Étude des Stratégies de réplication de données prenant en compte la consommation énergétique vs. le profit économique dans les systèmes Cloud

Morgan Séguéla, Riad Mokadem, Jean-Marc Pierson

▶ **To cite this version:**

Morgan Séguéla, Riad Mokadem, Jean-Marc Pierson. Étude des Stratégies de réplication de données prenant en compte la consommation énergétique vs. le profit économique dans les systèmes Cloud. Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS 2019), Jun 2019, Anglet, France. pp.1-8. hal-02887496

HAL Id: hal-02887496
https://hal.science/hal-02887496

Submitted on 2 Jul 2020

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.

Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
http://oatao.univ-toulouse.fr/26215

To cite this version: Seguela, Morgan and Mokadem, Riad and Pierson, Jean-Marc *Étude des Stratégies de réplication de données prenant en compte la consommation énergétique vs. le profit économique dans les systèmes Cloud.* (2019) In: Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS 2019), 25 June 2019 - 28 June 2019 (Anglet, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Étude des stratégies de réplication de données prenant en compte la consommation énergétique vs. le profit économique dans les systèmes Cloud

Morgan Séguéla, Riad Mokadem, Jean-Marc Pierson

Université Toulouse 3 Paul Sabatier,
Institut de Recherche en Informatique de Toulouse - 118 Route de Narbonne
31062 TOULOUSE CEDEX 9
prénom.nom@irit.fr

Résumé
Du fait de l'importance de la quantité de données produites, il devient de plus en plus difficile de les manipuler et d'y accéder rapidement. Dans ce contexte, le Cloud offre des ressources de traitement en apparence illimitées. La réplication de données est une technique bien connue qui permet de répondre à de tels problèmes. La plupart des stratégies de réplication de données ne répondent pas simultanément à des objectifs de performance, de profit du fournisseur, et de prise en compte de la consommation énergétique. Dans ce papier nous allons donc nous atteler à la comparaison entre deux stratégies de réplication de données avec d'un coté une stratégie qui considère le profit économique, et d'un autre une stratégie qui considère la consommation énergétique. Cela nous permettra de constater que peu de stratégies tentent de répondre à ces deux problématiques, malgré la forte demande de l'industrie qui se lance dans une course à la réduction de la consommation énergétique et de leur impact carbone.

Mots-clés : Cloud, Réplication de données, Profit économique, Consommation énergétique

1. Introduction

Depuis plusieurs décennies, de plus en plus de données sont générées par les utilisateurs. Ces données, qui peuvent aussi bien être du texte que des images satellites, ont besoin d'être accessibles de plus en plus rapidement, où que l'on soit dans le monde. La réplication de données permet de répondre à ce type de problématique. Elle est utilisée dans différents environnements, dont le Cloud Computing [3].
Le Cloud est caractérisé, comme les autres architectures, par des attributs qui lui sont propres, notamment, le *Service Level Agreement* [17] qui lie le fournisseur et le locataire par un contrat, ou encore la gestion élastique des ressources [7], qui permet à un locataire de ne payer que ce qu'il consomme, i.e., le modèle *Pay-as-you-go*.
Les stratégies de réplication de données tentent de répondre à des problématiques de disponibilité, de performance [9] et de tolérance aux fautes [16]. Dans le système Cloud, ces stratégies doivent aussi tenir compte de contraintes financières et énergétiques. Pour cela, certaines stratégies s'intéressent au mix énergétique utilisé dans un Centre de Données (CD), d'autres répliquent de manière économique sur des serveurs à plus faible coût. Cependant, peu de straté-

gies traitent ce type de sujet indépendemment et il existe encore moins de stratégies qui traitent de ces deux sujets simultanément.
Le papier s'organise de la manière suivante. Nous allons dans un premier temps faire un état de l'art sur les stratégies de réplication de données, et notamment celles qui traitent de la consommation énergétique et du profit (2). Puis nous allons exposer notre modèle d'estimation de la consommation énergétique (3). Nous confronterons ensuite deux stratégies de réplication de données, afin de comparer leurs impacts sur la consommation énergétique et sur leurs dépenses (4). Enfin, nous finirons cet article avec une conclusion et les futurs travaux (5).

