

UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR



ECOLE DOCTORALE: MATHEMATIQUES ET INFORMATIQUE

Année : 2019 N° d'ordre : 122

THESE DE DOCTORAT

Mention : INFORMATIQUE

Présentée par:

Fatoumata BALDE KASSE

Titre:

Réduction de la consommation d'énergie dans les réseaux de transport de données et les infrastructures Cloud

Soutenu le 12 Janvier 2019 devant le jury composé de :

Président	Karim KONATE	Professeur	UCAD, Sénégal
Rapporteurs	Ousmane THIARÉ	Professeur	UGB, Sénégal
	Joel RODRIGUES	Professeur	INATEL, Brésil
Examineurs	Mbaye SENE	Maître de Conférences (CAMES)	UCAD, Sénégal
	Ibrahima NIANG	Professeur	UCAD, Sénégal
Directeur de thèse	Cheikh Ahmadou Bamba GUEYE	Maître de Conférences (CAMES)	UCAD, Sénégal

REMERCIEMENTS

Je remercie tout d'abord ALLAH(SWT) qui m'a donné la force et la possibilité de continuer.

Je tiens à remercier mon directeur de thèse, Pr. Cheikh Ahmadou Bamba GUEYE, pour m'avoir fait confiance malgré les connaissances plutôt légères que j'avais au début de ma thèse en 2012 sur le Cloud, puis pour m'avoir guidée, encouragée, et conseillée pendant des années tout en me laissant une grande liberté et en me faisant l'honneur de me déléguer plusieurs responsabilités dont j'espère avoir été à la hauteur. Sa disponibilité et ses conseils ont été d'une grande qualité.

Tous mes remerciements au Pr. Karim KONATE pour avoir accepté d'être le président du jury. Je le remercie également pour ses conseils sa disponibilité, sa simplicité et pour avoir participé à ma formation.

Je remercie aussi Pr. Ousmane THIARE et Pr. Joel RODRIGUES qui ont accepté de rapporter cette thèse, malgré les contraintes de temps et leurs responsabilités.

Je remercie également Pr. Ibrahima NIANG et Pr. Mbaye SENE pour avoir accepté d'être les examinateurs de cette thèse, mais aussi pour avoir participé à ma formation.

Je remercie aussi mon mari Youssou KASSE pour les discussions que j'ai eu avec lui, ses encouragements et son assistance morale qui m'ont permis de faire cette thèse dans de bonnes conditions.

Je remercie également ma mère pour son assistance et ses prières et l'éducation qu'elle m'a inculquée.

Je remercie mes parents, beaux-parents et mes amis pour leurs encouragements et prières tout au long de la réalisation de ce mémoire.

Je remercie également la professeure Halima Elbiaze de L'UQAM pour ses contributions par rapport à mes travaux, son soutien son assistance et sa disponibilité.

DÉDICACES

À la mémoire de mon défunt père.

À la meilleure et merveilleuse maman sur terre,

À l'homme le plus généreux, le plus patient et le plus doux que je n'ai jamais rencontré, je veux parler de mon cher et tendre époux.

À mes chers enfants : Awa, Sergine et Diadia.

À mes frères et à ma sœur.

À ma tante et homonyme.

À Cheikh Ahmadou Bamba Gueye qui n'arrête jamais de me motiver.

À mes beaux parents et à toute ma famille.

À tous mes amis et collègues.

À tous ceux qui, par un mot, m'ont donné la force de continuer

TABLE DES MATIERES

TABLE DES MATIERES	v
LISTE DES FIGURES.....	viii
LISTE DES TABLEAUX.....	x
LISTE DES ABREVIATIONS.....	xi
RÉSUMÉ	xii
Chapitre 1 : INTRODUCTION GENERALE	1
1.1. Contexte	1
1.2. Objectifs	2
1.3. Organisation de la thèse	3
Chapitre 2 : ETAT DE L'ART SUR LE CLOUD ET LES TOPOLOGIES DES CENTRES DE DONNEES	4
2.1 Introduction	5
2.2 Cloud	6
2.2.1 Historique.....	6
2.2.2 Définition	8
2.2.3 Caractéristiques	10
2.2.4 Architecture.....	13
2.2.5 Modèle de déploiement	17
2.2.6 Outils d'implémentation et de simulation	20
2.3 Topologies des centres de données	23
2.3.1 Historique.....	23
2.3.2 Topologies des réseaux de centre de données.....	24
2.3.4 Comparaison des performances de quelques topologies.....	37
2.4 Conclusion	41
Chapitre 3 : ETAT DE L'ART SUR L'ECONOMIE D'ENERGIE DANS LE CLOUD COMPUTING	42
3.1 Introduction	43
3.2 Les principales causes des pertes d'énergie	44
3.2.1 Les serveurs.....	45

3.2.2 Le système de refroidissement.....	46
3.3 Gestion et analyse du trafic dans le Cloud privé et public	46
3.4 Techniques de réduction de la consommation d'énergie dans les centres de données.....	48
3.4.1 Approches basées sur les méthodes de virtualisation et de migration de machines virtuelles.....	48
3.4.2 Approches existantes pour réduire les besoins en énergie du réseau.....	53
3.4.3 Économie d'énergie dans les réseaux d'accès	55
3.4.4 Économie d'énergie basée sur l'organisation du centre de données.....	56
3.5 Conclusion.....	58
Chapitre 4 : "GREEN CLOUD" : Une architecture efficace pour la réduction de la consommation d'énergie	60
4.1 Introduction.....	61
4.2 Optimisation de la virtualisation des réseaux pour un transport éco-énergétique.....	62
4.3 Virtualisation Réseau/Dispositif	63
4.4 Présentation de GCA (Green Cloud Architecture).....	65
4.4.1 Objectif.....	66
4.4.2 Évaluations analytiques de GCA	68
4.4.3 Algorithme d'allocation de ressources.....	73
4.5 Évaluation de performance	76
4.5.1 Paramètres de simulation	76
4.5.2 Résultats	77
4.6 Conclusion.....	82
Chapitre 5 :Gestion de la qualité de service dans les centres de données (GREENPOD).....	83
5.1 Introduction.....	84
5.2 Contexte de la proposition.....	84
5.3 Présentation du modèle.....	86
5.3.1 Modèle analytique.....	92
5.3.2 Calcul des probabilités	94
5.3.3 Calcul des métriques de performance	96
5.4 Résultats numériques.....	97

5.5 Conclusion	109
Chapitre 6 : CONCLUSION GÉNÉRALE.....	110
6.1 Bilan	110
6.2 Perspectives	111
6.3 Publications relatives à la thèse	112
BIBLIOGRAPHIE	i

LISTE DES FIGURES

Figure 2.1: Cloud(©Wikipedia).....	10
Figure 2.2: Différence entre le modèle de paiement classique et celui du Cloud[HAMZE, Thèse doctorat 2015]	12
Figure 2.3: Les couches du Cloud.....	13
Figure 2.4: Abstraction du Cloud Computing[Marinos 2009].....	14
Figure 2.5: Une vue technique du Cloud [Young-Chan 2012]	18
Figure 2.6: Taxonomie des topologies de centre de données [Yang et al. 2013]	25
Figure 2.7: Topologie arborescente à trois niveaux [Kliazovich et al. 2010].....	26
Figure 2.8: Topologie Fat Tree [Yang et al. 2013]	28
Figure 2.9: La topologie Fat-tree de Facebook [Andreyev 2014].....	29
Figure 2.10: La topologie Clos Network [Yang et al. 2013]	30
Figure 2.11: Construire un DCell1 à partir de DCell0s avec $n = 4$ [Chirgwin 2015].	32
Figure 2.12: Construction de Bcube1 à partir de Bcube0 pour $n=4$ [Chirgwin 2015]	33
Figure 2.13: Topologie C-Through [Yang et al. 2013].....	35
Figure 2.14: Topologie Helios [Yang et al. 2013]	36
Figure 3.1: Répartition de la consommation d'énergie dans un centre de données [Jing et al. 2017].....	45
Figure 4.1: Affectation des nœuds virtuels sur un réseau physique.....	64
Figure 4.2: Green Cloud Architecture (GCA)	66
Figure 4.3: Fonction de distribution cumulative des lignes de carte activées en fonction de la charge	78
Figure 4.4: Consommation globale du réseau.....	79
Figure 4.5: Pourcentage des nœuds virtuels bloqués	80
Figure 4.6: Nombre de réseaux virtuels mappés et médiane des contraintes de nœud	81
Figure 4.7: Réseaux virtuels bloqués	82
Figure 5.1: Topologies Fat-tree.....	87
Figure 5.2: Modèle du système	88
Figure 5.3: Différents états d'un serveur du premier groupe	89
Figure 5.4: Comportement d'un serveur des groupes restants.....	90
Figure 5.5: Chaîne de Markov du modèle proposé	93
Figure 5.6: La puissance consommée en fonction du nombre de clients dans le système.....	97
Figure 5.7: Consommation d'énergie en fonction du temps de simulation pour un système peu sollicité	99
Figure 5.8: Consommation d'énergie en fonction du temps de simulation pour un système moyennement sollicité	100
Figure 5.9: Consommation d'énergie et temps d'attente de notre proposition en fonction du temps de simulation pour un système moyennement sollicité.....	101

Figure 5.10: Consommation d'énergie en fonction du temps de simulation pour un système très sollicité	102
Figure 5.11: Consommation d'énergie et temps d'attente de notre proposition en fonction du temps de simulation pour un système très sollicité.....	103
Figure 5.12: Temps d'attente des deux propositions en fonction du temps de simulation.....	103
Figure 5.13: Consommation d'énergie des deux propositions en fonction du nombre de serveurs.....	104
Figure 5.14: Consommation d'énergie de notre proposition par rapport à un centre de données par classique.....	105
Figure 5.15: Consommation d'énergie des Switch par rapport au deux propositions	106
Figure 5.16: Consommation totale d'énergie (Switch+serveurs) par rapport au deux propositions.....	106
Figure 5.17: Consommation totale d'énergie (Switch+serveurs) de notre proposition par rapport à un centre de données par classique.....	107
Figure 5.18: Consommation d'énergie des serveurs en fonction de la charge du système.....	108
Figure 5.19: Energie consommée (Serveurs + Switch) en fonction de la charge	109

LISTE DES TABLEAUX

Tableau 2.1: Exemple d'outils d'implémentation de Cloud	21
Tableau 2.2: Logiciels de simulation de Cloud.....	22
Tableau 2.3: Résumé de la performance des topologies [Yang et al. 2013].....	39
Tableau 2.4: Résumé de la redondance matérielle [Yang et al. 2013].....	40
Tableau 4.1: Résumé de la consommation d'énergie des routeurs	76
Tableau 5.1: Nombre de serveurs actifs et les seuils d'activation et de désactivation à chaque niveau.....	92

LISTE DES ABREVIATIONS

AP	Access Point	Point d'accès
DCN	Data center Network	Réseau de centre de données
DVFS	Dynamic Voltage FrequencyScaling	Mise à l'échelle dynamique de la fréquence de tension
EA- VNE	Energy-Aware Virtual Network Embedding	Intégration de réseaux virtuels éco- énergétiques
EMS	Energy Management System	Système de gestion d'énergie
EoR	End of Row	End of Row
EPA	Environmental Protection Agency	Agence de protection environnementale
IaaS	Infrastructure as a Service	Infrastructure en tant que service
ISP	Infrasutstructure service Provider	Fournisseur de service
MBFD	Modified Best Fit Decreasing	Algorithme (MBFD)
NaaS	Network as a Service	Réseau en tant que service
NPA	No Power Aware	
PaaS	Plateforme as a service	Plateforme en tant que service
Pod	Performance Optimized Datacenter	Optimisé les performances des DC
QoS	Quality of service	Qualité de service
SaaS	Software as a Service	Logiciel en tant que service
SLA	Service Level Agreement	Accord de niveau de service
ST	Single Threshold	Seuil unique
ToR	Top of Rack	Top of Rack
UADB	Universite Alioune Diop de Bambey	Universite Alioune Diop de Bambey
VM	Virtual Machine	Machine virtuelle
WDM	wavelength-division multiplexing	Multiplexage en longueur d'onde

RÉSUMÉ

Réduire l'énergie consommée dans le Cloud devient une direction de recherche de plus en plus difficile en raison de la croissance exponentielle des services qui y sont hébergés et livrés. De nos jours, la réduction de la consommation d'énergie dans les centres de données a reçu beaucoup d'attentions des chercheurs, l'objectif étant d'augmenter l'efficacité énergétique dans le Cloud. Malgré ces efforts, des travaux récents ont montré que le Cloud n'est pas toujours une technologie verte.

Les coûts énergétiques élevés liés à l'exploitation d'un centre de données ont conduit à une réflexion sur le fonctionnement efficace d'un centre de données. Les centres de données sont conçus pour supporter la charge de trafic maximale attendue. La quantité de données transportée entre les clients et les fournisseurs étant de plus en plus importante, il serait nécessaire de trouver des mécanismes permettant d'atténuer les coûts énergétiques occasionnés. En fonction aussi de la charge du trafic actuel, le fournisseur doit pouvoir adapter de manière dynamique le nombre de ressources offertes.

Dans nos travaux, nous avons proposé une architecture de Cloud écologique appelée Green Cloud Architecture (GCA), dans laquelle on peut instancier des routeurs virtuels sur des routeurs physiques. Cette proposition intègre la possibilité de migrer des routeurs virtuels vers d'autres nœuds physiques avec la possibilité de les éteindre ou les mettre en veille pour des besoins d'économie d'énergie. Pour réduire la consommation d'énergie dans les centres de données, nous avons conçu un algorithme d'allocation de ressources basé sur l'énergie courante qui intègre les routeurs virtuels sur les routeurs physiques tout en cherchant à minimiser l'énergie consommée pendant le transport.

Dans la troisième proposition appelée GREENPOD, nous proposons un modèle basé sur la théorie des files d'attente permettant de réduire la consommation d'énergie du

centre de données (routeurs, Switch et serveurs) tout en garantissant un temps de réponse acceptable. Nous utilisons des politiques de gestion dynamique qui activent/désactivent un certain nombre de Pod dans une topologie Fat-tree k_array en fonction du trafic. Nous développons un modèle de file d'attente avec des seuils d'activation qui sont proportionnels à la valeur de k pour activer les serveurs et les Switch des autres Pod.

Mots clés : Consommation d'énergie, Cloud Computing, Centre de données, Files d'attente.

Abstract:

Reducing the energy consumed in the cloud is becoming an increasingly difficult research direction due to the exponential growth of the services that are hosted and delivered. Nowadays, the reduction of energy consumption in data centers has received a lot of attention from researchers, the aim being to increase energy efficiency in the Cloud. Despite these efforts, recent work has shown that the cloud is not always a green technology.

The high energy costs of operating a data center led to a reflection on the efficient operation of a data center. Data centers are designed to support the expected maximum traffic load. As the amount of data transported between customers and providers becomes more important, it would be necessary to find mechanisms to mitigate the energy costs involved. Also depending on the current traffic load, the provider must be able to dynamically adapt the number of resources available.

In this thesis, firstly we proposed an ecological cloud architecture called “Green Cloud Architecture (GCA)”, where we can instantiate virtual routers on physical routers. GCA includes the ability either to migrate virtual routers to other physical nodes, or to power them off, or put them to sleep for energy-saving purposes.

Secondly, in order to reduce energy consumption within data centers, we designed an energy-based resource allocation algorithm that integrates virtual routers on physical routers while seeking to minimize the energy consumed during transport.

Finally, we proposed GREENPOD a Fat-tree based topology which leverages queuing network theory for reducing energy consumption within data center (routers, switches and servers) while guaranteeing an acceptable response time. We considered dynamic management policies that enable/disable several number of PODs (clusters) in a Fat-tree k -ary topology based on traffic load. We designed a queue model with activation thresholds that are proportional to the value of k to activate the servers and switches of other Pods.

Keywords: Energy consumption, Green Cloud, Queuing theory, Dynamic management.

Chapitre 1 : INTRODUCTION GENERALE

1.1. Contexte

La réduction de la consommation d'énergie dans le « *cloud computing* » (Informatique dans les nuages) est actuellement un gros challenge du fait de la croissance exponentielle des services qui y sont hébergés et offerts. Malgré les efforts consentis, plusieurs études ont montré que le Cloud est énergivore. Dans un article paru dans le journal RTL, les auteurs affirment que la pollution numérique pourrait rapidement dépasser celle de l'aéronautique [Sulmont et al 2016]. Cette pollution représentait selon les auteurs 2% des émissions en dioxyde de carbone et devrait doubler en 2019. En 2014, la consommation d'énergie des centres de données américains est estimée à 70 milliards de kWh, soit environ 1,8% de la consommation totale des États-Unis [Brackbill 2016]. Les coûts énergétiques élevés liés à l'exploitation d'un centre de données ont conduit à une réflexion sur leur fonctionnement efficace.

Une architecture réseaux efficace ainsi que des algorithmes d'allocation de ressources basés sur l'économie d'énergie doivent être mis en place. Les centres de données sont conçus pour supporter un maximum de charge de travail, une organisation permettant de faire fonctionner les équipements en cas de besoin est nécessaire.

Dans le but de réduire considérablement le coût énergétique, la tendance est de construire les centres de données dans des zones géographiques où l'accès à l'énergie coûte moins cher ou dans des zones très froides comme les régions arctiques. Cette option a généralement pour conséquence d'augmenter de manière considérable la distance géographique entre les centres de données et leurs utilisateurs. En raison de l'effet des applications Internet, qui sont évidemment basées par exemple sur le web,

sur le pair-à-pair et sur les services de vidéo à la demande sur le web, les quantités de données qui seront transférées sur Internet que ce soit des utilisateurs vers le Cloud ou des Cloud vers les utilisateurs selon les modèles de services Cloud existant, va augmenter de manière significative [Bianzino et al. 2012]. Puisque le trafic Internet ne cesse de se développer, l'équipement nécessaire pour acheminer ce trafic devrait suivre cette tendance et cela entrainera de manière inéluctable une augmentation de la consommation d'énergie.

1.2. Objectifs

L'objectif général de cette thèse est de mettre en place des outils et des mécanismes qui permettent de réduire la consommation d'énergie dans le Cloud. En utilisant des algorithmes d'allocation de ressources et le formalisme des réseaux de file d'attente, nous avons évalué les quantités d'énergie consommées dans le Cloud depuis les utilisateurs jusqu'aux serveurs en passant par les différents types de routeurs. De manière spécifique, il s'agit de :

- Proposer une architecture de Cloud écologique appelée Green Cloud Architecture. Dans cette architecture, nous travaillons avec des routeurs virtuels instanciés sur des routeurs physiques. Pour des raisons d'économie d'énergie, certains routeurs physiques peuvent être éteints et les routeurs virtuels qui y étaient instanciés, migrés vers d'autres.
- Définir un algorithme de réduction de la consommation d'énergie des équipements réseaux lors du transport des données depuis l'utilisateur jusqu'au Cloud.
- Mettre en place une topologie et une organisation du traitement des requêtes dans les centres de données appelée GREENPOD. Cette proposition se base sur la topologie Fat-tree et permet, pour des raisons d'économie d'énergie, d'activer ou de désactiver des groupes de serveurs en fonction du trafic.

1.3. Organisation de la thèse

Le chapitre 2 de la thèse illustre la technologie Cloud Computing ainsi que les différentes topologies des centres de données. Dans le chapitre 3, nous présentons un état de l'art sur les techniques de réduction de la consommation d'énergie dans les centres de données. Nous montrons dans ce chapitre, les différentes causes des pertes d'énergie dans les centres de données. Un ensemble d'approches permettant de réduire la consommation d'énergie est par la suite présenté. Parmi celles-ci, nous avons des approches basées sur les méthodes de virtualisation et de migration de machines virtuelles, des approches permettant de réduire les besoins en énergie du réseau, des mécanismes permettant d'économiser de l'énergie dans les réseaux d'accès et enfin des mécanismes basés sur l'organisation du centre de données.

Les chapitres 4 et 5 présentent nos contributions. Dans le chapitre 4, nous présentons notre architecture de Cloud non énergivore appelé Green Cloud Architecture (GCA) où la virtualisation des routeurs sera le pilier pour le transport des données. Par la suite nous évaluerons notre architecture GCA en concevant un algorithme d'allocation de ressources basé sur l'énergie courante.

Le chapitre 5 présente le modèle que nous avons proposé pour réduire la consommation d'énergie dans les centres de données en utilisant le formalisme des réseaux de file d'attente. Dans ce modèle nous utilisons une topologie Fat-tree avec activation et désactivation des groupes de serveurs dépendant du trafic. Cette gestion dynamique des serveurs dépendra de la taille de la file d'attente. La conclusion générale de la thèse est faite dans le chapitre 6. Le bilan et les perspectives de nos travaux y sont présentés.

Chapitre 2 : ETAT DE L'ART SUR LE CLOUD ET LES TOPOLOGIES DES CENTRES DE DONNEES

2.1 Introduction

Le Cloud Computing ou « informatique dans les nuages » en français représente à la fois un nouveau modèle d'utilisation des ressources de calculs, mais aussi un nouveau modèle économique pour les éditeurs de logiciels. Les fondements du Cloud Computing sont : l'externalisation des services et le paiement à la demande [Le LOUËT 2014]. Jusqu'à récemment, la solution qui prédominait était de disposer de ressources locales avec plusieurs paramètres à gérer. Les entreprises doivent elles-mêmes assurer la maintenance des infrastructures et des logiciels, elles doivent assurer les mises à jour des logiciels, s'occuper du refroidissement des serveurs et en même temps assurer entre autres la sécurité et les sauvegardes des données.

Un centre de données (en anglais data center), est défini par l'encyclopédie informatique Wikipédia comme étant un site physique sur lequel se trouvent regroupés des équipements constituant le système d'information de l'entreprise (ordinateurs centraux, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.). Les centres de données à grande échelle constituent le support d'infrastructure de base pour les services du Cloud en constante expansion. Ainsi, les caractéristiques de performance et de fiabilité des centres de données auront un impact significatif sur l'évolutivité de ces services. En particulier, le réseau du centre de données doit être agile et reconfigurable afin de répondre rapidement aux demandes d'applications et aux besoins de service en constante évolution. Des travaux de recherche importants ont été réalisés sur la conception des topologies de réseau de centres de données afin d'améliorer les performances des centres de données [Niranjan et al. 2009], [Al-Fares et al. 2010], [Greenberg et al. 2009], [Guo et al. 2009].

Bien que dans sa forme actuelle, le Cloud Computing (ou simplement le Cloud) soit un phénomène relativement récent, nous sommes probablement dans un monde où nous utilisons le Cloud sans nous en rendre compte. Si nous utilisons par exemple un fournisseur de messagerie Web comme «Gmail» ou «Hotmail», des appels vidéo avec

«Skype» ou des interfaces vidéo comme «Vimeo» ou «YouTube», et si nous sauvegardons des données sur Internet plutôt que sur un appareil externe, alors nous utilisons le Cloud [HAMZE, Thèse doctorat 2015]. Aujourd'hui, le Cloud est présent partout avec un intérêt croissant grâce à une offre de ressources à la demande. Toutes les grandes entreprises du domaine informatique s'engagent très fortement dans cette nouvelle approche.

Le cabinet américain d'analyse, le Gartner estimait que 43% des Directeurs des Systèmes d'information (DSI) dans le monde s'attendent, dès 2015, que plus de la moitié des transactions de leur organisation passeront par une infrastructure de type Cloud. Leur taux d'adoption devrait s'élever à 74% en 2020. Quant au marché mondial du « cloud » public, toujours selon Gartner, il devrait progresser de 17,9% entre 2011 et 2016 pour atteindre 211,9 milliards de dollars (165 milliards d'euros) [GORELIK, Master 2013].

Dans ce chapitre, nous présentons une historique et une description générale du Cloud avec quelques définitions, leurs caractéristiques, leurs modèles de service et de déploiement. Nous montrons également quelques travaux sur les conceptions et les topologies de réseaux de centres de données publiées récemment.

2.2 Cloud

2.2.1 Historique

Plusieurs dates sont avancées dans la littérature comme marquant le début du Cloud. Mais plusieurs auteurs s'accordent sur le fait que même si on ne les appelait pas comme tel, la technologie Cloud a vu le jour depuis que l'utilisation de services à distance s'est développée. Selon [HESS 2013], l'idée de fournir un service informatique centralisé remonte aux années 1960, lorsque des services informatiques étaient fournis sur un réseau et utilisant la technologie « time-sharing » sur les mainframes. L'auteur indique qu'en 1966, l'ingénieur canadien Douglass Parkhill publiait son livre intitulé « The Challenge of the Computer Utility » [PARKHILL

1966], dans lequel il décrit l'idée d'une informatique comme un utilitaire public avec une installation centralisée à laquelle de nombreux utilisateurs distants se connectent via un réseau.

Dans les années 80, les entreprises ont réalisé que les serveurs basés sur des ordinateurs normales pourraient être installés à moindre coût à la place des mainframes. Progressivement, ces mainframes ont été remplacés par des ordinateurs normales en raison aussi de la baisse des prix des ordinateurs personnels. Dans les années 90, avec le début de la globalisation de l'Internet, la mode qui permet à plusieurs ordinateurs d'accéder à un serveur refait surface. À cette époque des serveurs web avec une grande puissance étaient requis pour répondre aux multiples demandes des clients faites sur Internet. Depuis ce temps, de plus en plus de services sont offerts sur Internet et les utilisateurs de ces services demandent plus de capacité de stockage. De plus en plus, la logique des applications est déplacée vers des serveurs sur Internet à partir des ordinateurs personnels et cela grâce entre autres à l'augmentation de la vitesse d'accès aux réseaux et à la facilité d'accès.

De nos jours, le calcul massif nécessaire peut être réalisé par des prestataires qui se consacrent à cela. Ainsi, plusieurs utilisateurs peuvent partager la même infrastructure, maximisant ainsi l'efficacité de celle-ci et minimisant en même temps les coûts pour les utilisateurs. Cela a conduit à la mise en place de technologies comme les grilles de calcul et le Cloud. De nombreux experts soutiendraient que le Cloud provient des grilles de calcul [GORELIK, Master 2013]. Les objectifs principaux des deux architectures sont de réduire les coûts informatiques, d'accroître la flexibilité et la fiabilité. Il convient de noter que le Cloud fait un pas de plus, en ce sens qu'il s'appuie sur les technologies de virtualisation [Nurmi et al. 2009] [Buyya et al. 2009]. Cependant, bien qu'il existe de nombreuses similitudes entre le Cloud et les grilles de calcul, les méthodologies ne sont pas les mêmes.

Le terme informatique dans les nuages « Cloud Computing » commence à apparaître au 21^{ème} siècle [Mohamad HAMZE, Thèse doctorat 2015]. En effet, certains auteurs pensent que la discipline n'est apparue que dans les années 2000, comme une solution à un problème rencontré par Amazon c'est-à-dire le surdimensionnement de son parc de serveurs hors périodes de fêtes. Amazon avait décidé d'investir sur un groupement de machines, dimensionnées pour absorber les charges importantes des commandes faites sur son site pendant les périodes de fêtes. Ces serveurs étaient inutilisés pendant le reste de l'année et pour les rentabiliser, il décida de les louer à d'autres entreprises à la demande. Par la suite de nombreuses entreprises entrent sur le marché du Cloud telles qu'IBM, Microsoft et Google. Ensuite un certain nombre d'universités s'engagent à grande échelle sur des projets de recherche portant sur le Cloud.

2.2.2 Définition

Il n'existe pas une définition officielle ou formelle du Cloud Computing. Nous rencontrons dans la littérature plusieurs définitions du terme Cloud. Mais le terme Cloud fait référence à un nuage tel que nous avons l'habitude de l'utiliser dans les schémas techniques lorsque nous voulons représenter Internet. Mais mêmes si les experts ne sont pas d'accords sur sa définition exacte, la plupart s'accordent à dire qu'elle inclut la notion de services disponibles à la demande et extensible à volonté. Nous donnons ici quelques définitions parmi celles rencontrées dans la littérature :

- Le journal **Gartner** définit le cloud comme suit: « **Cloud computing is as a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies¹.**»
- **University of Berkeley**, une des plus importantes université technique dans le monde définit le Cloud comme suit: «Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services »[Armbrust et al. 2009].

¹<https://www.gartner.com/it-glossary/cloud-computing>

- L'encyclopédie informatique **Wikipedia** définit le cloud comme étant: « L'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet. Ces serveurs sont loués à la demande, le plus souvent par tranche d'utilisation selon des critères techniques (puissance, bande passante, etc.) mais également au forfait². ».
- La version finale de la définition du Cloud pour le National Institute of Standards and Technology (**NIST**), publiée en Janvier 2017 est : « Le Cloud Computing est un modèle qui permet un accès réseau pratique et sur demande à un pool partagé de ressources informatiques configurables (par exemple, des réseaux, des serveurs, du stockage, des applications et des services) qui peut être rapidement approvisionné et disponible sans trop d'efforts de gestion ou d'interaction d'opérateurs. Ce modèle de cloud favorise la disponibilité et est composé de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement³. ».

²https://fr.wikipedia.org/wiki/Cloud_computing

³<https://cloudhosting.tv/cloud-computing-2/final-version-of-nist-cloud-computing-definition-published.php>

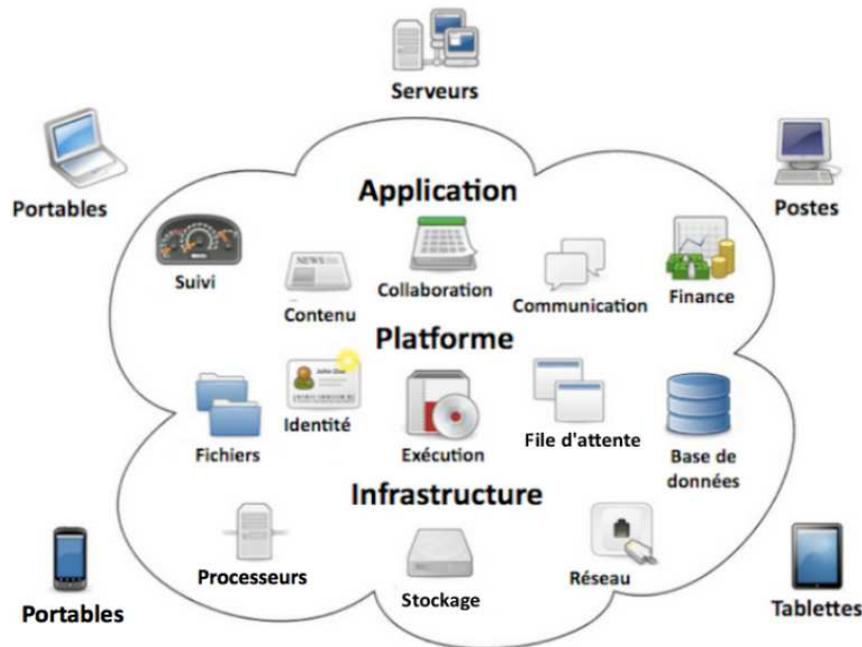


Figure 2.1: Cloud(©Wikipedia)

La figure 2.1 est une illustration du Cloud. Des terminaux hétérogènes peuvent interagir de manière transparente et on y retrouve les trois modèles de service du Cloud.

L'avantage considérable des Cloud par rapport à une infrastructure propre à une entreprise où les serveurs sont très souvent sous-utilisés est sa scalabilité, c'est-à-dire sa capacité à maintenir particulièrement ses fonctionnalités et ses performances en cas de forte demande, mais aussi son extensibilité à la demande et son élasticité. On ne paie que ce qu'on utilise. Le Cloud permet également aux consommateurs d'acquérir de la puissance de calcul sans acheter du matériel ou des logiciels. On se sert de la puissance de calcul mise à notre disposition par des fournisseurs via Internet.

2.2.3 Caractéristiques

Le Cloud possède cinq caractéristiques selon le NIST [Mell&Grance 2011]. Nous allons dans la suite les présenter :

- ***Accès à la demande et en libre-service par l'utilisateur:*** Dans le Cloud, le logiciel est accessible lorsque l'utilisateur le souhaite et il n'y a pas de recours à un administrateur informatique interne ni d'action du côté du fournisseur de service. En outre, pour atteindre des performances élevées en termes d'accessibilité, le Cloud est composé de centres de données situés à plusieurs endroits à travers le monde.
- ***Accès Internet avec une large bande passante :*** Puisque les services demandés sont situés sur des serveurs distants, un accès à Internet est nécessaire. La construction du service peut poser problème lorsque l'application est utilisée depuis un lieu où la couverture réseau n'est pas bonne. Les techniques standardisées utilisées permettent de s'en servir aussi bien avec un ordinateur qu'un téléphone ou une tablette.
- ***Accès à un pool de ressources :*** Les ressources informatiques du fournisseur sont mises en commun pour servir plusieurs consommateurs à l'aide d'un modèle multi-locataires avec différentes ressources physiques et virtuelles assignées de manière dynamique et réaffectées selon la demande du consommateur. Le consommateur n'a généralement aucun contrôle ni de connaissance sur l'emplacement exact des ressources fournies. Les ressources peuvent être des serveurs de stockage, des serveurs de calcul, de la mémoire ou de la bande passante. Cette caractéristique est évidente pour un utilisateur classique du web, mais l'environnement Cloud peut être facilement personnalisé et ajusté en fonction de la demande de l'utilisateur. Par contre, les environnements classiques mutualisés sont généralement statiques et plus difficiles à modifier.
- ***Elasticité :*** L'utilisateur a accès à autant de ressources que nécessaire. Les ressources informatiques sont mobilisables rapidement, autrement dit, l'utilisateur peut ajouter par exemple de l'espace de stockage et d'autres

utilisateurs en moins de temps qu'il ne faut pour commander et installer un ordinateur. De même, l'utilisateur peut aussi décider de diminuer l'espace de stockage ou le nombre d'utilisateurs.

