Approximation Algorithms for Energy Minimization in Cloud Service Allocation under Reliability Constraints

Olivier Beaumont, Philippe Duchon, Paul Renaud-Goud

Approximation Algorithms for Energy Minimization in Cloud Service Allocation under Reliability Constraints

Olivier Beaumont
Inria
Bordeaux, France
Email: Olivier.Beaumont@inria.fr

Philippe Duchon
University of Bordeaux
Bordeaux, France
Email: Philippe.Duchon@labri.fr

Paul Renaud-Goud
Inria
Bordeaux, France
Email: paul.renaud-goud@inria.fr

***Abstract*—We consider allocation problems that arise in the context of service allocation in Clouds. More specifically, we assume on the one part that each computing resource is associated with a capacity, that can be chosen using the Dynamic Voltage and Frequency Scaling (DVFS) method, and with a probability of failure. On the other hand, we assume that the services run as a set of independent instances of identical Virtual Machines (VMs). Moreover, there exists a Service Level Agreement (SLA) between the Cloud provider and the client that can be expressed as follows: the client comes with a minimal number of service instances that must be alive at anytime, and the Cloud provider offers a list of pairs (price, compensation), the compensation having to be paid by the Cloud provider if it fails to keep alive the required number of services. On the Cloud provider side, each pair actually corresponds to a guaranteed reliability of fulfilling the constraint on the minimal number of instances.**

In this context, given a minimal number of instances and a probability of success, the question for the Cloud provider is to find the number of necessary resources, their clock frequency and an allocation of the instances (possibly using replication) onto machines. This solution should satisfy all types of constraints (both capacity and reliability constraints). Moreover, it should remain valid during a time period (with a given reliability in presence of failures) while minimizing the energy consumption of used resources. We assume in this paper that this time period, that typically takes place between two redistributions, is fixed and known in advance. We prove deterministic approximation ratios on the consumed energy for algorithms that provide guaranteed reliability and we provide an extensive set of simulations that prove that homogeneous solutions are close to optimal.

***Keywords*—Cloud, reliability, approximation, energy savings**

I. Introduction

A. Reliability and Energy Savings in Cloud Computing

This paper considers energy savings and reliability issues that arise when allocating instances of an application consisting in a set of independent services running as Virtual Machines (VMs) onto Physical Machines (PMs) in a Cloud Computing platform. Cloud Computing [1]–[4] has emerged as a well-suited paradigm for service providing over the Internet. Using virtualization, it is possible to run several Virtual Machines on top of a given Physical Machine. Since each VM hosts its complete software stack (Operating System, Middleware, Application), it is moreover possible to migrate VMs from a PM to another in order to dynamically balance the load.

In the static case, mapping VMs with heterogeneous computing demands onto PMs with (possibly heterogeneous) capacities can be modeled as a multi-dimensional bin-packing problem. Indeed, in this context, each physical machine is characterized by its computing capacity (*i.e.* the number of flops it can process during one time-unit), its memory capacity (*i.e.* the number of different VMs that it can handle simultaneously, given that each VM comes with its complete software stack) and its failure rate (*i.e.* the probability that the machine will fail during the next time period) and each service comes with its requirements, in terms of CPU and memory demands, and reliability constraints.

In order to deal with capacity constraints in resource allocation problems, several sophisticated techniques have been developed in order to optimally allocate VMs onto PMs, either to achieve good load balancing [5]–[7] or to minimize energy consumption [8], [9]. Most of the works in this domain have therefore concentrated on designing offline [10] and online [11], [12] solutions of Bin Packing variants.

Reliability constraints have received much less attention in the context of Cloud computing, as underlined by Walfredo Cirne in [13]. Nevertheless, related questions have been addressed in the context of more distributed and less reliable systems such as Peer-to-Peer networks. In such systems, efficient data sharing is complicated by erratic node failure, unreliable network connectivity and limited bandwidth. Thus, data replication can be used to improve both availability and response time and the question is to determine where to replicate data in order to meet performance and availability requirements in large-scale systems [14]–[18]. Reliability issues have also been addressed by the High Performance Computing community. Indeed, recently, a lot of efforts has been done to build systems capable of reaching the Exaflop performance [19], [20] and such exascale systems are expected to gather billions of processing units, thus increasing the importance of fault tolerance issues [21]. Solutions for fault tolerance in Exascale systems are based on replication strategies [22] and rollback recovery relying on checkpointing protocols [23], [24].

This work is a follow-up of [25], where the question of how to evaluate the reliability of a general allocation has been addressed and a set of deterministic and randomized heuristics have been proposed. In this paper, we concentrate on energy savings issues and we propose proved approximation

algorithms. In order to minimize energy consumption, we assume that sophisticated mechanisms exist in order to fix the clock frequency of the PMs, such as DVFS (see [26]–[30]). In this context, the capacity of the PM can be expressed as a function of the clock frequency. In general, the probability of failure may itself depend on the clock frequency (see for instance [31]); nevertheless, we did not find in the literature a widely admitted model stating how clock frequency and failures relate and we leave this issue for future works.

To assess precisely the specific complexity of energy minimization introduced by reliability constraints in the context of services allocation in Clouds, we concentrate on a simple context, that nevertheless captures the main difficulties. First, we consider that the applications running on the Cloud platform can be seen as a set of independent services, and that the services themselves consist in a number of identical (in terms of requirements) and independent instances. Therefore, we do not consider the problems introduced by heterogeneity, that have already been considered (see for instance [6], [7]). Indeed, as soon as heterogeneity is considered, basic allocation problems are amenable to Bin Packing problem and are therefore intrinsically difficult. Then, we consider static allocation problems only, in the sense that our goal is to find the allocation that optimizes the reliability during a time period. This time period corresponds to the time period between two phases of migrations and reconfiguration of the allocation of VMs onto PMs. During this time period, the goal for the provider is to ensure that a minimal number of instances of each service is running whatever the machine failures. In order to enforce reliability constraints, the provider will over-provision resources by allocating and running more instances than actually required by the services in order to cope with failures. Combining these static and dynamic phases is out of the scope of this paper. Therefore, our work enables to assess precisely the complexity introduced by machine failures and service reliability demands on energy minimization.

Throughout this paper, we assume that the characteristics of the applications and their requirements (in terms of reliability in particular) have been negotiated between a client and the provider through a Service Level Agreement (SLA). In the SLA, each service is characterized by its demand in terms of processing capability (*i.e.* the minimal number of instances of VMs that must be running simultaneously) and in terms of reliability (*i.e.* the maximal probability so that the service will not benefit from this number of instances at some point during the next time period). Equivalently, the reliability requirement may be negotiated through the payment of a fine by the Cloud Provider if it fails to provide the required amount of resources. In the case where it may be difficult for the user to *a priori* decide the level of reliability, we discuss in Section V how reliability can be proposed by the cloud provider as a list of $(price, compensation)$ pairs. In all cases, the goal, from the provider point of view, is therefore to determine the cost of reliability, since a higher reliability will induce more replication and therefore more energy consumption. Our goal in this paper is to find allocations that minimize energy consumption while enforcing reliability constraints, and therefore to determine the price of reliability.

