Is the Price of Anarchy the Right Measure for Load-Balancing Games?

JOSU DONCEL, LAAS-CNRS and Université de Toulouse
URTZI AYESTA, LAAS-CNRS, Université de Toulouse, IKERBASQUE, and UPV/EHU
OLIVIER BRUN and BALAKRISHNA PRABHU, LAAS-CNRS and Université de Toulouse

Price of anarchy is an oft-used worst-case measure of the inefficiency of noncooperative decentralized architectures. For a noncooperative load-balancing game with two classes of servers and for a finite or infinite number of dispatchers, we show that the price of anarchy is an overly pessimistic measure that does not reflect the performance obtained in most instances of the problem. We explicitly characterize the worst-case traffic conditions for the efficiency of noncooperative load-balancing schemes and show that, contrary to a common belief, the worst inefficiency is in general not achieved in heavy traffic.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Performance

Additional Key Words and Phrases: Load balancing, game theory, inefficiency, price of anarchy, atomic games

ACM Reference Format:
Josu Doncel, Urtzi Ayesta, Olivier Brun, and Balakrishna Prabhu. 2014. Is the price of anarchy the right measure for load-balancing games? ACM Trans. Internet Technol. 14, 2-3, Article 18 (October 2014), 20 pages.
DOI: http://dx.doi.org/10.1145/2663498

1. INTRODUCTION

Server farms are commonly used in a variety of applications, including cluster computing, Web hosting, scientific simulation, or even the rendering of 3D computer-generated imagery. A central problem arising in the management of the distributed computing resources of a datacenter is that of balancing the load over the servers so that the overall performance is optimized.[1] In a centralized architecture, a single dispatcher or routing agent routes incoming jobs to a set of servers so as to optimize a certain performance objective, such as the mean processing time of jobs, for instance. However, modern data centers commonly have thousands of processors and up, hence it becomes difficult or even impossible to centrally implement a globally optimal load-balancing solution. For instance, Akamai Technologies revealed in March of 2012 that it operates 105,000 servers [Miller 2009b]. Similarly, it is estimated that Google has more than 900,000 servers and the company recently revealed that the container data center holds more than 45,000 servers in a single facility built in 2005 [Miller 2009a]. The ever-growing size and complexity of modern server farms thus call for decentralized control schemes.

In a decentralized routing architecture, several dispatchers are used, each routing a certain portion of the traffic. There are several possible approaches for the

[1]We shall use the terms load-balancing and routing interchangeably.

This work has been partially supported by grant ANR-11-INFR-001.
Authors' addresses: J. Doncel (corresponding author), U. Ayesta, O. Brun, and B. Prabhu, LAAS-CNRS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France and Université de Toulouse, LAAS, F-31400, Toulouse, France; email: {j.doncel, urtzi, brun, bala}@laas.fr.

DOI: http://dx.doi.org/10.1145/2663498

implementation of decentralized routing mechanisms. Approaches based on distributed optimization techniques [Bertsekas and Tsitsiklis 1989; Mosk-aoyama et al. 2010] can be cumbersome to implement and can have significant synchronisation and communication overheads, thus reducing the scalability of the decentralized routing scheme.

An alternative approach is based on autonomous, self-interested agents [Roughgarden 2005]. Such routing schemes are also known as "selfish routing", since each dispatcher independently seeks to optimize the performance perceived by the jobs it routes. This setting can be analysed within the framework of a noncooperative routing game. The strategy that rational agents will choose under these circumstances is called a Nash equilibrium and is such that a unilateral deviation will not help any routing agent in improving the performance perceived by the traffic it routes. When the number of dispatchers grows to infinity (every incoming job is handled by a dispatcher and takes its own routing decision), the corresponding equilibrium is given by the notion of Wardrop equilibrium [Wardrop 1952], where journey times are minimal and equal in all routes.

Apart from the obvious gain in scalability with respect to a centralized setting, there are wide-ranging advantages to noncooperative routing schemes: ease of deployment, no need for coordination between the routing agents that just react to the observed performances of the servers, and robustness to failures and environmental disturbances. However, it is well known that noncooperative routing mechanisms are potentially inefficient. Indeed, in general, the Nash equilibrium resulting from the interactions of many self-interested routing agents with conflicting objectives does not correspond to an optimal routing solution; hence, the lack of regulation carries the cost of decreased overall performance.

A standard measure of the inefficiency of selfish routing is the Price of Anarchy (PoA) introduced by Koutsoupias and Papadimitriou [1999]. It is defined as the ratio between the performance obtained by the worst Nash equilibrium and the global optimal solution. Thus the PoA measures the cost of having no central authority, irrespective of a specific data center architecture. A value of the PoA close to 1 indicates that, in the worst case, the gap between a Nash equilibrium and the optimal routing solution is not significant and thus good performances can be achieved even without a centralized control. On the contrary, a high PoA value indicates that, under certain circumstances, the selfish behaviour of the dispatchers leads to a significant performance degradation.

Several recent works have shown that noncooperative load balancing can be very inefficient in the presence of nonlinear delay functions; see, for example, Haviv and Roughgarden [2007], Bell and Stidham [1983], Ayesta et al. [2011], Suri et al. [2004], and Chen et al. [2009]. We just mention two of them here. First, Haviv and Roughgarden [2007] have considered the so-called nonatomic scenario where every arriving job can select the server in which it will be served. They have shown that, in this scenario, the PoA corresponds to the number of servers, implying that, in a server farm with S servers, the mean response time of jobs can be as high as S times the optimal one! Another important result on the PoA was proved by Ayesta et al. [2011]. They investigate the price of anarchy of a load-balancing game with a finite number, say K, of dispatchers, the so-called atomic case, and with a price per unit time to be paid for processing a job, which depends on the server. They prove that, for a system with two or more servers, the price of anarchy is of the order of $\sqrt{K}$, independently of the number of servers, implying that when the number of dispatchers grows large, the PoA grows unboundedly. The fact that the Nash equilibrium can be very inefficient has paved the way to a lot of research on mechanism design that aims at coming up with Nash equilibria that are efficient with respect to the centralized setting [Korilis et al. 1997, 2006; Roughgarden 2005, 2009; Christodoulou and Koutsoupias 2005].

In this article, we adopt the view that the worst-case analysis (PoA) of the inefficiency of selfish routing is overly pessimistic and that high PoAs are obtained in pathological instances that hardly occur in practice. For example, in Haviv and Roughgarden [2007], the worst-case architecture has one server whose capacity is much larger (tending to infinity) compared to that of the other servers. It is doubtful that such asymmetries will occur in data-centers where processors are more than likely to have similar characteristics.

While the architecture of a data-center is more or less fixed, the incoming traffic volume can vary as a function of time. Thus, for applications such as data-centers, it seems more appropriate to compare the performance of selfish routing and the centralized setting for different traffic profiles and a *fixed data-center architecture* (number of servers and their capacities). For this reason, we define the *inefficiency* for a fixed architecture of a data-center as the performance ratio between the worst-case Nash equilibrium and the global optimal. The worst-case scenario is taken over all possible traffic profiles that the routing agents can be asked to route. As is true of the PoA, inefficiency can take values between 1 and ∞. A higher value of inefficiency indicates a worse performance of selfish routing compared to centralized routing. As opposed to the PoA, the inefficiency depends on the parameters (the server speeds and the number of servers in our case) of the architecture. By calculating the worst possible inefficiency, one retrieves the PoA.

The main contributions in this work are the following.

—For an arbitrary architecture in the system, we characterize the traffic conditions (or load) associated with the inefficiency. Contrary to classical queueing theory, we show that the inefficiency is in general not achieved in heavy traffic or close to saturation conditions. In fact, we show that the inefficiency is close to 1 in heavy traffic. We also provide examples for which it is obtained for fairly low values of the utilization rate.
—We observe (see beginning of Section 4) that, for each problem instance with arbitrary server capacities, we can construct a scenario with two server speeds whose inefficiency is worse. We thus conjecture that the case of two classes of servers is the worst and that our conclusions on the efficiency of noncooperative load balancing extend beyond this case.
—In the case of two server classes, we show the inefficiency is obtained when selfish routing uses only one class of servers and is marginally using the second class. This scenario was used in Haviv and Roughgarden [2007] and Ayesta et al. [2011] to obtain a lower bound on the PoA for their models. We give a formal proof on why this is indeed the worst-case scenario for selfish routing. Further, we obtain a closed-form formula for the inefficiency that, in particular, depends only on the ratio of the number of servers in each class and on that of the capacities of each class (but not on the total nor on their capacities). When the number of servers is large, we also show that the PoA is equal to $\frac{K}{2\sqrt{K}-1}$, where K is the number of dispatchers.
—We then show that the inefficiency is very close to 1 in most cases and approaches the known upper bound (given by the PoA) only in a very specific setting, namely, when there is only one fast server that is infinitely faster than the slower ones.
—For an infinite number of dispatchers, namely, the nonatomic case, and an arbitrary architecture of the system, we show that the performance of the decentralized and the centralized settings are not equal in the heavy-traffic regime, in contrast with the case of a finite number of dispatchers. For the case of two servers classes, we give an expression of the inefficiency that depends only on the ratio of the number of servers in each class and on the ratio of the capacities of each class. We show that the PoA equals the number of servers, which coincides with the result presented in

Haviv and Roughgarden [2007], and that the PoA is achieved only when there is only one fast server infinitely faster than the slower ones. In all the other configurations, we observe the inefficiency is very close to one.

