

# Optimisation multi-critère pour l'allocation de ressources sur clouds distribués avec prise en compte de l'énergie

Yacine Kessaci, Nouredine Melab, El-Ghazali Talbi

## Abstract

Reducing energy consumption is an increasingly important issue in cloud computing, more specifically when dealing with High Performance Computing (HPC). Minimizing energy consumption can significantly reduce the amount of energy bills and then increases the provider's profit. In addition, the reduction of energy decreases greenhouse gas emissions. Therefore, many researches are carried out to develop new methods in order to consume less energy. In this paper, we present a multi-objective genetic algorithm (MO-GA) that optimizes the energy consumption,  $CO_2$  emissions and the generated profit of a geographically distributed cloud computing infrastructure. We also propose a heuristic that aims to maximize the number of scheduled applications in order to compare it with the MO-GA. The two approaches have been experimented using realistic workload traces from Feitelson's Parallel Workload Archive (PWA). The results show that MO-GA outperforms the heuristic by a significant margin in terms of energy consumption and  $CO_2$  emissions. In addition, MO-GA is also proved to be slightly better in terms of profit while scheduling more applications. We also propose in the perspectives how to integrate our approach in the project StratusLab for the exploitation of the geographical dispersion offered by EGI.

## I. INTRODUCTION

Le cloud computing est de nos jours de plus en plus adopté dans de nombreux domaines tel que le calcul haute performance (HPC). Cependant, il subsiste le problème de la consommation croissante d'énergie liée aux ordinateurs. Par conséquent, offrir une des solutions plus efficaces est crucial pour les grands systèmes gourmands en consommation d'énergie. Une étude récente [1] montre qu'en 2005, la puissance utilisée par les serveurs représente environ 0,6 % de la consommation totale d'électricité des États-Unis. Cette proportion passe à 1,2 % lorsque les infrastructures de refroidissement et les appareils auxiliaires sont inclus. De plus, l'électricité totale consommée par les serveurs a doublé au cours de la période 2000 à 2005 dans le monde entier.

D'autre part, les émissions de gaz à effet de serre sont en train d'atteindre un seuil critique [2] et représente aujourd'hui 2 % de la quantité totale des émissions de  $CO_2$  à travers le monde. La consommation d'énergie a un autre inconvénient. En effet, cela affecte de manière directe le bénéfice des fournisseurs. Selon l'estimation d'Amazone [3], le montant des coûts liés à l'énergie représente 42% du budget total d'un datacenter. Ainsi, plusieurs ouvrages ont été proposés dans le domaine de *l'energy-aware computing*. Dans [4], [5] par exemple, une technique matérielle (DVFS) est proposée. Elle consiste à faire varier la fréquence du CPU afin de minimiser la consommation d'énergie. Une autre approche est proposée dans [6]. Ce travail utilise les possibilités offertes par la virtualisation afin d'appliquer une consolidation de tâches en vue de maximiser l'utilisation des ressources pour économiser l'énergie. Hormis les travaux proposés dans [7] qui traitent de la réduction de la consommation d'énergie à grande échelle comme dans le cas de la grille de calcul Grid'5000, tous les travaux précédents visent à réduire la consommation énergétique d'un seul datacenter ou de plusieurs serveurs de la même zone géographique. Le travail proposé par Garg *et al* [8] traite pour sa part trois objectifs (énergie,  $CO_2$ , profit). Cependant, cette approche est une agrégation d'objectifs (c'est-à-dire qu'elle ne peut optimiser qu'un seul objectif à la fois). Or, il apparaît clairement que tous les objectifs cités précédemment sont liés et doivent donc

être traités simultanément. Ainsi, dans cet article, nous présentons une nouvelle approche qui vise à faire face à l'énergie, aux émissions de gaz à effet de serre ainsi qu'au profit en même temps (au sens Pareto). En effet, nous proposons un méta-ordonnanceur qui utilise un algorithme génétique multi-objectifs (MO-GA) pour trouver le meilleur ordonnancement en fonction de ces trois objectifs. Cette approche utilise la répartition géographique des datacenters pour trouver le meilleur ordonnancement, puisque l'énergie, les taux de  $CO_2$  et le prix de l'électricité diffèrent d'une zone à l'autre. Notre approche vise également à donner la meilleure qualité de service (QoS) en respectant au maximum les délais de terminaison de toutes les applications des clients.

