



**HAL**  
open science

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

Olivier Beaumont, Philippe Duchon, Paul Renaud-Goud

► **To cite this version:**

Olivier Beaumont, Philippe Duchon, Paul Renaud-Goud. Approximation Algorithms for Energy Minimization in Cloud Service Allocation under Reliability Constraints. [Research Report] RR-8241, 2013, pp.20. hal-00788964v1

**HAL Id: hal-00788964**

**<https://inria.hal.science/hal-00788964v1>**

Submitted on 18 Feb 2013 (v1), last revised 10 Oct 2013 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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

Olivier Beaumont, Philippe Duchon, Paul Renaud-Goud

**RESEARCH  
REPORT**

**N° 8241**

February 2013

Project-Teams CEPAGE





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

Olivier Beaumont\*, Philippe Duchon\*, Paul Renaud-Goud\*

Project-Teams CEPAGE

Research Report n° 8241 — February 2013 — 17 pages

**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 to a capacity constraint, that can be chosen using Dynamic Voltage and Frequency Scaling (DVFS) method, and to a probability of failure. On the other hand, we assume that the service runs as a set of independent instances of identical Virtual Machines. 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 which must be alive at the end of the day, and the Cloud provider offers a list of pairs (*price, compensation*), this compensation being paid by the Cloud provider if it fails to keep alive the required number of services. On the Cloud provider side, each pair corresponds actually to a guaranteed success probability 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 during a given time period while minimizing the energy consumption of used resources. We consider two energy consumption models based on DVFS techniques, where the clock frequency of physical resources can be changed. For each allocation problem and each energy model, we prove deterministic approximation ratios on the consumed energy for algorithms that provide guaranteed probability failures, as well as an efficient heuristic, whose energy ratio is not guaranteed.

**Key-words:** Cloud, reliability, approximation, energy savings

---

\* Inria, University of Bordeaux

RESEARCH CENTRE  
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour  
33405 Talence Cedex

## **Algorithmes d'approximation sur la minimisation d'énergie pour l'allocation de services dans un Cloud sous contraintes de fiabilités**

**Résumé :** Nous considérons un problème d'allocation de services dans des *Clouds*. Les ressources de calcul sont caractérisées par une probabilité de panne, et une contrainte de capacité, qui peut être ajustée grâce à la technique dite de *Dynamic Voltage and Frequency Scaling* (DVFS). Il existe un contrat entre le fournisseur et le client, le fournisseur assurant au client qu'un certain nombre d'instances du service du client sera toujours en train de s'exécuter à la fin de la journée, avec une certaine probabilité. La question est donc de savoir à quelle vitesse devront tourner les processeurs, et à quel point les services devront être répliqués sur les machines. Nous exhibons des algorithmes d'approximation, prouvons leurs facteurs d'approximation sur l'énergie consommée, et décrivons des heuristiques performantes.

**Mots-clés :** Cloud, fiabilité, approximation, économie d'énergie

# 1 Introduction

## 1.1 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 instances running as Virtual Machines (VMs) onto Physical Machines (PMs) in a Cloud computing platform. Cloud Computing [41, 1, 10, 24] 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 requirement, 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 [40, 11, 4] or to minimize energy consumption [7, 5]. Most of the works in this domain have therefore concentrated on designing offline [23] and online [21, 26] solutions of Bin Packing variants.

Reliability constraints have received much less attention in the context of Cloud computing, as underlined by Walfredo Cirne in [17]. 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 [36, 18, 32, 28, 37]. Reliability issues have also been addressed by 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 [12]. Solutions for fault tolerance in Exascale systems are based on replication strategies [22] and rollback recovery relying on checkpointing protocols [8, 13].

This work is a follow-up of [3], where the question of how to evaluate the reliability of an 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 we can rely on sophisticated mechanisms in order to fix the clock frequency of the PMs and we rely on techniques such as DVFS (see [29, 34, 14, 2, 15]). 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 [35]). (we will nevertheless not consider this case in this paper and we leave it 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 service running on the Cloud platform consists of 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 [11, 4]). 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 prob-

lems only, in the sense that our goal is to find the allocation that optimizes the reliability during a time period (say at the end of the day), instead of relying on VM migrations and creations to ensure that a minimal number of instances of each service is running whatever the machine failures. 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 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 both cases, the goal, from the provider point of view, is 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 cost of reliability. This cost of reliability can then be directly translated into a set of (price, fine in case of SLA violation) offers by the Cloud provider.

