Limits of Energy Saving for the Allocation of Data Center Resources to Networked Applications

Xavier León and Leandro Navarro
Departament d'Arquitectura de Computadors,
Universitat Politècnica de Catalunya,
Barcelona, Spain
{xleon, leandro}@ac.upc.edu

***Abstract*—Energy related costs are becoming one of the largest contributors to the overall cost of operating a data center, whereas the degree of data center utilization continues to be very low. Energy-aware dynamic provision of resources based on the consolidation of existing application instances can simultaneously address under-utilization of servers while highly reducing energy costs. Thus, energy costs cannot be treated separately from resource provision and allocation. However, current scheduling techniques based on market mechanisms do not specifically deal with such scenario. In this paper we model the problem of minimizing energy consumption of the allocation of resources to networked applications as a Stackelberg leadership game to find an upper bound of energy saving. The model is applied to a proportional-share mechanism where resource providers can maximize their profit by minimizing energy costs while users can select resources ensuring the minimum requirements are satisfied. We show that our mechanism can determine the optimal set of resources on and off, even in realistic conditions considering incomplete information, and heterogeneous applications.**

I. Introduction

Energy consumption is a key concern in networked computing systems, including service overlays, content distribution systems and many other distributed systems. These systems require a collection of networked computing resources from one or multiple providers on data centers spread over the world.

All data centers or cloud computing providers face the problems of high energy costs, a discouraging low server utilization, and changing resource demand of applications. For instance, Hoelzle et al. [1] look at the average CPU utilization of 5,000 Google servers during a 6-month period. It shows that, on average, servers spend relatively little aggregate time at high load levels, but most of the time at the 10–50% CPU utilization range where they are most inefficient in terms of energy.

Two popular methods for effectively matching power consumption to workload requirements in a data center are via *processor speed scaling* and *powering down* nodes.

The goal of this paper is to develop a theoretical framework to analyze the limits of energy saving, and strategies to determine the right set of computing nodes on and off to minimize energy consumption while keeping the right level of service for networked applications using these computational resources. We consider several characteristics of modern applications and resources –e.g. proportional share of resources, granularity of application processes, level of parallelism, and migration and consolidation of virtual machines.

The focus and contributions of this paper are as follows: i) we derive a simple energetic model for computing elements based on SPEC benchmarks using real data; ii) we model the problem of energy minimization in resource allocation as a Stackelberg competition game to derive an algorithm to compute the upper bound on energy saving, which optimizes both energy consumption and resource requirements of applications; iii) although the model assumes complete information, we also explore the effect of having incomplete information, a key aspect in large-scale systems; iv) we also conduct a comprehensive evaluation by simulation looking at the influence of users' budgets, its diversity, profit maximization and energy minimization.

In this paper we present an energy consumption model for computing elements in a data center in Section II. Sections III and IV establish a competition model, with complete and incomplete information between a resource provider and consumers. Section V provides the experimental results and findings based on simulation. Section VI presents the related work, conclusions and discuss future work.

II. Model

A. Energy consumption model

In this section we introduce a model for energy consumption of computing elements present in a data center. The energy curve of a server can be characterized in terms of the CPU load by:

$$f_e(s) = \sigma_e + \mu_e s^{\alpha}$$

where σ_e represents the fixed cost of maintaining a server powered on and available and μ_e, α are parameters of the device which characterizes its dynamic energy consumption as a function of the load $s \in [0, 1]$.

To understand the current trend in power consumption over the last years, we examined a comprehensive set of 500 server profiles publicly available through the SPEC (Standard Performance Evaluation Corporation) website [2] using the `SPECpower_ssj2008` benchmark. It is an industry-standard benchmark that evaluates the power and performance characteristics of volume server class and multi-node class computers. The results of the fitting procedure of the data to the above energy function are presented in Table I.

The value of μ_e should be understood as the maximum amount of energy consumed by a device at full capacity without considering the energy consumed when the server is idle. The trend is to decrease the *idle* consumption while the *dynamic* consumption is increased. This trend is due to the increasing interest from companies to achieve *energy proportionality*, which refers to the goal in which the amount of energy consumed by a device is proportional to the workload supported.

However, we also observe that the values of the fixed term σ_e are still an order of magnitude greater than μ_e in the worst

TABLE I
ESTIMATE OF SERVER PARAMETERS (IN WATTS)

Parameter	2008	2009	2010	Mean
σ_e	133.60	120.14	96.32	126.46
μ_e	1.39	5.31	10.82	4.55
α	0.95	0.83	0.70	0.86

case –i.e. when the node is fully used. From this fact, we can assume that the fixed term is dominant compared to the term proportional to the load in the energy curve. Thus, we argue that most savings in large-scale infrastructures would be achieved if we consider the problem of powering up or down nodes instead on focusing on scheduling decisions based on dynamic power scaling of nodes.