2. État de l'art

Dans cet article, nous nous concentrons sur les stratégies de réplication de données qui ont pour objectif de maximiser le profit ou de réduire la consommation énergétique.
Pour commencer avec les problèmes économiques, une première stratégie sur le Cloud [10] essaie de réduire au maximum le nombre de répliques en gardant la même probabilité de perte de données, afin de faire des économies. Plus récemment, la stratégie [6], s'est intéressée à la minimisation des dépenses selon sur quelle couche de CDs la réplication est réalisée. [24] prend en compte la question de la dépense dans la réplication en se basant sur les informations financières et les qualités de services de chaque noeud. La stratégie [21] s'intéresse pour sa part à la maximisation du profit en estimant le profit généré par une réplication.
Une intéressante stratégie de réplication de données a été proposée [1], qui prend en compte la consommation énergétique afin de réduire le coût d'une requête. C'est-à-dire qu'elle réduit la consommation énergétique pour réduire le coût global.
Pour répondre aux problèmes de la consommation énergétique, l'une des premières stratégies de réplication de données est [11]. Elle cherche à optimiser une fonction objectif, qui contient plusieurs buts dont la disponibilité, le temps de service, la charge de travail, la latence moyenne et enfin la consommation énergétique qui sont pondérés par des poids définis par l'administrateur. Une autre stratégie proposée [2] modélise la consommation énergétique et l'utilisation de la bande-passante. Pour réduire sa consommation énergétique, elle va répliquer la donnée dans des bases de données plus proches (CD, racks) des serveurs de calculs si ces dernières consomment moins d'énergie et de bande-passante que la base de données du niveau supérieur (centrale, CD). Cela s'applique lorsque le nombre d'accès à cette donnée dépasse un certain seuil. Une autre stratégie de réplication de données [25] prend le problème d'une autre manière. Le but de la réplication est alors de regrouper la charge de travail sur quelques noeuds afin de pouvoir mettre en veille voire éteindre les noeuds inactifs. Enfin, une dernière stratégie qui prend en compte la consommation énergétique est [23] qui cherche à réduire l'émission de gaz à effet de serre. Pour cela, elle va essayer de réduire la consommation énergétique en répliquant dans des CDs qui ont un mix énergétique plus propre.
Nous voyons à la suite de cet état de l'art, qu'il y a presque autant de stratégies de réplication de données qui prennent en compte la consommation énergétique que de stratégies qui prennent en compte le profit du fournisseur. Cependant, ils correspondent à un petit ensemble comparé au reste des stratégies, qui traitent des performances et de la disponibilité. De plus nous voyons qu'à l'intersection de ces sous ensembles d'objectifs, très peu de stratégies sont proposées.

3. Modélisation de la consommation énergétique

Dans l'objectif de comparer plusieurs stratégies qui essaient de maximiser le profit et de minimiser la consommation énergétique, une modélisation de la consommation énergétique à

l'échelle d'un Cloud est proposée. Ainsi nous allons estimer la consommation énergétique de chaque composant en estimant sa puissance et son temps d'utilisation.

3.1. Composants

Nous proposons des modèles de consommation énergétique pour chaque composant, que nous allons ensuite utiliser pour modéliser l'interaction entre ces derniers.

Processeur : Le *Central Processing Unit* ou CPU, est le composant qui consomme le plus dans un serveur [12]. Nous allons donc nous baser sur le modèle de [14] qui a été repris par [22] pour estimer la consommation énergétique d'un hôte en se basant sur la charge du CPU :

$$E_{CPU}(l, f) = E_{CPU_{act}}(l, f) + E_{CPU_{idle}}(f) \tag{1}$$

$$= \sum_{t=0}^{T} (P^f_{CPU_{max}} - P^f_{CPU_{idle}}) * \alpha * \frac{1}{l} * \frac{NbInstruct}{f} + P^f_{CPU_{idle}} * T_{tot} \tag{2}$$

Où $P^f_{CPU_{max}}$ et $P^f_{CPU_{idle}}$ correspondent à la puissance maximale et en inactivité respectivement fournit par le CPU en Watt (W) pour une fréquence f en gigahertz (GHz).La première soustraction de l'équation 2 représente la puissance dynamique du processeur. T et *NbInstruct* correspondent respectivement au nombre de tâches à effectuer sur un processeur ainsi que le nombre d'instructions de chacune des tâches. T_{tot} représente le temps total d'exécution et de transfert en seconde (s) Et enfin, l et α correspondent respectivement au pourcentage du processeur alloué à une tâche et au pourcentage de la puissance dynamique associé à l. Nous supposons, comme dans [14], une linéarité entre l et la puissance dynamique, donc $\alpha = l$.

$$E_{CPU}(f) = \sum_{t=0}^{T} (P^f_{CPU_{max}} - P^f_{CPU_{idle}}) * \frac{NbInstruct}{f} + P^f_{CPU_{idle}} * T_{tot} \tag{3}$$

Cette équation permet d'obtenir l'énergie consommée par le CPU lors de l'exécution d'une suite de tâches en Joules (J).

