On the Efficiency of Non-Cooperative Load Balancing

J. Doncel[a,c], U. Ayesta[a,b,c,d], O. Brun[a,c], B.J. Prabhu[a,c]

[a] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[b] IKERBASQUE — Basque Foundation for Science, 48011 Bilbao, Spain
[c] Univ. de Toulouse, LAAS, F-31400 Toulouse, France
[d] Univ. of the Basque Country, Dept. of Computer Science, 20018 Donostia, Spain

***Abstract*—Price of Anarchy is an oft-used worst-case measure of the inefficiency of non-cooperative decentralized architectures. In practice, though, the worst-case scenario may occur rarely, if at all. For non-cooperative decentralized load-balancing in server farms, 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. In the case of two classes of servers, we show that non-cooperative load-balancing provides a close-to-optimal solution in most cases, and that the worst-case performance given by the Price of Anarchy occurs only in a very specific setting, namely, when the slower servers are infinitely more numerous and infinitely slower than the faster ones. We explicitly characterize the worst-case traffic conditions for the efficiency of non-cooperative load-balancing schemes, and show that, contrary to a common belief, the worst inefficiency is in general not achieved in heavy-traffic or close to saturation conditions.**

I. 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 data center is that of balancing the load over the servers so that the overall performance is optimized. In a centralized architecture, a single dispatcher, or a 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, and it becomes difficult or even impossible to centrally implement a globally optimal load-balancing solution. For instance, Akamai Technologies revealed, in march 2012, that it operates 105,000 servers [1]. Similarly, it is estimated that Google has more than 900,000 servers, and the company recently revealed that container data center holds more than 45,000 servers in a single facility built in 2005 [2]. The ever growing size and complexity of modern server farms thus calls for decentralized control schemes.

In a decentralized routing architecture, several dispatchers are used with each one routing a certain portion of the traffic. There are several possible approaches for the implementation of decentralized routing mechanisms. Approaches based on distributed optimization techniques [3], [4], 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 [5]. 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 non-cooperative routing game. The strategy that rational agents will choose under these circumstances is called a Nash Equilibrium and it 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 dispatcher grows to infinity (every incoming job is handled by a dispatcher and it takes its own routing decision) the corresponding equilibrium is given by the notion of Wardrop Equilibrium [6].

Apart from the obvious gain in scalability with respect to a centralized setting, there are wide-ranging advantages to non-cooperative 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 non-cooperative 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) which was introduced by Koutsoupias and Papadimitriou [7]. 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 that 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 non-cooperative load-balancing[1] can be very inefficient in the presence of non-linear delay functions, see, for example, [8], [9], [10], and [11]. We just mention two of them here. First, Haviv and Roughgarden have considered in [8] the so-called non-atomic 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.* in [10]. They investigate the price of anarchy of a load balancing game with a finite number, say K, of dispatchers, 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 [12], [13], [5].

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

In this paper, 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 [8], 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 case 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 the *inefficiency* is obtained for fairly low values of the utilization rate.
- In the case of two server classes, we show that the *inefficiency* is obtained when selfish routing uses only one class of servers and is marginally using the second class of servers. This scenario was used in [8], [10] 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* which in particular depends only on the ratio of the number of servers in each class and on the ratio 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 that it approaches the known upper bound (given by the PoA) only in a very specific setting, namely, when the slower servers are infinitely more numerous and infinitely slower than the faster ones.

The rest of the paper is organized as follows. In section II we describe the model. In section III we show that the *inefficiency* of selfish routing does not occur in heavy-traffic. In section IV, we give more precise results for server farms with two classes of servers. We give the expression for the load which leads to *inefficiency*, and the corresponding value of the *inefficiency*. Finally, the main conclusions of this work are presented in section V.

Due to lack of space we have omitted the proofs of our main results and for full details we refer to [14].

