiently low loads as the mean service time of the fast servers is less than one.

More importantly, one may wonder how much gain in the mean response time one achieves by optimizing p_1. For this purpose we now study the relative gain in the mean response time of the optimal p_1 with the following three less complex assignment policies:

- Proportional: in this case a job is assigned to class k with probability

$$p_k = \frac{\gamma_k \mu_k}{\sum_{j=1}^{K} \gamma_j \mu_j},$$



(a) $d = 5$, Erlang job lengths

(b) $d = 5$, HEXP job lengths

(c) $d = 2$, HEXP job lengths

Fig. 7: Ratio $\text{mrt}_{\text{prop}}/\text{mrt}_{\text{opt}}$ as a function of λ with $\mu_r = 2$.

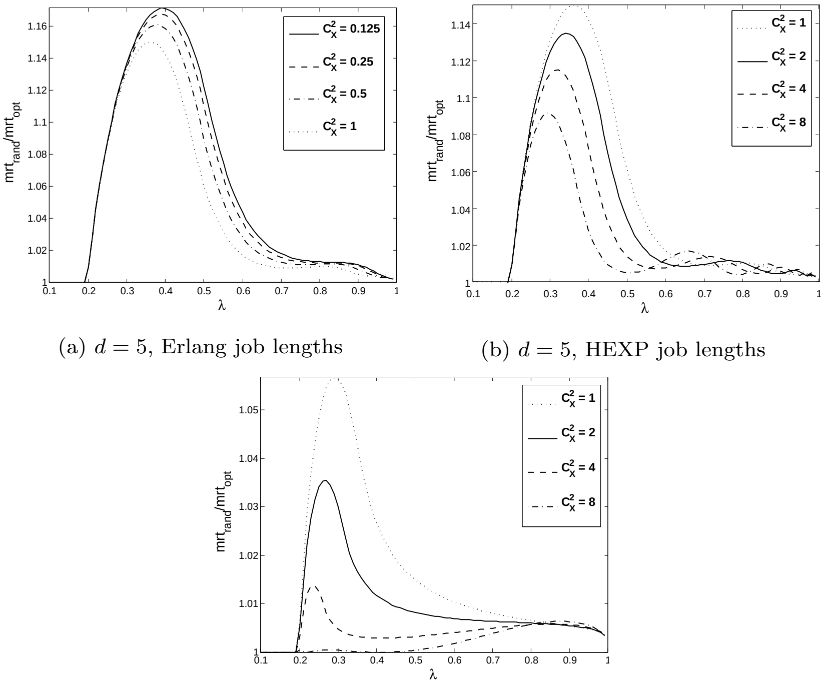
such that each of the K classes experiences the same load.

- Random within a class: in this case we use the optimized probability p_k by assuming that a random server is selected within a class and job lengths are exponential. Hence, p_k is given by the explicit formula in [3,13], which can be written as

$$p_k = \frac{1}{\lambda}\frac{\gamma_k \mu_k}{\sum_{j=1}^{K}\gamma_j \mu_j} + (1 - \frac{1}{\lambda})\frac{\gamma_k \sqrt{\mu_k}}{\sum_{j=1}^{K}\gamma_j \sqrt{\mu_j}}, \tag{8}$$

  where p_k is set to one (zero) when the above formula results in a p_k larger than one (less than zero). Note as λ approaches one, these probabilities converge to the proportional scheme.
- Exponential job size: in this case we optimze p_1 by solving the convex optimization problem of (1). Hence we optimize assuming that the job lengths are exponential.

Note that for all three policies the probabilities $p_1, \ldots, p_K$ only depend on the server speeds and the mean job size (which equals one), either by means of an explicit formula or via a simple convex optimization problem. When $K = 2$ we denote p_1 for the above three policies as p_{prop}, p_{rand} and p_{exp}. Their corresponding mean response times are denoted as $\text{mrt}_{\text{prop}}, \text{mrt}_{\text{rand}}$ and mrt_{exp}, respectively.



(a) $d = 5$, Erlang job lengths

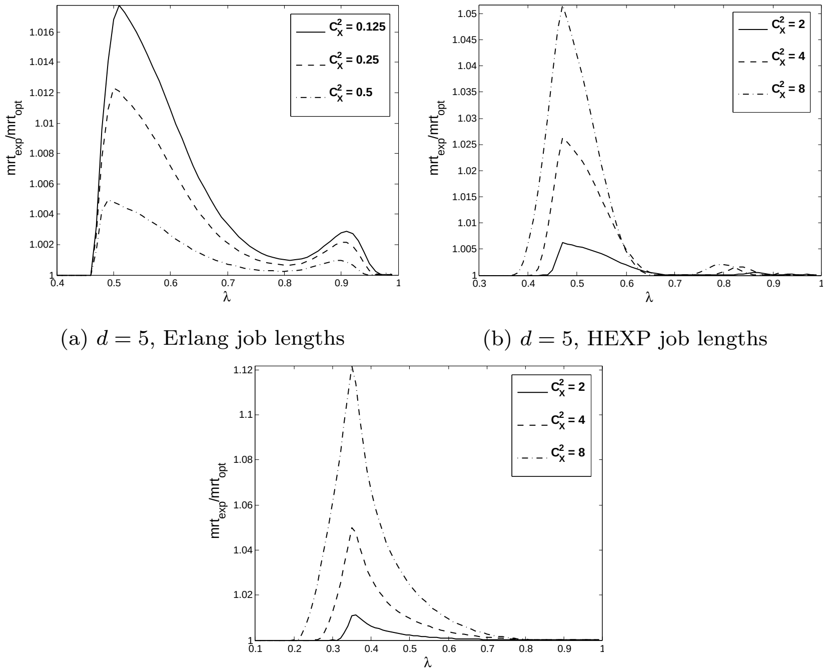
(b) $d = 5$, HEXP job lengths

(c) $d = 2$, HEXP job lengths

Fig. 8: Ratio $\mathrm{mrt}_{\mathrm{rand}}/\mathrm{mrt}_{\mathrm{opt}}$ as a function of λ with $\mu_r = 2$.