- ***L'utilisation est mesurée et facturée au détail:*** Dans le Cloud, la quantité de service consommée est mesurée à des fins de contrôle, d'adaptation des moyens techniques et de facturation. La facturation est liée à l'usage de paramètres tels que: l'heure d'utilisation, le nombre d'utilisateurs, l'espace disque, les fonctions utilisées etc. L'utilisation des ressources peut être surveillée, contrôlée, et signalée assurant ainsi la transparence tant au niveau du fournisseur que du consommateur.

La figure 2.2 montre la différence entre le modèle classique et le modèle du Cloud sur l'utilisation des ressources en fonction du temps. Le modèle Cloud offre l'avantage de payer une capacité (RAM, processeur, disque) qu'en fonction de la demande. Ce qui n'est pas le cas pour le modèle classique. Dans ce dernier cas, les entreprises sont obligées d'investir sur une capacité maximale, mais peuvent se rendre compte dans le temps que celle-ci n'est pas toujours nécessaire.

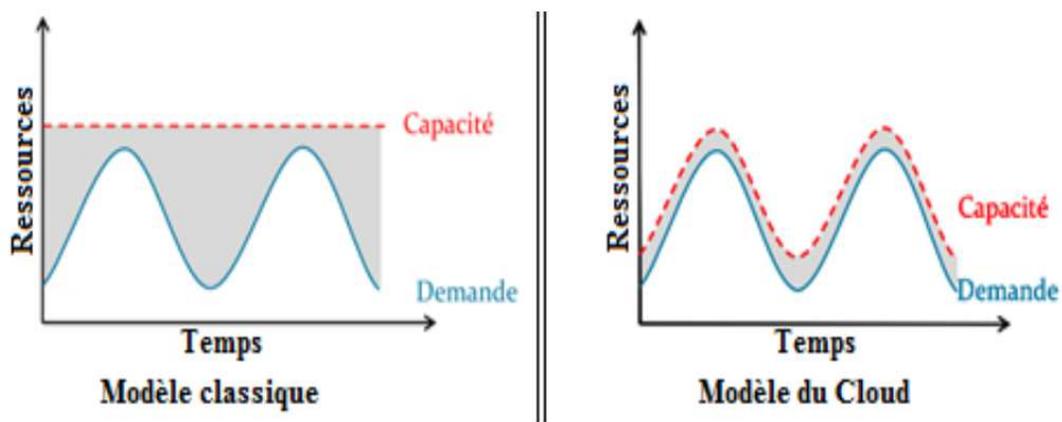


Figure 2.2: Différence entre le modèle de paiement classique et celui du Cloud[HAMZE, Thèse doctorat 2015]

2.2.4 Architecture

Le Cloud Computing est basé sur l'offre de service. Nous pouvons noter principalement trois couches dans le modèle de service du Cloud. La figure 2.3 présente ces différentes couches de la moins visible aux utilisateurs finaux à la plus visible.

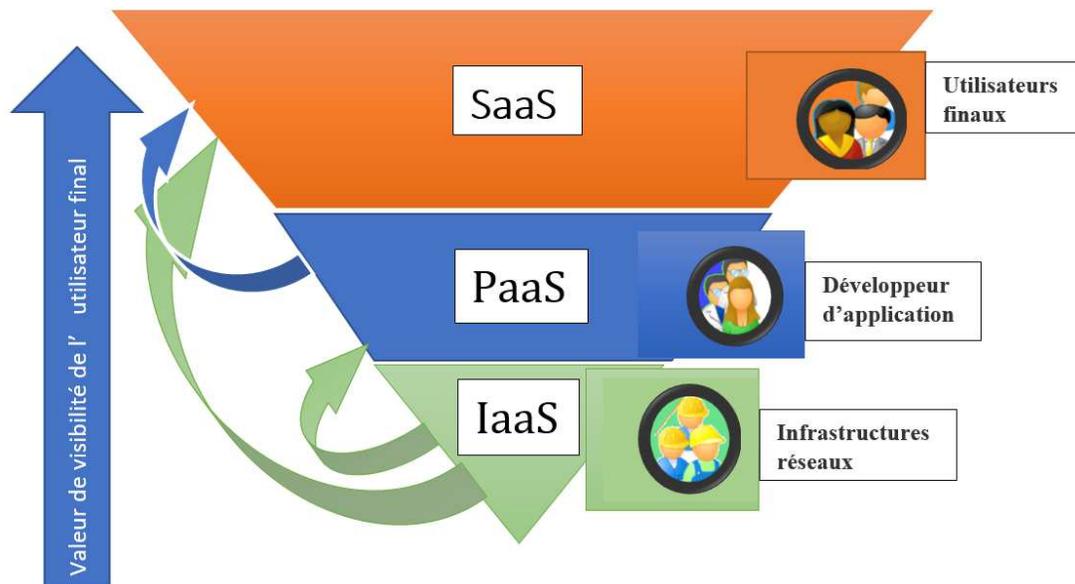


Figure 2.3: Les couches du Cloud

L'Infrastructure as a Service (IaaS) est gérée par les architectures réseaux. La couche Platform as a Service (PaaS) est destinée aux développeurs d'applications et la couche Software as a Service (SaaS) est le produit final pour les utilisateurs. La figure 2.4 constitue une abstraction du Cloud et montre les différentes interactions entre ses acteurs et ses composants. Le fournisseur de services « Vendor » peut fournir tout type de services. Les développeurs d'applications « developers » utilisent les ressources fournies au niveau des couches IaaS et PaaS pour construire des services pour les utilisateurs finaux « End Users ».

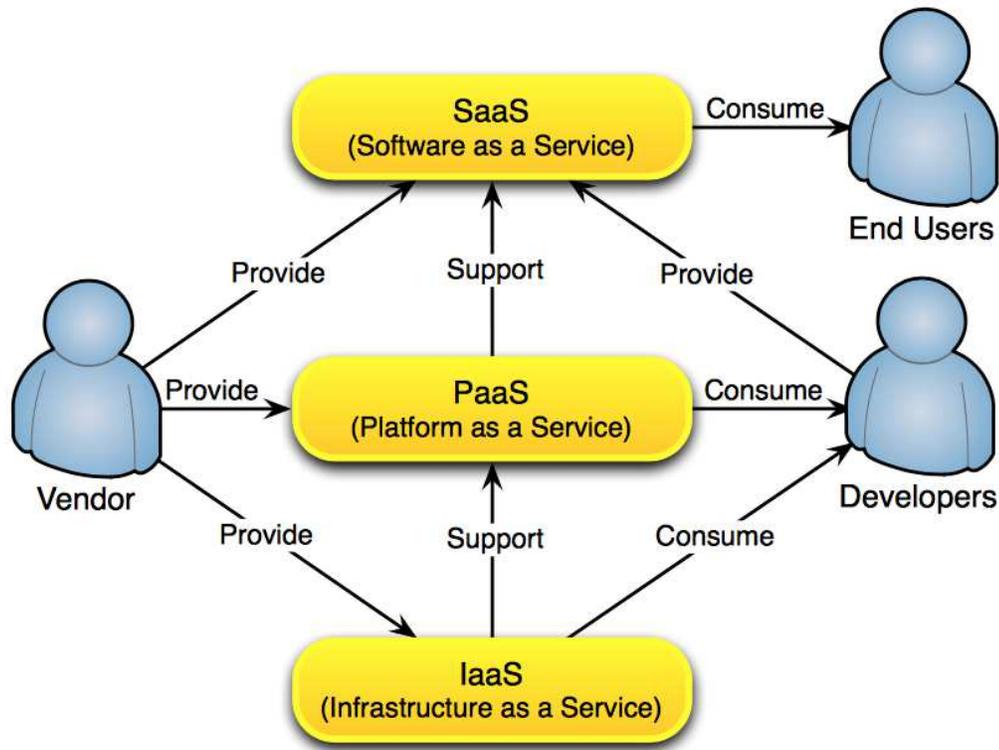


Figure 2.4: Abstraction du Cloud Computing[Marinos 2009]

2.2.4.1 Software as a Service (SaaS)

Le logiciel en tant que service, plus connu en anglais sous l'acronyme « SaaS » repose sur la mise à la disposition des utilisateurs plusieurs applications à travers Internet. C'est le type de service le plus utilisé dans le monde [Grevet 2009]. Dans ce modèle de déploiement, le fournisseur loue des applications clé en main à ses clients en tant que service à la demande au lieu de les facturer des licences. Le client ne se préoccupe pas de la mise à jour et de la maintenance des logiciels et serveurs et cela permet de réduire un certain coût relatif aux logiciels et aux serveurs. C'est certes un avantage, mais cela rend le client dépendant du mode de fonctionnement de l'application et mis à part quelques réglages, le client ne peut pas personnaliser à sa guise l'application. Comme exemple d'applications de type SaaS, nous pouvons citer

l'outil de traitement de texte collaboratif de Google, **Google Docs**⁴ et le logiciel de retouche d'image en ligne **Adobe Photoshop Online**⁵.

2.2.4.2 Platform as a Service (PaaS)

Le Cloud de type plateforme en tant que service « PaaS » met à la disposition des entreprises un environnement d'exécution rapidement disponible. Ce modèle de service est destiné principalement aux entreprises. Il est situé juste en-dessous du SaaS dans l'architecture puisqu'il permet de fournir aux développeurs des ressources pour leur environnement d'exécution. La gestion des applications est du ressort des entreprises, qui peuvent elles-mêmes s'occuper de leurs installations et configurations. Le maintien de la plateforme d'exécution de ces applications est de la responsabilité du fournisseur de Cloud. Le fournisseur met à la disposition du client le matériel des serveurs, les logiciels de base tels que le système d'exploitation et l'infrastructure de connexion, de stockage et de sauvegarde. C'est une architecture sans téléchargement ou installation de logiciels pour les développeurs, responsables informatiques ou utilisateurs finaux. Elle est également connue sous le nom de «*cloudware*» [Grevet 2009]. Comme exemple de plateforme de type PaaS, nous pouvons citer **Google App Engine**⁶, qui est une plateforme de conception et d'hébergement d'applications web et **Azure Services Platform**⁷, qui est la plateforme applicative en nuage de Microsoft.

2.2.4.3 Infrastructure as a Service (IaaS)

Le modèle de services Infrastructure en tant que service « IaaS » est situé juste en-dessous du PaaS, puisqu'il permet de créer un pool de ressources virtuelles en divisant les ressources physiques à l'aide des technologies de virtualisation [HAMZE, Thèse doctorat 2015]. Cette infrastructure fournit des capacités de calcul, de stockage

⁴<http://docs.google.com>

⁵<http://www.photoshop.com/express>

⁶<http://appengine.google.com>

⁷<http://www.microsoft.com/azure>

et une bande passante suffisante. Plutôt que d'acheter des serveurs, des logiciels, de l'espace dans un centre de traitement de données et/ou de l'équipement réseau, les clients n'ont plus qu'à louer ces ressources auprès de prestataires de services complètement externalisés et directement compétents dans le domaine [Grevet 2009]. La tarification est faite en fonction de l'utilisation et de la quantité de ressources consommées. Ce type d'hébergement présente des avantages par rapport aux hébergements classiques comme l'hébergement mutualisé et l'hébergement dédié.

Dans le cas de l'hébergement mutualisé, une même machine est partagée entre plusieurs clients, gérée par un prestataire de service et dont les clients paient le même prix peu importe leur utilisation.

Pour l'hébergement dédié, nous avons une machine par client, gérée le plus souvent par le client lui-même et pour laquelle le client paie le même prix chaque mois peu importe son utilisation.

Nous pouvons citer comme exemple d'infrastructure de type IaaS le *GoGrid*⁸, qui héberge des machines virtuelles Linux et Windows gérées par un panneau de contrôle multi-serveurs et une API RESTful et *Amazon ElasticCompute Cloud* ou *EC2*⁹, qui est un service proposé par Amazone permettant à des tiers de louer des serveurs sur lesquels exécuter leurs propres applications web.

Même si ces trois acronymes (SaaS, PaaS et IaaS) constituent les trois composants de base dans l'architecture Cloud, nous avons rencontré dans la littérature plusieurs acronymes se terminant par *aaS* parlant des Cloud computing.

Il y a par exemple le « *Network as a Service* » (*NaaS*), considérant le réseau comme un service et permet d'offrir l'ensemble de ses ressources en tant que service. Mais nous pouvons considérer ce type de service comme une partie de l'IaaS. Il y a aussi le

⁸<http://www.gogrid.com>

⁹<http://aws.amazon.com>

« *CaaS* », défini comme « *Communication as a Service* » et qui consiste à fournir des moyens de communication en tant que service via Internet. Comme exemple de service de type CaaS, on peut par exemple noter Skype, Windows Live Messenger, Whatsapp, Viber etc. En réalité, il ne s'agit pas d'une nouveauté, mais consiste ici à mettre des noms sur des services existants. Le modèle de service « *Data storage as a Service* » (*DaaS*) est aussi développé et il permet aux utilisateurs de stocker leurs données sur des disques distants et d'y accéder à tout moment à partir de n'importe quel endroit. Ce modèle de service peut être considéré comme un sous-groupe du modèle de service IaaS.

2.2.5 Modèle de déploiement

Au fur et à mesure que le Cloud s'est développé et a gagné en popularité, plusieurs modèles et stratégies de déploiement sont apparus pour répondre aux besoins spécifiques des différents utilisateurs. Les architectures de Cloud peuvent être publiques, privés, hybride, communautaires ou privés virtuels [Mell&Grance2009][Armbrust et al. 2009][Zhang et al. 2010]. La figure 2.5 offre une vue d'ensemble du modèle visuel de la définition du Cloud à partir des travaux du NIST [Young-Chan 2012]. Cette figure résume les différents modèles de déploiement du Cloud (Couche inférieure), les couches de son architecture (couche du milieu), ainsi que ses caractéristiques (couche supérieure) selon sa définition du NIST.

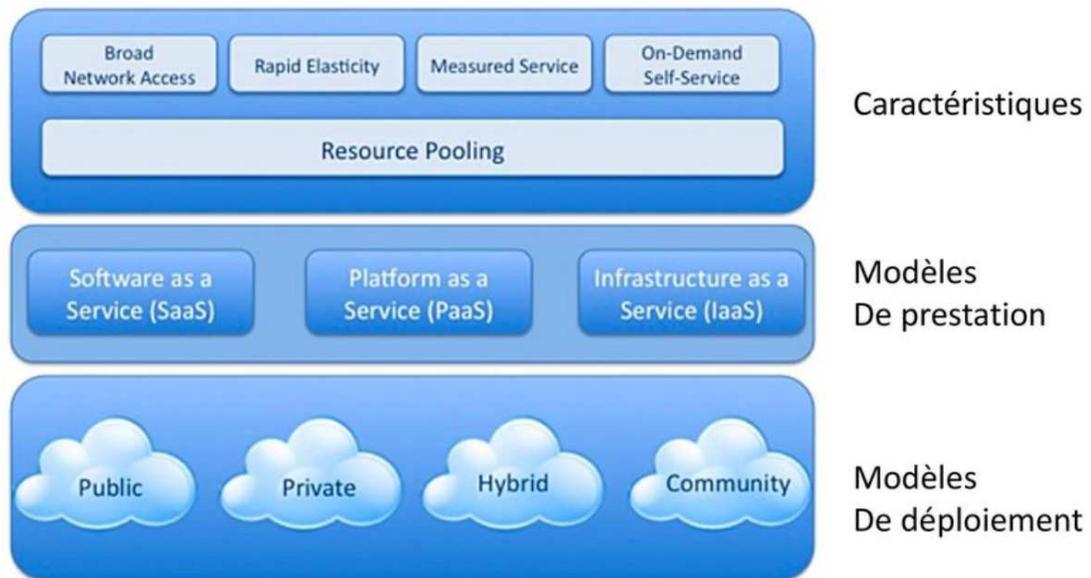


Figure 2.5: Une vue technique du Cloud [Young-Chan 2012]

2.2.5.1 Cloud Public

Le modèle de Cloud public est hébergé sur Internet et les fournisseurs de services offrent leurs ressources en tant que services au grand public. Les services du Cloud appelés également modèle de livraison « Delivery Models », comme illustré sur la figure 2.5 tels que le SaaS, le PaaS et l'IaaS, sont offerts au public. Les utilisateurs du Cloud public tels que les entreprises peuvent éliminer les coûts initiaux associés à la construction des centres de données ainsi qu'à l'achat et à la maintenance des serveurs. Le Cloud public ne veut pas dire que vos données sont accessibles à n'importe qui. Toutefois, les données sont hébergées sur une multitude de serveurs qui sont accessibles par un nombre déterminé d'utilisateurs. Cela peut conduire à un manque de contrôle sur les données, le réseau et sur les paramètres de sécurité et ainsi entraver l'efficacité de ce modèle de déploiement. De nos jours, il existe des préoccupations au sujet de la sécurité et des questions législatives sur les données hébergées dans le Cloud public. Mais un des avantages du Cloud public est que les dépenses initiales sont pratiquement réduites à néant puisqu'on vient se greffer à une

infrastructure déjà existante et en règle générale, on paie en fonction de l'utilisation de nos données.

2.2.5.2 Cloud Privé

Il existe deux types de Cloud privés: le Cloud privé interne géré par l'entreprise elle-même et le Cloud privé externe. En général, lorsqu'on parle de Cloud privé, on fait référence aux Cloud privés externes dédiés à une seule entreprise, mais dont la gestion est externalisée vers un prestataire (voir section 2.2.2). Dans ce type de Cloud, seule l'entreprise et ses collaborateurs peuvent accéder aux ressources. Cependant, il est souvent critiqué car il est similaire aux serveurs propriétaires traditionnels qui ne fournissent pas les avantages du Cloud comme l'absence de coûts d'investissement élevés [HAMZE, Thèse doctorat 2015]. Il est aussi moins réactif en cas de montée en charges et il est plus coûteux à mettre en place. Son avantage est qu'on a un contrôle beaucoup plus complet sur ses données et sur son infrastructure.

2.2.5.3 Cloud Communautaire

C'est un concept qui aspire à remplacer le Cloud de fournisseurs en combinant les ressources sous-utilisées des machines des utilisateurs pour former un Cloud communautaire [Marinos 2009]. C'est un modèle de déploiement qui offre à la fois les avantages d'un Cloud public du point de vue de la facturation, mais aussi les avantages d'un Cloud privé en termes de confidentialité et de sécurité de façon générale. L'idée principale est de supprimer la dépendance vis-à-vis des fournisseurs de Cloud et de mettre en place une structure sociale qui sera partagée par plusieurs organisations qui ont leurs propres préoccupations.

2.2.5.4 Cloud Hybride

Le Cloud hybride est une combinaison du modèle de déploiement des précédents types de Cloud en essayant de résoudre les problèmes des Cloud publics et privés. En fait, avec le Cloud hybride, certaines données, applications et services peuvent être hébergés en local, tandis qu'une partie restante s'exécute sur un Cloud public. Les

avantages de ce modèle sont qu'une entreprise donnée peut construire un Cloud privé pour garder ses services et ses données critiques, mais utilise également un Cloud public pour les applications peu sensibles, les informations et traitements non critiques.

2.2.5.5 Cloud Privé Virtuel

Le Cloud privé virtuel peut être vu comme des services de Cloud qui simulent des expériences de Cloud privé sur une infrastructure de Cloud public. Le Cloud privé virtuel est essentiellement une plateforme qui tourne sur un Cloud public en utilisant la technologie VPN (Virtual Private Network). C'est un « Cloud dans le Cloud » et il est dit privé car il est entièrement dédié au client, c'est-à-dire complètement isolé sur le plan logique. Le Cloud privé virtuel devient ainsi une extension du réseau interne du client. Il bénéficie du même mode de facturation qu'un Cloud public. Il permet également aux entreprises qui n'ont pas les moyens techniques, humains ou financiers d'administrer elles-mêmes leurs propres centres de données de disposer d'un Cloud privé. Une des raisons qui justifie également l'utilisation de ce type de Cloud est la sécurité liée à l'utilisation du Cloud public, mais aussi la complexité liée aux Cloud hybrides.

2.2.6 Outils d'implémentation et de simulation

Il existe sur le marché quelques outils d'implémentation et de simulation du Cloud. Ils sont généralement axés sur des expériences et une modélisation simplifiée du Cloud. Ils peuvent permettre d'adapter le système avant de le déployer sur un vrai Cloud et de l'expérimenter avec différentes charges de travail et scénarios de performances. Les tableaux 2.1 et 2.2 présentent respectivement quelques-uns des outils d'implémentation et des logiciels de simulation.

Ils donnent pour chacun d'eux, le modèle de service, le modèle de déploiement ainsi que leur but.

Nom	Modèle de service	Modèle de déploiement	Open Source	But
OpenNebula ¹⁰	IaaS	Cloud privé	Oui	Organiser le fonctionnement d'un ensemble de serveurs physiques. Fournir une boîte à outils complète permettant de gérer de façon centralisée une infrastructure virtuelle hétérogène.
Nimbus ¹¹	IaaS	Cloud public	Oui	Fournir une infrastructure en tant que service dans le Cloud à ses clients. Fournir la puissance et la polyvalence des infrastructures Cloud aux utilisateurs scientifiques.
OpenStack ¹²	Paas, IaaS, NaaS	Cloud public, privé ou hybride	Oui	Déployer des infrastructures de Cloud et contrôler les différentes ressources des machines virtuelles, de stockage ou encore de réseau dans un centre de données.
Apache CloudStack ¹³	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Gérer et déployer de grands réseaux de machines virtuelles avec une interface web conviviale et/ou des outils de commandes en ligne.

Tableau 2.1: Exemple d'outils d'implémentation de Cloud

¹⁰<http://opennebula.org>

¹¹<http://www.nimbusproject.org>

¹²<https://www.openstack.org>

¹³<https://cloudstack.apache.org>

Nom	Modèle de service	Modèle de déploiement	Open Source	But
CloudSim ¹⁴	PaaS, SaaS, IaaS, NaaS	Cloud public, privé ou hybride	Oui	Modéliser et simuler l'environnement du Cloud Computing et ses services. Il est réalisé en Java.
GreenCloud ¹⁵	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Simuler des centres de données du Cloud pour évaluer leurs coûts d'exploitation en termes d'énergie. Il s'agit d'une extension du simulateur NS2.
iCanCloud ¹⁶	IaaS	Cloud public, privé ou hybride	Oui	Prédire les compromis entre le coût et la performance d'un ensemble donné d'applications exécutées dans un matériel spécifique, puis fournir aux utilisateurs des informations utiles sur ces coûts.
RealCloudSim ¹⁷	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Allocation de machines virtuelles basée sur CloudSim et fourniture d'une interface graphique pour lire les topologies de réseaux basées sur le format BRITE.

Tableau 2.2: Logiciels de simulation de Cloud

¹⁴<http://www.cloudbus.org/cloudsim>

¹⁵<https://greencloud.gforge.uni.lu>

¹⁶<http://www.arcos.inf.uc3m.es/~icancloud/Home.html>

¹⁷<https://sourceforge.net/projects/realcloudsim>

2.3 Topologies des centres de données

2.3.1 Historique

La conception de l'infrastructure des centres de données a récemment suscité un vif intérêt dans le milieu universitaire et de l'industrie, notamment en raison de la croissance des centres de données pour supporter les applications Internet en croissance rapide, comme la recherche (Google, Bing), l'hébergement et la distribution de contenu vidéo (par exemple, YouTube, NetFlix), les réseaux sociaux (par exemple Facebook, Twitter) et les calculs à grande échelle (par exemple exploration de données, bioinformatique, indexation). Par exemple, les services en ligne Microsoft Live sont pris en charge par un centre de données basé à Chicago, qui est l'un des plus grands centres de données jamais créés, couvrant plus de 700 000 pieds carrés. En particulier, le Cloud Computing est caractérisé comme l'aboutissement de l'intégration des infrastructures informatiques et de données pour fournir une approche évolutive, agile et rentable. Tout cet investissement est fait pour répondre aux besoins informatiques critiques en constante augmentation (en termes de calcul, de stockage et d'applications) dans les entreprises [Fox et al. 2009] [Greenberg et al. 2009].

Les centres de données massifs fournissant le stockage forment le noyau de l'infrastructure pour le Cloud [Greenberg et al. 2009]. Il est donc impératif que l'infrastructure du centre de données, y compris la mise en réseau des centres de données, soient bien conçues pour que le déploiement et la maintenance de l'infrastructure soient rentables. Avec la disponibilité des données et la sécurité en jeu, le rôle du centre de données est plus critique que jamais.

La topologie du réseau interconnectant les serveurs a un impact significatif sur l'agilité et la reconfigurabilité de l'infrastructure du centre de données afin de répondre aux demandes de changement d'applications et aux exigences de service. Aujourd'hui, les réseaux des centres de données utilisent principalement des

commutateurs Top of Rack (ToR) interconnectés par des commutateurs de End of Row (EoR), eux-mêmes connectés via des commutateurs centraux. Avec ToR, un Switch d'accès est mis dans chaque baie de serveurs pour offrir une connectivité nécessaire à tous les serveurs qui y sont rackés. Le but est de garder le plus gros du câblage à l'intérieur de la baie. Tandis qu'avec le EoR, aucun Switch n'est mis dans les baies serveurs. Un ou plusieurs Switch assurent la connectivité de tous les serveurs de toutes les baies.

Cette approche conduit à une surabondance de bande passante significative sur les liens dans le noyau du réseau [Al-Fares et al. 2008], et a incité plusieurs chercheurs à suggérer d'autres approches pour des architectures de réseau économiques et évolutives. Selon la reconfigurabilité de la topologie après le déploiement du centre de données, il existe des architectures fixes et des architectures flexibles. Les architectures fixes peuvent être classées en catégories : les architectures arborescentes telles que fat-tree [Al-Fares et al. 2008] et Clos Network [Hamilton et al. 2009], et les topologies récursives comme DCell [Guo et al. 2008], BCube [Guo et al. 2009]. Les architectures flexibles telles que cThrough [Wang et al. 2010], Helios [Farrington et al. 2010] et OSA [Chen et al. 2012] permettent la reconfigurabilité de la topologie du réseau.

2.3.2 Topologies des réseaux de centre de données

Dans cette section, nous présentons les différentes topologies des centres de données. Compte tenu du coût et de la disponibilité, Ethernet haut débit (1 Gbps, 10 Gbps) est le plus souvent utilisé dans la construction de réseau interne de centres de données [Kant 2009]. Cependant, bien que la plupart des centres de données utilisent des commutateurs Ethernet pour interconnecter les serveurs, il existe encore de nombreuses façons de mettre en œuvre les interconnexions, ce qui conduit à différentes topologies de réseau de centre de données. Chacune de ces différentes topologies est caractérisée par différentes exigences de ressources, visant à apporter

des améliorations à la performance des centres de données. Si les serveurs et les commutateurs sont considérés comme des sommets, les fils comme des arêtes, la topologie de chaque réseau de centre de données peut être représentée sous forme de graphe.

Toutes les topologies abordées dans cette section peuvent être classées en deux catégories: fixe et flexible. Si la topologie du réseau ne peut pas être modifiée après le déploiement du réseau, nous la classons comme une architecture fixe; sinon c'est une architecture flexible.

La figure 2.6 représente la taxonomie des différents réseaux de centre de données.

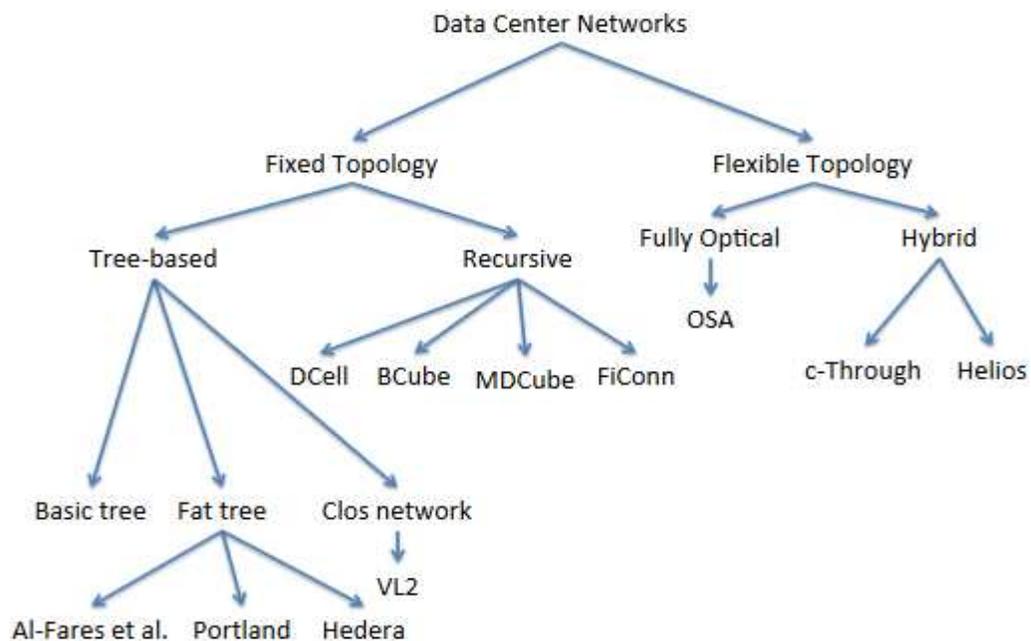


Figure 2.6: Taxonomie des topologies de centre de données [Yang et al. 2013]

Ci-dessous les notations utilisées dans la suite.

n : nombre de ports d'un commutateur dans une architecture.

k : nombre de ports d'un serveur dans une architecture.

N : nombre total de serveurs dans un centre de données.

Il est à noter que n et k peuvent varier en fonction de la position du nœud.

2.3.2.1 Architecture fixe

a. Topologie tree-based

L'architecture arborescente standard et leurs variantes sont largement utilisées dans la conception de réseaux de centres de données [Al-Fares et al. 2008], [Niranjan et al. 2009], [Al-Fares et al. 2010].

❖ Topologie Basic-tree

Les centres de données reposent généralement sur l'architecture hiérarchique à trois niveaux basée sur une structure arborescente, illustrée à la figure 2.7.

Dans une topologie à trois niveaux, il existe un niveau cœur qui est la racine de l'arbre, un niveau d'agrégation qui est au milieu. Un groupe d'hôtes (principalement 16/32) est connecté à un commutateur, appelé commutateur ToR (Top of Rack) créant ainsi la couche d'accès à la topologie.

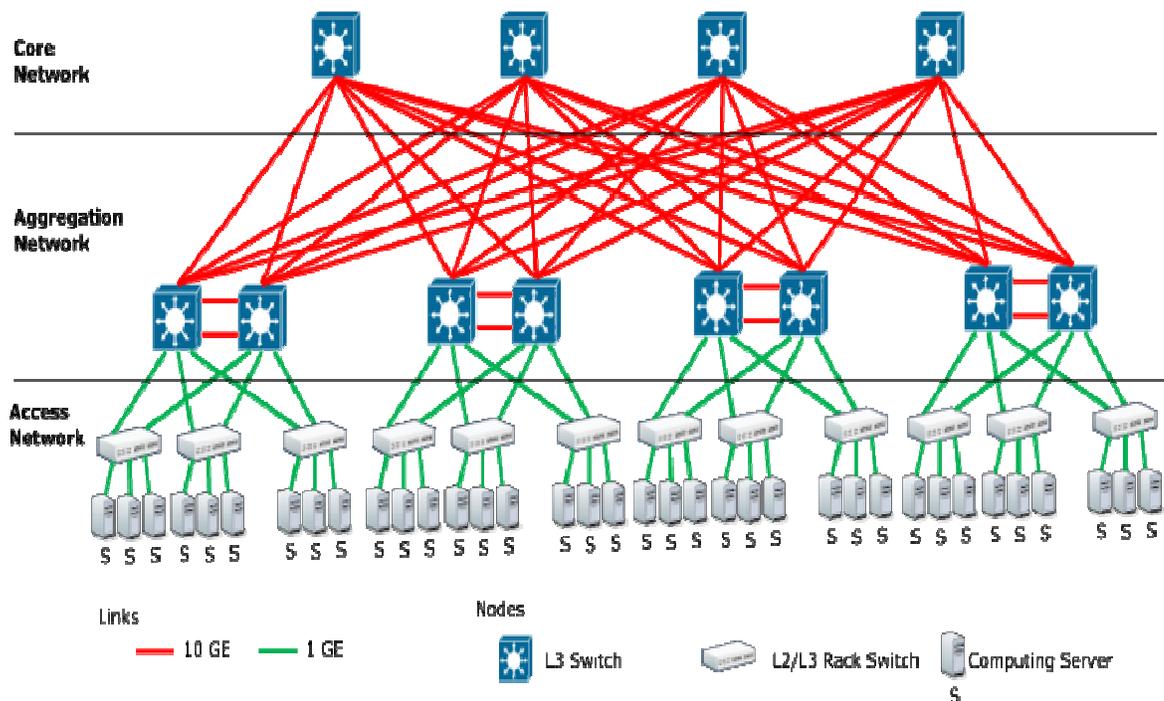


Figure 2.7: Topologie arborescente à trois niveaux [Kliazovich et al. 2010]

Dans une topologie basic-Tree, les commutateurs de niveau supérieur doivent prendre en charge la communication de données entre un grand nombre de serveurs.

Des commutateurs offrant des performances et une fiabilité supérieure sont donc nécessaires dans ces niveaux. Le nombre de serveurs dans une architecture arborescente est limité par le nombre de ports sur les commutateurs.

❖ Topologie Fat-tree

Fat-tree est une version étendue d'une topologie arborescente [Leiserson 1985]. Fat-tree a été utilisé comme topologie pour les centres de données par plusieurs chercheurs [Al-Fares 2008], [Niranjan et al 2009]. C'est un réseau basé sur un arbre binaire complet.

Parmi les n -ports, le commutateur au niveau edge est connecté à $\frac{n}{2}$ serveurs et à $\frac{n}{2}$ commutateurs au niveau agrégation.

Les $\frac{n}{2}$ commutateurs niveau agrégation, les $\frac{n}{2}$ commutateurs niveau edge et les serveurs connectés aux commutateurs niveau edge forment ce qu'on appelle un Pod.