II. Problem Formulation

We consider a non-cooperative routing game with K dispatchers and S Processor-Sharing servers. Denote $\mathcal{C} = \{1,\dots,K\}$ to be the set of dispatchers and $\mathcal{S} = \{1,\dots,S\}$ to be 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, i.e., if $m \le n$, then $r_m \ge r_n$. Let $\mathbf{r} = (r_j)_{j\in\mathcal{S}}$ denote the vector of server capacities and let $\overline{r} = \sum_{n\in\mathcal{S}} r_n$ 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, i.e., 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 $\lambda_i \le \lambda_j$ for $i \le j$. Moreover, it will also be assumed that the vector $\boldsymbol{\lambda}$ of traffic intensities belongs to the following set: $\Lambda(\bar{\lambda}) = \left\{\boldsymbol{\lambda} \in \mathbb{R}^K \ : \ \sum_{i\in\mathcal{C}} \lambda_i = \bar{\lambda}\right\}$, where $\bar{\lambda}$ denotes the total incoming traffic intensity. It will be assumed throughout the paper that $\bar{\lambda} < \overline{r}$, which is the necessary and sufficient condition to guarantee the stability of the system.

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:

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

subject to

$$\sum_{j\in\mathcal{S}} x_{i,j} = \lambda_i, \quad i = 1,\dots,K, \tag{1}$$

$$0 \le x_{i,j} \le r_j, \quad \forall j \in \mathcal{S}, \tag{2}$$

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, i.e., the set of routing strategies satisfying constraints (1)-(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 dispatcher $i \in \mathcal{C}$.

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

$$D_K(\boldsymbol{\lambda}, \mathbf{r}) = \sum_{i\in\mathcal{C}} T_i(\mathbf{x}) = \sum_{j\in\mathcal{S}} \frac{y_j}{r_j - y_j}, \tag{3}$$

where the offered traffic y_j are those at the NEP. The above 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:

$$\textit{inefficiency } I_K^S(\mathbf{r}) = \sup_{\boldsymbol{\lambda}\in\Lambda(\bar{\lambda}),\bar{\lambda}<\overline{r}} \frac{D_K(\boldsymbol{\lambda},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}. \quad (4)$$

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 [10] can be retrieved by looking at the worst *inefficiency*, i.e.,

$$PoA(K,S) = \sup_{\mathbf{r}} I_K^S(\mathbf{r}). \quad (5)$$

III. Inefficiency is not in heavy-traffic

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 [10]. 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,

Theorem 1 ([10]):

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

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

Thus, we have identified the traffic vector in the set $\Lambda(\bar{\lambda})$ which has the worst-ratio of global cost at the NEP to the global optimal cost. It follows from the above result that

Corollary 1:

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda}<\overline{r}} \frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}. \quad (6)$$

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* [15], 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 the computation of the NEP. Another important consequence of the above results is that the *inefficiency* depends only on the total traffic intensity and not on individual traffic flows to each of the dispatcher.

Another consequence of theorem 1 is that the inefficiency of decentralized routing increases with the number of dispatchers, that is,

Lemma 1:

$$I_K^S(\mathbf{r}) \leq I_{K+1}^S(\mathbf{r}), \quad \forall K \geq 1. \quad (7)$$

Proof: We have for all $\bar{\lambda} < \overline{r}$, $D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r}) = D_{K+1}(\left(\frac{\bar{\lambda}}{K}\mathbf{e},0\right),\mathbf{r}) \leq D_{K+1}(\frac{\bar{\lambda}}{K+1}\mathbf{e},\mathbf{r})$, where the last inequality follows from theorem 1. It yields

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

i.e., $I_K(\mathbf{r}) \leq I_{K+1}(\mathbf{r})$. ■

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 $\rho = \bar{\lambda}/\overline{r}$, as is shown in figure 1 for two and five dispatchers. The data-center 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 it moves towards 1 as the load moves towards saturation. In the numerical experiments, we noted that the peaks corresponded to the total traffic intensity when selfish routing started to use one more server. Moreover, just after these peaks the number of servers used by selfish routing and the centralized one was the same. A similar behaviour was observed on different sets of experiments.