p_{prop} versus p_{opt}: Figure 7 depicts the relative increase in the mean response time if we replace the optimal policy (i.e., p_1 value) with a simple proportional assignment for $\mu_r = 2$. Similar results were obtained for other choices of μ_r. While the proportional scheme is very simple, it results in poor performance for low to medium loads and this loss in performance compared to the optimal policy grows as the number of choices d increases (see Figure 7b versus 7c). This is as expected as the optimal strategy under low load exists in sending all the jobs to the fast servers, while the proportional scheme balances the load among the servers.

p_{rand} versus p_{opt}: Figure 8 depicts the relative increase in the mean response time when relying on (8) instead of using the optimal value for p_k. For small λ both policies (that is, the optimal and the random within a class policy) assign all the jobs to the fast servers. For somewhat higher arrival rates (at about 0.2 in Figure 8) the random within a class policy starts utilizing the slower servers as well, while the optimal strategy continues to assign all the jobs to the fast servers. Indeed, when all the jobs are assigned to the fast servers, the risk of assigning a job to a busy server increases as d decreases, thus the smaller d the sooner one needs to utilize the slow servers. Figure 8 illustrates that assuming a random assignment (i.e., $d = 1$) results in a performance loss of up to 15% that tends to increase with the number of choices d and that decreases when the job sizes become more variable. The latter can be understood by looking at Figure 3 which indicates that



(a) $d = 5$, Erlang job lengths

(b) $d = 5$, HEXP job lengths

(c) $d = 2$, HEXP job lengths

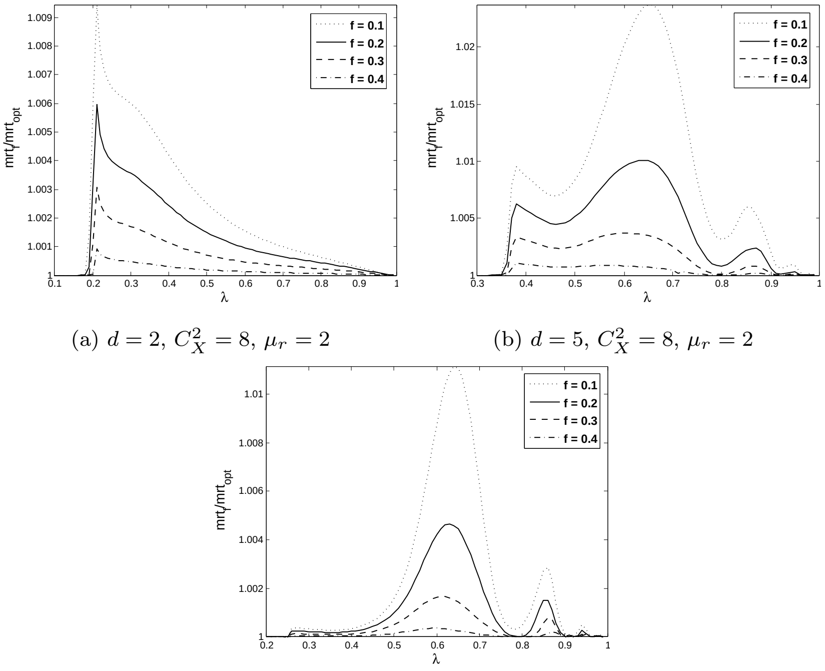
Fig. 9: Ratio $\text{mrt}_{\text{exp}}/\text{mrt}_{\text{opt}}$ as a function of λ with $\mu_r = 2$.

under low to medium loads, p_{opt} increases as a function of d and decreases as a function of C_X^2. Therefore p_{opt} and p_{rand} are more alike for small d and large C_X^2. We note that in the limit as λ goes to one, both policies use proportional assignment and thus perform alike.

When comparing the relative errors of the proportional scheme with the random within a class policy (compare Figures 7 and 8), we see that the latter results in lower relative errors. We should however note that the proportional scheme is easier to implement as it does require an estimation of the arrival rate λ.

p_{exp} versus p_{opt}: Figure 9 studies the relative increase in the mean response time when we only neglect the higher moments of the job size distribution when optimizing p_1. When $d = 5$ this results in errors below 5% and the error is only significant in a fairly small load region. Thus for large enough d, taking the job size variability into account is not paramount (this was confirmed for other μ_r values). When $d = 2$ the relative increase does surge up to 12% in case of highly variable job sizes when λ is close to 0.35. The load at which the relative error is the highest corresponds to the largest arrival rate λ for which p_{exp} still equals 1. Thus, for highly variable job sizes the region where the relative error surges up corresponds to the settings where p_{opt} drops below 1, but p_{exp} remains 1.

Note that solving the convex optimization problem (1) or computing (8) both requires one to estimate the arrival rate λ. When comparing Figures 8 and 9, it



(a) $d = 2$, $C_X^2 = 8$, $\mu_r = 2$

(b) $d = 5$, $C_X^2 = 8$, $\mu_r = 2$

(c) $d = 5$, $C_X^2 = 8$, $\mu_r = \sqrt{2}$

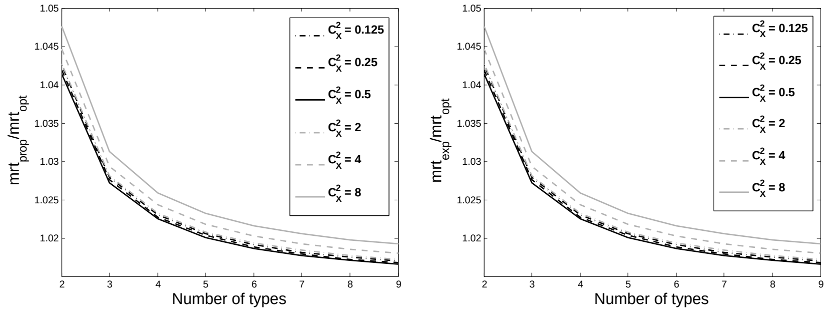
Fig. 10: Ratio $\mathrm{mrt_f}/\mathrm{mrt_{opt}}$ as a function of λ.

is clear that solving the convex optimization problem (which can be done in a fraction of a second) is far more effective than relying on (8) for $d = 5$, i.e., larger d values. Indeed, the convex optimization problem takes the value of d into account and therefore causes smaller errors for d large. A somewhat unexpected result is that (8) does result in smaller relative error when $d = 2$, in case of medium loads and highly variable job sizes. The explanation is that when computing p_{rand} we make two errors that mostly cancel each other in this case: we assume that $d = 1$ and that jobs have exponential sizes. For p_{exp} only the latter error is present.