Mémoire : Dans le cadre de cet article, seule la mémoire RAM sera modélisée. Nous nous basons sur les modèles de puissance de la mémoire RAM proposés par [15] et utilisé plus récemment dans la fiche technique de calcul de puissance de Micron [13]. Cela permet de mettre en place différents états d'utilisation (inactif, lecture, écriture), qui dépendent de variables comme les taux d'écritures ou de lectures. Nous obtenons donc la formule suivante :

$$E_{RAM} = E_{RAM_{read}} + E_{RAM_{write}} + E_{RAM_{idle}} \tag{4}$$

$$= \sum_{r=0}^{R} (P_{RAM_{read}} * \frac{size(File)}{BW_{RAM}}) + \sum_{w=0}^{W} (P_{RAM_{write}} * \frac{size(File)}{BW_{RAM}}) + P_{RAM_{idle}} * T_{tot} \tag{5}$$

Nous avons dans un premier temps R et W signifiant le nombre de lectures et d'écritures à faire respectivement, puis la taille du fichier qui est notée size(File) en mégaoctet (Mo) et la bande-passante de la mémoire RAM, notée BW_{RAM} en mégaoctet par seconde (Mo/s).

Disque dur : le disque dur modélisé est un *Hard Drive Disk*. Le modèle est proposé par [8] pour estimer la consommation énergétique en se basant sur les fiches techniques. [18] l'a utilisé plus récemment pour estimer la consommation énergétique de différent disques dur de type *Solid-State Drive*. Le modèle est le suivant :

$$E_{HDD} = E_{HDD_{act}} + E_{HDD_{idle}} \tag{6}$$

$$= \sum_{i=0}^{N} P_{HDD_{Active}} * \frac{size(File)}{BW_{HDD}} + \sum_{j=0}^{S} P_{HDD_{Seek}} * T_{Seek} + P_{HDD_{idle}} * T_{tot} \tag{7}$$

Où $P_{HDD_{Active}}$ (W) correspond à la puissance nécessaire au disque dur pour lire ou écrire, la taille du fichier et la bande-passante sont notées size(File) (Mo) et BW_{HDD} (Mo/s) respectivement. $P_{HDD_{Seek}}$ (W) correspond à la consommation énergétique lors de la recherche sur le disque et T_{Seek} (s) le temps de recherche.

Carte réseau : Pour prendre en compte la consommation énergétique de la carte réseau, nous nous sommes basés sur le document qui présente le standard 802.3az, qui permet de larges réductions de la consommation énergétique lorsqu'il est inactif [5]. Nous utilisons ainsi le modèle suivant pour un nombre T de transferts :

$$E_{NIC} = E_{NIC_{act}} + E_{NIC_{idle}} \tag{8}$$

$$= \sum_{t=0}^{T} (P_{NIC_{Active}} * \frac{size(File)}{BW_NIC}) + P_{NIC_{idle}} * T_{tot} \tag{9}$$

3.2. Utilisation des modèles de consommation énergétique

Dans cette partie, nous considérons deux types de consommation énergétique. La première est la consommation statique, qui correspond à la consommation de base de tous les composants. La seconde est la consommation dynamique, qui correspond à la consommation en activité de ces composants qui est ajouté à la première. Cette dernière est représentée par l'exécution d'une tâche, le transfert de données, et le stockage de données.