## 1.2 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  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_m$ . As already noted, we assume that Machine  $\mathcal{M}_j$  is able to handle the execution of  $\text{CAPA}_j$  instances of service. We also assume that we can rely on Dynamic Voltage Frequency Scaling (DVFS) mechanism in order to adapt  $\text{CAPA}_j$ . The energy consumed by machine  $\mathcal{M}_j$  when running at capacity (speed proportional to)  $\text{CAPA}_j$  is given by  $E = E_{\text{stat}}(j) + E_{\text{dyn}}(j)$ , where  $E_{\text{dyn}}(j) = e_j \text{CAPA}_j^\alpha$ , with  $\alpha \geq 2$ . 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. For the sake of simplicity, we will assume throughout this paper that any  $\text{CAPA}_j$  can be achieved by Machine  $\mathcal{M}_j$ , as advocated in [30, 31, 33].

On this Cloud platform, our goal is to run (all through a given time period, say a day, as defined in the SLA) a service  $\mathcal{S}$ . DEM identical and independent instances of service  $\mathcal{S}$  are required, and the instances run as Virtual Machines. Several instances can therefore run concurrently and independently on the same physical machine. We will denote by  $\mathcal{A}_j$  the number of instances running on  $\mathcal{M}_j$ , that has to be smaller than  $\text{CAPA}_j$ .  $\sum_j \mathcal{A}_j$  represents the overall number of running instances of  $\mathcal{S}$ . In general,  $\sum_j \mathcal{A}_j$  is larger than DEM 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  $\text{FAIL}_j$ , that represents the probability of failure of  $\mathcal{M}_j$  during the time period. During the time period, we will not reallocate instances of service to physical machines but rather provision extra instances for the service (replicas) that will actually be used if some machines fail. In practice,  $\text{FAIL}_j$  depends on the clock frequency (although no clear consensus exist in the literature on how, see for instance [9, 25, 38]) and therefore on  $\text{CAPA}_j$ . As said previously, we will assume for the results proved in this paper that  $\text{FAIL}_j$  does not depend on  $\text{CAPA}_j$ .

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 running on  $\mathcal{M}_j$  is  $\mathcal{A}_j$  if  $\mathcal{M}_j \in \text{ALIVE}$  and 0 otherwise. Therefore,  $\text{ALIVEINST} = \sum_{\mathcal{M}_j \in \text{ALIVE}} \mathcal{A}_j$  denotes the overall number of running instances at the end of the time period and the service is running properly at the end of the time period if and only if  $\sum_{\mathcal{M}_j \in \text{ALIVE}} \mathcal{A}_j \geq \text{DEM}$ .

Of course, our goal is not that all services 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 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 for the service (together with the penalty paid by the Cloud Provider if  $\mathcal{S}$  does not run with at least DEM instances at the end of the period). Therefore, the Cloud provider faces the following optimization problem:

**BestEnergy**( $m, \text{DEM}, \text{REL}$ ): Find the set ON of machines that are on and the clock frequency assigned to Machine  $\mathcal{M}_j$ , represented by  $\text{CAPA}_j$  and an allocation  $\mathcal{A}$  of instances to machines  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_m$  such that (i)  $\forall j \in \text{ON}, \mathcal{A}_j \leq \text{CAPA}_j$ , (ii)  $\mathcal{P}(\text{ALIVEINST} \geq \text{DEM}) \geq 1 - \text{REL}$ , i.e. the probability that a least DEM instances of  $\mathcal{S}$  are running on alive machines after the time period is larger than the reliability requirement  $1 - \text{REL}$ , (iii) and the overall energy consumption  $\sum_{j \in \text{ON}} E_{\text{stat}}(j) + e_j \text{CAPA}_j^\alpha$  is minimized.