5.3 Impact of the 3rd and higher moments of the job size variability

In the previous section we studied the impact of neglecting the job size variability when optimizing p by comparing the performance gain obtained by using p_{opt} instead of p_{exp}. In this section we look at the impact of the higher moments (3rd and beyond). To investigate their impact we consider a hyperexponential distribution as defined in Section 4.3, where we matched the mean $EX = 1$, the squared coefficient of variation C_X^2 and the fraction f of the workload contributed by the *short* jobs. Note that changing f influences the higher moments of the job size distribution, but not the mean or variance.

To assess the impact of the higher moments we therefore consider job size distributions with $f \neq 1/2$ and compare the mean response time in a system with



(a) Proportional policy

(b) Exponential job sizes policy

Fig. 11: Relative increase in mean response time when a suboptimal policy is used as a function of the number K of server types for $\lambda = 0.75$, $d = 2$ and $\mu_1/\mu_K = 2$.

p_1 optimized for $f = 1/2$, denoted as p_f, with the optimal p_1. Figure 10 depicts the relative gain obtained by using the optimal p_1 instead of p_f when $C_X^2 = 8$ (smaller C_X^2 values result in even smaller relative gains) for $f = 0.1$ to 0.4. While the shape of these curves is very unpredictable and irregular, it is also clear that the relative gain is very minor and less than 2.5% in all cases considered. This indicates that there is little use in taking these higher moments into account when optimizing p_1 (which is good as they are harder to estimate in practice compared to the mean or variance).

5.4 Beyond 2 server types

In the previous subsections we assumed that the system consists of two types of servers only. In this section we illustrate that as the number of server types K increases, the differences between the mean response time of the simple policies considered in subsection 5.2 and the optimal choice of $p_1, \ldots, p_K$ decreases. In other words, the scenario with $K = 2$ in a way provides an upper bound on how much one gains by optimizing the p_k probabilities.

In Figure 11 we set $d = 2$, $\lambda = 0.75$, $\gamma_k = 1/K$ and ordered the server types such that $\mu_1 > \mu_2 > \ldots \mu_K$ with $\mu_1/\mu_K = 2$ and $\mu_{k+1} - \mu_k = \mu_k - \mu_{k-1}$ for $k = 2, \ldots, K-1$. We depict the increase in the mean job response time when the *proportional* and *exponential job size* policies are used instead of the optimal p_k values with both low and high job size variability (with $f = 1/2$). The results confirm our intuition that this increase tends to diminish as more server types K are introduced.

6 Analysis for size based dispatching

In this section we indicate how to determine the mean response time in case the class k is selected based on the job size x of an incoming job when the number of servers becomes large. Given the thresholds $0 = T_0 < T_1 < \ldots < T_K = \infty$, the system is equivalent to a set of K homogeneous POD LB systems. The k-th system is equivalent to a system with arrival rate is $\lambda N/(\gamma_k\mu_k)$ times $P[T_{k-1} \leq X < T_k]$, service rate 1 and the job size distribution, denoted as $X^{(k)}$, is a truncated phase-type distribution, that is:

$$P[X^{(k)} > x] = \begin{cases} 1 & x \leq T_{k-1}, \\ \frac{\alpha(e^{Sx} - e^{ST_k})e}{\alpha(e^{ST_{k-1}} - e^{ST_k})e} & T_{k-1} < x < T_k, \\ 0 & x \geq T_k. \end{cases}$$

Due to the truncated job size distribution, we cannot define an ODE-based mean field model as for the randomized dispatching policy in Section 4. Instead we make use of the cavity method introduced in [7]. This method can be used to assess the mean response time of the POD LB in a large homogeneous system with general job size distributions and thus also for truncated phase-type distributions. The accuracy of the cavity method for large finite systems was already numerically validated in [7] and as such we do not include such a numerical validation here. In the remainder of this section we outline how to compute the queue length distribution for a homogeneous system with POD LB and truncated phase-type jobs sizes. The mean response time of the heterogeneous system is then obtained by applying this method K times combined with Little's law.

The idea of the cavity method exists in starting with some set of arrival rates $(\lambda_0^{(0)}, \lambda_1^{(0)}, \lambda_2^{(0)}, \ldots)$ and to compute the queue length distribution $Q^{(0)}$ of the $M_n/G/1$ queue, which is an M/G/1 queue with queue length dependent arrival rates $\lambda_n^{(0)}$. Next, a new set of arrival rates $(\lambda_0^{(1)}, \lambda_1^{(1)}, \lambda_2^{(1)}, \ldots)$ is computed from $Q^{(0)}$ and $Q^{(1)}$ is computed as the queue length distribution of the $M_n/G/1$ queue with arrival rates $(\lambda_0^{(1)}, \lambda_1^{(1)}, \lambda_2^{(1)}, \ldots)$. This procedure is repeated iteratively until convergence takes place.