Au niveau cœur, il ya $(\frac{n}{2})^2$ commutateurs, chacun est connecté à chaque Pod parmi les " n " Pod.

La figure 2.8 présente une topologie Fat-Tree avec $n=4$. Contrairement à la topologie basic Tree, tous les trois niveaux utilisent le même type de commutateur. Les commutateurs de haute performance ne sont pas nécessaires dans les niveaux cœur et agrégation. Le nombre maximum de serveurs dans une topologie Fat-Tree à n -ports est $\frac{n^3}{4}$ [Al-Fares 2008].

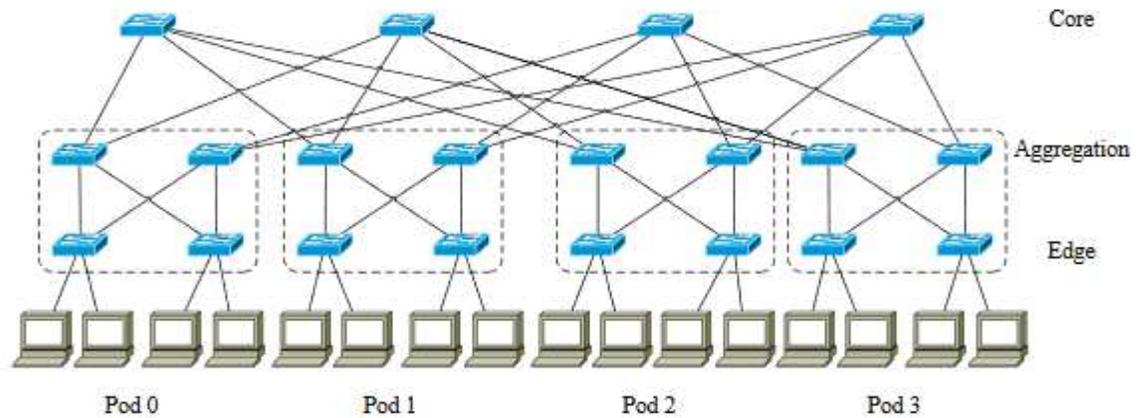


Figure 2.8: Topologie Fat Tree [Yang et al. 2013]

Facebook [Andreyev 2014] a aussi déployé une version de la topologie Fat-Tree afin d'obtenir une bande passante de bisection élevée, un déploiement réseau rapide et une évolutivité des performances pour rester en phase avec la nature agile des applications exécutées dans les centres de données.

Il se compose de Pods, une unité de réseau standard comme le montre la figure 2.10. La bande passante de liaison ascendante de chaque TOR est 4 fois ($4 * 40G = 16 * 10G$) de la bande passante descendante pour chaque serveur connecté.

Pour mettre en œuvre la connectivité à l'échelle du bâtiment, il a créé quatre «plans» indépendants de commutateurs (commutateur de niveau 3), chacun pouvant être mis à l'échelle jusqu'à 48 périphériques indépendants dans un plan. Border Gateway Protocol (BGP4) est utilisé comme un protocole de contrôle pour le routage tandis qu'un contrôleur centralisé est déployé pour pouvoir remplacer les chemins de routage chaque fois que nécessaire, en adoptant une approche de «distributed control, centralized override». Afin d'utiliser tous les chemins disponibles entre deux hôtes, le routage ECMP (EqualCost Multiple Path) avec le hachage basé sur le flux est implémenté.

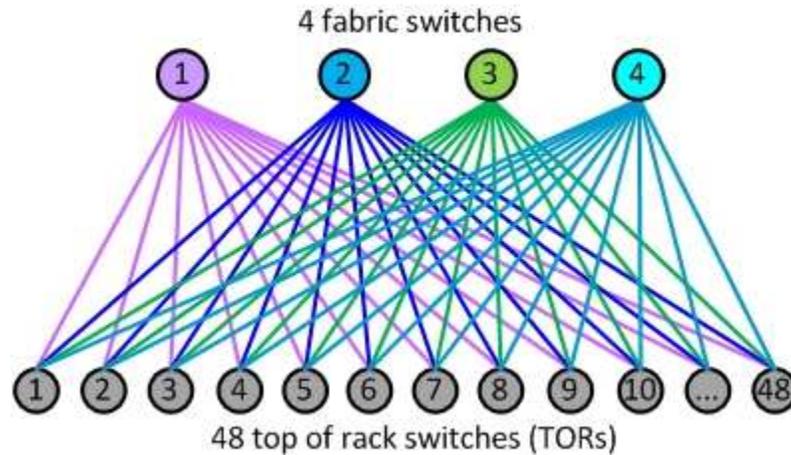


Figure 2.9: La topologie Fat-tree de Facebook [Andreyev 2014]

❖ **Topologie Clos Network :**

Greenberg et al. ont proposé une architecture de réseau de centre de données utilisant une topologie Clos Network dans [Greenberg et al. 2009]. Un Clos Network est également un arbre à plusieurs racines. Un Clos Network se compose généralement de trois niveaux de commutateurs : les commutateurs ToR directement connectés aux serveurs, les commutateurs d'agrégation connectés aux commutateurs ToR et les commutateurs intermédiaires connectés aux commutateurs d'agrégation. Ces trois niveaux de commutateurs sont appelés commutateur "Input", "middle", et "output" dans le Clos original [Dally & Towles 2004]. Le nombre de commutateurs est déterminé par le nombre de ports sur les commutateurs intermédiaires et commutateurs d'agrégation. Si chacun de ces commutateurs a " n " ports, il y aura " n " commutateurs d'agrégation et $\frac{n}{2}$ commutateurs intermédiaire. Il y a exactement un lien entre chaque commutateur intermédiaire et chaque commutateur d'agrégation. Les $\frac{n}{2}$ ports restant sur chaque commutateur d'agrégation sont connecté à $\frac{n}{2}$ commutateurs ToR. Chacun des commutateurs ToR est connecté à $\frac{n}{2}$ différents commutateurs d'agrégation et les ports restants sur les commutateurs ToR sont connectés aux serveurs.

Il y a $\frac{n^2}{4}$ commutateurs ToR. Pendant que les commutateurs intermédiaires et les commutateurs d'agrégation doivent avoir le même nombre de ports, le nombre de ports d'un commutateur ToR n'est pas limité. Si " n_{ToR} " ports de chaque commutateur ToR sont connectés aux serveurs, il y aura $\frac{n^2}{4} * n_{ToR}$ serveurs dans le réseau. La figure 2.10 montre un Clos Network avec $n=4$ et " $n_{ToR} = 2$ ".

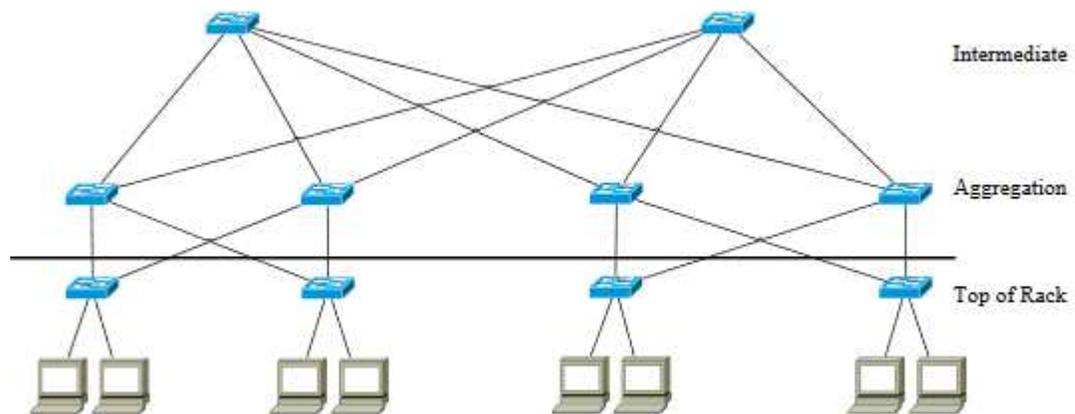


Figure 2.10: La topologie Clos Network [Yang et al. 2013]

b. Topologies récursives

Les topologies arborescentes peuvent être mises à l'échelle en insérant plusieurs niveaux de commutateurs, tandis que chaque serveur est connecté à un seul des commutateurs de niveau inférieur. Les topologies récursives ont toujours des niveaux comme dans les topologies arborescentes. Cependant, les topologies récursives utilisent des structures de niveau inférieur comme des cellules pour construire des structures de niveau supérieur, et les serveurs dans des topologies récursives peuvent être connectés à des commutateurs de différents niveaux ou même d'autres serveurs. Il existe plusieurs ports réseau sur les serveurs de topologies récursives, ce qui les rend significativement différents des topologies arborescentes. Les graphes, plutôt que les arbres multi-racines, sont adaptés pour représenter des architectures

récurrentes. Certaines topologies récurrentes représentatives seront abordées dans les sections suivantes.

❖ **Topologie DCell:**

L'élément le plus fondamental d'un DCell, appelé $DCell_0$, consiste à avoir n serveurs et un commutateur à n -ports. Chaque serveur dans un $DCell_0$ est connecté au commutateur dans le même $DCell_0$. Soit $DCell_k$ un DCell de niveau- k . La première étape consiste à construire un $DCell_1$ à partir de plusieurs $DCell_0$. Chaque $DCell_1$ a $n + 1$ $DCell_0$, et chaque serveur de chaque $DCell_0$ dans un $DCell_1$ est connecté à un serveur dans un autre $DCell_0$, respectivement. En conséquence, les $DCell_0$ sont connectés les uns aux autres, avec exactement un lien entre chaque paire de $DCell_0$. Une procédure similaire est utilisée pour construire un $DCell_k$ à partir de plusieurs $DCell_{k-1}$. Dans un $DCell_k$, chaque serveur aura finalement $k + 1$ liens : le premier lien ou le lien de niveau 0 connecté à un commutateur lors de la formation d'un $DCell_0$, et un lien de niveau i connecté à un serveur dans le même $DCell_i$ mais à un $DCell_{i-1}$ différent.

Supposons que chaque $DCell_{k-1}$ ait des serveurs t_{k-1} , alors un $DCell_k$ sera constitué de t_k $DCell_{k-1}$, et par conséquent de $t_{k-1} \times t_k$.

Évidemment, nous avons $t_k = t_{k-1} \times (t_{k-1} + 1)$. La figure 2.11 montre un $DCell_1$ avec $n = 4$. On peut voir que le nombre de serveurs dans un DCell croît de façon exponentielle, et le nombre total de niveaux dans un DCell est limité par le nombre de carte d'interface réseau sur les serveurs qu'il contient. DCell peut évoluer vers un très grand nombre de serveurs en utilisant des commutateurs et des serveurs avec très peu de ports. Par exemple, lorsque $n = 6$, $k = 3$, un DCell entièrement construit peut comprendre plus de 3 millions de serveurs [Guo et al. 2008].

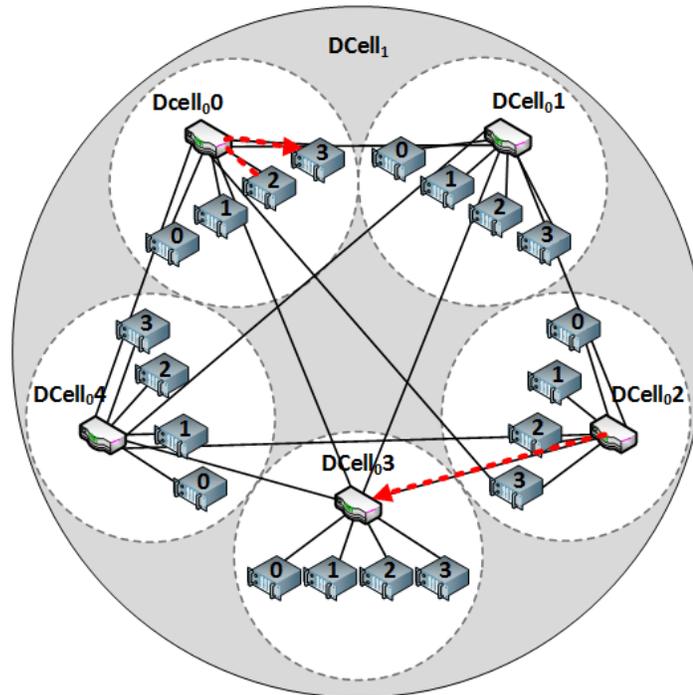


Figure 2.11: Construire un DCell1 à partir de DCell0s avec $n = 4$ [Chirgwin 2015]

❖ Topologie Bcube

BCube [Guo et al. 2009] est une topologie récursive spécialement conçue pour les centres de données modulaires embarqués. La construction d'un cluster de centre de données dans un conteneur d'expédition de 20 ou 40 pieds le rend très portable. Comme les demandes changent dans différents centres de données, le cluster entier peut être facilement déplacé. Bien que le temps de déploiement soit considérablement plus court, l'inconvénient de cet environnement est qu'en raison des contraintes opérationnelles et d'espace, une fois déployé sur le terrain, il est difficile de desservir le cluster. L'élément le plus fondamental d'un BCube, nommé BCube₀, est également le même que celui d'un DCell₀: n serveurs connectés à un commutateur n -ports.

La principale différence entre BCube et DCell réside dans la façon dont ils évoluent. BCube utilise plus de commutateurs lors de la construction d'architectures de plus

haut niveau. Lors de la construction d'un $BCube_1$, " n " commutateurs supplémentaires sont utilisés, se connectant à exactement un serveur dans chaque $BCube_0$. Par conséquent, un $BCube_1$ contient " n " $BCube_0$ s et " n " commutateurs supplémentaires (Si les commutateurs des $BCube_0$ sont pris en compte, il y a " $2n$ " commutateurs dans un $BCube_1$). Plus généralement, un $BCube_k$ est construit à partir de " n " commutateurs $BCube_{k-1}$ s et " nk " extra n -ports. Ces commutateurs supplémentaires sont connectés à exactement un serveur dans chaque $BCube_{k-1}$. Dans un $BCube$ de niveau- k , chaque niveau requiert " nk " commutateurs (dont chacun est un commutateur n -ports). $BCube$ utilise plus de commutateurs lors de la construction de structures de niveau supérieur, alors que $DCell$ n'utilise que des commutateurs n -ports de niveau 0.

Les serveurs doivent cependant avoir " $k + 1$ " cartes d'interface réseau. La figure 2.12 montre un $BCube_1$ avec $n = 4$. Tout comme le $DCell$, le nombre de niveaux dans $BCube$ dépend du nombre de ports sur les serveurs. Le nombre de serveurs dans $BCube$ croît exponentiellement avec les niveaux, beaucoup plus lentement que $DCell$. Par exemple, lorsque $n = 6$, $k = 3$, un $BCube$ entièrement construit peut contenir 1296 serveurs. Considérant que $BCube$ est conçu pour des centres de données basés sur des conteneurs, une telle évolutivité est suffisante [Guo et al. 2009].

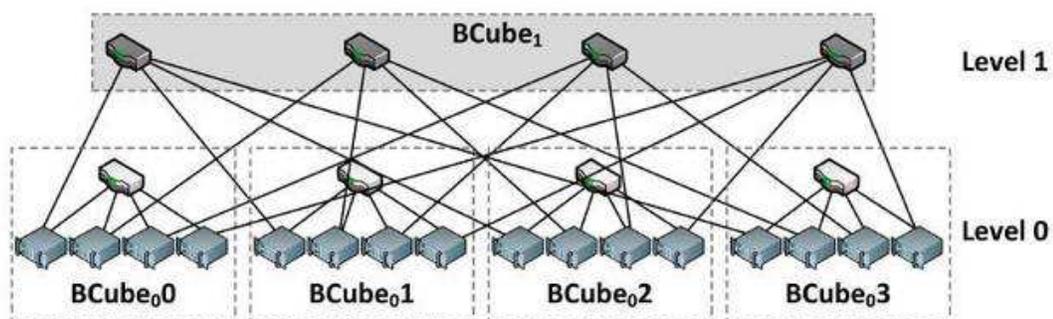


Figure 2.12: Construction de Bcube1 à partir de Bcube0 pour $n=4$ [Chirgwin 2015]

2.3.2.2 Architectures flexibles

Récemment, de plus en plus de chercheurs ([Wang et al. 2010], [Farrington et al 2010], [Chen et al. 2012]) ont envisagé d'utiliser la technologie de commutation optique pour construire des DCN. En plus d'offrir une bande passante élevée (jusqu'à une fibre Tbps avec des techniques WDM), les réseaux optiques ont une flexibilité significative de reconfiguration de la topologie pendant l'opération. Cette caractéristique est importante compte tenu des schémas de trafic déséquilibrés et en constante évolution dans les DCN. Dans cette section, nous présentons trois des architectures flexibles : c-Through [Wang et al. 2010], Helios[Farrington et al 2010], et OSA [Chen et al. 2012].

a. Architecture c-through

c-Through [Wang et al. 2010] est une architecture de réseau hybride qui utilise à la fois un réseau de commutation de paquets électrique et un réseau de commutation de circuit optique. Le réseau hybride de c-Through, ou réseau dit HyPaC (Hybrid Packet and Circuit), se compose de deux parties : un réseau électrique qui maintient la connectivité entre chaque paire de commutateurs ToR, et un réseau optique reconfigurable qui offre une interconnexion à bande passante élevée entre certains racks. En raison du coût relativement élevé du réseau optique et de la grande largeur de bande des liaisons optiques, il est inutile et coûteux de maintenir un lien optique entre chaque paire de disques. À la place, c-Through connecte chaque rack à un autre rack à la fois.

Par conséquent, les liaisons optiques à haute capacité sont temporairement offertes à des paires de baies en fonction de la demande de trafic. L'estimation du trafic entre les racks et la reconfiguration du réseau optique est réalisée par le plan de contrôle du système.

La figure 2.13 montre un réseau c-Through (les serveurs directement connectés aux commutateurs ToR ne sont pas représentés). Pour configurer la partie optique du réseau c-Through, le trafic entre les racks doit être connu. c-Through estime les demandes de trafic rack-rack en observant l'occupation de la mémoire tampon socket. Puisqu'une seule liaison optique est offerte à chaque rack, la topologie doit être configurée de telle sorte que la plus grande partie du trafic estimée puisse être satisfaite. Dans [Wang et al. 2010], le problème est résolu en utilisant l'algorithme d'appariement parfait de poids maximum (l'algorithme d'Edmonds) [Edmonds 1965].

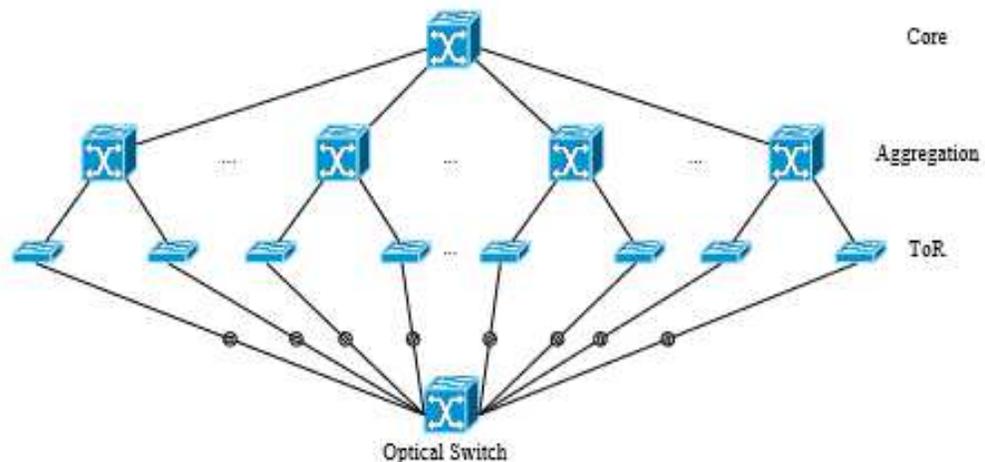


Figure 2.13: Topologie C-Through [Yang et al. 2013]

b. Architecture Helios

Helios [Farrington et al 2010] est un autre réseau hybride avec des commutateurs électriques et optiques. Helios est un arbre multi-racines à deux niveaux de commutateurs : le noyau et le Pod (ToR). Les commutateurs de noyau peuvent être électriques ou optiques, de manière à utiliser pleinement les deux techniques complémentaires. Contrairement à c-Through, Helios utilise le réseau électrique de commutation de paquets pour distribuer le trafic, tandis que la partie de commutation de circuit optique offre une bande passante de base au trafic de changement lent. Sur chacun des commutateurs Pod, les liaisons montantes sont équipées d'un émetteur-

récepteur optique. La moitié des liaisons montantes sont connecté aux commutateurs électriques, tandis que l'autre moitié est connectée au commutateur optique par l'intermédiaire d'un multiplexeur optique. Le multiplexeur combine les liens qui lui sont liés pour constituer un "super-lien" et permet une affectation de bande passante flexible sur ce super-lien.

La figure 2.14 montre une topologie de réseau Helios (les émetteurs-récepteurs ne sont pas représentés). Helios utilise un algorithme de Hedera [Al-Fares et al. 2010] pour estimer la demande de trafic entre les racks, qui alloue une bande passante juste pour les flux TCP. La configuration du réseau optique est la même que celle du c-Through pour maximiser la demande de trafic qui peut être satisfaite. Helios formule également le problème d'optimisation en tant que correspondance de poids maximum et le résout en utilisant l'algorithme d'Edmonds.

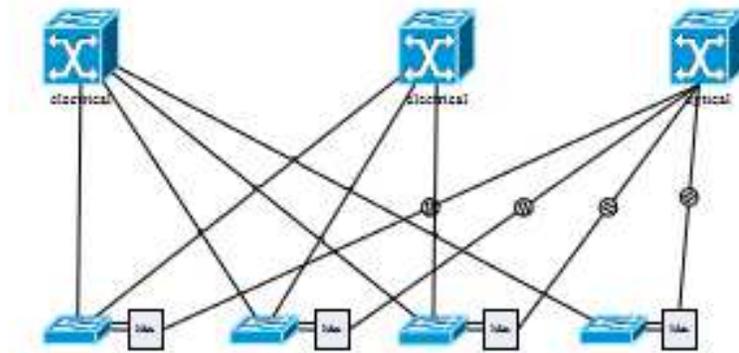


Figure 2.14: Topologie Helios [Yang et al. 2013]

c. OSA

OSA [Chen et al. 2012] est un réseau de commutation optique «pur», ce qui signifie qu'il abandonne les commutateurs du noyau électrique et n'utilise que des commutateurs optiques pour construire le noyau de commutation. Les commutateurs ToR sont toujours électriques, convertissant les signaux électriques et optiques entre les serveurs et le noyau de commutation. OSA permet plusieurs connexions au cœur de commutation sur chaque commutateur ToR (la quantité typique de ces liens est 4).

Cependant, le schéma de connexion est déterminé de manière flexible en fonction de la demande de trafic. Etant donné que le réseau n'assure pas de liaison optique directe entre chaque paire de racks avec demande de trafic, le système de contrôle construit la topologie pour en faire un graphe connecté et les commutateurs ToR sont chargés de relayer le trafic entre les autres commutateurs ToR.

OSA estime la demande de trafic avec la même méthode que Helios. Cependant, dans OSA, il peut y avoir plusieurs liaisons optiques offertes à chaque rack, et le problème ne peut plus être formulé comme le problème de correspondance de poids maximum. C'est un problème d'écoulement de plusieurs produits avec des contraintes de degré, qui est NP-difficile.

2.3.4 Comparaison des performances de quelques topologies

Le tableau 2.3 présente les comparaisons de certaines métriques de performance de différentes topologies.

2.3.4.1 Bande passante

Les quatre premières lignes du tableau 2.3 montrent la bande passante que les topologies peuvent offrir sous différents modèles de trafic. "One-to-one" signifie la bande passante maximale que la topologie peut offrir lorsqu'un serveur arbitraire envoie des données à un autre serveur arbitraire, et ainsi de suite.

La bande passante "All to All" signifie que chaque serveur établit un flux vers tous les autres serveurs. Chaque type de modèle de trafic est significatif pour différentes applications. Par exemple, un trafic one-to-severals survient lorsque le système de fichiers fait des répliques, le trafic one-to-all se produit lors de la mise à jour de certains logiciels sur tous les serveurs et le trafic all-to-all est très courant dans MapReduce[Yang et al. 2013]. Dans le tableau 2.3, les largeurs de bande sont exprimées en nombre de liens, ce qui implique que chaque lien dans un centre de données a la même bande passante. Il est montré dans le tableau 2.3 que les bandes

passantes one-to-one, one-to-severals et one-to-all sont en fait limitées par le nombre de ports sur les serveurs, ou le degré de serveurs. La topologie Basic tree offre la plus petite bande passante all-to-all, limitée par le nombre de ports du commutateur à la racine. Comme fat-tree introduit des commutateurs redondants, il peut offrir autant de chemins disjoints que le nombre de serveurs.

La largeur de bissection est le nombre minimal de connexions qu'il faut retirer pour avoir deux parties égales non connectées (mesure la capacité du réseau à transmettre des messages simultanément). Plus elle est grande, meilleure est la capacité de tolérance aux pannes qu'une topologie aura. Basic tree a la plus petite largeur de bissection, ce qui signifie qu'il est le plus vulnérable de tous. Une dégradation progressive des performances implique que lorsque de plus en plus de composants tombent en panne dans le centre de données, les performances diminuent lentement sans chutes dramatiques. Lors de la construction d'un nouveau centre de données, il est habituel qu'une topologie partielle soit construite en premier et que d'autres composants soient ajoutés en fonction des besoins futurs. Ce réseau partiel peut être vu comme un réseau avec beaucoup d'échecs. Construire un réseau partiel avec une topologie de dégradation gracieuse signifie que les gens peuvent obtenir les performances attendues avec moins de composants. [Guo et al. 2009] donne une comparaison entre fat-tree, DCell et BCube sur cette métrique avec des résultats de simulation.

	Tree-Based Architecture			Recursive Architecture	
	Basic Tree	Fat Tree	Clos Network	Dcell	Bcube
One-to-one	1	1	1	$k + 1$	$k + 1$
One-to-several	1	1	1	$k + 1$	$k + 1$
One-to-all	1	1	1	$k + 1$	$k + 1$
All-to-one	n	N	$\frac{2N}{n_{TOR}}$	$> \frac{N}{2^k}$	$\frac{n}{n-1}(N-1)$
BisectionWidth	$\frac{n}{2}$	$\frac{N}{2}$	$\frac{N}{n_{TOR}}$	$> \frac{N}{4 \log_n N}$	$\frac{N}{2}$

Tableau 2.3: Résumé de la performance des topologies [Yang et al. 2013]

2.3.4.2 Redondance matérielle

Le tableau 2.4 montre la redondance matérielle offerte par différentes topologies.

Avant de discuter du tableau 2.4, certaines définitions doivent être clarifiées :

Définition 1 (Node-Disjoint Paths)[Yang et al. 2013] : Le minimum du nombre total de chemins qui ne partagent aucun nœud intermédiaire commun entre des serveurs arbitraires.

Définition 2 (chemins disjoints)[Yang et al. 2013] : nombre minimal de chemins qui ne partagent aucun nœud edge commun entre des serveurs arbitraires.

Définition 3 (tolérance f-fault) [Duato et al. 2003]: Un réseau est f-fault tolérant si pour tout composant défectueux du réseau, le réseau est toujours connecté.

Définition 4 (Niveau de redondance)[Duato et al. 2003] : Un réseau a un niveau de redondance égal à r si et seulement si, après avoir supprimé tout ensemble de liens r , il reste connecté, et qu'il existe un ensemble de liens $r + 1$ tels qu'après les avoir supprimés, le réseau n'est plus connecté.

			Tree-Based Architecture			Recursive Architecture	
			Basic Tree	Fat Tree	Clos	DCell	BCube
Node-disjoint Paths			1	1	1	k+1	k+1
Edge-disjoint Paths			1	1	1	k+1	k+1
Redondancy Level	Switch	Edge/ToR	0	0	0	k	k
		Aggregation	0	$\frac{n}{2} - 1$	1		
		Core/Intermediate	0	$\frac{n}{4} - 1$	$\frac{n}{2} - 1$		
	Server		-	-	-	k	k

Tableau 2.4: Résumé de la redondance matérielle [Yang et al. 2013]

Dans le tableau 2.4, les niveaux de redondance des différents composants sont indiqués respectivement, ce qui reflète l'influence des différentes défaillances. Les nombres de chemins de nœuds disjoints et de chemins de bord-disjoints sont les mêmes dans chacune des topologies. En effet, il existe exactement un lien entre chaque paire de nœuds directement connectés (serveur à serveur, serveur à commutateur, commutateur à commutateur) dans ces topologies. Si des liens redondants sont ajoutés rendant la topologie multi-graphes, il y aura plus de chemins edge-disjoints que de chemins node-disjoints. Le nombre de chemins node-disjoints montre la capacité d'un système à gérer les défaillances de nœuds, tandis que le nombre de chemins edge-disjoints montre la capacité à gérer les échecs de liaison. Une défaillance de nœud peut avoir un impact plus grave qu'un échec de liaison, car elle entraîne généralement plusieurs échecs de liaison.

Les topologies tree-based ont moins de chemins node-disjoint et de chemins edge-disjoint par rapport à DCell et BCube. La principale raison à cela est que les

topologies arborescentes placent moins de ports sur les serveurs. Chaque serveur est connecté à un commutateur de périphérique avec une connexion unique dans les topologies arborescentes. Moins de ports sur les serveurs, rend les connexions entre les serveurs et les commutateurs vulnérables aux défaillances. Par exemple, dans les topologies arborescentes, si un commutateur Edge échoue, tous les serveurs qui lui sont connectés seront séparés du reste du centre de données. Cela explique également pourquoi il n'y a pas de redondance des commutateurs edge. DCell et BCube, cependant, ont plus de chemins node-disjoints et les chemins edge-disjoints à cause des ports multiples sur leurs serveurs, ce qui entraîne un niveau de redondance plus élevé. Il est à noter que le niveau de redondance de DCell ou BCube est déterminé par le nombre de ports sur les serveurs. Ce tableau ne montre pas la différence entre le taux de défaillance des différents composants.

2.4 Conclusion

Actuellement, il est possible de déployer facilement une application à travers le monde grâce au Cloud. Cela offre une grande opportunité aux entreprises de pouvoir satisfaire leurs clients locaux et internationaux en leur proposant des services.

Dans ce chapitre, nous avons effectué une présentation générale du Cloud, son historique, ses caractéristiques, son architecture et ses modèles de services. Nous avons aussi fait un petit résumé sur les topologies représentatives des réseaux de centres de données. Nous avons discuté des propriétés de ces topologies, et des techniques tolérantes aux pannes. De nouvelles topologies potentielles pour les réseaux existants sont toujours proposées. Une topologie doit trouver un équilibre entre performance, coût et durabilité.

Ainsi, dans le chapitre suivant, nous présentons un état de l'art sur les techniques de réduction de la consommation d'énergie dans le Cloud.

**Chapitre 3 : ETAT DE L'ART SUR L'ECONOMIE
D'ENERGIE DANS LE CLOUD COMPUTING**

3.1 Introduction

Un centre de données, appelé aussi centre de traitement de données est un site physique sur lequel se trouvent regroupés des équipements constituant du système d'information de l'entreprise (ordinateurs centraux, serveurs rangés dans des baies, baies de stockage, équipement réseaux et de télécommunication, etc.)¹⁸. Ils doivent être construits pour supporter les demandes croissantes de stockage des données et de calcul. Selon [Da COSTA 2015], en 2014 déjà, les centres de calcul classiques pouvaient contenir plusieurs dizaines de milliers de serveurs sur des surfaces de l'ordre de dix mille mètres carrés. Cette taille permettait d'organiser de manière efficace la gestion des fluides (refroidissement, électricité). Selon la même source, pour un centre de données classique d'environ 40 000 serveurs utilisés par 500 000 services, la consommation électrique représente environ 10 mégawatts. Cette consommation est telle que le prix d'achat des serveurs est dépassé en quelques années par le prix de leur consommation électrique.

Le directeur général Jim Witham de GaNSystems, leader des dispositifs de gestion de l'alimentation à base de nitrure de gallium (GaN), spécialisés dans la conversion de puissance, les semi-conducteurs et les transistors, affirme dans un article¹⁹ paru en Février 2018, qu'en 2017, environ 8 millions de centres de données dans le monde (allant des petits serveurs aux grands centres de données) géraient la charge des données. Ils avaient besoin pour cela de 416,2 térawattheures d'électricité. Cela représente 2% de la consommation mondiale totale et devrait atteindre 5% de la consommation mondiale d'ici 2020.

Il est alors important de mettre en place des mécanismes permettant d'économiser de l'énergie pour l'ensemble des composants du Cloud.

¹⁸https://fr.wikipedia.org/wiki/Centre_de_donn%C3%A9es

¹⁹<https://www.datacenterknowledge.com/industry-perspectives/achieving-data-center-energy-efficiency>

Pour une bonne maîtrise de l'énergie dans le Cloud, on s'intéresse généralement à ses différents composants tels que les centres de données, les serveurs et le système de refroidissement. Selon un rapport d'EPA [EPA 2007], les serveurs consomment 80% de l'énergie totale de l'infrastructure informatique et 40% de l'énergie totale des centres de données. Ils constituent l'une des cibles prioritaires dans le dispositif de mise en place d'une politique d'économie d'énergie dans les centres de données [Schappi et al. 2010]. L'énergie consommée par le système de refroidissement est importante, mais elle est proportionnelle à la consommation d'énergie de l'infrastructure informatique.

Des systèmes de refroidissement gratuits utilisés par les grandes compagnies telles que Google, Facebook, ebay sont intéressants pour réduire la consommation d'énergie liée au système de refroidissement. Ces approches réduisent la température dans les centres de données en utilisant de l'air ou de l'eau naturellement fraîche au lieu de la réfrigération mécanique et contribue ainsi à diminuer de manière drastique la consommation d'énergie liée au système de refroidissement. Ce gain en énergie pour le refroidissement peut atteindre les 100% dans certaines conditions climatiques.

