Operations Research Letters 49 (2021) 459–464

Contents lists available at ScienceDirect

Operations Research Letters

www.elsevier.com/locate/orl

Optimal performance of parallel-server systems with job size prediction errors

Josu Doncel [a,*], Vincenzo Mancuso [b]

[a] University of the Basque Country, UPV/EHU, Leioa, Spain
[b] IMDEA Networks Institute, Madrid, Spain

ARTICLE INFO

Article history:
Received 24 July 2020
Received in revised form 3 March 2021
Accepted 4 May 2021
Available online 12 May 2021

Keywords:
Parallel-server systems
Size-based routing
Prediction errors

ABSTRACT

Modern communication networks integrate distributed computing architectures, in which customers are processed in parallel. We show how to minimize the waiting time of customer's jobs by leveraging a simple threshold-based job dispatching policy. The optimal policy leverages the SITA routing, which assigns jobs to servers according to the size of the job. Moreover, the optimal policy permits to optimize system performance even when the job size is not known a priori and is estimated by means of error-prone predictors.

1. Introduction

Communication networks have evolved towards the integration of connectivity and computing elements in a distributed platform, so as to offer the possibility to create, customize and tear down network services in a flexible way [11]. In particular, there exist many services and applications that require network connectivity with low and controllable latency, e.g., for the control of industrial processes in smart factories, the (remote) monitoring of patients using e-Health platforms, and the management of smart cities with vehicle platooning systems, just to name a few. The latency performance of the network resulting from the composition of connectivity devices (e.g., switches) and computing elements (e.g., edge cloud servers) mostly depends on the waiting time experienced by messages to exchange—for the connectivity part—and by data analysis tasks—for the computing part. These "connect-compute" jobs are served in parallel by multiple servers, they are strongly heterogeneous and incur variable delay, whose minimization strategy is the object of this work.

Specifically, we consider a system where jobs arrive according to a Poisson process to a dispatcher, e.g., a network switch. The dispatcher performs a load balancing task according to the Size-Interval Task Assignment (SITA) policy, in which the "size" of the task is the computing and/or data transmission time required to respond to the needs of the application that generated the job. In this routing policy, the job size distribution is divided in intervals and jobs whose sizes are in the same interval are executed in the same server. The main advantage of this routing policy is that the variability of jobs is reduced in the servers, which, when the servers are FCFS, leads to a performance improvement with respect to other routing policies such as the random assignment or Least-Work-Left. Another advantage of SITA is that, unlike other policies such as Power of Two, it does not require signaling between servers and router.

The implementation of SITA requires to determine which interval the size of each incoming job belongs to. Job sizes of incoming tasks are often not known, but they can be estimated using classification algorithms, using, e.g., machine learning tools [12]. However, here we are not interested in how to improve the accuracy of the prediction itself, but rather in its impact on system performance and on the optimal strategy to enforce when routing jobs towards the available servers. So, we consider that there is an oracle that predicts the interval that each job size is associated to. We consider that the oracle makes errors in the predictions and, therefore, it may occur that jobs whose sizes belong to different intervals are executed in the same server. This causes a performance degradation, which we tackle in this article.

In practice, the system designer will build a non-ideal error-prone classifier, and we show the impact of errors on network performance. Thus, the system designer could rely on our performance analysis in order to tune the desired accuracy of the classifier. We therefore assume that the classification error probabilities are known and fixed because a real system will have to learn how to classify incoming jobs and foresee their size. Given enough training, the learning process will reach the desired level

[*] Corresponding author at: University of the Basque Country, UPV/EHU, Barrio Sarriena s/n., 48940 Leioa, Spain.
E-mail address: josu.doncel@ehu.eus (J. Doncel).

https://doi.org/10.1016/j.orl.2021.05.007

of accuracy, if compatible with the Bayes error rate of the classifier. Therefore, after a training transient phase, the system will reach a steady state regime in which the classification error probability matches the target accuracy of the classification learning process.

We consider that the system designer is aware of prediction errors and can decide whether to trust or not the oracle. The strategy of the system designer consists in setting the optimal probability of trusting the oracle's prediction, i.e., the probability to minimize the average waiting time for the jobs in the system. For simplicity, we first consider a system with two servers and we show that the strategy of always trust the prediction and never trust the prediction lead to the same performance. We then consider that the router implements the SITA-E policy, which is a SITA policy that equalizes the load of both servers, and we show that, when prediction errors are present in both types of jobs, the optimal strategies consist again in always following or never following the prediction of the oracle. We then consider that there are prediction errors only in one type of jobs and, under a low load assumption, we characterize the optimal strategy, which is unique. We notice that the latter strategy can be provided as an approximation of the optimal strategy for an arbitrary arrival rate and we study its performance numerically. We also show how to extend the analysis to more than two servers.