### 1.3 Methodology

Throughout the paper, we will rely on the same general approach. Through Section 2 to Section 3 and Section 4, we complicate the problem by considering more general problem (from the problem of assigning a fractional number of instances onto a fixed number of resources, to the problem of allocating integer number of instances onto a set of resources to be determined).

In order to prove claimed approximation ratios, we will rely on the following techniques.

For the lower bounds, we prove that for a service, given the reliability constraints of the 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 [27].

For the upper bounds, we concentrate on two special allocation schemes, namely *Homogeneous* and *Step*. In a solution of *Homogeneous*, for each service, we assign to every machine the same number of instances (that may be fractional or integral depending on the context), i.e.  $\forall i, \forall j \in \text{ON}, \mathcal{A}_{i,j} = \mathcal{A}_i$ . In a *Step* solution, we authorize one unit step, i.e.  $\forall i, \forall j \in \text{ON}, 0 \leq \mathcal{A}_{i,j} - \mathcal{A}_i \leq 1$ . Using these two allocation schemes, we are able to derive theoretical bounds relying on Chernoff bounds [16]. 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.

### 1.4 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  $\text{REL} = 8 \cdot 10^{-6}$ , 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 various 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 day, in order to satisfy reliability constraints. The shape of the optimal solution reflects the complexity of the problem. Indeed, Indeed, it has been proved in [3] that even in the case with a single service and even if the allocation is given, then estimating its reliability is  $\#P'$ -complete. The  $\#P$  complexity class has been introduced by Valiant [39] 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, 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. By counting the number of such valid configurations (weighted by their probability) leads to the reliability of the allocation. We can notice that the optimal solution involves 60 instances against around 67 for best fractional homogeneous solution, and 64 in the best step solution. Nevertheless, we will use fractional homogeneous and step solutions in order both to derive approximation algorithms and upper bounds on the number of required resources, and we will see that they are in general close to the optimal.

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. On the other hand, if  $\alpha = 3$  for instance, then the best step solution consumes less energy than the solution minimizing the number of instances. We will prove in this paper that step and homogeneous fractional solutions are in fact asymptotically optimal when the overall demand, or the number of machines involved in the solution, becomes arbitrarily large.

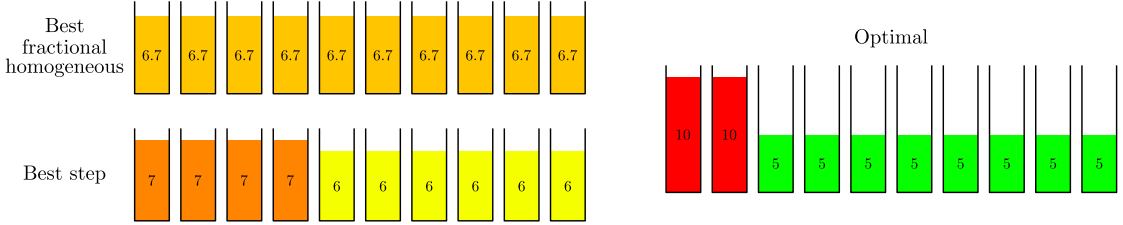


Figure 1: Motivating example

## 1.5 Outline of the Paper

As we have noticed through the motivating example, **BestEnergy** is in general difficult. 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 [6] the non-linearity of energy consumption makes the optimization problems harder. In our context, approximation ratio are smaller for energy minimization than for classical replica (that would correspond to makespan or load balancing in other contexts) minimization.

To prove this result, we progressively come to the most general problem through the study of more simple objective functions. Firstly, we address in Section 2 the case where we are given a single service, where the set of machines that are switched on is given and where the number of instances allocated to a machine is allowed to be fractional. Finding allocations such that the number of instances we place is minimum is denoted as MIN-ENERGY-NO-SHUTDOWN problem. Then, we address in Section 3 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. Finally, in Section 4, we tackle the problem of designing more realistic solutions, where the number of instances on each machine must be an integer. In *Homogeneous*, all participating machines are allocated the same number of instances whereas in *Step*, the number of instances allocated to a machine can differ by at most one (either  $a$  or  $a + 1$  for some value of  $a$ ).