We now outline how to compute $(\lambda_0^{(n+1)}, \lambda_1^{(n+1)}, \lambda_2^{(n+1)}, \ldots)$ from $Q^{(n)}$ and $Q^{(n)}$ from $(\lambda_0^{(n)}, \lambda_1^{(n)}, \lambda_2^{(n)}, \ldots)$. Assume that the total arrival rate in the homogeneous system is $\lambda' N$ and the service time distribution is a truncated phase-type distribution on $[a, b]$ (for our k-th system we have $\lambda' = \lambda/(\gamma_k\mu_k)$, $a = T_{k-1}$ and $b = T_k$). Then,

$$\lambda_i^{(n+1)} = \lambda' d \sum_{k=0}^{d-1} \binom{d-1}{k} \frac{P[Q^{(n)} = i]^k P[Q^{(n)} > i]^{d-1-k}}{k+1}, \tag{9}$$

which is the arrival rate to a tagged queue of length i if the d queue lengths are independent and ties are broken uniformly at random.

The distribution $Q^{(n)}$ is simply the queue length distribution of an $M_n/G/1$ queue with queue length dependent arrival rates $(\lambda_0^{(n)}, \lambda_1^{(n)}, \lambda_2^{(n)}, \ldots)$. Although elegant expressions for the queue length distribution of an $M_n/G/1$ queue have been derived before [10], these formulas tend to result in numerical issues. We therefore present a different approach that exploits the fact that the service time is a phase-type distribution truncated on $[a, b]$.

We start our iterative scheme with $\lambda_0^{(0)} = \lambda'$ and $\lambda_i^{(0)} = 0$ for $i > 0$. Therefore, we have $\lambda_i^{(n)} = 0$ for $i > n$ due to (9) and $P(Q^{(n)} > n + 1) = 0$. To compute $P(Q^{(n)} = i)$ for $i \leq n$ we first define an $n + 1$ state Markov chain by observing the $M_n/G/1$ queue at service completion times. The main challenge is to determine the probability $Q_{i,j}$ that the queue length changes from i at the start of service to $j-1$ when the service is completed. This requires some care as the arrival rate depends on the queue length and is therefore not necessarily fixed during the service of a single customer.

Define the bidiagonal matrix M as

$$M = \begin{bmatrix} -\lambda_1^{(n)} & \lambda_1^{(n)} & & & & \\ & -\lambda_2^{(n)} & \lambda_2^{(n)} & & & \\ & & \ddots & \ddots & & \\ & & & -\lambda_n^{(n)} & \lambda_n^{(n)} \\ & & & & 0 \end{bmatrix},$$

then $W_{i,j}$ is given by entry (i,j) of the matrix W with

$$\begin{aligned} W &= \frac{(\alpha \otimes I) \int_a^b (e^{Ss} \otimes e^{Ms}) ds (-Se \otimes I)}{\alpha(e^{Sa} - e^{Sb})e} \\ &= \frac{(\alpha \otimes I)(S \oplus M)^{-1}(e^{(S\oplus M)b} - e^{(S\oplus M)a})(-Se \otimes I)}{\alpha(e^{Sa} - e^{Sb})e}, \end{aligned}$$

where $\otimes$ is the Kronecker product and $S \oplus M = S \otimes I + I \otimes M$. Using these probabilities $W_{i,j}$ we can easily compute the queue length distribution at service completion times. To determine the queue length distribution $Q^{(n)}$ at a random point in time, it now suffices to compute the mean time between two service completions and $X_{j,i}$, which is the expected amount of time that the queue length equals i during the service of a job that started service when the queue length equaled $j \geq 1$. Let X be the matrix with entries $X_{j,i}$, then

$$\begin{aligned} X &= \int_0^\infty P(\text{ service time } > s) e^{Ms} ds \\ &= \int_0^a e^{Ms} ds + \int_a^b \frac{\alpha(e^{Ss} - e^{Sb})e}{\alpha(e^{Sa} - e^{Sb})e} e^{Ms} ds. \end{aligned}$$

The latter integral can be expressed as

$$(\alpha \otimes I) \int_a^b \frac{(e^{Ss} - e^{Sb}) \otimes e^{Ms}}{\alpha(e^{Sa} - e^{Sb})e} ds (e \times I),$$

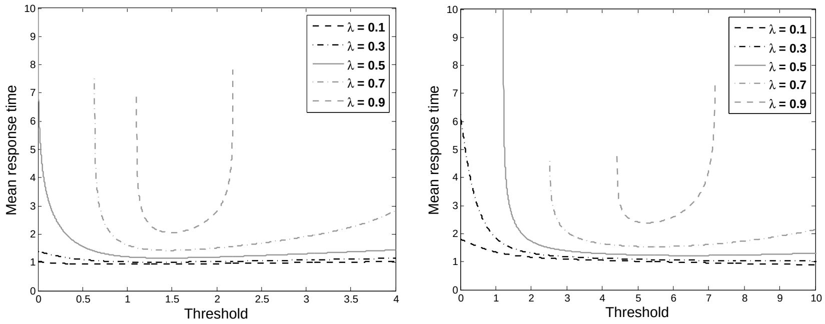
which simplifies to

$$\frac{(\alpha \otimes I)(S \oplus M)^{-1}(e^{(S\oplus M)b} - e^{(S\oplus M)a})(e \otimes I)}{\alpha(e^{Sa} - e^{Sb})e} - \frac{\alpha e^{Sb} e}{\alpha(e^{Sa} - e^{Sb})e} \int_a^b e^{Ms} ds.$$

7 Numerical results for size based dispatching

We focus on the case with $K = 2$ server types and denote T_1 as T. Thus, all jobs with a size $x < T$ go to one class and the remaining jobs go to the other class. We refer to jobs with a size below T as short jobs and to jobs with a size that exceeds T as long jobs. There are two strategies of selecting the class:

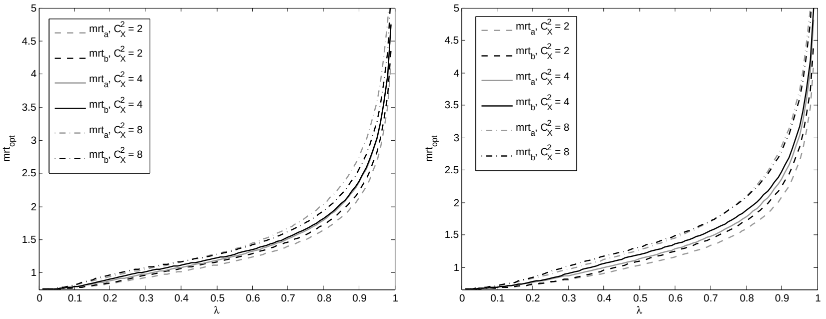
- Long jobs to fast servers, short jobs to slow servers



(a) Long jobs to fast servers

(b) Short jobs to fast servers

Fig. 12: Mean response time as a function of the job size threshold for $\gamma_1 = \gamma_2 = 1/2$, $\mu_r = 2$ and $C_X^2 = 4$.