2. Related work

The SITA policy is a size-based routing policy introduced in [10]. The basic idea of this policy consists of separating the execution of jobs of different sizes in different servers. The authors in [7] show that when the service demand is known but the queues are non-observable and the servers are FCFS, there exists a set of intervals such that the derived SITA policy minimizes the response time of jobs in the system. Some authors have characterized the intervals of the optimal SITA policy in the asymptotic regime of the Bounded Pareto distribution [13,3]. However, the characterization of these intervals in a general setting remains as an open question even in a two-server system [9]. Therefore, some authors focus on variants of the optimal SITA such as Equiload [5], TAGS [8,4] or the SITA policy that balances the performance of the queues [1].

3. Model description

We consider a system formed by two homogeneous servers and a single router. We assume that the service rate of both servers is equal to one. Servers are FCFS queues. We assume that job size incoming tasks form an i.i.d. sequence with a common distribution, where X denotes a generic job size. Let $F(x)=\mathbb{P}(X\leq x)$. We assume $F(x)$ to be differentiable and we write $f(x)=\frac{dF(x)}{dx}$. Let x_S and x_L be respectively the size of the shortest and the longest job. The incoming traffic arrives to the system according to a Poisson process of rate λ. We denote by X_i the random variable of jobs executed in server i.

3.1. SITA routing

We assume that the router performs the load balancing task using a size-based policy called Size Interval Task Assignment (SITA). Under this policy, there is threshold value x such jobs whose size is smaller than x are sent to Server 1 and jobs whose size is larger than x to Server 2. Roughly speaking, short jobs are executed in one server and long jobs in the other one. We assume that $x_L<\infty$, which is a common assumption of the literature of SITA policies (even though our results generalize to distributions with unbounded support). Therefore, it follows from this definition that the probability for a job to be executed in Server 1 is given by $\bar{q}_1(x)=\int_{x_S}^{x} f(z)dz$ and in Server 2 by $\bar{q}_2(x)=\int_{x}^{x_L} f(z)dz=1-\bar{q}_1(x)$. The arrival rate to Server 1 is $\lambda\bar{q}_1(x)$ and to Server 2 is $\lambda\bar{q}_2(x)$. Using conditional probabilities, we obtain that the mean service time in Server 1 is $\mathbb{E}[X_1]=\int_{x_S}^{x} z\frac{f(z)}{F(x)-F(x_S)}dz$ and in Server 2 it is $\mathbb{E}[X_2]=\int_{x}^{x_L} z\frac{f(z)}{F(x_L)-F(x)}dz$. Therefore, using that $\bar{q}_1(x)=F(x)-F(x_S)$, it results that the load of Server 1 is

$$\lambda\bar{q}_1(x)\int_{x_S}^{x}\frac{zf(z)dz}{F(x)-F(x_S)}=\lambda\int_{x_S}^{x} zf(z)dz,$$

and, using that $\bar{q}_2(x)=F(x_L)-F(x)$, the load of Server 2 is $\lambda\bar{q}_2(x)\int_{x}^{x_L}\frac{zf(z)dz}{F(x_L)-F(x)}=\lambda\int_{x}^{x_L} zf(z)dz$.

Using the Pollaczek-Kinchine formula, the mean waiting time of a system with two servers operating under the SITA policy is given by

$$\bar{g}(x)=\frac{\lambda}{2}\left(\frac{\bar{q}_1(x)\bar{v}_1(x)}{1-\lambda\bar{b}_1(x)}+\frac{\bar{q}_2(x)\bar{v}_2(x)}{1-\lambda\bar{b}_2(x)}\right), \tag{1}$$

where $\bar{v}_1(x)=\int_{x_S}^{x} z^2f(z)dz$, $\bar{b}_1(x)=\int_{x_S}^{x} zf(z)dz$, $\bar{v}_2(x)=\int_{x}^{x_L} z^2f(z)dz$ and $\bar{b}_2(x)=\int_{x}^{x_L} zf(z)dz$.

