



# Étude de la corrélation entre la topologie et les performances dans le Cloud Computing

MEMOIR DE FIN D'ÉTUDE EN INFORMATIQUE, OPTION :  
RSD

BENFRIHA SIHEM & BELLIFA SANA

L'ANCADREUR : MR. BELHOCINE

UNIVERSITÉ ABOU BEKR BELKAID



République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

*Option: Réseau et Système et Distribué(R.S.D)*

*Thème*

**Étude de la corrélation entre la topologie  
et les performances dans le Cloud**

**Réalisé par :**

- Benfriha Sihem
- Bellifa Sana

Présenté le 06 Juillet 2019 devant le jury composé de :

**Lehsaini Mohamed**

**(Président)**

**Benmammar Badr**

**(Examineur)**

**Belhocine Amin**

**(Encadreur)**

## **Remerciement**

Avant tout, je tiens à remercier le bon dieu de m'avoir aidé et m'avoir donné la force et le courage de mener à bien ce modeste travail, également je remercie infiniment mon mari et mes parents, qui m'ont encouragé et qui m'ont aidé à arriver à ce stade de ma formation.

Je voudrais dans un premier temps remercier, mon encadreur de mémoire Mr. Belhocine, pour sa patience, sa disponibilité, grâce à lui, j'ai pu arriver au but de mon projet. Je remercie aussi les enseignants Mr. Benmammam et Mr. Lehsaini pour avoir accepté d'être membre de jury dans notre soutenance.

Je remercie également tous les enseignants qui ont été avec moi pendant tout mon cursus universitaire. J'ai appris de chacun d'eux des choses qui vont m'aider dans ma vie.

## **Dédicace**

Je dédie ce travail qui n'aura jamais pu voir le jour sans les soutiens indéfectibles et sans limite de mes chers parents qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui. Que dieux vous protège et que la réussite soit toujours à ma portée pour que je puisse vous combler de bonheur.

Sans oublier de dédie ce travail à mon très cher mari qui m'a soutenu toute la période de mon mémoire, en m'encourageant et en me remontant le moral, que dieu le protège.

Je dédie aussi ce travail à :

Mes grands-parents qui ont prié chaque jour pour ma réussite et pour me voir heureuse aussi.

Mes chères sœurs et leur famille

Mon très cher et unique frère.

Toute ma famille qui m'a encouragé

Tous mes amis, mes collègues et tous ceux qui m'estiment.

# Table des matières

<b>Remerciement</b>	
<b>Dédicace</b>	
<b>Liste des figures</b>	
<b>Liste des tableaux</b>	
<b>Liste des abréviations</b>	
<b>Introduction générale</b> .....	1
<b>Chapitre 1</b>	
<b>Le Cloud Computing</b>	
1. Introduction.....	4
2. La naissance du Cloud et son évolution.....	4
3. La technologie Cloud Computing .....	5
3.1. Définition .....	5
3.2. Caractéristique du Cloud.....	5
4. Modèles de déploiement .....	6
4.1. Cloud publique.....	6
4.2. Cloud privé.....	8
4.3. Cloud hybride.....	8
4.4. Cloud communautaire .....	9
5. Les différents types de Cloud.....	9
5.1. IaaS.....	10
5.2. PaaS.....	10
5.3. SaaS.....	11
5.4. Serverless .....	11
5.5. DWaaS .....	12
5.6. DaaS .....	12
5.7. STaaS (Storage as a Service) .....	13
6. Les avantages .....	14
7. Exemple sur les domaines d'utilisations du Cloud .....	15
Le Cloud & La robotique .....	15
8. Les limites de Cloud.....	16
9. Conclusion .....	17
<b>Chapitre 2</b>	
<b>La consommation d'énergie &amp; les topologies</b>	
1. Introduction.....	19
2. Les Data Centers .....	19
3. Définition de la consommation d'énergie .....	19
4. Comment calculer la consommation d'énergie.....	19

5.	Raisons de la consommation élevée d'énergie des Data Center .....	20
6.	La virtualisation .....	20
6.1.	Techniques de virtualisation .....	21
6.2.	Les avantages de la virtualisation .....	22
6.3.	Les inconvénients de la virtualisation .....	23
7.	La migration.....	23
	Les avantages de la migration .....	25
8.	Les topologies .....	25
8.1.	Définition .....	25
8.2.	Les types .....	26
9.	Le SLA.....	33
10.	Conclusion .....	34

### **Chapitre 3**

#### **Implémentation et Simulation**

1.	Introduction.....	36
2.	Langage et environnement de développement .....	36
2.1.	Le langage de programmation Java.....	36
2.2.	L'environnement de développement.....	37
2.3.	Le simulateur CloudSim .....	37
2.3.1.	Définition .....	37
2.3.2.	L'architecture du CloudSim.....	37
3.	Réalisation de l'application.....	38
3.1.	La partie Présentation.....	38
3.2.	La partie Fonctionnement .....	41
3.3.	Persistance.....	43
4.	Résultats obtenues.....	44
5.	Comparaison .....	49
6.	Conclusion .....	50
	<b>Conclusion générale</b> .....	51

#### **Référence Bibliographie**

#### **Annexe**

## Liste des figures

Fig. 1.1 Chronologie simplifiée du Cloud .....	4
Fig. 1.2 Le symbole d'internet .....	5
Fig. 1.3 Le Cloud Computing .....	5
Fig. 1.4 Les modèles de déploiement du Cloud .....	6
Fig. 1.5 Cloud public .....	7
Fig. 1.6 Cloud privé .....	8
Fig. 1.7 Cloud hybride .....	9
Fig. 1.8 Cloud communautaire .....	9
Fig. 1.9 Les différents types du Cloud .....	10
Fig. 1.10 Le service IaaS .....	10
Fig. 1.11 PaaS .....	11
Fig. 1.12 Le SaaS .....	12
Fig. 1.13 Serverless .....	12
Fig. 1.14 Pourquoi les clients déplacent-ils leur DWaaS vers le Cloud .....	13
Fig. 1.15 DaaS .....	13
Fig. 1.16 Storage as a Service .....	13
Fig. 1.17 Top 10 des vendeurs Cloud Computing .....	13
Fig. 1.18 Cloud robotics .....	15
Fig. 1.19 Le robot Sophia .....	16
Fig. 2.1 Schéma de la virtualisation .....	20
Fig. 2.2 Schéma d'émulation .....	21
Fig. 2.3 Virtualisation complète .....	21
Fig. 2.4 Paravirtualisation .....	22
Fig. 2.5 L'architecture d'hyperviseur .....	22
Fig. 2.6 La migration .....	24
Fig. 2.7 Processus de la migration à froid .....	24
Fig. 2.8 Migration à chaud .....	25
Fig. 2.9 La topologie .....	25
Fig. 2.10 Topologie en bus .....	26
Fig. 2.11 Topologie en étoile .....	26
Fig. 2.12 Topologie anneau .....	27
Fig. 2.13 Topologie en arbre .....	27
Fig. 2.14 La topologie Fat-tree .....	28

Fig. 2.15 Topologie maillée .....	29
Fig. 2.16 Topologie hybride .....	29
Fig. 2.17 L'architecture de la topologie BCube .....	30
Fig. 2.18 Topologie DCell .....	31
Fig. 2.19 Topologie FiConn .....	32
Fig. 2.20 Une vue verticale de la topologie DPillar .....	32
Fig. 2.21 Topologie DPillar .....	33
Fig. 2.22 La structure de la topologie VL2 .....	33
Fig. 3.1 Schéma qui représente l'architecture de CloudSim .....	38
Fig. 3.2 L'interface de la page d'accueil .....	38
Fig. 3.3 L'interface de la configuration du Data Center .....	39
Fig. 3.4 L'interface de la configuration des Hôtes .....	39
Fig. 3.5 La configuration des VMs .....	40
Fig. 3.6 l'interface des topologies utilisées .....	40
Fig. 3.7 l'interface pour afficher le résultat .....	41
Fig. 3.8 Schéma d'interaction entre les classes (DCell) .....	43
Fig. 3.9 La base de données de notre application .....	44
Fig. 3.10 Un aperçu sur le fichier bride .....	44
Fig. 3.11 Host graph DCell .....	45
Fig. 3.12 Switch graph DCell .....	45
Fig. 3.13 Data center graph DCell .....	45
Fig. 3.14 Host graph BCube .....	46
Fig. 3.15 Switch graph BCube .....	46
Fig. 3.16 Data center graph BCube .....	46
Fig. 3.17 Host graph Fat-tree .....	47
Fig. 3.18 Switch graph Fat-tree .....	47
Fig. 3.19 Data center graph Fat-tree .....	47
Fig. 3.20 Host graph VL2 .....	48
Fig. 3.21 Switch graph VL2 .....	48
Fig. 3.22 Data center graph VL2 .....	48
Fig. 3.23 Graphe des quatre topologies .....	49
Fig. 3.24 Graphe de pourcentage d'éléments réseau pour chaque topologie .....	49

## **Liste des tableaux**

Tab. 1.1 Quelques exemples de fournisseurs avec le type du Cloud utilisé .....	14
Tabl. 3.1 Tableau de résultats de DCell .....	45
Tabl. 3.2 Tableau de résultats de BCube .....	46
Tabl. 3.3 Tableau de résultats de Fat-tree .....	47
Tabl. 3.4 Tableau de résultats de VL2 .....	48

# Liste des abréviations

## **A**

API: Application Programming Interface.

ASP : Application Service Provider, fournisseur de services d'applications.

AWS: Amazon Web Services.

## **B**

BSD: Berkeley Software Distribution.

## **C**

CDDL: Common Development and Distribution License.

## **D**

DaaS: Desktop as a Service.

DWaaS: Data Warehousing as a Service.

## **E**

EoR: End-of-Row.

## **F**

FaaS: Functions as a Service.

FDDI: Fiber Distributed Data Interface.

## **I**

IaaS: Infrastructure as a service.

IBM: International Business Machines Corporation.

IDC: International Data Corporation.

IDE: Integrated Development Environment.

IT: Information Technology.

## **M**

MAU: Multistation Access Unit.

MDC: Modular Data Center.

## **O**

ONG: organisation non gouvernementale.

OS: Operating System.

## **P**

PaaS: Platform as a Service.

PDA: Personal Digital Assistant.

## **R**

RaaS: Robot as a Service.

## **S**

SaaS: Software as a Service.

SE: Système d'Exploitation.

SLA: Services Level Agreement.

STaaS: Storage as a Service.

## **T**

ToR: Top of Rack.

## **V**

VDI: Virtual Desktop Infrastructure.

VM: Virtual Machine.

VMM: Virtual Machine Monitor

VPN: Virtual Private Network.

# Introduction générale

**L**e Cloud Computing est une technologie qui est très connue dans le monde informatique, il a plusieurs services qui sont utilisés selon le besoin d'utilisateur. Des services comme le calcul, le logiciel, l'accès aux données et le stockage sont fournis sans connaissance préalable de leur emplacement et leur mode de fonctionnement. La base du Cloud c'est les Data Centers, là où les services sont traités. Ils contiennent un ensemble de serveurs, des équipements réseaux tel que les commutateurs...etc., et des systèmes de refroidissement qui sont les plus responsables en termes de consommation d'énergie.

La consommation d'énergie joue un rôle très important, le fait qu'elle soit très élevée, augmente les coûts d'exploitation pour les fournisseurs, mais le côté qui est plus négatif c'est la grande quantité de CO2 qui est émise. Ce problème provoque l'augmentation de réchauffement de la planète. Il n'est pas dû uniquement par le grand nombre de ressources physiques, mais aussi par l'inefficacité d'utilisation des ressources physiques et par le choix de leur emplacement dans un réseau. Il y a aussi un autre point important pour le fournisseur et qui est le SLA (qualité de service) qui est un contrat dont le respect est nécessaire au risque d'avoir un problème avec le client, engendrant une augmentation des frais pour le fournisseur.

Pour le problème de la consommation d'énergie, il existe des solutions qui sont utilisées. Les plus classiques sont la virtualisation et la migration. La virtualisation est une technologie qui permet aux fournisseurs de Cloud de résoudre le problème de l'inefficacité énergétique. Son rôle est de créer plusieurs instances de machines virtuelles dans un même serveur physique ce qui minimise leur nombre dans un Data Center. Les VMs peuvent être transférées vers des autres serveurs physiques en utilisant la migration.

D'autres facteurs peuvent avoir un rôle important dans la consommation d'énergie dans le cloud dont la topologie. Cette dernière détermine la structure du réseau ainsi que l'emplacement des équipements réseau et des machines physiques dans le Data Center. Il existe plusieurs types de topologie ; classiques tel que la topologie en bus, étoile...etc., ou plus aboutis tel que : DCell, BCube, FiConn...etc. chaque topologie impacte différemment la consommation d'énergie.

Le but de cette étude est de mettre en évidence l'influence de la topologie dans la consommation d'énergie dans le cloud, ainsi que la détermination de la meilleure structure parmi celles étudiées.

Nous allons présenter dans ce qui suit un aperçu de chaque chapitre :

**Chapitre 1** : Dans ce chapitre, nous présenterons le Cloud Computing et tout ce que nous devons savoir sur cette technologie. Commencant par sa description et un brève historique sur son apparition et sa continuation jusqu'à maintenant. Nous passerons ensuite vers ses différents types et ses services proposés, bien sûr nous parlerons un peu de quelques domaines d'utilisation et ses grands fournisseurs, sans oublier ses avantages et ses limites.

**Chapitre 2** : Cette partie est consacrée pour la consommation d'énergie, en commençant par les Data Centers vu que c'est eux qui consomment, en passant par les solutions utilisées pour réduire la consommation d'énergie. Nous citerons plusieurs méthodes dont deux technologies qui sont la virtualisation et la migration. Le plus

important pour nous dans ce travail c'est les topologies qui représentent un autre facteur important dans la consommation d'énergie.

**Chapitre 3** : Le dernier chapitre dans notre mémoire, inclut toutes les informations à propos de notre application. Nous présenterons le langage utilisé Java et son environnement de développement NetBeans, le simulateur CloudSim, son architecture ainsi que ses classes avec lesquels nous allons travailler. Nous terminerons notre chapitre par les résultats obtenus et une comparaison entre eux.

# Chapitre 1

## Le Cloud Computing

---

### Sommaire

1. Introduction.....	4
2. La naissance du Cloud et son évolution.....	4
3. La technologie Cloud Computing.....	5
3.1. Définition.....	5
3.2. Caractéristique du Cloud.....	5
4. Modèles de déploiement.....	6
4.1. Cloud publique.....	6
4.2. Cloud privé.....	8
4.3. Cloud hybride.....	8
4.4. Cloud communautaire.....	9
5. Les différents types de Cloud.....	9
5.1. IaaS.....	10
5.2. PaaS.....	10
5.3. SaaS.....	11
5.4. Serverless.....	11
5.5. DWaaS.....	12
5.6. DaaS.....	12
5.7. STaaS (Storage as a Service).....	13
6. Les avantages.....	14
7. Exemple sur les domaines d'utilisations du Cloud.....	15
Le Cloud & La robotique.....	15
8. Les limites de Cloud.....	16
9. Conclusion.....	17

## 1. Introduction

Internet nous a facilité la vie dans tous les domaines, il nous aide pour le stockage de nos informations et nos données avec la possibilité de les récupérer grâce au Cloud, qu'on utilise presque tous les jours sans nous rendre compte. Non pas uniquement pour le stockage des données mais aussi pour la messagerie électronique comme le mail et bien d'autres services sophistiqués, exploitables à volonté depuis n'importe où dans le monde à partir du moment où l'on dispose d'une connexion Internet.