We believe that our work opens a new avenue in the study of the PoA and hope that future research will be done not only on the PoA, but also on the inefficiency.

The rest of the article is organized as follows. In Section 2 we describe the model. In Section 3 we investigate the worst-case traffic conditions. In Section 4, we give more precise results for server farms with two classes of servers. We give the expression for the load that leads to inefficiency and the corresponding value of the inefficiency. We study the inefficiency of a server farm with an infinite number of dispatchers in Section 5. Finally, the main conclusions of this work are presented in Section 6.

A conference version of this article appeared in Doncel et al. [2013]. Appendices C and D are available in the Online Appendix to this article available in the ACM Digital Library.

2. PROBLEM FORMULATION

We consider a noncooperative routing game with K dispatchers and S processor-sharing servers. Denote $\mathcal{C} = \{1, \ldots, K\}$ to be the set of dispatchers and $\mathcal{S} = \{1, \ldots, S\}$ as the set of servers. Jobs received by dispatcher i are said to be jobs of stream i.

Server $j \in \mathcal{S}$ has capacity r_j. It is assumed that servers are numbered in the order of decreasing capacity, that is, if $m \leq n$, then $r_m \geq r_n$. Let $\mathbf{r} = (r_j)_{j\in\mathcal{S}}$ denote the vector of server capacities and let $\bar{r} = \sum_{n\in\mathcal{S}} r_n$ denote the total capacity of the system.

Jobs of stream $i \in \mathcal{C}$ arrive to the system according to a Poisson process and have generally distributed service times. We do not specify the arrival rate and the characteristics of the service-time distribution due to the fact that, in an $M/G/1 - PS$ queue, the mean number of jobs depends on the arrival process and service-time distribution only through the traffic intensity, namely, the product of the arrival rate and the mean service time. Let λ_i be the traffic intensity of stream i. It is assumed that $\lambda_i \leq \lambda_j$ for $i \leq j$. Moreover, it will also be assumed that the vector $\boldsymbol{\lambda}$ of traffic intensities belongs to the set $\Lambda(\bar{\lambda}) = \{\boldsymbol{\lambda} \in \mathbb{R}^K \ : \ \sum_{i\in\mathcal{C}} \lambda_i = \bar{\lambda}\}$, where $\bar{\lambda}$ denotes the total incoming traffic intensity. It will be assumed throughout the article that $\bar{\lambda} < \bar{r}$, which is the necessary and sufficient condition to guarantee the stability of the system. We denote by $\rho = \frac{\bar{\lambda}}{\bar{r}}$ the total traffic of the system.

We will sometimes be interested in what happens when $\bar{\lambda} \to \bar{r}$, a regime we will refer to as heavy-traffic ($\rho \to 1$).

Let $\mathbf{x}_i = (x_{i,j})_{j\in\mathcal{S}}$ denote the routing strategy of dispatcher i, with $x_{i,j}$ being the amount of traffic it sends towards server j. Dispatcher i seeks to find a routing strategy that minimizes the mean sojourn times of its jobs, which, by Little's law, is equivalent to minimizing the mean number of jobs in the system as seen by this stream. This optimization problem can be formulated as follows:

$$\text{minimize } T_i(\mathbf{x}) = \sum_{j\in\mathcal{S}} \frac{x_{i,j}}{r_j - y_j} \qquad \text{(ROUTE-}i\text{)}$$

$$\text{subject to } \sum_{j\in\mathcal{S}} x_{i,j} = \lambda_i, \quad i = 1, \ldots, K, \qquad (1)$$

$$\text{and } \ 0 \leq x_{i,j} \leq r_j, \quad \forall j \in \mathcal{S}, \qquad (2)$$

where $y_j = \sum_{k\in\mathcal{C}} x_{k,j}$ is the traffic offered to server j. Note that the optimization problem solved by dispatcher i depends on the routing decisions of the other dispatchers since $y_j = x_{i,j} + \sum_{k\neq i} x_{k,j}$. We let $\mathcal{X}_i$ denote the set of feasible routing strategies for dispatcher

i, namely, the set of routing strategies satisfying constraints (1) and (2). A vector $\mathbf{x} = (\mathbf{x}_i)_{i\in\mathcal{C}}$ belonging to the product strategy space $\mathcal{X} = \bigotimes_{i\in\mathcal{C}} \mathcal{X}_i$ is called a strategy profile.

A Nash equilibrium of the routing game is a strategy profile from which no dispatcher finds it beneficial to deviate unilaterally. Hence $\mathbf{x} \in \mathcal{X}$ is a Nash Equilibrium Point (NEP) if $\mathbf{x}_i$ is an optimal solution of problem (ROUTE-i) for all dispatchers $i \in \mathcal{C}$.

Let $\mathbf{x}$ be an NEP for the system with K dispatchers. The global performance of the system can be assessed using the global cost

$$D_K(\boldsymbol{\lambda}, \mathbf{r}) = \sum_{i\in\mathcal{C}} T_i(\mathbf{x}) = \sum_{j\in\mathcal{S}} \frac{y_j}{r_j - y_j},$$

where the offered traffic y_j are those at the NEP. The preceding cost represents the mean number of jobs in the system. Note that, when there is a single dispatcher, we have a single dispatcher with $\lambda_1 = \bar{\lambda}$. The global cost can therefore be written as $D_1(\bar{\lambda}, \mathbf{r})$ in this case.

We shall use the ratio between the performance obtained by the Nash equilibrium and the global optimal solution as a metric in order to assess the inefficiency of a decentralized scheme with K dispatchers and S servers. We define the inefficiency as the performance ratio under the worst possible traffic conditions, namely

$$\textit{inefficiency } I_K^S(\mathbf{r}) = \sup_{\boldsymbol{\lambda}\in\Lambda(\bar{\lambda}),\bar{\lambda}<\bar{r}} \frac{D_K(\boldsymbol{\lambda}, \mathbf{r})}{D_1(\bar{\lambda}, \mathbf{r})}. \tag{3}$$

The rationale for this definition is that, in practice, the system administrator controls neither the total incoming traffic nor how it is split between the dispatchers, whereas the number of servers and their capacities are fixed. Therefore it makes sense to consider the worst traffic conditions for the inefficiency of selfish routing, provided the system is stable.

The PoA for this system as defined in Ayesta et al. [2011] can be retrieved by looking at the worst inefficiency, namely,

$$PoA(K, S) = \sup_{\mathbf{r}} I_K^S(\mathbf{r}). \tag{4}$$

3. WORST-CASE TRAFFIC CONDITIONS

In this section we show that for a sufficiently large load, the ratio $\frac{D_K(\boldsymbol{\lambda},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ decreases with $\bar{\lambda}$. This result implies that the inefficiency of a data-center is not achieved in the heavy-traffic regime. Moreover, we also prove that when the system is in heavy traffic, namely, when $\bar{\lambda} \to \bar{r}$, the performance of both settings is the same.

The main difficulty in determining the behaviour of the inefficiency stems from the fact that for most cases there are no easy-to-compute explicit expressions for the NEP. A first simplification results from the following theorem which was proved in one of our previous works [Ayesta et al. 2011]. It states that, among all traffic vectors with total traffic intensity $\bar{\lambda}$, the global cost $D_K(\boldsymbol{\lambda}, \mathbf{r})$ achieves its maximum when all dispatchers control the same fraction of the total traffic. Formally, we have the following.

THEOREM 3.1 [AYESTA ET AL. 2011].

$$D_K(\boldsymbol{\lambda}, \mathbf{r}) \le D_K\left(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r}\right). \quad \forall \boldsymbol{\lambda} \in \Lambda(\bar{\lambda}),$$

where $\mathbf{e}$ *is the all-ones vector.*

Thus, we have identified the traffic vector in the set $\Lambda(\bar{\lambda})$ that has the worst ratio of global cost at the NEP to the global optimal cost. Corollary 3.2 follows from Theorem 3.1.

COROLLARY 3.2.