Consommation statique : La consommation énergétique statique correspond à la consommation de base de tous les éléments. Ainsi, nous prenons en compte la consommation énergétique tout au long de l'exécution des tâches.

$$E_{Idle} = (P_{CPU_{idle}} + P_{RAM_{idle}} + P_{HDD_{idle}} + P_{NIC_{idle}}) * T_{tot} \tag{10}$$

Exécution de tâches : Durant l'exécution d'une tâche, nous avons la donnée déjà dans la mémoire. Le CPU va donc lire la donnée présente dans la mémoire et effectuer sa tâche :

$$E_{Pro} = E_{CPU_{Act}} + E_{RAM_{read}} \tag{11}$$

Transfert de données : Afin d'exécuter une tâche la donnée associée à cette dernière peut être sur le noeud de calcul. Dans ce cas, il suffit de lire la donnée sur le disque dur et de l'écrire dans la mémoire (src=tgt). Sinon, cela signifie que la donnée est stockée sur un autre noeud, et il faudra donc prendre en compte le transfert entre la source (src) et le noeud de calcul (tgt).

$$\begin{aligned} E_{FileTransfer} =& E_{HDD_{read_{src}}} + \mathbb{1}_{src \neq tgt} * (E_{RAM_{write_{src}}} + E_{RAM_{read_{src}}} + \\ & E_{NIC_{src}} + E_{NIC_{tgt}}) + E_{RAM_{write_{tgt}}} \end{aligned} \tag{12}$$

Stockage d'une donnée

Tout comme le transfert de fichier, le stockage de la donnée, au moment de la réplication par exemple, peut s'effectuer sur le noeud qui fait le calcul. Ainsi, il suffit de lire la donnée sur la mémoire RAM et de l'écrire dans le disque dur (tgt=src). Sinon, la réplication se fait sur un autre noeud. Il faudra à nouveau prendre en compte la consommation du transfert de données entre celui qui l'a en mémoire (src) et celui sur lequel aura lieu la réplication (tgt).

$$\begin{aligned} E_{Replication} =& E_{RAM_{read_{src}}} + \mathbb{1}_{src \neq tgt} * (E_{NIC_{src}} + E_{NIC_{tgt}} + \\ & E_{RAM_{write_{tgt}}} + E_{RAM_{read_{tgt}}}) + E_{HDD_{write_{tgt}}} \end{aligned} \tag{13}$$

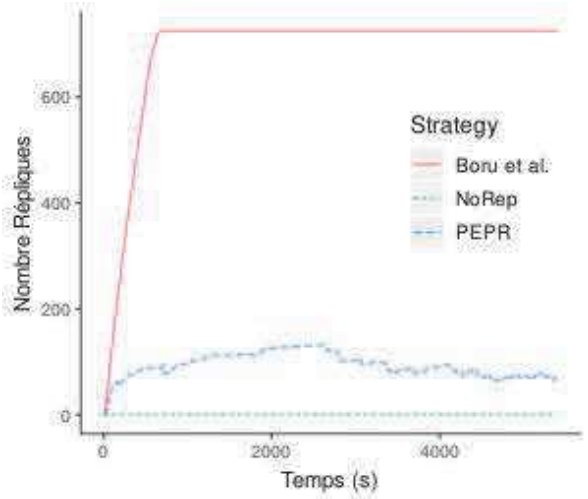

FIGURE 1 – Réplications durant l'exécution

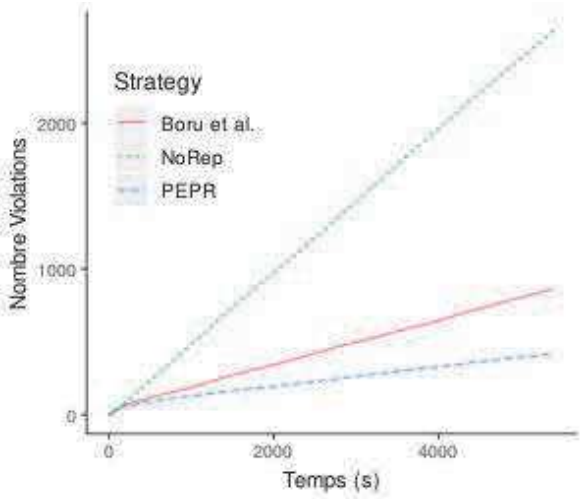

FIGURE 2 – Violations durant l'exécution

4. Expériences

4.1. Mise en place

Afin de comparer ces stratégies de réplication de données, nous avons mis en place des expériences sur le simulateur CloudSim [4]. Ce dernier a été étendu durant la thèse d'Uras Tos [19] afin de prendre en compte la réplication de données ainsi que les coûts monétaires des ressources [21]. Il a de nouveau été étendu pour implémenter nos modèles de consommation énergétique présentés dans la partie 3.2. Nous comparons la stratégie de réplication de données PEPR [21], la stratégie de Boru et al. [2], et l'absence de réplication (NoRep). Nous avons testé ces stratégies selon les paramètres présentés dans la Table 1 (en annexe) basés sur [20].