Le Cloud Computing est devenu très important, sans lui, on ne peut pas être tel qu'on est aujourd'hui, car toutes les technologies travaillent avec, les smartphones, les smartTVs, les robots surtout les nouveaux robots, dans les domaines téléphoniques ... etc.

Donc le Cloud Computing est devenu une tendance dont on parle souvent, mais qu'on ne voit pas, du coup ça nous pose beaucoup de questions. C'est quoi ce Cloud Computing, qu'est-ce que nous voulons dire par ce mot ? Qu'est-ce qu'il fait exactement, à quoi sert-il ? C'est quoi ces services et comment il fonctionne ? Est-ce qu'il a des avantages et des inconvénients, ou plutôt a-t-il des limites ?

Dans ce chapitre, on va répondre à toutes ces questions, du coup, on va avoir une idée générale sur ce Cloud et son fonctionnement pour pouvoir le maîtriser, en plus de ça, on va voir quelques exemples pour mieux le comprendre.

## 2. La naissance du Cloud et son évolution

L'idée du Cloud Computing a pris naissance en 1990 et surtout en 1991 par l'émergence des ASP [1] avec la naissance d'Internet et la mise sur le marché du logiciel CERN<sup>1</sup> qui a été le premier logiciel accessible par le Web. Ensuite, l'idée a progressé avec l'apparition de nouvelles solutions IT<sup>2</sup>, le lancement du navigateur Mosaic<sup>3</sup> en 1993 et celui du navigateur Netscape<sup>4</sup> en 1994. En 1995, la découverte d'EBay et d'Amazon a encore accéléré le processus. Enfin, le lancement en 1996 de l'assistant Palm PDA<sup>6</sup> a donné une nouvelle impulsion. En 2000, l'arrivée de Google a favorisé une réelle émergence du Cloud computing. Dans l'avenir, le Cloud computing devrait connaître une belle croissance car moins de 10% des sociétés y avaient recours en 2009. On prévoit que 30 à 50 % des sociétés devraient utiliser ce service bien que les entreprises semblent préférer le Cloud privé. Il se peut que l'arrivée de nouvelles solutions IT provoque une accélération du phénomène [2].

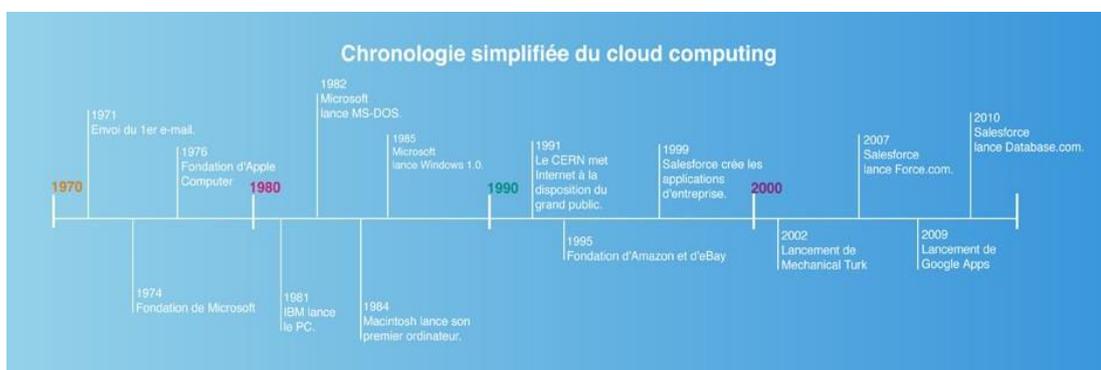


Fig. 1.1 : Chronologie simplifiée du Cloud [3]

## 3. La technologie Cloud Computing

### 3.1. Définition

- Le Cloud Computing ou l'informatique dans le nuage en français : c'est le fait d'utiliser le processeur, non pas la machine physique mais plutôt la puissance de la machine [4], Il permet d'accéder à des ressources informatiques qui sont quelque part (tel que des photos, des fichiers...etc.), à travers Internet, ça peut être gratuit comme c'est le cas avec Webmail<sup>6</sup>, ou sur abonnement, avec un niveau de service garanti [5]. L'origine de son concept provient du fait que les informaticiens symbolisent Internet sous la forme d'un nuage dans leurs schémas [1].

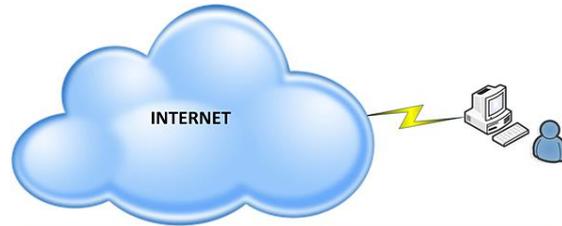


Fig. 1.2 Le symbole d'Internet [6]

- Le Cloud Computing est une technologie qui permet de mettre sur des serveurs localisés à distance des données de stockage ou des logiciels qui sont habituellement stockés sur l'ordinateur d'un utilisateur, voire sur des serveurs installés en réseau local au sein d'une entreprise.

Cette virtualisation des ressources permet donc à l'entreprise d'accéder à ses données sans avoir à gérer une infrastructure informatique, souvent complexe et qui représente un certain coût pour l'entreprise.

Le Cloud Computing est considéré par beaucoup, comme une évolution majeure de l'informatique et qui permet d'accéder depuis n'importe où à vos fichiers [7].

- Le cloud computing, en français l'informatique en nuage ou nuagique ou encore l'infonuagique (au Québec), consiste à exploiter la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet. Les serveurs sont loués à la demande, le plus souvent par tranche d'utilisation, selon des critères techniques (puissance, bande passante, etc.), mais, également, au forfait. [8].

- Le Cloud Computing est un modèle qui permet un accès omniprésent, pratique et à la demande, à un réseau partagé et à un ensemble de ressources informatiques configurables (comme par exemple : des réseaux, des serveurs, du stockage, des applications et des services) qui peuvent être provisionnées et libérées avec un minimum d'administration. Ce modèle est composé de 5 caractéristiques essentielles, de 3 modèles de services et de 4 modèles de déploiement.



Fig. 1.3 Le Cloud Computing [10]

On pourrait résumer de cette manière : « le Cloud Computing c'est de pouvoir utiliser des ressources informatiques sans les posséder » [9].

### 3.2. Caractéristique du Cloud

Le Cloud Computing a pour but d'offrir des services qui vont au-delà de ces offres classiques et que nous pouvons tenter de définir avec les cinq caractéristiques suivantes:

- Il s'agit d'une informatique distribuée ou les échanges sont gérés et centralisés par des serveurs distants, les applications étant stockées non plus sur le poste de travail,

mais Sur un «Cloud» de serveurs, accédées par une connexion Internet et un navigateur web.

- Les applications, plate formes et infrastructures nécessaires sont louées en fonction de l'usage qui en est fait, que ce soit pendant le développement de ces applications ou pendent leurs utilisations en production.
- Les applications, plateformes et infrastructures sont facilement extensibles.
- Les ressources peuvent être allouées dynamiquement en fonction du besoin.
- Les applications, plateformes et infrastructures restent disponibles en cas de panne d'une ressource [11].

## 4. Modèles de déploiement

Le Cloud Computing se catégorise en 4 modèles de déploiement, ils sont :

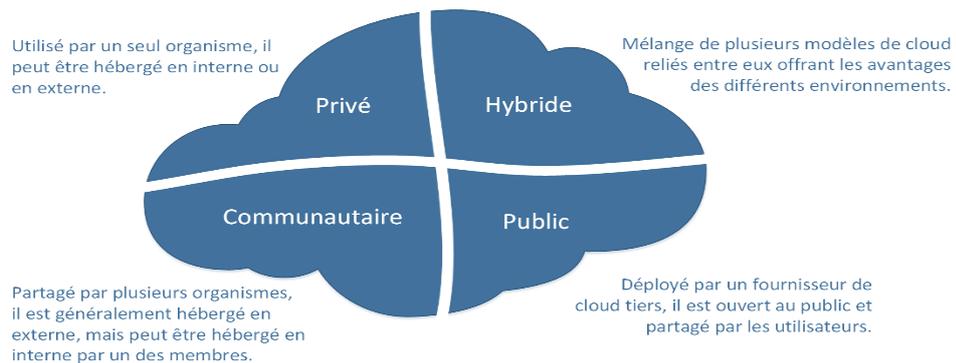


Fig. 1.4 Les modèles de déploiement du Cloud [12]

### 4.1. Cloud public

En tant qu'utilisateurs personnels, nous accédons via des services à des données stockées chez des fournisseurs/hébergeurs avec qui nous avons contractualisé le service et les niveaux de services associés. Ces fournisseurs/hébergeurs ayant une accessibilité directe sur Internet, nous parlons alors de Cloud public.

En première approximation, nous pouvons considérer que Cloud public = Internet !

Exemple : les réseaux sociaux, notre messagerie privée, l'extranet de notre société.

**Remarque** : il n'est pas toujours facile en tant qu'utilisateur de savoir si l'on se connecte réellement à un Cloud public [1].

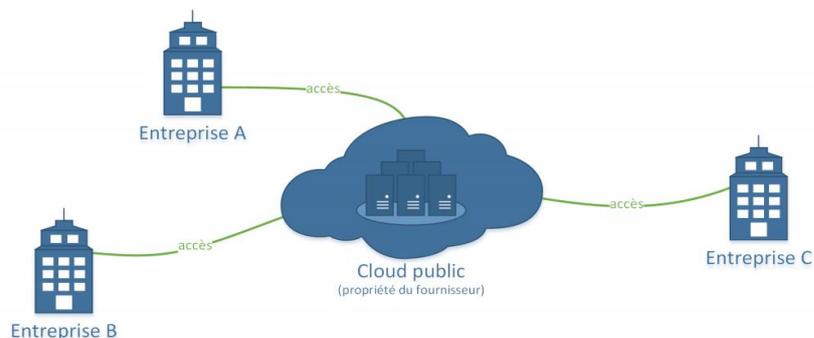


Fig. 1.5 Cloud public [12]

### Les fournisseurs du Cloud public :

Trois principaux fournisseurs de Cloud se distinguent en effet par l'étendue de leurs infrastructures: Amazon Web Services, Microsoft et Google. Ils disposent chacun au minimum, de 45 Data Centers dans le monde.

#### Amazon Web Services



Dans son Magic Quadrant<sup>7</sup>, Gartner<sup>8</sup> place Amazon Web Services dans le quadrant « Leaders » pour la 7<sup>e</sup> année consécutive. AWS est sans conteste la référence dès qu'on évoque le Cloud et en particulier l'IaaS. Sur ce segment qui a connu la plus forte croissance en 2016, Amazon détient près de 45 % du marché

AWS présente deux atouts déterminants :

Premièrement, son interface graphique : AWS Management Portal for vCenter<sup>9</sup> permet de bénéficier d'un chemin de migration des VMs (et notamment les VM VMware) très clair. Les instances des VMs peuvent être hébergées dans des régions spécifiques. En Europe, AWS dispose de 13 Data Centers, dont trois dans l'Hexagone.

Deuxième atout : sa fonction Enhanced Networking<sup>10</sup> assure une connectivité réseau qui est plus performante (faible latence notamment) que celle proposée en standard [13].

#### Microsoft



Concurrent direct d'AWS, Microsoft a créé son Cloud public Azure<sup>11</sup> par-dessus Windows Server et Hyper-V<sup>12</sup>. Cette proximité logicielle facilite la migration des VMs entre les Data Centers locaux et Azure. Il est possible également de connecter ce dernier à votre réseau d'entreprise via un VPN point à point.

Sur le marché de l'IaaS, l'approche de Microsoft est complète, surtout après le lancement d'Azure<sup>13</sup> Stack, sa plateforme de déploiement de Cloud hybride.

Microsoft a défini 17 régions pour Azure, situées un peu partout aux États-Unis, en Europe, en Asie, en Amérique du Sud et en Australie. L'éditeur possède deux Data Centers en France [13].

#### Google



Google Compute Engine<sup>14</sup> offre toutes les fonctionnalités de base de la connectivité réseau d'un Cloud, directement ou via un VPN. Mais il ne prend pas en charge les migrations des VMs dans le Cloud Google Compute Engine. Il est nécessaire de passer par des fournisseurs tiers.

Google permet de choisir les régions d'hébergement des VMs. Le géant a des Data Centers en Europe (Belgique, Royaume-Uni, Allemagne, Pays-Bas et Finlande). Mais pas en France [13].

## Rackspace



Racheté pour 4,3 milliards de dollars par Apollo Global Management<sup>15</sup>, l'hébergeur Rackspace ambitionne de devenir une référence en tant que fournisseur de services et spécialiste du Cloud managé.

Co-créateur avec la Nasa du système OpenStack, il ne permet pas de migrer les VM vers le Cloud ou à partir du Cloud, ni de retenir une zone en particulier pour leur hébergement.

Rackspace propose plusieurs options de connectivité réseau. Ce spécialiste possède des Data Centers aux États-Unis, en Angleterre, à Hong Kong et en Australie [13].

### 4.2. Cloud privé

Un Cloud privé est une infrastructure entièrement dédiée à une entreprise unique, pouvant être gérée en interne ou par un tiers, et hébergée en interne ou en externe. Ce modèle offre une versatilité aux entreprises, tout en préservant la gestion, le contrôle et la sécurité. Les avantages sont l'accès en self-service à l'interface de contrôle, permettant à l'équipe informatique un approvisionnement rapide, et l'allocation ou la livraison de ressources informatiques à la demande. De même, la gestion des ressources est automatisée, aussi bien pour le stockage ou l'analyse. De même, la sécurité et la gouvernance sont conçues sur mesure pour les besoins spécifiques de l'entreprise [14].

En première approximation, nous pouvons considérer que Cloud privé = réseau interne !

Exemple : le logiciel de gestion de l'organisation [1].

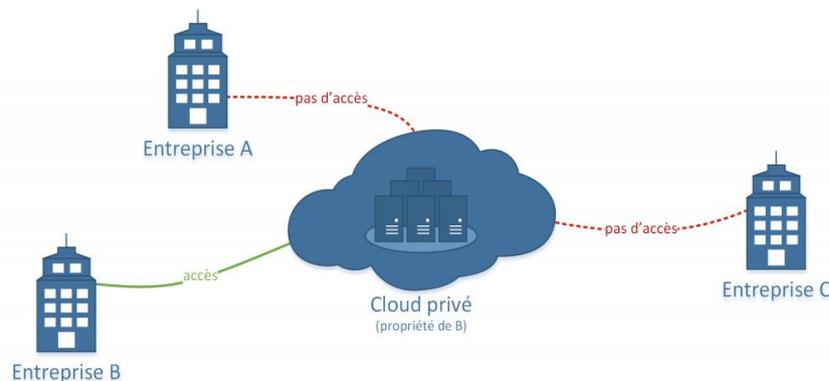


Fig. 1.6 Cloud privé [12]

### 4.3. Cloud hybride

Enfin, le Cloud Hybride est le croisement entre le Cloud public et le Cloud privé. Les entreprises peuvent par exemple effectuer des tâches très importantes ou des applications sensibles sur le Cloud privé, et utiliser le Cloud public pour les tâches nécessitant une scalabilité des ressources. L'objectif du Cloud hybride est de créer un environnement unifié, automatisé et scalable tirant avantage des infrastructures de Cloud public tout en maintenant un contrôle total sur les données [14].