## 2 Fractional MIN-ENERGY-NO-SHUTDOWN

### 2.1 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 day) 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.*  $\sum_j \mathcal{A}_j$ . 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  $s^\alpha$ . 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  $\text{ALIVEINST}_j$  is the number of instances of the service that are alive on machine  $\mathcal{M}_j$  at the end of the day.  $\text{ALIVEINST}_j$  is thus a random variable equal to  $\mathcal{A}_j$  with a probability  $1 - \text{FAIL}$  and to 0 with a probability  $\text{FAIL}$ .

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 [27]) 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  $t = \sqrt{-\ln(1 - \text{REL}) \sum_{j=1}^m \mathcal{A}_j^2 / 2}$ , so that  $\exp\left(-2 \frac{t^2}{\sum_{j=1}^m \mathcal{A}_j^2}\right) = 1 - \text{REL}$ . Noting  $K' = \frac{-\ln(1 - \text{REL})}{2}$ , 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 citehoelder, that states that if  $1/p + 1/q = 1$ , then  $\forall a_j, b_j \geq 0$ ,  $\sum a_j b_j \leq (\sum a_j^p)^{1/p}, (\sum b_j^q)^{1/q}$ . We assume in the following that  $\alpha > 2$ .

With  $p = q = 2$ ,  $a_j = b_j = \mathcal{A}_j$ , we obtain  $\sum \mathcal{A}_j^2 \leq (\sum \mathcal{A}_j)^2$ , so that  $(1 - \text{FAIL}) \sum_{j=1}^m \mathcal{A}_j + \sqrt{K' \times \sum_{j=1}^m \mathcal{A}_j^2} \leq (1 - \text{FAIL} + \sqrt{K'}) \times \sum_{j=1}^m \mathcal{A}_j$ . 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  $\text{MINREP}$  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  $\sum \mathcal{A}_j \leq (\sum \mathcal{A}_j^\alpha)^{1/\alpha} m^{1-1/\alpha}$ . Similarly, (remember that  $\alpha > 2$  so that  $\alpha/2 > 1$ ), with  $p = \alpha/2$ ,  $1/q = (1 - 2/\alpha)$ ,  $a_j = \mathcal{A}_j^2$  and

$b_j = 1$ , we obtain  $\sum \mathcal{A}_j^2 \leq (\sum \mathcal{A}_j^\alpha)^{2/\alpha} m^{1-2/\alpha}$ , so that

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

Also, we can derive another necessary condition defined as

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

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

Note that both results hold true for  $\alpha = 2$ .

## 2.2 Upper bound – Homogeneous

### 2.2.1 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.*  $\forall j, \mathcal{A}_{1,j} = \mathcal{A}$ ).

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

Hence, the expected value of  $|\text{ALIVE}|$  is given by  $\mathbb{E}(|\text{ALIVE}|) = (1 - \text{FAIL})m$ . Chernoff bound (see [16]) 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  $\{0, \dots, A\}$  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,  $\mathbb{P}(|\text{ALIVE}| \leq (1 - \text{FAIL} - \varepsilon)m) \leq e^{-2\varepsilon^2 m}$ . As we want to ensure that  $\mathbb{P}_{fail} \leq \text{REL}$ , we choose  $\varepsilon$  such that  $e^{-2\varepsilon^2 m} = \text{REL}$ , *i.e.*  $\varepsilon = \sqrt{K'/m}$  by noting  $K' = \frac{-\ln(\text{REL})}{2}$ . Finally, we obtain a sufficient condition, so that the reliability constraint is fulfilled for the service  $\mathcal{A}m \geq \frac{\text{DEM}}{1 - \text{FAIL} - \sqrt{\frac{K'}{m}}} = \text{MAXREP}$ .