B. Resource allocation model

Sharing of computing elements (resources) can significantly increase system's utilization by combining in a single resource different types of workloads with complementary resource requirements. Although our results are quite independent of the specific resource allocation model, we have decided to employ a market-based way of allocating computing resources in contrast to traditional centralized schedulers.

The rationale behind this decision is the complexity of allocating efficiently applications to resources given that each user has different requirements –i.e. private information which users' are not willing to disclose– and, therefore, centralized schedulers which optimize resource allocation have a high cost in terms of information needed and computational complexity. The use of market-based mechanisms allows such complete information to be reduced to a single piece of information –i.e. price– which allows the system to scale.

The simplest and most appealing market-based mechanism for shared divisible resources is the *proportional share allocation mechanism*, where each user submits bids for the different resources and receives a fraction proportional to user's bid and inversely proportional to the sum of all other users' bids for the same resource. More formally, the price of the resource is $Y_j = \sum_{i=1}^{k} x_{ij}$, where k is the number of bids on resource j and x_{ij} is a non-negative bid of user i for resource j. Following the proportional share allocation mechanism, user i receives a fraction $r_{ij} = \frac{x_{ij}}{Y_j}$ of resource j. This mechanism is straightforward for users to understand the response of the system to a specific bid. Besides, current virtual machine systems already provide proportional share mechanisms which makes even simpler its implementation from the technological point of view [3].

As introduced previously, our aim is to minimize energy consumption considering energy costs. We extend this mechanism to consider the state of a resource –i.e. *on* or *off* state. Therefore, the fraction of a resource received by user i is actually $r_{ij} = q_j \frac{x_{ij}}{Y_j}$, where $q_j \in \{0, 1\}$ is a binary variable representing the state of resource j –i.e. 0 if the resource is shut down and 1 if it is powered up.

III. STACKELBERG COMPETITION MODEL

As the provider is in a privileged position to decide in advance which resources keep on-line and which ones shut down, it is natural to model the data center economy, where users and providers interact considering energetic costs, as a Stackelberg game [4] of participation on two levels –the infrastructure operator in the leader role determining which resources to keep on and off and a set of strategic users buying resources as followers. Following that, we present the respective payoff functions of users and the infrastructure operator, and efficient algorithms to compute the best response of both roles considering each others' actions. This model will allow us to provide lower and upper bounds for metrics of interest to both users and provider.

A. User model

To model users[1] behavior, we consider the utility function presented in [5], a linear payoff function $U_i(r_{i1}, \ldots, r_{im}) = q_{i1}w_{i1}r_{i1} + \ldots + q_{im}w_{im}r_{im}$, where r_{ik} is the resource share obtained from resource k, $w_{ik} \geq 0$ is user i's private preference for resource k, and $q_{ik} \in \{0, 1\}$ represents whether the resource is off or on respectively. However, following the trends of resource providers of providing homogeneous resources to users in the form of virtual machines, these weights or preferences no longer apply and thus $\forall_{j,k} w_{ij} = w_{ik}$.

As typically, we also assume that users are selfish and strategic –i.e. they all act to maximize their own utility defined by their payoff functions. The best response of an agent given a set of resource prices is to find the set of bids on resources that maximizes their utility function. More formally, the best response is the solution to the following optimization problem (1):

$$\max\ U_i(x_1, \ldots, x_m) = \sum_{j=1}^{m} q_{ij} \frac{x_{ij}}{x_{ij} + y_j} \tag{1}$$

$$\text{subject to } \sum_{j=1}^{m} x_{ij} \leq X_i$$
$$\frac{x_{ij}}{x_{ij} + y_j} \geq \phi_i \in (0, 1]$$
$$|S| \geq \tau_i$$

Restrictions: We assume several restrictions for the optimal solution of each agent. The first one is that each user i has a budget (or money) constraint X_i and its total positive bids have to sum up to match its budget. This constraint represents the fact that each user does not own an infinite budget, as opposed to standard assumptions in traditional economic models.

The second restriction assumes a certain application granularity. An application i needs at least a minimum share at each node specified by ϕ_i. Therefore, the optimization problem should select those resources in which this constraint is satisfied. Notice that this restriction may be reformulated in terms of the minimum bid to announce: $x_{ij} \geq \frac{y_j \phi_i}{1 - \phi_i}$.