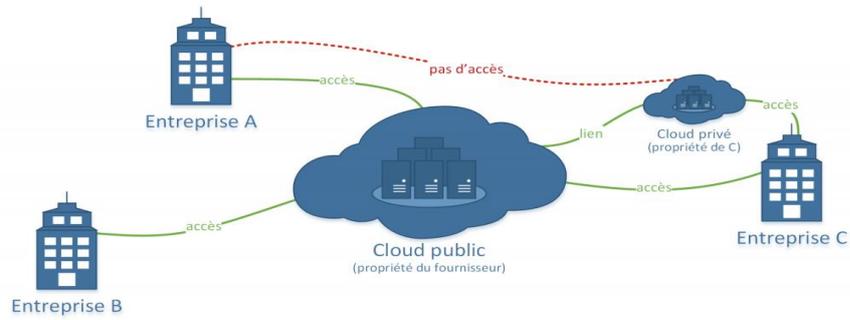


Fig. 1.7 Cloud hybrid [12]

#### 4.4. Cloud communautaire

C'est un nouveau type du Cloud plus récent, qui est multi-tenant<sup>16</sup> partagé entre plusieurs entreprises, il est régi, géré et sécurisé par un groupe de participants ou par un fournisseur de service. Le Cloud communautaire est sous forme hybride de plusieurs Clouds privés interconnectés sans ouverture vers l'extérieur. [12]

Exemple : des hôpitaux partageant des dossiers de patients, un logiciel de gestion mutualisé ... etc. [1]

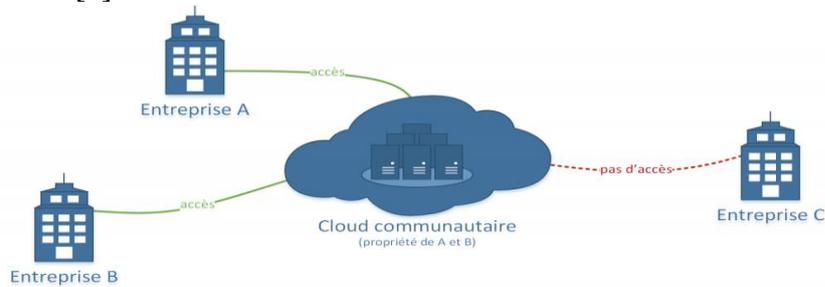


Fig. 1.8 Cloud communautaire [12]

### 5. Les différents types de Cloud

La plupart des services de Cloud Computing sont classés en quatre grandes catégories : IaaS, PaaS, serverless et SaaS. Elles s'appellent aussi pile de Cloud, car elles s'empilent les unes sur les autres.

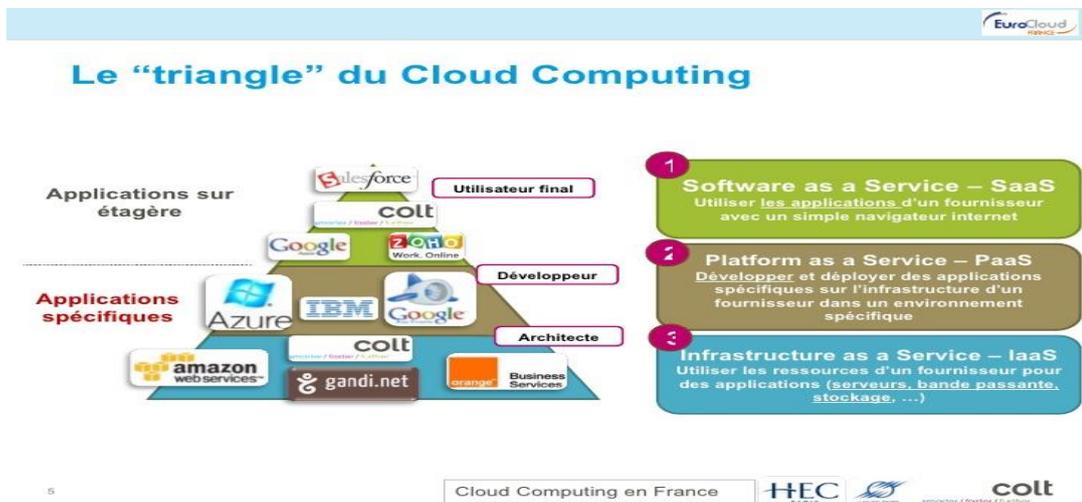


Fig. 1.9 Les différents types du Cloud [15]

## 5.1. IaaS



Fig. 1.10 Le service IaaS [18]

La catégorie la plus basique des services de Cloud Computing [16]. Il permet de fournir aux clients un socle d'infrastructure informatique virtualisé, distribué et automatisé [17] auprès d'un fournisseur de services Cloud [16]. Il est composé d'un ensemble de ressources sous forme de serveurs, de réseaux, de stockage dédié. On considère que l'unité de base dans ce Cloud est la machine virtuelle.

Il est responsable de la mise à disposition d'une infrastructure de traitement complète pour l'utilisateur ou le client. Celui-ci peut se connecter à une ou plusieurs machines virtuelles sur lesquelles il peut augmenter la taille de stockage très facilement, obtenir une capacité de calcul plus importante [17] et un paiement en fonction de l'utilisation [16]. Exemple: Amazon Web Services AWS, Windows Azure, Google Computer Engine [17].

## 5.2. PaaS

Plateforme en tant que service est un environnement de développement et de déploiement complet dans le cloud, avec les ressources nécessaires pour vous permettre de fournir n'importe quel service, de la simple application cloud aux applications d'entreprise sophistiquées [19]. Il qualifie les services de Cloud computing qui offrent un environnement à la demande pour développer, tester, fournir et gérer des applications logicielles [16]. Vous faites l'acquisition des ressources dont vous avez besoin auprès d'un fournisseur de services cloud, avec un paiement à l'utilisation, et vous y accédez via une connexion Internet sécurisée.

PaaS est conçu pour permettre aux développeurs de créer rapidement des applications web ou mobiles sans avoir à se préoccuper de la configuration ou de la gestion de l'infrastructure de serveurs, de stockage, de réseau et de bases de données nécessaire au développement[16]. Exemple : Google Apps, Microsoft Azure [17].

Tout comme IaaS, PaaS comprend l'infrastructure, à savoir les serveurs, le stockage et les composants réseau, mais aussi les intergiciels<sup>17</sup> (*Middleware*), les outils de développement, les services d'aide à la décision (BI, Business Intelligence), les systèmes de gestion de bases de données, etc. Le PaaS est conçu pour prendre en charge l'intégralité du cycle de vie de l'application web : conception, test, déploiement, gestion et mise à jour.

Le PaaS permet d'éviter les dépenses et problèmes liés à l'achat et à la gestion de licences logicielles, de l'infrastructure sous-jacente aux applications et du middleware ou des outils de développement et autres ressources. Vous gérez les services et les applications que vous développez, et le fournisseur de services cloud se charge en général du reste [16].



Fig. 1.11 PaaS [20]

### 5.3. SaaS

Le logiciel en tant que service est une méthode de diffusion d'applications logicielles via Internet, à la demande et en général sur abonnement [16]. Les exemples les plus courants sont les outils de messagerie comme Outlook, Hotmail ou Yahoo !, de calendrier et les outils de bureautique comme Microsoft Office 365 [21]. Il permet à une entreprise d'accéder à une application en mode hébergé indépendante de la plateforme d'infrastructure ou logicielle [17].

Le SaaS offre une solution logicielle complète pour laquelle vous payez en fonction de l'utilisation. Vous louez l'utilisation d'une application pour votre organisation, et vos utilisateurs s'y connectent par Internet, en général avec un navigateur web sur téléphone, sur tablette ou sur PC [16].

#### Son fonctionnement :

Le SaaS est lié aux modèles de livraison de service de logiciel ASP et l'informatique à la demande. Avec ce modèle, le fournisseur offre aux clients un accès basé sur le réseau à une simple copie d'une application spécifiquement créée par le fournisseur pour la distribution SaaS. Le code de l'application est le même pour tous les clients. Quand les nouvelles fonctionnalités sont déployées, tous les clients peuvent en profiter.

En fonction du niveau de service, les données du client peuvent être stockées localement, sur le Cloud, ou les deux à la fois. Les entreprises peuvent intégrer des applications à d'autres logiciels en utilisant des API. Par exemple, une entreprise peut développer ses propres outils logiciels et utiliser l'API du fournisseur de service pour intégrer ces outils à l'offre [22].



Fig. 1.5.12 Le SaaS [22]

Il y en a d'autres modèles qui sont présentés ci-dessous:

### 5.4. Serverless



Fig 1.13 Serverless [23]

Se chevauchant avec PaaS, l'informatique serverless se concentre sur la création de fonctionnalités applicatives sans perte de temps en lien avec la gestion permanente des serveurs et de l'infrastructure requis à cette fin. Le fournisseur de Cloud se charge de la configuration, de la planification de la capacité et de l'administration du serveur à votre place. Les architectures Serverless sont hautement scalables et basées sur

des événements. Elles n'utilisent des ressources que quand une fonction ou un déclencheur spécifique s'activent.

L'architecture serverless est un modèle dans lequel le fournisseur de services Cloud est responsable de l'exécution d'un morceau de code en allouant de manière dynamique les ressources. Et il ne facture que la quantité de ressources utilisées pour exécuter le code. Le code est généralement exécuté dans des conteneurs sans état pouvant être déclenchés par divers événements, notamment des requêtes http, des événements de base de données, des services de file d'attente, des alertes de surveillance, des téléchargements de fichiers, etc. le code envoyé au fournisseur de Cloud pour l'exécution est généralement sous la forme d'une fonction. Par conséquent, serverless est parfois appelé « «Functions as a Service » ou « FaaS » [23].

### 5.5. DWaaS

C'est un modèle d'impartition dans lequel un fournisseur de services configure et gère les ressources matérielles et logicielles requises par un entrepôt de données. Le client fournit les données et paie le service géré. Avec DWaaS, une entreprise n'a pas à déboursier d'argent pour créer un entrepôt de données sur site ni à se préoccuper de la configuration et de la gestion du matériel et des logiciels nécessaires [25]. Le DWaaS est un cousin de SaaS, il est utilisé pour faciliter l'accessibilité en toute sécurité aux données essentielles impliquant les activités métiers [17].



Fig. 1.14 Pourquoi les clients déplacent-ils leur DWaaS vers le Cloud [24]

### 5.6. DaaS



Fig. 1.15 DaaS [26]

Cette plateforme peut être considérée comme une solution prête pour fournir un poste de travail virtualisé et hébergé chez le fournisseur. Ce concept, issu du modèle client-serveur, consiste à déporter la gestion des environnements de travail (éventuellement des applications) dans le Cloud Computing. L'avantage de cette solution est tel qu'elle peut être distribuée sur des postes de travaux classiques, des ordinateurs portables, smartphones ou tablettes [17]. Le DaaS permet aux entreprises de réaliser rapidement tout le potentiel et les avantages associés aux environnements de bureau virtuel (VDI) – réduction de la complexité du déploiement ; amélioration de la gestion, de la sécurité et de la conformité ; et une productivité supérieure des utilisateurs finaux – sans les dépenses en capital et l'intégration de système complexe pour la création et le déploiement d'une solution interne personnalisée [26].

## 5.7. STaaS (Storage as a Service)

C'est un service qui permet de fournir la capacité de stockage de données à l'aide d'un accès Internet.

Il s'agit d'un modèle technologique qui propose de l'espace de stockage en ligne et dans lequel un prestataire de services propose de louer l'espace de son infrastructure de stockage sur une base d'abonnement mensuel ou annuel [17].



Fig. 1.16 Storage as a Service [27]

# CLOUD WARS

## Top 10 Rankings – Nov. 7, 2017

1. <b>Microsoft</b> – Nadella on \$20.4B run rate w/ <u>end-to-end customer-centric cloud</u>
2. <b>Amazon</b> – AWS needs software! <u>10 software companies Amazon might look at</u>
3. <b>IBM</b> – Rometty strikes gold helping customers convert legacy IT to private cloud
4. <b>Salesforce</b> – Benioff must extend SFDC impact from SaaS deeply into PaaS
5. <b>SAP</b> – McDermott accelerating major product-line overhaul to HANA and cloud
6. <b>Oracle</b> –Ellison on cybercrime: <u>'Make no mistake, this is a war—and we're losing'</u>
7. <b>Google</b> – Tons of potential but still unclear if/how it wants to play in enterprise
8. <u>ServiceNow</u> – Jumps ahead of Workday: revenue up 40%, new products boom
9. <b>Workday</b> – Q2 revenue surges 41% as <u>Bhusri</u> jumps into PaaS marketplace
10. <b>VMware</b> – revenue & stock jump on deals w/AMZN MSFT IBM GOOG for hybrid

Fig. 1.17 Top 10 des vendeurs Cloud Computing [28]

<b>Fournisseur</b>	<b>Type d'offre</b>	<b>Nom de l'offre</b>	<b>Type de paiement</b>	<b>Tarif</b>	<b>Caractéristiques principales</b>
<b>Amazon</b>	IaaS	EC2 (Elastic Compute Cloud)	A la consommation	A partir de quelques centimes d'euro par heure	Location de serveurs à la demande : Puissance de calcul, stockage, OS divers (Windows Serveur, Linux, OPenSolaris, Oracle, etc.).
<b>Google</b>	SaaS	Google Apps for Business	Abonnement annuel par utilisateur	40 € (Google Apps)	Plate-forme applicative : Gmail (messagerie), Gtalk (messagerie instantanée) Google Agenda, Google Docs (partage de documents), Google Grops (listes de diffusion), Google Sites (intranet), Google Vidéos pour entreprises.
<b>HP</b>	IaaS	CloudSystem		NC	Composants matériels et logiciels (serveurs, baies de stockage, composants réseaux, etc.)
<b>IBM</b>	SaaS	Lotus Life	Abonnement	A partir de 0.09 €/heure	Messagerie collaborative.
<b>Microsoft</b>	SaaS	Office 365 (disponible en juin)	Abonnement mensuel par utilisateur	De 5.25 à 22€	Messagerie (Exchange), partage de documents (Sharepoint), communications unifiées (Live Meeting), logiciels bureautique (suite Office).
<b>Salesforce.com</b>	SaaS	Chatter	Abonnement mensuel par utilisateur	Gratuit	Réseau social d'entreprise.

Tab. 1.1 Quelques exemples de fournisseurs avec le type du Cloud utilisé [29]

## 6. Les avantages

Le Cloud Computing peut permettre d'effectuer des économies, notamment grâce à la mutualisation des services sur un grand nombre de clients. Certains analystes indiquent que 20 à 25 % d'économies pourraient être réalisées par les gouvernements sur leur budget informatique s'ils migraient vers le Cloud Computing. Comme pour la

virtualisation, l'informatique dans le nuage peut être aussi intéressante pour le client grâce à son évolutivité. En effet, le coût est fonction de la durée de l'utilisation du service rendu et ne nécessite aucun investissement préalable (homme ou machine).

L'« élasticité » du nuage permet de fournir des services évolutifs et peut permettre de supporter des montées en charge. Inversement, le fournisseur a la maîtrise sur les investissements, est maître des tarifs et du catalogue des offres et peut se rémunérer d'autant plus facilement que les clients sont captifs.