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

## 2.3 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 the 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.

## 2.4 Can optimal solutions be strongly heterogeneous ?

Above results states 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  $\mathcal{A}_j$  (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 2.1) 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  $V' < \frac{\text{MINENERGY}}{m} - \left(\frac{\text{DEM}}{(1 - \text{FAIL})m - \sqrt{K'm}}\right)^\alpha$  or equivalently  $m^\alpha V' \leq \left(\frac{\text{DEM}}{1 - \text{FAIL} - \sqrt{\frac{K}{m}}}\right)^\alpha - \left(\frac{\text{DEM}}{1 - \text{FAIL} + \sqrt{K'}}\right)^\alpha$ .  $\square$

## 3 Fractional MIN-ENERGY

### 3.1 Lower bound

Let us now consider that the number of participating machines is to be determined. In this case, we need to take explicitly the leakage term into account (that was considered as constant in previous section since the number of switched on machine was fixed). In this case, given that  $k \in \{1, \dots, m\}$ , the goal is to minimize

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

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.

Let us apply above lemma with  $g_t(x) = x/x^{\alpha/2}$  (which is convex since  $\alpha > 2$ ) 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  $E^{(\text{low})} \xrightarrow{0} +\infty$  and  $E^{(\text{low})} \xrightarrow{\infty} +\infty$ ,  $(E^{(\text{low})})'$  is null at some point in  $[0, +\infty[$ , and let us define  $x_{\min}^{(\text{low})}$  such that  $(E^{(\text{low})})'(x_{\min}^{(\text{low})}) = 0$ , 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  $\min(\max(x_{\min}^{(\text{low})}, 1), m)$ .

We can also obtain a lower bound on the energy consumption if we restrict the search to integral number of machines. Due to the convexity of  $E^{(\text{low})}$ , the minimum is achieved either at  $\lceil \min(\max(x_{\min}^{(\text{low})}, 1), m) \rceil$  or  $\lfloor \min(\max(x_{\min}^{(\text{low})}, 1), m) \rfloor$ , so that

$$E \leq \min \left( E^{(\text{low})}(\lceil \min(\max(x_{\min}^{(\text{low})}, 1), m) \rceil), E^{(\text{low})}(\lfloor \min(\max(x_{\min}^{(\text{low})}, 1), m) \rfloor) \right).$$

### 3.2 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  $g_t(x) = d^\alpha/x^{\alpha-1}$  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,  $E^{(\text{up})}(x) \xrightarrow{x \rightarrow \infty} +\infty$  and  $E^{(\text{up})}(x) \xrightarrow{x \rightarrow 0} +\infty$  so that we can uniquely define  $x_{\min}^{(\text{up})}$  by  $(E^{(\text{up})})'(x_{\min}^{(\text{up})}) = 0$ , i.e.

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

Therefore, we end up with the following upper bound for the energy

$$E \geq \min \left( E^{(\text{low})}(\lceil \min(\max(x_{\min}^{(\text{low})}, 1), m) \rceil), E^{(\text{low})}(\lfloor \min(\max(x_{\min}^{(\text{low})}, 1), m) \rfloor) \right).$$

## 4 Algorithms for the Integral Case

In the service allocation problem in Clouds, demands represent a number of virtual machines that need to be allocated onto physical machines. Therefore, the number of instances allocated to each machine has to be an integer, and we need to adapt above results in order to obtain valid allocation schemes. Moreover,

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 of low number of machines. Hence, we derive in this section a few heuristics that return lower energy than those obtained in Section 2.2.

## 4.1 MIN-ENERGY-NO-SHUTDOWN

### 4.1.1 Algorithms

**lower.bound** In order to evaluate the performance of the heuristics, we rely on the lower bound proved in Section 2.1. Since this lower bound is valid even among fractional solutions, it is *a fortiori* valid for energy minimization for the integral problem.

**theo.homo** This algorithm builds a valid solution following the *Homogeneous* policy. We have exhibited such a solution in Section 2.2 on the fractional problem. In order to enforce the reliability constraint while turning this solution into an integral one, we have to round the number of instances assigned to each machine to the next integer, *i.e.*  $\mathcal{A} = \left\lceil \frac{\text{DEM}}{m(1-p-\sqrt{\frac{K}{m}})} \right\rceil$ , leading to an energy consumption of

$$E_{\text{dyn}} = m \times \left\lceil \frac{\text{DEM}}{m(1-p-\sqrt{\frac{K}{m}})} \right\rceil^{\alpha}.$$

**best.homo** This heuristic finds the best solution (*i.e.* the one that minimizes the energy consumption) following *Homogeneous* policy. It 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 write 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 solution that minimizes the common capacity of the machines, and if the reliability constraint is fulfilled for a given capacity, it is *a fortiori* true for a higher capacity. At each step, for a given capacity, we just have to check, using the table, whether the number of alive instances is large enough.