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda}<\bar{r}} \frac{D_K\left(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r}\right)}{D_1(\bar{\lambda}, \mathbf{r})}.$$

Routing games in which players have exactly the same strategy set are known as *symmetric* games. These games belong to the class of *potential games* [Monderer and Shapley 1996], that is, they have the property that there exists a function, called the *potential*, such that the NEP can be obtained as the solution of an optimization problem with the potential as the objective. This property considerably simplifies computation of the NEP. Another important consequence of the prior results is that the inefficiency depends only the total traffic intensity and not on individual traffic flows to each of the dispatchers.

Another consequence of Theorem 3.1 is that the inefficiency of decentralized routing increases with the number of dispatchers, as follows.

LEMMA 3.3.

$$I_K^S(\mathbf{r}) \le I_{K+1}^S(\mathbf{r}), \quad \forall K \ge 1.$$

PROOF. We have for all $\bar{\lambda} < \bar{r}$,

$$D_K\left(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r}\right) = D_{K+1}\left(\left(\frac{\bar{\lambda}}{K}\,\mathbf{e}, 0\right), \mathbf{r}\right) \le D_{K+1}\left(\frac{\bar{\lambda}}{K+1}\,\mathbf{e}, \mathbf{r}\right),$$

where the last inequality follows from Theorem 3.1. It yields

$$\sup_{\bar{\lambda}<\bar{r}} \frac{D_K\left(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r}\right)}{D_1(\bar{\lambda}, \mathbf{r})} \le \sup_{\bar{\lambda}<\bar{r}} \frac{D_{K+1}\left(\frac{\bar{\lambda}}{K+1}\,\mathbf{e}, \mathbf{r}\right)}{D_1(\bar{\lambda}, \mathbf{r})},$$

namely, $I_K(\mathbf{r}) \le I_{K+1}(\mathbf{r})$. □

Before going further, let us take a look at the ratio $\frac{D_K(\frac{\bar{\lambda}}{K}\,\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ as a function of the load $\rho = \bar{\lambda}/\bar{r}$, as shown in Figure 1 for two and five dispatchers. The datacenter characteristics are the following: 200 servers of speed 6,100 servers of speed 3,300 servers of speed 2, and 200 servers of speed 1. It can be observed that, as the load increases, the ratio goes through peaks and valleys and finally moves towards 1 as the load approaches 1. In the numerical experiments, we noted that the peaks corresponded to the total traffic intensity when selfish routing started to use one more class of servers. Moreover, just after these peaks, the number of servers used by selfish routing and the centralized version was the same. A similar behaviour is observed on different sets of experiments.

In general, it is not easy to make the earlier observation formal, that is to say, there are no simple expressions for the value of loads that corresponds to the peaks and the valleys. However, in heavy traffic, it helps to observe that both selfish and centralized routing will be using the same number of servers. Then, in order to show that heavy-traffic conditions are not inefficient, it is sufficient to show that the ratio decreases with load when both settings use the same number of servers.

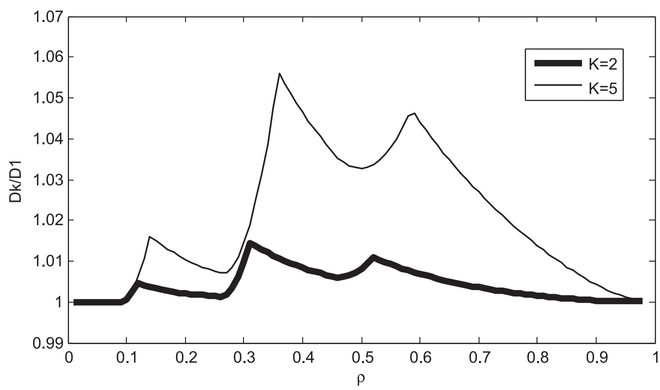

Fig. 1. Evolution of the ratio of social costs for $K = 2$ and $K = 5$ as the load in the system ranges from 0% to 100%.

PROPOSITION 3.4. *If the total traffic intensity $\bar{\lambda}$ is such that the centralized and the decentralized settings use the same number of servers (more than one), then the ratio of social costs $D_K(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ is decreasing with $\bar{\lambda}$.*

PROOF. See Appendix B.1. □

In the preceding result we exclude the case of one server so as to obtain a stronger result. If both the settings use just one server, then the ratio remains 1, and is nonincreasing. For a sufficiently high load, all the servers will be used by both settings in order to guarantee the stability of the system. It then follows that, in a server farm with an arbitrary number of servers and with arbitrary server capacities, a heavy-traffic regime is not inefficient. In fact, we can prove a stronger result which states that the inefficiency of the heavy-traffic regime is close to 1, that is, in heavy traffic both settings have similar performance. Formally, we have the following.

THEOREM 3.5. *For a fixed $K < \infty$,*

$$\lim_{\bar{\lambda}\to\bar{r}} \frac{D_K\left(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r}\right)}{D_1(\bar{\lambda}, \mathbf{r})} = 1.$$

PROOF. See Appendix B.2. □

It is important that the number of dispatchers be finite for the prior result to hold. As we show in Section 5, if the number of dispatchers is infinite as in the case of nonatomic games, the aforesaid limit may be a value larger than 1.

This result is important because it is widely believed that the maximum inefficiency of the decentralized routing scheme is obtained in the heavy-traffic regime. Theorem 3.5 shows that this belief is false. As can be observed in Figure 1, the worst-case traffic conditions can occur at low or moderate utilization rates (in fact, the worst total traffic intensity can be arbitrarily close to 0 if the server capacities are sufficiently close to each other). In heavy traffic, even though the cost in both the settings will grow, the rate of growth is the same that in a ratio close to 1.

The characterization of the exact traffic intensity which results in $I_K^S(\mathbf{r})$ proves a difficult task for arbitrary values of the capacities. In the following section we restrict ourselves to two server classes.

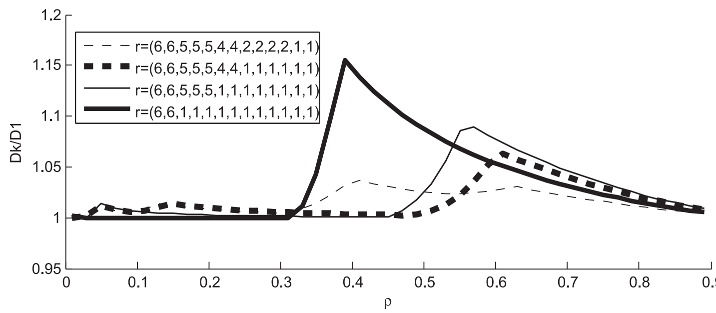

Fig. 2. Evolution of the ratio of social costs when $K = 5$ and $S = 13$ as the load in the system ranges from 0% to 90% for a server farm with different values of the capacities.

4. INEFFICIENCY FOR TWO-SERVER CLASSES

Before considering in detail the two-classes case, let us first observe how the inefficiency depends on the configuration of servers. Assume that there are 5 dispatchers and 13 servers. Figure 2 presents the evolution of the ratio $\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ according to the total load of the system for several vectors $\mathbf{r}$ of server capacities. We observe that the highest inefficiency is obtained in the case of two classes of servers with extreme capacity values. From extensive numerical experiments, we conjecture that, given the number of servers, for each problem instance with arbitrary server capacities we can construct a scenario with the same number of servers and two server speeds whose inefficiency is worse. This is formally stated in Conjecture 4.1.

Conjecture 4.1. For a data-center of S servers with $r_1 > r_S$,

$$I_K^S(\mathbf{r}) \geq I_K^S(\mathbf{r}^*),$$

where $\mathbf{r} = (\overbrace{r_1, \ldots, r_1}^{\text{m}}, \overbrace{r_S, \ldots, r_S}^{\text{S-m}})$ and $\mathbf{r}^* = (\overbrace{r_1, \ldots, r_1}^{\text{m}}, \overbrace{r_{m+1}, \ldots, r_{S-1}, r_S}^{\text{S-m}})$, with $r_1 > r_{m+1} \geq \cdots \geq r_{S-1} \geq r_S$ and $m \geq 1$.

A direct consequence of the prior conjecture is that, if for two classes of servers the inefficiency is very close to one except in some pathological cases, this should also be true for more than two classes.

In the following, we thus consider a server farm with two classes of servers. Let S_1 be the number of "fast" servers of capacity r_1 and let $S_2 = S - S_1$ be the number of "slow" servers, each of capacity r_2, either $r_1 > r_2$.[2] The behaviour of the ratio of social costs is illustrated in Figure 3 in the case of a server farm with $S_1 = 100$ fast servers of capacity $r_1 = 100$ and $S_2 = 300$ slow servers of capacity $r_2 = 10$. We plot the evolution of the ratio $D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ while the load on the system ranges from 0% to 1 for $K = 2$, $K = 5$.