In general, it is not easy to make formal the above observation, that is to say, there are no simple expressions for the value of loads which 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

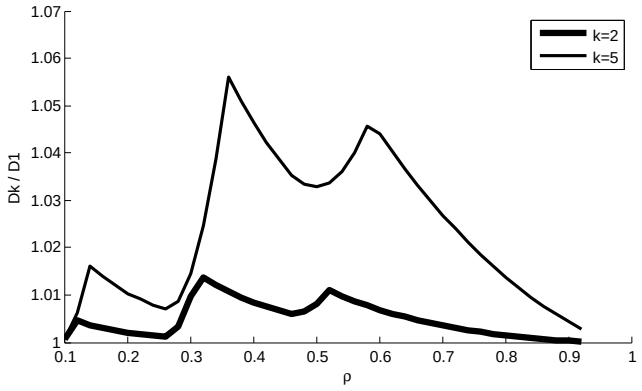

Fig. 1. Evolution of the ratio of social costs for $K = 2$ and $K = 5$ as the utilization rate ranges from 0% to 100%.

is sufficient to show that the ratio decreases with load when both the setting use the same number of servers.

Proposition 1: If the total traffic intensity $\bar{\lambda}$ is such that 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 $\bar{\lambda}$.

In the above 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, which is non-increasing.

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, 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 the settings have similar performance. Formally,

Theorem 2: For a fixed $K < \infty$,

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

It is important that the number of dispatchers be finite for the above result to hold. If the number of dispatchers is infinite, as in the case of non-atomic games, the above 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 heavy-traffic regime. Theorem 2 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 arbitrary close to 0 if the server capacities are sufficiently close from each other). In heavy-traffic, even though the cost in both the settings will grow, the rate of growth is the same which results in a ratio close to 1. This result is in sheer contrast with classical queueing theory as well. For example, in a $M/M/1$ queue the mean sojourn time is characterized by a factor $(1-\rho)^{-1}$, thus, as the load approaches one the mean sojourn time explodes.

The characterization of the exact traffic vector which results in I_K^S proves to be a difficult task. As a first attempt, in the following section we restrict ourselves to two server classes, which turns out to be more tractable than more number of classes.

IV. Inefficiency for two-server classes

Consider the case of two classes of servers: there are S_1 "fast" servers of capacity r_1, and $S_2 = S - S_1$ "slow" servers, each one of capacity $r_2 < r_1$ [2]. The behaviour of the ratio of social costs is illustrated in figure 2 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/D_1 as the load on the system ranges from 0% to 100% 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 load.

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

$$\bar{\lambda}^{OPT} = S_1\sqrt{r_1}(\sqrt{r_1} - \sqrt{r_2}), \tag{8}$$

[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.

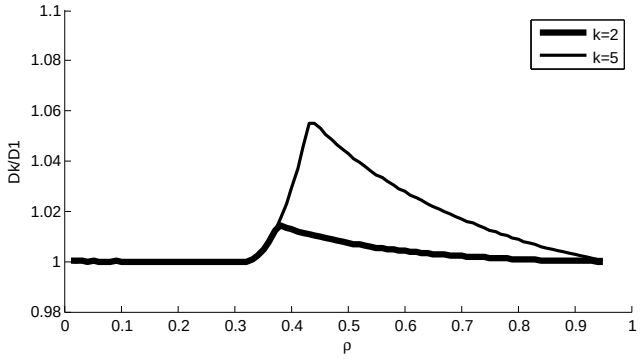

Fig. 2. Evolution of the ratio of social costs for $K = 2$ and $K = 5$ as the utilization rate ranges from 0% to 100%.

and

$$\bar{\lambda}^{NE} = S_1 r_1 \left(1 - \frac{2}{\sqrt{(K-1)^2 + 4K\frac{r_1}{r_2}} - (K-1)} \right). \tag{9}$$

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 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,
3) if $\bar{\lambda} > \bar{\lambda}^{NE}$, both settings use all servers.

Since $\bar{\lambda}^{OPT} < \bar{\lambda}^{NE}$, a consequence of lemma 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 2.