**best.step** This heuristic aims at relaxing the homogeneous constraint by finding the best solution on the following form: there exists CAPA such that the number of instances is either CAPA or CAPA  $- 1$ . To achieve this goal, it first calls the previous **best.homo** heuristic that returns CAPA. This ensures that an allocation of CAPA instances per machine leads to a valid solution, whereas if we allocate CAPA  $- 1$  instances to each machine, the reliability constraint is violated. Then, we perform another binary search on the number of machines that will hold CAPA  $- 1$  instances, instead of CAPA. The validity of a given allocation is checked thanks to the dynamic programming algorithm described in [3].

### 4.1.2 Results

In Figure 2, we compare the performance of all heuristics under the following settings: FAIL =  $10^{-2}$ , DEM = 500, REL =  $10^{-6}$ ,  $\alpha = 2$  and  $m$  varies between 1 and 600. **lower.bound** is depicted in red, **best.step** in pink, **best.homo** in blue, **theo.homo** in green and **best.step** in red. We can clearly see the

aliasing issue of *Homogeneous* solutions on integral problem: during some periods, it only increases the number of loaded machines without decreasing the overall capacity.

*Step* solutions almost solve completely this issue and softens the **best.homo** curve, still staying always above. The ratio between the energy dissipated by **best.homo** and **lower.bound** is under 1.5 as soon as  $m > 25$ .

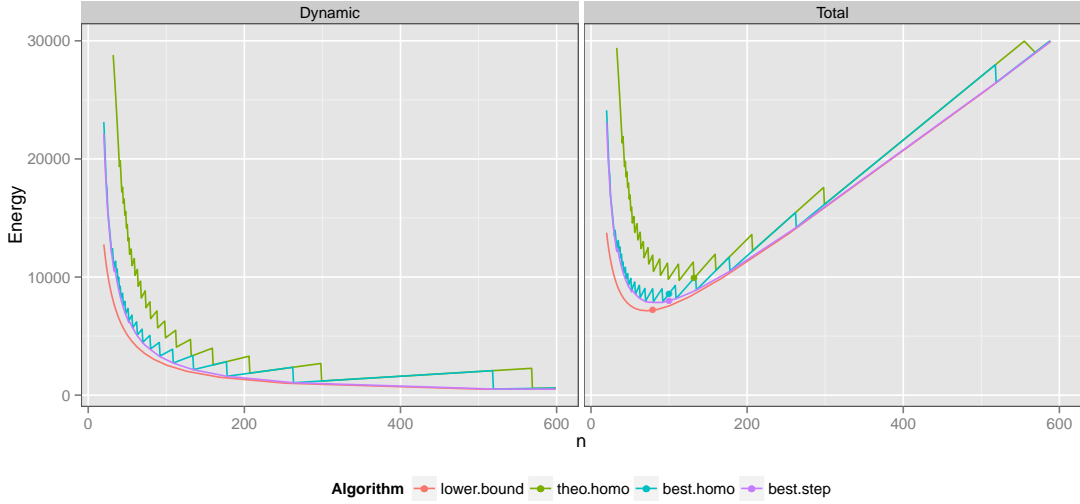


Figure 2: Simulation results for  $\text{FAIL} = 10^{-2}$ ,  $\text{DEM} = 500$ ,  $\text{REL} = 10^{-6}$ ,  $\alpha = 2$ ,  $1 \leq m \leq 600$ ,  $E_{\text{stat}} = 50$ .

## 4.2 MIN-ENERGY

When adding a non-zero static energy, all heuristics and 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 a close to the optimal number of machines for each algorithm.