3.2 Les principales causes des pertes d'énergie

Il existe plusieurs facteurs liés à la perte d'énergie dans les centres de données du Cloud. Les statistiques relatives à la consommation d'énergie des centres de données provenant de plusieurs enquêtes sont résumées à la figure 3.1, couvrant principalement le coût des ressources informatiques et physiques.

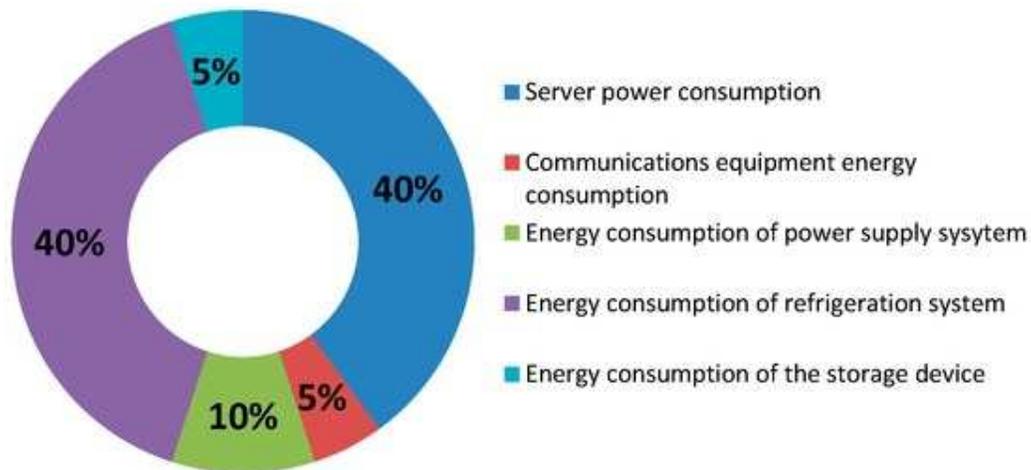


Figure 3.1: Répartition de la consommation d'énergie dans un centre de données [Jing et al. 2017]

3.2.1 Les serveurs

Comme indiqué dans la précédente partie, les serveurs font partie des éléments les plus énergivores dans les centres de données. Les principales raisons sont leurs sous-utilisations.

L'augmentation de la taille des centres de données a entraîné une augmentation considérable du nombre de serveurs dans les centres de données. Beaucoup d'études ont montré que les serveurs ne sont pas utilisés de manière efficiente²⁰, [Schappi et al. 2010]. L'*Uptime Institute*, dans son rapport de 2014 [Stansberry 2014], évalue à 20% les serveurs obsolètes, trop vieux ou inutilisés. Quant à Google [Barroso & U. Hölzle, 2007], une étude interne a montré qu'en 2007 une grande majorité de leurs serveurs étaient utilisés à moins de 50% de leur capacité. Ces serveurs contiennent pour la plupart des informations qui nécessitent leur disponibilité 24h/24 et 7 jours/7. Ces informations peuvent être des fois des fichiers attachés à certains mails datant de plusieurs années. La disponibilité de ces serveurs entraîne une grande consommation électrique et produit de la chaleur qu'il faut extraire.

²⁰<https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IP.pdf> 2014

A cela s'ajoute les pertes d'énergie des serveurs pendant leur état de repos. Dans [Naone 2009], il est indiqué qu'un serveur inactif consomme environ 70% de sa puissance maximale s'il est non utilisé. Les éléments qui sont les principaux responsables de la consommation d'énergie au niveau des serveurs sont le CPU, la mémoire vive, le disque de stockage, les blocs d'alimentation et les bus de connecteur d'extension [Beloglav 2013].

Si les CPU peuvent accepter de fonctionner en utilisant les modes de faible puissance, les autres composants ne peuvent être complètement ou partiellement mis hors tension. L'adoption des architectures multi-cœurs entraîne une diminution de la consommation d'énergie, par contre l'utilisation des puces sur les mémoires DRAM entraîne une augmentation de la consommation d'énergie [Djouhra, Thèse de doctorat 2016]. Ces pertes d'énergie des serveurs sont considérées comme l'une des causes majeures de l'inefficacité énergétique des centres de données.

3.2.2 Le système de refroidissement

Le refroidissement des centres de données est l'une des causes liées à la perte d'énergie dans les centres de données comme illustré sur la figure 3.1. La raison est que les serveurs sont des appareils électroniques de grandes tailles qui produisent énormément de chaleurs. Il est essentiel de maintenir des conditions de température et d'humidité appropriées pour s'assurer du bon fonctionnement des serveurs et de tous les équipements informatiques du centre de données. Ces conditions sont assurées par des systèmes de climatisation qui sont énergivores.

3.3 Gestion et analyse du trafic dans le Cloud privé et public

Baliga et al. [Baliga et al. 2011], en considérant trois différents modèles de services de Cloud comme le logiciel, le stockage et le traitement, étudie l'énergie consommée dans la commutation et la transmission, ainsi que le traitement et le stockage des données selon qu'on ait un Cloud public ou privé. Ces auteurs présentent une étude

sur l'énergie utilisée pour transmettre des données sur le Cloud. Ils ont constaté que le transport de données entre le Cloud et l'ordinateur personnel peut consommer des quantités d'énergie encore plus importantes que leur stockage.

En outre, en prenant en compte le stockage comme service du Cloud, ils soutiennent que le stockage dans le Cloud public peut consommer dans l'ordre de trois à quatre fois plus de puissance que le stockage dans le Cloud privé. Ils estiment la consommation d'énergie par bit de la transmission et la commutation pour un Cloud public à environ $2.7 \mu\text{J}/\text{b}$ [Baliga et al. 2011]. La raison principale est due au fait que les clients des Cloud publics utilisent ses services via Internet. En réalité, les paquets appartenant à un flux donné traversent en moyenne 12 à 14 sauts entre la source et la destination [Kliazovich et al. 2010], [Chase et al. 2001], et donc l'énergie consommée pour transporter ces paquets augmente.

La conclusion la plus importante notée dans [Baliga et al. 2011] est que lorsque l'on veut comparer la consommation d'énergie du Cloud basée sur les services (logiciels, stockage, traitement) avec un ordinateur personnel, nous devrions inclure la consommation d'énergie requise pour transporter les données du client vers les ressources du Cloud et vice versa. Il est obligatoire de prendre en compte le transport, si les clients utilisent les services du Cloud depuis un Cloud public accessible à travers Internet.

La tendance générale observée dans [Baliga et al. 2011] est que le Cloud privé est plus efficace que le Cloud public. Cette constatation est liée aux économies d'énergies dans le transport, et c'est donc particulièrement important de calibrer l'énergie consommée dans le transport de données entre les clients et le Cloud afin de progresser vers des Cloud plus économes en énergie.

3.4 Techniques de réduction de la consommation d'énergie dans les centres de données

Dans cette partie nous décrivons quelques approches qui permettent de réduire la consommation d'énergie dans les centres de données.

3.4.1 Approches basées sur les méthodes de virtualisation et de migration de machines virtuelles

Les centres de données ont connu ces dernières années une ascension fulgurante. De plus en plus, beaucoup de services sont proposés via le Cloud. Les fournisseurs sont ainsi amenés à adopter des technologies et des politiques plus efficaces dans la gestion de l'infrastructure de ces centres de données.

L'efficacité énergétique est devenue un enjeu majeur car ces centres de données ont tendance à consommer beaucoup d'énergies. Plusieurs solutions sont proposées dans la littérature pour réduire les impacts liés à l'environnement [Beloglazov et al. 2012], [Sekhar&Jeba2013], [Belbekkouche et al. 2012], [Beloglazov&Buyya, 2010], [Su et al. 2012]. Aujourd'hui, la question de l'écologie est omniprésente dans tous les domaines et l'informatique aussi se met au vert.

Le Cloud offre plusieurs techniques permettant la réduction la consommation d'énergie. De la virtualisation à la consolidation en passant par la migration de machines virtuelles, plusieurs possibilités s'offrent pour réduire la consommation d'énergie dans les centres de données. La gestion dynamique des composants du centre de données est aussi un moyen permettant de gérer de manière efficace la consommation d'énergie.

Nous présentons dans cette partie un état de l'art sur la gestion de la consommation d'énergie dans les centres de données.

La technologie de virtualisation fait partie des événements ayant contribué à l'événement du Cloud Computing. Elle permet d'instancier plusieurs machines dites

"virtuelles" sur une seule machine physique. Ainsi, cette technologie peut permettre de réduire le nombre de machines physiques en cours d'exécution dans un centre de données et participer à la réduction de la consommation d'énergie. Lorsque des machines physiques sont sur utilisées ou très peu utilisées, les machines virtuelles qui y sont instanciées peuvent être déplacées vers d'autres machines physiques. Cette migration des machines virtuelles peut ainsi participer à l'équilibrage de charge et à la réduction de la consommation d'énergie et peut se faire à chaud ou à froid. On parle dans ce cas de consolidation. La consolidation est une approche efficace pour parvenir à une meilleure efficacité énergétique, elle contribue à améliorer l'utilisation des ressources par de nombreuses migrations de machines virtuelles résidents sur plusieurs serveurs "sous-utilisés" vers un seul serveur [Djouhra, Thèse doctorat 2016].

Dans [Beloglazov et al. 2012], les auteurs ont utilisé la technique de migration de machines virtuelles. Les auteurs ont mis l'accent sur la consommation d'énergie du processeur. La mémoire vive, les interfaces réseaux et autres n'ont pas été pris en compte. Des seuils supérieurs et inférieurs d'utilisation du CPU des serveurs ont été définis dans leur proposition. Lorsque l'utilisation du processeur de l'hôte est en dessous du seuil inférieur, toutes les machines virtuelles doivent être transférées et la machine hôte est mise en veille afin d'économiser de l'énergie. Par contre, si l'utilisation du processeur dépasse le seuil supérieur, certaines machines virtuelles sont migrées vers d'autres machines physiques pour réduire l'utilisation du CPU. La deuxième phase consiste en la mise en place de la machine virtuelle sélectionnée. L'algorithme MBFD (Modified Best Fit Decreasing) a été utilisé. Plusieurs métriques de performances ont été considérées et les résultats obtenus par les auteurs ont montré que la consommation d'énergie pouvait être réduite jusqu'à 77% et 53% par rapport respectivement à NPA (No Power Aware) et DVFS (Dynamic Voltage Frequency Scaling) avec 5,4% des violations de SLA. NPA suppose que tous les hôtes fonctionnent à 100% d'utilisation du CPU et consomment une puissance maximale

tout le temps, tandis que la technique DVFS permet d'augmenter ou de réduire le voltage et la fréquence du processeur.

Les auteurs de [Beloglazov&Buyya, 2010] proposent une gestion des ressources énergétiques dans le Cloud en utilisant la technique de migration à chaud avec une activation/désactivation des nœuds utilisant la technique DVFS. L'algorithme est appliqué à une infrastructure de Cloud constituée de nœuds physiques hétérogènes ayant chacun un CPU multi-coeur performant et disposant de plusieurs machines virtuelles. L'architecture proposée est constituée de deux gestionnaires sur chaque nœud physique et d'un dispatcheur. Lorsqu'une requête est émise, le dispatcheur envoie la demande au gestionnaire global. Le gestionnaire global donne les informations sur l'utilisation de la ressource, les machines virtuelles qui doivent être allouées et l'émission des ordres de migration. Le gestionnaire des machines virtuelles effectue les redimensionnements, les commandes et les migrations selon les ordres reçus. La migration est effectuée lorsque les ressources sont très peu utilisées ou que leur utilisation atteint 100%. Cette migration s'effectue en deux étapes que sont la sélection de la machine à migrer et la mise en place de cette dernière. Plusieurs politiques sont simulées : NPA, DVFS, ST (Single Threshold), MM (Deux seuils) par le simulateur CloudSim et les résultats expérimentaux montrent la souplesse de la politique MM [Beloglazov&Buyya, 2010].

Une méthode pour détecter la sur-utilisation et la sous-utilisation des processeurs des serveurs a été proposée dans [Sekhar&Jeba2013]. Dans le cas d'une sur-utilisation du CPU, certaines machines virtuelles sont migrées vers d'autres hôtes afin de réduire l'utilisation du serveur sous un seuil déterminé. En cas de sous-utilisation, toutes les machines virtuelles sont migrées vers d'autres hôtes et cette dernière est éteinte à la fin pour réduire la consommation énergétique. Pour l'allocation des machines virtuelles, les auteurs ont utilisé des heuristiques. Le simulateur CloudSim a été

utilisé pour implémenter la politique de migration et les heuristiques. Les résultats obtenus sont meilleurs que ceux précédemment obtenus dans la littérature.

Dans [Belbekkouche et al. 2012], les études ont montré que lorsque les nœuds et les liens virtuels sont conjointement intégrés, ils procurent un meilleur mappage avec un moindre coût en un temps réduit comparé aux deux approches d'étape d'intégration [Su et al. 2012]. Les auteurs de [Su et al. 2012] ont proposé une intégration nommée "EA-VNE" (Energy-Aware Virtual Network Embedding) qui est basée sur les contraintes du CPU et de la bande passante, néanmoins, leur algorithme ne supporte pas le partage de chemin. De manière similaire, une allocation de ressources prenant en compte l'hétérogénéité des réseaux virtuels et physiques a été proposée dans [Nogueira et al. 2011]. Leurs contraintes d'élimination pour un réseau virtuel intégré sont basées sur l'emplacement, la charge et le nombre de CPU, la quantité de mémoire vive disponible entre autres.

L'intégration suggérée dans [Yu et al. 2008] permet le partage de chemin et la migration de lien. Cependant, pour une meilleure intégration comme suggérée dans [Nogueira et al. 2011], leur approche [Yu et al. 2008] devrait corrélérer leur intégration de lien avec le précédent mappage de nœud. Fondamentalement, la plupart de ces algorithmes ne prennent pas en compte le fait que les ressources appartenant à la fois au réseau physique et virtuel peuvent être hétérogènes.

L'avènement de l'Internet des objets(IoT), avec le développement rapide de la technologie informatique en nuage ainsi que la consommation d'énergie dans les centres de données causent beaucoup d'inquiétude. Le réseau défini par logiciel (SDN) offre une solution réalisable au problème rigide de l'architecture réseau traditionnelle. Des études [Liao & Wang 2018] beaucoup plus récentes ont montré que l'utilisation du SDN pour contrôler l'information de l'ensemble du réseau avec une gestion dynamique de l'équipement peut économiser environ la moitié de l'énergie consommée dans les centres de données.

Les centres de données Cloud peuvent aussi réduire la consommation d'énergie par le biais de la consolidation des routeurs [Wang et al. 2008], de tels sortes que plusieurs charges de travail (Worload) peuvent partager le même nœud physique en utilisant la virtualisation, et ainsi, les routeurs non utilisés peuvent être éteints [Bolla et al. 2011]. En effet, les techniques de virtualisation des routeurs permettent à un seul routeur physique de créer plusieurs routeurs virtuels [Wang et al. 2008]. Chaque routeur virtuel conserve plusieurs tables de routage pour servir le trafic provenant de réseaux multiples. Ce comportement est dû au fait qu'il existe des instances de routeurs distincts pour chaque routeur virtuel. Un routeur virtuel mobile appelé VROOM (Virtual Router On the Move) a été proposé dans [Wang et al. 2008]. En tirant partie de la séparation du plan de contrôle et du plan de données proposée dans les routeurs modernes, un routeur virtuel donné peut être migré d'un routeur physique à un autre sans perturber le trafic de données qu'il transporte. Cette solution (VROOM) peut migrer un routeur virtuel d'un routeur physique à un autre sans perturber le flux de trafic ou changer la topologie logique, ce qui évite la nécessité de reconfigurer les routeurs virtuels tout en évitant des retards de convergence sur les protocoles de routage.

Le trafic sans perturbation est dû au fait que le plan de contrôle d'un routeur virtuel donné est migré d'un routeur physique à un autre: Ce plan de contrôle clone par la suite l'état du plan de données au nouvel emplacement tout en continuant de mettre à jour l'état à l'ancien emplacement. La solution VROOM migre seulement le plan de contrôle, tout en continuant à acheminer le trafic à travers l'ancien plan de données. Ainsi le plan de contrôle s'exécute sur le nouvel emplacement et alimente le nouveau plan de données en mettant parallèlement à jour l'ancien plan de données. La raison pour laquelle les pertes de paquets de données ne sont pas remarquées est que la solution VROOM transmet le paquet en utilisant le plan de données qui est chargé dans l'ancien et le nouvel emplacement. Ainsi la migration asynchrone des liens de connectivité est prise en charge. Une fois que tous les liens appartenant à un routeur

virtuel qui est déplacé sont migrés, l'ancien plan de données et ses liens sortants peuvent être supprimés en toute sécurité. Notez que, avant le processus de migration, un canal pour rediriger le routage des messages est établi entre les deux routeurs physiques. Pour plus de détail sur le processus de migration, nous suggérons au lecteur de se référer au [Wang et al. 2008].

3.4.2 Approches existantes pour réduire les besoins en énergie du réseau

Des travaux précédents comme [Pries et al. 2011], [Gupta & Singh 2003] et [Zhang et al. 2008] proposent de faire des économies d'énergie au niveau de la couche réseau. Les auteurs de [Zhang et al. 2008] observent qu'un gain d'énergie peut être obtenu soit en reconfigurant de manière dynamique la topologie Internet via le routage, ou plus tôt pendant la phase de la conception du réseau. Nedeveschi et al. dans [Nedeveschi et al. 2008] proposent de mettre les composants du réseau en veille pendant les temps d'inactivité (ou dans un mode d'économie d'énergie) et ainsi retarder les lots de trafic sur les buffers situés aux bords du réseau. En retardant les paquets, ils peuvent réorganiser le trafic vers des rafales plus cohérentes, et donc produire une organisation plus efficace en énergie. Les auteurs affirment qu'en général, un dispositif qui est utilisé à une fréquence moindre peut permettre une réduction considérable de la consommation d'énergie.

Néanmoins, ces solutions peuvent augmenter le délai et les pourcentages de perte de paquets si un composant donné du réseau est éteint et qu'un lot de paquets arrive sur ses interfaces. En raison du comportement des flux dans le réseau filaire et les contraintes de QoS rigoureuses, des rotations des nœuds en mode On/Off, peuvent être très difficiles dans le contexte filaire. Il est important de noter que les protocoles Internet devraient être modifiés si nous prenons en compte les études comme [Gupta & Singh 2003], [Zhang et al. 2008] et [Pries et al. 2011]. En effet, Gupta et al. dans [Gupta & Singh 2003] décrivent les problèmes qui peuvent se produire sur deux

importants protocoles comme OSPF et IBGP lorsque nous mettons des interfaces de commutateurs et de routeurs en mode veille.

Pour s'attaquer au problème de la réduction de la consommation d'énergie, les auteurs de [Pries et al. 2011] ont proposé d'ajouter aux couches de pile un Système de gestion de l'énergie (EMS) qui a pour rôle de surveiller de manière proactive la consommation instantanée d'énergie de chaque lien pilote et routeur, les informations sur les chemins, les taux de trafic et l'état de la QoS des flux. Selon ces informations, le EMS est capable de recommander un ensemble de chemins qui optimisent les QoS des utilisateurs avec la plus basse consommation d'énergie. Par conséquent, un ensemble de nœuds ou certains liens peuvent être éteints afin de réduire la consommation d'énergie. Notez que l'EMS conçu s'exécute sur la couche IP supérieure [Pries et al. 2011]. Néanmoins, une grande quantité de données est collectée par les EMS avant la découverte d'un chemin donné qui diminue la consommation d'énergie avec un certain niveau de QoS.

Malgré un gain de 10 à 20% de consommation d'énergie, les résultats obtenus montrent que le délai mesuré et les paquets perdus augmentent lorsqu'un ensemble de routeurs sont mis hors tension. L'augmentation du délai mesuré ainsi que les pics de perte sont une sérieuse préoccupation pour les applications en temps réel. Un pic de perte momentané est également enregistré depuis des liens donnés pouvant être désactivés alors que certains paquets sont encore stockés dans la file d'attente du routeur [Pries et al. 2011].

Dans un but similaire, Berl et al. [Berl et al. 2010] suggèrent d'optimiser ou de redévelopper les protocoles réseau afin d'obtenir plus d'efficacité énergétique. En outre, ils suggèrent de réduire le coût énergétique des logiciels des centres de données qui exécutent les applications du Cloud. Une proposition intéressante est de prendre en compte la consommation d'énergie comme paramètre clé d'un fonctionnement

optimal lorsque les ingénieurs font le plan de la réplication du contenu et des algorithmes de diffusion pour les services du Cloud.

3.4.3 Économie d'énergie dans les réseaux d'accès

Compte tenu de la grande hétérogénéité des équipements, l'amélioration de l'efficacité énergétique du futur Internet constitue un enjeu majeur. En effet, les auteurs de [Bolla et al. 2011] ont montré qu'en ce qui concerne une configuration typique de réseau ISP/telco, les équipements d'accès (par exemple, les modems, les passerelles domestiques, les équipements de réseau à large bande sans fil, les multiplexeurs d'accès aux liens des abonnés numériques) consomment environ 70% de l'exigence globale en énergie du réseau, alors que la consommation d'énergie du réseau de transport/cœur est de 30%. La raison principale est due au fait que les nœuds d'accès sont estimés approximativement à 94% du réseau d'accès, et la consommation d'énergie approximative par appareil est de 10000, 6000 et 1600 Watt / h respectivement pour le cœur, le transport et le réseau d'accès [Bolla et al. 2011].

Dans [Gomma et al. 2011], les auteurs proposent le Broadband Hitch-Hiking (BH²), qui permet de rendre "vert" le réseau d'accès. L'objectif est de diminuer l'énergie consommée dans les réseaux d'accès côté utilisateur ainsi que sur les modems DSLAM du fournisseur de services Internet (ISP). Ils suggèrent d'agrèger le trafic de plusieurs utilisateurs dans quelques routeurs qui sont disponibles à proximité de la résidence des clients et donc laisser les routeurs restants en mode veille. Cette approche est réalisée en donnant la possibilité aux points d'accès sans fil (AP) de se connecter à d'autres AP, puis envoyer leur trafic pour le routage. Ainsi, le routeur local d'un client donné peut entrer en mode veille.

Plusieurs problèmes devraient être abordés avant un véritable déploiement de BH². Tout d'abord, il est obligatoire de comprendre comment la coopération demandée entre les propriétaires des passerelles (routeurs) dans la même zone géographique sera effectuée. Aussi, vérifier si l'échange de routage, avant d'arriver aux modems ISP

DSLAM peut être possible en dépit des fournisseurs du client. Enfin, il est nécessaire de prendre en compte les problèmes de sécurité et de confidentialité, car le trafic des clients sera acheminé via les AP voisins. Par exemple, un AP malveillant peut détourner le trafic ou modifier les données qui doivent être transmises et, par conséquent, protéger l'intégrité des données est nécessaire pour un déploiement réussi de BH².

Nonobstant le fait que les dispositifs d'accès ont de petites exigences en matière d'énergie, l'amélioration de l'efficacité énergétique de l'accès au cœur et au périphérique réseau est obligatoire car l'énergie consommée par les routeurs devient plus importante avec l'augmentation du taux d'accès des utilisateurs [Bolla 2011]. La consommation globale d'énergie augmentera au fur et à mesure que la capacité du réseau augmente. L'énergie consommée en revanche par les périphériques réseau appartenant aux réseaux d'accès est en grande partie indépendante du volume de trafic [Baliga 2011].

3.4.4 Économie d'énergie basée sur l'organisation du centre de données

Heller et al. [Heller et al. 2010] ont publié un document portant sur le compromis entre l'efficacité énergétique et la résilience. Pendant le fonctionnement normal du réseau, les commutateurs supplémentaires utilisés pour les chemins de sauvegarde sont désactivés et seront uniquement activés en cas de panne de liens ou de charge élevée. Le mécanisme proposé est mis en œuvre dans un banc d'essai où OpenFlow est utilisé pour la gestion des commutateurs. Cependant, ils n'arrêtent que les commutateurs et non les serveurs. Pries et al. [Pries et al. 2011] montrent que ceux-ci ne consomment qu'entre 5% et 10% de la consommation globale d'énergie.

Kliazovich et al. [Kliazovich et al. 2010] ont développé un environnement de simulation pour calculer la consommation d'énergie de plusieurs architectures de centres de données. Les auteurs ont, en plus de montrer la part de la consommation d'énergie du réseau et du serveur, indiqué la quantité d'énergie pouvant être

sauvegardée en utilisant le voltage et l'échelle de fréquence dynamique ou la gestion dynamique de l'alimentation. L'un des premiers articles présentant une gestion dynamique des ressources selon la charge proposée est présenté par Chase et al. [Chase et al. 2001]. Ils proposent une architecture où les clusters de serveurs sont redimensionnés dynamiquement conformément aux SLA négociés.

Une approche plus détaillée est présentée par Chen et al. [Chen et al. 2005]. Trois solutions sont proposées pour réduire la consommation d'énergie des serveurs dans un centre de données. Pour la première solution, le comportement de la charge de travail d'un avenir proche est prévu tandis que la seconde est une solution réactive, utilisant des retours d'informations périodiques du système d'exécution. La troisième solution proposée est une solution hybride utilisant une combinaison de prédiction et de rétroaction périodique.

Les auteurs de [Schwartz et al. 2012] évaluent les économies d'énergie en adaptant de manière dynamique le nombre de serveurs selon la charge actuelle en utilisant des seuils sur la file d'attente pour activer des serveurs supplémentaires et sur le nombre total de requêtes dans le système pour désactiver les serveurs. En outre, ils utilisent une véritable approche d'optimisations multi-objectives pour trouver un optimum entre le temps d'attente et la consommation d'énergie. Pour l'architecture du centre de données, ils utilisent une topologie Three-tier avec deux groupes de serveurs. Le premier groupe de serveurs est toujours actif. Si le nombre de requêtes dans la file d'attente dépasse le seuil fixé, le deuxième groupe de serveurs est activé. Ces derniers le resteront jusqu'à ce que le nombre total de requêtes dans le système dépassent le second seuil fixé. ChristianSchwartz et al.[Schwartz et al. 2012] cherchent à garantir une qualité de service (un temps de réponse acceptable) tout en économisant de l'énergie. Les résultats obtenus montrent que des configurations existent, de sorte que la consommation d'énergie peut être considérablement réduite tout en conservant un

temps d'attente moyen acceptable. Ainsi, ils garantissent des accords de niveau service à l'utilisateur final tout en conservant environ 40% d'énergie.

[Dan et al. 2015] abordent le problème de la consommation d'énergie en tenant compte des performances du système dans un cadre général d'optimisation. Plutôt que de considérer un seul groupe de sauvegarde, les auteurs de [Dan et al. 2015] proposent trois groupes de serveurs dont les deux sont des groupes de sauvegarde. En prenant en compte, les considérations précédentes, à savoir le compromis entre la consommation d'énergie et la performance, et l'attribution correcte de groupes de sauvegarde, ils étudient la relation entre la consommation totale d'énergie et le temps moyen d'attente du service dans une méthode de gestion dynamique élaborée. Étant donné que tous les serveurs sont divisés en trois groupes, les auteurs modélisent un centre de données avec des arrivées simples suivant une distribution poissonnienne, un tampon de capacité finie et un temps de service exponentiel. Ils évaluent les économies d'énergie en adaptant dynamiquement les groupes de serveurs en fonction de la charge actuelle en utilisant des seuils d'activation sur la taille de la file d'attente pour démarrer un certain groupe de serveurs. Afin de trouver les seuils d'activation optimum, ils formulent le problème d'optimisation de la minimisation de l'énergie selon les exigences de performance données en termes de temps moyen d'attente du service.

3.5 Conclusion

Dans ce chapitre, nous avons passé en revue les travaux relatifs à la consommation d'énergie dans les centres de données. Nous avons d'abord montré les principales causes des pertes d'énergies dans les centres de données, par la suite, nous avons présenté une comparaison en termes de consommation d'énergie entre un Cloud public et un Cloud privé. Dans le but d'apporter notre contribution pour un Cloud plus écologique, nous avons étudié plusieurs techniques de réduction de la consommation d'énergie dans le Cloud.

Cette revue de la littérature nous a permis de voir des points forts et des points faibles des propositions existantes. Elle nous a permis de voir par exemple que même si EA-VNE permet une intégration des nœuds et des liens virtuels dans le but d'économiser de l'énergie, son algorithme ne supporte pas le partage de chemin et plusieurs algorithmes proposés ne prennent pas en compte le fait que les ressources appartenant à la fois au réseau physique et virtuel peuvent être hétérogènes.

Sur l'organisation des centres de données, l'idéal est de faire en sorte qu'en fonction du trafic, certains serveurs puissent être éteints ou mis en veille. DAN et al. ont, dans la suite des travaux d'autres auteurs, proposé trois groupes de serveurs dont un seul est toujours actif. Le résultat obtenu est certes une avancée, mais nous avons montré qu'en utilisant la topologie Fat-Tree avec plus de groupes de serveurs, nous pouvons économiser plus d'énergie tout en gardant les mêmes performances. Dans les chapitres 4 et 5, nous présentons nos contributions.

**Chapitre 4 : "GREEN CLOUD" : Une architecture
efficiente pour la réduction de la consommation d'énergie**

4.1 Introduction

La réduction de la consommation d'énergie dans le Cloud est devenue l'une des directions de recherche les plus difficiles en raison de la très grande croissance des services qui y sont hébergés et fournis. L'efficacité énergétique est devenue un enjeu important de nos jours. Par conséquent, l'économie d'énergie dans les centres de données attire de plus en plus l'attention des chercheurs. Si certains s'intéressent à la consommation d'énergie au niveau du cœur des centres de données, d'autres tentent de voir des solutions pour réduire cette consommation d'énergie lors du transport des données. L'énergie consommée par le transport des données représente un pourcentage important de la consommation globale au niveau du Cloud.

Ainsi, en exploitant le réseau et la consolidation des routeurs, nous proposons une architecture de Cloud écologique appelé Green Cloud Architecture (GCA), dans laquelle on peut soit éteindre ou mettre en veille des routeurs virtuels. On peut aussi migrer des routeurs virtuels vers des routeurs physiques en fonction de l'énergie courant. Notre architecture verte réduit de manière considérable le nombre de routeurs sous tensions et donc la consommation d'énergie lors du transport des données jusqu'à 41% [Kasse et al. 2014] et [Gueye et al 2014].

Dans cette contribution, nous nous concentrons sur les stratégies d'économie d'énergies lors du transport et de la commutation dans le Cloud.

Dans un premier temps, nous proposons une architecture de Cloud écologique appelé Green Cloud Architecture (GCA) où la virtualisation des routeurs sera le pilier pour le transport des données. Dans ces cas de figures, les routeurs physiques peuvent créer plusieurs routeurs virtuels, ainsi nous concevrons des mécanismes d'acquisition et de contrôle de l'ensemble des ressources du réseau. Étant donné que la puissance consommée par un routeur dépend du nombre de ports activés [Bianzino et al. 2012], en activant des interfaces sur les instances de routeurs virtuels en veille, nous pouvons réduire la quantité d'énergie consommée par les routeurs ainsi que la

consommation globale d'électricité en raison des commutations. Par la suite, nous évaluerons notre architecture GCA en concevant un algorithme d'allocation de ressources basé sur l'énergie courante qui intègre les routeurs virtuels sur les routeurs physiques et cherche à minimiser l'énergie consommée pendant le transport.

Dans un second temps, nous utilisons l'intégration des réseaux virtuels (Virtual Network Embedding, en anglais) pour le mapping des routeurs virtuels sur des routeurs physiques tout en cherchant à minimiser la consommation énergétique pendant le transport.

4.2 Optimisation de la virtualisation des réseaux pour un transport éco-énergétique

La proportionnalité énergétique signifie que la consommation d'énergie par un dispositif de mise en réseau devrait être proportionnelle soit à la charge soit aux nombres de ports actifs sur ses interfaces. Nos mesures proposées tiennent compte de ces facteurs. La première mesure est liée à la puissance consommée par chaque routeur qui fait partie du réseau physique. Dans la suite, les termes "switch" et "routeur" sont utilisés de manière interchangeable.

$$Power_{switch} = \sum_{r=0}^{configs} numports_{configs_r} \times Power_{configs_r} + num_{linecards} \times Power_{linecards} + Power_{chassis}(1)$$

En effet, la puissance consommée par un routeur augmente de façon linéaire en fonction du nombre d'interfaces disponibles sur le routeur [Chabarek et al . 2012]. Etant donné que chaque interface peut avoir plusieurs ports ($numports_{configs_r}$) en cours d'exécution avec un taux "configs", " $Power_{configs_r}$ " donne la puissance consommée par un port en cours d'exécution de taux "r", où "r" peut être égal à 10 Mbps, 100 Mbps ou 1 Gbps. L'utilisation du port n'est pas considérée dans l'équation

1 puisque le trafic à travers des dispositifs de mise en réseau n'a pas un effet significatif [Chabarek et al . 2012].

La deuxième mesure, appelée indice de proportionnalité d'énergie (EPI, pour Energy Proportionality Index en anglais), présente la corrélation potentielle entre la puissance consommée par un routeur et sa charge [Greenberg et al. 2009]. L'indice EPI est égal à

$$\frac{M-I}{M} \times 100 \quad (2)$$

où M désigne la puissance maximale consommée et I désigne la puissance consommée par le routeur quand il est allumé mais ne traite pas de trafic (état de repos). I retourne la puissance du châssis ($Power_{chassis}$ Equation 1). En effet, la puissance maximale est la puissance consommée lorsque tous les composants qui sont installés sur le routeur fonctionnent.