B. Notations

In this section, we introduce the notations that will be used throughout the paper. Our target Cloud platform is made of m physical machines $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m$. As already noted, we assume that machine $\mathcal{M}_j$ is able to handle the execution of CAPA_j instances of services. We also assume that we can rely on Dynamic Voltage Frequency Scaling (DVFS) mechanism in order to adapt CAPA_j. The energy consumed by machine $\mathcal{M}_j$ when running at capacity (speed proportional to) CAPA_j is given by $E = E_{\text{stat}}(j) + E_{\text{dyn}}(j)$, where $E_{\text{dyn}}(j) = e_j \text{CAPA}_j^{\alpha}$. This means that the energy consumed by machine $\mathcal{M}_j$ can be seen as the sum of a leakage term (paid as soon as the machine is switched on) and of a term that depends (most of the works consider that $2 \leq \alpha \leq 3$) on its running speed. We assume in addition continuous speeds, which means that any CAPA_j can be achieved by machine $\mathcal{M}_j$ (as advocated in [32]–[34]), so that we can obtain readable and interesting results.

On this Cloud platform, our goal is to run (all through a given time period, as defined in the SLA) n services $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$. DEM_i identical and independent instances of service $\mathcal{S}_i$ are required, and the instances of the different services run as Virtual Machines. Several instances of the same service can therefore run concurrently and independently on the same physical machine, even if it lowers the service reliability. We will denote by $\mathcal{A}_{i,j}$ the number of instances of $\mathcal{S}_i$ running on $\mathcal{M}_j$. Therefore, $\sum_i \mathcal{A}_{i,j}$ represents the overall number of instances running on $\mathcal{M}_j$ and therefore, it has to be smaller than CAPA_j. Respectively, $\sum_j \mathcal{A}_{i,j}$ represents the overall number of running instances of $\mathcal{S}_i$. In general, $\sum_j \mathcal{A}_{i,j}$ is larger than DEM_i since replication, *i.e.* over-provisioning of services, is used in order to enforce reliability constraints.

More precisely, each machine $\mathcal{M}_j$ comes with a failure rate FAIL_j, that represents the probability of failure of $\mathcal{M}_j$ during the time period. During the time period, we will not reallocate instances of services to physical machines but rather provision extra instances for the services (replicas) that will actually be used if some machines fail. As said previously, we will assume for the results proved in this paper that FAIL_j does not depend on CAPA_j.

We will denote by ALIVE the set of running machines. In our model, at the end of the time period, the machines are either up or completely down, so that the number of instances of service $\mathcal{S}_i$ running on $\mathcal{M}_j$ is $\mathcal{A}_{i,j}$ if $j|\mathcal{M}_j \in \text{ALIVE}$, and 0 otherwise. Therefore, $\text{ALIVEINST}_i = \sum_{\mathcal{M}_j \in \text{ALIVE}} \mathcal{A}_{i,j}$ denotes the overall number of running instances of $\mathcal{S}_i$ at the end of the time period. In addition, $\mathcal{S}_i$ is running properly at the end of the time period if and only if $\sum_{j|\mathcal{M}_j \in \text{ALIVE}} \mathcal{A}_{i,j} \geq \text{DEM}_i$.

Of course, our goal is not that all instances should run properly at the end of the time period. Indeed, such a reliability cannot be achieved in practice since the probability that all machines fail is clearly larger than 0 in our model. In general, as noted in a recent paper of the NY Times [35], Data Centers usually over-provision resources (at the price of high energy consumption) in order to (quasi-)avoid failures. In our model, we assume a more sustainable model, where the SLA defines the reliability requirement REL_i for service $\mathcal{S}_i$ (together with the penalty paid by the Cloud Provider if $\mathcal{S}_i$ does not run with at least DEM_i instances at the end of the period). Therefore, the Cloud provider faces the following optimization problem:

BestEnergy$(m, n, \text{DEM}, \text{REL})$: Find the set ON of machines that are on and the clock frequency assigned to machine $\mathcal{M}_j$, represented by CAPA_j and an allocation $\mathcal{A}$ of instances of services $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ to machines $\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_m$ such that

$$\text{(i)} \forall j \in \text{ON}, \sum_{i=1}^{n} \mathcal{A}_{i,j} \leq \text{CAPA}_j,$$
$$\text{(ii)} \forall i,\ \mathbb{P}\left(\text{ALIVEINST}_i \geq \text{DEM}_i\right) \geq 1 - \text{REL}_i,$$

i.e. the probability that a least DEM_i instances of $\mathcal{S}_i$ are running on alive machines after the time period is larger than the reliability requirement $1 - \text{REL}_i$,

(iii) the overall energy consumption $\sum_{j \in \text{ON}} E_{\text{stat}}(j) + e_j \text{CAPA}_j^{\alpha}$ is minimized.

C. Methodology

Throughout the paper, we will rely on the same general approach. Through Section II to Section IV, in order to prove claimed approximation ratios, we rely on the following techniques.

For the lower bounds, we prove that for a service, given the reliability constraints of this service and given failure probabilities of the machines, at least a given number of instances, or at least a given level of energy is needed. These results are obtained through careful applications of Hoeffding Bounds [36].

For the upper bounds, we concentrate on a special allocation schemes, namely *Homogeneous*. In a solution of *Homogeneous*, for each service, we assign to every machine the same number of instances, *i.e.* $\forall i, \forall j \in \text{ON}, \mathcal{A}_{i,j} = \mathcal{A}_i$. Using this allocation scheme, we are able to derive theoretical bounds relying on Chernoff bounds [37]. Moreover, the comparison with the lower bound shows that the quality of obtained solutions is reasonably high, especially in the case of energy minimization and even asymptotically optimal when the size of the platform or the overall volume of service instances to be handled, becomes arbitrarily large.

D. Motivating example

In order to illustrate the objective functions that we consider throughout this paper and the notations, let us consider a service with a demand $\text{DEM} = 20$ and a reliability request of $\text{REL} = 4.5 \cdot 10^{-6}$, that has to be mapped onto a Cloud composed of $m = 10$ physical machines, whose failure probability is $\text{FAIL} = 10^{-1}$. Figure 1 depicts the kind of solutions that we consider in this paper. In terms of minimizing the number of instances, the best solution consists in allocating 10 instances of the service to the first 2 machines and 5 instances to the 8 remaining machines. Therefore, the optimal solutions allocate a total of 60 instances, whereas 20 instances only are required at the end of the time period, in order to satisfy reliability constraints. The shape of the optimal solution reflects the complexity of the problem. Indeed, it has been proved in [25] that even in the case of a single service and even if the allocation is given, estimating its reliability is a $\#P$-complete problem. The $\#P$ complexity class has been introduced by Valiant [38] in order to classify the problems where the goal is not to determine whether there exists a solution (captured by NP-completeness notion) but rather to determine the number of solutions. In our context, the reliability of an allocation is related to the number (weighted by their probability) of ALIVE sets that lead to an allocation where all service demands are satisfied. In this example, in order to check that the reliability is larger (in fact equal to) than REL, we can observe that all configurations where at least 4 machines are alive are acceptable (since at least 20 instances are alive as soon as 4 machines are up), together with all configurations with 3 machines, as soon as a machine loaded with 10 instances is involved, and the solution with only the first two machines alive. Counting the number of such valid configurations (weighted by their probability) leads to the reliability of the allocation.