It was observed that, for low loads, both the settings used the fast servers. The ratio in this regime was 1. After a certain point, the centralized setting started to use the slow servers as well, and the ratio increased with the load until the point when the decentralized setting also started to use the slow servers. From this point on, the ratio decreased with increase in the load.

[2]In the case $r_2 = r_1$, it is easy to see that the NEP is always an optimal routing solution, whatever the total traffic intensity.

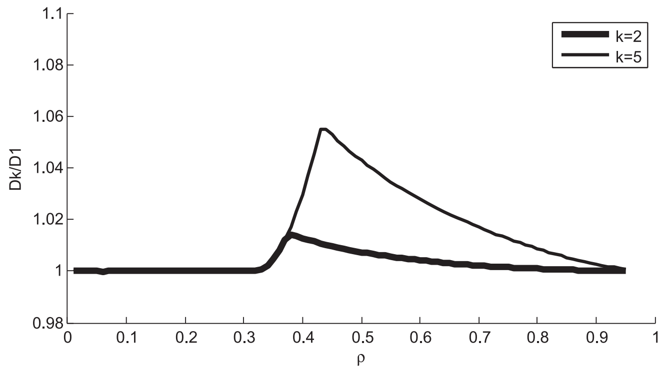

Fig. 3. Evolution of the ratio of social costs for $K = 2$ and $K = 5$ as the load in the system ranges from 0% to 100%.

We shall now characterize the point where the ratio starts to increase and where the peak occurs. Define

$$\bar{\lambda}^{OPT} = S_1\sqrt{r_1}(\sqrt{r_1} - \sqrt{r_2}),$$

and

$$\bar{\lambda}^{NE} = S_1 r_1 \left(1 - \frac{2}{\sqrt{(K-1)^2 + 4K\frac{r_1}{r_2}} - (K-1)}\right).$$

The following lemma gives the conditions on $\bar{\lambda}$ under which the centralized setting and the decentralized one use only the fast class of servers, or both classes.

LEMMA 4.2. *For $K \geq 2$, $\bar{\lambda}^{OPT} < \bar{\lambda}^{NE}$, and:*

(1) *if $\bar{\lambda} \leq \bar{\lambda}^{OPT}$, both settings use only the fast servers;*
(2) *if $\bar{\lambda}^{OPT} \leq \bar{\lambda} \leq \bar{\lambda}^{NE}$, the decentralized setting uses only the fast servers, while the centralized one uses all servers; and*
(3) *if $\bar{\lambda} > \bar{\lambda}^{NE}$, both settings use all servers.*

PROOF. See Appendix C.1. □

Since $\bar{\lambda}^{OPT} < \bar{\lambda}^{NE}$, a consequence of Lemma 4.2 is that the decentralized setting always uses a subset of the servers used by the centralized one. We immediately obtain expressions of the social cost in the centralized and decentralized settings as given in Corollary 4.3.

COROLLARY 4.3. *For the centralized setting, if $\bar{\lambda} < \bar{\lambda}^{OPT}$, then*

$$D_1(\bar{\lambda}, \mathbf{r}) = \bar{\lambda} \Big/ \left(r_1 - \frac{\bar{\lambda}}{S_1}\right),$$

otherwise

$$D_1(\bar{\lambda}, \mathbf{r}) = \left[\bar{\lambda}\sqrt{\frac{r_1}{r_2}} + S_1 y_1 \left(1 - \sqrt{\frac{r_1}{r_2}}\right)\right] \frac{1}{r_1 - y_1},$$

where$y_1 = \sqrt{r_1}\,\frac{\bar{\lambda} - S_2\sqrt{r_2}(\sqrt{r_2} - \sqrt{r_1})}{S_1\sqrt{r_1} + S_2\sqrt{r_2}}$ and $y_2 = (\bar{\lambda} - S_1 y_1)/S_2$ are the loads on each fast server and on each slow server in the case $\bar{\lambda} > \bar{\lambda}^{OPT}$, respectively. Similarly, if $\bar{\lambda} < \bar{\lambda}^{NE}$, *then*

$$D_K\left(\frac{\bar\lambda}{K}\mathbf{e},\mathbf{r}\right)=\bar\lambda\Big/\left(r_1-\frac{\bar\lambda}{S_1}\right)$$

and

$$D_K\left(\frac{\bar\lambda}{K}\mathbf{e},\mathbf{r}\right)=\frac{1}{2}\sum_{j=1}^{2}S_j\left[\sqrt{(K-1)^2+4Kr_j\gamma(K)}-(K+1)\right]$$

otherwise.

PROOF. See Appendix C.2. □

In Lemma 4.2, we identified three intervals, namely $[0,\bar\lambda^{OPT})$, $[\bar\lambda^{OPT},\bar\lambda^{NE})$, $[\bar\lambda^{NE},\bar r)$, each corresponding to a different set of servers used by the two settings. In Proposition 4.4, we describe how the ratio of the social costs evolves in each of these three intervals.

PROPOSITION 4.4. *The ratio* $D_K(\frac{\bar\lambda}{K}\mathbf{e},\mathbf{r})/D_1(\bar\lambda,\mathbf{r})$ *is*

(a) equal to 1 for $0\le\bar\lambda\le\bar\lambda^{OPT}$*;*
(b) strictly increasing over the interval $(\bar\lambda^{OPT},\bar\lambda^{NE})$*; and*
(c) strictly decreasing over the interval $(\bar\lambda^{NE},\bar r)$*.*

PROOF. See Appendix C.3. □

Moreover, the ratio of social costs has the following property.

LEMMA 4.5. *The ratio* $D_K(\frac{\bar\lambda}{K}\mathbf{e},\mathbf{r})/D_1(\bar\lambda,\mathbf{r})$ *is a continuous function of* $\bar\lambda$ *over the interval* $[0,\bar r)$*.*

PROOF. See Appendix C.4. □

We can now state the main result of this section.

THEOREM 4.6. *The inefficiency is worst when the total arriving traffic intensity equals* $\bar\lambda^{NE}$*, namely*

$$I_K^S(\mathbf{r})=\frac{D_K\left(\frac{\bar\lambda^{NE}}{K}\mathbf{e},\mathbf{r}\right)}{D_1(\bar\lambda^{NE},\mathbf{r})}.$$

PROOF. It was shown in Lemma 4.5 that $D_K(\frac{\bar\lambda}{K}\mathbf{e},\mathbf{r})/D_1(\bar\lambda,\mathbf{r})$ is a continuous function of $\bar\lambda$ over the interval $[0,\bar r)$. Proposition 4.4.(a) states that the ratio is minimum for $0\le\bar\lambda\le\bar\lambda^{OPT}$. For $\bar\lambda$ in $(\bar\lambda^{OPT},\bar\lambda^{NE})$, we know from Proposition 4.4.(b) that this ratio is strictly increasing, which implies that $I_K^S(\mathbf{r})\ge D_K(\frac{\bar\lambda^{NE}}{K}\mathbf{e},\mathbf{r})/D_1(\bar\lambda^{NE},\mathbf{r})$ by continuity. Since, according to Proposition 4.4.(c), the ratio is decreasing over the interval $(\bar\lambda^{NE},\bar r)$, we can conclude that its maximum value is obtained for $\bar\lambda=\bar\lambda^{NE}$. □

Theorem 4.6 fully characterizes the worst-case traffic conditions for a server farm with two classes of servers. It states that the worst inefficiency of the decentralized setting is achieved when: (a) each dispatcher controls the same amount of traffic and (b) the total traffic intensity is such that the decentralized setting only starts using the slow servers. The behaviour described by Proposition 4.4 can easily be observed in Figure 3.

For more than two classes of servers, we were unfortunately not able to prove the preceding results concerning the worst traffic conditions. Nevertheless, we conjecture

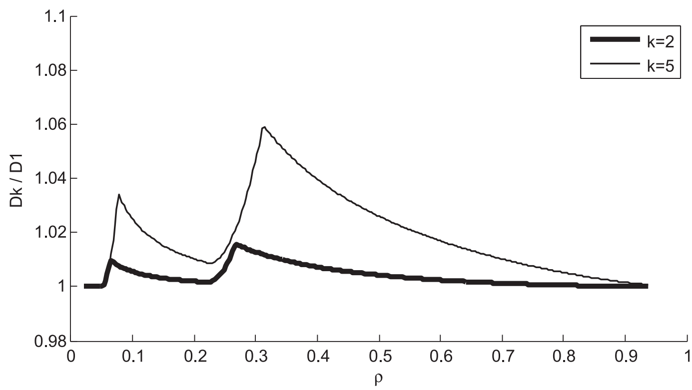

Fig. 4. The evolution of the ratio of social costs for $K = 2$ and $K = 5$ with respect to ρ in a server farm with 3 server classes.

that a similar behaviour happens also in this case. As another illustration of this behaviour, in Figure 4 we plot the ratio of social costs as a function of the load on the system, for a server farm with 3 server classes (and for $K = 2$, $K = 5$) with $S_1 = 100$ fast servers of capacity $r_1 = 30$, $S_2 = 200$ intermediate servers of capacity $r_2 = 20$, and $S_3 = 100$ slow servers of capacity $r_3 = 10$.