(a) $\mu_r = 2$

(b) $\mu_r = 3$

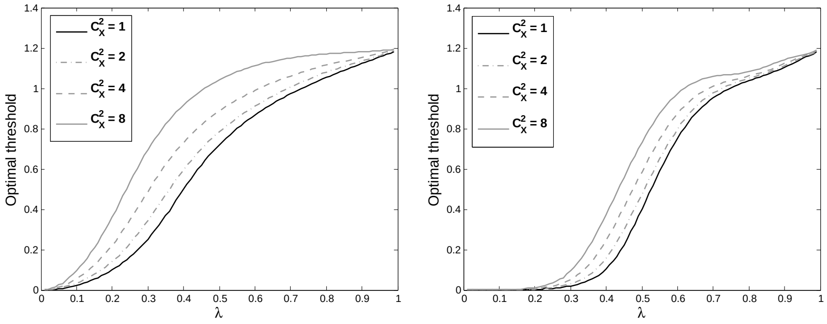
Fig. 13: Optimal mean response times in function of λ for both job size interval assignment approaches for $\gamma_1 = \gamma_2 = 1/2$ and $d = 2$.

- Short jobs to fast servers, long jobs to slow servers

As can be seen in Figure 12, for both strategies there appears to be a unique value for T for which the mean response time is minimal (and not all choices of T result in a stable system). Both strategies do not necessarily yield the same optimal mean response time. In fact, for the job size distributions considered in this paper, it turns out that it is typically more efficient to send the long jobs to the fast servers as illustrated in Figure 13 (mrt_a is the optimal mean response time achieved by sending long jobs to fast servers, mrt_b for sending short jobs to fast servers). In the remainder of this section we therefore limit ourselves to the first strategy.

7.1 Optimal threshold T

In this section we look at the impact of various system parameters on the optimal threshold value T using Figures 14 and 15.



(a) $d = 2$ (b) $d = 5$

Fig. 14: Optimal threshold as a function of the arrival rate λ for $\gamma_1 = \gamma_2 = 1/2$, $\mu_r = 2$.

Arrival rate λ*:* For low loads, more jobs should be sent to the faster servers, resulting in a low optimal threshold T. For low and high loads the optimal threshold is not very sensitive to the arrival rate. As λ approaches 1, the optimal threshold T is such that both server types have the same load.

Job size variability C_X^2*:* For higher job size variability, the optimal threshold T starts to increase at lower loads and increases more slowly at higher loads. This is expected as with more variable job sizes the risk of being stuck behind a long job increases and therefore it is preferential to send more short jobs to the slower servers, where there is no risk of being stuck behind a long job.

Number of choices d*:* For a higher number of choices d, the optimal threshold T is less sensitive to the job size variability. Further, the optimal threshold stays close to zero for a larger range of arrival rates when d increases. This is intuitively clear as a higher number of choices increases the probability of selecting an idle server.

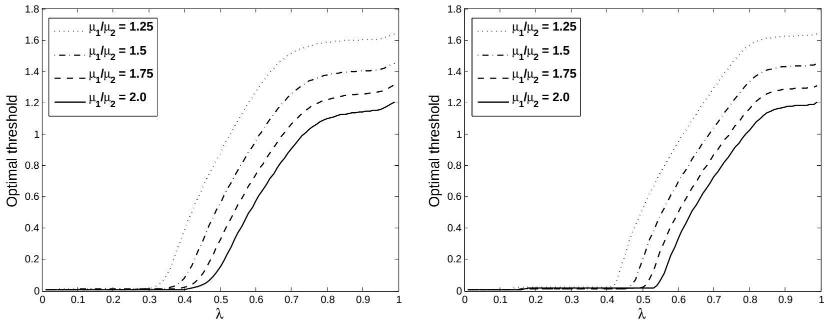
System heterogeneity μ_r*:* Figure 15 shows that a higher ratio μ_r leads to a lower optimal threshold, which is to be expected, as this means that the faster servers can handle more jobs in less time.

7.2 Accuracy of simple suboptimal strategies

In this section we study the gain of the optimal threshold T (which requires some effort to compute) with a few basic suboptimal strategies.

- Proportional: In this case T is set such that there is an equal load on both server types. This heuristic was also proposed in [18,9].
- Load dependent threshold: In this case, we determine the load on the fast servers ρ_1 using a parameter ρ_m with the following function:

$$\rho_1(\rho_m, \lambda, \mu_1, \gamma_1) = \begin{cases} \frac{\lambda}{\mu_1\gamma_1} & : \rho_m > \frac{\lambda}{\mu_1\gamma_1} \\ 1 - \frac{(1-\lambda)(1-\rho_m)}{1-\rho_m\mu_1\gamma_1} & : \rho_m \leq \frac{\lambda}{\mu_1\gamma_1} \end{cases}$$



(a) $d = 10$

(b) $d = 20$

Fig. 15: Optimal threshold as a function of λ for $\gamma_1 = \gamma_2 = 1/2$, $C_X^2 = 2$ and different values of $\mu_r = \mu_1/\mu_2$.

```
Input: $\lambda, \gamma, \mu, C_X^2, d$
Output: $\rho_m$
$\rho_m = 0$;
$\lambda = 0.01$; while $\lambda < 1$ do
    $\rho_m = \frac{\lambda - 0.01}{\mu_1\gamma_1}$;
    $mrt_{\rho_m} = mrt(\lambda, \mu_r, C_X^2, d, T_{\rho_m})$;
    $mrt_0 = mrt(\lambda, \mu_r, C_X^2, d, 0)$;
    if $mrt_0 > mrt_{\rho_m}$ then
        break;
    end
    $\lambda = \lambda + 0.01$
end
```