We have proved the convexity of the energy function returned by **lower.bound**. Thus, solving Equation 1 using binary search, is enough in order to obtain the optimal  $m$ . When turning from fractional *Homogeneous* solutions to integral ones, convexity is lost and there is no easy way to find the optimal  $m$ . Therefore, we try all possible number of machines and keep the one that minimizes the consumed energy.

Concerning **best.homo** and **best.step**, trying all possible number of machines would be too expensive, since computing the consumed energy for a given  $m$  is in general  $\#P'$ -complete. As the dynamic energy returned by **best.homo** or **best.step** lies between the dynamic energy given by the lower and upper bounds of the fractional problem, the number of machines for **best.homo** and **best.step** lies between the solutions of Equation 1 and Equation 2. Thus, we choose  $m$  for **best.homo** and **best.step** the mean of previous solutions. The results for  $E_{\text{stat}} = 50$  are depicted in Figure 2.

## 5 Conclusion and Open Problems

In this problem, we have considered approximation algorithms for minimizing both the number of used resources and the dissipated energy in the context of service allocation under reliability constraints on Clouds. For both optimization problems, we have given lower bounds and 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 number of machines becomes arbitrarily large. Such a result is important since it enables, for the Cloud provider point of view, to associate a price to reliability (or to fix penalties in case of SLA violation). This work opens many perspectives. First, it seems possible to improve, relying on different techniques, better approximation ratio in the case of low number of resources. Then, the extension to several services is easy: all results can be generalized except the lower bound on the energy consumption. Still we can use the lower bound, obtained for resource minimization and extend it to the energy minimization. At last, it would be interesting to take explicitly 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 solution shapes, by enforcing to limit the number of participating physical machines in the deployment of each individual service.



## References

- [1] 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.
- [2] H. Aydin and Q. Yang. Energy-aware partitioning for multiprocessor real-time systems. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, pages 113–121, 2003.
- [3] 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.
- [4] O. Beaumont, L. Eyraud-Dubois, H. Rejeb, and C. Thraves. Heterogeneous Resource Allocation under Degree Constraints. *IEEE Transactions on Parallel and Distributed Systems*, 2012.
- [5] 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*, pages 577–578. IEEE, 2010.
- [6] A. Benoit, P. Renaud-Goud, and Y. Robert. Power-aware replica placement and update strategies in tree networks. In *IPDPS*, pages 2–13, 2011.
- [7] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Dang, and K. Pentikousis. Energy-efficient cloud computing. *The Computer Journal*, 53(7):1045, 2010.
- [8] 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*, pages 1–11. IEEE, 2011.
- [9] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger. Comparison of error rates in combinational and sequential logic. *Nuclear Science, IEEE Transactions on*, 44(6):2209–2216, dec 1997.
- [10] 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*, 25(6):599 – 616, 2009.
- [11] 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*, pages 518–525. IEEE, 2009.
- [12] F. Cappello. Fault tolerance in petascale/exascale systems: Current knowledge, challenges and research opportunities. *International Journal of High Performance Computing Applications*, 23(3):212–226, 2009.
- [13] F. Cappello, H. Casanova, and Y. Robert. Checkpointing vs. migration for post-petascale supercomputers. *ICPP'2010*, 2010.
- [14] A. P. Chandrakasan and A. Sinha. Jouletrack: A web based tool for software energy profiling. In *Design Automation Conference*, pages 220–225. IEEE CS Press, 2001.