A titre d'exemple, une entreprise possédant une boutique en ligne pourra facilement mettre en œuvre des serveurs supplémentaires pour faire face à un pic d'activité très limité dans le temps, tel que la période de Noël ou des soldes, puis les supprimer après coup. Cela lui reviendra certainement moins cher que si elle avait dû acheter et gérer toute l'année une infrastructure informatique capable d'absorber cette charge importante mais éphémère.

L'abonnement à des services de Cloud Computing peut permettre à l'entreprise de ne plus avoir à acquérir des actifs informatiques comptabilisés dans le bilan sous forme de CAPEX<sup>18</sup> et nécessitant une durée d'amortissement. Les dépenses informatiques peuvent être comptabilisées en tant que dépenses de fonctionnement.

La maintenance, la sécurisation et les évolutions des services étant à la charge exclusive du prestataire, dont c'est généralement le cœur de métier, celles-ci ont tendance à être mieux réalisées et plus rapidement que si elles étaient sous la responsabilité du client (principalement lorsque celui-ci n'est pas une organisation à vocation informatique) [30].

## 7. Exemple sur les domaines d'utilisations du Cloud

### Le Cloud & La robotique

Pour parler du Cloud et de la robotique, il faut parler de RaaS et Cloud robotics, ces eux qui nous permettent de réunir ces deux termes. Mais que cela veut dire ?



Fig. 1.18 Cloud robotics [31]

**Le RaaS :** un nouveau service du Cloud, qui sert à combiner le Cloud Computing, l'intelligence artificielle, la robotique et les services partagés. Ce service permet de stocker les données collectées par les robots sur le Cloud. Il est utile surtout pour les entreprises qui utilisent des centaines de robots sur le Cloud

**Le Cloud robotics :** c'est un autre concept qui réunit le Cloud et la robotique. Il s'agit de d'utiliser les ressources de Cloud

Computing pour augmenter la mémoire, la puissance de calcul, l'apprentissage collectif et l'inter-connectivité<sup>19</sup> des applications robotiques, ce qui aide les robots a partagé leurs connaissances entre eux [31].

**Exemple :** le robot Sophia qui est un humanoïde<sup>20</sup> robot, il est développé par la compagnie Hanson Robotics qui se trouve à Hong-Kong, ce robot est basé sur le Cloud, car il stocke chaque information, chaque image et même chaque geste, c'est sa particularité, d'imiter l'être humain, dans la tristesse, la joie, pour rigoler aussi et faire des blagues à son niveau, elle a fait déjà des interviews ce qui confirme sa capacité d'enregistrer ce qu'elle capte. Elle est arrivée au point d'avoir une nationalité, alors que ça n'a jamais été arrivé pour n'importe quel robot.



Fig. 1.19 le robot Sophia [32]

## 8. Les limites de Cloud

Plusieurs catégories d'inconvénients existent :

- l'utilisation des réseaux publics, dans le cas du Cloud public, entraîne des risques liés à la sécurité du Cloud. En effet, la connexion entre les postes et les serveurs applicatifs passe par le réseau Internet et expose à des risques supplémentaires de cyberattaques et de violation de confidentialité. Le risque existe pour les particuliers, mais aussi pour les grandes et moyennes entreprises, qui ont depuis longtemps protégé leurs serveurs et leurs applications des attaques venues de l'extérieur, grâce à des réseaux internes cloisonnés;

- le client d'un service de Cloud Computing devient très dépendant de la qualité du réseau pour accéder à ce service. Aucun fournisseur de service Cloud ne peut garantir une disponibilité de 100 %. Par exemple, des défaillances sur les services Cloud sont référencées par l'International Working Group of Cloud Resiliency

- les entreprises perdent la maîtrise de l'implantation de leurs données. De ce fait, les interfaces interapplicatives (qui peuvent être volumineuses) deviennent beaucoup plus complexes à mettre en œuvre que sur une architecture hébergée en interne;

- les entreprises n'ont plus de garanties (autres que contractuelles) de l'utilisation qui est faite de leurs données, puisqu'elles les confient à des tiers;

- des questions juridiques sont posées par l'absence de localisation précise des données du Cloud Computing. Les lois en vigueur s'appliquent, mais pour quel serveur, quel centre de données et, surtout, quel pays?

- comme les logiciels installés localement, les services de Cloud Computing permettent de lancer des attaques (cassage de mot de passe, déni de service...). En 2009 par exemple, un cheval de Troie a utilisé illégalement un service du Cloud public d'Amazon pour infecter des ordinateurs;

- du fait que l'on ne peut pas toujours exporter les données d'un service Cloud, la réversibilité (ou les coûts de sortie associés) n'est pas toujours prise en compte dans le cadre du projet. Le client se trouve souvent « piégé » par son prestataire, et c'est seulement lorsqu'il y a des problèmes (changement des termes du contrat ou des conditions générales d'utilisation, augmentation du prix du service, besoin d'accéder à

ses données en local, etc.) qu'il se rend compte de l'enfermement propriétaire (vendor lock-in) dans lequel il se trouve.

L'informatique en nuage induit une consommation d'énergie croissante et contribue au réchauffement climatique et des milieux par leur énergie grise<sup>21</sup> et via les fermes de serveurs nécessaires à l'informatique en nuage. L'ONG Greenpeace<sup>22</sup>, par exemple dénonce ceci dans un rapport de 2010 sur l'impact écologique du secteur informatique ; Ainsi, si toutes les fermes de serveurs d'informatique en nuage constituaient un pays, celui-ci viendrait au quatrième rang mondial des plus grands consommateurs d'énergie [33].

## 9. Conclusion

Comme on a pu voir, le Cloud Computing est l'une des notions les plus importantes dans le domaine informatique et technologique. Toutes les nouvelles technologies et inventions fonctionnent avec le Cloud, sur lequel sont basés les plus grandes entreprises.

Dans ce chapitre, nous avons abordé les notions les plus importantes du cloud. Le Cloud est caractérisé par son infrastructure distribué et sa gestion centralisé. Ses ressources sont loués dynamiquement et en fonction du besoin. Il a quatre modèles de déploiement, public, privé, hybride et communautaire, et il offre plusieurs services.

Le cloud consomme une quantité importante d'énergie, ce qui en fait un élément à prendre en considération et à bien comprendre. Dans le chapitre suivant nous allons aborder cette notion plus en détail.

## Chapitre 2

# La consommation d'énergie & les topologies

---

### Sommaire :

1.	Introduction .....	19
2.	Les Data Centers .....	19
3.	Définition de la consommation d'énergie .....	19
4.	Comment calculer la consommation d'énergie .....	19
5.	Raisons de la consommation élevée d'énergie des Data Center .....	20
6.	La virtualisation .....	20
6.1.	Techniques de virtualisation .....	21
6.2.	Les avantages de la virtualisation .....	22
6.3.	Les inconvénients de la virtualisation .....	23
7.	La migration .....	23
	Les avantages de la migration .....	25
<b>8.</b>	Les topologies .....	25
8.1.	Définition .....	25
8.2.	Les types .....	26
9.	Le SLA .....	33
10.	Conclusion .....	34

## 1. Introduction

L'augmentation de la consommation d'énergie dans le Cloud Computing soulève une grande préoccupation pour les gouvernements et les fournisseurs de services. Leur objectif principal est de fournir des services aux clients tout en augmentant leurs revenus et en consommant plus efficacement l'énergie consommée par leurs infrastructures. Outre les coûts d'exploitation écrasants et le coût total d'acquisition causés par la consommation d'énergie élevée, une autre préoccupation croissante est l'impact sur l'environnement en termes de dioxyde de carbone (CO<sub>2</sub>).

Ce chapitre est consacré à la consommation d'énergie. Nous allons voir comment ça se passe, comment pouvoir la réduire en utilisant des techniques et des technologies, en jetant un coup d'œil sur leurs types, avantages et leurs inconvénients, nous parlerons aussi des différents types de topologies.

## 2. Les Data Centers

Un Data Center ou le centre de données en français, est un regroupement de plusieurs équipements (ordinateurs centraux, serveurs rangés dans les baies, les baies de stockage, équipements réseaux et de télécommunication, etc.). Il peut être utilisé par les entreprises pour organiser, traiter, stocker et entreposer de grandes quantités de données. Pour qu'il puisse fonctionner correctement, il doit avoir une infrastructure qui contient un système de distribution d'énergie, un commutateur électrique, des réserves d'énergie, des générateurs dédiés au backup<sup>23</sup>, un système de ventilation et de refroidissement et une puissante connexion à Internet [34]. Pour cela les Data Centers doivent être protégés et ils doivent prévenir les risques de coupures électriques ou d'intrusion, les risques d'incendie et l'accès de personnes malveillantes sur les serveurs.

Quelques exemples sur les Data Center : ceux de Microsoft, Google et Facebook.

Les Data Center sont généralement faites pour soutenir la demande croissante de calcul et de stockage de données d'entreprises et de l'industrie mondiale [35]. Les études faites en 2017 par le cabinet IDC avaient prévu une forte croissance du nombre des Data Centers pour arriver jusqu'à 8.6 millions deux ans plus tard avec une surface de plus de 186 millions de mètres carrés [36].

## 3. Définition de la consommation d'énergie

Quand on parle de la consommation d'énergie, on parle de la quantité d'énergie utilisée ou consommée par un appareil qui utilise une source énergétique, cette énergie est mesurée par le Watt [37].

## 4. Comment calculer la consommation d'énergie

Le calcul de la consommation d'énergie permet de faire des économies en choisissant les meilleurs appareils ou comme dans notre cas de recherche, on choisit la meilleure topologie. Donc il faut connaître la consommation en Kw et faire le calcul nécessaire. Pour cela, on a besoin de 3 éléments qui doivent être pris en considération qui sont :

- La puissance.
- La durée d'utilisation quotidienne en heure,
- La durée d'utilisation annuelle en jour

Il suffit de multiplier la puissance par le nombre d'heures d'utilisation, puis par le nombre de jours ; si la puissance était en W, alors le résultat sera en W aussi, et là, on doit le divisé par 1000 pour l'obtenir en KW [37].

## 5. Raisons de la consommation élevée d'énergie des Data Center

La partie principale qui consomme le plus d'énergie dans les Data Centers du Cloud Computing se trouve au niveau des infrastructures matérielles (les serveurs, le stockage et les périphériques réseau). Ces périphériques atteignent une consommation important même s'ils sont en repos, du coup le gaspillage d'énergie est énorme. Ils fonctionnent tous les jours sans arrêt pour qu'ils puissent répondre à n'importe quelle heure et ne pas perdre les données des clients [35] [37].

La consommation des Data Centers ne cesse de croître, au point de représenter 4 % de la consommation énergétique mondiale en 2015. La climatisation et les systèmes de refroidissement représentent de 40 à 50 % de la consommation énergétique des Data Centers. Les Data Centers américains ont consommé 91 milliards de kWh en 2013 et 56 milliards en Europe (prévision : 104 milliards en 2020) [38].

Dans ce qui suit, nous allons aborder des techniques importantes dans la réduction de l'énergie dans le Cloud.

## 6. La virtualisation

La machine virtuelle a été introduite par IBM vers 1960. La technique de la virtualisation représente la notion principale du Cloud Computing, le plus important c'est le fait de pouvoir installer sur la même machine physique plusieurs systèmes d'exploitation sur différentes machines virtuelles, ce qui nous permet de utiliser peut de matériels, donc peut de consommation d'énergie.

Le but de la virtualisation c'est de cacher les caractéristiques physiques des ressources informatiques afin que les autres systèmes, les applications ou les utilisateurs puissent interagir avec ces ressources. Un système d'exploitation principal appelé « système hôte » est installé sur un serveur physique unique, il sert à accueillir d'autres systèmes d'exploitation. Un logiciel de virtualisation appelé « hyperviseur » ou bien « VMM » se trouve entre les deux couches.

Ce logiciel permet de créer les environnements clos et indépendants qui sont les VM, où les autres systèmes d'exploitations seront installés « systèmes invités ». Chaque VM à un accès aux ressources du serveur physique (CPU, mémoire, espace disque, ...etc.). Le VMM prend le contrôle du multiplexage des ressources, il est utilisé aussi pour la gestion d'énergie du système. [35] [37].

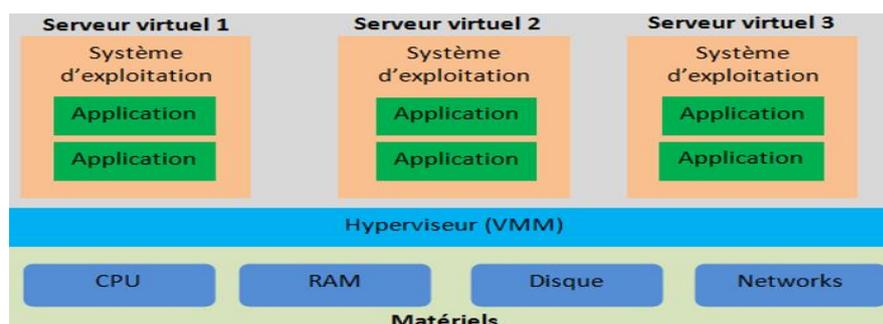


Fig. 2.6.1 Schéma de la virtualisation [35]

## 6.1. Techniques de virtualisation

Il existe plusieurs manières de réaliser la virtualisation telles que l'émulation, l'hyperviseur, la virtualisation complète et la paravirtualisation.

### - Émulation :

C'est la technique la plus ancienne de la virtualisation, elle s'appelle aussi la virtualisation binaire. Dans cette technique, l'hyperviseur présente un matériel virtuel complet (CPU, interface réseau, carte vidéo,...) au système invité qui peut être différent du système hôte.

L'émulation permet d'isoler les systèmes d'exploitation invités, elle fournit une grande flexibilité à ces systèmes, mais la vitesse du processus est très faible par rapport à l'hyperviseur et elle doit avoir une configuration élevée de ressources matérielles pour l'exécution de logiciel.

Exemple sur les émulateurs qui sont connus : Virtualbox, Lismoresystems Guest PC (émulateur de plateforme x86) [35] [37].



Fig. 2.6.2 Schéma d'émulation [35]

### - La virtualisation complète :

Dans cette technique, l'hyperviseur crée un environnement isolé entre le système invité et le système hôte. Les systèmes d'exploitation peuvent accéder directement aux contrôleurs de matériel et de ses périphériques sans être informé de l'environnement virtualisé et les modifications requises. Ce type de virtualisation intercepte et émule les instructions privilégiées et les instructions sensibles au moment de l'exécution.

L'inconvénient principal de cette méthode tient aux performances médiocres de la conversion et il est difficile d'accélérer cette dernière [35] [37].

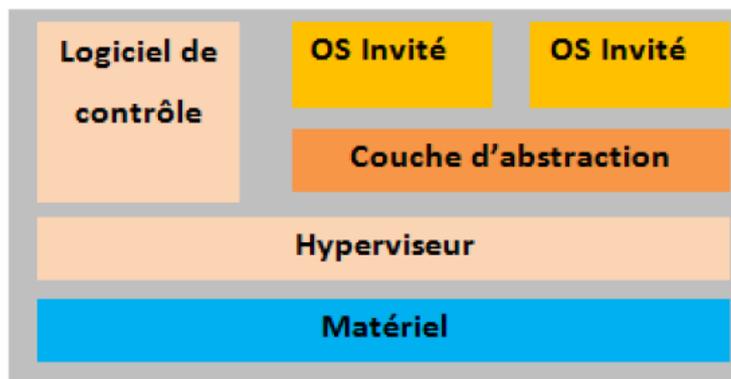


Fig. 2.6.3 Virtualisation complète [35]

### - Paravirtualisation :

C'est la technique inventé par le projet Xen<sup>24</sup>. Elle a été rendu possible grâce aux logiciels libres, parce qu'il était possible de modifier et d'adapter le noyau Linux ou BSD ou Solaris<sup>25</sup>.