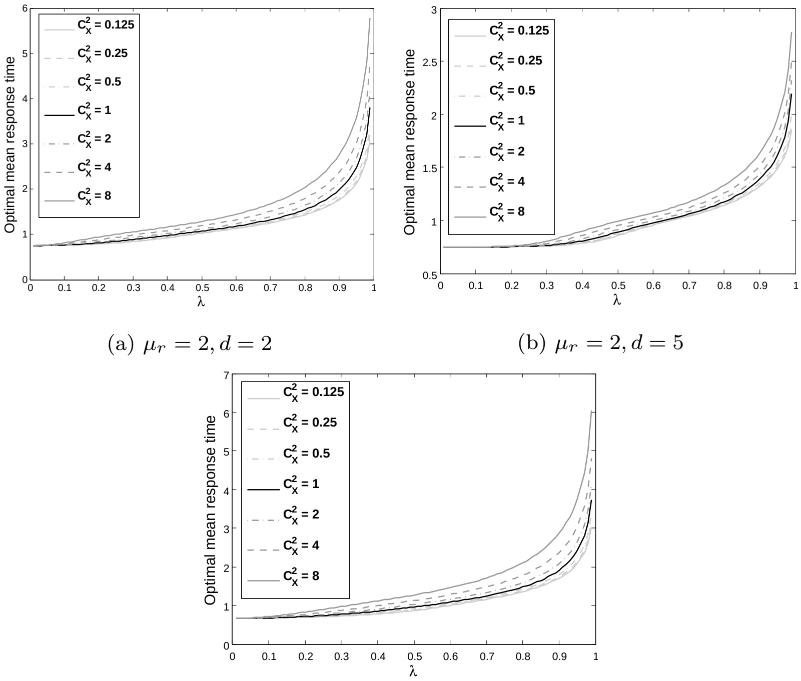
Algorithm 1: Algorithm to set ρ_m

With this heuristic all traffic is sent to the fast servers if the resulting load on these servers remains below ρ_m. For higher loads, the load on the fast servers increases linearly (towards proportional load for $\lambda \to 1$). The resulting ρ_1 is subsequently used to determine the threshold T_{ρ_m} of the system. Figure 17 illustrates the value of ρ_1 and its respective threshold under this scheme.

A reasonable value for this parameter ρ_m can be computed using Algorithm 1. This algorithm finds the lowest load (that is a multiple of 0.01) such that it is no longer better to simply send all jobs to the fast servers.

T_{prop} versus T_{opt}: In Figure 18 we can see that the proportional strategy gives poor results for low loads, as a more proportional division of jobs is best suited for highly loaded systems (to avoid unstable systems). For lower values of d the performance loss of the proportional heuristic at low loads clearly reduces. This performance loss is also less pronounced for more variable job sizes.

T_{ρ_m} versus T_{opt}: Figure 19 indicates that the load dependent heuristic performs better than the proportional one, especially at low loads. It is however somewhat more complex as we need to set the parameter ρ_m (which was done using Algorithm



(a) $\mu_r = 2, d = 2$

(b) $\mu_r = 2, d = 5$

(c) $\mu_r = 3, d = 2$

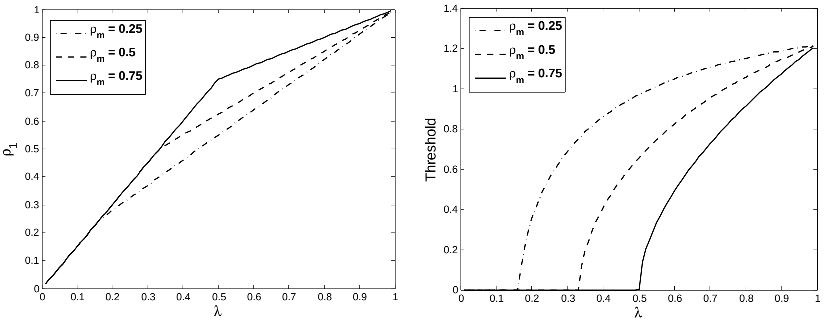
Fig. 16: Optimal mean response time as a function of the arrival rate λ for $\gamma_1 = \gamma_2 = 1/2$.

1). When compared to the optimal choice of T, this heuristic increases the mean response time by less than 5% when $\mu_r = 2$ for the job size distributions under consideration.

7.3 Randomized versus size based dispatching

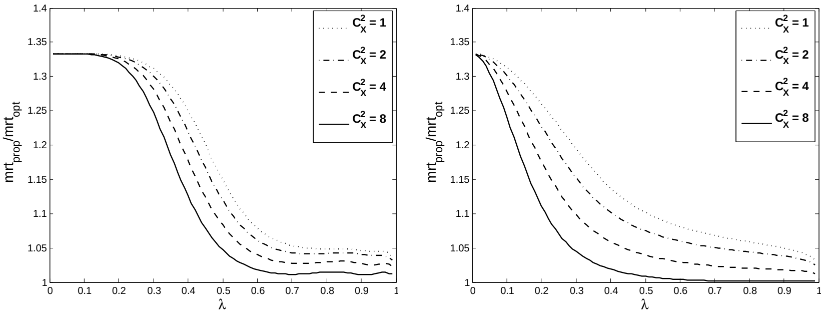
In this section we compare the mean response time of the randomized and size based dispatching. The size based dispatching policy requires knowledge of the job lengths, but should result in a better mean response time. The main objective is to get some insight on the performance gain that can be obtained if such job size information is available.