Generally speaking, the question of determining the optimal solution remains open and all the references to the optimal in the paper rely either on comparisons to a lower bound or on exhaustive enumeration of the solutions (for instance, the optimality statement for the example of this section has been obtained through exhaustive search). Nevertheless, we will concentrate on *Homogeneous* solutions, *i.e.* those where all PMs are given the same number of instances. We provide in Section II algorithms to compute the BestHomogeneous solution.

We can notice that the optimal solution involves 60 instances against around 67 for best fractional homogeneous solution. Indeed, the best fractional solution allocates $20/3$ instances to each machine, so that all configurations with 3 alive machines are enough, thus leading to a better reliability (at a higher cost). Note that this case has been determined using exhaustive search among all possible allocations with 10 machines and where the number of instances given to each PM is an integer, so that this example can be seen as a worst case.

As far as energy minimization is concerned, we can notice that if we assume $\alpha = 2$, despite the bad load balancing among the machines in the optimal solution for the number of instances, this solution remains optimal. Indeed,

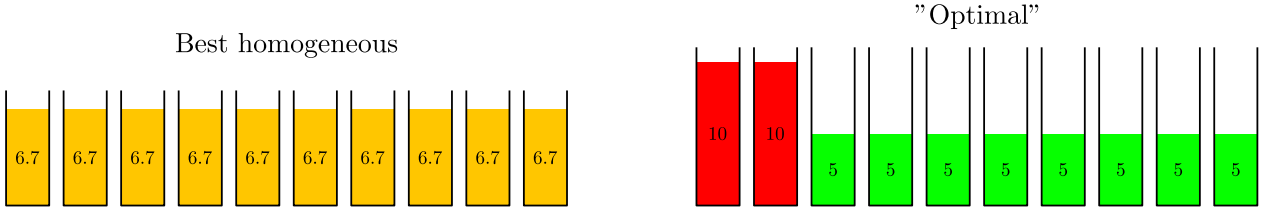


Figure 1. Motivating example

the dynamic energy of the unbalanced solution is given by $2 * 10^2 + 8 * 5^2 = 400$ and the energy of the homogeneous one is given by $10 * (20/3)^2 = 445$. On the other hand, if $\alpha = 3$ for instance, then the homogeneous solution consumes less energy ($10*(20/3)^3 = 2967$) than the unbalanced solution ($2*10^3*8*5^3 = 3000$). Thus, we can observe on this example that minimizing the dynamic energy (rather than minimizing the number of instances) favors homogeneous solutions.

Therefore, in the rest of this paper, we will use fractional homogeneous solutions in order both to derive approximation algorithms and upper bounds on the number of required resources. Indeed, we prove in Section II that homogeneous allocations are asymptotically optimal for dynamic energy minimization when the number of involved PMs becomes large. In Section IV, we provide an extensive set of simulations that prove that homogeneous solutions are in general close to optimal for general energy minimization in a large number of situations.

E. Outline of the Paper

As we have noticed through the motivating example, **BestEnergy** is in general difficult since verifying that a given allocation satisfies a given reliability constraint is already $\#P-$complete. Nevertheless, we prove in this paper that even when the allocation is to be determined, it is possible to provide low-complexity deterministic approximation algorithms, that are even asymptotically optimal when the sum of the demands becomes arbitrarily large. Another original result that we prove in this paper is that minimizing the energy (relying on DVFS) induced by replication is easier than minimizing the number of replicas, whereas in many contexts (see [39]) the non-linearity of energy consumption makes the optimization problems harder. In our context, approximation ratio are smaller for energy minimization than for classical replication minimization (that would correspond to makespan or load balancing in other contexts).

To prove this result, we progressively come to the most general problem through the study of more simple objective functions. Firstly, we consider several models for energy minimization. First, we address in Section II the case where dynamic energy only is concerned, *i.e.* without taking explicitly the leakage term into account. Then, we introduce the static energy part in Section III and the more general MIN-ENERGY problem. For MIN-ENERGY, the setting is the same except that the number of participating machines is to be determined and DVFS can be used to determine the capacity of each machine. At last, in Section IV, we perform some simulations in order to show that homogeneous solutions are in fact very close to optimal.

II. DYNAMIC ENERGY MINIMIZATION USING DVFS

In this section, we concentrate on the dynamic energy minimization problem. Therefore, we assume that the number of resources that are switched on is fixed in advance. Then, since no reallocation or VM migration will take place during the considered period, our goal is to actually run more instances than what is actually required by the demand of the service, so as to cope automatically with machine failures during the period. Indeed, since we are considering services typically serving requests and where the demand is given as a minimal number of request per time unit, it is both sufficient and necessary to enforce that the remaining serving capacity given failures is large enough with the reliability expressed in the SLA .

A. Lower bound

Let us consider the case of a single service to be mapped onto a fixed number of machines when the objective is to minimize the amount of resources necessary to enforce the conditions defined in the SLA in terms of quantity (of alive instances at the end of the time period) and reliability. The problem comes into two flavours depending on the resources we want to optimize. Recall that $\mathcal{A}_j$ is the number of instances of the service initially allocated to machine $\mathcal{M}_j$. In its physical machines version, the optimization problem consists in minimizing the number of instances allocated to the different machines, *i.e.* minimizing $\sum_j \mathcal{A}_j$. In its energy minimization version, we rely on DVFS mechanism in order to adapt the voltage of a machine to the need of the instances allocated to it. In general, energy consumption models assume that the energy dissipated by a processor running at speed s is proportional to s^α. Therefore, the energy dissipated by a processor running $\mathcal{A}_j$ instances will be proportional to $\mathcal{A}_j^\alpha$ and the overall objective is to minimize the overall dissipated energy, *i.e.* $\sum_j \mathcal{A}_j^\alpha$.

In order to find the lower bound, let us consider any allocation (where $\mathcal{A}_j$ is the number service instances initially allocated to machine $\mathcal{M}_j$) and let us prove that if the amount of resources is too small, then reliability constraints cannot be met. Recall that ALIVEINST_j is the number of instances of the service that are alive on machine $\mathcal{M}_j$ at the end of the time period. ALIVEINST_j is thus a random variable equal to $\mathcal{A}_j$ with a probability $1 - \text{FAIL}$ and to 0 with a probability FAIL.