Dans l'article [2], l'architecture du Cloud n'est pas la même que celle de PEPR. Dans Boru et al., seules les bases de données peuvent stocker des données et sont réparties sur trois niveaux : centrale, CD, racks. Dans PEPR, il n'y a pas de telle hiérarchie entre les noeuds, et chaque noeud peut stocker des données. Chaque noeud appartient à un CD qui est situé dans une région. NoRep est mis en place sur la même topologie que PEPR. Afin de respecter l'architecture de Boru et al. au sein de PEPR, nous avons ajouté à l'architecture proposée dans les paramètres, un CD par région qui correspond aux Bases de Données de la région, et une région qui correspond à la base de données centrale. Cependant, afin d'avoir une certaine équité, ces bases de données supplémentaires ne sont pas prises en compte dans le calcul de l'énergie. En effet, les bases de données supplémentaires dans l'architecture de Boru et al. existent car les noeuds ne stockent pas de données, contrairement à l'architecture de PEPR. Cela nous permet donc de comparer les 2 stratégies le plus équitablement possible avec la même capacité de stockage et de calcul.

Pour comparer ces différentes stratégies de réplication de données, nous avons mis en place quatre métriques. La première métrique utilisée est le nombre de répliques créées, qui permet de voir la gestion des répliques par les stratégies. La seconde est le nombre de violations de SLO, c'est-à-dire le nombre d'exécutions dont le temps de réponse est supérieur à l'objectif présenté dans la Table 1. Enfin, les deux dernières métriques correspondent à la consommation énergétique (J) et aux dépenses du Cloud en dollars.

4.2. Résultats

Nous voyons sur la Figure 1, que la réplication dans le cadre de la stratégie de Boru et al. augmente fortement au début afin de répondre aux problématiques énergétique et de bande-passante. Pour PEPR, cela est légèrement différent. En effet, nous voyons que PEPR augmente

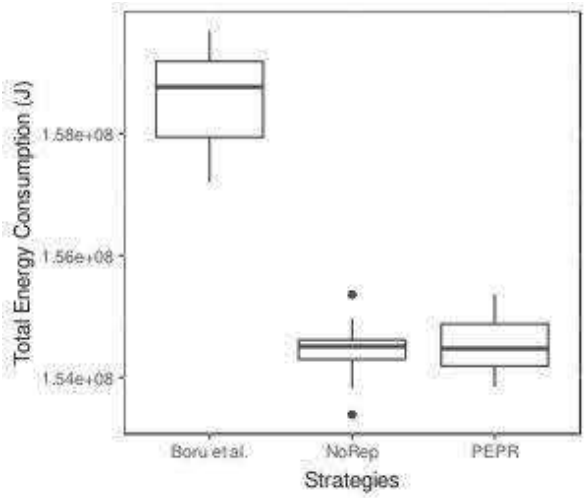

FIGURE 3 – Énergie consommée

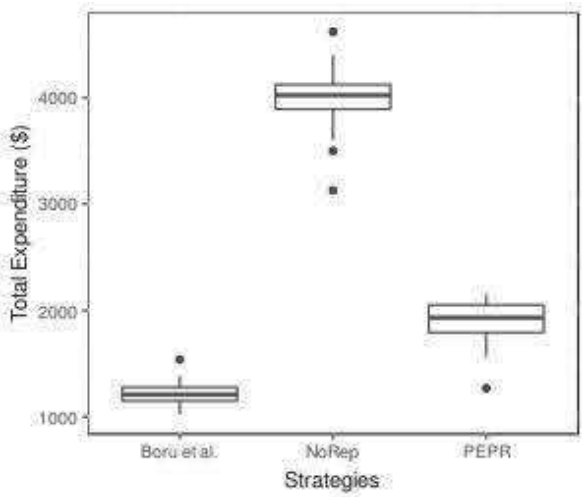

FIGURE 4 – Total des dépenses

ou diminue le nombre de répliques dans le temps. Cela permet d'adapter la stratégie selon les violations et selon les coûts. Nous voyons sur ces stratégies que cela a un impact sur le nombre de violations (Figure 2). Cependant, ces violations peuvent aussi provenir de l'architecture du Cloud et des CDs, car dans PEPR, toutes les VMs peuvent stocker des données, ce qui n'est pas le cas de Boru et al. où un rack par cluster peut stocker des données.