In Figure 20 we plot the mean response time of the optimal randomized and size based policy and the two size based heuristics, where the parameter ρ_m was set using Algorithm 1. As expected the optimal size based strategy outperforms the optimal randomized strategy and the margin of improvement increases as the load increases. The load dependent heuristic is also very close to the optimal size based strategy. More surprisingly, even the simple proportional size based heuristic outperforms optimal randomized dispatching, except for small loads.



(a) ρ_1 as a function of λ and ρ_m (b) Threshold as a function of λ and ρ_m

Fig. 17: Load on fast servers and job size threshold in function of ρ_m for $\gamma_1 = \gamma_2 = 1/2$, $\mu_r = 2$ and $C_X^2 = 4$.



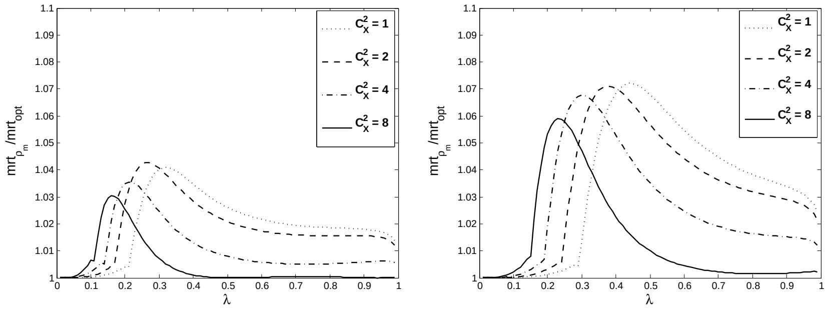
(a) $d = 5$, HEXP job lengths (b) $d = 2$, HEXP job lengths

Fig. 18: Ratio $\text{mrt}_{\text{prop}}/\text{mrt}_{\text{opt}}$ as a function of λ with $\mu_r = 2$

Figures 21 and 22 show the relative difference between the optimal randomized and optimal size based strategies. For higher λ or job size variability, the mean response times can be lowered significantly by a size based load balancing approach. An interesting observation is that the relative difference between both approaches becomes less pronounced as the system becomes more heterogeneous. Intuitively this makes sense as in a system where the fast servers are much faster than the slow ones, most of the jobs should go to the fast servers irrespective of whether randomized or size based dispatching is used.

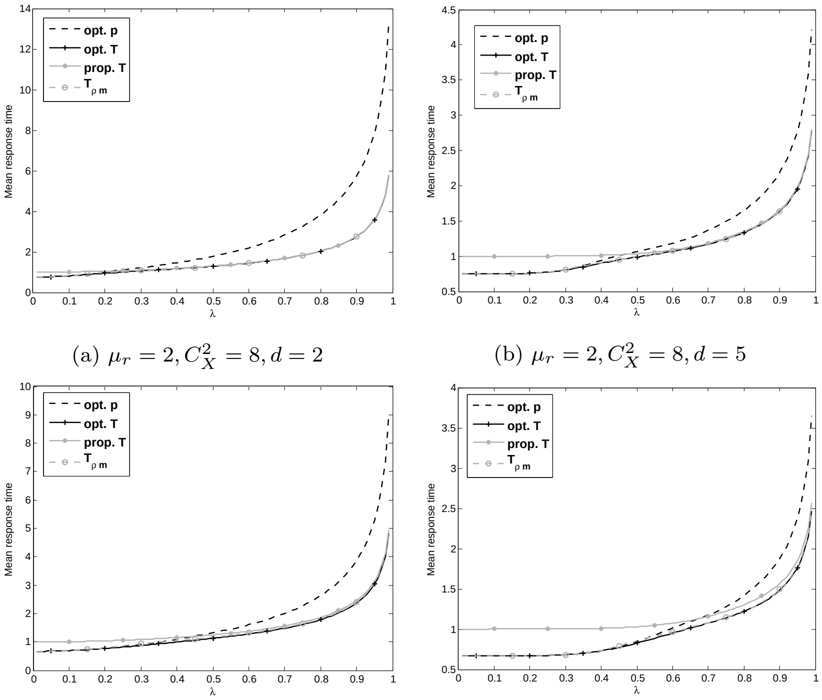
8 Conclusion

A class of load balancing schemes for a heterogeneous set of FCFS servers is analyzed. The servers are partitioned in K classes of servers: within each class all servers are identical, while servers belonging to different classes only differ in their server speed. Jobs are assigned to a server class either via randomization or based on their size x. A power-of-d choices rule is used to select a server within



(a) $\mu_r = 2$, HEXP job lengths (b) $\mu_r = 3$, HEXP job lengths

Fig. 19: Ratio $\mathrm{mrt}_{\rho_m}/\mathrm{mrt}_{\mathrm{opt}}$ as a function of λ with $d = 2$



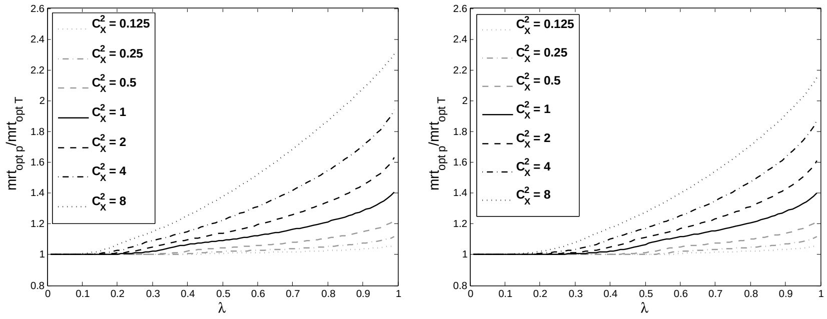
(a) $\mu_r = 2, C_X^2 = 8, d = 2$ (b) $\mu_r = 2, C_X^2 = 8, d = 5$

(c) $\mu_r = 3, C_X^2 = 4, d = 2$ (d) $\mu_r = 3, C_X^2 = 4, d = 5$

Fig. 20: Comparison of different strategies: mean response times in function of λ