Corollary 2: For the centralized setting, if $\bar{\lambda} \leq \bar{\lambda}^{OPT}$

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

otherwise

$$D_1(\bar{\lambda}, \mathbf{r}) = \left[\bar{\lambda}\sqrt{\frac{r_1}{r_2}} + S_1 y_1 \left(1 - \sqrt{\frac{r_1}{r_2}} \right) \right] \frac{1}{r_1 - y_1}, \tag{10}$$

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 $\bar{\lambda} \geq \bar{\lambda}^{OPT}$, respectively. Similarly, if $\bar{\lambda} \leq \bar{\lambda}^{NE}$

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

and

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

otherwise.

In lemma 2, we identified three intervals, namely, $[0, \bar{\lambda}^{OPT})$, $[\bar{\lambda}^{OPT}, \bar{\lambda}^{NE})$, $[\bar{\lambda}^{NE}, \overline{r})$, each one corresponding to a different set of servers used by the two settings. In proposition 2, we describe how the ratio of the social costs evolves in each of these three intervals.

Proposition 2: The ratio $D_K(\frac{\bar{\lambda}}{K}\,\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}, \mathbf{r})$ is

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

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

Lemma 3: 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 $[0, \overline{r})$.

We can now state the main result of this section.

Theorem 3: The inefficiency is worst when the total arriving traffic intensity equals $\bar{\lambda}^{NE}$, namely,

$$I_K^S(\mathbf{r}) = \frac{D_K(\frac{\bar{\lambda}^{NE}}{K}\,\mathbf{e}, \mathbf{r})}{D_1(\bar{\lambda}^{NE}, \mathbf{r})}, \tag{11}$$

Proof: It was shown in lemma 3 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 $[0, \overline{r})$. Proposition 2.(a) states that the ratio is minimum for $0 \leq \bar{\lambda} \leq \bar{\lambda}^{OPT}$. For $\bar{\lambda}$ in $\left(\bar{\lambda}^{OPT}, \bar{\lambda}^{NE}\right)$, we know from proposition 2.(b) that this ratio is strictly increasing, which implies that $I_K^S(\mathbf{r}) \geq D_K(\frac{\bar{\lambda}^{NE}}{K}\,\mathbf{e}, \mathbf{r})/D_1(\bar{\lambda}^{NE}, \mathbf{r})$ by continuity. Since, according to proposition 2.(c), the ratio is decreasing over the interval $(\bar{\lambda}^{NE}, \overline{r})$, we can conclude that its maximum value is obtained for $\bar{\lambda} = \bar{\lambda}^{NE}$. ■

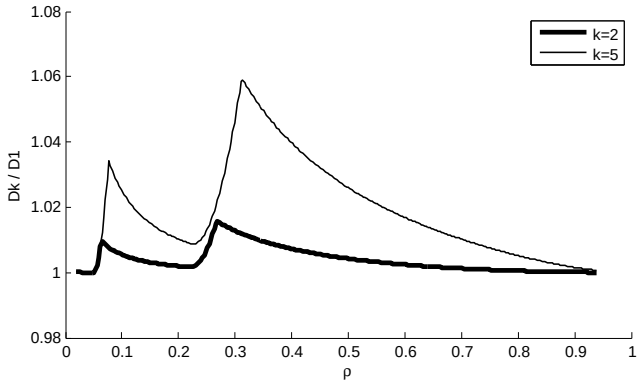

Fig. 3. The evolution of the ratio of social costs for $K = 2$ and $K = 5$ with respect to ρ in a server farm with 3 server classes.