3.2. Prediction errors

We assume that there is an oracle that predicts the job size of the incoming jobs and the router balances the traffic according to the SITA routing. Besides, we consider that the oracle makes errors in the predictions. More precisely, a short job is predicted as a long job with probability p_1 a long job as a short job with probability p_2. Hence, for this case, a proportion p_1 of short jobs are executed in Server 2 and a proportion p_2 of long jobs to Server 1. Hence, the mean waiting time of the system is given by

$$\tilde{g}(p_1,p_2,x)=\frac{\lambda}{2}\left(\frac{\tilde{q}_1(p_1,p_2,x)\tilde{v}_1(p_1,p_2,x)}{1-\lambda\tilde{b}_1(p_1,p_2,x)}+\frac{\tilde{q}_2(p_1,p_2,x)\tilde{v}_2(x)}{1-\lambda\tilde{b}_2(p_1,p_2,x)}\right), \tag{2}$$

where $\tilde{q}_1(p_1,p_2,x)=\bar{q}_1(x)(1-p_1)+\bar{q}_2(x)p_2$, $\tilde{v}_1(p_1,p_2,x)=\bar{v}_1(x)(1-p_1)+\bar{v}_2(x)p_2$, $\tilde{b}_1(p_1,p_2,x)=\bar{b}_1(x)(1-p_1)+\bar{b}_2(x)p_2$, $\tilde{q}_2(p_1,p_2,x)=\bar{q}_2(x)(1-p_2)+\bar{q}_1(x)p_1$, $\tilde{v}_2(p_1,p_2,x)=\bar{v}_2(x)(1-p_2)+\bar{v}_1(x)p_1$ and $\tilde{b}_2(p_1,p_2,x)=\bar{b}_2(x)(1-p_2)+\bar{b}_1(x)p_1$.

Before going further, let us present two nice properties of this model. First, we note that

$$\bar{g}(x)=\tilde{g}(0,0,x)=\tilde{g}(1,1,x), \forall x\in[x_S,x_L], \tag{3}$$

which means that the performance of the system with perfect predictions (i.e. when $p_1=p_2=0$) and with completely incorrect predictions (i.e., when $p_1=p_2=1$) is the same. In other words, (3) says that swapping from an oracle with perfect predictions to an oracle with completely wrong predictions does not change the performance of the system. Another interesting property of this model is the existence of a performance degradation when there are prediction errors, that is,

$$\bar{g}(x)\leq\tilde{g}(p_1,p_2,x), \forall x\in[x_S,x_L]. \tag{4}$$

We assume that the values of p_1 and p_2 are known and fixed. We also assume that the value of the threshold x is chosen such that the load of both servers is equal. We denote by x^E such a threshold value and by λb the load of each server, i.e., $b=\tilde{b}_1(p_1,p_2,x^E)=\tilde{b}_2(p_1,p_2,x^E)$. Therefore, the threshold value is also fixed and it is omitted its dependence in the above function. As a result, (2) gives

$$\frac{\lambda(q_1 v_1 + q_2 v_2)}{2(1-\lambda b)}, \tag{5}$$

where $q_1 = \tilde{q}_1(p_1, p_2, x^E)$, $v_1 = \tilde{v}_1(p_1, p_2, x^E)$, $q_2 = \tilde{q}_2(p_1, p_2, x^E)$ and $v_2 = \tilde{v}_2(p_1, p_2, x^E)$. We remark that, in the above expression, we have omitted the dependence of p_1, p_2 and x^E to avoid heavy notations since they are fixed.

3.3. Strategies

We consider that the system designer can decide to trust or not the prediction of the oracle. Therefore, the strategy of the system designer is defined as the probability α to trust the prediction of the oracle and its goal is to find the value of α that optimizes the performance of the system, i.e., that minimizes the waiting time.

Using (4), it follows that, if the predictions are perfect, i.e., $p_1 = p_2 = 0$, then the optimal strategy is to always trust the prediction of the oracle, i.e., $\alpha = 1$, in which case all the short jobs are executed in Server 1 and the long jobs in Server 2. Besides, from (3), the strategy $\alpha = 0$ is also optimal in the case of perfect predictions, because $\alpha = 0$ corresponds to swapping the two servers, so as to run all the long jobs in Server 1 and the short ones in Server 2.

Remark 1. Using the same arguments as above, one can also show that the strategy $\alpha = 0$ or $\alpha = 1$ is also optimal when $p_1 = p_2 = 1$.

In Section 4, we study the optimal strategy for the system designer when there are errors in the prediction of the job sizes that the oracle performs, that is, the strategy that minimizes the mean waiting time of the system. Prior to that, we present our main assumptions on the job size distribution.

3.4. Assumptions

Besides, we know from Lemma 3 of [6] that $q_1 \geq q_2$ and $v_1 \leq v_2$ when the SITA-E policy is implemented. In this model, we assume that (i) $q_1 > q_2$ and (ii) $v_1 < v_2$. As we will see now, making these assumptions is equivalent to consider a non-deterministic job size distribution.

We first focus on (i). Let us consider that $q_1 = q_2$. Since the load of the servers is the same, we have that

$$\lambda q_1 \mathbb{E}[X_1] = \lambda q_2 \mathbb{E}[X_2] \iff \mathbb{E}[X_1] = \mathbb{E}[X_2],$$

which means that the mean of the job size distribution of jobs executed in both servers is the same. This is impossible for SITA type systems (note that jobs of different sizes are executed in different servers) unless the size of all the incoming jobs is equal. Therefore, from the above considerations, it is clear that $q_1 > q_2$ when the job size distribution is non-deterministic, and the case $q_1 = q_2$ holds for the deterministic case.