## II. MODÈLE DU CLOUD DISTRIBUÉ

### A. Modèle du système

Notre approche est basée sur le modèle de cloud *Infrastructure as a Service* (IAAS). En effet, nous avons affaire dans notre cas à une architecture deux tiers avec d'un côté le fournisseur du cloud distribué et de l'autre les clients. Ces derniers ont accès au cloud en demandant des ressources au fournisseur. Le service proposé par le fournisseur de cloud dans notre approche consiste en des infrastructures afin de permettre aux clients d'exécuter leurs applications HPC. Le rôle de ce travail est d'aider le fournisseur à optimiser un certain nombre de critères tout en proposant son service. De plus, notre cloud est géographiquement distribué. L'originalité de cette approche est de proposer un algorithme de méta-ordonnancement qui utilise un algorithme génétique multi-objectifs, ceci dans le but de trouver le meilleur ordonnancement pour les applications à travers le temps. Les objectifs traités dans notre algorithme sont la réduction de la consommation d'énergie et des émissions de gaz à effet de serre tout en maximisant le profit. De plus, l'algorithme de méta-ordonnancement veille à garantir la meilleure qualité de service (QoS), tout en respectant les délais de terminaison des applications et les contraintes du modèle. L'optimisation des objectifs est due principalement aux caractéristiques offertes par la répartition géographique des datacenters.

### B. Modèle énergétique

La consommation énergétique d'un datacenter résulte des équipements informatiques et des équipements auxiliaires. Dans notre travail, nous prenons en compte la consommation énergétique dû au refroidissement (équipements auxiliaires) et de la consommation énergétique dû au calcul (équipements informatiques). Notre modèle d'énergie est dérivé de celui des processeurs CMOS (Complementary Metal-Oxide Semiconductor) dont la formule est  $P = \alpha f^3 + \beta$ . Quant à l'énergie utilisée pour le refroidissement des datacenters, celle-ci est déduite grâce à un coefficient noté  $COP$  qui représente le ratio entre l'énergie consacrée au calcul et l'énergie utilisée pour le refroidissement du système. Cette dernière est étroitement liée à la zone géographique où se trouve le datacenter. En effet chaque région diffère de l'autre à cause de la température qui y règne.

### C. Description de la problématique

Notre problème met en relation deux tierces parties. La première partie est le fournisseur de cloud. Ce dernier possède  $N$  datacenters géographiquement répartis sur différentes zones dans le monde. La seconde partie est représentée par les clients avec  $J$  applications HPC qui doivent être exécutées sur les datacenters. Le problème consiste donc à ordonnancer  $J$  applications sur  $N$  datacenters. Dans notre description, le fournisseur est tenu de payer une somme d'argent pour pouvoir fournir ses services aux clients. Ce prix est le résultat de la consommation d'électricité de chacun de ses datacenters durant le calcul. En fonction de cette dépense, le fournisseur fixe un prix de vente du service pour les clients. La quantité de  $CO_2$  de chaque datacenter  $i$  est calculée à partir d'un ratio noté  $r_i^{CO_2}$ . Ce ratio est une valeur moyenne qui varie selon la façon dont l'électricité du datacenter a été produite. Lors du processus d'ordonnancement, l'utilisateur soumet une requête pour l'exécution d'une application HPC  $j$ . Cette requête est définie par

un triplet noté  $(e_j, n_j, d_j)$ . Toutes les informations relatives à ce triplet sont fournies par l'utilisateur lors de la réservation, à l'exception du temps de démarrage de l'application ( $t_j$ ) qui est déduit par le système à partir du temps de soumission. Les éléments du triplet représentent respectivement, le temps d'exécution de l'application, le nombre de processeurs requis par l'utilisateur pour son application et enfin le délai après lequel l'application sera considérée comme ayant échoué.