4.1. Inefficiency for a Given Architecture

We now give the expression for the inefficiency of selfish routing for data-centers with two classes of servers. Using Theorem 4.6 we assume the worst traffic conditions for the inefficiency of selfish routing, that is, the symmetric game obtained for $\bar{\lambda} = \bar{\lambda}^{NE}$.

PROPOSITION 4.7. *Let $\beta = \frac{r_1}{r_2} > 1$ and $\alpha = \frac{S_1}{S_2} > 0$, then*

$$I_K^S(\mathbf{r}) = \frac{1}{2} \frac{\sqrt{(K-1)^2 + 4K\beta} - (K+1)}{\frac{(\frac{1}{\alpha}+\sqrt{\beta})^2}{\frac{1}{\alpha}+\frac{2\beta}{\sqrt{(K-1)^2+4K\beta}-(K-1)}} - (\frac{1}{\alpha}+1)}. \tag{5}$$

PROOF. According to Theorem 4.6, we have $I_K^S(\mathbf{r}) = D_K(\frac{\bar{\lambda}^{NE}}{K}\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}^{NE}, \mathbf{r})$. The proof is then obtained after some algebra by using the expressions for $D_K(\frac{\bar{\lambda}^{NE}}{K}\mathbf{e}, \mathbf{r})$ and $D_1(\bar{\lambda}^{NE}, \mathbf{r})$ given in Corollary 4.3, and the expression for $\bar{\lambda}^{NE}$ given in Lemma 4.2. □

The inefficiency $I_K^S(\mathbf{r})$ does not depend on the total number of servers S, but only on the ratio of server capacities and on the ratio of the numbers of servers of each type. In Figure 5, we plot the inefficiency $I_K(\mathbf{r})$ of the noncooperative routing scheme with $K = 5$ dispatchers and $S = 1000$ servers as the parameters α and β change from $\frac{1}{S-1}$ to 2 and from 1 to 1,000, respectively. It can be observed that, even for unbalanced scenarios (i.e., α small and β large), the inefficiency is always fairly close to 1, indicating that even in the worst-case traffic conditions the gap between the NEP and the optimal routing solution is not significant. With slight abuse of notation, let us denote the right-hand side of (5) by $I_K(\alpha, \beta)$.

LEMMA 4.8. *The function $I_K(\alpha, \beta)$ is decreasing with α.*

PROOF. See Appendix C.5. □

A consequence of the previous result is that, given the ratio of server speeds in a data-center, the inefficiency is largest when there is one fast server and all the other

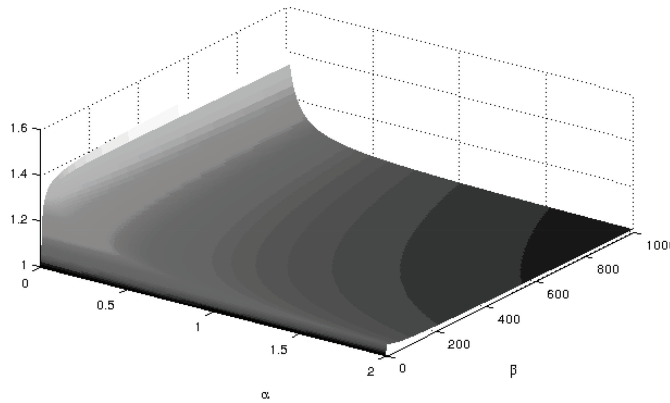

Fig. 5. Evolution of the inefficiency as a function of α and β for $K = 5$ dispatchers and $S = 1{,}000$ servers.

servers are slow. Selfish routing has the tendency to use the fast servers more than the slow ones. When there is just one fast server, its performance tends to be the worst as compared to that of the centralized routing that reduces its cost by sending traffic to the slower ones as well. Thus, in decentralized routing architectures, it is best to avoid server configurations with this particular kind of asymmetry.

4.2. Price of Anarchy

The PoA is defined as the worst possible inefficiency when the server capacities are varied. Then, from (3), (4), and Proposition 4.7, it follows that

$$PoA(K, S) = \sup_{\alpha,\beta} I_K(\alpha, \beta).$$

From Lemma 4.8 and the fact that, for a fixed S, α can take values in $\{\frac{1}{S-1}, \frac{2}{S-2}, \ldots, S-1\}$, we can deduce that

$$PoA(K, S) = \sup_{\beta} I_K\left(\frac{1}{S-1}, \beta\right). \tag{6}$$

We are able to prove that $I_K(\frac{1}{S-1}, \beta)$ is increasing with β. This means that the PoA of a server farm with S servers and K dispatchers is achieved when $\alpha = \frac{1}{S-1}$ and β infinity, that is, when there is only one fast server and it is infinitely faster than the slower ones. While there is no simple expression for the PoA in terms of K and S, we can nonetheless derive a certain number of properties from the preceding set of results.

PROPOSITION 4.9. *The price of anarchy has the following properties.*

(*1*) *For fixed K, $PoA(K, S)$ is increasing in S.*
(*2*) *For a fixed S, $PoA(K, S)$ is increasing in K.*

PROOF. For fixed K and for every β, from Lemma 4.8 and (6), we have

$$I_K\left(\frac{1}{S-1}, \beta\right) \le I_K\left(\frac{1}{S}, \beta\right) \le \sup_{\beta} I_K\left(\frac{1}{S}, \beta\right) = PoA(K, S+1),$$

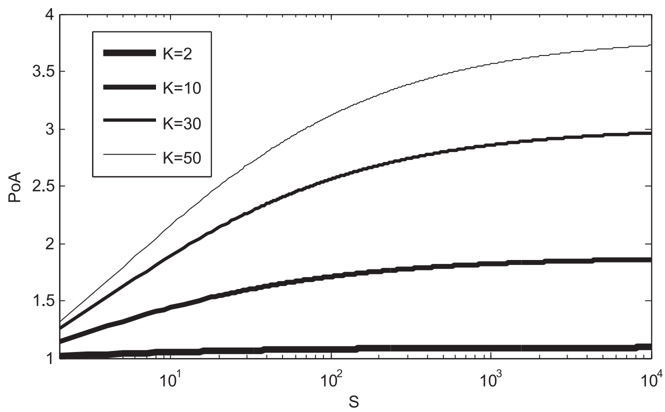

Fig. 6. The price of anarchy as a function of the number of servers for different values of the number of dispatcher.

where the last equality follows from (6). Taking the supremum over β in the prior inequality, we obtain, for a fixed K,

$$PoA(K, S) \leq PoA(K, S+1),$$

which proves the first property.

For a fixed S and β, from Lemma 3.3,

$$I_K\left(\frac{1}{S-1}, \beta\right) \leq I_{K+1}\left(\frac{1}{S-1}, \beta\right) \leq \sup_{\beta} I_{K+1}\left(\frac{1}{S-1}, \beta\right) = PoA(K+1, S).$$

Again, taking the supremum over β in the preceding inequality, we obtain, for a fixed S,

$$PoA(K, S) \leq PoA(K+1, S),$$

which proves the second property. □

In Figure 6, the PoA is plotted as a function of S for different values of K. It is observed that this value remains modest even when the number of servers is 10,000.

We now give an upper bound the PoA. For this, we first need the following result.

LEMMA 4.10. *For a server farm with two server classes and K dispatchers,*

$$\lim_{S\to\infty} PoA(K, S) = \frac{K}{2\sqrt{K}-1}.$$

PROOF. See Appendix C.6. □

PROPOSITION 4.11. *For a server farm with two server classes and K dispatchers and for all K and S,*

$$PoA(K, S) \leq \min\left(\frac{K}{2\sqrt{K}-1}, S\right).$$

PROOF. From Proposition 4.9, $PoA(K, S)$ is increasing with S. Combining this fact with Lemma 4.10, we can conclude that

$$PoA(K, S) \leq \frac{K}{2\sqrt{K}-1}.$$

Moreover, it was shown in Haviv and Roughgarden [2007] that, for the Wardrop case that is the limit of $K \to \infty$, $PoA(\infty, S) \leq S$. Thus

$$PoA(K, S) \leq S.$$

We can deduce the desired result from the preceding two inequalities. □

In server farms with a large number of servers, it follows from Lemma 4.10 that the PoA will be $\frac{K}{2\sqrt{K}-1}$. In Ayesta et al. [2011], it was shown that this value was a lower bound on the PoA. The model in that paper had server-dependent holding cost per unit time. The lower bound was obtained in an extreme case with negligible (tending to 0) holding cost on the fast servers and the decentralized setting marginally using the slow servers. Our present results show that the lower bound is indeed tight. Moreover, even in a less asymmetrical setting of equal holding costs per unit time, one can construct examples in which the PoA is attained.