Hence, the expected number of alive instances is given by $\mathbb{E}(\text{ALIVEINST}) = (1 - \text{FAIL}) \sum_{j=1}^m \text{ALIVEINST}_j$. Hoeffding inequality (see [36]) says how much the number of alive resources may differ from its expected value. In particular, for the lower bound, we will use it in the following form, that bounds the chance of being lucky, *i.e.* to find a correct allocation with few instances. More precisely, it states that for all $t > 0$:

$$\mathbb{P}(\text{ALIVEINST} \geq \mathbb{E}(\text{ALIVEINST}) + t) \leq \exp\left(-2\frac{t^2}{\sum_{j=1}^m \mathcal{A}_j^2}\right).$$

Let us choose $t = \sqrt{-ln(1 - \text{REL}) \sum_{j=1}^n \mathcal{A}_j^2/2}$, so that $\exp\left(-2\frac{t^2}{\sum_{j=1}^m \mathcal{A}_j^2}\right) = 1 - \text{REL}$. Noting $K' = \frac{-ln(1-\text{REL})}{2}$, and since $\mathbb{E}(\text{ALIVEINST}) = (1 - \text{FAIL}) \sum_{j=1}^m \mathcal{A}_j$, the previous equation becomes

$$\mathbb{P}\left(\text{ALIVEINST} \geq \sum_{j=1}^m \mathcal{A}_j + \sqrt{K' \times \sum_{j=1}^m \mathcal{A}_j^2}\right) \leq 1 - \text{REL}.$$

Now, if a given allocation succeeds, then, by definition, $\mathbb{P}(\text{ALIVEINST} \geq \text{DEM}) \leq 1 - \text{REL}$.

Thus we obtain that a necessary condition on the $\mathcal{A}_j$'s so that the reliability constraint is enforced is given by

$$(1-\text{FAIL})\sum_{j=1}^{m}\mathcal{A}_j+\sqrt{K'\times\sum_{j=1}^{m}\mathcal{A}_j^2}\geq \text{DEM}.$$

As stated in the introduction of this section, we are interested either in minimizing $\sum_j \mathcal{A}_j$ for resource use minimization, and $\sum_j \mathcal{A}_j^\alpha$ for energy minimization. To obtain lower bounds on these quantities in order to achieve quantitative (number of alive instances) and qualitative (reliability constraints), we rely on Hoelder's inequality, that states that if $1/p+1/q=1$, then

$$\forall a_j,b_j\geq 0,\ \sum_j a_j b_j\leq\left(\sum_j {a_j}^p\right)^{1/p}\times\left(\sum_j {b_j}^q\right)^{1/q}.$$

With $p=q=2, a_j=b_j=\mathcal{A}_j$, we obtain $\sum\mathcal{A}_j^2\leq\left(\sum\mathcal{A}_j\right)^2$, so that

$$(1-\text{FAIL})\sum_{j=1}^{m}\mathcal{A}_j+\sqrt{K'\times\sum_{j=1}^{m}\mathcal{A}_j^2}\leq\left(1-\text{FAIL}+\sqrt{K'}\right)\times\sum_{j=1}^{m}\mathcal{A}_j.$$

Hence a necessary condition in order to satisfy the constraints is given by

$$\sum_{j=1}^{m}\mathcal{A}_j\geq\frac{\text{DEM}}{1-\text{FAIL}+\sqrt{K'}}=\text{MINREP}.$$

Therefore, any solution that satisfies quantitative and qualitative constraints must allocate at least MINREP instances, whatever the distribution of instances onto machines is.

With $p=\alpha,\ 1/q=(1-1/\alpha),\ a_j=\mathcal{A}_j$ and $b_j=1$, we obtain $\sum\mathcal{A}_j\leq\left(\sum\mathcal{A}_j^\alpha\right)^{1/\alpha}m^{1-1/\alpha}$.

Similarly, assuming that $\alpha>2$ hence $\alpha/2>1$, with $p=\alpha/2,\ 1/q=(1-2/\alpha),\ a_j=\mathcal{A}_j^2$ and $b_j=1$, we obtain

$$\sum_{j=1}^{m}\mathcal{A}_j^2\leq\left(\sum_{j=1}^{m}\mathcal{A}_j^\alpha\right)^{2/\alpha}m^{1-2/\alpha},\ \text{ so that}$$

$$(1-\text{FAIL})\sum_{j=1}^{m}\mathcal{A}_j+\sqrt{K'\times\sum_{j=1}^{m}\mathcal{A}_j^2}$$

$$\leq\left((1-\text{FAIL})m^{1-1/\alpha}+\sqrt{K'}m^{1/2-1/\alpha}\right)\times\left(\sum_{j=1}^{m}\mathcal{A}_j^\alpha\right)^{1/\alpha}.$$

Also, we can derive another necessary condition defined as

$$\left(\sum_{j=1}^{m}\mathcal{A}_j^\alpha\right)\geq\left(\frac{\text{DEM}}{(1-\text{FAIL})m^{1-1/\alpha}+\sqrt{K'}m^{1/2-1/\alpha}}\right)^\alpha=\text{MINENERGY},$$

which also holds true for $\alpha=2$.

Therefore, any solution that satisfies quantitative and qualitative constraints must consume at least MINENERGY, whatever the distribution of instances onto machines is.

B. Upper bound – Homogeneous

1) MIN-REPLICATION*:* As explained above, in order to obtain upper bounds on the amount of necessary resources (either in terms of number of instances or energy), it is enough to exhibit a valid solution (that satisfies the constraints defined in the SLA). To achieve this, we will concentrate in this part on homogeneous (fractional) solutions, with an equally-balanced allocation among all machines (*i.e.* $\forall j,\mathcal{A}_j=\mathcal{A}$).

An assignment is considered as failed when there are not enough instances of the service that are running at the end of the time period, hence $\mathbb{P}_{fail}=\mathbb{P}\left(\text{ALIVEINST}<\text{DEM}\right)$. From the homogeneous characteristics of the allocations, we derive that $\text{ALIVEINST}=\mathcal{A}\times|\text{ALIVE}|$, then $\mathbb{P}_{fail}=\mathbb{P}\left(|\text{ALIVE}|<\frac{\text{DEM}}{\mathcal{A}}\right)$. $|\text{ALIVE}|$ can be described as the sum of random independent variables $\sum_{j=1}^{m}X_j$, where, for all $j\in\{1,\ldots,m\}$, X_j depicts the fact that machine $\mathcal{M}_j$ is alive at the end of the time period (X_j is equal to 1 with probability $1-\text{FAIL}$, and to 0 with probability FAIL).