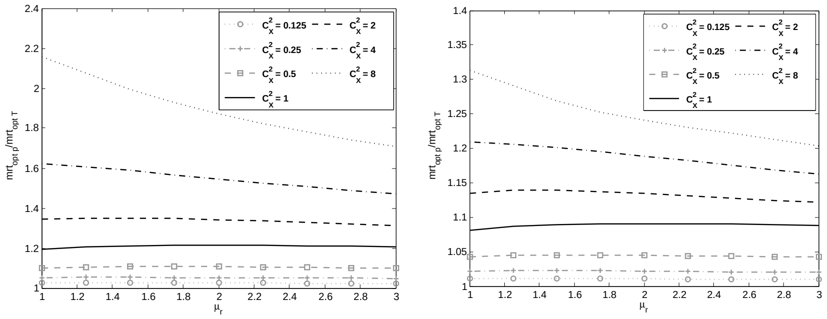
the selected class. We developed an ODE-based mean field model to estimate the mean job response time in a system with many servers in case of randomized dispatching and used the cavity method to assess the mean response time for size based dispatching.

While the impact of the different system parameters (like the job size variability or number of choices d) on the optimal randomization probabilities is not



(a) $\mu_r = 2, d = 2$ (b) $\mu_r = 3, d = 2$

Fig. 21: Comparison of opt. p vs opt. threshold: ratio of optimal mean response times in function of λ



(a) $d = 2$ (b) $d = 5$

Fig. 22: Comparison of opt. p vs opt. threshold: ratio of optimal mean response times in function of μ_r for $\lambda = 0.8$

always easy to predict (due to oscillations in some of the curves), the main insight provided is that only taking the mean job sizes into account when determining the randomization probabilities (via convex optimization) often results in a very limited loss in performance compared to the optimal probabilities. For size based dispatching we showed that a simple load dependent heuristic often achieves a mean response time that is close to optimal.

Finally, we illustrated that size based dispatching outperforms optimal randomized dispatching even if we use a simple heuristic to estimate the threshold value.

References

1. A. K. A. Mukhopadhyay and R. R. Mazumdar. Randomized assignment of jobs to servers in heterogeneous clusters of shared servers for low delay. *Stochastic Systems*, 6(1):90 – 131, 2016.

2. R. Aghajani, X. Li, and K. Ramanan. The pde method for the analysis of randomized load balancing networks. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2):38:1–38:28, Dec. 2017.
3. E. Altman, U. Ayesta, and B. Prabhu. Load balancing in processor sharing systems. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '08, pages 12:1–12:10, 2008.
4. E. Bachmat and H. Sarfati. Analysis of SITA policies. *Performance Evaluation*, 67(2):102 – 120, 2010.
5. F. Bause, P. Buchholz, and J. Kriege. Profido - the processes fitting toolkit dortmund. In *2010 Seventh International Conference on the Quantitative Evaluation of Systems*, pages 87–96, Sept 2010.
6. M. Bramson. Stability of join the shortest queue networks. *Ann. Appl. Probab.*, 21(4):1568–1625, 2011.
7. M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. In *ACM SIGMETRICS 2010*, pages 275–286, 2010.
8. M. Bramson, Y. Lu, and B. Prabhakar. Asymptotic independence of queues under randomized load balancing. *Queueing Syst.*, 71(3):247–292, 2012.
9. G. Ciardo, A. Riska, and E. Smirni. Equiload: a load balancing policy for clustered web servers. *Performance Evaluation*, 46(2):101 – 124, 2001. Advanced Performance Modeling.
10. A. Economou and A. Manou. A probabilistic approach for the analysis of the M_n/G/1 queue. *Annals of Operations Research*, pages 1–9, 2015.
11. S. Ethier and T. Kurtz. *Markov processes: characterization and convergence.* Wiley, 1986.
12. H. Feng, V. Misra, and D. Rubenstein. Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems. *Perform. Eval.*, 62(1-4):475–492, Oct. 2005.
13. A. Gandhi, X. Zhang, and N. Mittal. HALO: heterogeneity-aware load balancing. In *23rd IEEE MASCOTS 2015, Atlanta, GA, USA, October 5-7, 2015*, pages 242–251, 2015.
14. N. Gast. Expected values estimated via mean-field approximation are 1/n-accurate. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1):17:1–17:26, June 2017.
15. N. Gast and B. Van Houdt. A refined mean field approximation. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2):33:1–33:28, Dec. 2017.
16. V. Gupta, M. Harchol Balter, K. Sigman, and W. Whitt. Analysis of join-the-shortest-queue routing for web server farms. *Performance Evaluation*, 64:1062–1081, 2007.
17. M. Harchol-Balter. Task assignment with unknown duration. *J. ACM*, 49(2):260–288, Mar. 2002.
18. M. Harchol-Balter, M. E. Crovella, and C. D. Murta. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, 59(2):204 – 228, 1999.
19. M. Harchol-Balter, A. Scheller-Wolf, and A. R. Young. Surprising results on task assignment in server farms with high-variability workloads. *SIGMETRICS Perform. Eval. Rev.*, 37(1):287–298, June 2009.
20. G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods and stochastic modeling.* SIAM, Philadelphia, 1999.
21. M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12:1094–1104, October 2001.
22. A. Mukhopadhyay and R. R. Mazumdar. Rate-based randomized routing in large heterogeneous processor sharing systems. In *26th International Teletraffic Congress (ITC), Karlskrona, Sweden*, pages 1–9, 2014.
23. J. F. Pérez and G. Riaño. jphase: An object-oriented tool for modeling phase-type distributions. In *Proceeding from the 2006 Workshop on Tools for Solving Structured Markov Chains*, SMCtools '06, New York, NY, USA, 2006. ACM.
24. N. Vvedenskaya, R. Dobrushin, and F. Karpelevich. Queueing system with selection of the shortest of two queues: an asymptotic approach. *Problemy Peredachi Informatsii*, 32:15–27, 1996.