- [15] J.-J. Chen and T.-W. Kuo. Multiprocessor energy-efficient scheduling for real-time tasks. In *Proceedings of International Conference on Parallel Processing (ICPP)*, pages 13–20. IEEE CS Press, 2005.
- [16] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.
- [17] W. Cirne. Scheduling at google. In *16th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), in conjunction with IPDPS 2012*, 2011.
- [18] D. da Silva, W. Cirne, and F. Brasileiro. Trading cycles for information: Using replication to schedule bag-of-tasks applications on computational grids. In H. Kosch, L. Böszörményi, and H. Hellwagner, editors, *Euro-Par 2003 Parallel Processing*, volume 2790 of *Lecture Notes in Computer Science*, pages 169–180. Springer Berlin / Heidelberg, 2003.
- [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*, 23(4):309–322, 2009.
- [20] Eesi, "the european exascale software initiative", 2011. <http://www.eesi-project.eu/pages/menu/homepage.php>.
- [21] L. Epstein and R. van Stee. Online bin packing with resource augmentation. *Discrete Optimization*, 4(3-4):322–333, 2007.
- [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*, page 44. ACM, 2011.
- [23] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [24] 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.*, 39(1):68–73, Dec. 2008.
- [25] K. J. Hass, J. W. Gambles, B. Walker, and M. Zampaglione. Mitigating single event upsets from combinational logic. In *7th NASA Symposium on VLSI Design*, volume 4, pages 1–4, 1998.
- [26] D. Hochbaum. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.
- [27] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [28] H.-I. Hsiao and D. J. Dewitt. A performance study of three high availability data replication strategies. *Distributed and Parallel Databases*, 1:53–79, 1993. 10.1007/BF01277520.
- [29] 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)*, pages 197–202. ACM Press, 1998.
- [30] 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)*, pages 197–202, New York, NY, USA, 1998. ACM Press.

- [31] P. Langen and B. Juurlink. Leakage-aware multiprocessor scheduling. *Journal of Signal Processing Systems*, 57(1):73–88, 2009.
- [32] M. Lei, S. V. Vrbsky, and X. Hong. An on-line replication strategy to increase availability in data grids. *Future Generation Computer Systems*, 24(2):85 – 98, 2008.
- [33] 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)*, pages 21–29, 2003.
- [34] K. Pruhs, R. van Stee, and P. Uthaisombut. Speed scaling of tasks with precedence constraints. *Theory of Computing Systems*, 43:67–80, 2008.
- [35] X. Qi, D. Zhu, and H. Aydin. Global reliability-aware power management for multiprocessor real-time systems. In *RTCSA*, pages 183–192, 2010.
- [36] 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*, page 376, may 2002.
- [37] E. Santos-Neto, W. Cirne, F. Brasileiro, and A. Lima. Exploiting replication and data reuse to efficiently schedule data-intensive applications on grids. In D. Feitelson, L. Rudolph, and U. Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, volume 3277 of *Lecture Notes in Computer Science*, pages 54–103. Springer Berlin / Heidelberg, 2005.
- [38] N. Seifert, D. Moyer, N. Leland, and R. Hokinson. Historical trend in alpha-particle induced soft error rates of the alpha<sup>tm</sup> microprocessor. In *Reliability Physics Symposium, 2001. Proceedings. 39th Annual. 2001 IEEE International*, pages 259–265. IEEE, 2001.
- [39] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.
- [40] 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*, pages 357–362. IEEE, 2009.
- [41] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Reliability and Energy Savings in Cloud Computing . . . . .	3
1.2	Notations . . . . .	4
1.3	Methodology . . . . .	5
1.4	Motivating example . . . . .	5
1.5	Outline of the Paper . . . . .	6
<b>2</b>	<b>Fractional MIN-ENERGY-NO-SHUTDOWN</b>	<b>7</b>
2.1	Lower bound . . . . .	7
2.2	Upper bound – <i>Homogeneous</i> . . . . .	8
2.2.1	MIN-REPLICATION . . . . .	8
2.3	Comparison . . . . .	8
2.4	Can optimal solutions be strongly heterogeneous ? . . . . .	9
<b>3</b>	<b>Fractional MIN-ENERGY</b>	<b>9</b>
3.1	Lower bound . . . . .	9
3.2	Upper bound – <i>Homogeneous</i> . . . . .	10
<b>4</b>	<b>Algorithms for the Integral Case</b>	<b>10</b>
4.1	MIN-ENERGY-NO-SHUTDOWN . . . . .	11
4.1.1	Algorithms . . . . .	11
4.1.2	Results . . . . .	11
4.2	MIN-ENERGY . . . . .	12
<b>5</b>	<b>Conclusion and Open Problems</b>	<b>13</b>



**RESEARCH CENTRE  
BORDEAUX – SUD-OUEST**

200 avenue de la Vieille Tour  
33405 Talence Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399