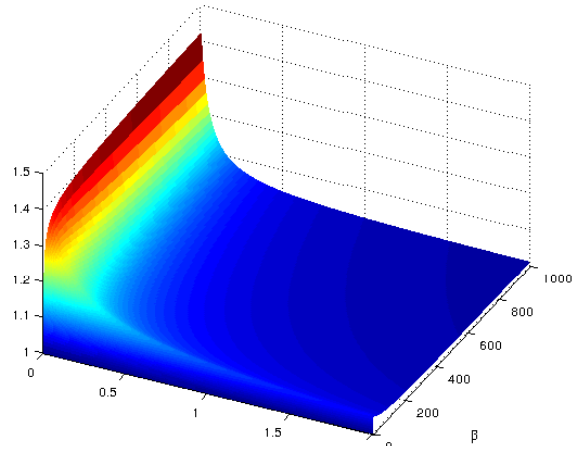

Fig. 4. Evolution of the *inefficiency* as a function of α and β for $K = 5$ dispatchers.

Theorem 3 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 2 can easily be observed in figure 2.

For more than two classes of servers, we were unfortunately not able to prove the above results concerning the worst traffic conditions. Nevertheless, we conjecture that a similar behaviour happens also in this case. As another illustration of this behaviour, in figure 3 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 $r_3 = 10$.

A. 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 3 we assume the worst traffic conditions for the inefficiency of selfish routing, i.e., the symmetric game obtained for $\bar{\lambda} = \bar{\lambda}^{NE}$.

Proposition 3:

$$I_K^S(\mathbf{r}) = \frac{1}{2}\frac{\sqrt{(K-1)^2+4K\beta}-(K+1)}{\frac{(\frac{1}{\alpha}+\sqrt{\beta})^2}{\frac{1}{\alpha}+\frac{2\beta}{\sqrt{(K-1)^2+4K\beta}-(K-1)}}-(\frac{1}{\alpha}+1)} \tag{12}$$

where $\beta = \frac{r_1}{r_2} \geq 1$ and $\alpha = \frac{S_1}{S_2} > 0$.

Proof: According to theorem 3, 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 2, and with the expression for $\bar{\lambda}^{NE}$ given in lemma 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 4, we plot the *inefficiency* $I_K(\mathbf{r})$ of the non-cooperative routing scheme with $K = 5$ dispatchers as the parameters α and β change. It can be observed that even for unbalanced scenarios (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 RHS of (12) by $I_K(\alpha, \beta)$.

Lemma 4: The function $I_K(\alpha, \beta)$ is decreasing with α.

A consequence of the above 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 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 which 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.

B. Price of Anarchy

The PoA is defined as the worst possible *inefficiency* when the server capacities are varied. Then, from (4), (5) and proposition 3,

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

From lemma 4 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

$$PoA(K,S) = \sup_{\beta} I_K\left(\frac{1}{S-1},\beta\right). \tag{13}$$

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: 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 and (13),

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

where the last equality follows from (13). Taking the supremum over β in the above inequality, we obtain, for a fixed K,

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

which proves the first property.

For a fixed S and β, from lemma 1,

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

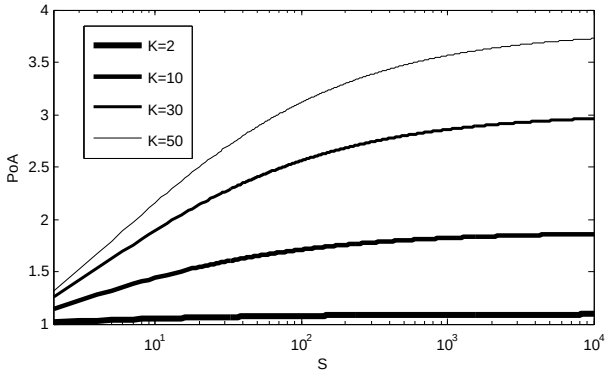

Fig. 5. The Price of Anarchy as a function of the number of servers for different values of the number of dispatcher

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

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

which proves the second property. ■

In figure 5, 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 of the PoA. For this, we first need the following result.