We now concentrate on (ii). If $v_1 = v_2$, since all the servers are equivalent and x^2 is an increasing function, we have that

$$v_2 = \int_{x^E}^{x_L} z^2 f(z)dz \geq x^E \int_{x^E}^{x_L} z f(z)dz = x^E \int_{x_S}^{x^E} z f(z)dz \geq \int_{x_S}^{x^E} z^2 f(z)dz = v_1$$

which implies that the above inequalities are equalities, which is only possible if $x_L = x^E = x_S$, that is, the job size distribution is deterministic. Therefore, from the above considerations, it is clear that $v_1 < v_2$ when the job size distribution is non-deterministic, and the case $v_1 = v_2$ holds for the deterministic case.

4. Analysis for $p_1 \neq 0$ and $p_2 \neq 0$

In this section, we assume that the oracle makes errors when it predicts short and long jobs, i.e., $p_1 \neq 0$ and $p_2 \neq 0$. Besides, we consider that the system designer trusts the prediction of the oracle with probability α, regardless of the job has been predicted as short or long. Under these assumptions, the arrival rate at Server 1 becomes $(\alpha q_1 + (1-\alpha) q_2)\lambda$, and the rest goes to Server 2. Therefore, the performance of the two-server system described so far (i.e., the average waiting time experienced by jobs dispatched to either server 1 or Server 2 with SITA routing and prediction errors on both short and long jobs) is

$$h_2(\alpha) = \frac{\lambda}{2}\left(\frac{(\alpha q_1 + (1-\alpha)q_2)(\alpha v_1 + (1-\alpha)v_2)}{1-\lambda(\alpha b + (1-\alpha)b)} + \frac{((1-\alpha)q_1 + \alpha q_2)((1-\alpha)v_1 + \alpha v_2)}{1-\lambda((1-\alpha)b + \alpha b)}\right). \tag{6}$$

We note that the above expression can be simplified as follows:

$$h_2(\alpha) = \frac{\lambda}{2(1-\lambda b)}((\alpha q_1 + (1-\alpha)q_2)(\alpha v_1 + (1-\alpha)v_2) + ((1-\alpha)q_1 + \alpha q_2)((1-\alpha)v_1 + \alpha v_2)). \tag{7}$$

We notice that, as we said before, $h_2(0) = h_2(1)$ that is, the mean waiting time of the system when we always follow the strategy of the oracle and when we never follow the prediction is the same.

We denote by $\alpha_2^* \in \operatorname{argmin} h_2(\alpha)$. After some simplifications, the sign of the derivative with respect to α of the above expression is the same as the sign of $\lambda(q_1 - q_2)(v_1 - v_2)(2\alpha - 1)$. From the assumptions of Section 3.4, the above expression is zero when $\alpha = 0.5$. Besides, from our assumptions, we conclude that the above expression is increasing with α when $\alpha < 0.5$ and decreasing when $\alpha > 0.5$. Therefore, since $h_2(0) = h_2(1)$, the following result is proven.

Proposition 1. *The optimal strategies consist in either always follow the prediction of the oracle or never follow the prediction of the oracle, i.e., $\alpha^* = 0$ and $\alpha^* = 1$ are both optimal.*

From the above result, it follows that the mean waiting time under the optimal strategy in a system with prediction errors is given in (5).

5. Approximate analysis for $p_1 \in (0, 1)$ and $p_2 = 0$

We assume that the oracle makes errors when it predicts short jobs and it makes perfect predictions for long jobs, i.e., $p_1 \in (0, 1)$ and $p_2 = 0$. The system designer is aware of this prediction errors and, therefore, when the oracle predicts a job as short, it always trusts the result. However, when the oracle predicts a job as long, the prediction is accepted with probability α. For this case, the performance of the system is given by

$$h_1(\alpha) = \frac{\lambda}{2}\left(\frac{(q_1 + q_2(1-\alpha))(v_1 + v_2(1-\alpha))}{1-\lambda(b + b(1-\alpha))} + \frac{q_2 v_2 \alpha^2}{1-\lambda b\alpha}\right) \tag{8}$$

We denote by α_1^* a strategy that is optimal for this case, i.e., $\alpha_1^* \in \operatorname{argmin} h_1(\alpha)$. Given the difficulty of the derived expression, we did not succeed in obtaining an analytical expression of the value of α_1^*. Instead, we analyze the value of α that minimizes the following expression:

$$\frac{\lambda}{2}\left((q_1+q_2(1-\alpha))(v_1+v_2(1-\alpha))+q_2v_2\alpha^2\right) \qquad \text{(LT-APP)}$$

We now explain that the above expression consists of the first order Taylor expansion of (8) at $\lambda=0$. Therefore, it can be seen as a light-traffic approximation of the mean waiting time of the system. We analyze the accuracy of this approximation in the numerical section.