The third restriction is on the level of parallelism[2]. Let $S = \{r_{ij} : r_{ij} \geq \phi_i\}$ be the set of nodes in which the second restriction is satisfied. Then, we assume that an application has a minimum level of parallelism (defined as the cardinality of S) of up to τ_i nodes. If this restriction is not satisfied, the user has no incentive to participate in the resource market as its requirements will not be met. To simplify the notation

[1]We may use the word *user* and *application* indistinctly through the paper to describe an entity which consumes resources from the infrastructure.

[2]Parallelism understood as the need for several instances of a virtual machine in separate nodes. This need would arise when dealing with fault-tolerance, load-balancing, etc.

Fig. 1. User i's best response algorithm

```
Require: $\phi$ {user $i$'s minimum share}
Require: $X$ {user $i$'s budget}
Require: $\{y_1, \ldots, y_m\}$ {list of resource prices}
Require: $\{q_1, \ldots, q_m\}$ {list of resource states (on/off)}
  $M = \{y_j : q_j = 1\}$ {list of prices of on machines}
  Sort the set $M$ by $y_j$ in increasing order
  Compute largest $k$ such that
      $\frac{\sqrt{y_k}}{\sum_{i=1}^{k}\sqrt{y_i}}(X + \sum_{i=1}^{k} y_i) - y_k \geq \frac{y_j\phi}{1-\phi}$
  Set $x_j = 0$ for $j > k$, and for $1 \leq j \leq k$, set:
      $x_j = \frac{\sqrt{y_j}}{\sum_{i=1}^{k}\sqrt{y_i}}(X + \sum_{i=1}^{k} y_i) - y_j$
  return $(x_1, \ldots, x_m)$
```

through the rest of the paper, we denote this restriction as a boolean named *satisfaction*.

The second and third restrictions improve the representation of real distributed applications. Previous models did not consider that an optimal allocation for the above problem would contain such a small share that is not possible (or impractical) to be allocated in a real computing resource.

Although this model is more realistic in terms of application requirements, we still do not consider the cost in time of moving virtual machines between nodes. However, recent results for live migration in several virtual machine monitors show migration latencies at the millisecond level, supporting our assumption of disregarding this cost [6].

User's best response algorithm: Figure 1 summarizes the algorithm for computing the best response –i.e. the bidding vector $x^* = (x_1^*, \ldots, x_m^*)$ that maximizes the payoff function U_i– considering the above optimization problem (see [5] for details). In short, it determines the maximum number of bids x_j on a resource j such that $x_j \geq \frac{y_j\phi_i}{1-\phi_i}$.

B. Providers model

The payoff function of the resource provider is that of maximizing its profit considering the actual income (price paid by users for each resource) and the actual cost of maintaining the infrastructure (price paid by the provider for its *on* resources).

In this game, the provider acts as the leader (deciding the values of q_j) and users act as followers deciding their optimal bidding vector once the values of q_j are announced.

More formally, the best response is the solution to the following optimization problem (2):

$$\begin{aligned} \max \quad & P(q_1, \ldots, q_m) = \sum_{j=1}^{m}(q_j \sum_{i=1}^{n} x_{ij}) - \sum_{j=1}^{m} q_j c_j \quad (2) \\ \text{subject to} \quad & \forall_i\, |S_i| \geq \tau_i \end{aligned}$$

Restrictions: We assume that the provider has no incentive to perform admission control on users given that there are enough resources to accommodate user's aggregate demand. In other words, the provider's restriction is to satisfy every user –i.e. minimum resource requirements are met– if there are enough resources. An interesting line of future work would be to consider admission control on users to select users which provide higher income with a minimum associated cost to increase the profit. However, in this paper we consider that admission control is made beforehand through a contract between a user and the provider. Therefore, our goal is to satisfy all contracts already admitted.

Fig. 2. Provider's best response algorithm

```
Require: $\{\phi_1, \ldots, \phi_n\}$ {users minimum share}
Require: $\{\tau_1, \ldots, \tau_n\}$ {users minimum number of nodes}
Require: $\{X_i, \ldots, X_n\}$ {users budget}
Require: $M = \{c_1, \ldots, c_m\}$ {list of resource costs}
  Sort the set M by $c_j$ in increasing order
  k=0
  repeat
    $k \leftarrow k + 1$
    Set $q_j = 1$ for $j \leq k$
    Set $q_j = 0$ for $j > k$
    repeat
      for all user $i$ do
        UpdateBestResponse($\phi_i$, $\tau_i$, $X_i$)
      end for
    until convergence
  until $\forall_i\, |S_i| \geq \tau_i$ OR $k = m$
  return $(q_1, \ldots, q_m)$
```

Provider's best response algorithm: The Stackelberg model can be solved by finding the subgame perfect Nash equilibrium (SPNE) –i.e. the strategy profile that serves best each player, given the strategies of other players and that entails every player playing in a Nash equilibrium.