Enfin, la troisième métrique appelée puissance normalisée permet d'estimer la consommation d'énergie par bit pendant la transmission et la commutation. Il est défini comme suit:

$$NormalizedPower = \frac{M}{G} \quad (3)$$

où G représente la bande passante totale agrégée qu'il peut supporter. En effet, plus la valeur de $NormalizedPower$ est petite, plus le routeur est économe en énergie. L'indice EPI et les métriques de la puissance normalisée sont utilisés pour estimer la proportionnalité énergétique des dispositifs de mise en réseau. Ces facteurs représentent des contraintes éliminatoires tout au long de la sélection des routeurs (nœuds physiques).

4.3 Virtualisation Réseau/Dispositif

Pour parvenir à une certaine efficacité énergétique, nous proposons la création de plusieurs machines virtuelles sur un nœud physique. Nous réduisons ainsi la quantité de matériels utilisés et améliorons l'utilisation des ressources. En effet, la

consolidation de nœud permet de déplacer les machines virtuelles tournant sur plusieurs nœuds sous-utilisés sur un seul nœud ou sur un nombre minimal de nœuds physiques et donc les nœuds restants peuvent être réglés sur les modes d'économie d'énergie.

Par exemple, la figure 4.1 montre un environnement de virtualisation de réseau où deux topologies de réseaux virtuels sont installées sur une seule infrastructure physique partagée. Dans les réseaux virtuels, les valeurs numériques à côté des liens expriment la largeur de la bande passante demandée par les liens virtuels de ces deux topologies. En revanche, par rapport au réseau physique, les valeurs à côté des nœuds et des liens représentent des contraintes (C'est-à-dire les ressources disponibles) de ces nœuds et liens à un instant t donné.

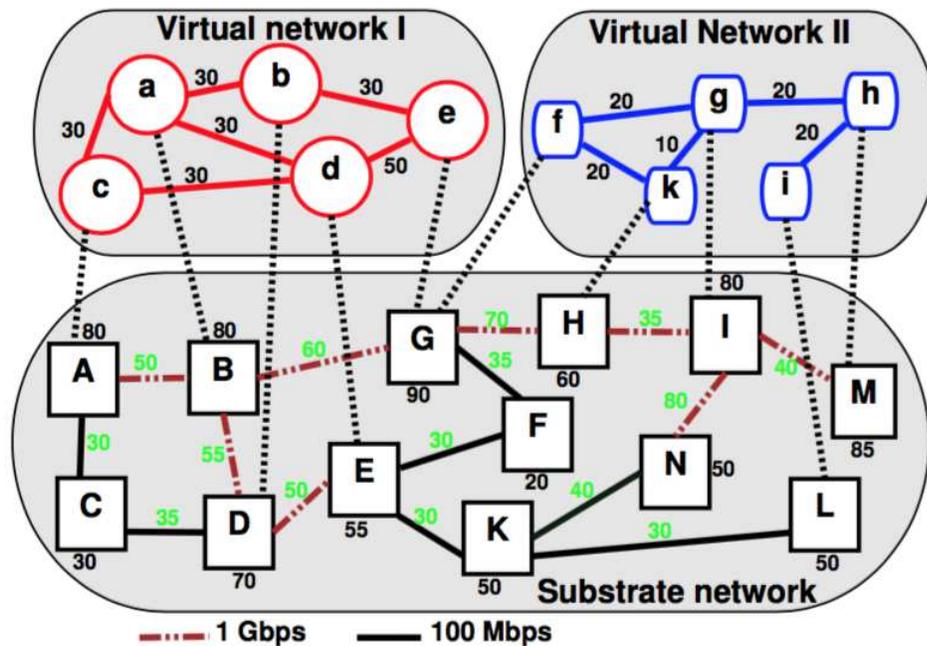


Figure 4.1: Affectation des nœuds virtuels sur un réseau physique

4.4 Présentation de GCA (Green Cloud Architecture)

Comme indiqué dans [Mahadevan et al. 2009], la consommation d'énergie d'un routeur augmente linéairement en fonction du nombre d'interfaces disponibles dans le routeur. Cette même tendance est également notée sur le nombre de ports actifs au niveau de chaque interface. Afin de réduire l'énergie consommée lors des communications dans le Cloud, nous proposons une nouvelle approche où la virtualisation des routeurs est le pilier de cette nouvelle architecture (figure 4.2).

La structure générale de notre architecture appelée GCA (figure 4.2) est formée par:

- (i) un réseau standard d'un fournisseur d'accès à Internet (ISP) logiquement divisé en un réseau d'accès, un réseau métropolitain et un réseau central;
- (ii) deux réseaux identiques de centre de données à trois niveaux, formés par la couche de base (Core), la couche d'agrégation (Aggregation) et la couche d'accès (Access) [Zhang et al. 2010].

Nous notons que les commutateurs situés au niveau de l'agrégation ont une capacité de commutation importante afin d'agréger le trafic provenant de la couche d'accès, par opposition à ceux situés aux niveaux des racks de serveurs.

La distribution des routeurs virtuels [Yu et al. 2008] et non virtuels est effectuée de manière aléatoire sur la figure 4.2. Comme on pouvait s'y attendre les deux types de routeurs peuvent partager la même topologie et travailler ensemble de façon transparente. Le Cloud permet une virtualisation des centres de données et des applications qui seront présentés sous forme de services.

L'architecture proposée peut distribuer les trafics des utilisateurs entre le principal centre de données et le centre de données secondaire en respectant la quantité de trafic observée dans un centre de données désigné ou en fonction de la proximité géographique des utilisateurs du Cloud. Par conséquent, nous faisons la promotion d'une haute disponibilité des applications et de l'accès aux données ainsi que

l'évolutivité de performance avec moins d'énergie consommée en considérant les routeurs virtualisés.

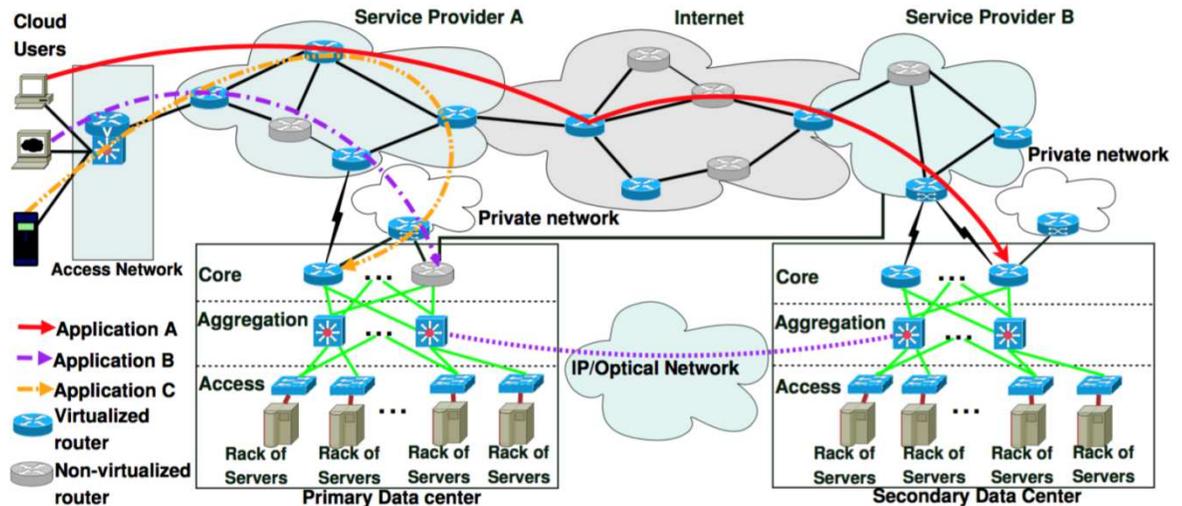


Figure 4.2: Green Cloud Architecture (GCA)

4.4.1 Objectif

Dans cette proposition, notre objectif est d'identifier et de gérer le trafic du Cloud en redirigeant le trafic Internet vers leurs destinations (par exemple en fonction du service ciblé, la localisation des centres de données). Nous pouvons par conséquent affecter les flux correspondants à une interface donnée d'un routeur fixe prenant en charge plusieurs instances de routeurs virtuels. En effet, chaque interface est composée d'un hyperviseur, appelé également gestionnaire de machines virtuelles. L'hyperviseur est un programme informatique qui permet à plusieurs systèmes d'exploitation de partager un seul hôte matériel. Il permet de créer, de gérer et de libérer les routeurs virtuels d'un routeur physique. En plus, les routeurs virtuels créés et gérés par les hyperviseurs ont leurs propres tables de routage, leurs propres tables de transfert et leur propre mémoire tampon en locale. Chaque routeur virtuel fonctionne indépendamment des autres et il est possible de le migrer ou de l'arrêter sans affecter les autres.

Par exemple, la figure 4.2 illustre un équilibreur de charge, situé au niveau de la couche agrégation dont le rôle est de transférer le trafic vers un autre centre de données. Nous décrivons un scénario où trois utilisateurs demandent un service Cloud. Par exemple, « Application A », « Application B » et « Application C » peuvent être considérées comme étant des requêtes de réseaux virtuels qui doivent être intégrées dans le réseau physique. Suivant un routage optimal, les applications des utilisateurs sont redirigées sur les deux centres de données en prenant des chemins différents au sein du réseau.

Tenant compte de GCA, nous sommes en mesure de mettre certains ou l'ensemble des routeurs virtuels d'un nœud physique particulier, soit à l'état de veille (faible consommation d'énergie), soit l'arrêter selon les matrices de trafic. On peut aussi migrer des routeurs virtuels pour déplacer la charge loin des liens physiques congestionnés ou prenant en compte les prix de l'électricité dans des zones géographiquement différentes. Puisque les routeurs virtuels sont isolés, nous pouvons redémarrer un routeur virtuel sans affecter les autres services sur le routeur physique. En outre, les ressources disponibles peuvent être réaffectées au besoin à chaque routeur virtuel conduisant à économiser encore plus l'énergie des routeurs. Par conséquent, nous permettons la consolidation du réseau tout comme celle des serveurs et des centres de données en réduisant les traitements électriques sur les secteurs intermédiaires.

Il est bien sûr obligatoire de définir des horaires de veille appropriés par rapport aux interfaces des routeurs. Les instances de routeurs virtuels, situés dans des zones où le trafic est envoyé des utilisateurs de Cloud vers les services Cloud, peuvent probablement être mises en veille pendant un long intervalle de temps durant les périodes d'inactivité (par exemple la nuit, le week-end, les contraintes géographiques), ou pendant une courte période par rapport au comportement du trafic. Nous devons éviter la perte de paquets, mais aussi un grand nombre de

tentatives pour s'assurer que les instances de routeurs sont en veilles ou en sommeil. Afin de diminuer le nombre de tentatives Allumer/Eteindre sur les routeurs, nous considérons une métrique de contrainte sur les nœuds, qui permettra à notre algorithme de mise sous tension, d'instancier sur demande, les nœuds virtuels situés sur des nœuds physiques ayant la plus grande contrainte.

4.4.2 Évaluations analytiques de GCA

La virtualisation des réseaux est un moyen puissant pour exécuter plusieurs architectures ou des expériences en même temps sur une infrastructure partagée [Yu et al. 2008]. Cependant, l'utilisation efficace de ressources sous-jacentes nécessite des techniques efficaces pour l'intégration ou le mapping de chaque réseau virtuel à des nœuds et des liens spécifiques dans le réseau physique.

Un réseau virtuel intégré ou Virtual Network Embedding (VNE) en anglais gère le mappage efficace des nœuds virtuels sur les ressources du réseau physique. L'algorithme doit trouver un ensemble de nœuds virtuels N^V qui doivent être mappés sur un ensemble de nœuds physiques N^S et un ensemble de liens virtuels E^V qui doivent aussi être mappés sur un ensemble de liens physiques E^S . En d'autres termes, le réseau physique est un graphe $G^S = (N^S, E^S)$ et le nœud virtuel à incorporer est un graphe $G^V = (N^V, E^V)$. Par conséquent, chaque nœud physique (routeur physique) $n_i^S \in N^S$ a une valeur associée appelée indice de proportionnalité d'énergie et une capacité de puissance normalisée. EPI désigne la corrélation potentielle entre la puissance consommée par un routeur et sa charge de travail, et la puissance normalisée permet d'estimer la consommation d'énergie par bit pendant la transmission et la commutation. Un lien physiques $= (n_i^S, n_j^S) \in E^S$ entre les nœuds physiques $n_i^S, n_j^S \in N^S$ a une capacité de bande passante agrégée $BW(s)$.

Les nœuds et les liens ayant la contrainte maximale auront la plus grande priorité au cours du processus d'intégration.

En effet, la contrainte de liaison (LS) d'un lien $i \in E^S$ représente le taux d'utilisation de ce lien, tandis que la contrainte de nœud (NS) d'un nœud donné $i \in N^S$ donne sa consommation globale d'énergie. Les calculs de ces deux métriques sont décrits dans la suite. Par conséquent, selon le problème d'efficacité énergétique et la qualité de service demandée par la topologie du nœud virtuel (VN), un lien virtuel (VL) peut être divisé et intégré sur plusieurs liens physiques en fonction des contraintes fixées. Le fractionnement des chemins permet de réaliser des économies d'énergie en exploitant des nœuds physiques et des liens qui tendent vers une forte proportionnalité énergétique pour gérer le VN. La possibilité que les liens virtuels puissent être fractionnés sur plusieurs chemins physiques est en faveur de la réduction de la consommation d'énergie.

4.4.3 Méthode d'allocation de ressources

Etant donné que les ressources du réseau physique ne sont pas infinies, chaque nouveau réseau virtuel satisfait réduit les ressources résiduelles du réseau physique.

Par conséquent, la capacité résiduelle d'un lien $i \in E^S$ est définie dans l'équation 4:

$$R_E(t^-, i) = BW_{thres}(i) - \sum_{j=1}^{N_{vnet}} \sum_{l=1}^{N_{vlinkj}} ((LS_j^V(l_j) | l_j \ni i)) \quad (4)$$

où t^- est l'instant précédent immédiatement l'arrivée d'un réseau virtuel, N_{vnet} est le nombre de réseaux virtuels existant, N_{vlinkj} est le nombre de liens virtuels hébergés par le réseau virtuel j , LS_j^V qui représente la contrainte du lien virtuel, retourne la largeur de bande passante attribuée au lien virtuel l_j .

Si la largeur de la bande passante non consommée permet toujours de répondre aux besoins réels du réseau virtuel, elle doit être utilisée par le lien virtuel $l_j \in E^V$. Tant qu'il est adéquat, les ressources non consommées restent sur un lien $i \in E^S$ (ayant la

plus grande contrainte) et $i= 1 \dots N_{slink}$, où N_{slink} , est le nombre de liens du réseau physique. On devrait commencer par assigner un réseau virtuel sur ces liens.

Cette intégration est réalisée si uniquement nous ne dépassons pas le seuil de la bande passante selon la qualité de service(QoS) requise.

Selon notre heuristique VNE, les nœuds et les liens ayant une contrainte maximale auront la plus haute priorité lors du processus d'intégration. Par conséquent, en fonction de l'efficacité énergétique et de la qualité de service (QoS) requises par la topologie du réseau virtuel, un lien virtuel peut être fractionné et intégré sur plusieurs liaisons physiques en fonction des contraintes fixées.

En considérant le fractionnement des chemins, le problème de l'incorporation de la liaison peut être résolu en un temps polynomial, car il peut être réduit au problème de flux multi-commodité bien connu [Yu & al, 2008]. Par contre, pour chaque arrivé d'un réseau virtuel, nous prenons en compte la capacité résiduelle du lien physique. Nous attribuons un lien virtuel sur un lien physique que si nous ne dépassons pas le seuil de la bande passante selon les exigences de la qualité de service(QoS).

Le seuil de la bande passante est défini comme suit :

$$BW_{thres}(s) = \alpha \times BW(s) \quad (5)$$

Les valeurs de α sont comprises entre 0 et 1. Notre objectif est d'éviter la congestion du réseau, et donc respecter les contraintes de délais imposées par les applications réseau.

L'algorithme 1 évalue la contrainte de liaison de chaque lien $e \in E^S$ afin de déterminer l'ensemble des liens physiques pouvant être utilisés par les réseaux virtuels. A cet effet, nous supposons comme [Nogueira et al. 2011] que la contrainte de lien virtuel LS_j^V d'un lien donné l_j appartenant au j ème lien virtuel est :

$$LS_j^V(l_j) = BW(l_j) \quad (6)$$

$BW(l_j)$ renvoie la bande passante allouée au lien virtuel l_j qui représente le numéro de lien d'un lien virtuel appartenant au j ème $VN \in G^V$. Les lignes 1 à 8 du pseudo-code représenté dans l'algorithme 1 montrent cette procédure. Nous minimisons le nombre de liens physiques utilisés en chargeant chaque lien autant que possible. L'objectif principal est d'atténuer le nombre de nœuds virtuels activés et nous réalisons ainsi des économies d'énergie dans le transport.

Algorithme 1 Computing link stresses (Upon i 'th VN arrival)

Inputs:

- $G^S = (N^S, E^S)$: substrate topology;
- $LS(t_i^-, c), \forall c \in E^S$: current link stress;
- $R_E(t_i^-, e), \forall e \in E^S$: residual link capacity;
- $G_i^V = (N_i^V, E_i^V)$: VNet topology;

Output: Candidates substrate link that can be assigned to the set of virtual links E_i^V

```

1  foreach Link  $e \in E^S$  do
2    foreach Vnet  $j \in G^V$  do
3      foreach Link  $k \in E_j^V$  do
4        if Link  $k_j \supseteq$  Link  $e$  then
5           $LS(t_i^-, e) += BW(k_j)$ 
6        end if
7      end for
8    end for
9     $R_E(t_i^-, e) = BW_{thres}(e) - LS(t_i^-, e)$ 
10 end for
11 foreach Link  $k_i \in E_i^V$  do
12   foreach Link  $e \in E^S$  do
13     if  $R_E(t_i^-, e) \geq BW(k_i)$  then
14        $\wedge(t_i^-, k_i, e) = 1$ 
15     else
16        $\wedge(t_i^-, k_i, e) = 0$ 
17     end if
18      $LS(t_i^-, e) = \sum_{j=1}^{N_{vnet}} \sum_{l=1}^{N_{vlink_j}} ((LS_{j^V}(l_j) | l_j \supseteq e)) \wedge (t_i^-, k_i, e)$ 
19   end for
20 end for

```

La fonction \wedge définie dans l'équation 7 et utilisée dans l'équation 8, détermine si oui ou non le lien $i \in E^S$ est un chemin admissible à l'instant t .

$$\Lambda(t, l_j, i) = \begin{cases} 1 & \text{if } R_E(t^-, i) \geq BW(l_j) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

où t^- est l'instant précédent immédiatement l'arrivée d'un réseau virtuel, et $BW(l_j)$ est la bande passante demandée par un lien virtuel donné et numéroté l_j appartenant à l'arrivée d'une demande de la $j^{\text{ème}}$ topologie de réseau virtuel. Ensuite, la contrainte de liaison (LS) d'un lien $i \in E^S$, qui représente le taux d'utilisation du lien à l'instant t est donné par:

$$LS(t, i) = \sum_{j=1}^{N_{vnet}} \sum_{l=1}^{N_{vlinkj}} \left((LS_j^v(l_j) | l_j \ni i) \right) \Lambda(t, l_j, i) \quad (8)$$

Contrairement à [Baliga et al. 2011], notre contrainte de liaison définie dans l'équation 8 prend en compte la largeur de la bande passante préalablement allouée à chaque lien physique et la capacité de la bande passante résiduelle sur ce lien. En effet, nous exploitons la bande passante résiduelle dans le but de pondérer chaque lien candidat $i \in E^S$.

De la même manière, nous voulons savoir si un nœud virtuel n_j donné (par exemple un routeur virtuel) appartenant au $j^{\text{ème}}$ réseau virtuel est actif ou non et qu'il est en cours d'exécution sur un $i^{\text{ème}}$ nœud physique. En outre, la fonction Λ définie dans l'équation 9 ainsi que dans l'équation 10, tient compte de ces exigences à un instant t .

$$\Lambda(t, n_j, i) = \begin{cases} 1 & \text{if } (n_j \ni i) \wedge (n_j \text{ is active}) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

L'algorithme 2 évalue la contrainte des nœuds physiques capables d'héberger l'ensemble des nœuds virtuels. Selon notre formule de contrainte de nœud (NS) (Eq. 6), nous considérons la proportionnalité énergétique de chaque routeur physique

(métriques EPI et NPower) et sa consommation d'énergie instantanée ($Power_{switch}$).

La NS d'un nœud donné $i \in N^S$ donne sa consommation énergétique globale.

La NS du $i^{\text{ème}}$ nœud à l'instant t est:

$$NS(t, i) = \frac{\sum_{j=1}^{N_{vnet}} \sum_{n=1}^{N_{vnode_i}} \wedge(t, n_j, i)}{\frac{1}{EPI} \times NPower} \times Power_{switch} \quad (10)$$

Algorithme 2 Computing node stresses (Upon i 'th VN arrival)

Inputs:

$G^S = (N^S, E^S)$: substrate topology;

$\mathcal{NS}(t_i^-, n), \forall n \in N^S$: current node stress;

$G_i^V = (N_i^V, E_i^V)$: VNet topology;

Output: Candidates substrate node that can host the set of virtual nodes N_i^V

```

1 foreach Node  $n \in N^S$  do
2    $\mathcal{NS}(t_i^-, n) = \frac{\sum_{j=1}^{N_{vnet}} \sum_{n=1}^{N_{vnode_j}} \wedge(t_i^-, n_j, n)}{\frac{1}{EPI} \times NPower} \times Power_{switch}$ 
3 end for
4 foreach Node  $v \in N_i^V$  do
5   foreach Node  $n \in N^S$  do
6     if SatisfiesConstraints( $v, n$ ) then
7        $v.Candidates.Add(n)$ 
8     end if
9   end for
10 end for

```

En effet, le numérateur de la première partie de l'équation 10 renvoie le nombre total d'instances de nœuds virtuels s'exécutant sur le nœud $i \in E^S$, où N_{vnode_i} représente le nombre de nœuds qui forment le $j^{\text{ème}}$ réseau virtuel.

4.4.3 Algorithme d'allocation de ressources

Après avoir pris en compte les problèmes d'hétérogénéité entre les nœuds et les liens en termes de consommation d'énergie et de capacité de la bande passante,

Pour réduire la consommation d'énergie pendant les commutations, notre algorithme d'allocation de ressources prend en considération:

- (i) l'affectation des nœuds et des liens sont effectués simultanément;
- (ii) un contrôle d'admission pour vérifier si l'arrivée du réseau virtuel pourra être satisfaite ou non;
- (iii) la possibilité que les liens virtuels puissent être divisés sur plusieurs chemins physiques en faveur de la réduction de la consommation d'énergie ainsi qu'une maximisation du trafic envoyé par les liens, tout en minimisant l'utilisation des ressources du réseau physique.

Algorithme 3 Energy-aware embedding algorithm (Upon i 'th VN arrival)

Inputs:

$\mathbb{G}^S = (\mathbb{N}^S, \mathbb{E}^S)$: substrate topology;

$\mathbb{G}_i^V = (\mathbb{N}_i^V, \mathbb{E}_i^V)$: VNet topology;

Output: VNet_{Embed} (Embedded Virtual Network)

- 1 Rank virtual nodes $v \in \mathbb{N}_i^V$ according to their number of candidates $n \in \mathbb{N}^S$
- 2 Firstly assign nodes $v \in \mathbb{N}_i^V$ that have fewer substrate candidate nodes
- 3 **foreach** Node $v \in \mathbb{N}_i^V$ **do**
- 4 **foreach** Link $k \in \mathbb{E}_i^V$ connected to v **do**
- 5 LinkedVNode = GetLinkDestination(k)
- 6 **foreach** SourceCandidate s in $v.Candidates$ **do**
- 7 CostNRG(s, d) = 0
- 8 **foreach** DestCandidate d in $LinkedVNode.Candidates$ **do**
- 9 PathCost(s, d) = CostNRG(s, d)
- 10 **end for**
- 11 CostNRG(s, d) = CostNRG(s, d) + $\frac{\sum_{e \in L(s, d)} \mathcal{L}S(t_i^-, e) \times \mathcal{N}S(t_i^-, s)}{\text{Count}(v.Candidates)}$
- 12 **end for**
- 13 **end for**
- 14 $v.Embed = s$ | Path-cost is minimized with respect to power savings
- 15 **end for**

Après le calcul des contraintes de liaison et de nœud, un classement de tous les nœuds suivant leurs contraintes est effectué. Le nœud $i \in N^S$ ayant la contrainte la plus grande sera le premier potentiel nœud parmi l'ensemble des candidats pour accueillir un nœud virtuel. Le coût du trajet de chaque chemin potentiel entre le candidat source $s \in v.Candidates$ vers toutes les autres destinations $d \in LinkedVNode.Candidates$ est estimé dans l'algorithme 3 entre les lignes 3 et 12:

- (i) La contrainte de nœud de chaque nœud physique qui agit comme potentiel nœud source par rapport au chemin qui relie un nœud virtuel fixé est désigné comme destination.
- (ii) La contrainte de lien de l'ensemble des liens physiques $L(s, d)$ qui peuvent former le chemin vers le potentiel nœud physique qui héberge le nœud virtuel.
- (iii) Le nombre de nœuds physiques qui répondaient à des contraintes pour l'hébergement d'un nœud virtuel fixé afin de traiter équitablement les nœuds physiques possédant un nombre différent de liens virtuels possibles. Ainsi, le nœud candidat qui offre un chemin avec des économies d'énergie (C'est-à-dire la plus petite valeur de $costNRG$) est sélectionné (ligne 14).

4.5 Évaluation de performance

4.5.1 Paramètres de simulation

(a) Cisco GSR 12008

Line card (LC) type	Power	Qty	EPI (in %)	NPower
4 Port GE	92	2	43.12	0.058
4 Port OC-12/POS	72	1		
1 Port OC-48/POS	70	1		

(b) Cisco 7507

Line card (LC) type	Power	Qty	EPI (in %)	NPower
1 Port FE	26	3	42.77	0.085
4 Port GE	30	1		
1 Port 1.544 Mbps DS1	49	1		

Tableau 4.1: Résumé de la consommation d'énergie des routeurs

Le tableau 4.1 montre la puissance moyenne consommée par les différents composants de deux routeurs donnés tels que Cisco GSR 12008 et Cisco 7507. Les consommations des interfaces des routeurs (Lines Cards) illustrées dans les tableaux 4.1 sont dérivées de [Chabarek et al 2008]. Dans le tableau 4.1, les colonnes « Power » et « NPower » sont respectivement exprimées en « Watts » et « Watts/Mbps ». La colonne « Qty » désigne le nombre de type d'interfaces installés sur chaque routeur. Les termes « FE » et « GE » signifient respectivement Fast Ethernet et Gigabit Ethernet. À l'état Idle (inactif), GSR 12008 consomme environ 430 Watts, tandis que le 7507 consomme environ 270 Watts pour le même état. Ici, l'état Idle se réfère à la consommation d'énergie du châssis d'un routeur donné. Les valeurs des deux paramètres « EPI » et « NPower » sont calculées sur la base des formules illustrés dans la section 4.2. Les valeurs des indices EPI correspondant aux deux routeurs sont assez proches, mais il existe un gap important selon la métrique de normalisation de la puissance « NPower ». La tendance observée est que GSR a une meilleure proportionnalité d'énergie par rapport au routeur 7507.

Nous avons implémenté notre algorithme d'allocation de ressources basé sur l'économie d'énergie en Java. Notre simulateur d'événements discrets considère les mêmes paramètres décrits dans [Yu et al. 2008] au cours des nombreuses expériences de simulations. De ce point de vue, le réseau physique est une topologie constituée de 100 nœuds et de 354 liens, il est généré de manière aléatoire par l'outil GT-ITM. Les nœuds physiques sont choisis de manière aléatoire comme étant des routeurs de type GSR ou 7507. Nous supposons que les requêtes de la topologie des réseaux virtuels arrivent suivant un processus de Poisson de paramètre $\lambda = 5$ par unité de temps en moyenne. Ces requêtes sont regroupées pendant une fenêtre coulissante fixe égale à 10 unités de temps et seront exécutées durant la prochaine fenêtre comme dans [Yu et al. 2008]. Dans cette configuration, lorsque les contraintes de ressource des nœuds virtuels sont satisfaites, les requêtes sont traitées avec un temps de service exponentiel de paramètre " μ " dont la valeur moyenne est égale à 10 unités de temps. Dans le cas contraire, les demandes ayant échoué pour l'instanciation des nœuds virtuels seront reportées en file d'attente à un moment approprié.

Par ailleurs, le nombre de nœuds virtuels est uniformément réparti entre 2 et 10 et chaque paire de nœuds virtuels est connectée au hasard avec une probabilité de 0,5. Cela signifie que pour un réseau virtuel de n nœuds, nous avons $\frac{n(n-1)}{4}$ liaisons en moyenne [Yu et al. 2008]. Contrairement aux travaux précédents [Belbekkouche et al. 2012], nous avons mis en place un mécanisme de contrôle d'admission. Nous réalisons notre simulation pendant 500 unités de temps, ce qui correspond en moyenne à 2500 requêtes environ dans une instance de simulation. Les nœuds virtuels peuvent demander une bande passante d'une capacité 100 Mbps ou de 1 Gbps.

4.5.2 Résultats

La figure 4.3 montre la fonction de distribution cumulative (CDF) du nombre moyen d'interfaces activés par nœud pour différents taux d'arrivée (λ) des nœuds virtuels. À

noter que λ varie entre 5 et 40, également pendant cette expérience, nous avons fixé la fenêtre de temps à 10 unités de temps. Nous remarquons sur la figure 4.3 l'efficacité de notre algorithme d'allocation de ressource basé sur l'énergie. En effet, notre heuristique de réseau virtuel intégré (VNE) réduit l'utilisation des ressources en n'activant qu'un nombre minimal d'interfaces par rapport à la charge du réseau. Pour $\lambda = 5$, en moyenne 20% des nœuds ont zéro interface activée, ce qui signifie une économie d'énergie importante. D'autre part, le nombre d'interfaces utilisées par nœud croît en fonction de l'augmentation du taux d'arrivée des requêtes des réseaux virtuels.

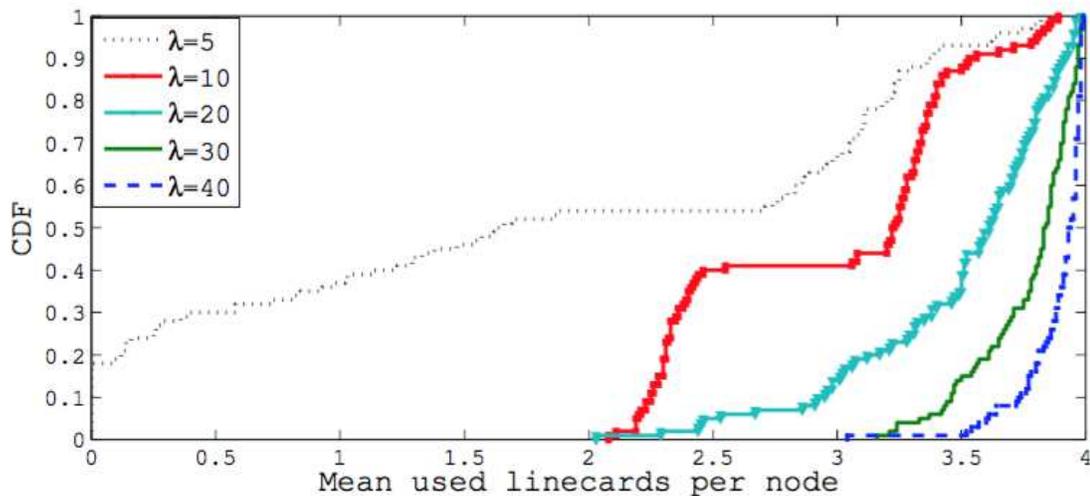


Figure 4.3: Fonction de distribution cumulative des lignes de carte activées en fonction de la charge

Afin d'évaluer les performances de notre algorithme, nous avons implémenté l'algorithme EA-VNE défini dans [Su et al. 2012]. Nous avons considéré les mêmes paramètres d'évaluation de performance à l'exception du nombre de nœuds physiques fixé à 100. La figure 4.4 montre l'énergie consommée par l'ensemble du réseau lorsque les nœuds virtuels sont intégrés selon des contraintes d'énergie. Un réseau non-efficace signifie un réseau où tous les dispositifs de mise en réseau fonctionnent avec leurs énergies maximales. Il est facile de vérifier qu'après 10 temps window, la consommation d'énergie de GCA-VNE est dans son état stable. La consommation

moyenne d'énergie pour GCA-VNE et pour EA-VNE sont respectivement 37 000 Watts et 19 6000 Watts. Alors que l'énergie consommée par un réseau non efficient est toujours égale en moyenne à 63 000 Watts pour GCA-VNE et 300 000 Watts pour EA-VNE. Ainsi, GCA-VNE et EA-VNE peuvent respectivement économiser jusqu'à 41% et 35% de l'énergie par rapport à un réseau non efficient. Ce gain est dû au fait que GCA-VNE favorise l'équilibrage de charge.