Les fonctions objectives de notre approche ont pour but de **minimiser** la consommation d'énergie et les émissions de gaz carbonique tout en **maximisant** le profit du fournisseur du cloud distribué. Ces opérations sont réalisées uniquement sous le respect des contraintes suivantes :

- L'application  $j$  doit finir avant  $d_j$  sinon l'ordonnancement est rejeté.
- Chaque application  $j$  peut être affectée sur un et un seul datacenter  $j$ .

### III. MÉTA-ORDONNANCEUR GÉNÉTIQUE MULTI-OBJECTIF

#### A. Étapes de l'algorithme de méta-ordonnancement

Avant chaque ordonnancement, le méta-ordonnancement attend durant une période fixe appelée *cycle d'ordonnancement*. Cette période permet de réunir un ensemble d'applications afin d'avoir un choix plus large et donc optimiser les résultats d'ordonnancement à venir. Une fois cette phase terminée, le groupe d'application est traité par le MO-GA pour trouver les meilleurs ordonnancements possibles sur les datacenters du cloud distribué. Le résultat de ce calcul est stocké dans une archive Pareto. Une fois l'ensemble des solutions Pareto (ordonnements) défini, l'algorithme choisit un ordonnancement selon le choix du fournisseur. La solution choisie à partir du front Pareto est utilisée comme un état de transition pour les différents datacenters. Cet état sera une base à partir de laquelle la prochaine itération de l'algorithme fera une autre exécution sur un nouvel ensemble d'applications. L'algorithme poursuivra ses itérations en proposant à chaque fois des ordonnancements pour chaque ensemble d'applications jusqu'à ce qu'il n'y ait plus d'applications qui arrivent.

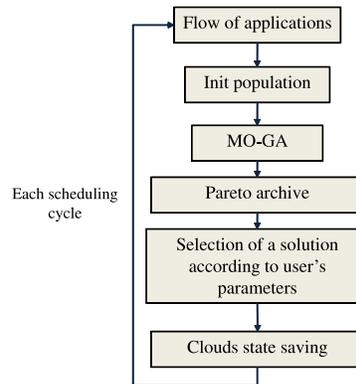


Fig. 1. The Flowchart of the Meta-scheduler Algorithm / Organigramme de l'algorithme de méta-ordonnancement

### IV. EXPÉRIMENTATIONS ET RÉSULTATS

#### A. Paramètres d'expérimentation

Notre approche étant portée sur les applications HPC, nous avons effectué les expérimentations sur des traces de charge de travail réalistes tirées du Feitelson's Parallel Workload Archive (PWA) [9]. La charge de travail des traces s'étend sur une période de cinq mois de requêtes d'applications (Janvier 2007 à Juin 2007). Les traces que nous avons utilisées sont les Lawrence Livermore National Laboratory (LLNL) du cluster Thunder. Les informations que nous avons extraites des traces LLNL sont : les temps de soumission des applications, leur temps d'exécution et le nombre de processeurs requis. Les traces étaient dépourvues d'information sur les délais de terminaison des applications. Nous avons donc

généralisés ces derniers suivant une distribution bimodale. 80% des applications appartiennent au groupe LU (Urgence faible) et 20% au groupe HU (Urgence élevée). Une application (HU) a 3 fois moins de temps en moyenne pour terminer son exécution qu'une application (LU). Dans notre approche, nous avons utilisé huit datacenters géographiquement distribués avec les mêmes spécifications que dans [8]. Le prix de l'électricité et le taux d'émission de gaz à effet de serre sont issus respectivement du rapport de l'US Energy Information Administration (EIA) [10] et du US Department of Energy (DOE) [11].