The PoA obtained in the nonatomic case in Haviv and Roughgarden [2007] comes into play when there are few servers and a relatively large number of dispatchers. However, for data-centers the configuration is reversed; there are a few dispatchers and a large number of servers. In this case it is more appropriate to use the upper bound given in Lemma 4.10.

5. INEFFICIENCY WITH AN INFINITE NUMBER OF DISPATCHERS

We know from Lemma 3.3 that the inefficiency increases with the number of dispatchers K. This motivates the analysis of the inefficiency when the number of dispatchers K grows to infinity. In this section we show that, in the heavy-traffic regime the inefficiency is not one, in contrast to the case of finite K. For the case of two classes of servers we give the expression of the inefficiency, we characterize the situation under which the PoA is achieved, and show it is equal to the number of servers.

We define inefficiency of a server farm with S servers and an infinite number of dispatchers as

$$I_{\infty}^{S}(\mathbf{r}) = \lim_{K\to\infty} I_{K}^{S}(\mathbf{r}). \tag{7}$$

5.1. Heavy-Traffic Analysis

We study the performance of a data center with an arbitrary number of servers S when the system is in the heavy-traffic regime. We give the value of $\lim_{K\to\infty} \frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ in the following proposition.

PROPOSITION 5.1. *For a data-center with S servers, we have*

$$\lim_{\bar{\lambda}\to\bar{r}} \lim_{K\to\infty} \frac{D_K\big(\frac{\bar{\lambda}}{K}\ \mathbf{e}, \mathbf{r}\big)}{D_1(\bar{\lambda}, \mathbf{r})} = \frac{S\,\bar{r}}{\Big(\sum_{i=1}^{S}\sqrt{r_i}\Big)^2}.$$

PROOF. See Appendix D.1. □

We observe that this result does not coincide with the one obtained for the Nash equilibrium for K large, as given in Theorem 3.5. This implies that the limits of the number of dispatchers and heavy traffic do not interchange, namely

$$\lim_{K\to\infty} \lim_{\bar{\lambda}\to\bar{r}} \frac{D_K\big(\frac{\bar{\lambda}}{K}\ \mathbf{e}, \mathbf{r}\big)}{D_1(\bar{\lambda}, \mathbf{r})} \neq \lim_{\bar{\lambda}\to\bar{r}} \lim_{K\to\infty} \frac{D_K\big(\frac{\bar{\lambda}}{K}\ \mathbf{e}, \mathbf{r}\big)}{D_1(\bar{\lambda}, \mathbf{r})}.$$

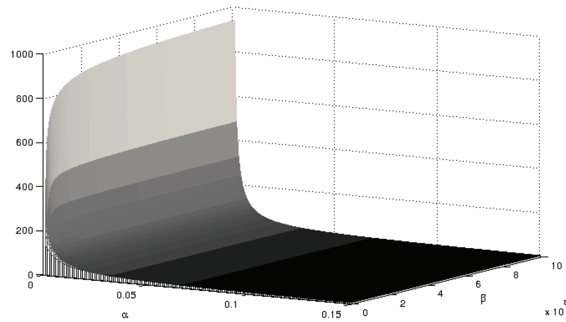

Fig. 7. Evolution of the inefficiency as a function of α and β for $K = 10^6$ and $S = 1{,}000$.

5.2. The Case of Two-Server Classes

We focus on the case of a data center with servers of two different speeds, r_1 and r_2 respectively, where $r_1 > r_2$. For the case of $K = \infty$, we conjecture that the worst inefficiency is obtained for two server classes, that is, Conjecture 4.1 holds when the number of dispachers grows to infinity.

Let S_1 be the number of fast servers and $S_2 = S - S_1$ the number of slow. We give the expression of the inefficiency for a data center with two classes of servers in terms only of the parameters $\alpha = \frac{S_1}{S_2}$ and $\beta = \frac{r_1}{r_2} > 1$.

COROLLARY 5.2.

$$I_\infty(\alpha, \beta) = \frac{(\beta - 1)(1 + \frac{1}{\alpha})}{(\sqrt{\beta} + \frac{1}{\alpha})^2 - (\frac{1}{\alpha} + 1)^2}. \tag{8}$$

PROOF. The result follows from Theorem 4.6 and (7), taking into account that $\sqrt{(K-1)^2 + 4Kx} - (K - 1) = 2x$, when $K \to \infty$, $\forall x \geq 1$. □

Using this expression, we can use the parameters $\alpha \in \{\frac{1}{S-1}, \frac{2}{S-2}, \ldots, \frac{S-2}{2}, S - 1\}$ and β to characterize the inefficiency for a datacenter with two-server classes and infinite number of servers. Lemma 5.3 states the main properties of the inefficiency for an infinite number of dispatchers.

LEMMA 5.3. *We have that $I_\infty(\alpha, \beta)$ is decreasing with α for all β and that $I_\infty(\alpha, \beta)$ is increasing with β for all α.*

PROOF. The result follows from Corollary 5.2. □

A direct consequence of Corollary 5.2 and Lemma 5.3 is that the price of anarchy of a datacenter with two classes of servers and infinite number of dispatchers is equal to the number of servers S.

PROPOSITION 5.4. $PoA(\infty, S) = S$.

PROOF. According to Lemma 5.3, the worst inefficiency is obtained when $\alpha = \frac{1}{S-1}$ and $\beta \to \infty$. Using the formula of Corollary 5.2 for these values of α and β, we get the desired result. □

We observe this result coincides with that given by Haviv and Roughgarden [2007].

We illustrate in Figure 7 the evolution of the inefficiency of a server farm of $S = 1{,}000$ servers and $K = 10^6$ dispatchers when α changes from $\frac{1}{S-1}$ to 0.15 and β from 1 to 10^8. We observe that the inefficiency equals the PoA when $\alpha = \frac{1}{S-1}$ and $\beta = 10^8$, that is, when there is only one fast server and it is 10^8 times faster than the slower ones. We also see in Figure 7 that the inefficiency stays very close to 1 in most cases, even if the worst inefficiency is 1,000. Thus we can conclude that, although the inefficiency can be as bad as the number of servers when the number of dispatchers is infinity, the decentralized setting is almost always as efficient as the centralized one.

5.3. The Price of Anarchy

We have seen that the PoA equals the number of servers in the case of an infinity number of dispatchers, while for K finite the upper bound is given by the minimum of $\frac{K}{2\sqrt{K}-1}$ and S. We observe that, given the number of servers S, there exists a K^* such that:

—if $K \leq K^*$, then $PoA \leq \frac{K}{2\sqrt{K}-1}$; and
—if $K \geq K^*$, then $PoA \leq S$.

For a sufficiently large S, we can say that $S = \frac{K^*}{2\sqrt{K^*}-1} \approx 0.5\sqrt{K^*}$. Thus we claim that, for a sufficiently large S, then $K^* \approx 4S^2$ and this means that, if the number of dispatchers is smaller than $4S^2$, the upper bound on the PoA is given by $\frac{K}{2\sqrt{K}-1}$ and by the number of servers S otherwise.

6. CONCLUSIONS

Price of anarchy is an oft-used worst-case measure of the inefficiency of noncooperative decentralized architectures. In spite of its popularity, we have observed that the price of anarchy is an overly pessimistic measure that does not reflect the performance obtained in most instances of the load-balancing game. For an arbitrary architecture in the system, we have seen that (contrary to a common belief) the inefficiency is in general not achieved in the heavy-traffic regime. Surprisingly, we have shown that inefficiency might be achieved at arbitrarily low load. For the case of two classes of servers we give an explicit expression of the inefficiency, and we have shown that noncooperative load balancing has close-to-optimal performance in most cases. We also show that the worst-case performance given by the price of anarchy occurs only in a very specific setting, namely when there is only one fast server and it is infinitely faster than the slower ones. We conjecture that our conclusions will also be true for more than two classes of servers.

We believe that our study opens up a new complementary point of view on the PoA and hope that, in the future, researchers will not only investigate the PoA but also the inefficiency. As our work suggests, even if the PoA is very bad, the inefficiency might be low in most instances of the problem. We believe that this issue should be investigated for other models.