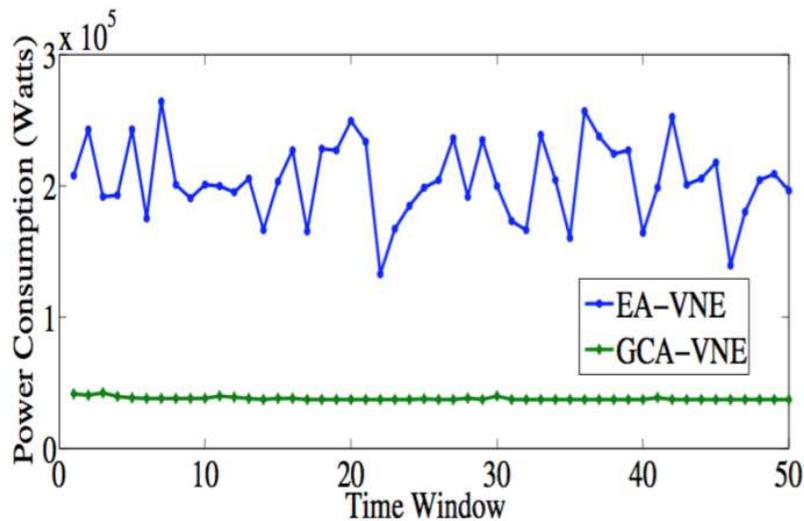


Figure 4.4: Consommation globale du réseau

La figure 4.5 illustre le pourcentage de nœuds virtuels rejetés selon le taux moyen λ variant entre 5 et 50. Il faudra noter que les requêtes des nœuds virtuels sont rassemblées pendant une fenêtre de temps fixe égale à 10 unités de temps et traitées à la fenêtre suivante. Le pourcentage de nœuds virtuels rejetés dans EA-VNE est supérieur à celui de GCA-VNE. Lorsque λ est égal à 5, tous les nœuds virtuels arrivant sont acceptés pour l'approche GCA-VNE, contrairement à EA-VNE où 8% des nœuds virtuels sont rejetés. On remarquera que pour des valeurs de λ supérieures à 40, le taux de blocage reste stable pour l'approche GCA-VNE.

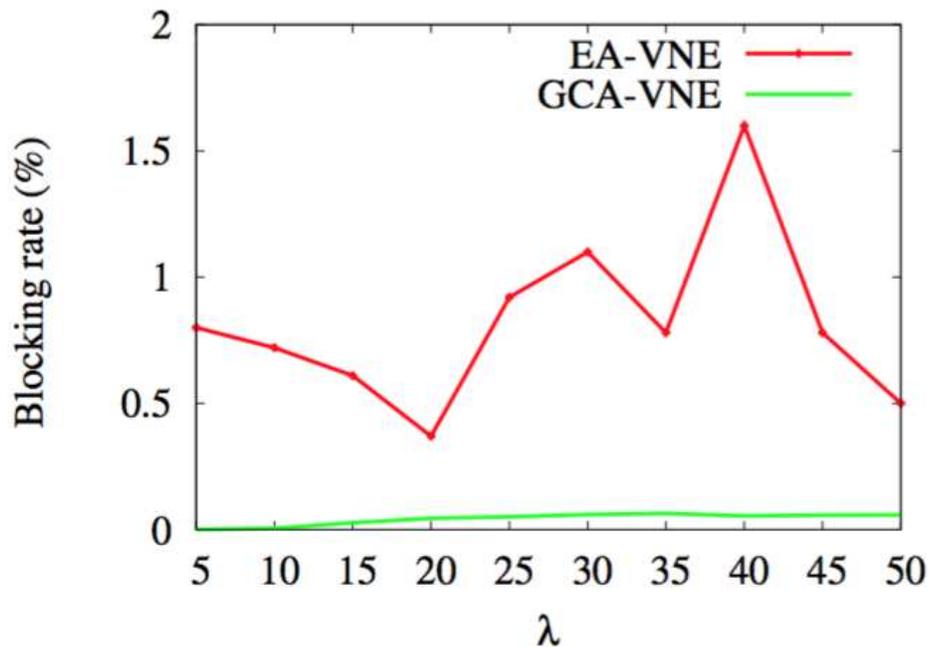


Figure 4.5: Pourcentage des nœuds virtuels bloqués

La figure 4.6 montre les contraintes de nœud de l'ensemble des nœuds du réseau ainsi que le nombre de réseaux virtuels mappés sur le réseau physique en fonction du temps. L'axe vertical gauche donne la contrainte de nœud calculée pour chaque nœud alors que l'axe vertical droit donne le nombre de réseaux virtuels intégrés. En effet, lorsque les réseaux virtuels quittent le réseau physique, les ressources utilisées sont libérées. Par conséquent, les contraintes de nœud des nœuds physiques qui ont hébergé ces réseaux virtuels sont réduites. La courbe bleue libellée "Node Stresses" représente le pourcentage et la médiane des contraintes de nœud globales des nœuds du réseau pendant chaque temps window. Les valeurs médianes des contraintes de nœuds montrent une corrélation par rapport à l'affectation des réseaux virtuels (courbe verte libellé "Virtual Networks Number"). Dans ce sens, les pics notés par rapport aux contraintes de nœud correspondent à l'instant où l'on note le nombre maximum de réseaux virtuels intégrés. En effet, notre algorithme instancie en premier lieu les routeurs virtuels sur les nœuds physiques ayant la plus grande contrainte.

Pendant plusieurs temps window aussi, nous remarquons que 50% des routeurs sont à l'état Idle (Node Stresses = 0) par rapport à leurs consommations d'énergie.

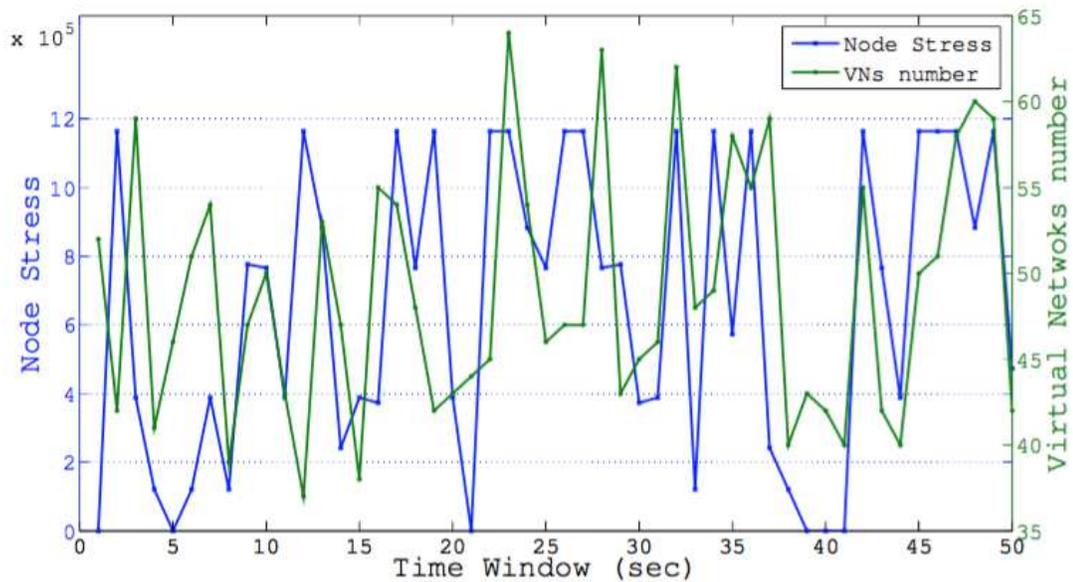


Figure 4.6: Nombre de réseaux virtuels mappés et médiane des contraintes de nœud

La figure 4.7 montre le pourcentage de réseaux virtuels rejetés en fonction de la moyenne du taux d'arrivée λ qui varie entre 5 et 50. Il faudra noter que les demandes des réseaux virtuels sont regroupées pendant un temps window fixe qui varie entre 10 et 50 unités de temps et traitées au temps window suivant. Chaque courbe montre une simulation donnée en fonction d'une unité de temps window fixe. Le pourcentage de réseaux virtuels rejetés augmente en fonction des valeurs du temps window ainsi que du taux d'arrivée. Lorsque λ est égal à 5 et que le temps window est fixé à 10, toutes les arrivées de réseaux virtuels sont acceptées. Notons que pour des valeurs de λ supérieure à 40, le taux de blocage est sensiblement stable.

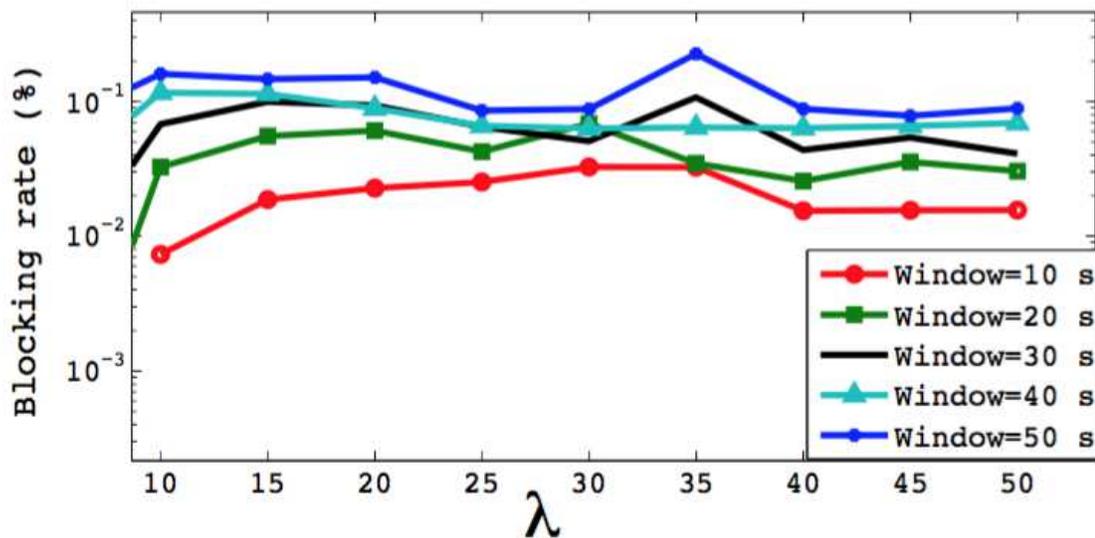


Figure 4.7: Réseaux virtuels bloqués

4.6 Conclusion

La virtualisation des routeurs physiques impliqués dans le transport des données contribue à la réduction de la consommation d'énergie par les routeurs ainsi que l'énergie globale consommée pendant le transport. Dans cette contribution, nous avons proposé un algorithme d'allocation de ressources qui prend en compte le partage de chemins et le support de réseaux virtuels et physiques hétérogènes. De plus, GCA-VNE favorise l'attribution simultanée des liens et des nœuds. Les résultats obtenus montrent que notre algorithme d'allocation de ressources (GCA-VNE) peut économiser jusqu'à 41% du coût énergétique par rapport à un réseau non efficace. Dans le chapitre suivant, nous nous intéressons à la réduction de la consommation d'énergie au cœur des centres de données.

**Chapitre 5 : Gestion de la qualité de service dans les centres
de données (GREENPOD)**

5.1 Introduction

La demande croissante de services Internet et de Cloud a considérablement augmenté le nombre de serveurs et de Switch opérant dans les grands centres de données. Cela entraîne également une augmentation de la consommation d'énergie dans ces centres de données massifs. Malgré que les centres de données soient mis en place à des fins et applications différentes, la structure de base est similaire. Le fournisseur de services vise à réaliser des revenus élevés tout en garantissant les accords de niveau de service (SLA).

Pour réduire la consommation d'énergie, il est important de développer un moyen efficace d'éteindre les serveurs inutilisés pendant les heures de charge de travail faible [Grid 2010], [Gandhi et al. 2010].

Dans notre proposition [Kasse et al. 2018], nous avons utilisé la topologie Fat-tree K-aray. Nous proposons de trouver un compromis entre la réduction de la consommation d'énergie et un temps d'attente acceptable. Les serveurs étant divisés en un nombre de groupes dépendant de la valeur de " K ", la décision de les activer ou de les désactiver est basée sur un nombre de seuils qui doivent être correctement dimensionnés en fonction du niveau de charge exprimé par la longueur de la file d'attente.

5.2 Contexte de la proposition

La consommation d'énergie des centres de données ne cesse d'augmenter ces dernières années. Le chercheur suédois et expert en analyse du cycle de vie chez Huawei, Anders Andrae [Andrea 2017], prédit que d'ici 2025, les centres de données représenteront la plus grande part des TIC dans la production mondiale d'électricité (33%), suivis des smartphones (15%) des réseaux (10%) et de la télévision (9%).

En 2010, leur consommation représentait environ 1,3% de la consommation totale d'électricité dans le monde [Kooimey 2011] et leur consommation d'énergie peut être triplée de 2010 à 2020 si leur efficacité énergétique reste au niveau de 2010 [Shehabi

et al. 2016]. Les deux principaux consommateurs d'énergie des centres de données étant son équipement informatique plus particulièrement les serveurs, et ses systèmes de refroidissement [Dayarathna et al. 2016], les technologies visant à réduire la consommation énergétique de ces deux composants dans les centres de données sont en développement rapide ces dernières années [Ni & Bai 2017].

Plusieurs auteurs proposent de nouvelles architectures pour les centres de données dont le déploiement est moins coûteux et offre plus de résilience [Baliga et al. 2011] - [Fischer et al. 2013]. Des techniques existent et visent principalement à réduire la consommation d'énergie des serveurs [Mardani et al. 2015]. Actuellement, les fournisseurs de services dans le cloud trouvent d'autres moyens de réduire la quantité de carbone de leur infrastructure [Mardani et al. 2015]. Les grands fournisseurs du Cloud tels que Google, Amazon, Microsoft et IBM espèrent alimenter leurs centres de données en utilisant des sources d'énergie renouvelables [Sing et al. 2017]. Les futurs centres de données Cloud sont tenus de fournir des services avec un minimum d'émissions de carbone et de chaleur sous forme de gaz à effet de serre [Buyya&Sing 2018].

Des travaux ont étudié des cas dynamiques de gestion des serveurs dans le Cloud [Gandhi et al. 2010], [Ghamkhari & Mohsenian-Rad 2013], [Guenter et al. 2011] qui activent et désactivent les serveurs en fonction de la charge actuelle.

Notre proposition est comparée aux travaux de [Dan et al. 2015]. En effet, dans leurs propositions, ils divisent le nombre de serveurs en trois groupes. Le premier est toujours actif et les deux autres sont utilisés dynamiquement en fonction des seuils d'activation basés sur la taille de la file d'attente. Ils travaillent avec une architecture tree-tiers. Dans le cas d'un grand centre de données, constitué de milliers de serveurs, il peut arriver régulièrement qu'on active avec leur proposition, un groupe de serveurs pour traiter quelques requêtes. Le nombre de serveurs d'un groupe dans de tels

centres de données peut être très grand. Dans certains cas, quelques serveurs du groupe peuvent régler le problème.

Nous utilisons la topologie Fat-tree et nous démarrons avec le quart des serveurs et Switch c'est à dire $\frac{K}{4} * \left(\frac{K}{2}\right)^2$ serveurs et $\frac{K}{4} * K$ Switchs. Nous utilisons des modèles mathématiques et de simulation pour réduire la consommation d'énergie dans les centres de données.

Les autres groupes (un groupe de serveurs constitue le nombre de serveurs dans un Pod) sont activés et désactivés en fonction des seuils d'activation basés sur la taille de la file d'attente. Pour éviter des pertes de requêtes, on vérifie si les serveurs à désactiver ont terminé leur traitement, au cas échéant on les désactive sinon on replace les requêtes restantes en tête de file avant de désactiver les serveurs. Nous montrons par la suite que nous économisons plus d'énergie que la proposition de Dan et al. mais aussi nous avons un gain important en énergie par rapport à un centre de données par défaut.

5.3 Présentation du modèle

Le modèle de centre de données que nous considérons est un K-array Fat-Tree et nous démarrons initialement le quart des Pod. Nous avons utilisé Fat-Tree dont la topologie est illustrée sur la figure 5.1 parce que :

- Fat-Tree a la même bande passante au niveau de toutes les bisections;
- Chaque couche a la même bande passante agrégée;
- Qu'elle est construite en utilisant des dispositifs avec une capacité uniforme;
- Chaque port supporte la même vitesse que l'hôte final;
- Tous les appareils peuvent transmettre à la vitesse de la ligne si les paquets sont; uniformément distribués le long des chemins disponibles;

- Elle offre une grande évolutivité: le commutateur k-port prend en charge $\frac{K^3}{4}$ serveurs.

En fonction de la demande, nous activerons les autres Pod mis initialement hors tension et si la demande diminue, nous les désactiverons avec les ports des Switchs Core correspondants. Pour parvenir à cela, nous allons fixer des seuils d'activation et de désactivation.

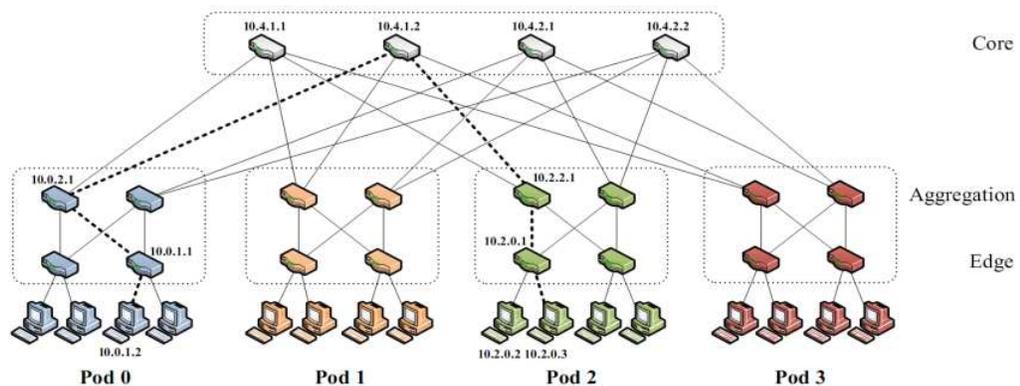


Figure 5.1: Topologies Fat-tree

Contrairement aux propositions [Schwartz et al. 2012] et [Dan et al. 2015], nous utiliserons la topologie Fat_Tree permettant d'économiser plus d'énergie par rapport aux autres topologies. Nous ne limiterons pas non plus le nombre de groupes de serveurs à ajouter à un ou deux, ce nombre dépendra de "K", c'est-à-dire du nombre de Pod, mais aussi les Switchs non utilisés seront éteints. Notre objectif est d'évaluer la quantité d'énergie consommée dans le centre de données depuis les routeurs core jusqu'aux serveurs et la quantité d'énergie économisée quand les serveurs et Switchs non utilisés sont mis hors tension. Initialement, seuls les switch core et le quart des Pods sont actifs, ce qui permet d'économiser aussi bien de l'énergie au niveau des serveurs mais aussi au niveau des Switchs.

Considérons le modèle de la figure 5.2 suivante constitués de serveurs homogènes.

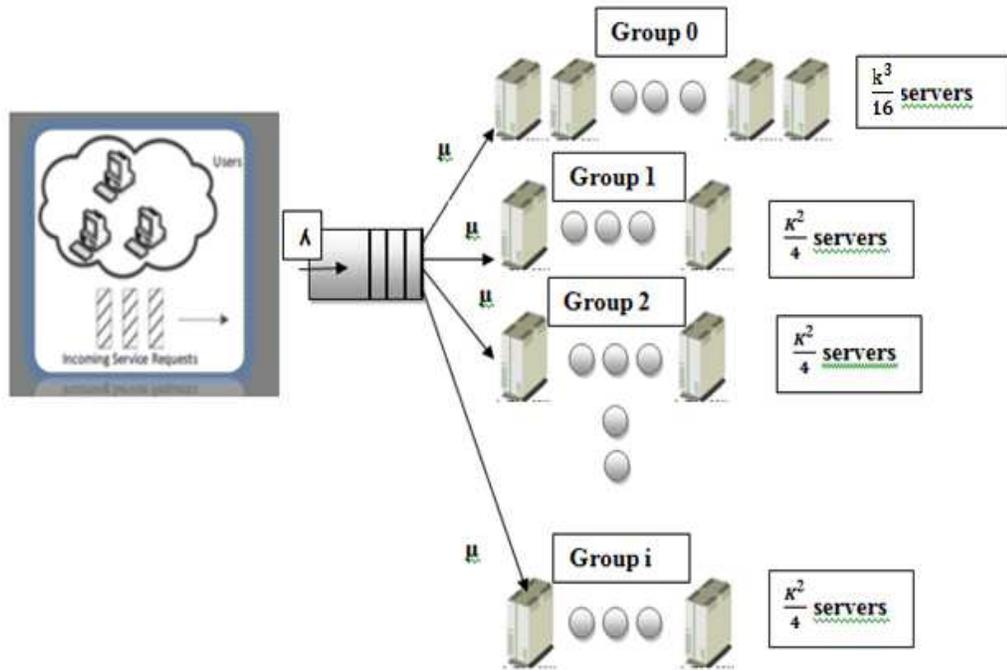


Figure 5.2: Modèle du système

Nous supposons que les requêtes arrivent suivant une distribution poissonnienne de taux λ et que chaque serveur exécute une seule requête à la fois avec un temps de service suivant une distribution exponentielle de taux μ . Le système peut être modélisé comme une file $M/M/\frac{K^3}{4}$, avec $\frac{K^3}{4}$, le nombre total de serveurs dans une topologie fat-tree k-array.

Quand une requête arrive dans le système, elle est allouée à un serveur à l'état idle parmi les $\frac{K^3}{16}$ constituant le quart des serveurs qui restent toujours allumés quel que soit le nombre de requêtes dans le système. Si les serveurs sont occupés, la requête est mise en attente jusqu'à ce qu'un serveur soit disponible. Le comportement d'un serveur de ce groupe est illustré dans la figure 5.3.

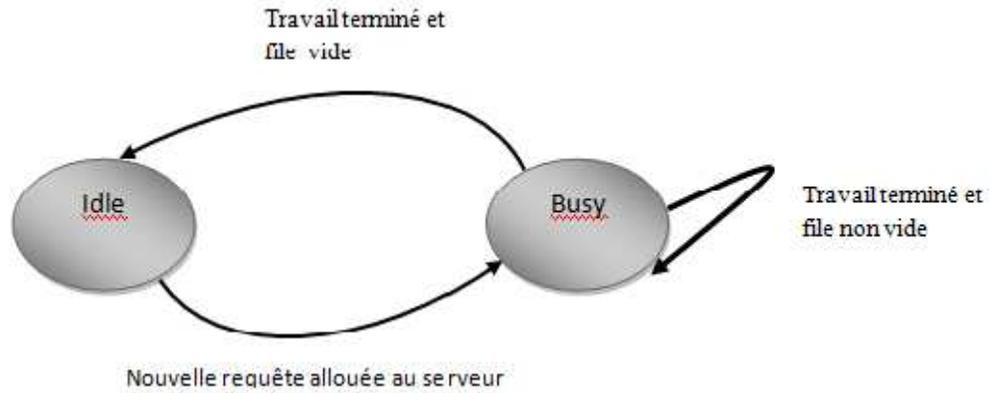


Figure 5.3: Différents états d'un serveur du premier groupe

Ce premier groupe de serveurs se comporte comme un centre de données par défaut. Soit P une variable aléatoire qui détermine le nombre de requêtes dans le système et P_j la probabilité qu'il y ait j requêtes dans le système. Initialement les $\frac{K}{2}$ Switchs edge et agrégations dans chaque Pod sont sous tension ainsi que les $\left(\frac{k}{2}\right)^2$ Switch core. Si $j < \frac{K^3}{4}$ requêtes, les j serveurs sont occupés et consomment e_{busy} et les $\frac{K^3}{4} - j$ serveurs sont inoccupés et consomment e_{idle} . La borne supérieure de la consommation d'énergie est :

$$E_{max} = \left[\sum_{j=0}^{\frac{k^3}{4}} P_j * (j * e_{busy} + \left(\frac{k^3}{4} - j\right) e_{idle}) + e_{busy} * \sum_{j=\frac{k^3}{4}+1}^{\infty} P_j \right] + \frac{k^2}{2} (e_{busyS_e} + e_{busyS_a}) + \frac{k^2}{2} e_{busyS_c} \quad (11)$$

Avec

e_{busy} l'énergie du serveur en traitement

e_{idle} l'énergie du serveur à l'état idle

e_{busyS_a} l'énergie du switch aggregation

e_{busyS_e} l'énergie du switch edge

e_{busyS_c} l'énergie du switch core

La borne minimale de la consommation d'énergie correspond au cas où les serveurs inoccupés sont automatiquement éteints et consomment e_{off} .

$E_{min} =$

$$\sum_{j=0}^{\frac{k^3}{4}} P_j * \left(j * e_{busy} + \left(\frac{k^3}{4} - j \right) e_{off} \right) + e_{busy} * \sum_{j=\frac{k^3}{4}+1}^{\infty} p_j + \left(nbP * \frac{k}{2} \right) (e_{busySe} + e_{busySa}) + \left(\frac{k^2}{2} - nbP * \frac{k}{2} \right) e_{off} + \frac{k^2}{8} * e_{busySc}$$

(12)

Avec nbP=nombre de Pod actifs.

Le comportement d'un serveur parmi ceux qui sont réservés est illustré à la figure 5.4. Dans les groupes restants, les serveurs sont initialement éteints. Ils sont démarrés et passent à l'état Idle dès que leurs seuils d'activation sont atteints. Le passage à l'état occupé (Busy) se produit dès qu'on leur affecte une requête. Ils y restent jusqu'à ce que leurs seuils de désactivation soient atteints.

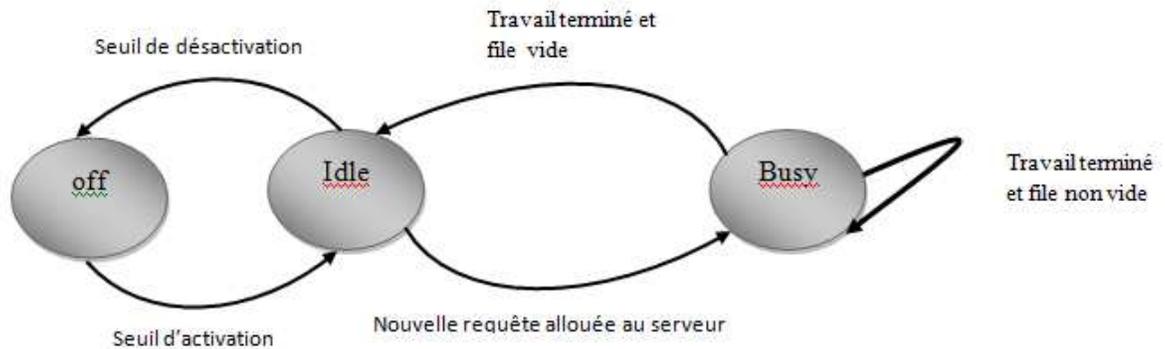


Figure 5.4: Comportement d'un serveur des groupes restants

Dans notre approche, nous avons " i " seuils d'activation et " i " seuils de désactivation. Le système est divisé en " K " groupes de serveurs (Pod). Initialement, les $\frac{K}{4}$ groupes de serveurs sont désactivés et nous avons un nombre de serveurs de base actifs " S_{a0} "

égal à $\frac{K}{4} * \left(\frac{K}{2}\right)^2$ (nombre de POD actifs * nombre de serveurs dans chaque POD) ce qui donne $\frac{K^3}{16}$ serveurs actifs. Le $i^{\text{ème}}$ groupe de serveur sera activé si le nombre de requêtes dans la file dépasse le seuil θ_i , avec $\theta_i \in [S_{a_{i-1}}, S_{a_{i-1}} + \left(\frac{K}{2}\right)^2]$ et $S_{a_{i-1}}$, le nombre de serveurs actifs au niveau $i-1$. Autrement dit dès qu'il y a $S_{a_{i-1}} + \theta_i$ requêtes dans le système, le groupe i est activé. Ce groupe de serveurs restera actif jusqu'à ce que le nombre total de requêtes dans la file soit égal à $S_{a_{i-1}}$.

Exemple : Le deuxième groupe de serveurs sera activé si le nombre de requêtes dans la file dépasse le seuil θ_1 avec $\theta_1 \in \left[\frac{K^3}{16}, \frac{K^3}{16} + \left(\frac{K}{2}\right)^2\right]$ c'est-à-dire dès qu'il y ait $\frac{K^3}{16} + \theta_1$ requêtes dans le système. Ce groupe de serveurs sera désactivé si le nombre de requêtes en attente dans la file est inférieur ou égal à $\frac{K^3}{16}$.

Le tableau 5.1 suivant, présente pour chaque niveau, son seuil d'activation et de désactivation ainsi que le nombre de serveurs actifs.

Niveau	Nombre de serveurs actifs S_i	Seuil d'activation θ_i (Nombre de requêtes)		Seuil de désactivation (Nombre de requêtes)
		En attente (θ_i)	Système($S_{i-1}+\theta_i$)	En attente
0	$S_0 = \frac{K^3}{16}$	-	-	-
1	$S_1 = S_0 + \frac{K^2}{4} = \frac{K^3}{16} + \frac{K^2}{4}$	$\theta_1 \in [S_0, S_0 + \frac{K^2}{4}]$	$S_0 + \theta_1$	S_0
2	$S_2 = \frac{K^3}{16} + \frac{K^2}{2}$	$\theta_2 \in [S_1, S_1 + \frac{K^2}{4}]$	$S_1 + \theta_2$	S_1
3	$S_3 = \frac{K^3}{16} + \frac{3K^2}{4}$	$\theta_3 \in [S_2, S_2 + \frac{K^2}{4}]$	$S_2 + \theta_3$	S_2
...
i	$S_i = \frac{K^3}{16} + i \frac{K^2}{4}$	$\theta_i \in [S_{i-1}, S_{i-1} + \frac{K^2}{4}]$	$S_{i-1} + \theta_i$	S_{i-1}

Tableau 5.1: Nombre de serveurs actifs et les seuils d'activation et de désactivation à chaque niveau

5.3.1 Modèle analytique

Dans cette partie, nous utilisons le modèle de la théorie des files d'attente pour la réduction de la consommation d'énergie dans les centres de données.

Les files d'attente sont essentiellement caractérisées par 3 paramètres notés sous la forme A/S/N (notation de Kendall) où:

A Représente la loi associée aux arrivées dans la file.

S Loi associée au temps de service des serveurs.

N Nombre de serveurs.

Quand il n'y a pas d'autres paramètres, les autres caractéristiques sont par défaut:

- Capacité maximale de la file d'attente infinie.
- Nombre d'utilisateurs infini.
- Discipline de service FIFO.

Les lois sont classées parmi:

"M" Loi exponentielle (Markovienne).

"D" Loi déterministe.

"G" Loi générale quelconque.

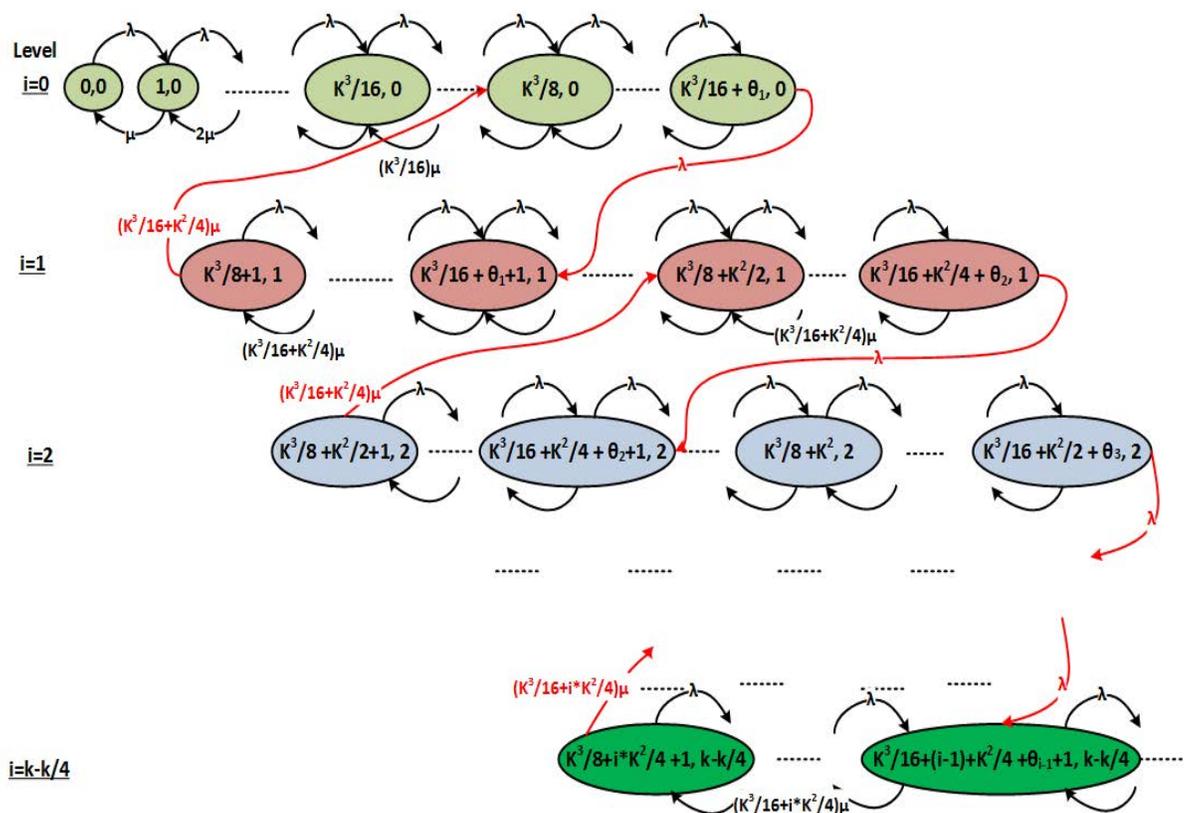


Figure 5.5: Chaîne de Markov du modèle proposé

La figure 5.5 représente la chaîne de Markov de notre modèle proposé. Pour représenter l'espace d'état de notre système, notre modèle est considéré comme un ensemble de systèmes S_i^j où "j" est le nombre de requêtes dans le système.

$$i = \begin{cases} 0 & \text{si seuls les serveurs de base } S_{a_0} \text{ sont activés} \\ 1, 2, \dots, 3 * \frac{k}{4} & \text{si les serveurs réservés sont activés} \end{cases}$$

Le système active le groupe de serveurs i s'il y a au moins θ_i requêtes dans la file d'attente c'est-à-dire s'il ya $S_{a_{i-1}} + \theta_i$ requêtes dans le système. Ce groupe i de serveurs est désactivé si le nombre de requêtes en attente est égal à $S_{a_{i-1}}$.