Dans ce type de virtualisation, il y a un compilateur intelligent qui permet d'assister les OS invités, et qui remplace les instructions non virtualisables par des hyperappels<sup>26</sup>. Elle concerne la communication entre l'hyperviseur et le SE invité pour améliorer l'efficacité et la performance [35] [37].



Fig. 2.4 Paravirtualisation [35]

### - Hyperviseur :

C'est un programme qui permet aux nombreux systèmes d'exploitation de fonctionner simultanément sur une seule machine hôte. Le VMM crée des VMs isolées où chacune a un SE, ce dernier gère les demandes de VM pour accéder au matériel alors, il crée pour chacune d'eux un environnement similaire à l'environnement réel.

On peut trouver plusieurs exemples sur les hyperviseurs, on prend comme exemple : Xen qui est technologie open source, y a aussi VMware qui supporte la gestion d'énergie au niveau machine physique [35].

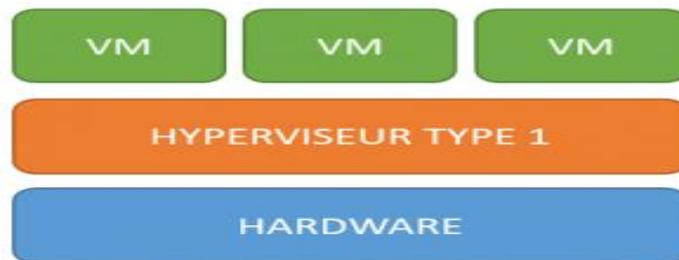


Fig. 2.5 L'architecture d'hyperviseur [35]

## 6.2. Les avantages de la virtualisation

La virtualisation a plusieurs avantages, on va citer quelques un qui sont les plus importants :

- Moins de serveurs : c'est le premier point qu'on peut citer. Comme on a pu voir, plusieurs machines virtuelles peuvent fonctionner sur un seul serveur physique ce qui permet de minimiser les occurrences de ce dernier.

- Consommer moins d'énergie : ce point est relié avec le premier point, vu qu'on peut diminuer le nombre de serveurs, donc logiquement la consommation d'énergie est réduite, car un serveur est fait dans le but de rester longtemps en place. Si on calcule le cout de dépenses en électricité d'un serveur tout au long de sa vie, on se rend compte que ça sera une somme non négligeable.

- La portabilité : un serveur virtuel n'est qu'une suite de fichiers, du coup il est possible de les déplacer sur n'importe quel type de matériel sans risque d'erreurs au démarrage, il peut même être copié sur un espace de stockage mobile tel que le disque dur pour le réinstaller sur une autre machine.

- la restauration d'une machine virtuelle : comme il est possible de copier et de déplacer le serveur virtuel, il est possible aussi de le restaurer facilement, sans avoir besoin de réinstaller les logiciels.

- Environnement de test : le serveur virtuel peut être utilisé comme un environnement de test, car il permet de faire des copies à un instant T, ce qui permet de revenir à une ancienne sauvegarde en cas d'erreur, ce qui permet de faire des modifications risquées sur son travail, en cas de problème il peut revenir en arrière.

- Licensing : Microsoft propose une offre pour Windows Server pour les environnements virtuels. Alors lorsqu'on achète une licence « Entreprise », on a le droit d'installer jusqu'à 4 serveurs virtuels. Si on achète une licence « Datacenter », le nombre de systèmes Windows virtuel devient illimité.

- Une meilleure disponibilité : la possibilité de déplacer les machines virtuelles d'un serveur physique à un autre sans avoir besoin de les stopper. On peut le faire en cas d'une panne d'un serveur, du coup le second serveur récupère automatiquement les VMs.

- Un monitoring simplifié : Il y a plusieurs solutions pour monitorer des serveurs. Quand on utilise la virtualisation, un serveur physique unique est à monitorer [39].

### 6.3. Les inconvénients de la virtualisation

Malgré tous ses avantages, il y a des inconvénients qui sont :

- Le coût important : pour avoir une architecture virtualisée qui marche très bien, il faut avoir un serveur physique qui a plusieurs processeurs et mémoires, donc ça coûte cher.

- Les pannes généralisées : en cas où le serveur tombe en panne, les VMs seront en panne aussi

- La vulnérabilité généralisée : si l'hyperviseur s'expose à une attaque vulnérable, les VMs seront attaquées aussi, et elles ne sont plus protégées. Plus les couches logicielles augmentent, plus les possibilités d'attaques augmentent aussi [35].

## 7. La migration

Cette méthode est une caractéristique importante de la virtualisation [35], elle a pour but de déplacer les machines virtuelles d'une ressource de calcul ou d'un emplacement de stockage vers un autre ou plutôt d'une machine physique à une autre [32]. La migration permet donc de libérer les machines pour ensuite les éteindre, et que les autres machines qui fonctionnent seront utilisées à leur capacité maximale [37]. Quand on dit la migration des VMs, on dit aussi le transfert de la mémoire, le CPU, les entrées/sorties vers la machine destination [35]

La migration a deux types, à froid et à chaud, le choix du type se fait en fonction de l'état d'alimentation de la machine virtuelle que vous migrez [40].

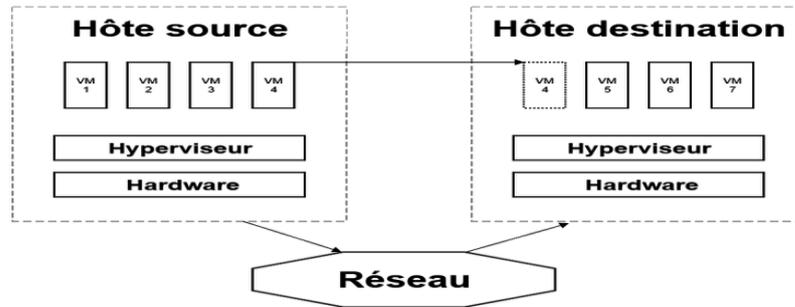


Fig. 2.6 La migration [41]

- **La migration à froid :**

En anglais c'est Stop and copy migration ou Non Live Migration, ici il faut mettre hors tension la machine virtuelle sur la machine source et migrer son exécution avec sa mémoire vers la machine destination. Une fois les pages mémoires transférées, la VM fonctionne dans la machine destination. Dans ce type de migration le temps total est égal au temps d'arrêt.

L'avantage de cette stratégie c'est qu'elle ne fait aucune faute lors de la migration de la mémoire, cette dernière ne sera pas altérée sur l'hôte source vue qu'elle sera arrêtée.

Dans cette figure, nous pouvons voir les étapes de la migration à froid, nous allons voir les étapes de migration de la VM3 de la machine 1 vers la machine 2.

1. L'exécution de la VM3 est arrêtée dans la machine source
2. L'état d'exécution et les pages mémoires de la VM3 sont transférés vers la machine 2.
3. La VM3 est activée dans la machine 2 [35].

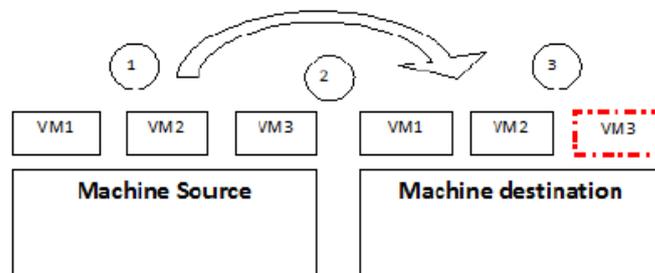


Fig. 2.7 Processus de la migration à froid [35]

- **La migration à chaud :**

Appelée aussi Live Migration en anglais, elle est plus utile, vue qu'elle permet la migration de SEs complets et même sans interruption de la VM, et aussi pallier les problèmes de la migration à froid. Cette technologie facilite l'équilibrage de charge, la gestion des pannes, la maintenance du système et la réduction de la consommation d'énergie.

Quelques scénarios où nous pouvons trouver la migration à chaud :

- Migrer une VM d'un serveur physique défaillant à un autre non défaillant.
- Migrer les VMs au mode repos sur une machine physique vers d'autres hôtes pour l'optimisation des ressources.

- Migrer les VMs sur un nœud physique chargé vers différents nœuds pour équilibrer les charges [35].

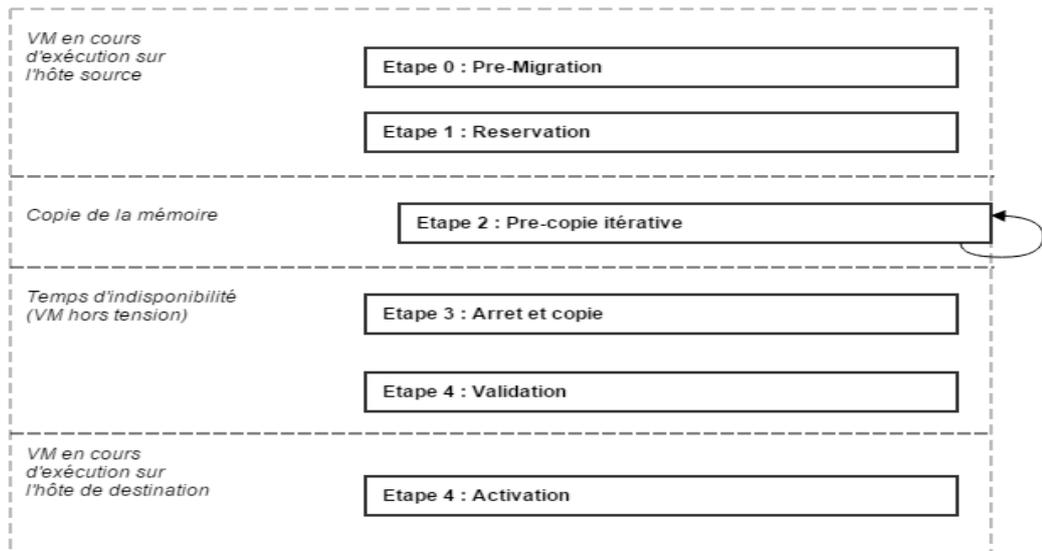


Fig. 2.8 migration à chaud [41]

### Les avantages de la migration

- **Tolérance aux pannes** : cet avantage permet aux machines virtuelles de continuer leurs tâches, même s'il y a une panne ou une partie du système échoue. Il se base sur la prédiction de la panne lorsqu'il migre une machine virtuelle d'un serveur physique à un autre, cela permet d'améliorer la disponibilité du serveur et d'éviter la dégradation des performances des applications.

- **Equilibrage de charge** : cela vise à répartir la charge du travail sur les serveurs physiques pour l'amélioration de l'évolutivité des serveurs physiques dans un environnement de Cloud. [35].

- **Réduction de la consommation d'énergie** : les Data Centers sont connues par leur consommation d'énergie lors du refroidissement qui peut atteindre jusqu'à 70% des ressources, même dans le cas d'une utilisation faible de ressources. Mais grâce à la migration, ce problème va disparaître, car cette technique conserve l'énergie consommée par les serveurs avec une utilisation optimale des ressources [37].

Puisque l'objet de notre étude est la réduction de l'énergie à travers la manipulation de la topologie dans le Data Center, dans la section suivante nous allons aborder en détail les différentes topologies utilisées.

## 8. Les topologies

### 8.1. Définition

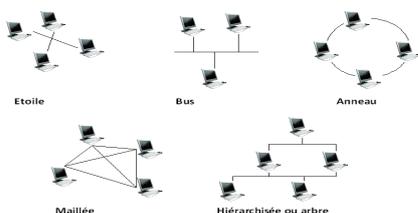


Fig. 2.9 La topologie [45]

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce à des lignes de communication (câbles réseaux, etc.) et des éléments matériels (cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique, c'est-à-dire la configuration spatiale du réseau est appelé topologie physique. La

topologie d'un réseau peut définir aussi la façon dont les données transitent dans les lignes de communication cela s'appelle topologie logique. On distingue généralement les topologies suivantes :

En bus, en étoile, en anneau, en arbre, maillé, mesh DCell, hybride, BCube, FiConn, DPillar, Rooted tree, VL2, fat-tree [43].

Il y a d'autres topologies qui sont utilisées que dans des réseaux conçus pour des tâches particulières, souvent scientifiques, ou pour effectuer des calculs distribués : le réseau en grille, le réseau en hyper-cube [44].

## 8.2. Les types

### **La topologie en bus :**

Ce type de topologie est le plus simple, comme son nom l'indique, bus, c'est le fait que, toutes les machines soient reliées à une même ligne de transmission par un câble, généralement le câble coaxial.

L'avantage de cette topologie c'est le fait d'être facile à mettre en œuvre et de posséder un fonctionnement simple. Mais elle a un inconvénient, c'est qu'elle est vulnérable car si l'une des connexions est défectueuse, l'ensemble du réseau en est affecté [45].



Fig. 2.10 Topologie en bus [46]

### **La topologie en étoile :**

Cette topologie utilise un concentrateur ou hub en anglais, où les machines sont reliées à ce dernier. Ce matériel est une boîte qui contient un certain nombre de jonctions, auxquelles les câbles réseau des machines sont raccordés, il assure la communication entre les jonctions.

Contrairement à la topologie en bus, elle est moins vulnérable car une connexion peut être débranchée sans affecter le réseau. L'inconvénient, c'est la présence du hub, car sans lui aucune communication entre les machines ne peut être possible. En plus de ça, la topologie en étoile est plus coûteuse qu'une topologie en bus à cause du hub [45].

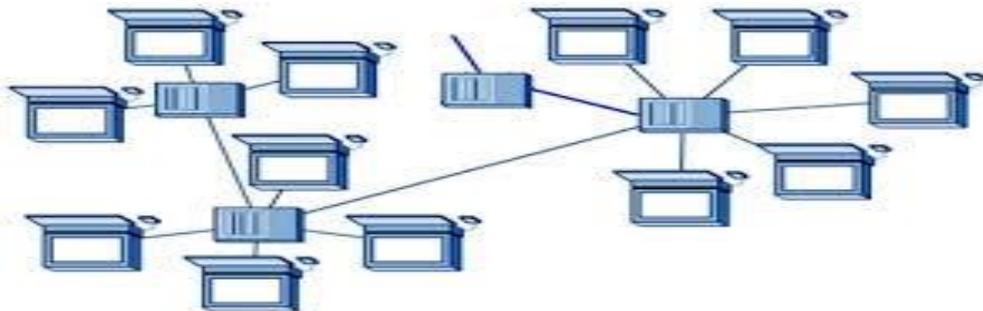


Fig.2.11 Topologie en étoile [47]

## La topologie en anneau :

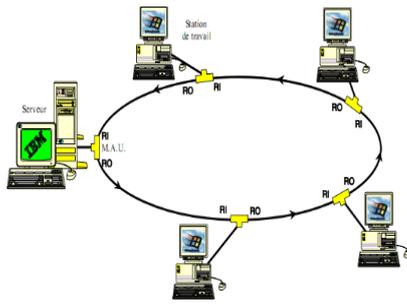


Fig. 2.12 Topologie anneau [48]

Ici les machines se trouvent en boucle et chacune communique à son tour, c'est d'après ce qu'on voit, car y a pas d'extrémité comme un réseau linéaire, mais en vérité, elles sont relié à un répartiteur qui est appelé MAU. Il est fait pour gérer la communication entre les hôtes, en donnant un temps de parole à chacun d'entre eux. Ce qui est bien dans cette architecture c'est que la défaillance d'un hôte n'affecte pas tout le réseau. Token ring<sup>27</sup> et FDDI qui sont des topologies logiques utilisent cette topologie physique [45].