APPENDIXES

A. SOME KNOWN RESULTS

The results in this section are taken from Ayesta et al. [2011]. Since they are cited several times in the present work, we choose to present them here for its easy perusal. Let $W(K,z) = \sum_{j\in\mathcal{S}} W_j(K,z)$, we define the function

$$W_j(K,z) = \mathbb{1}_{\{z\in[\frac{1}{r_j},\frac{1}{r_{j+1}})\}} \cdot \left(\sum_{s=1}^{j} \frac{2r_s}{\sqrt{(K-1)^2+4Kr_s z}-(K-1)} - \sum_{s=1}^{j} r_s + \bar{\lambda} \right). \quad (9)$$

The following proposition gives the solution of the symmetric game.

PROPOSITION A.1. *The subset of servers used at the NEP is $\mathcal{S}^*(K) = \{1, 2, \ldots, j^*(K)\}$, where $j^*(K)$ is the greatest value of j such that $W(K, 1/r_{j+1}) \le 0 < W(K, 1/r_j)$. The equilibrium flows are $x_{i,j}(K) = y_j(K)/K$, $i \in \mathcal{C}$, $j \in \mathcal{S}^*(K)$, where the offered traffic of server j is given by*

$$y_j(K) = r_j \frac{\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K+1)}{\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K-1)} \qquad (10)$$

with $\gamma(K)$ the unique root of $W(K, z) = 0$ in $[\frac{1}{r_1}, \infty)$.

B. PROOFS OF THE RESULTS IN SECTION 3

B.1. Proof of Proposition 3.4

Before proving Proposition 3.4, we establish closed-form expressions for the value of the social costs in the centralized and decentralised settings. Recall that we assume a server farm with S servers with decreasing values of the capacities, namely, $r_i \le r_j$, if $i > j$.

LEMMA B.1. *Let n be the number of servers that the centralized setting uses, for $n = 1\ldots, S$, then $D_1(\bar{\lambda}, \mathbf{r}) = \frac{(\sum_{j=1}^n \sqrt{r_j})^2}{\sum_{j=1}^n r_j - \bar{\lambda}} - n$. Similarly, if the decentralized setting uses n servers, we have $D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r}) = \frac{1}{2}\sum_{j=1}^n [\sqrt{(K-1)^2 + 4Kr_j\gamma(K)} - (K+1)]$.*

PROOF. We first prove the results for the centralized setting. When the centralized setting uses n servers, Proposition A.1 states that, in this case,

$$y_j(1) = r_j \left(1 - \frac{1}{\sqrt{\gamma(1)}\sqrt{r_j}}\right), \qquad j = 1, \ldots, n,$$

that yields $\frac{y_j(1)}{r_j - y_j(1)} = \sqrt{\gamma(1)}\sqrt{r_j} - 1$, for $j = 1, \ldots, n$. We thus obtain that $D_1(\bar{\lambda}, \mathbf{r}) = \sum_{j=1}^n \frac{y_j(1)}{r_j - y_j(1)} = \sqrt{\gamma(1)} \sum_{j=1}^n \sqrt{r_j} - n$. Since $\gamma(1)$ is the unique root of $W(K, \gamma(1)) = 0$ as defined in Appendix A, according to Proposition A.1, then $\gamma(1)$ is the solution of

$$\frac{1}{\sqrt{\gamma(1)}} \sum_{j=1}^n \sqrt{r_j} = \sum_{j=1}^n r_j - \bar{\lambda}. \qquad (11)$$

Thus it follows that $\sqrt{\gamma(1)} = \sum_{j=1}^n \sqrt{r_j}/(\sum_{j=1}^n r_j - \bar{\lambda})$ and $D_1(\bar{\lambda}, \mathbf{r}) = \frac{(\sum_{j=1}^n \sqrt{r_j})^2}{\sum_{j=1}^n r_j - \bar{\lambda}} - n$. Let us now consider the decentralized setting. If the number of servers used by the decentralized setting is n, then (10) gives that for $j = 1, \ldots, n$

$$1 - \frac{y_j(K)}{r_j} = \frac{2}{\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K-1)}. \qquad (12)$$

From (10) and (12) yields the desired result $D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r}) = \sum_{j=1}^n \frac{y_j(K)}{r_j}(1 - \frac{y_j(K)}{r_j})^{-1} = \frac{1}{2}\sum_{j=1}^n [\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K+1)]$. □

We first show in the following lemma an important property to prove Proposition 3.4.

LEMMA B.2. *Let $a_k = \sqrt{(K-1)^2 + 4K\gamma(K)r_k} + (K-1)$, then for all $i > j$, $\frac{a_j}{a_i}$ is increasing with $\bar{\lambda}$.*

Proof. First, we define $b_j = \sqrt{(K-1)^2 + 4K\gamma(K)r_j}$ and we see that $\frac{b_j}{b_i}$ is increasing with $\bar{\lambda}$ if $\frac{(K-1)^2+4K\gamma(K)r_j}{(K-1)^2+4K\gamma(K)r_i}$ is increasing with $\bar{\lambda}$ because $\frac{b_j}{b_i}$ is positive and thus:

$$\frac{\partial}{\partial\bar{\lambda}}\left(\frac{(K-1)^2 + 4K\gamma(K)r_j}{(K-1)^2 + 4K\gamma(K)r_i}\right) \geq 0 \iff 4K\frac{\partial\gamma(K)}{\partial\bar{\lambda}}(K-1)^2(r_j - r_i) \geq 0$$

and is true due to $r_j \geq r_i$ if $i > j$. Hence, we have proved that $\frac{(K-1)^2+4K\gamma(K)r_j}{(K-1)^2+4K\gamma(K)r_i}$ is increasing with $\bar{\lambda}$ and this implies that $\frac{b_j}{b_i}$ is increasing with $\bar{\lambda}$. We also observe that $\frac{\partial b_j}{\partial\bar{\lambda}} \geq \frac{\partial b_i}{\partial\bar{\lambda}}$, if $i > j$ as

$$\frac{\partial b_j}{\partial\bar{\lambda}} \geq \frac{\partial b_i}{\partial\bar{\lambda}} \iff \frac{2K\frac{\partial\gamma(K)}{\partial\bar{\lambda}}r_j}{b_j} \geq \frac{2K\frac{\partial\gamma(K)}{\partial\bar{\lambda}}r_i}{b_i} \iff \frac{1}{\sqrt{\frac{(K-1)^2}{r_j^2} + \frac{4K\gamma(K)}{r_j}}} \geq \frac{1}{\sqrt{\frac{(K-1)^2}{r_i^2} + \frac{4K\gamma(K)}{r_i}}}$$

and this inequality holds since $r_j \geq r_i$ when $i > j$.

As $\frac{\partial b_j}{\partial\bar{\lambda}} = \frac{\partial a_j}{\partial\bar{\lambda}}$ and $a_j = b_j + (K-1)$, for $j = 1, \ldots, n$, we are able to state that if $\frac{b_j}{b_i}$ is increasing, then $\frac{a_j}{a_i}$ is increasing with $\bar{\lambda}$

$$\frac{\partial}{\partial\bar{\lambda}}\left(\frac{a_j}{a_i}\right) > 0 \iff \frac{\frac{\partial b_j}{\partial\bar{\lambda}}a_i - \frac{\partial b_i}{\partial\bar{\lambda}}a_j}{a_i^2} > 0 \iff \frac{\partial b_j}{\partial\bar{\lambda}}b_i - \frac{\partial b_i}{\partial\bar{\lambda}}b_j + (K-1)\left(\frac{\partial b_j}{\partial\bar{\lambda}} - \frac{\partial b_i}{\partial\bar{\lambda}}\right) > 0$$

and we know the inequality is satisfied because $\frac{\partial(\frac{b_j}{b_i})}{\partial\bar{\lambda}} > 0$ and $\frac{\partial b_j}{\partial\bar{\lambda}} > \frac{\partial b_i}{\partial\bar{\lambda}}$. □

Proof of Proposition 3.4. We show that, when both settings use n servers ($n = 1, \ldots, S$), then the ratio $\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ is decreasing with $\bar{\lambda}$. We use the expressions of $D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})$ and $D_1(\bar{\lambda},\mathbf{r})$ given in Lemma B.1 and we modify the ratio $\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ as follows:

$$\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})} = \frac{\frac{1}{2}\sum_{j=1}^n\left[\sqrt{(K-1)^2+4Kr_j\gamma(K)} - (K+1)\right]}{-n + \sqrt{\gamma(1)}\sum_{j=1}^n\sqrt{r_j}}$$
$$= \frac{-n + \frac{1}{2}\sum_{j=1}^n\left[\sqrt{(K-1)^2+4Kr_j\gamma(K)} - (K-1)\right]}{-n + \sqrt{\gamma(1)}\sum_{j=1}^n\sqrt{r_j}} = \frac{f_1 + f_2}{f_1 + g_2},$$

where we define $f_1 = \frac{-n}{\sqrt{\gamma(1)}}$, $g_2 = \sum_{j=1}^n\sqrt{r_j}$ and $f_2 = \frac{1}{2\sqrt{\gamma(1)}}\sum_{j=1}^n [\sqrt{(K-1)^2+4Kr_j\gamma(K)} - (K-1)]$.