A common method for determining subgame perfect equilibrium in the case of a finite game is backward induction. To calculate the SPNE, the best response functions of the follower must first be calculated. As we already provide an efficient algorithm to compute the follower's best response, we now present an algorithm to compute the best response of the provider (the leader in our game).

To compute the best response, we first sort machines by its energetic cost in increasing order. Without loss of generality, suppose that $c_1 \leq c_2 \leq \ldots c_m$. Suppose that $q^* = (q_1^*, \ldots, q_m^*)$ is the optimal solution (values of 0 and 1 representing *off* and *on* machines).

Claim 1. *If $q_i^* = 0$, then it must hold that for any $j > i$, $q_j^* = 0$.*

Proof: (Sketch) By the optimality constraint in Equation 2, we have that $\forall_i\, |S_i| \geq \tau_i$, and therefore, each user participating in the market spends all its budget in the optimal subset of *on* resources. Therefore, the income is a constant value equal to $\sum_{i=1}^{n} X_i$ and we need to minimize $\sum_{j=1}^{m} q_j^* c_j$ to maximize $P(q_1^*, \ldots, q_m^*)$.
Given that the set of machines is sorted in ascending order of energetic cost, q^* already contains the nodes with a lowest cost and therefore maximizes the provider's revenue. □

The remaining question is to determine the number of *on* nodes –i.e. nodes with $q_j = 1$. It is the minimum number of nodes such that each user is satisfied[3] with its allocation. Thus, the algorithm derived to compute the best response of a provider is presented in Figure 2.

Intuitively, it checks if the SPNE when adding a single machine into the set of *on* machines satisfies the *satisfaction*

[3]Recall that the satisfaction of each user is determined by its minimum requirements in terms of share in a node ϕ and number of minimum nodes τ

constraint for all users. This algorithm stops when every user is satisfied with its allocation or the provider needs to keep every machine on.

IV. Strategies with Incomplete Information

The algorithm presented in Figure 2 is useful for understanding the upper and lower bounds on metrics of interest for the provider. However, the main problem with this model is the complete information required by the provider in order to compute the SPNE. Thus, we present two strategies which use only aggregated information (either economic or technical) available to the provider –i.e. without knowing the private information of each user.

Proportional cost threshold strategy: In this case, the provider will decide to shutdown those machines which do not cover its own energetic cost. More formally, it will shutdown the set of machines $OFF = \{j : \forall_j c_j > \sum_i^n x_{ij}\}$. The intuitive idea behind this is to shutdown those nodes which do not report a positive profit for the provider because of its high energetic cost.

Demand aware strategy: In this case, instead of relying on the energetic cost of each resource, the provider will make use of the aggregate demand present in its resource pool. This way, it will compute a rough number of nodes which could potentially carry out the current load. More formally, the number of nodes k to maintain on-line, considering a mean resource usage of $p \in [0, 1]$ and a confidence parameter $\alpha \in [0, \infty]$, will be $k = (1 + \alpha) * p * m$.

Then, as in the complete information case, we sort machines by cost in increasing order and set $q_j = 1$ for $j \leq k$ and $q_j = 0$ for $j > k$. Intuitively, we are *estimating* the optimal number of machines necessary to satisfy every user. The parameter α is related to the confidence on the estimation of the mean resource usage p. Besides, α represents the trade-off between user's satisfaction and the reduction of energetic costs. Thus, a higher α will lead to an *overestimation* of the necessary on-line resources to satisfy users, while a lower α will be more aggressive and risk user's insatisfaction at the benefit of lower costs.

V. Experimental Results

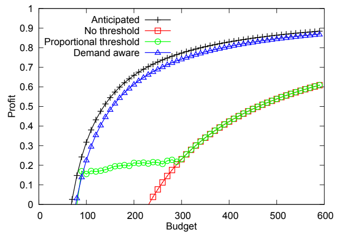
In this section, we provide the detailed experimental findings. In our simulations we fix the number m of nodes to 1000 and a fixed number n of users to 250. We assign a individual cost of c_j in Watts to each node drawn from a uniform distribution $U \sim (96.32, 133.60)$ extracted from real measurements summarized in Table I. We assume that the economic cost of each Watt is one monetary unit.