## La topologie en arbre :

Appelé aussi la topologie hiérarchique. Elle relie les éléments d'une manière pyramidale qui est faite de trois niveaux [49] qui sont :

**Le noyau :** qui est le niveau supérieur de la topologie en arbre, il contient des commutateurs connectés aux commutateurs d'agrégation qui sont d'un niveau plus bas. Il représente le point d'accès du Cloud, où les requêtes envoyées passent premièrement par lui.

**Niveau d'agrégation :** ce niveau contient les commutateurs connectés aux commutateurs d'accès et aux commutateurs du noyau. Son rôle principal est de faire le chemin pour les requêtes des utilisateurs du Cloud aux commutateurs d'accès.

**Niveau d'accès :** il fait la connectivité pour pouvoir accéder aux ressources des serveurs disponibles. Sa conception est influencée par des critères de décision, la virtualisation des serveurs par exemple. Ce niveau utilise plusieurs genres de commutateurs, parmi eux il y a : EoR et ToR [50].

Dans cette topologie en arbre, on trouve deux types : le tree basic et le fat tree qui est plus vaste [50].

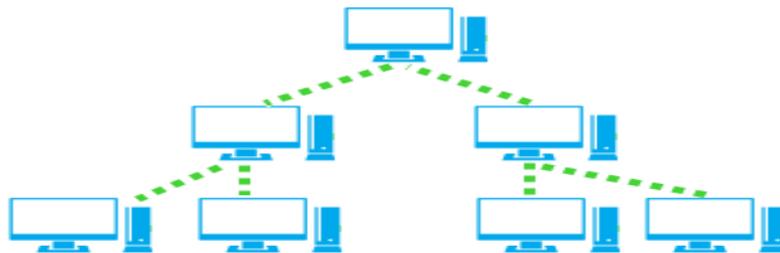


Fig. 2.13 Topologie en arbre [51]

## La topologie Fat-tree :

**Fat-tree** qui un autre type de la topologie en arbre et un cas particulier du réseau clos, il est utilisé surtout pour un réseau d'un grand nombre de serveurs physiques dans un grand Data Center. Selon son architecture, il y a trois niveaux, chaque niveau utilise un type de commutateur. Le premier niveau qui est le principal est pour les commutateurs Core, *Core Switches*, le deuxième utilise les commutateurs d'agrégation, *Aggregation Switches* et le dernier c'est pour les commutateurs périphériques, *Edge switches*.

Le nombre de port est le même dans chaque commutateur qui est symbolisé par « k », à partir de la valeur de k, on peut déduire le nombre de commutateurs Core, de

commutateurs d'agrégation, de commutateurs Edge et le nombre maximale de serveurs qui peuvent se connecter.

Le pod : est un ensemble de  $(k/2)$  nombre de commutateurs d'agrégation et de  $(k/2)$  nombre de commutateurs Edge qui sont combinés entre eux. Le nombre de pods qu'on peut avoir c'est « k » qui est le nombre de ports

La construction d'un Fat-tree :

- Avec k port pour chaque commutateur, le nombre de commutateurs Core est  $(k/2)^2$ , chaque module est constitué de  $(k/2)$  commutateurs d'agrégation et  $(k/2)$  commutateurs Edge.
- Chaque commutateur d'agrégation d'un pod est connecté à  $(k/2)$  commutateur Core et à  $(k/2)$  commutateur Edge
- Chaque commutateur Edge d'un pod est connecté à  $(k/2)$  serveurs et  $(k/2)$  commutateurs d'agrégation du même pod [52].

Ce qui est bien dans la topologie Fat-tree c'est qu'elle permet la communication des serveurs en utilisant la capacité totale de leurs interfaces réseau, en plus de ça, tous les modules de réseau sont identiques, du coup ça évite les commutateurs coûteux à haute densité de ports [53].

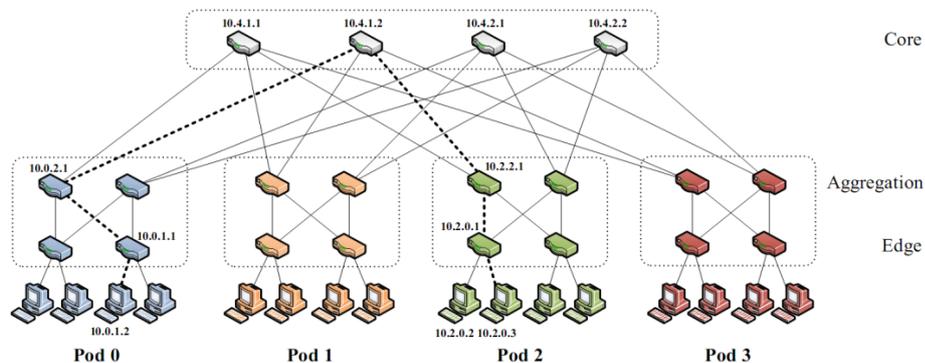


Fig. 2.14 La topologie Fat-tree [54]

### La topologie maillée :

C'est une topologie étoile mais plus évoluée, elle est utilisée pour plusieurs liaisons point à point. Dans ce type chaque terminal est relié à tous les autres terminaux. Son inconvénient est le nombre de liaisons qui peut devenir très grand, car chaque unité peut avoir  $(1, N)$  connexions vers plusieurs unités.

On trouve ce genre de structure dans les grands réseaux de distribution comme Internet, ou même aussi chez l'armée, parce que même dans le cas d'une rupture de lien, l'information peut être acheminée. Elle se trouve aussi dans le cas de couverture Wi-Fi [45].

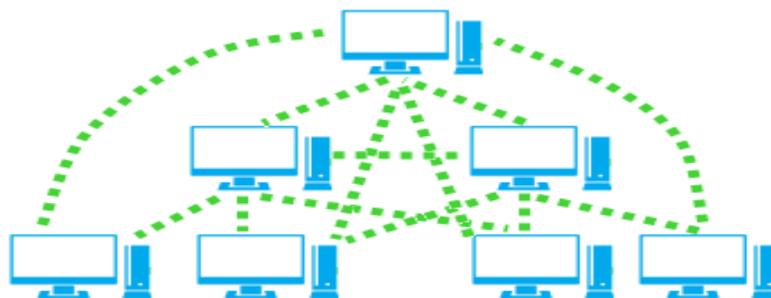


Fig. 2.15 Topologie maillée [51]

### La topologie hybride :

Une topologie hybride est structure qui combine une topologie avec une autre, elle existe en deux types, un qui est une topologie hybride physique qui contient deux topologies physique ou plus dans un réseau, par exemple, une partie peut être un anneau physique et la deuxième une étoile physique, ou bien l'exemple qu'on peut voir le plus, c'est d'avoir plusieurs commutateurs dans un réseau, et que chacun d'entre eux sera relié en tant que bus, mais chaque commutateur donnera une étoile physique à partir de celui-là, le résultat obtenu est une connexion de bus physique avec les commutateurs et des étoiles logiques dans chaque commutateur.

L'autre type est une topologie hybride physique-logique, c'est différent par rapport au premier type, car il ressemble physiquement à une topologie, mais il fonctionne comme une autre. La topologie la plus connue dans ce type est un réseau qui fonctionne comme un anneau, mais il ressemble à une étoile [56].

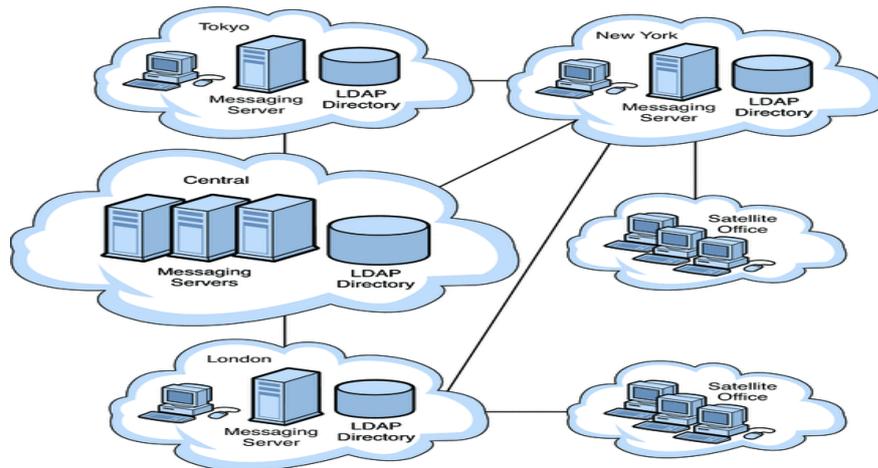


Fig. 2.16 Topologie hybride [57]

### La topologie BCube :

Elle a été proposée pour l'utiliser dans un centre de données modulaire (MDC). Elle est classée dans la catégorie des topologies récursives, elle comporte des couches de mini-commutateurs et des serveurs, qui servent à transférer les paquets

Ces serveurs ont donc plusieurs interfaces réseau, généralement pas plus de cinq. Le module principal d'une topologie BCube est BCube0, qui consiste en un commutateur unique avec  $n$  ports connectés à  $n$  serveurs. Un BCube1, en revanche, est construit en utilisant  $n$  réseaux BCube0 et  $n$  commutateurs. Chaque commutateur est connecté à tous les réseaux BCube0 via sa connexion avec un serveur de chaque système BCube0. La figure suivante montre un réseau BCube1. Plus généralement, un réseau BCubek ( $k \geq 1$ ) est constitué de  $n$  BCubek-1s et de  $nk$  commutateurs de  $n$  ports. Pour construire un BCubek, les  $n$  Bcubek-1 sont numérotés de 0 à  $n - 1$  et les serveurs de chacun de 0 à  $nk - 1$ . Ensuite, le port de niveau  $k$  du  $i$ -ème serveur ( $i \in [0, nk - 1]$ ) du  $j$ -ième BCubek ( $j [0, n - 1]$ ) est connecté au  $j$ -ième port du commutateur de niveau  $k$ . Un réseau BCubek peut avoir  $nk + 1$  serveurs. La figure suivante montre que, dans BCube00, le serveur 0 communique via un commutateur vers le serveur 1. D'autre part, le serveur 1 de BCube01 utilise son commutateur local pour transférer ses paquets au serveur 2, qui peut transférer le paquet à la destination, dans ce cas le serveur 2 dans le réseau BCube02. Toutefois, la communication entre différents BCubes du même niveau peut se faire en utilisant uniquement un commutateur de niveau supérieur, comme dans le cas du serveur 3 de BCube02 avec le serveur 3 dans BCube03. En résumé, les serveurs BCube peuvent participer au transfert de paquets en fonction de la paire en communication [53].

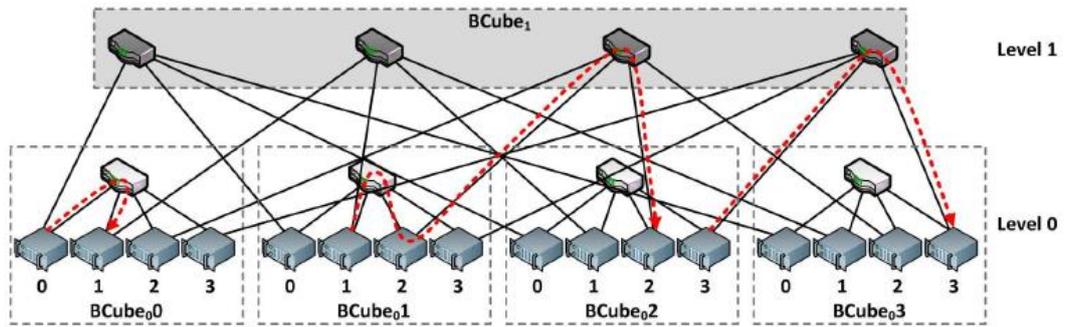


Fig. 2.17 L'architecture de la topologie BCube [53]

### **La topologie DCell :**

De la même manière que BCube, DCell est défini de manière récursive et utilise des serveurs et des mini-commutateurs pour le transfert de paquets. Le module principal de cette topologie est DCell0 qui, en tant que BCube0, est composé d'un commutateur connecté à  $n$  serveurs. Un DCell1 est construit en connectant  $n + 1$  réseaux DCell0 et un DCell0 est connecté à toutes les autres cellules DCell0 par un lien d'un de ses serveurs à un serveur d'un autre DCell0. Un réseau DCell1 est illustré à la figure suivante.

Notez que la communication à l'intérieur d'une cellule est effectuée localement à l'aide d'un commutateur, comme indiqué dans la communication entre le serveur 2 et le serveur 3 à partir de DCell00.

La communication entre les serveurs de cellules différentes est effectuée directement, par exemple entre le serveur 1 dans DCell02 et le serveur 2 dans DCell03, ou en utilisant une combinaison de serveurs et de commutateurs. Ce dernier est indiqué dans la communication entre le serveur 1 de DCell01 et le serveur 1 de DCell04.

Notez que dans un DCell, à la différence du BCube, les commutateurs sont connectés uniquement aux serveurs de son même DCell et la connexion entre différents réseaux DCell se fait toujours à l'aide de serveurs. Pour construire un DCell $k$ , il faut  $n + 1$  réseaux DCell $k - 1$ . Chaque serveur dans un DCell $k$  a  $k + 1$  liens. Sur chaque serveur, le premier lien (lien de niveau 0) est connecté au commutateur de son DCell0 et le deuxième lien connecte le serveur à un nœud de son DCell1, mais dans un autre DCell0. Généralement, la liaison de niveau  $i$  d'un serveur le connecte à un autre DCell $i-1$  situé dans le même DCell $i$ . La procédure de construction d'une DCell est plus complexe que celle d'un BCube et est exécutée par un algorithme proposé par Guo et al.

La capacité DCell en nombre de serveurs peut être évaluée de manière récursive, en utilisant les équations suivantes:  $g_k = t_{k-1} + 1$  et  $t_k = g_k \times t_{k-1}$ , où  $g_k$  est le nombre de réseaux DCell $k - 1$  dans un DCell $k$ ,  $k$  est le nombre de serveurs dans un DCell $k$ . Un réseau DCell0 est un cas particulier dans lequel  $g_0 = 1$  et  $t_0 = n$  [53].