We want to prove that the derivative of $\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ with respect to $\bar{\lambda}$ is negative:

$$\frac{\partial}{\partial\bar{\lambda}}\left(\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}\right) < 0 \iff \frac{\partial f_1}{\partial\bar{\lambda}}(g_2 - f_2) + \frac{\partial f_2}{\partial\bar{\lambda}}\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{\sqrt{\gamma(1)}} < 0.$$

We observe that f_1 is increasing with $\bar{\lambda}$, because $\gamma(1)$ increases with $\bar{\lambda}$, and $D_1(\bar{\lambda},\mathbf{r}) \leq D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})$ implies that $g_2 \leq f_2$. Therefore, if we show that f_2 is decreasing with $\bar{\lambda}$ and we can conclude that $\frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}$ is decreasing with $\bar{\lambda}$. From (11) and (9), if both settings use n servers then

$$\frac{1}{\sqrt{\gamma(1)}} = \frac{\sum_{j=1}^{n} r_j - \bar{\lambda}}{\sum_{j=1}^{n} \sqrt{r_j}} = \frac{1}{\sum_{j=1}^{n} \sqrt{r_j}} \sum_{s=1}^{n} \frac{2r_s}{\overline{a_s}},$$

where $\overline{a_s} = \sqrt{(K-1)^2 + 4K\gamma(K)r_s} - (K-1)$. We rewrite f_2 as follows:

$$f_2 = \frac{1}{\sum_{j=1}^{n} \sqrt{r_j}} \sum_{j=1}^{n} \overline{a_j} \sum_{s=1}^{n} \frac{r_s}{\overline{a_s}} = \frac{1}{\sum_{j=1}^{n} \sqrt{r_j}} \left(\sum_{j=1}^{n} r_j + \sum_{j=1}^{n} \sum_{i>j} \left[r_j \frac{\overline{a_i}}{\overline{a_j}} + r_i \frac{\overline{a_j}}{\overline{a_i}} \right] \right).$$

We define $a_s = \sqrt{(K-1)^2 + 4K\gamma(K)r_s} + (K-1)$ and notice that, if we multiply and divide $\overline{a_s}$ by a_s it yields $\overline{a_s} = \frac{4K\gamma(K)r_s}{a_s}$. So f_2 gets modified as follows with this property:

$$f_2 = \frac{1}{\sum_{j=1}^{n} \sqrt{r_j}} \left(\sum_{j=1}^{n} r_j + \sum_{j=1}^{n} \sum_{i>j} \left[r_j \frac{a_i}{a_j} + r_i \frac{a_j}{a_i} \right] \right).$$

Now, we show that $r_j/a_j^2 > r_i/a_i^2$ for all $i > j$ since $\frac{r_k}{a_k^2}$ is decreasing with k because we can write it as $\frac{r_k}{a_k^2} = [(\sqrt{\frac{(K-1)^2}{r_k} + 4K\gamma(K)} + \frac{K-1}{\sqrt{r_k}})^{-1}]^2$ and r_k decreases with k. Finally, we see that f_2 is decreasing with $\bar{\lambda}$

$$\frac{\partial f_2}{\partial \bar{\lambda}} = \frac{1}{\sum_{j=1}^{n} \sqrt{r_j}} \sum_{j=1}^{n} \sum_{i>j} \left[\left(\frac{\partial a_j}{\partial \bar{\lambda}} a_i - \frac{\partial a_i}{\partial \bar{\lambda}} a_j \right) \left(\frac{r_i}{a_i^2} - \frac{r_j}{a_j^2} \right) \right] < 0$$

and we conclude that this is true because, from Lemma B.2, $\frac{a_j}{a_i}$ is increasing with $\bar{\lambda}$ if $i > j$ (so that $\frac{\partial a_j}{\partial \bar{\lambda}} a_i - \frac{\partial a_i}{\partial \bar{\lambda}} a_j > 0$) and we have observed that $r_j/a_j^2 > r_i/a_i^2$ when $i > j$. □

B.2. Proof of Theorem 3.5

PROOF. First, we know that in heavy traffic all the servers are used, so we consider that S servers are used in both settings. We also observe that in heavy traffic $\gamma(K)$, as defined in Proposition A.1, tends to ∞, and the following approximation is satisfied.

$$\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K-1) \approx 2\sqrt{K\gamma(K)r_j} \tag{13}$$

From (13) and the definition of $\gamma(K)$, we obtain $\sqrt{K\gamma(K)} \approx \frac{\sum_{j=1}^{S} \sqrt{r_j}}{\sum_{j=1}^{S} r_j - \bar{\lambda}}$. Now, using this expression, (13), and Lemma B.1, we show that $D_K(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r}) \approx D_1(\bar{\lambda}, \mathbf{r})$ in heavy traffic:

$$\begin{aligned} D_K\left(\frac{\bar{\lambda}}{K}\mathbf{e}, \mathbf{r}\right) &= \frac{1}{2} \sum_{j=1}^{S} \left[\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K+1) \right] \\ &= -S + \frac{1}{2} \sum_{j=1}^{S} \left[\sqrt{(K-1)^2 + 4K\gamma(K)r_j} - (K-1) \right] \\ &\approx -S + \sqrt{K\gamma(K)} \sum_{j=1}^{S} \sqrt{r_j} = -S + \frac{\left(\sum_{j=1}^{S} \sqrt{r_j}\right)^2}{\sum_{j=1}^{S} r_j - \bar{\lambda}} = D_1(\bar{\lambda}, \mathbf{r}). \quad \square \end{aligned}$$

REFERENCES

U. Ayesta, O. Brun, and B. J. Prabhu. 2011. Price of anarchy in non-cooperative load-balancing games. *Perform. Eval.* 68, 1312–1332.

C. H. Bell and S. Stidham. 1983. Individual versus social optimization in the allocation of customers to alternative servers. *Manag. Sci.* 29, 831–839.

D. P. Bertsekas and J. N. Tsitsiklis. 1989. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall.

H. L. Chen, J. Marden, and A. Wierman. 2009. The effect of local scheduling in load balancing designs. *ACM Perform. Eval. Rev.* 36, 2, 110–112.

G. Christodoulou and E. Koutsoupias. 2005. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing.*

J. Doncel, U. Ayesta, O. Brun, and B. Prabhu. 2013. On the efficiency of non-cooperative load balancing. In *Proceedings of the IFIP Networking Conference*. 1–9.

J. Doncel, U. Ayesta, O. Brun, and B. Prabhu. 2014. Supplementary appendix: Is price of anarchy the right measure for load-balancing games? *ACM Trans. Internet Technol.* 14(2–3).

M. Haviv and T. Roughgarden. 2007. The price of anarchy in an exponential multi-server. *Oper. Res. Lett.* 35, 421–426.

Y. A. Korilis, A. A. Lazar, and A. Orda. 2006. Architecting noncooperative networks. *IEEE J. Select. Areas Comm.* 13, 7, 1241–1251.

Y. A. Korilis, A. A. Lazar, and A. Orda. 1997. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Trans. Netw.* 5, 1.

E. Koutsoupias and C. H. Papadimitriou. 1999. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS'99).* 404–413.

R. Miller. 2009a. Google unveils its container data center. http://www.datacenterknowledge.com/archives/2009/04/01/google-unveils-its-container-data-center/.

R. Miller. 2009b. Who has the most web servers? http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/.

D. Monderer and L. S. Shapley. 1996. Potential games. *Games Econ. Behav.* 14, 124–143.

D. Mosk-Aoyama, T. Roughgarden, and D. Shah. 2010. Fully distributed algorithms for convex optimization problems. *SIAM J. Optim.* 20, 6, 3260–3279.

T. Roughgarden. 2005. *Selfish Routing and the Price of Anarchy*. MIT Press.

T. Roughgarden. 2009. Intrinsic robustness of the price of anarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing.*

S. Suri, C. D. Toth, and Y. Zhou. 2004. Selfish load balancing and atomic congestion games. In *Proceedings of the 6th Annual ACM Symposium on Parallelism in Algorithms and Architectures*.

J. G. Wardrop. 1952. Some theoretical aspects of road traffic research. *Proc. Instit. Civil Engin.* 1, 325–378.

Received October 2013; revised March 2014; accepted June 2014