La Figure 3 montre que la stratégie de réplication proposée par Boru et al. consomme plus d'énergie que PEPR et NoRep. Cette différence est notamment due à une consommation accrue des cartes réseaux. En effet, Boru et al. réplique les données indépendemment de l'exécution d'une tâche. La réplication se fait donc en plus des tâches à exécuter. Contrairement à PEPR, où la donnée est répliquée au mieux sur le noeud qui calcule, ou au pire sur un noeud proche (même CD ou même région), ce qui implique des coûts supplémentaires.

À l'inverse dans la Figure 4, on constate que la stratégie de Boru et al. est moins dépensière que PEPR, mais que ces stratégies restent bien plus économe que NoRep. En effet, si PEPR ne génère pas de profit sur la réplication, alors il ne réplique pas, ce qui peut avoir des effets dramatiques sur la dépense totale. Car cela implique une absence de réplication qui peut entraîner un augmentation des violations de SLA et donc une augmentation du coût global. D'autant plus qu'une suppression des répliques est mise en place dans PEPR. Pour NoRep, l'absence de réplication force les données à être transférées entre les régions ce qui génère un coût de communication plus important. Boru et al. économise sur les transferts inter-région car il va avant tout répliquer les données fréquemment utilisées dans chaque région pour diminuer la consommation énergétique et surtout diminuer la charge de la bande-passante.

5. Conclusion et perspectives

Nous avons comparé les performances de deux stratégies de réplication ayant des objectifs différents. La stratégie PEPR [21] permet de réduire les coûts sans aucun effet sur la consommation énergétique. Contrairement à Boru et al. [2] qui bénéficie de la réplication pour réduire ses coûts économiques mais avec une consommation importante liée à la communication entre serveurs. Cette comparaison pourrait être étendue en ajoutant notamment l'énergie consommée par l'infrastructure réseau. De plus, il serait intéressant d'établir une nouvelle stratégie de réplication de données qui prenne en compte les deux objectifs pour une optimisation multi-critères qui permettra de réaliser de meilleurs compromis.

Bibliographie

1. Alghamdi (M.), Tang (B.) et Chen (Y.). – Profit-based file replication in data intensive cloud data centers. – In *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, Paris, France, mai 2017. IEEE.
2. Boru (D.), Kliazovich (D.), Granelli (F.), Bouvry (P.) et Zomaya (A. Y.). – Energy-efficient data replication in cloud computing datacenters. *Cluster Computing*, vol. 18, n1, mars 2015, pp. 385–402.
3. Buyya (R.), Yeo (C. S.), Venugopal (S.), Broberg (J.) et Brandic (I.). – Cloud computing and emerging IT platforms : Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, vol. 25, n6, juin 2009, pp. 599–616.
4. Calheiros (R. N.), Ranjan (R.), Beloglazov (A.), De Rose (C. A. F.) et Buyya (R.). – CloudSim : a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software : Practice and Experience*, vol. 41, n1, janvier 2011, pp. 23–50.
5. Christensen (K.), Reviriego (P.), Nordman (B.), Bennett (M.), Mostowfi (M.) et Maestro (J.). – IEEE 802.3az : the road to energy efficient ethernet. *IEEE Communications Magazine*, vol. 48, n11, novembre 2010, pp. 50–56.
6. Gill (N. K.) et Singh (S.). – A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers. *Future Generation Computer Systems*, vol. 65, décembre 2016, pp. 10–32.
7. Herbst (N. R.), Kounev (S.) et Reussner (R.). – Elasticity in Cloud Computing : What It Is, and What It Is Not. *ICAC*, vol. 13, 2013, pp. 23–27.
8. Hylick (A.) et Sohan (R.). – A methodology for generating disk drive energy models using performance data. *Energy (Joules)*, vol. 80, 2009, p. 100.
9. Kumar (K. A.), Quamar (A.), Deshpande (A.) et Khuller (S.). – SWORD : workload-aware data placement and replica selection for cloud data management systems. *The VLDB Journal*, vol. 23, n6, décembre 2014, pp. 845–870.
10. Li (W.), Yang (Y.) et Yuan (D.). – A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres. – In *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, pp. 496–502, Sydney, Australia, décembre 2011. IEEE.
11. Long (S.-Q.), Zhao (Y.-L.) et Chen (W.). – MORM : A Multi-objective Optimized Replication Management strategy for cloud storage cluster. *Journal of Systems Architecture*, vol. 60, n2, février 2014, pp. 234–244.
12. Mastelic (T.), Oleksiak (A.), Claussen (H.), Brandic (I.), Pierson (J.-M.) et Vasilakos (A. V.). – Cloud Computing : Survey on Energy Efficiency. *ACM Computing Surveys*, vol. 47, n2, décembre 2014, pp. 1–36.
13. Micron. – TN-40-07 : Calculating Memory Power for DDR4 SDRAM, 2017.
14. Minas (L.) et Ellison (B.). – The Problem of Power Consumption in Servers, 2009.
15. Rhu (M.), Sullivan (M.), Leng (J.) et Erez (M.). – A locality-aware memory hierarchy for energy-efficient GPU architectures. – In *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 86–98, Davis, California, décembre 2013. ACM Press.
16. Selvi (S. A. E.) et Anbuselvi (R.). – RAAES : Reliability-Assured and Availability-Enhanced Storage for Cloud Environment. *International Journal of Pure and Applied Mathematics*, vol. 118, février 2018, pp. 103–112.
17. Serrano (D.), Bouchenak (S.), Kouki (Y.), de Oliveira Jr. (F. A.), Ledoux (T.), Lejeune (J.),