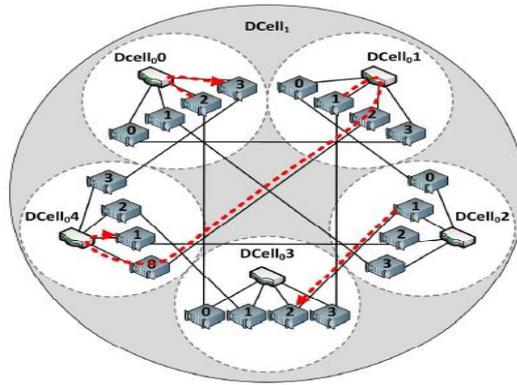


Fig. 2.18 Topologie DCell [53]

### La topologie FiConn :

Dans FiConn, le premier DCN centré sur le serveur à double port à être proposé et, par conséquent, généralement considéré comme la base de ce type de DCN. Pour tout  $n \geq 2$  et  $k \geq 0$ ,  $\text{FiConn}_k, n$  est un DCN défini de manière récursive où  $k$  indique le niveau de la construction récursive et  $n$  le nombre de nœuds de serveur directement connectés à un nœud de commutateur (donc, tous les nœuds de commutation ont le degré  $n$ ).  $\text{FiConn}_0, n$  est constitué de  $n$  nœuds de serveur et d'un nœud de commutateur, auxquels tous les nœuds de serveur sont connectés. Supposons que  $\text{FiConn}_k, n$  ait  $b$  nœuds de serveur de degré 1 ( $b = n$  lorsque  $k = 0$ ; de plus, quelle que soit la valeur de  $k$ ,  $b$  peut toujours être montrée comme étant égale). Afin de construire  $\text{FiConn}_{k+1}, n$ , nous prenons  $b + 1$  copies de  $\text{FiConn}_k, n$  et pour chaque copie, nous connectons un nœud de serveur de degré 1 à chacune des autres  $b$  copies (ces liens supplémentaires sont appelés liens de niveau  $k$ ). Notez que tous les DCN que nous considérons ici viennent en familles paramétrées; Ainsi, lorsque nous disons «le DCN FiConn», nous entendons vraiment «la famille des DCN  $\{\text{FiConn}_k, n; k \geq 0, n \geq 2 \text{ est paire}\}$ ». [58].

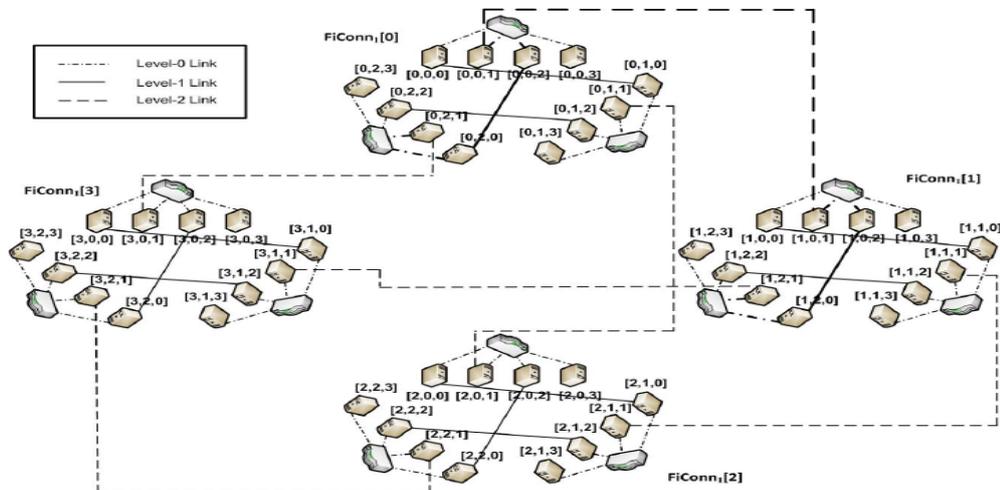


Fig. 2.19 Topologie FiConn [59]

### La topologie DPillar :

C'est une architecture d'interconnexion des Data Centers très évoluées, elle est différente par rapport aux autres, elle travaille qu'avec des serveurs et des commutateurs PC standard bas de gamme. Les switches sont des périphériques Plug-and-Play<sup>28</sup> de couche 2 économiques et les serveurs sont des ordinateurs de bureau à deux ports. Cette topologie utilise un nombre quelconque de serveurs sans qu'il soit nécessaire de mettre à niveau physiquement les serveurs existants, c'est son point principal [60].

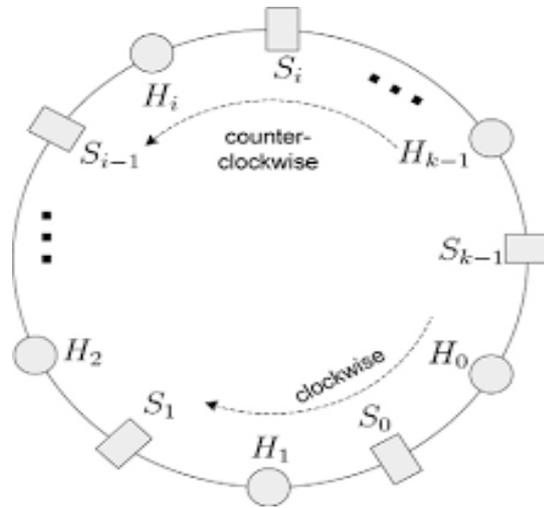


Fig. 2.20 Une vue verticale de la topologie DPillar [61]

Cette figure représente un exemple sur un réseau DPillar qui a 8 switches de 8 ports chacun. Il y a deux colonnes  $S_0$  et  $S_1$ , chacune contient 4 switches. Il y a aussi deux colonnes d'hôtes  $H_0$  et  $H_1$ , chacune a 16 hôtes. Nous allons diviser sa surface qui est cylindrique le long de la colonne  $H_0$  pour avoir une troisième colonne  $H_0$  qui est dupliquée. Chaque switch a deux colonnes d'hôtes ( $n=8, k=2$ ) [61].

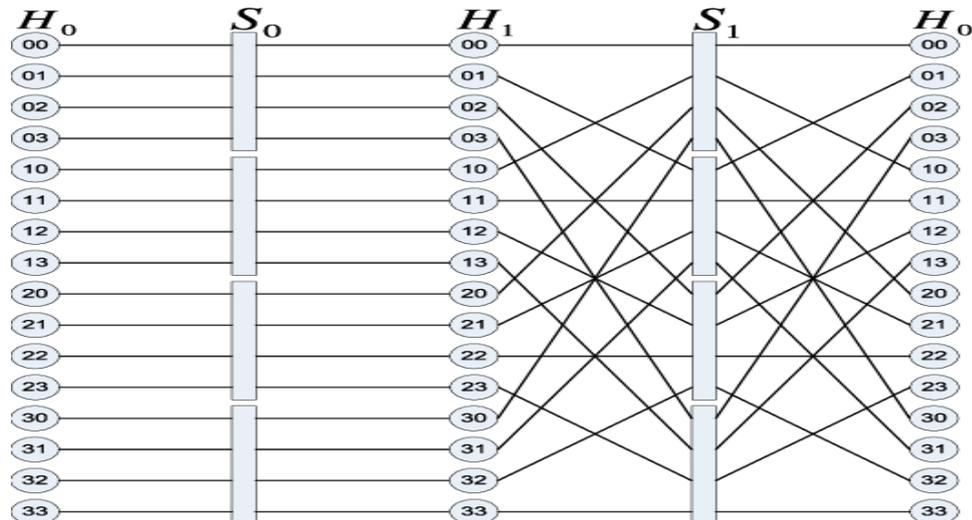


Fig. 2.21 Topologie DPillar [61]

### La topologie VL2 :

VL2 utilise également un réseau Clos<sup>29</sup>, il veut dire Virtual Layers ou les couches virtuelles en français, selon son nom, on comprend qu'il y a des couches, et dans chaque couche il y a des matériaux. Comme on peut voir dans la Figure suivante, la deuxième couche est faite pour les serveurs et les switches, la troisième est pour les routeurs et la dernière couche est pour Internet. Cette topologie est très utile, elle est faite pour prendre en charge d'énormes Data Centers avec une grande capacité qui est la même pour tous les serveurs, une isolation des performances entre les services et une sémantique Ethernet de couche 2 [62].

Dans la couche 2 : constitué de grilles de serveurs connectés à des switches qui sont connectés à d'autres switches. Ces derniers sont connectés au Load Balancer ou l'équilibreur de charge.

Dans la couche 3 : les derniers switches qui sont connectés avec les Load Balancer, ce connectent avec les routeurs d'accès, ces derniers se connectent avec les routeurs frontière qui partent à Internet qui est la dernière couche.

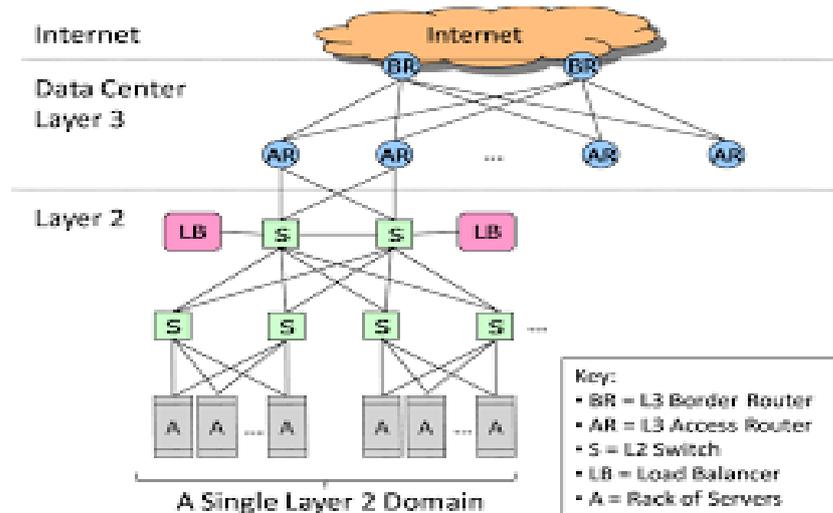


Fig. 2.22 La structure de la topologie VL2 [63]

## 9. Le SLA

Le Service Level Agreement ou plus connu en tant que SLA, en français, signifie engagements de service ou accord de niveau de service ou encore convention de services [64]. C'est un contrat relié à l'univers du Cloud, il concerne le niveau de service que le prestataire informatique s'engage à délivrer à un ou plusieurs utilisateurs. Il permet de garantir aux clients certains niveaux de sécurité dans le stockage et la gestion de leurs données personnelles, il formalise aussi leurs besoins de façon à les rendre compréhensibles et de fournir des critères d'évaluation de la question [65].

La rédaction d'un SLA peut changer selon le prestataire, du coup même les critères peuvent être différents. La rédaction peut être de type réseau, applications ou hébergement. Généralement, on trouve les points suivants :

- la liste des services, leur description, leur étendue
- la définition des niveaux de service et leurs critères de mesure, il y a aussi le temps de réponse du prestataire en cas de panne ou d'incidents ou encore, le taux de disponibilité du serveur
- le prix et les conditions de paiement
- la sécurité et la récupération des données
- les dates de début et de fin du contrat
- les conditions de renouvellement et de résiliation
- l'obligation d'information et de conseil dû par le prestataire
- les relations avec les tiers
- les litiges, les plaintes, la propriété intellectuelle et le droit applicable

Mais il y a aussi des points qu'on doit éviter dans la rédaction du SLA :

- à définir précisément ce que recouvre la notion de disponibilité
- à prévoir non seulement les solutions en cas de problème, mais aussi les résultats atteindre en temps normal de fonctionnement

- à rédiger le contrat dans des termes intelligibles par le client (le SLA est souvent considéré comme un document hautement technique)

## 10. Conclusion

Les Data Centers rendent un service très important à la communauté mondiale. Ils supportent n'importe quel type de formation. Ils sont connus par leur consommation d'énergie vue précédemment.

Le fait de vouloir réduire la consommation d'énergie nous permet aussi de réduire les coûts et protéger l'environnement parce que ça implique la réduction du CO2.

Dans ce chapitre, nous avons parlé de quelques techniques pour réduire la consommation d'énergie des Data Centers, comme nous avons vu aussi deux technologies avec leurs types, leurs avantages et leurs inconvénients. Nous avons terminé le chapitre avec différentes topologies du réseau, dont les plus importantes seront testées dans notre application à fin de prouver l'influence de la topologie sur la consommation d'énergie et déterminer quelle est la meilleure topologie dans ce cadre.

### Sommaire :

1. Introduction.....	36
2. Langage et environnement de développement.....	36
2.1. Le langage de programmation Java.....	36
2.2. L'environnement de développement.....	37
2.3. Le simulateur CloudSim .....	37
2.3.1. Définition .....	37
2.3.2. L'architecture du CloudSim.....	37
3. Réalisation de l'application.....	38
3.1. La partie Présentation.....	38
3.2. La partie Fonctionnement .....	41
3.3. Persistance.....	43
4. Résultats obtenues.....	44
5. Comparaison .....	49
6. Conclusion .....	50

## 1. Introduction

Pour atteindre l'objectif de notre étude nous avons réalisé une application qui permet de prouver et confirmer nos déclarations théoriques sur l'influence de la topologie dans la consommation d'énergie du Cloud et découvrir qu'elle est la meilleure topologie.

Dans ce chapitre nous allons présenter le contexte ainsi que notre application qui nous permet de simuler les 4 topologies réseau DCell, BCube, Fat-tree et V12 avec 100 entités réseau chacune.

## 2. Langage et environnement de développement

Notre travail essentiel de ce mémoire qui l'application a été implémenter et tester dans un environnement qui a les caractères suivants :

- Une machine avec un processeur Intel(R) Core(TM) i5-2430 CPU, une vitesse de 2.40Ghz et une capacité mémoire de 6Go. Le SE utilisé est Windows 7 de 64bits.
- Le simulateur CloudSim version 3.0.3 développé avec Java.
- Le langage de programmation Java.
- L'IDE NetBeans.

Avant de voir les résultats de l'application, on va jeter un coup d'œil sur le simulateur, le langage et l'environnement de développement utilisés.

### 2.1. Le langage de programmation Java



C'est un langage de programmation informatique orienté objet, et en même temps un environnement d'exécution portable [66], développé par Sun Microsystems. Il est simple, robuste, dynamique et sécurisé, indépendant de la plateforme matérielle. Il permet de créer des logiciels compatibles avec de nombreux SEs, et développer des programmes pour téléphones portables et bien d'autres applications [67]. Java n'est pas comme le langage C qui est compilé et pas comme PHP qui est interprété, c'est un langage intermédiaire qui veut dire semi Compilé/Semi Interprété. Il a une bibliothèque très importante sans elle, il ne marche pas, c'est le JDK API. Il est devenu OpenSource depuis Novembre 2006, toutes les sources sont disponibles. Java appartient maintenant à Oracle vu que ce dernier a acheté la société Sun en 2009.

Il a trois environnements d'exécutions différents :

- Java ME pour PDA, téléphone
- Java SE (Standard Edition) pour le desktop
- Java EE (Entreprise Edition) pour serveur

Servlet/JSP/JSTL

Portlet/JSF

JTA/JTS, JDO/EJB

JavaMail, ... etc. [66]

## 2.2. L'environnement de développement



L'éditeur que nous avons utilisé est NetBeans, qui est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il se télécharge à partir de son site <http://netbeans.org/downloads>. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS [35].

## 2.3. Le simulateur CloudSim

### 2.3.1. Définition



CloudSim est une nouvelle structure de simulation généralisée et extensible qui permet la modélisation des environnements hétérogènes, la simulation et l'expérimentation de Cloud émergent des infrastructures de calcul et des services d'application. Il offre les fonctionnalités suivantes :

- Une plate-forme indépendante pour la modélisation des Data Centers, des Brokers, de l'ordonnancement et des politiques d'allocation des ressources ;
- Un support pour la modélisation et la simulation à grande échelle d'infrastructure de Cloud Computing, y compris des centres de données sur un seul nœud physique.