We compare the behavior of the strategies presented previously by varying users' budget and its minimum requirements ϕ and τ. Following, we present the metrics under study.

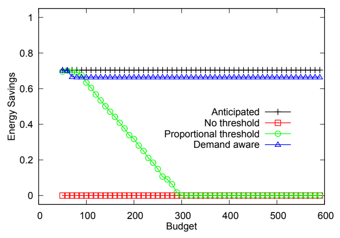
Profit. It is the outcome of the providers utility's function defined in Equation 2. That is, it is the total income provided by users by bidding on available machines minus the energetic cost of these available machines. As we are not interested in the actual profit but the ratio between the profit and the available money in the system –i.e. the money owned by users– we normalize the resulting profit by $\sum_i^n X_i$.

Energy saving. It is the ratio between the cost saving of shutting down nodes and the total cost if all nodes were available.

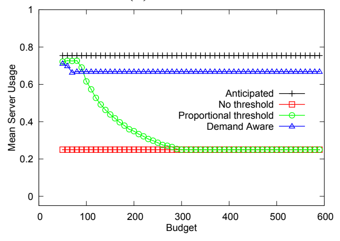
Server usage. It is the mean server usage of the whole infrastructure. In our simulations, users consume a share up



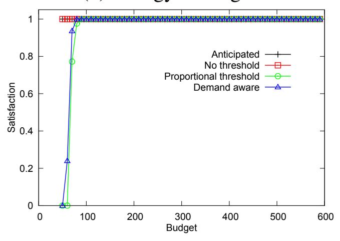
(a) Profit



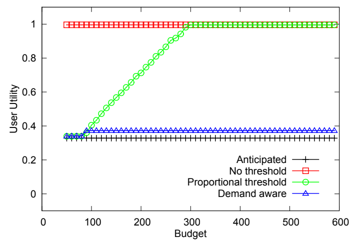
(b) Energy Savings



(c) Server Usage



(d) Satisfaction



(e) User Efficiency

Fig. 3. Results with variable budget per user

to ϕ of a single resource. For example, if a user's share on a resource is $r_{ij} = 0.7$ and its $\phi = 0.3$, then 40% of that resource will be unused. Recall that $r_{ij} \geq \phi$ is always true by the optimality restrictions of user's payoff function presented in Equation 1. Instead of a percentage, we present a ratio in the range $[0, 1]$ for comparison purposes.

Satisfaction. This metric shows the efficiency of our strategies in terms of user satisfaction. It is the ratio between the number k of satisfied users (those fulfilling their satisfaction criteria) and the total number n of users.

User efficiency. User efficiency or price of anarchy (PoA) is defined as the ratio between the social welfare $U(\omega)$ considering an actual allocation ω and the social welfare at the optimal allocation U^* where $U^* = \max(U(\omega))\ \forall\omega$. For an explanation on how to compute the optimal social welfare U^* we refer the reader to [5].

For the sake of comparison, we introduce another simple strategy which is to maintain all nodes always available (*No threshold* in the figures). This strategy will allow us to compare the behavior of different strategies and provide a lower bound on how the system would behave no matter the strategy followed by the provider.

We first experiment with homogeneous users –i.e. running equivalent applications. Thus, for each user i we fix $\phi_i = 0.1$ and a $\tau_i = 5$. This way, we introduce a constant light-weight workload which could be supported by a subset of the whole infrastructure. Through subsequent iterations, we increase the budget of each user in the range $[60, 600]$ with steps of 10 monetary units. The expected behavior is that the more budget is available to users, the more resources they buy increasing its own utility.

The results show that for the simplest strategy of maintaining the whole infrastructure available (*No threshold* in Figures 3(a)-(e)) the provider has no incentive to participate in the market –i.e. its profit is less than zero– until the level of income is at least equal to the cost of maintaining the infrastructure (around 220 monetary units per user). Besides, the energy savings and server mean usage are the lowest possible as the whole infrastructure is active and the workload is constant. However, in this case, user's satisfaction and efficiency are the highest possible as there are always enough resources to meet each one's requirements. This scenario is interesting as it provides lower bounds for energy savings and profit while providing upper bounds on user's satisfaction and user's utility.