Hence, the expected value of $|\text{ALIVE}|$ is given by $\mathbb{E}\left(|\text{ALIVE}|\right)=(1-\text{FAIL})m$. Chernoff bound (see [37]) says how much the number of alive machines may differ from its expected value. We use in this part Chernoff bounds rather than Hoeffding bounds because the random variables take their value in $\{0,1\}$ instead of $\{0,\ldots,A\}$ and Chernoff bounds are more accurate in this case. In particular, for the upper bound, we will use it in the following form, that bounds the chance of being unlucky, *i.e.* to fail having a correct allocation while allocating a large number of instances. More specifically, Chernoff bound gives that for all $\varepsilon>0$, $\mathbb{P}\left(|\text{ALIVE}|\leq(1-\text{FAIL}-\varepsilon)m\right)\leq e^{-2\varepsilon^2 m}$. As we want to ensure that $\mathbb{P}_{fail}\leq\text{REL}$, we choose ε such that $e^{-2\varepsilon^2 m}=\text{REL}$, i.e. $\varepsilon=\sqrt{K/m}$ by noting $K=\frac{-ln(\text{REL})}{2}$. This allows to rewrite the previous equation into: $\mathbb{P}\left(|\text{ALIVE}|\leq(1-\text{FAIL}-\sqrt{K/m})m\right)\leq\text{REL}$. Finally, we obtain a sufficient condition, so that the reliability constraint is fulfilled for the service:

$$\mathcal{A}m\geq\frac{\text{DEM}}{1-\text{FAIL}-\sqrt{\frac{K}{m}}}=\text{MAXREP},$$

since then

$$\begin{aligned}\mathbb{P}_{fail}&=\mathbb{P}\left(\text{ALIVEINST}<\text{DEM}\right)\\&=\mathbb{P}\left(|\text{ALIVE}|\mathcal{A}<\text{DEM}\right)\\&\leq\mathbb{P}\left(|\text{ALIVE}|\leq(1-\text{FAIL}-\sqrt{K/m})m\right)\\\mathbb{P}_{fail}&\leq\text{REL}.\end{aligned}$$

Therefore, it is possible to satisfy the SLA with at most MAXREP instances of the service. Similarly, we can derive an upper bound of the energy needed to enforce the SLA. Indeed,

with the same value of $\mathcal{A}$, we obtain

$$\mathcal{A}^\alpha m \geq \left(\frac{\text{DEM}}{(1-\text{FAIL})m^{1-1/\alpha} - \sqrt{K}m^{1/2-1/\alpha}}\right)^\alpha = \text{MAXENERGY}.$$

C. Comparison

When minimizing the number of necessary instances to enforce the SLA, we obtain $\frac{\text{MAXREP}}{\text{MINREP}} = \frac{1-\text{FAIL}+\sqrt{K'}}{1-\text{FAIL}-\sqrt{\frac{K}{m}}}$. For realistic values of the parameters, above approximation ratio is good (close to one), since both $\sqrt{K'} = \sqrt{\frac{-ln(1-\text{REL})}{2}}$ and $\sqrt{\frac{K}{m}} = \sqrt{\frac{-ln(\text{REL})}{2m}}$ are small as soon as m is large. Nevertheless, the ratio is not asymptotically optimal when m becomes large.

On the other hand, for energy minimization, we have

$$\frac{\text{MAXENERGY}}{\text{MINENERGY}} = \left(\frac{(1-\text{FAIL})m^{1-1/\alpha}+\sqrt{K'}m^{1/2-1/\alpha}}{(1-\text{FAIL})m^{1-1/\alpha}-\sqrt{K}m^{1/2-1/\alpha}}\right)^\alpha = \left(\frac{(1-\text{FAIL})+\sqrt{\frac{K'}{m}}}{(1-\text{FAIL})-\sqrt{\frac{K}{m}}}\right)^\alpha,$$

so that this ratio tends to 1 when m becomes arbitrarily large. This shows that for energy minimization, homogeneous fractional solutions provide very good results when m is large enough. In the following section, we prove that an allocation with a large dispersion (in a sense described precisely below) of the number of instances allocated to the machines cannot achieve SLA constraints with optimal energy.

D. Can optimal solutions be strongly heterogeneous ?

Above results state that for the minimization of the number of instances and for the minimization of the energy, homogeneous allocations provide good solutions. Nevertheless, we know from the example depicted in Figure 1 that optimal solutions, for both the minimization of the number of instances and the minimization of the energy are not always homogeneous. In the case of energy minimization, the dispersion of an allocation cannot be too large, as stated more formally in the following theorem.

Theorem 1: Let us consider a valid allocation $\mathcal{A}_j$ whose energy is not larger than MAXENERGY, the upper bound on the energy consumed by an homogeneous allocation. Then, if $V' = \frac{\sum(\mathcal{A}_j^2)^{\alpha/2}}{m} - \left(\frac{\sum \mathcal{A}_j^2}{m}\right)^{\alpha/2}$ is used as the measure of dispersion of the $\mathcal{A}_j$'s (related to the $\alpha/2$-th moment of their square values), then

$$m^\alpha V \leq \left(\frac{\text{DEM}}{1-\text{FAIL}-\sqrt{\frac{K}{m}}}\right)^\alpha - \left(\frac{\text{DEM}}{1-\text{FAIL}+\sqrt{K'}}\right)^\alpha.$$

Proof: Let us first introduce $V = \frac{\sum \mathcal{A}_j^\alpha}{m} - \left(\frac{\sum \mathcal{A}_j}{m}\right)^\alpha$. Then $V \geq V'$. Indeed, $V - V' = \left(\frac{\sum \mathcal{A}_j^2}{m}\right)^{\alpha/2} - \left(\frac{\sum \mathcal{A}_j}{m}\right)^\alpha$ that has the same sign as $\left(\frac{\sum \mathcal{A}_j^2}{m}\right)^{1/2} - \left(\frac{\sum \mathcal{A}_j}{m}\right)$ that is non-negative by application of Hoelder's inequality.

Moreover, we have seen that a necessary condition (see Section II-A) for allocation $\mathcal{A}_j$ to be valid is given by

$$(1-\text{FAIL})\sum_{j=1}^m \mathcal{A}_j + \sqrt{K' \times \sum_{j=1}^m \mathcal{A}_j^2} \geq \text{DEM},$$

what induces $(1-\text{FAIL})\left(\frac{\text{MINENERGY}}{m} - V\right)^{1/\alpha} + \sqrt{K'm}\left(\frac{\text{MINENERGY}}{m} - V'\right)^{1/\alpha} \geq \text{DEM}$ and finally $V' < \frac{\text{MINENERGY}}{m} - \left(\frac{\text{DEM}}{(1-\text{FAIL})m-\sqrt{Km}}\right)^\alpha$ or equivalently $m^\alpha V' \leq \left(\frac{\text{DEM}}{1-\text{FAIL}-\sqrt{\frac{K}{m}}}\right)^\alpha - \left(\frac{\text{DEM}}{1-\text{FAIL}+\sqrt{K'}}\right)^\alpha$. ∎

III. OVERALL ENERGY MINIMIZATION

In above section, we have considered the case where the number of used machines is fixed in advance. In this context, the leakage term is paid for all machines, and is a constant. In general, in the context of a Cloud platform, both the set of used resources and the voltage associated to them have to be determined. In this case, given that $k \in \{1, \ldots, m\}$, the goal is to minimize

$$E^{(\text{low})}(k) = k \times E_{\text{stat}} + k \times \left(\frac{\text{DEM}}{(1-\text{FAIL})k + \sqrt{K'k}}\right)^\alpha.$$

In above problem, there is intuitively an interesting compromise to be done. Since $\alpha \geq 2$, the machines are more efficient in terms of requests per watt when running at a low frequency. On the other hand, running the machines at a lower frequency requires a larger number of machines and therefore induces a higher leakage term.