Soit P une variable aléatoire, P_j^i , la probabilité qu'il ait j requêtes dans le système à l'état i .

Nous commençons par formuler les premières équations d'état du système S_0^j qui correspond à l'état du système où aucun groupe de serveurs réservés n'est encore activé.

$$j\mu P_j^0 = \lambda P_{j-1}^0 \quad 0 < j < S_{a_0}$$

$$S_{a_0} \mu P_j^0 = \lambda P_{j-1}^0 S_{a_0} \quad S_{a_0} \leq j < 2S_{a_0}$$

$$S_{a_0} \mu P_j^0 + S_{a_1} \mu P_{2S_{a_0}}^1 = \lambda P_{j-1}^0 2S_{a_0} \quad S_{a_0} + \theta_1 \leq j < S_{a_0} + \theta_1$$

Les équations des états des systèmes S_i^j (état du système où les groupes de serveurs i sont activés) sont les suivantes :

$$P_{our i} \leftarrow 1 \text{ à } 3 * \frac{k}{4}$$

$$S_{a_i} \mu P_j^i = \lambda P_{j-1}^i + \lambda P_{S_{a_{i-1}} + \theta_i}^{i-1} 2S_{a_{i-1}} \quad S_{a_{i-1}} + \theta_i < j < S_{a_{i-1}} + \theta_i$$

$$S_{a_i} \mu P_j^i = \lambda P_{j-1}^i S_{a_{i-1}} + \theta_i \quad S_{a_{i-1}} + \theta_i \leq j < 2S_{a_i}$$

$$S_{a_i} \mu P_j^i + S_{a_{i+1}} \mu P_{2S_{a_i}}^1 = \lambda P_{j-1}^i 2S_{a_i} \quad 2S_{a_i} \leq j < S_{a_i} + \theta_{i+1}$$

$$\frac{k^3}{4} \mu P_j^2 = \lambda P_{j-1}^2 j \geq \frac{k^3}{2}$$

5.3.2 Calcul des probabilités

$$P_j^0 = \left(\frac{\lambda}{\mu} \right)^j \frac{1}{j!} P_0^0 \quad \text{Pour } 0 < j \leq S_{a_0}$$

$$P_j^0 = \frac{\left(\frac{\lambda}{\mu}\right)^j}{S_{a_0}^{j-S_{a_0}} S_{a_0}!} P_0^0 \text{ Pour } S_{a_0} < j \leq 2S_{a_0}$$

$$P_j^0 = \left(\frac{\lambda}{S_{a_0}\mu}\right)^{j-2S_{a_0}} * \frac{\left(\frac{\lambda}{\mu}\right)^{2S_{a_0}}}{S_{a_0}! S_{a_0}^{S_{a_0}}} P_0^0 - \sum_{m=0}^{j-2S_{a_0}-1} \frac{S_{a_1}}{S_{a_0}} \left(\frac{\lambda}{S_{a_0}\mu}\right)^m P_{2S_{a_0}+1}^1 \quad 2S_{a_0} < j \leq S_{a_0} + \theta_1$$

$$\Rightarrow P_j^0 = \left(\frac{\lambda}{S_{a_0}\mu}\right)^{j-2S_{a_0}} * \frac{\left(\frac{\lambda}{\mu}\right)^{2S_{a_0}}}{S_{a_0}! S_{a_0}^{S_{a_0}}} P_0^0 - \frac{S_{a_1}}{S_{a_0}} * \frac{\left(1 - \frac{\lambda}{S_{a_0}\mu}\right)^{j-2S_{a_0}}}{1 - \frac{\lambda}{S_{a_0}\mu}} P_{2S_{a_0}+1}^1 \quad 2S_{a_0} < j \leq S_{a_0} + \theta_1$$

Avec $\lambda P_{S_{a_0}+\theta_1}^0 = S_{a_1} \mu P_{2S_{a_0}+1}^1$

$$\Rightarrow P_{2S_{a_0}+1}^1 = \frac{\left(\frac{\lambda}{S_{a_0}\mu}\right)^{\theta_1-S_{a_0}} * \frac{\left(\frac{\lambda}{\mu}\right)^{2S_{a_0}}}{S_{a_0}! S_{a_0}^{S_{a_0}}} P_0^0}{\frac{S_{a_1}}{S_{a_0}} * \frac{\left(1 - \frac{\lambda}{S_{a_0}\mu}\right)^{\theta_1-S_{a_0}}}{1 - \frac{\lambda}{S_{a_0}\mu}} + \frac{S_{a_1}\mu}{\lambda}}$$

Pouri $\leftarrow 1 \text{ à } 3 * \frac{k}{4}$

$$P_{2S_{a_{i-1}}+1}^i = \frac{\left(\frac{\lambda}{S_{a_{i-1}}\mu}\right)^{\theta_i-S_{a_{i-1}}}}{\frac{S_{a_i}\mu}{\lambda} + \frac{S_{a_i}}{S_{a_{i-1}}} * \sum_{m=0}^{\theta_i-S_{a_{i-1}}-1} \left(\frac{\lambda}{S_{a_{i-1}}\mu}\right)^m} P_{2S_{a_{i-1}}}^{i-1}$$

$$P_j^i = \sum_{m=0}^{j-2S_{a_{i-1}}} \left(\frac{\lambda}{S_{a_i} \mu} \right)^m P_{2S_{a_{i-1}}+1}^i \quad 2S_{a_{i-1}} < j \leq S_{a_{i-1}} + \theta_i$$

$$P_j^i = \left(\frac{\lambda}{S_{a_i} \mu} \right)^{j-(S_{a_{i-1}}+\theta_i+1)} P_{S_{a_{i-1}}+\theta_i+1}^i + \sum_{m=0}^{j-1-(S_{a_{i-1}}+\theta_i+1)} \left(\frac{\lambda}{S_{a_i} \mu} \right)^m * \frac{1}{S_{a_i} \mu} \left(S_{a_i} \mu P_{S_{a_{i-1}}+\theta_i+1}^i - \lambda P_{S_{a_{i-1}}+\theta_i}^{i-1} - \lambda P_{S_{a_{i-1}}+\theta_i}^i \right)$$

$$S_{a_{i-1}} + \theta_i < j \leq 2S_{a_i}$$

$$P_j^{\frac{k}{2}} = \left(\frac{\lambda}{\frac{k^3}{4} \mu} \right)^{j-2S_{a_{\frac{3^*k}{4}}}+1} P_{2S_{a_{\frac{3^*k}{4}}}+1}^{\frac{k}{2}} \quad j \geq \frac{k^3}{2}$$

$$\text{Avec } \sum_{j=0}^{S_{a_0}+\theta_1} P_j^0 + \sum_{i=1}^{\frac{k}{2}} \sum_{j=2S_{a_{i-1}}+1}^{S_{a_{i-1}}+\theta_i} P_j^i + \sum_{j=2S_{a_{\frac{3^*k}{4}}-1}}^{\infty} P_j^{\frac{3^*k}{4}} = 1$$

5.3.3 Calcul des métriques de performance

$$\text{Le taux d'utilisation } \rho = \frac{\lambda}{\frac{k^3}{4} \mu}$$

Le nombre de clients en attente est:

$$\Omega = \sum_{i=0}^{\frac{3^*k}{4}-1} \left(\sum_{j=S_{a_i}}^{S_{a_i}+\theta_{i+1}} (j - S_{a_i}) P_j^i \right) + \sum_{j=S_{a_{\frac{3^*k}{4}}}}^{\infty} \left(j - S_{a_{\frac{3^*k}{4}}} \right) P_j^{\frac{3^*k}{4}} \quad (10)$$

D'après la formule de Little, le temps d'attente moyen $T_a = \frac{\Omega}{\lambda}$

L'énergie consommée au niveau des serveurs et au niveau des Switchs edge agrégation et core est représentée dans la formule suivante :

$$E = \left[\begin{aligned} & \sum_{j=0}^{S_{a_0}} P_j^0 \left(j e_{busy} + (S_{a_0} - j) e_{idle} + (S_{a_{\frac{3^*k}{4}}} - S_{a_0}) e_{off} \right) + \sum_{j=S_{a_0}}^{2S_{a_0}} P_j^0 \left(S_{a_0} e_{busy} + (S_{a_{\frac{3^*k}{4}}} - S_{a_0}) e_{off} \right) \\ & + \sum_{i=1}^{\frac{3k}{4}-1} \sum_{j=2S_{a_{i-1}}}^{S_{a_{i-1}}+\theta_i} P_j^i \left(S_{a_i} e_{busy} + (S_{a_{\frac{3^*k}{4}}} - S_{a_i}) e_{off} \right) + \sum_{i=1}^{\frac{3^*k}{4}-1} \sum_{j=S_{a_{i-1}}+\theta_i+1}^{2S_{a_i}} P_j^i \left(S_{a_i} e_{busy} + (S_{a_{\frac{3^*k}{4}}} - S_{a_i}) e_{off} \right) + P_{j > \frac{k^3}{2}}^{i} S_{a_{\frac{3^*k}{4}}} e_{busy} \\ & + nbP^* \left(\frac{k}{2} e_{busyS_e} + \frac{k}{2} e_{busyS_a} \right) + \frac{k^2}{4} e_{busyS_c} \end{aligned} \right] \quad (13)$$

Cette quantité d'énergie correspond à l'énergie consommée depuis les Switch core jusqu'aux serveurs.

5.4 Résultats numériques

Nous avons commencé par le modèle analytique pour montrer que notre proposition est meilleure que celle de Dan et al. en termes d'économie d'énergie. Nous avons utilisé $k=8$ i.e. 128 serveurs avec un nombre de requêtes dans le système variant entre 32 et 400 avec une charge du système $\rho=0,7$.

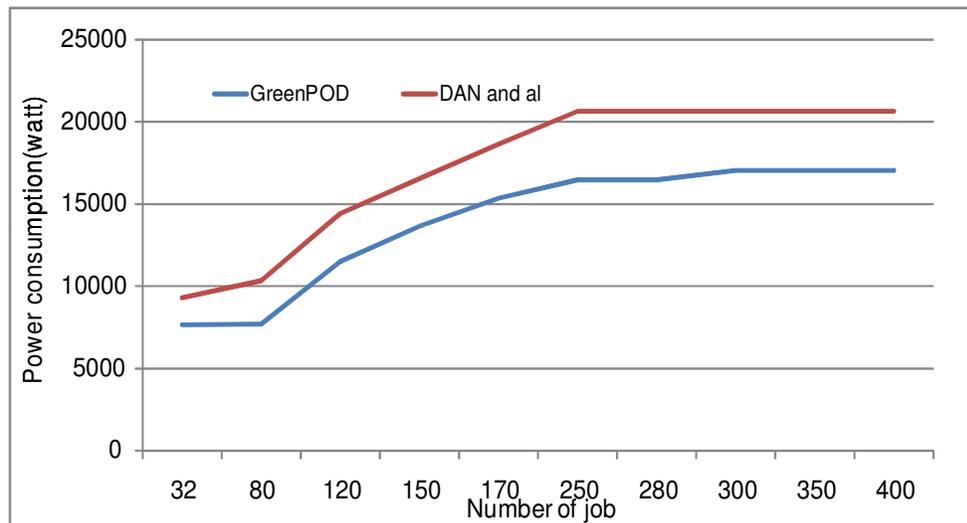


Figure 5.6: La puissance consommée en fonction du nombre de clients dans le système

La figure 5.6 montre que l'énergie augmente en fonction du nombre de clients dans le système puis devient stable si le nombre total de serveurs est atteint.

Dans la simulation, nous avons considéré un système composé de 432 serveurs ($k=12$ pour une topologie Fat-Tree) ayant chacun un temps de service de 0.2 seconde, la puissance consommée par un serveur occupé est $e_{busy}=240w$, celle d'un serveur à l'état idle est $e_{idle}=150w$ et celle d'un serveur éteint est $e_{off}=0w$. Pour notre proposition et celle de Dan et al., nous avons subdivisé les serveurs en groupes. Au début de la simulation, nous activons un groupe de serveurs donné. En fonction de la taille de la file d'attente, nous activons ou désactivons un nouveau groupe de serveurs. Pour la proposition de Dan et al., les serveurs sont subdivisés en trois groupes de n , m_1 et m_2 serveurs. n , m_1 et m_2 étant des entiers. Dès que les seuils d'activation Θ_1 et Θ_2 sont atteints, on active le groupe suivant. Nous avons choisi ces seuils d'activation et de désactivation en rapport avec leur proposition ($n \leq \Theta_1 \leq n+m_1$ et $m_1 \leq \Theta_2 \leq m_2+m_1$). On désactive un groupe si le nombre de clients dans le système passe en dessous du nombre de serveurs activés. Pour notre proposition, nous avons choisi d'activer un nouveau groupe de serveurs dès que la taille de la queue dépasse le nombre de serveurs après activation. Nous désactivons un groupe dès que la taille de la queue est inférieure au nombre de serveurs activés. Lors de la désactivation, notre algorithme récupère l'ensemble des requêtes en cours d'exécution sur les serveurs à désactiver et les replace en tête de file avec leurs temps d'arrivée dans le système pour éviter de les perdre ou les obliger à reprendre leurs demandes. Les résultats obtenus montrent que dans toutes les configurations (système peu sollicité, moyennement sollicité ou trop sollicité), nous économisons plus d'énergie tout en gardant un temps de réponse acceptable par rapport aux autres propositions.

Pour un système relativement peu sollicité (*Probabilité des inter-arrivées* =0,45), nous avons noté une consommation moyenne d'énergie pour notre proposition de 24 167.61 watt. Pour la proposition de Dan et al., cette consommation moyenne

est estimée à 29 567.61 watt, tandis que pour un centre de données par défaut où on démarre l'ensemble des serveurs, la consommation moyenne est estimée à 72 767.61 watt. Ainsi, par rapport au centre de données par défaut, nous économisons 66,79% d'énergie, tandis que Dan et al. économise 59,37%. La figure 5.7, montre la consommation d'énergie en fonction du temps de simulation. Les systèmes étant peu sollicités, aucun nouveau groupe de serveurs n'est activé.

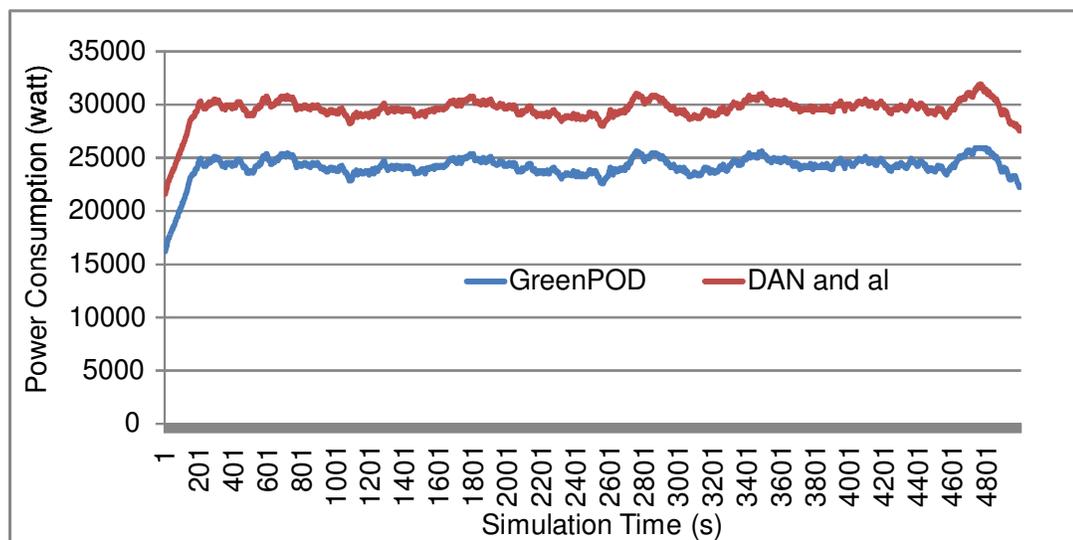


Figure 5.7: Consommation d'énergie en fonction du temps de simulation pour un système peu sollicité

Pour un système un peu plus sollicité (*Probabilité des inter-arrivées* =0.65), notre consommation moyenne d'énergie est estimée à 29 242.66 watt. Pour la proposition de Dan et al., cette consommation moyenne est estimée à 33 189.05 watt, tandis que pour le centre de données par défaut, elle est égale à 76 389.05 watt. Donc par rapport à un centre de données par défaut, nous économisons 61,72% d'énergie, tandis que Dan et al. économise 56,55% d'énergie.

La figure 5.8 présente les consommations d'énergie en fonction du temps de simulation, et nous remarquons des activations et désactivations pour notre

proposition. Dan et al. ayant démarré plus de serveurs, le système n'a pas eu besoin de démarrer un nouveau groupe de serveurs.

La figure 5.9 présente la consommation d'énergie ainsi que le temps d'attente des clients en fonction du temps de simulation. Nous pouvons remarquer qu'à chaque fois que nous activons un groupe de serveurs, le temps d'attente diminue légèrement. Lors des différentes désactivations, nous remarquons une augmentation brusque du temps d'attente. Cela s'explique par le fait que les clients qui étaient en train d'être servis au niveau des serveurs à désactiver sont replacés en tête de file avec leurs temps d'arrivée initiaux dans le système qui sont forcément inférieurs aux temps d'arrivée des autres clients en attente. Leurs réels temps d'attente sont donc égaux au temps courant moins leurs temps d'arrivées initiaux dans le système. Ayant repris leurs services, ces requêtes ont plus attendu que les autres, mais cela est plus avantageux que de les abandonner ou de les replacer en queue de file. Une fois que les requêtes ayant repris leurs services terminent leur traitement, le temps d'attente redevient normal. Cette hypothèse n'a pas été évoquée dans les travaux de Dan et al.

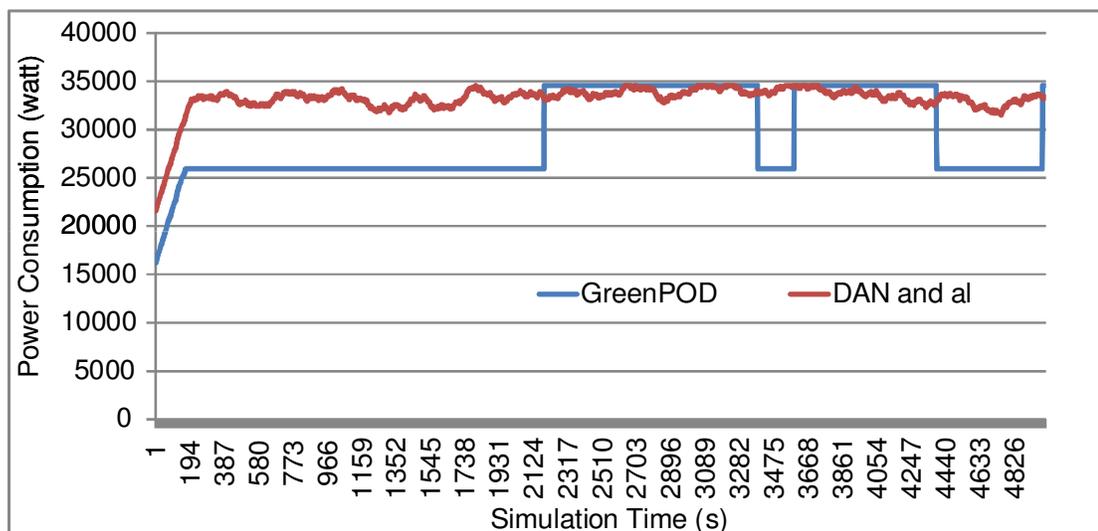


Figure 5.8: Consommation d'énergie en fonction du temps de simulation pour un système moyennement sollicité

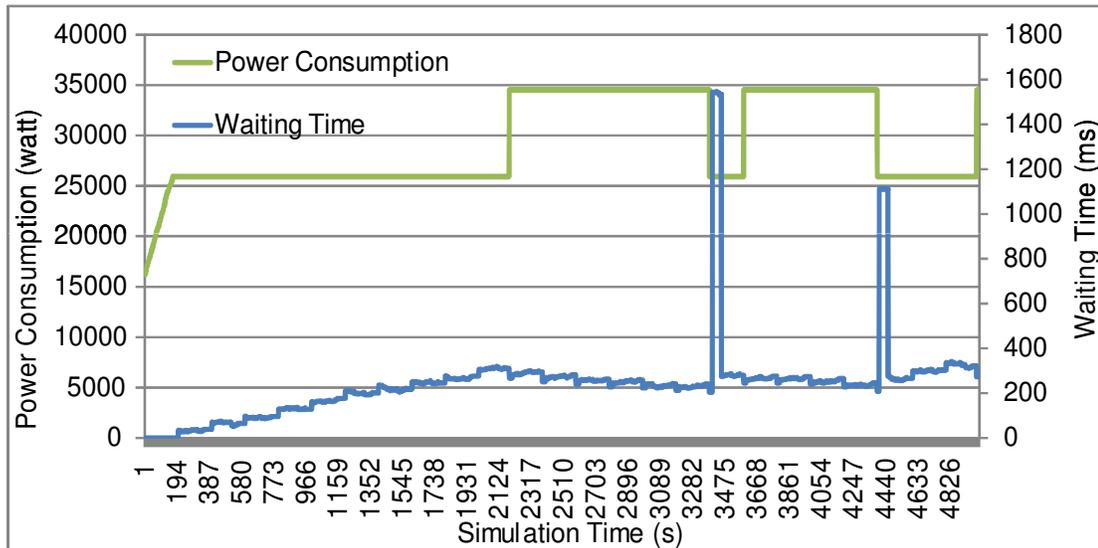


Figure 5.9: Consommation d'énergie et temps d'attente de notre proposition en fonction du temps de simulation pour un système moyennement sollicité

Pour un système très sollicité (*Probabilité des inter-arrivées* =0.95), la consommation moyenne d'énergie pour notre proposition est de 42 652.39 watt. Pour la proposition de Dan et al., cette consommation moyenne est estimée à 45 447.26 watt, tandis que pour un centre de données par défaut, elle est estimée à 81 637.16 watt. Par rapport au centre de données par défaut, nous économisons 47,75% d'énergie, tandis que Dan et al. économise 44,33%.

La figure 5.10 présente les consommations d'énergie en fonction du temps de simulation. Les systèmes étant très sollicités, dans toutes les deux propositions, il y a eu des activations et désactivations de groupes de serveurs.

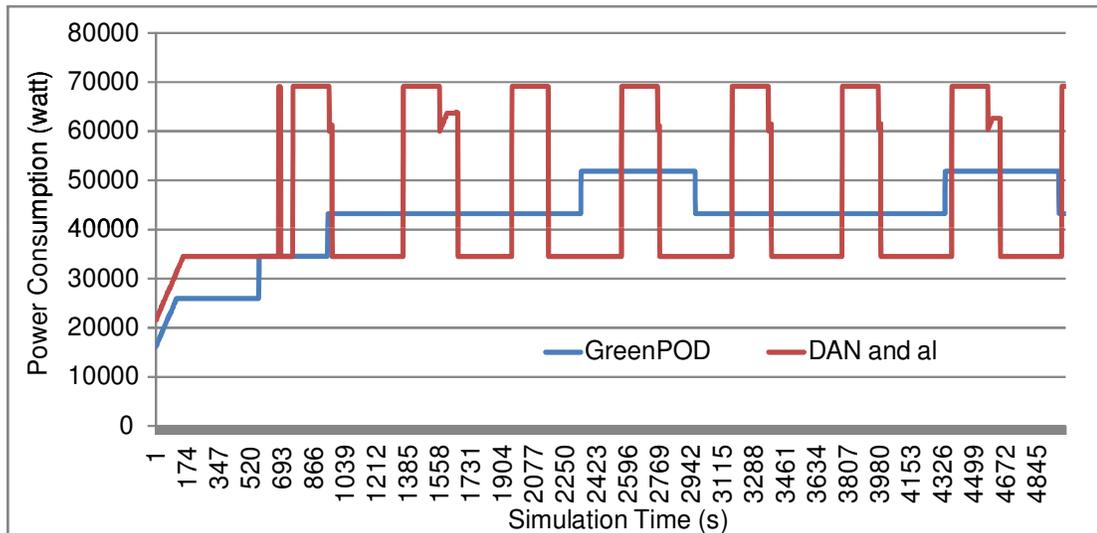


Figure 5.10: Consommation d'énergie en fonction du temps de simulation pour un système très sollicité

La figure 5.11 présente les consommations d'énergie et temps d'attente de notre proposition pour un système très sollicité. Nous faisons la même remarque sur la variation du temps d'attente. À chaque désactivation d'un groupe de serveurs, les requêtes qui y étaient ne sont pas perdues et sont replacées en tête de queue. Cet évènement se produit sur la figure 5.11 vers la 2977^{ième} seconde. Le temps qui augmente brusquement est celui des requêtes ayant repris leurs services et replacées en tête de queue.

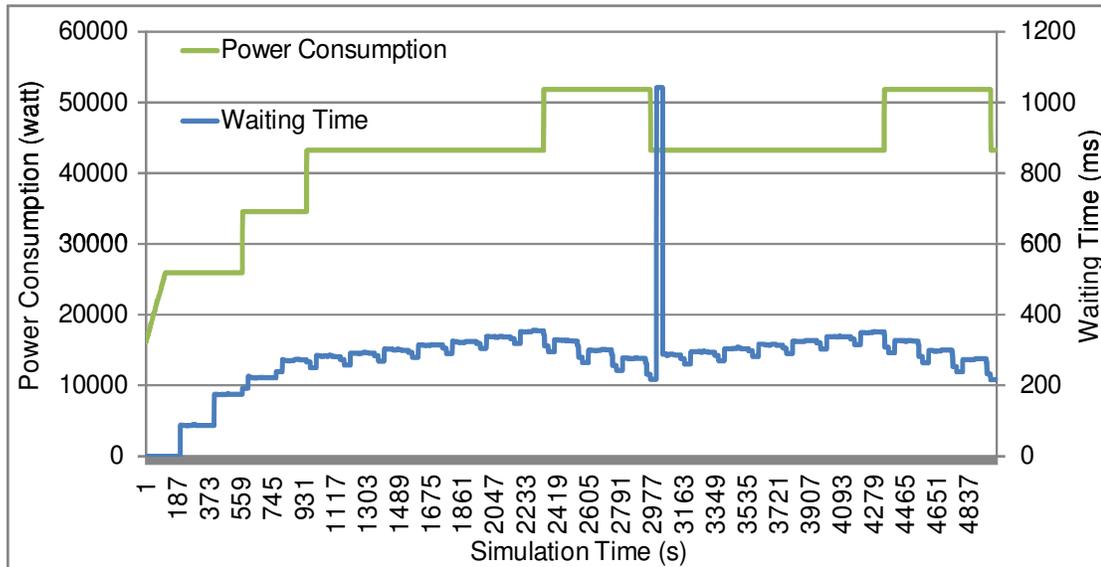


Figure 5.11: Consommation d'énergie et temps d'attente de notre proposition en fonction du temps de simulation pour un système très sollicité

La figure 5.12 représente le temps de réponse des deux propositions. Nous pouvons remarquer que le temps d'attente est légèrement mieux sur la proposition de Dan et al.

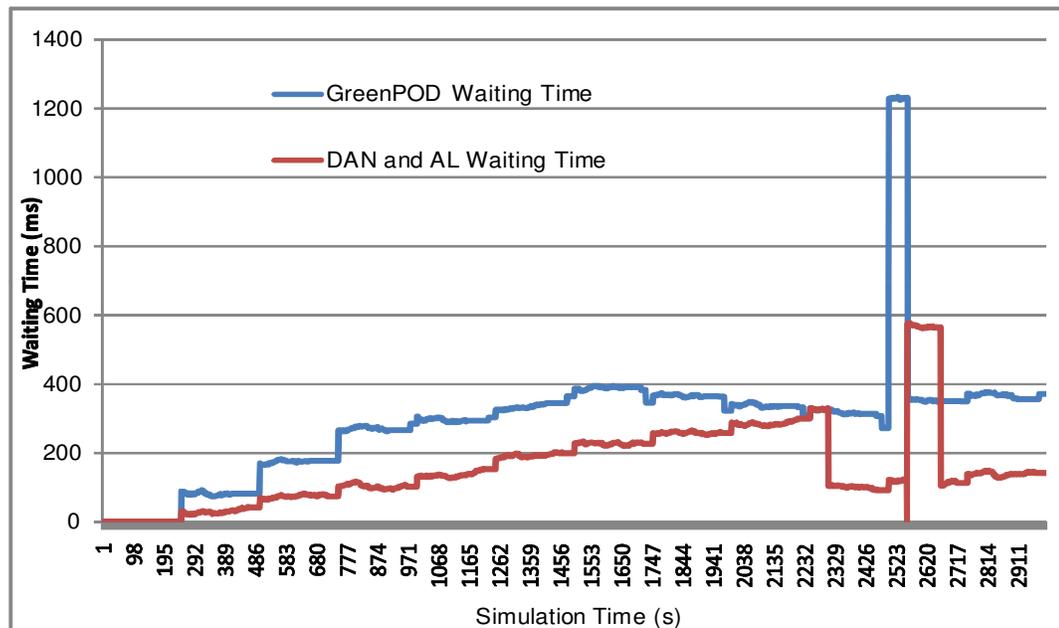


Figure 5.12: Temps d'attente des deux propositions en fonction du temps de simulation

La figure 5.13 représente la consommation d'énergie en fonction du nombre de serveurs dans chaque modèle. Nous remarquons que plus le nombre de serveurs augmente plus le gain en énergie devient significatif pour notre proposition. Dans la simulation, nous avons augmenté le temps de simulation en fonction de la valeur de k_aray . Cela nous permet d'avoir des résultats plus précis.

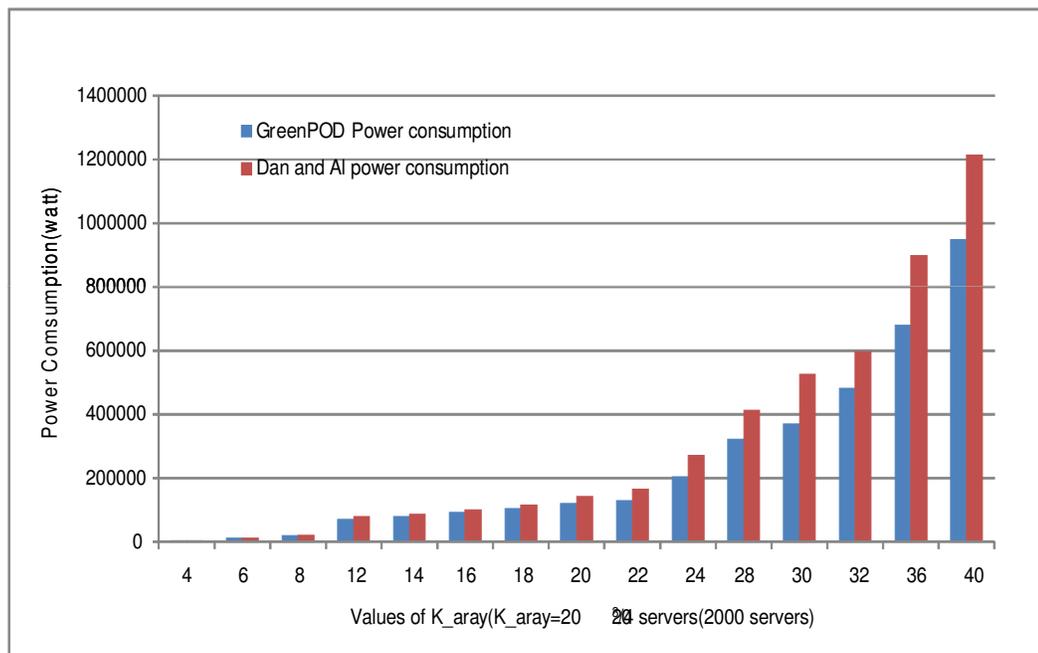


Figure 5.13: Consommation d'énergie des deux propositions en fonction du nombre de serveurs

La figure 5.14 nous permet de savoir le gain en énergie par rapport à un centre de données par défaut. L'axe des abscisses représente la valeur de K dans une topologie Fat-Tree. Il donne ainsi le nombre de serveurs dans chaque situation $(\frac{K^3}{4})$.

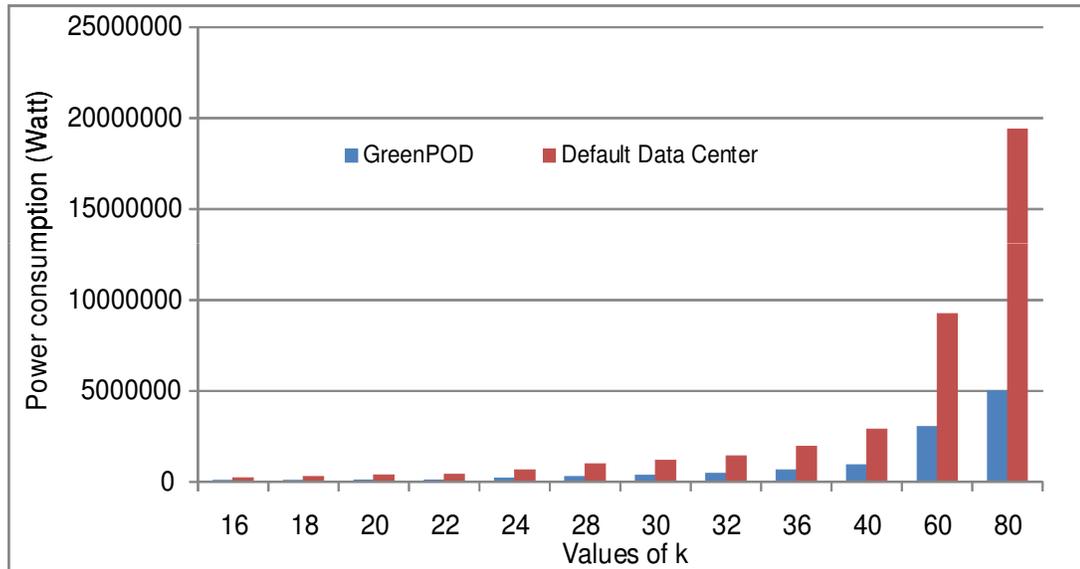


Figure 5.14: Consommation d'énergie de notre proposition par rapport à un centre de données par classe

Les résultats de la figure 5.14 montrent que pour un k_array dont les valeurs sont comprises entre 16 et 80, nous arrivons à obtenir un gain en énergie variant entre 59% et 74%.