In contrast, we look at the strategy derived from the Stackelberg game (*Anticipated* in Figures 3(a)-(e)) which is proven to be the best response of the Stackelberg game. Figure 3(a) shows that the more budget is available to users the more profit is obtained by the provider because it is able to adjust the number of available machines for a given workload (which is fixed in these simulations) and, at the same time, the mean server utilization (Figure 3(c)) is higher than in the simplest strategy because of the consolidation of applications into a subset of the infrastructure. More importantly, this adjustment of available resources to a given workload implies a drastic reduction in energetic costs making the infrastructure more environmentally and economically efficient (Figure 3(b)). However, this privileged position gives the provider a great advantage over users as shown in Figure 3(e). In this case, the provider's profit is maximized whereas the efficiency considering user's utility is decreased to the minimum possible considering its minimum requirements of *satisfaction*. This scenario, unrealistic for the need of complete information, is interesting as it provides an upper bound on energy savings, profit and mean server usage whereas it provides a lower bound for user efficiency.

Whatever other strategy should lie between these two scenarios. The *proportional threshold* strategy behaves similarly to the *anticipated* strategy when there is a small amount of money in the system, whereas its performance decreases as more money is introduced. For example, Figure 3(a) shows that the profit slowly increases as more budget is introduced by users in contrast with the rapid increase of the *anticipated* strategy. This is because users are able to pay for the specific costs of each machine as more money they have, independently of the workload. We can see a point of *saturation* (around a budget of 300 monetary units) in which users have enough money to wake up all the resources of the infrastructure and then the system behaves as the *no threshold* strategy.

Interestingly, Figures 3(a)–3(e) also show the behavior of the *demand aware* strategy in which the infrastructure considers the aggregate demand to estimate the number of nodes to shutdown with an $\alpha = 0.25$. This value of α represents a trade-off between provider's satisfaction (increasing revenue) and user's satisfaction. This scenario is also interesting as tuning this parameter would allow system administrators to decide at which point of the solution space operates between the *no threshold* (lower bound) and *anticipated* (upper bound) strategies, and using only aggregated information. This information is commonly present in currently deployed monitoring infrastructures.

VI. Related Work and Conclusions

Most of current work focus on scheduling techniques and architectures for consolidating virtual machines in fewer servers to reduce energy consumption. Takeda et al. [7] and Verma et al. [8] propose mechanisms for virtual machine placement considering power and cost-aware algorithms. These works primary concern is the design and evaluation of a specific piece of software with specific workloads without considering the economic issues behind user's behavior and the cost of managing complete information to meet applications' requirements. At the network level, different techniques to reduce energy consumption have been proposed. Most of the work focus on exploring models related to speed scaling and power down methods [9] and energy minimization when operational voltage and frequency of transmission can be scaled [10].

This paper investigates the limits of energy saving for the allocation of data center resources. We define an energy consumption model of resources obtained from real available data and we model the data center economy as a Stackelberg game which allows us to derive algorithms to compute the limits of energy savings and profit of the resource provider (leader) considering the competition among users to obtain resources (followers). The results can be extended in many directions. Energy-aware admission control of users might help to select the most profitable set of users in terms of income and associated cost. The resource model can be extended looking at nodes with multiple power levels depending on load instead of just switching them on and off. Besides, the intersection of energy minimization at the network level and server level should go together. Another direction is exploring competition models with incomplete information about the strategy of users (Bayesian games).

Acknowledgment

This work is being supported by a FPI-UPC grant from the Universitat Politècnica de Catalunya (UPC) and partially supported by the Spanish Government (project TIN2010-20140-C03-01) and the Generalitat de Catalunya (2009-SGR-1167).

References

[1] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 2009.

[2] "Spec power benchmarks," 2010. [Online]. Available: http://www.spec.org/power_ssj2008/results/

[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003, pp. 164–177.

[4] D. Fudenberg and J. Tirole, "Game Theory MIT Press," *Cambridge, MA*, 1991.

[5] M. Feldman, K. Lai, and L. Zhang, "The proportional-share allocation market for computational resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1075–1088, 2009.

[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2005, pp. 25–25.

[7] S. Takeda and T. Takemura, "A rank-based vm consolidation method for power saving in datacenters," *IPSJ Online Transactions*, vol. 3, pp. 88–96, 2010.

[8] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in *Middleware '08: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2008, pp. 243–264.

[9] M. Andrews, A. Fernandez, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *INFOCOM'10: Proceedings of the 29th conference on Information communications*. Piscataway, NJ, USA: IEEE Press, 2010, pp. 2435–2443.

[10] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008, pp. 323–336.