A. Lower bound

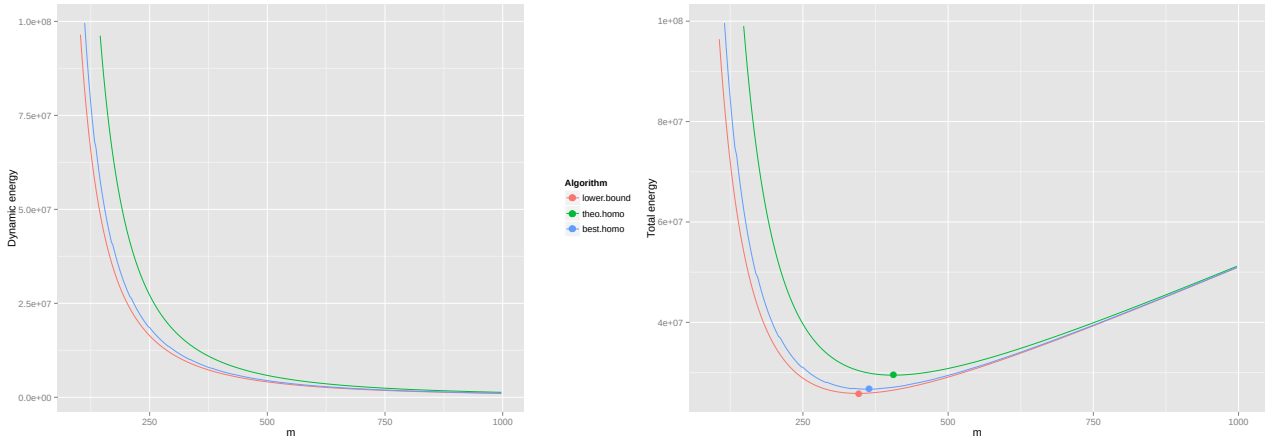
Let g be the function defined on $]0, +\infty[$ by $g(x) = g_t(x)/g_d^\alpha(x)$. Let us prove that if g_d is non-decreasing, concave, positive, and g_t is non-increasing, convex and positive, then g is convex. On the one hand, if g_d fulfills its constraints, then $g_d^{-\alpha}$ is non-increasing, convex and positive, and on the other hand, the product of two non-increasing, convex and positive is a convex function (this can be easily seen on the derivative).

Let us apply above lemma with $g_t(x) = x/x^{\alpha/2}$ (which is convex since $\alpha \geq 2$) and $g_d(x) = (1-\text{FAIL})\sqrt{x} + \sqrt{K'}$, and deduce easily that $E^{(\text{low})}$ is convex.

Therefore, $E^{(\text{low})}$ admits a unique minimum on $[1, m]$. Since $E^{(\text{low})} \underset{0}{\to} +\infty$ and $E^{(\text{low})} \underset{\infty}{\to} +\infty$, $\left(E^{(\text{low})}\right)'$ is null at some point in $[0, +\infty[$, and let us define $x_{\min}^{(\text{low})}$ such that $\left(E^{(\text{low})}\right)'(x_{\min}^{(\text{low})}) = 0$, *i.e.* as

$$E_{\text{stat}} + \left(\frac{\text{DEM}}{(1-\text{FAIL})x_{\min}^{(\text{low})} + \sqrt{K'x_{\min}^{(\text{low})}}}\right)^\alpha \times \left(-(\alpha-1)(1-\text{FAIL}) + \left(1-\frac{\alpha}{2}\right)\sqrt{\frac{K'}{x_{\min}^{(\text{low})}}}\right) = 0. \quad (1)$$

The minimum of function $E^{(\text{low})}$ is reached on $[1, m]$ for $\min(\max(x_{\min}^{(\text{low})}, 1), m)$.



(a) Dynamic energy consumption.

(b) Energy consumption.

Figure 2. Simulation results for $\text{FAIL} = 10^{-2}$, $\text{DEM} = 10^4$, $\text{REL} = 10^{-5}$, $\alpha = 3$, $E_{\text{stat}} = 5 \times 10^4$.

B. Upper bound – Homogeneous

The energy consumption of an *Homogeneous* solution on k machines is given by

$$E^{(\text{up})}(k) = k \times E_{\text{stat}} + \frac{1}{k^{\alpha-1}} \left(\frac{\text{DEM}}{(1 - \text{FAIL}) - \sqrt{\frac{K}{k}}} \right)^{\alpha}.$$

Let us apply again above lemma with $g_t(x) = \text{DEM}^{\alpha}/x^{\alpha-1}$ and $g_d(x) = 1 - \text{FAIL} - \sqrt{\frac{K}{x}}$ to prove that $E^{(\text{up})}$ is convex and consequently admits a unique minimum on $[1, m]$. Moreover, $E^{(\text{up})}(x) \underset{x \to \infty}{\longrightarrow} +\infty$ and $E^{(\text{up})}(x) \underset{x \to 0}{\longrightarrow} +\infty$ so that we can uniquely define $x_{\min}^{(\text{up})}$ by $\left(E^{(\text{up})}\right)'(x_{\min}^{(\text{up})}) = 0$, *i.e.*

$$E_{\text{stat}} = \left(\frac{\text{DEM}}{(1 - \text{FAIL})x_{\min}^{(\text{up})} - \sqrt{K x_{\min}^{(\text{up})}}} \right)^{\alpha} \times \left((\alpha - 1)(1 - \text{FAIL}) + \left(1 - \frac{\alpha}{2}\right) \sqrt{\frac{K'}{x_{\min}^{(\text{up})}}} \right). \quad (2)$$

IV. Simulations

The application of Chernoff bounds enables to find valid solutions (satisfying the reliability constraints) and to obtain theoretical upper bounds, but Chernoff bounds are in general too pessimistic, especially in the case when the number of machines is small. Hence, we derive in this section a heuristic that returns a homogeneous allocation with lower energy than the one obtained in Section II-B.

A. Algorithms for MIN-ENERGY-NO-SHUTDOWN Problem

In this section, we concentrate of the dynamic energy part only, and we assume that the overall number of running PMs is fixed so that the leakage term has to be paid for all PMs.

*1) **lower.bound**:* In order to evaluate the performance of the heuristics, we rely on the lower bound proved in Section II-A. This is a lower bound on the energy consumption that is required in order to fulfill the reliability constraint.

*2) **theo.homo**:* This algorithm builds a valid solution following the *Homogeneous* policy. We have exhibited such a solution in Section II-B. In order to determine the frequency at which each PM should be run, we rely on Chernoff bounds to estimate the reliability of the allocation. Therefore, due to the application of conservative Chernoff bounds, this solution is in general pessimistic, in the sense that induced energy may not be optimal.