Sopena (J.), Arantes (L.) et Sens (P.). – SLA guarantees for cloud services. *Future Generation Computer Systems*, vol. 54, janvier 2016, pp. 233–246.
18. Song (M.). – Minimizing Power Consumption in Video Servers by the Combined Use of Solid-State Disks and Multi-Speed Disks. *IEEE Access*, vol. 6, juin 2018, pp. 25737–25746.
19. Tos (U.). – *Data replication in large-scale data management systems*. – Theses, Université Paul Sabatier - Toulouse III, juin 2017.
20. Tos (U.), Mokadem (R.), Hameurlain (A.), Ayav (T.) et Bora (S.). – A Performance and Profit Oriented Data Replication Strategy for Cloud System. – In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, p. 10, Toulouse, France, juillet 2016. IEEE.
21. Tos (U.), Mokadem (R.), Hameurlain (A.), Ayav (T.) et Bora (S.). – Ensuring performance and provider profit through data replication in cloud systems. *Cluster Computing*, décembre 2017.
22. Wang (H.) et Tianfield (H.). – Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters. *IEEE Access*, vol. 6, mars 2018, pp. 15259–15273.
23. Xu (Z.), Deng (N.), Stewart (C.) et Wang (X.). – CADRE : Carbon-Aware Data Replication for Geo-Diverse Services. – In *2015 IEEE International Conference on Autonomic Computing*, pp. 177–186, Grenoble, France, juillet 2015. IEEE.
24. Zeng (L.), Xu (S.), Wang (Y.), Kent (K. B.), Bremner (D.) et Xu (C.). – Toward cost-effective replica placements in cloud storage systems with QoS-awareness : Toward cost-effective replica placements in cloud storage systems with QoS-awareness. *Software : Practice and Experience*, vol. 47, n6, juin 2017, pp. 813–829.
25. Zhang (L.), Deng (Y.), Zhu (W.), Zhou (J.) et Wang (F.). – Skewly replicating hot data to construct a power-efficient storage cluster. *Journal of Network and Computer Applications*, vol. 50, avril 2015, pp. 168–179.

Paramètres	Valeurs	Paramètres	Valeurs
Nombre de fichiers	30	Délai entre les région	150 ms
Taille moyenne des fichiers	600 MB	Délai dans une région	50 ms
Nombre de tâches	100,000	Nb de VMs par CD	10
Taille minimale de la tâche	1,000 MI	Nb de CDs par région	3
Taille maximale de la tâche	7,500 MI	Nb de régions par Cloud	5
Durée de la simulation	1h30	Revenu par tâche	0.01$
Capacité de calcul d'une VM	1,500 MIPS	Coût du calcul d'une tâche	$4\$/10^9$ MI
BP entre les région	500 Mbit/s	Coût de stockage	0.04$/GB
BP dans une région	3 Gbit/s	Coût de transfert entre régions	0.04$/GB
BP dans un centre	10 Gbit/s	Coût de transfert d'une région	0.004$/GB
Délai dans un CD	10 ms	Coût de transfert dans un CD	$8*10^{-4}\$$/GB
Objectif de temps de réponse	35s	Pénalité par violation de SLO	0.001$

TABLE 1 – Annexe - Paramètres de la simulation