We denote by $\bar{\alpha}_1$ the value of α that minimizes (LT-APP). The sign of the derivative with respect to α of (LT-APP) is the same as the sign of

$$\begin{aligned}-q_2v_1-q_1v_2-2(1-\alpha)q_2v_2+2\alpha q_2v_1 &=\\ -q_2v_1-q_1v_2-2q_2v_2+4\alpha q_2\tilde{v}_2. &\quad (9)\end{aligned}$$

This expression is clearly negative when $\alpha=0$. Therefore, since it is linear on α, it is negative for all $\alpha\in[0,1]$ if and only if it is negative when $\alpha=1$, i.e., if $-q_2v_1-q_1v_2-2q_2v_2+4q_2v_2<0$. Rearranging both sides of this expression it is easy to see that this condition is equivalent to the following one: $\frac{v_1}{v_2}+\frac{q_1}{q_2}>2$.

As a result, the minimum of (9) over $\alpha\in[0,1]$ is achieved when $\alpha=1$ when the above condition is satisfied. We now consider that $\frac{v_1}{v_2}+\frac{q_1}{q_2}\le 2$, in which case if (9) equals zero, it results $\alpha=\frac{1}{4}\left(\frac{v_1}{v_2}+\frac{q_1}{q_2}+2\right)$.

We remark that when $\frac{v_1}{v_2}+\frac{q_1}{q_2}=2$, the above expression gives $\alpha=1$. As a result of the reasoning, it follows the next result.

Proposition 2. *If the oracle makes errors only in the prediction of short jobs, then*

$$\bar{\alpha}_1=\min\left\{1,\frac{1}{4}\left(\frac{v_1}{v_2}+\frac{q_1}{q_2}+2\right)\right\}.$$

This result means that, at low load, the optimal strategy for the system designer is to always follow the prediction of the oracle if $\frac{v_1}{v_2}+\frac{q_1}{q_2}$ is larger or equal than two and, otherwise, the optimal probability is to follow the prediction of the oracle with probability $\frac{1}{4}\left(\frac{v_1}{v_2}+\frac{q_1}{q_2}+2\right)$.

5.1. The case $p_1=0$ and $p_2\in(0,1)$

The case $p_1=0$ and $p_2\in(0,1)$ is symmetric to the case we study above. Therefore, using the same arguments, one can easily derive that $\bar{\alpha}_1=\min\left\{1,\frac{1}{4}\left(\frac{v_2}{v_1}+\frac{q_2}{q_1}+2\right)\right\}$. This means that, for $p_1=0$ and $p_2\neq 0$ and at low load, the optimal strategy consists of always following the prediction of the oracle if $\frac{v_2}{v_1}+\frac{q_2}{q_1}$ is larger or equal than 2 and, otherwise, the optimal probability is to follow the prediction of the oracle with probability $\frac{1}{4}\left(\frac{v_2}{v_1}+\frac{q_2}{q_1}+2\right)$.

6. Extensions

6.1. More than two servers

In a system formed by K servers, there are $K-1$ thresholds $x_1,\ldots,x_{K-1}$ and, with perfect predictions, a job sent to server i if its size is between x_{i-1} and x_i. For this case, the modeling of the prediction errors becomes very difficult. To see this, let us consider $K=3$. For this case, there are short, medium and large jobs and, in the system with predictions errors, jobs that are short can be predicted as medium or large, medium jobs as short or large and large jobs as short or medium. Hence, for an arbitrary K, the huge amount of possibilities that arise shows the difficulty the extension of the above results to more than two servers.

However, we now present how the above results can be applied to a system with $2K$ servers. The system is divided in two group of servers, each of them formed by K servers and a front-end router. We assume that the front-end router of each group of servers receives all the traffic to be executed and sends the same proportion of traffic to each server of that type. We note that traffic received by each front-end router is also a Poisson process. Therefore, since each server executes a proportion of $1/K$ of the traffic, the mean waiting time of the system with 2K servers is obtained replacing λ by λ/K in (5). And, therefore, it is clear that the main results of this article for two servers also hold for this case.