### B. Paramètres de l'algorithme de méta-ordonnancement

Dans nos expérimentations, nous avons utilisé différents paramètres (Variation du taux d'arrivée des applications, prix du service client, cycle d'ordonnancement). Nous avons utilisé quatre taux d'arrivages (bas, moyen, élevé et très élevé). Chaque passage d'un taux d'arrivée à l'autre représente 10 fois plus d'applications à ordonnancer durant la même période de temps. Le prix du client est fixé comme étant deux fois le coût moyen de l'énergie électrique d'un datacenter (\$0.40/CPU/h). Le cycle d'ordonnancement quant à lui a été fixé à (50 secondes).

### C. Évaluation des performances

TABLE I  
RÉSULTATS DU MÉTA-ORDONNANCEUR MO-GA FACE À L'HEURISTIQUE

Valeur objectif Taux d'arrivée	MO-GA				Heuristique			
	Énergie (kW h)	CO <sub>2</sub> (Kg)	Profit (\$)	Applications échouées	Énergie (kW h)	CO <sub>2</sub> (Kg)	Profit (\$)	Applications échouées
Bas	1.807e+06	709050	4.726e+06	<b>2270</b>	1.807e+06	709039	4.726e+06	2270
Moyen	1.813e+06	717417	4.700e+06	<b>2588</b>	1.833e+06	727748	4.696e+06	2635
Élevé	1.931e+06	837531	4.683e+06	<b>3045</b>	2.026e+06	939550	4.677e+06	3357
Très élevé	2.036e+06	972912	4.660e+06	<b>5168</b>	2.036e+06	972912	4.660e+06	5168

A notre connaissance aucune approche existante ne traite le problème d'ordonnancement multi-objectif au sens Pareto sur un cloud géographiquement distribué. Dans notre approche, en plus d'optimiser les trois objectifs cités précédemment, l'approche doit d'abord satisfaire un nombre maximum de clients (applications). Une comparaison entre notre approche et une heuristique qui maximise le nombre d'application traités nous semble donc être le meilleur choix pour évaluer notre travail. Les résultats sont présentés dans Table I. Les expériences montrent que notre approche améliore les résultats obtenus par l'heuristique en ce qui concerne les objectifs tout en réussissant à ordonnancer plus d'applications. Pour résumer, dans notre approche, la consommation d'énergie est réduite jusqu'à **4,66%**, les émissions de CO<sub>2</sub> jusqu'à **10,85%** et le profit maximisé jusqu'à **1,62%** par rapport à l'heuristique, tandis que le taux d'ordonnancement est en moyenne amélioré de **2,67%**.

## V. PERSPECTIVES

Dans ce papier, nous avons présenté un nouvel algorithme de méta-ordonnancement animé par un algorithme génétique multi-objectif. Les objectifs sont de minimiser la consommation d'énergie et les émissions de gaz à effet de serre ainsi que de maximiser le profit du fournisseur, tout en respectant les délais de terminaison des applications. Dans notre approche, l'économie d'énergie se fait par le biais de la répartition géographique des datacenters qui composent le cloud distribué. Le travail que nous avons proposé est le premier à avoir traité le problème au sens Pareto. Notre nouvelle approche a été évaluée avec des traces de charges de travail réalistes à partir du Feitelson's Parallel Workload Archive (PWA) [9]. Les expérimentations montrent que l'algorithme génétique multi-objectif améliore les résultats obtenus par

l'heuristique, en particulier dans la réduction des émissions de  $CO_2$ . En outre, notre approche ordonnance en moyenne **2,67 %** d'applications en plus que l'heuristique.

Concernant les perspectives de production à des fins de résultats pratiques, nous projetons de collaborer avec le projet StratusLab. Ainsi, le méta-ordonnanceur génétique multi-objectif que nous proposons prendra place comme une partie des modules qui composent StratusLab (voir Figure 2). En effet notre approche viendra se greffer au sein de la partie VM manager (OpenNebula) de StratusLab, plus particulièrement dans sa partie ordonnancement pour offrir un ordonnanceur plus intelligent. Cet ordonnancement va permettre d'exploiter de manière optimale le cloud géographiquement distribué offert par StratusLab à travers l'architecture matérielle qu'est la grille EGI. Ainsi, on donnera la possibilité de profiter de la dispersion géographique qu'offre cette dernière au niveau européen à des fins économiques, énergétiques et/ou environnementales.