Les résultats des figures 5.15, 5.16 et 5.17 montrent la consommation d'énergie au niveau des serveurs. Cependant nous avons calculé la consommation d'énergie au niveau des Switch. Dans notre simulation nous avons fait varier le k_array entre 4 et 36 par pas de "2" avec un temps de simulation inférieur au résultat ci-dessus et nous avons évalué la consommation d'énergie au niveau des Switch et des serveurs dans le pire des cas c'est-à-dire que nous supposons que les Switch allumés sont en service et ils consomment une puissance $e_{busyS_e} = e_{busyS_a} = 300w$ et ceux qui sont éteints consomment une puissance $e_{off} = 0w$.

La figure 5.15 montre la consommation d'énergie au niveau des Switch. La figure 5.16 représente la consommation d'énergie des Switch et des serveurs des deux propositions. L'axe des abscisses représente le nombre de serveurs dans les deux cas.

Nous pouvons remarquer que plus le nombre de serveurs est important, plus le gain en énergie est important par rapport à la proposition de Dan et al.

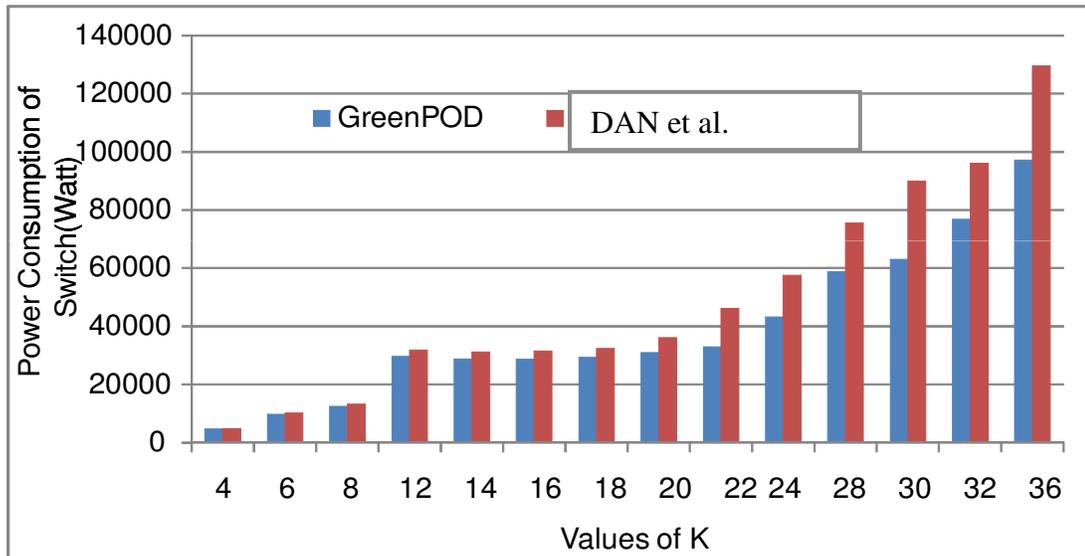


Figure 5.15: Consommation d'énergie des Switch par rapport au deux propositions

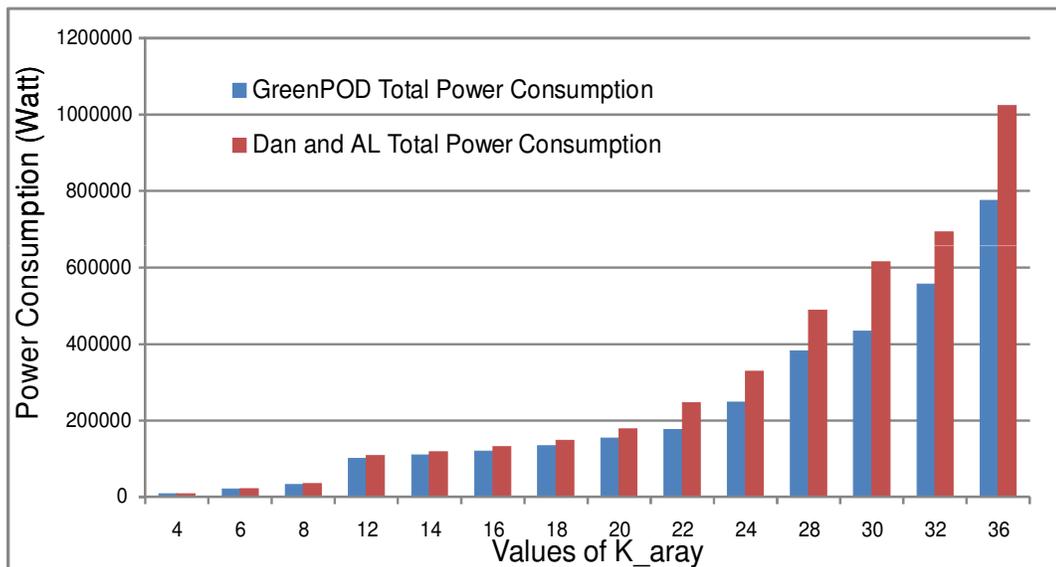


Figure 5.16: Consommation totale d'énergie (Switch+serveurs) par rapport au deux propositions

Enfin la figure 5.17 montre la consommation totale d'énergie de notre proposition par rapport à un centre de données classique. Nous remarquons la quantité d'énergie que l'on peut économiser en adoptant une gestion dynamique des serveurs.

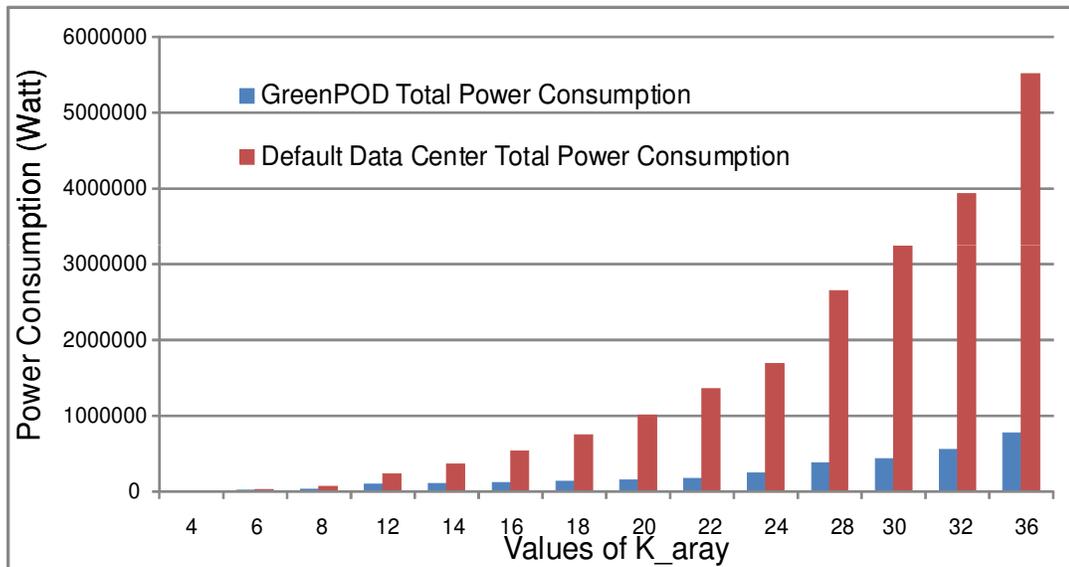


Figure 5.17: Consommation totale d'énergie (Switch+serveurs) de notre proposition par rapport à un centre de données par classique

Les trois courbes suivantes présentent l'énergie consommée en fonction de la charge du système. La figure 5.18 présente la consommation d'énergie des serveurs en fonction de la charge du système. Par exemple, pour une charge de 40% avec un temps de simulation de 2000 millisecondes, GreenPOD consomme 47 2984.5 watt contre 61 8784.5 watt pour Dan et al.

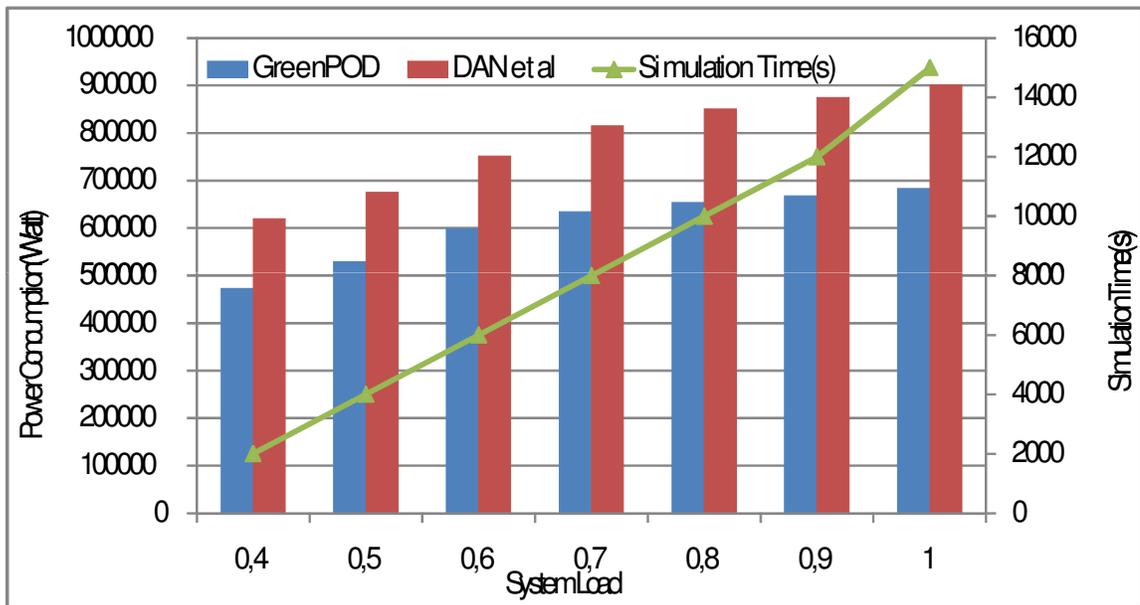


Figure 5.18: Consommation d'énergie des serveurs en fonction de la charge du système

La figure 5.19 présente l'énergie consommée pour l'ensemble (serveur + Switch) en fonction de la charge du système.

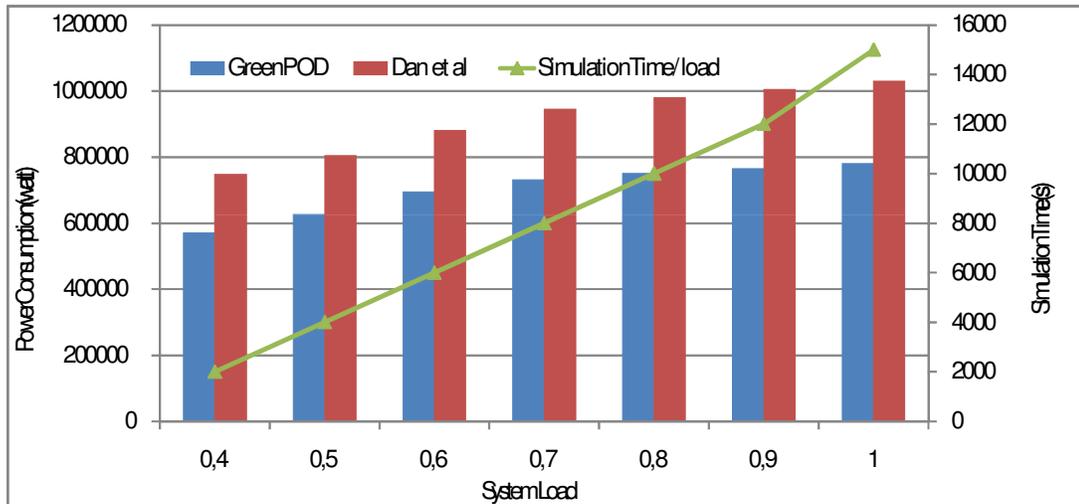


Figure 5.19: Energie consommée (Serveurs + Switch) en fonction de la charge

5.5 Conclusion

Dans ce chapitre, nous avons proposé un modèle de fonctionnement avec différents seuils d'activation pour réduire la consommation d'énergie dans un centre de données avec la topologie Fat-tree. Pour cela nous avons ajusté le nombre de serveurs et Switch actifs dans le système tout en éteignant les autres. La décision de savoir si les serveurs doivent être activés est basée sur le nombre de requêtes dans la file. Nous avons comparé notre modèle avec celui de Dan et al. et les résultats montrent que nous économisons beaucoup plus d'énergie par rapport à eux avec un temps de réponse acceptable.

Ainsi, nous pouvons garantir les accords de niveau de service à l'utilisateur final tout en économisant environ 74% de l'énergie avec l'adaptation des switch et serveurs.

Chapitre 6 : CONCLUSION GÉNÉRALE

6.1 Bilan

Dans cette thèse, nous nous sommes intéressés à la réduction de la consommation d'énergie dans les centres de données en utilisant le formalisme des réseaux de file d'attente. Après une revue de la littérature qui nous a permis de bien comprendre les enjeux et l'état des lieux sur les économies d'énergie dans les centres de données, nous avons fait quelques propositions.

Nous avons, dans les premiers résultats, proposé une architecture de Cloud écologique appelée Green Cloud Architecture. Nous avons, dans cette architecture, travaillé avec des routeurs virtuels instanciés sur des routeurs physiques. Cette architecture présente l'avantage de pouvoir éteindre certains routeurs physiques pour des raisons d'économie d'énergie. Les routeurs virtuels précédemment instanciés sur ces derniers routeurs physiques peuvent être migrés vers d'autres routeurs. Le résultat obtenu montre que notre algorithme d'allocation de ressources peut économiser jusqu'à 41% d'énergie.

Les centres de données sont généralement localisés sur des sites distants par rapport aux utilisateurs. Certains fournisseurs ont tendance à les construire dans des zones polaires où le refroidissement peut coûter moins cher. Cette démarche entraîne une augmentation de la quantité de données transférée entre fournisseurs et clients. Les équipements requis pour acheminer le trafic aussi augmentent, entraînant ainsi la quantité d'énergie produite dans le Cloud. Nous nous sommes intéressés dans la deuxième contribution à l'économie d'énergie lors du transport des données dans le Cloud. L'algorithme de réduction de la consommation d'énergie des équipements réseaux lors du transport des données a permis d'économiser 41% d'énergie par rapport à un réseau non efficient.

Dans la troisième contribution, nous sommes focalisés sur la réduction de la consommation d'énergie au cœur des centres des données. Nous avons proposé un modèle d'organisation des centres de données et un système de traitement des requêtes en utilisant le formalisme des réseaux de file d'attente. Le modèle GREENPOD que nous avons proposé est basé sur la topologie Fat-Tree. Cette topologie organise les équipements réseaux et les serveurs sous forme de Pod. Nous avons choisi de démarrer le quart des Pod et en fonction de la charge du système, les autres Pod seront démarrés. Le résultat obtenu montre que nous économisons bien de l'énergie par rapport à [Dan et al. 2015] et surtout par rapport à un centre de données par défaut.

6.2 Perspectives

En se basant sur les travaux et résultats obtenus dans le cadre de cette thèse, nous envisageons définir des contraintes plus spécifiques concernant l'hétérogénéité des serveurs. Nous prévoyons aussi des simulations basées sur des SLA différents ou bien des taux d'arrivés différents. Nous pourrons aussi tester le modèle avec une file M/G/ 1 avec des classes de clients différentes. Une autre approche envisagée est de

combiner les Data Centers avec les nano Data Centers pour mieux économiser l'énergie.

6.3 Publications relatives à la thèse

1. Fatoumata B. Kassé, Halima Elbiaze, Bamba Gueye, "GreenPOD: Leveraging Queuing Networks for Reducing Energy Consumption in Data Centers", IEEE Conference on Innovation in Clouds, Internet and Networks (ICIN), ISSN: 2472-8144, February 2018, Paris, France
2. Fatoumata B. Kassé, Bamba Gueye, Halima Elbiaze, "Green Cloud Architecture: Low Power Routers for an Energy-Aware Data Transport", IEEE Globecom Workshop, ISSN: 2166-0077, Austin 2014, Texas, USA.
3. Fatoumata B. Kassé, Bamba Gueye, Halima Elbiaze, "Leveraging Network Virtualization for Energy-Efficient Cloud: Future Directions", IEEE Conference on Local Computer Networks (LCN), ISSN: 0742-1303, September 2014, Edmonton, Canada.

BIBLIOGRAPHIE

[Al-Fares et al. 2008] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proceedings of the ACM SIGCOMM 2008 conference on Data communication. ACM, 2008, pp. 63–74.

[Al-Fares et al. 2010] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” in Proceedings of the 7th USENIX conference on Networked systems design and implementation. USENIX Association, 2010, p. 19.

[Andrea 2017] Dr. Anders S.G. Andrae (Huawei), “Total Consumer Power Consumption Forecast” at the Nordic Digital Business Summit, Helsinki, Finland, October 5, 2017.

[Andreyev 2014], A. Andreyev, “Introducing data center fabric, the next-generation facebook data center network”, Novembre 2014, [Online], <https://code.facebook.com/posts/360346274145943/>

[Armbrust et al. 2009] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Kon-winski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” Electr. Eng. Comput. Sci. Dept., University California, Berkeley, CA, UCB/EECS-2009-28, Tech. Rep., February 2009. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>

[Baliga et al. 2011] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, “Green Cloud Computing: Balancing Energy in Processing, Storage and Transport,” Proceedings of the IEEE, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[Barroso & U. Hölzle, 2007] L. A. Barroso et UrsHölzle. “The Case for Energy-Proportional Computing”. In :*IEEE Computer* 40 (2007). url : http://www.computer.org/portal/site/computer/index.jsp?ageID=computer_level1&path=computer/homepage/Dec07&file=feature.xml&xsl=article.xsl.

[Belbekkouche et al. 2012] A. Belbekkouche, M. Hasan, and A. Karmouch, “Resource discovery and allocation in network virtualization,” *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1–15, June 2012

[Beloglazov&Buyya, 2010] A. Beloglazov and R. Buyya, "Energy efficient management in virtualized cloud data centers", in Cluster, cloud and Grid Computing (CCGrid), 10th IEEE/ACM International Conference, (Melbourne, Australia), p. 826-831, 2010.

[Beloglazov et al. 2012] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems*, vol. 28, no 5, p. 755-768, 2012.

[Beloglav 2013] A. Beloglav, Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing. PhD thesis, Departement of Computing and Information Systems, The university of MELBOURNE, April 2013.

[Berl et al. 2010] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, “Energy-Efficient Cloud Computing,” *The Computer Journal*, vol. 53, no. 7, pp. 1045–1051, Sept. 2010.

[Bianzino et al. 2012] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, “A survey of green networking research,” *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.

[Bolla et al. 2011] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures,” *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 223–244, 2011.

[Brackbill. 2016]N. Brackbill, "The Energy of the Cloud," Physics 240, Stanford University Fall 2016.

[Buyya et al. 2009] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, June 2009.

[Buyya& Sing 2018] R. Buyya and S. Singh. “Sustainable Cloud Computing: Foundations and Future Directions.” *Business Technology & Digital Transformation Strategies*, Cutter Consortium, Vol. 21, no. 6, Pages 1-9,2018.

[Chase et al. 2001] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, “Managing energy and server resources in hosting centers,” in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Chateau Lake Louise, Banff, Canada, October 2001.

[Chabarek et al 2008] J. Chabarek, J. Sommers, P. Barford, C. Egan, D. Tsang, and S. Wright, “Power awareness in network design and routing,” in *Proc. INFOCOM*, April 2008, pp. 457–465.

[Chen et al. 2005] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, “Managing server energy and operational costs in hosting centers,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 303–314, 2005.

[Chen et al. 2012] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, “Osa: An optical switching architecture for data center networks with unprecedented flexibility,” in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 18–18.

[Chirgwin 2015] Richard Chirgwin :Which data centre network topology's best? Depends on what you want to break : https://www.theregister.co.uk/2015/10/13/which_data_centre_topology_is_best_depends_on_what_you_want_to_break/

[Da COSTA 2015] G. Da COSTA, "Evaluation et optimisation de performance énergétique des centres de calcul", Thèse en vue de l'obtention de l'habilitation de l'université de Toulouse, présentée et soutenue le 12 Novembre 2015.

[Dally & Towles 2004] W. Dally and B. Towles, Principles and practices of interconnection networks. Morgan Kaufmann, 2004.

[Dayarathna et al. 2016] M. Dayarathna, Y. Wen and F. R., “Data Center Energy Consumption Modeling: A Survey,” IEEE Communications Surveys and Tutorials, vol. 18, no. 1, pp. 732-794, 2016.

[Djoughra, Thèse de doctorat 2016] D. Djoughra, Optimisation des performances des data centers des cloud sous contrainte d'énergie consommée, thèse pour le diplôme de doctorat en sciences, présentée et soutenue le 21/04/2016.

[Duato et al. 2003] J. Duato, S. Yalamanchili, and L. Ni, Interconnection networks: An engineering approach. Morgan Kaufmann, 2003.

[Edmonds 1965] J. Edmonds, “Paths, trees, and flowers,” Canadian Journal of mathematics, vol. 17, no. 3, pp. 449–467, 1965.

[EPA 2007] Report to congress on server and data center energy efficiency, environmental protection agency,2007.www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf.

[ETSI 2013] ETSI, “Leading operators create ETSI standards group for network functions virtualization,” 22 January 2013 - <http://www.etsi.org/news-events/news/644-2013-01-isg-nfv-created>.

[Farrington et al. 2010] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 339–350.

[Fischer et al. 2013] A. Fischer, J. F. Botero, M. T. Beck, H. D. Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[Foster et al. 2008] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” in *IEEE Grid Computing Environments Workshop*, November 2008, pp. 1–10.

[Gandhi et al. 2010] A. Gandhi, V. Gupta, M. Harchol-Balter, M. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. In: *Proceedings of the 28th Performance*, 2010.

[Ghamkhari&Mohsenian-Rad 2013] M. Ghamkhari, H. Mohsenian-Rad. Energy and Performance Management of Green Data Centers: A Profit Maximization Approach. *IEEE Transactions on Smart Grid*, Vol. 4, No. 2, JUNE 2013, pp.1017-1025

[Ghribi 2014] C. Ghribi, "Energy efficient resource allocation in cloud computing environments. Net-working and Internet Architecture. Institut National des Télécommunications, 2014.

[Gomma et al. 2011] E. Goma, M. Canini, A. L. Toledo, N. Laoutaris, D. Kostić, P. Rodriguez, R. Stanojević, and P. Y. Valentin, "Insomnia in the access: or how to curb access network related energy consumption," in *Proceedings of the ACM SIGCOMM 2011 conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, Aug. 2011, pp. 338–349.

[GORELIK, Master 2013] E. GORELIK, "Cloud Computing Models, Comparison of Cloud Computing Service and Deployment Models", Master of Management and Master of Engineering, Massachusetts Institute of Technology, January 2013.

[Greenberg et al. 2009] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *IGCOMMComput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.

[Grevet 2009] N. Grevet, "Le Cloud computing: évolution ou révolution? pourquoi, quand, comment et surtout faut-il prendre les risques?", *Mémoire de recherche*, Aout 2009.

[Grid 2010] International Conference on Grid Computing (Grid 2010) Energy Efficient Grids, Clouds and Clusters Workshop (E2GC2-2010), pp.313-320, Brussels, Belgium, October 2010

[Guenter et al. 2011] B. Guenter, N. Jain, C. Williams, "Managing cost, performance and reliability tradeoffs for energy-aware server provisioning", *IEEE Infocom 2011*, pp.1332-1340.

[Gueye et al. 2014] Fatoumata B. Kassé, Bamba Gueye, Halima Elbiaze, "Green Cloud Architecture: Low Power Routers for an Energy-Aware Data Transport", IEEE Globecom Workshop, ISSN: 2166-0077, Austin 2014, Texas, USA.

[Guo et al. 2008] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 75–86, 2008

[Guo et al. 2009] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: A high performance, server-centric network architecture for modular data centers," ACM SIGCOMM Computer Communication Review, vol. 39, no. 4, pp. 63–74, 2009.

[Gupta & Singh 2003] M. Gupta and S. Singh, "Greening of the internet," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. Karlsruhe,Germany: ACM, 2003, pp. 19–26.

[Hamilton et al. 2009] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "V12: a scalable and flexible data center network," SIGCOMM Comput. Commun. Rev., vol. 39, no. 4, pp. 51–62, 2009.

[HAMZE, Thèse doctorat 2015] M. HAMZE, "Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing », Thèse de doctorat, Soutenue publiquement le 07 décembre 2015, Université de Bourgogne.

[Heller et al. 2010] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastic tree: Saving energy in data center networks," in *7th USENIX Symposium on Networked System Design and Implementation (NSDI)*, San Jose, CA, USA, April 2010, pp. 249–264.

[HESS 2013] E. HESS, "Comment le « cloud » bouleverse l'économie de l'informatique", <http://business.lesechos.fr/directions-numeriques/technologie/cloud-computing/comment-le-cloud-bouleverse-l-economie-de-l-informatique-5976.php?SWZABh952l6RTe9P.99>, 09/04/2013.

[Jinget al. 2017] Jing Ni, Bowen Jin, Bo Zhang and Xiaowei Wang, "Simulation of Thermal Distribution and Airflow for Efficient Energy Consumption in a Small Data Centers", *Sustainability* 2017, 9, 664; doi:10.3390/su9040664.

[Kant 2009] K. Kant, "Data center evolution: A tutorial on state of the art, issues, and challenges," *Computer Networks*, vol. 53, no. 17, pp. 2939–2965, 2009.

[Kasse et al. 2014] F. B. Kasse, B. Gueye, and H. Elbiaze, "Leveraging network virtualization for energy-efficient cloud: Future directions," in *Proc. IEEE LCN*, Edmonton, Canada, September 2014, pp. 494–497.

[Kasse et al. 2018] Fatoumata B. Kassé, Halima Elbiaze, Bamba Gueye, "GreenPOD: Leveraging Queuing Networks for Reducing Energy Consumption in Data Centers", *IEEE Conference on Innovation in Clouds, Internet and Networks (ICIN)*, ISSN: 2472-8144, February 2018, Paris, France

[Kliazovich et al. 2010] D. Kliazovich, P. Bounvry, Y. Audzevich, and S. U. Khan, "Greencloud: A packet-level simulator of energy-aware cloud computing data centers," in *IEEE Globecom*, Miami, FL, USA, December 2010.

[Koomey 2011] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," Oakland, CA, Analytics Press, 2011.

[Le LOUËT 2014] G. Le LOUËT, "Maîtrise énergétique des centres de données virtualisés : D'un scénario de charge à l'optimisation du placement des calculs", Thèse de doctorat, Soutenue le 12 mai 2014, Ecole des Mines de Nantes.

[Leiserson 1985] C. E. Leiserson, “Fat-trees: universal networks for hardware-efficient supercomputing,” *IEEE Trans. Comput.*, vol. 34, no. 10, pp. 892–901, 1985.

[Dan et al. 2015] D. Dan, Ke Li, Gang Sun, Vishal Anand, Yu Gong, Zhi Tan « Energy and Performance Management in Large Data Centers A Queuing Theory Perspective » *Computing, Networking and Communications (ICNC)*, 2015 International Conference on (2012) IEEE.

[Liao & Wang 2018] Q. Dan, Z. Wang, “Energy Consumption Optimization Scheme of Cloud Data Center Based on SDN”, 8th International Congress of Information and Communication Technology (ICICT- 2018), *Procedia Computer Science* 131 (2018) 1318–1327.

[Mahadevan et al. 2009] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, “Energy aware network operations,” in *Proc. of IEEE GIS*, ser. INFOCOM, Rio de Janeiro, Brazil, 2009, pp. 25–30.

[Mardani et al. 2015] Abbas Mardani, Ahmad Jusoh, Edmundas Kazimieras Zavadskas, Fausto Cavallaro, and Zainab Khalifah. “Sustainable and renewable energy: An Overview of the Application of Multiple Criteria Decision Making Techniques and Approaches”, *Sustainability*, vol. 7, no. 10, pp. 13947–13984, 2015.

[Marinos 2009] G. B. A. Marinos, “Community cloud computing,” in *First International Conference on Cloud Computing*, Beijing, China, December 2009.

[Mazucco et al. 2010] M. Mazucco, D. Dyachuk, and R. Deters, « Maximizing cloud providers revenues via energy aware allocation policies », *IEEE International Conference on Cloud Computing*, Miami, Florida, July 2010, pp. 131–138.

[Mell &Grance 2009] P. Mell and T. Grance, “The nist definition of cloud computing,” National Institute of Standards and Technology, vol. 53, no. 6, p. 50, 2009.

[Mell &Grance 2011] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", Special Publication 800-145, September 2011.

[Mohit 2010] M. Mohit, “A comprehensive solution to cloud traffic tribulations,” in *International Journal on Web Service Computing*, vol. 1, no. 2, December 2010, pp. 1–13.

[Müller–Hannemann& Schwartz 2000] M. Müller -Hannemann and A. Schwartz, “Implementing weighted bmatching algorithms: insights from a computational study,” *Journal of Experimental Algorithmics (JEA)*, vol. 5, p. 8, 2000.

[Naone 2009] E. Naone. Conjuring clouds. *Technology Review*, 112(4):54-56, 2009.

[Nedeveschi et al. 2008] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, “Reducing network energy consumption via sleeping and rate adaptation,” in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI’08. Berkeley, CA, USA: USENIX Association, 2008, pp. 323–336. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1387589.1387612>.

[Ni & Bai 2017] J. Ni and X. Bai, “A review of air conditioning energy performance in data centers,” *Renewable and Sustainable Energy Reviews*, vol. 67, pp.625-640, 2017.

[Niranjan et al. 2009] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, “Portland: a scalable fault-tolerant layer 2 data center network fabric,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 39–50, 2009.

[Nogueira et al. 2011] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," *IEEE Symposium on Computers and Communication*, pp. 438–444, June 2011.

[Nurmi et al. 2009] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, ser. CCGRID '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 124–131.

[PARKHILL 1966] PARKHILL, D. « The Challenge of the Computer Utility ». Addison-Wesley Educational Publishers Inc., US, 1966.

[Pries et al. 2011] R. Pries, M. Jarschel, D. Schlosser, M. Klopff, and P. Tran-Gia, "Power Consumption Analysis of Data Center Architectures," in *GreenNets*, Colmar, France, Oct. 2011.

[Schappi et al. 2010] B. Schappi, T. Bogner, B. P. S. Weeren, F. Bellosa, A. Anglade, and B. Harrison, "Guide d'achat et de gestion des équipements et des infrastructures pour des serveurs sobres en énergie", 2010. Austrian energy Agency, Université de Karlsruhe, Sun Micro system.

[Schwartz et al. 2012] C. Schwartz, R. Pries, P. Tran-Gia, " A Queuing Analysis of an Energy-Saving Mechanism in Data Centers", *ICOIN 2012*, pp 70-75, 2012.

[Sekhar&Jeba 2013] J. Sekhar and G. Jeba, "Energy efficient vm live migration in cloud data centers", *International Journal of Computer Science and Network (IJCSN)*, vol. 2, no. 2, p. 71-75, 2013.

[Shehabi et al. 2016] A. Shehabi, S. J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin and W. Lintner, "United States

DataCenter Energy Usage Report,” Lawrence Berkeley National Laboratory, Berkeley, California, 2016.

[Sing et al. 2017] Sukhpal Singh, Inderveer Chana and RajkumarBuyya. “STAR: SLA-aware autonomic management of cloud resources.” *IEEE Transactions on Cloud Computing*(2017)

[Stansberry 2014] M. Stansberry. *2014 Data Center Industry Survey*. Rapp. tech. Uptime Institute, 2014.

[Su et al. 2012] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, and J. Wang, “Energy aware virtual network embedding through consolidation,” in *INFOCOMWorkshops*, 2012, pp. 127–132.

[Sulmont et al. 2016], Rémi Sulmontet Loïc Farge, La "pollution numérique" pourrait rapidement dépasser celle de l'aéronautique, mis à jour le 12/02/2016, [En ligne], <https://www.rtl.fr/actu/debats-societe/la-pollution-numerique-pourrait-rapidement-depasser-celle-de-l-aeronautique-7781822692>.

[Wang et al. 2008] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, “Virtual routers on the move: live router migration as a network management primitive,” in *Proc. of SIGCOMM*, Seattle, WA, USA, 2008, pp. 231–242.

[Wang et al. 2010] G. Wang, D. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, “c-through: Part-time optics in data centers,” in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 327–338.

[Wiki-DC, consulté le 15/09/2017] "Centre de données", https://fr.wikipedia.org/wiki/Centre_de_donn%E9es, consulté le 15/09/2017.

[Yang 2013], Liu Yang, Jogesh K. Muppala and MalathiVeeraraghavan, “A Survey of Data Center Network Architectures.” , 2013.

[Young-Chan 2012], Lee, Young-Chan, “A Deployment Model for Cloud Computing using the Analytic Hierarchy Process and BCOR Analysis”, 18th Americas Conference on Information Systems, Août 2012.

[Yu et al. 2008] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” *SIGCOMMComput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008. consumption,” in *Proceedings of theACM SIGCOMM 2011 conference*, ser. SIGCOMM ’11. New York, NY, USA: ACM, Aug. 2011, pp. 338–349.

[Zhang et al. 2008] G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, and T. Zhou, “Evolution of the Internet and Its Cores,” *New Journal of Physics*, vol. 10, Dec. 2008.

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