6.2. Approximated analysis with asymmetric strategies for $p_1\in(0,1)$ and $p_2\in(0,1)$

In this section, we consider that, when the oracle predicts a job as long, the prediction is accepted with probability α_1, whereas when the oracle predicts a job as short, the prediction is accepted with probability α_2. For this case, the mean waiting time of the system is given by

$$\frac{\lambda}{2}\left(\frac{(\alpha_1q_1+(1-\alpha_2)q_2)(\alpha_1v_1+(1-\alpha_2)v_2)}{1-\lambda(\alpha_1b+(1-\alpha_2)b)}+\frac{((1-\alpha_1)q_1+\alpha_2q_2)((1-\alpha_1)v_1+\alpha_2v_2)}{1-\lambda((1-\alpha_1)b+\alpha_2b)}\right).$$

As we do in (LT-APP), we use the first order Taylor to say that, at low load, the above expression is approximately $\frac{\lambda}{2}((\alpha_1q_1+(1-\alpha_2)q_2)(\alpha_1v_1+(1-\alpha_2)v_2)+((1-\alpha_1)q_1+\alpha_2q_2)((1-\alpha_1)v_1+\alpha_2v_2))$. Simple calculus arguments show that the minimum of this expression in which $\alpha_1=\alpha_2$ is given when α_1 and α_2 are both zero and when they are both one.

6.3. Approximated analysis with optimal SITA

So far, we have assumed that the router implements the SITA-E policy, that is, the threshold that determines the short and large jobs is chosen in a way that the load of both servers is the same. The threshold could be also chosen optimally, that is, to minimize the mean waiting time of the system, which, if x^* is the optimal threshold, is

$$\frac{\lambda}{2}\left(\frac{(\alpha q_1^*+(1-\alpha)q_2^*)(\alpha v_1^*+(1-\alpha)v_2^*)}{1-\lambda(\alpha b_1^*+(1-\alpha)b_2^*)}+\frac{((1-\alpha)q_1^*+\alpha q_2^*)((1-\alpha)v_1^*+\alpha v_2^*)}{1-\lambda((1-\alpha)b_1^*+\alpha b_2^*)}\right), \quad (10)$$

where $q_1^*=\tilde{q}_1(p_1,p_2,x^*)$, $v_1^*=\tilde{v}_1(p_1,p_2,x^*)$, $b_1^*=\tilde{b}_1(p_1,p_2,x^*)$, $q_2^*=\tilde{q}_2(p_1,p_2,x^*)$, $v_2^*=\tilde{v}_2(p_1,p_2,x^*)$ and $b_2^*=\tilde{b}_2(p_1,p_2,x^*)$. We now explain that, for this case, as we do in (LT-APP), one can use the first order Taylor to derive that, at low load, the derivative of the mean waiting time has the same sign of

$$\lambda(q_1^*-q_2^*)(v_1^*-v_2^*)(2\alpha-1).$$

Hence, from the above expression, we conclude that, for the SITA with optimal threshold and at low load, the strategy that minimizes the mean waiting time is (a) $\bar{\alpha}_2=0.5$ if $(q_1^*-q_2^*)(v_1^*-v_2^*)$ is positive and (b) $\bar{\alpha}_2=0$ and $\bar{\alpha}_2=1$ otherwise. We compare the performance of the system under this strategy and under the strategy α_2^* that minimizes the mean waiting time in the numerical section.

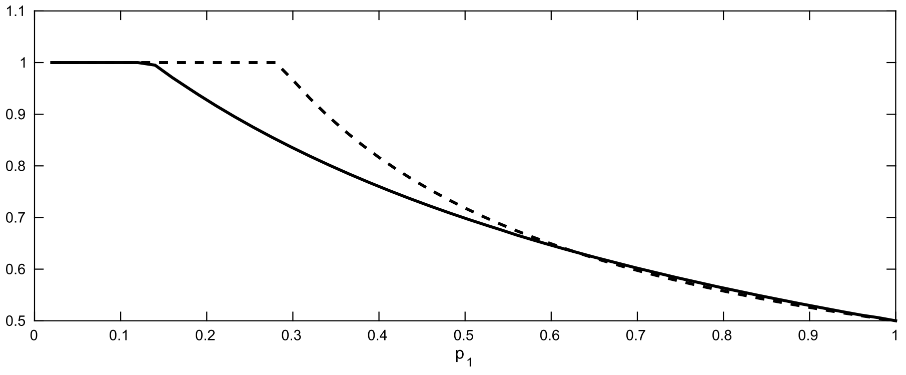

Fig. 1. Comparison of $\bar{\alpha}_1$ (dashed line) and α_1^* (solid line) for different values of p_1.