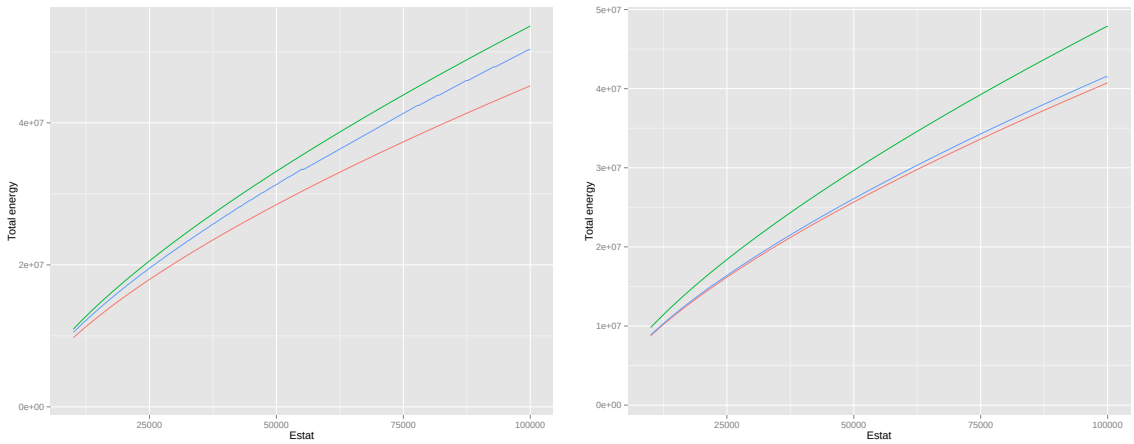
*3) **best.homo**:* In order to cope with the limitations of **theo.homo** algorithm, **best.homo** finds the best solution (*i.e.* the one that minimizes the energy consumption) following *Homogeneous* policy. To do this, we need to estimate precisely the reliability of an allocation, instead of relying on a lower bound as in **theo.homo**. **best.homo** can be decomposed into an off-line and an on-line phase; the former is executed once and for all, while the latter is to be run for each reliability constraint.

In the off-line phase, we rely a double-entry table, where a row is associated with a number of machines m and a column corresponds to a reliability requirement REL. The value of a cell indicates the maximum number m' such that the probability of having $m' \leq m$ alive machines among the m initial machines at the end of the day is not less than $1 - \text{REL}$. Those values can be obtained thanks to a cumulative binomial distribution.

In the on-line phase, we perform a binary search on the machine capacity, so that we end up with a valid solution minimizing the energy. Obviously, this solution is the one that minimizes the common clock frequency of the machines, and if the reliability constraint is fulfilled for a given capacity, it is *a fortiori* true for a higher frequencies. At each step, for a given frequency, we just have to check, using the table, whether the number of alive instances is large enough.

B. Algorithms for MIN-ENERGY problem

Let us now consider the case when both static (leakage) and dynamic energy have to be taken into account, and when both the number of PMs and their frequency have to determined. When adding a non-zero static energy, all heuristics and



(a) FAIL $= 10^{-1}$, REL $= 5 \times 10^{-7}$

(b) FAIL $= 10^{-3}$, REL $= 5 \times 10^{-7}$

(c) FAIL $= 10^{-2}$, REL $= 10^{-5}$

(d) FAIL $= 10^{-2}$, REL $= 10^{-9}$

(e) Legend

Figure 3. Simulation results for DEM $= 10^4$ and $\alpha = 3$

bounds are such that the overall dissipated energy tends to $+\infty$ if the number of machines tends to 0 (because of the dynamic energy) or to $+\infty$ (because of the static energy). There remains to find for each of them the optimal number of machines.

We have proved the convexity of the energy function returned by **lower.bound**. Thus, solving Equation 1 using binary searc is enough in order to obtain the optimal m. We operate in the same way for **theo.homo**, solving Equation 2 thanks to a binary search. Since the energy consumption of the best homogeneous allocation is also convex (as a function of the number of machines), we also rely on the same technique for **best.homo** on the MIN-ENERGY problem. More specifically, we perform a binary search in order to obtain the number of used machines that leads to minimum energy consumption.

C. *Results for* MIN-ENERGY *problem*

1) For a single configuration: In Figure 2, we compare the performance of all three heuristics under the following settings: FAIL $= 10^{-2}$, DEM $= 10^4$, REL $= 10^{-5}$, $\alpha = 3$, $E_{\text{stat}} = 5 \times 10^4$ and m varies between 1 and 250. **lower.bound** is depicted in red, **best.homo** in blue, **theo.homo** in green. As expected, the dynamic energy decreases with the number of machines and as we proved in Section II-C, the lower and the upper bound converge when the number of machines becomes large. When both leakage and dynamic energy terms are taken into account, then the plots obtained for **lower.bound**, **best.homo** and **theo.homo** are convex, as proved in Section III. Using binary search for each plot, we are able to determine, for each heuristic, the point that minimizes the overall energy (respectively the red point for **lower.bound**, the blue point for **best.homo** and the green point for **theo.homo**).

In this example, the energy consumed by **lower.bound** is 2.58×10^7, while **best.homo** consumes 2.67×10^7 and **theo.homo** 2.94×10^7, showing that **theo.homo** is 14% larger than the lower bound and that **best.homo** only 4% larger than the lower bound.

2) Simulation Results: In order to study the influence of the different parameters, we performed a large set of simulations, whose results are depicted on Figure 3. Each point in Figure 3 corresponds to the results of an experiment for a single configuration described in Section IV-C1. For instance, the results of the configuration depicted in previous section can be read on Figure 3(c) when $E_{\mathrm{stat}} = 5 \times 10^4$.

In general, we can observe that the simulation results prove the efficiency of homogeneous distributions with respect to energy minimization. Indeed, the red plots correspond to a lower bound that holds true for any (possibly heterogeneous) solutions. In all cases, the ratio between the upper bound **theo.homo** and the lower bound **lower.bound** is always smaller than 1.2 and that the ratio between the upper bound **best.homo** and the lower bound **lower.bound** is always smaller than 1.08.

Therefore, our simulations results prove both that **lower.bound** is always very close to the lower bound and that the approximation ratio provided by **theo.homo** is in general not too pessimistic.

V. Pricing Issues

In practice, it may be difficult for the cloud user to evaluate the reliability requirements for the service they are running. On the other hand, our work enables the cloud provider to price the reliability constraint since it is possible to estimate the overall price of the energy $\textsc{Price}(E(\textsc{Rel}))$ that is required to enforce reliability $\textsc{Rel}$ for a given service. From this information, it is possible for the Cloud provider to turn its offer into a list of pairs $(price, compensation)$, so that

$$(1-\textsc{Rel}) \times price - \textsc{Rel} \times compensation = \textsc{Price}(E(\textsc{Rel})).$$

In this case, the expectation of the price received by the provider is equal to its actual energy cost.