Pour ce qui est de l'amélioration de notre approche, l'une des principales perspectives du travail présenté dans cet article est de déterminer d'une part un moyen de minimiser plus la consommation d'énergie en utilisant le Dynamic Voltage Scaling (DVS) au sein des datacenters, et d'autre part, un algorithme de méta-ordonnancement dynamique qui aura pour but une réaffectation des applications en temps réel sur différents datacenters pour l'optimisation de l'énergie et / ou du profit. Toutefois, cela dépendra de la flexibilité, du coût de transfert de données et de la complexité des applications, puisque le traitement porte sur des applications HPC.

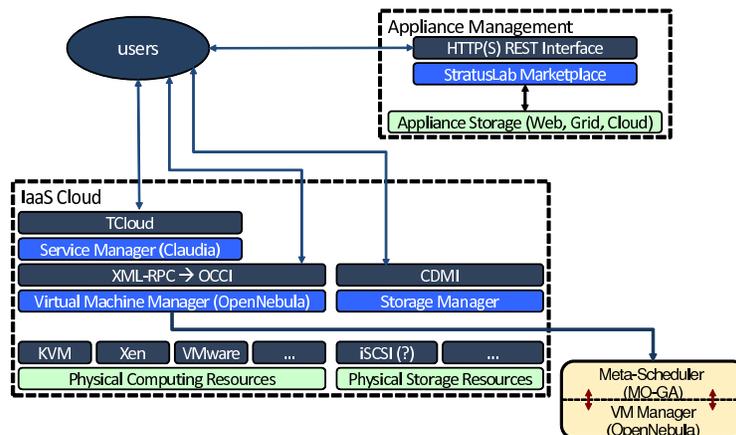


Fig. 2. Integration of the MO-GA Meta-Scheduler in StratusLab / Integration du méta-ordonnanceur MO-GA dans StratusLab

## REFERENCES

- [1] J. G. Koomey, "Estimating total power consumption by servers in the u.s. and the world."
- [2] Gartner. Gartner estimates ict industry accounts for 2 percent of global co2 emissions. <http://www.gartner.com/it/page.jsp?id=503867>.
- [3] J. Hamilton, "Cooperative expendable micro-slice servers (cems): Low cost, low power servers for internet-scale services," in *Proceedings of 4th Biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California, USA, January, 2009.
- [4] Y. C. Lee and A. Y. Zomaya, "Minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling," in *CCGRID'09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009, pp. 92–99.
- [5] N. B. Rizvandi, J. Taheri, A. Y. Zomaya, and Y. C. Lee, "Linear combinations of dvfs-enabled processor frequencies to modify the energy-aware scheduling algorithms," *Cluster Computing and the Grid, IEEE International Symposium on*, vol. 0, pp. 388–397, 2010.
- [6] Y. Lee and A. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, pp. 1–13, 2010, 10.1007/s11227-010-0421-3.
- [7] A.-C. Orgerie, L. Lefevre, and J.-P. Gelas, "Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems," in *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, 2008, pp. 171–178.
- [8] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers," *Journal of Parallel and Distributed Computing*, vol. In Press, Corrected Proof, pp. –, 2010.
- [9] D. Feitelson. (2009, Aug) Parallel workloads archive. <http://www.cs.huji.ac.il/labs/parallel/workload>.
- [10] (2007) Us energy information administration (eia) report. [http://www.eia.doe.gov/cneaf/electricity/epm/table5\\_6\\_a.html](http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html).
- [11] (2007) Us department of energy, voluntary reporting of greenhouse gases: Appendix f. [http://www.eia.doe.gov/oiaf/1605/pdf/Appendix20F\\_r071023.pdf](http://www.eia.doe.gov/oiaf/1605/pdf/Appendix20F_r071023.pdf).