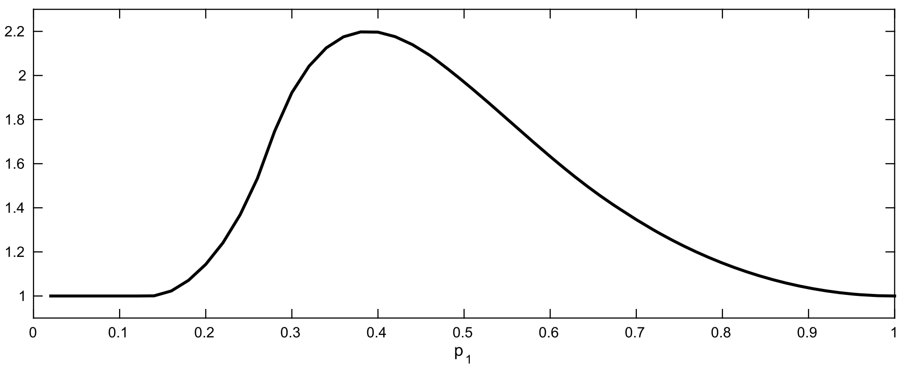

Fig. 2. Evolution over p_1 of $\frac{h_1(\bar{\alpha}_1)}{h_1(\alpha_1^*)}$.

7. Numerical experiments

We now present the numerical experiments we have carried out in this work. In all the experiments, we consider the same setting as in Figures 4-6 of [7] and Figures 5-7 of [2], that is, the Bounded Pareto distribution with shape parameter 1.5, $\mathbb{E}[X] = 1$ and $x_L/x_S = 10^4$. For this setting, we get that $x^E = 1.3203$ and that the optimal threshold is 1.9490.

We focus on the approximated analysis of Section 5. In Fig. 1, we represent by a solid line the strategy α_1^*, which and by a dashed line the strategy $\bar{\alpha}_1$. We recall that α_1^* is the strategy that minimizes $h_1(\alpha)$ and $\bar{\alpha}_1^*$ is the strategy that minimizes (LT-APP). As it can be observed, the range of values for which for strategies coincide is very large. Indeed, it coincides with $\bar{\alpha}_1$ when p_1 is very small and very large, whereas if $0.14 \leq p_1 \leq 0.28$, then $\bar{\alpha}_1 = 1$ and α_1^* is smaller than 1. In Fig. 2, we represent the value of $\frac{h_1(\bar{\alpha}_1)}{h_1(\alpha_1^*)}$ for different values of p_1. We observe that, for the values of p_1 where the strategy α_1^* is optimal, we have that $\frac{h_1(\bar{\alpha}_1)}{h_1(\alpha_1^*)} = 1$. On the other hand, when α_1^* and $\bar{\alpha}_1$ do not coincide, we see that the value of $\frac{h_1(\bar{\alpha}_1)}{h_1(\alpha_1^*)}$ is at most 2.2, which means that the strategy $\bar{\alpha}_1$ is almost optimal. We consider the SITA policy with optimal thresholds when $p_1 \neq 0$ and $p_2 \neq 0$. In Section 6.3, we explain that, at low load, the strategy that minimizes the mean waiting time is $\bar{\alpha}_2 = 0.5$ if $(q_1 - q_2)(v_1 - v_2)$ is positive and $\bar{\alpha}_2 = 0$ and $\bar{\alpha}_2 = 1$ otherwise. We aim to compare the performance of the system under strategy $\bar{\alpha}_2$ with the performance under α_2^*, which is the optimal strategy, i.e., $\alpha_2^* \in$ argmin $h_2(\alpha)$. In Fig. 3, we represent the performance ratio $\frac{h_2(\bar{\alpha}_2)}{h_2(\alpha_2^*)}$ for different values of p_1 and p_2. We observe that, for all the values of p_1 and p_2, the value of the ratio $\frac{h_2(\bar{\alpha}_2)}{h_2(\alpha_2^*)}$ is very close to 1, hence the approximation is almost optimal for almost all the values under consideration even if the value of the arrival rate under consideration is 1.5.

8. Conclusion

We have characterized the performance of a system with two or more parallel servers that receive network jobs according to a SITA policy, i.e., their load is balanced by routing jobs to servers according to job sizes, so as to make job sizes as homogeneous as possible within the same server. Even if servers are meant to deal with jobs of similar size, so as to reduce the waiting time in the overall system, it is possible that jobs are erroneously dispatched, so that the system suffers performance degradations. This is possible because the job size is not necessarily know at service request time. Nonetheless, the analysis shown in this work leads to the identification of a simple probabilistic strategy that achieves near-minimal job waiting time when using error-prone job size predictors, and which tends to the optimal strategy when the load becomes low. We have validated the assumptions and approximations introduced in the analysis by means of numerical experiments, which also show that the near-optimal policy achieves results very close to ones of the optimal policy not only under low load conditions, but also when the load becomes high.