VI. Conclusion and Open Problems

In this paper, we have proposed approximation algorithms for minimizing both the number of used resources and the dissipated energy in the context of static service allocation under reliability constraints in Clouds. For both optimization problems, we have given lower bounds and we have exhibited algorithms that achieve claimed reliability. In the case of energy minimization, we have even been able to prove that proposed algorithm is asymptotically optimal when the overall demand or the number of machines becomes arbitrarily large. Such a result is important since it enables, for the point of view of the Cloud provider, to associate a price to reliability (or equivalently to fix penalties in case of SLA violation). This work opens many perspectives. First, relying on different techniques, better approximation ratio in the case of low number of resources are needed. Then, the extension to several services is trivial in the case of resource usage minimization, but not trivial in the case of energy minimization. It would also be interesting to explicitly take into account the memory print of the services, so as to limit the number of different services that a machine can handle. This would lead to different results, by enforcing to limit the number of participating physical machines to the deployment of each individual service. At last, we concentrate in this work on the static phase, and we assume that migrations and redistributions take place at regular time steps. It would be very interesting to mix both migrations and static allocations in order to minimize the overall required energy, since more frequent redistributions induce less energy consumed by replication but more energy wasted by migration phases.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "Above the clouds: A berkeley view of cloud computing," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*, 2009.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599 – 616, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X08001957

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1496091.1496103

[5] H. Van, F. Tran, and J. Menaud, "SLA-aware virtual resource management for cloud infrastructures," in *IEEE Ninth International Conference on Computer and Information Technology*. IEEE, 2009, pp. 357–362.

[6] R. Calheiros, R. Buyya, and C. De Rose, "A heuristic for mapping virtual machines and links in emulation testbeds," in *2009 International Conference on Parallel Processing*. IEEE, 2009, pp. 518–525.

[7] O. Beaumont, L. Eyraud-Dubois, H. Rejeb, and C. Thraves, "Heterogeneous Resource Allocation under Degree Constraints," *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[8] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *The Computer Journal*, vol. 53, no. 7, p. 1045, 2010.

[9] A. Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE, 2010, pp. 577–578.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[11] L. Epstein and R. van Stee, "Online bin packing with resource augmentation." *Discrete Optimization*, vol. 4, no. 3-4, pp. 322–333, 2007.

[12] D. Hochbaum, *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

[13] W. Cirne, "Scheduling at google," in *16th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), in conjunction with IPDPS 2012*, 2011.

[14] K. Ranganathan, A. Iamnitchi, and I. Foster, "Improving data availability through dynamic model-driven replication in large peer-to-peer communities," in *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, may 2002, p. 376.

[15] D. da Silva, W. Cirne, and F. Brasileiro, "Trading cycles for information: Using replication to schedule bag-of-tasks applications on computational grids," in *Euro-Par 2003 Parallel Processing*, ser. Lecture Notes in Computer Science, H. Kosch, L. Böszörményi, and H. Hellwagner, Eds. Springer Berlin / Heidelberg, 2003, vol. 2790, pp. 169–180.

[16] M. Lei, S. V. Vrbsky, and X. Hong, "An on-line replication strategy to increase availability in data grids," *Future Generation Computer Systems*, vol. 24, no. 2, pp. 85 – 98, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X07000830

[17] H.-I. Hsiao and D. J. Dewitt, "A performance study of three high availability data replication strategies," *Distributed and Parallel Databases*, vol. 1, pp. 53–79, 1993, 10.1007/BF01277520. [Online]. Available: http://dx.doi.org/10.1007/BF01277520

[18] E. Santos-Neto, W. Cirne, F. Brasileiro, and A. Lima, "Exploiting replication and data reuse to efficiently schedule data-intensive applications on grids," in *Job Scheduling Strategies for Parallel Processing*, ser. Lecture Notes in Computer Science, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Springer Berlin / Heidelberg, 2005, vol. 3277, pp. 54–103.

[19] J. Dongarra, P. Beckman, P. Aerts, F. Cappello, T. Lippert, S. Matsuoka, P. Messina, T. Moore, R. Stevens, A. Trefethen *et al.*, "The international exascale software project: a call to cooperative action by the global high-performance community," *International Journal of High Performance Computing Applications*, vol. 23, no. 4, pp. 309–322, 2009.

[20] "Eesi, "the european exascale software initiative", 2011," http://www.eesi-project.eu/pages/menu/homepage.php.

[21] F. Cappello, "Fault tolerance in petascale/exascale systems: Current knowledge, challenges and research opportunities," *International Journal of High Performance Computing Applications*, vol. 23, no. 3, pp. 212–226, 2009.

[22] K. Ferreira, J. Stearley, J. Laros III, R. Oldfield, K. Pedretti, R. Brightwell, R. Riesen, P. Bridges, and D. Arnold, "Evaluating the viability of process replication reliability for exascale systems," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011, p. 44.

[23] M. Bougeret, H. Casanova, M. Rabie, Y. Robert, and F. Vivien, "Checkpointing strategies for parallel jobs," in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*. IEEE, 2011, pp. 1–11.

[24] F. Cappello, H. Casanova, and Y. Robert, "Checkpointing vs. migration for post-petascale supercomputers," *ICPP'2010*, 2010.

[25] O. Beaumont, L. Eyraud-Dubois, and H. Larchevêque, "Reliable Service Allocation in Clouds," in *IPDPS 2013 - 27th IEEE International Parallel & Distributed Processing Symposium*, Boston, États-Unis, 2013. [Online]. Available: http://hal.inria.fr/hal-00743524

[26] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED)*. ACM Press, 1998, pp. 197–202.

[27] K. Pruhs, R. van Stee, and P. Uthaisombut, "Speed scaling of tasks with precedence constraints," *Theory of Computing Systems*, vol. 43, pp. 67–80, 2008.

[28] A. P. Chandrakasan and A. Sinha, "Jouletrack: A web based tool for software energy profiling," in *Design Automation Conference*. IEEE CS Press, 2001, pp. 220–225.

[29] H. Aydin and Q. Yang, "Energy-aware partitioning for multiprocessor real-time systems," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, 2003, pp. 113–121.

[30] J.-J. Chen and T.-W. Kuo, "Multiprocessor energy-efficient scheduling for real-time tasks," in *Proceedings of International Conference on Parallel Processing (ICPP)*. IEEE CS Press, 2005, pp. 13–20.

[31] X. Qi, D. Zhu, and H. Aydin, "Global reliability-aware power management for multiprocessor real-time systems," in *RTCSA*, 2010, pp. 183–192.

[32] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED)*. New York, NY, USA: ACM Press, 1998, pp. 197–202.

[33] P. Langen and B. Juurlink, "Leakage-aware multiprocessor scheduling," *Journal of Signal Processing Systems*, vol. 57, no. 1, pp. 73–88, 2009.

[34] R. Mishra, N. Rastogi, D. Zhu, D. Mossé, and R. Melhem, "Energy aware scheduling for distributed real-time systems," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, 2003, pp. 21–29.

[35] "Data centers waste vast amounts of energy belying industry image," http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html.

[36] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[37] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *The Annals of Mathematical Statistics*, vol. 23, no. 4, pp. 493–507, 1952.

[38] L. Valiant, "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, no. 2, pp. 189 – 201, 1979. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304397579900446

[39] A. Benoit, P. Renaud-Goud, and Y. Robert, "Power-aware replica placement and update strategies in tree networks," in *IPDPS*, 2011, pp. 2–13.