Lemma 5: For a server farm with two server classes and K dispatchers,

$$\lim_{S\to\infty} PoA(K,S) = \frac{K}{2\sqrt{K}-1}. \tag{14}$$

Proposition 5: For a server farm with two server classes and K dispatchers, and for all K and S,

$$PoA(K,S) \le \min\left(\frac{K}{2\sqrt{K}-1}, S\right). \tag{15}$$

Proof: From proposition 4, $PoA(K,S)$ is increasing with S. Combining this fact with lemma 5, we can conclude that

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

Moreover, it was shown in [8] that, for the Wardrop case which is the limit of $K \to \infty$, $PoA(\infty,S) \le S$. Thus,

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

We can deduce the desired result from the above two inequalities. ■

In server farms with large number of servers, it follows from lemma 5 that the PoA will be $\frac{K}{2\sqrt{K}-1}$. In [10], 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 non-atomic case in [8] comes into play when there are few servers and a relatively large number of dispatcher. 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 (15).

V. Conclusions

Price of Anarchy is an oft-used worst-case measure of the inefficiency of non-cooperative decentralized architectures. In spite of its popularity, we have shown that the Price of Anarchy is an overly pessimistic measure that does not reflect the performance obtained in most instances of the problem. For an arbitrary architecture in the system, we have seen that, contrary to a common belief, the *inefficiency* is in general not achieved in heavy-traffic or close to saturation conditions. Surprisingly, we have shown that *inefficiency* might be achieved at arbitrarily low load. In the case of two classes of servers, we have explicitly characterized the traffic conditions (or load) associated with the *inefficiency*. This has allowed us to obtain a refined upper bound on the Price of Anarchy and to show that non-cooperative load-balancing has close-to-optimal performances in most cases. The worst-case performances given by the Price of Anarchy occur only in a very specific setting, namely, when the slower servers are infinitely more numerous and infinitely slower than the faster ones. In future research we plan to generalize some of the results to an arbitrary number of classes of servers. It will also be worthwhile to investigate what happens when the number of dispatchers grows to infinity, that is, when the equilibrium traffic pattern is characterized by the so-called Wardrop equilibrium.

VI. Acknowledgements

This work has been partially supported by grant ANR-11-INFR-001.

References

[1] R. Miller. Who has the most web servers? [Online]. Available: http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/

[2] ——. Google unveils its container data center. [Online]. Available: http://www.datacenterknowledge.com/archives/2009/04/01/google-unveils-its-container-data-center/

[3] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[4] D. Mosk-aoyama, T. Roughgarden, and D. Shah, "Fully distributed algorithms for convex optimization problems," *SIAM Journal on Optimization*, 2010.

[5] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. MIT Press, May 2005.

[6] J. Wardrop, "Some theoretical aspects of road traffic research," *Proceedings of the Institute of Civil Engineers*, vol. 1, pp. 325–378, 1952.

[7] E. Koutsoupias and C. H. Papadimitriou, "Worst-case equilibria." in *STACS 1999*, 1999.

[8] M. Haviv and T. Roughgarden, "The price of anarchy in an exponential multi-server," *Operations Research Letters*, vol. 35, pp. 421–426, 2007.

[9] C. H. Bell and S. Stidham, "Individual versus social optimization in the allocation of customers to alternative servers," *Management Science*, vol. 29, pp. 831–839, 1983.

[10] U. Ayesta, O. Brun, and B. J. Prabhu, "Price of anarchy in non-cooperative load-balancing games," *Performance Evaluation*, vol. 68, pp. 1312–1332, 2011.

[11] H. L. Chen, J. Marden, and A. Wierman, "The effect of local scheduling in load balancing designs," in *Proceedings of IEEE Infocom*, 2009.

[12] Y. A. Korilis, A. A. Lazar, and A. Orda, "Achieving network optima using stackelberg routing strategies," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 5, no. 1, February 1997.

[13] Y. Korilis, A. Lazar, and A. Orda, "Architecting noncooperative networks," *IEEE J.Sel. A. Commun.*, vol. 13, no. 7, September 2006.

[14] J. Doncel, U. Ayesta, O. Brun, and B. Prabhu, "On the Efficiency of Non-Cooperative Load Balancing," Tech. Rep. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00768339

[15] D. Monderer and L. S. Shapley, "Potential games," *Games and Econ. Behavior*, vol. 14, pp. 124–143, 1996.