Acknowledgements

The work of Josu Doncel has been supported by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1294-19), by the Marie Sklodowska-Curie grant agreement No 777778 and by the Spanish Ministry of Science and Innovation with reference PID2019-108111RB-I00 (FEDER/AEI). The work of Vincenzo Mancuso has been supported by the Ramon y Cajal grant RYC-2014-16285 from the Spanish Ministry of Economy and Competitiveness, and by

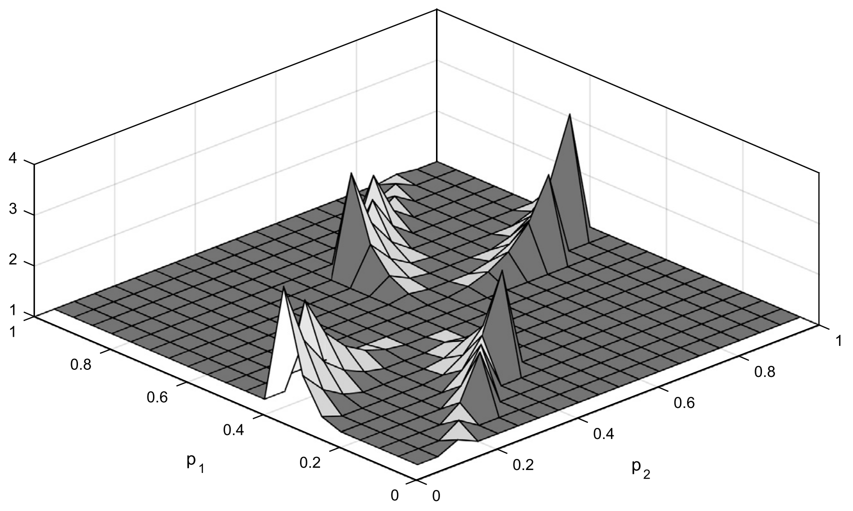

Fig. 3. Evolution over p_1 and p_2 of $\frac{h_2(\tilde{\alpha}_2)}{h_2(\alpha_2^*)}$.

the Region of Madrid through the TAPIR-CM program (S2018/TCS-4496).

References

[1] M. Abidini, O. Boxma, J. Doncel, Size-based routing to balance performance of the queues, in: Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools, ACM, 2017, pp. 198–205.
[2] J. Anselmi, J. Doncel, Asymptotically optimal size-interval task assignments, IEEE Trans. Parallel Distrib. Syst. 30 (11) (2019) 2422–2433, https://doi.org/10.1109/TPDS.2019.2920121.
[3] E. Bachmat, H. Sarfati, Analysis of SITA policies, Perform. Eval. 67 (2) (2010) 102–120.
[4] E. Bachmat, J. Doncel, H. Sarfati, Analysis of the task assignment based on guessing size policy, Perform. Eval. (2020) 102122.
[5] G. Ciardo, A. Riska, E. Smirni, Equiload: a load balancing policy for clustered web servers, Perform. Eval. 46 (2–3) (2001) 101–124.
[6] J. Doncel, S. Aalto, U. Ayesta, Performance degradation in parallel-server systems, IEEE/ACM Trans. Netw. 27 (2) (2019) 875–888, https://doi.org/10.1109/TNET.2019.2902531.
[7] H. Feng, V. Misra, D. Rubenstein, Optimal state-free, size-aware dispatching for heterogeneous m/g/-type systems, Perform. Eval. 62 (1) (2005) 475–492, performance 2005.
[8] M. Harchol-Balter, Task assignment with unknown duration, in: Proceedings 20th IEEE International Conference on Distributed Computing Systems, IEEE, 2000, pp. 214–224.
[9] M. Harchol-Balter, R. Vesilo, To balance or unbalance load in size-interval task allocation, Probab. Eng. Inf. Sci. 24 (2) (2010) 219–244.
[10] M. Harchol-Balter, M.E. Crovella, C.D. Murta, On choosing a task assignment policy for a distributed server system, J. Parallel Distrib. Comput. 59 (2) (1999) 204–228.
[11] K. Kaur, S. Kumar, A. Baliyan, 5G: a new era of wireless communication, International Journal of Information Technology 12 (2) (2020) 619–624.
[12] P.C. Sen, M. Hajra, M. Ghosh, Supervised classification algorithms in machine learning: a survey and review, in: J.K. Mandal, D. Bhattacharya (Eds.), Emerging Technology in Modelling and Graphics, Springer Singapore, Singapore, 2020, pp. 99–111.
[13] R. Vesilo, Asymptotic analysis of load distribution for size-interval task allocation with bounded Pareto job sizes, in: 2008 14th IEEE International Conference on Parallel and Distributed Systems, 2008, pp. 129–137.