Parmi les principales caractéristiques du CloudSim, nous pouvons citer :

- La disponibilité d'un moteur de virtualisation qui facilite la création et la gestion indépendante des services ainsi que l'hébergement des services virtualisés sur un nœud d'un Data Center ;
- Un support pour la simulation des connexions réseau entre les éléments du système de simulation ;
- Une simulation de la définition de matériel de centre de traitement des données (Data Center) en termes de machines physiques composées de processeurs, de dispositif de stockage, de mémoire et de largeur de bande passante ;
- Une simulation des spécifications, de la création et de la destruction de machines virtuelles ;

Une simulation de l'exécution des programmes utilisateurs ou des tâches (Cloudlet) sur les machines virtuelles [35].

### 2.3.2. L'architecture du CloudSim

La structure logicielle de CloudSim et ses composants sont représentés par une architecture en couches comme indiqué dans la Figure 3.1. Les premières versions de CloudSim utilisent SimJava, un moteur de simulation d'événement discret qui met en œuvre les principales fonctionnalités requises pour des structures de simulation de haut niveau. Parmi les fonctionnalités, nous avons la formation d'une file d'attente et le traitement d'événements, la création de composants système (les services, les machines (Host), le centre de données (Data Center), le courtier (Broker), les machines virtuelles), la communication entre les composants et la gestion de l'horloge de simulation [35].

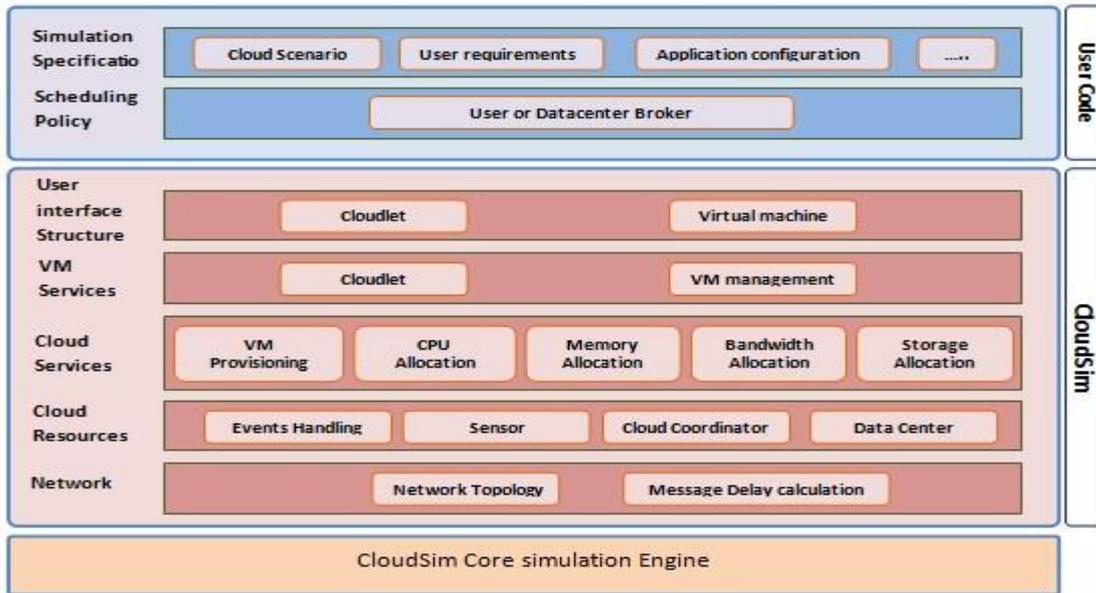


Fig. 3.1 Schéma qui représente l'architecture de CloudSim [62]

### 3. Réalisation de l'application

#### 3.1. La partie Présentation

##### - Les interfaces graphiques :

1) **PremierePage.java** : représente la page d'accueil.



Fig. 3.2 L'interface de la page d'accueil

2) **ConfDatacenter.java** : c'est la page de configuration du Data Center permet à l'utilisateur de préciser les caractéristiques du Data Center afin de les stocker dans la base de données.

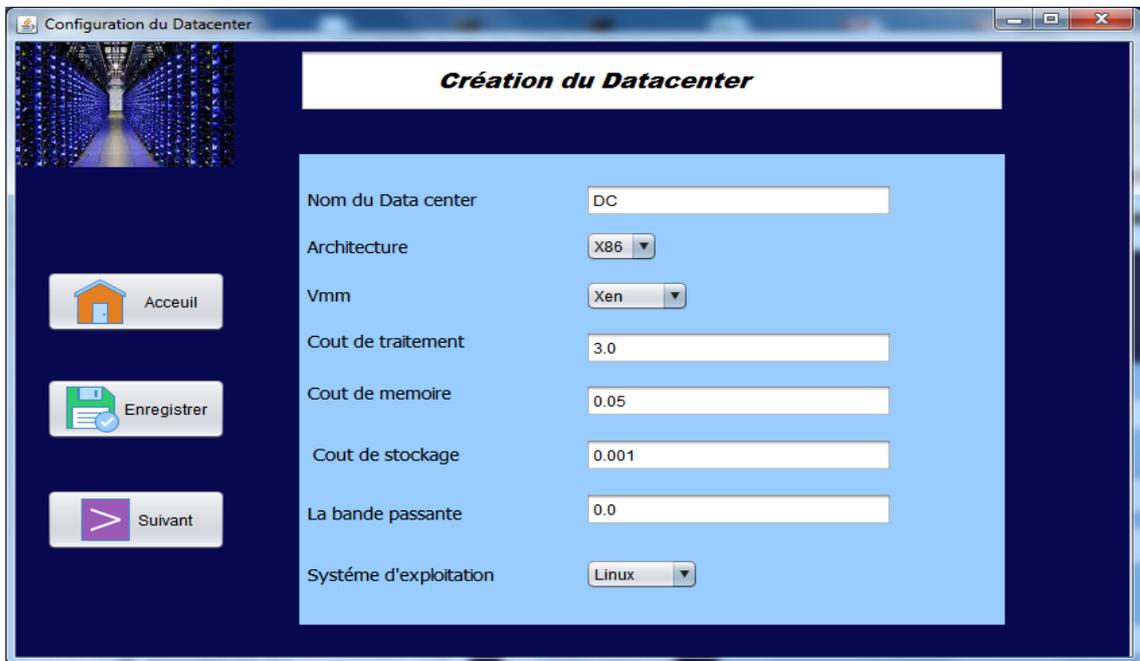


Fig. 3.3 L'interface de la configuration du Data Center

3) **Host.java**: sert à configurer les machines physiques et stocker les caractéristiques dans la BDD.



Fig. 3.4 L'interface de la configuration des Hôtes

4) **Vm.java**: c'est la page de configuration des machines virtuelles et stocke les caractéristiques dans la BDD.

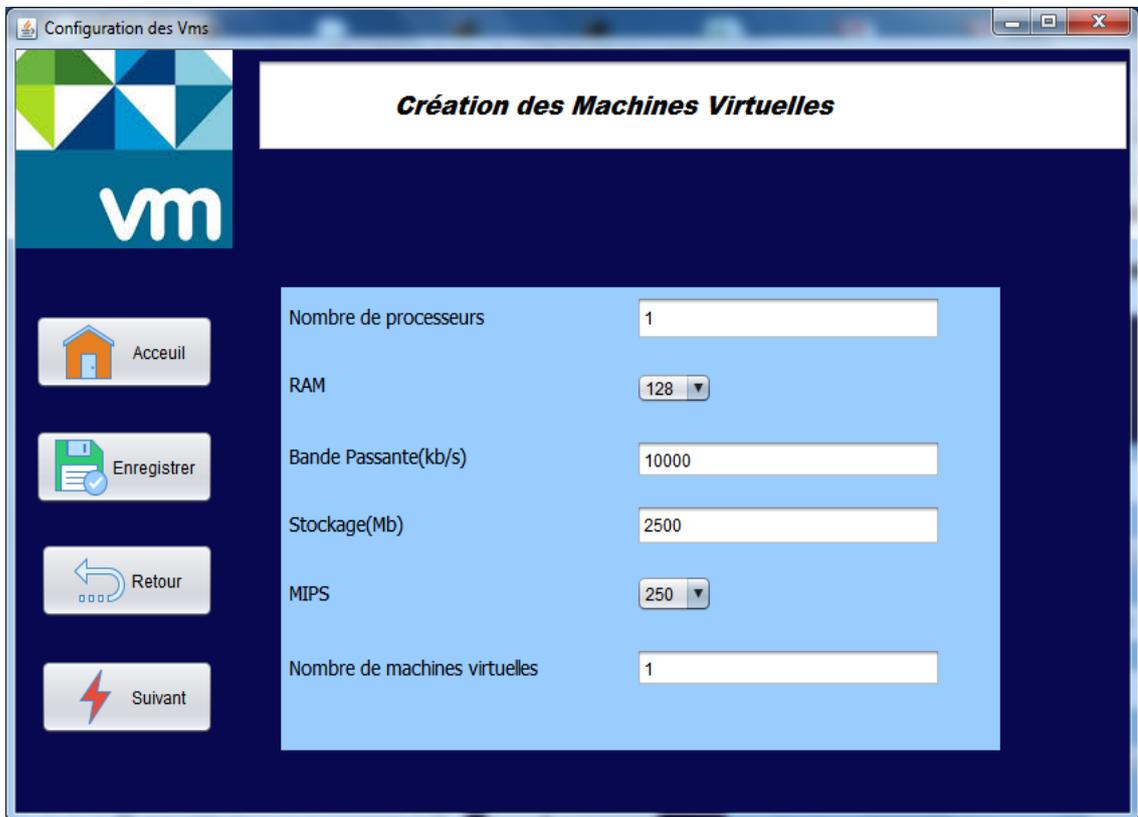


Fig. 3.5 La configuration des VMs

5) **Topologie.java**: c'est la page des topologies sert à montrer les graphes une fois choisir le type de graphe et cliquer sur l'image correspondante.

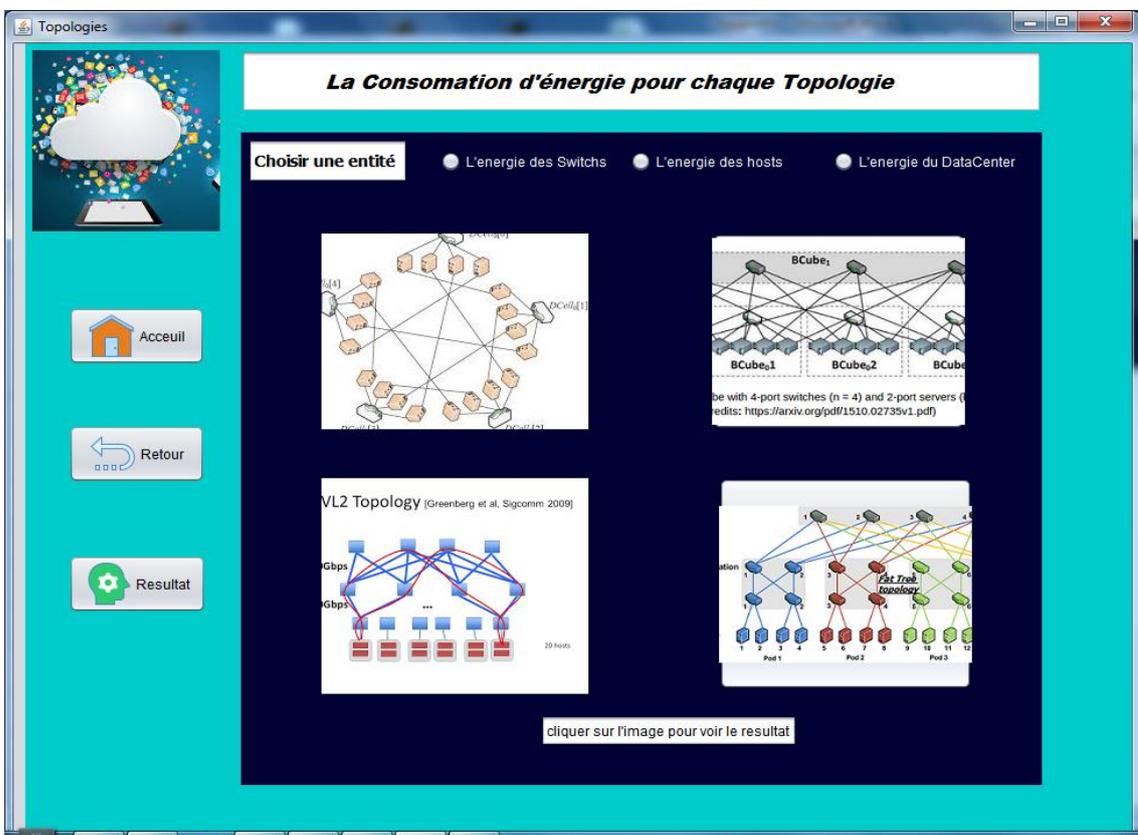


Fig. 3.6 l'interface des topologies utilisées

6) **Resultat.java**: sert à montrer les résultats de cette étude.



Fig. 3.7 l'interface pour afficher le résultat

### 3.2. La partie Fonctionnement

Dans notre projet, nous avons utilisé deux genres de classes, les classes propres à CloudSim, et d'autres que nous avons créées pour le besoin.

#### - Les classes du CloudSim :

**SimEntity:** il s'agit d'une classe abstraite, elle représente l'entité de simulation qui est capable d'envoyer des messages à d'autres entités et de gérer les messages reçus ainsi que les événements. Toutes les entités doivent étendre cette classe.

**DatacenterBroker:** cette classe modélise le Broker qui est responsable de la médiation entre les utilisateurs et les prestataires de service selon les conditions de la QoS des utilisateurs et déploie les tâches de service à travers le Cloud. Le Broker au nom des utilisateurs agit sur les prestataires du service approprié par le service d'information du Cloud CIS et négocie avec eux pour une allocation des ressources qui répond aux besoins de QoS des utilisateurs.

**VM:** cette classe représente une instance de machine virtuelle qui est gérée et hébergée par une machine physique.

**PowerHost:** cette classe étend la classe Host qui modélise une ressource physique comme le serveur de stockage ou de calcul. Elle a comme particularité la méthode `getPower()` qui retourne l'énergie consommée par le host.

**PowerDatacenterBroker:** cette classe hérite du Data CenterBroker et ses machines physiques sont de type PowerHost

**Cloudlet:** cette classe modélise les services d'application, chaque service a une taille d'instruction pré-assignée et la quantité de flux de transfert de données qu'il doit entreprendre au cours de son cycle de vie.

**Datacenter:** cette classe modélise les services au niveau matériel qui sont offerts par les fournisseurs de Cloud, elle encapsule un ensemble d'hôtes qui peuvent être homogènes ou hétérogènes par rapport à leurs configuration matérielles (mémoire, noyaux, capacité et stockage).

**Datacentercharacteristics:** cette classe contient les informations sur la configuration des ressources des centres de données.

**NetworkTopology:** NetworkTopology est une classe qui implémente la couche réseau dans CloudSim. Elle lit un fichier BRITE et génère la topologie du réseau à partir de celui-ci. Ses informations sont utilisées pour simuler la latence du trafic réseau dans CloudSim.

**Le fichier topology.brite:** peut contenir plus de nœuds qui représentent le nombre d'entités dans la simulation. Il permet aux utilisateurs d'augmenter l'échelle de la simulation.

**Pe:** (Processing Element) cette classe représente l'unité CPU, définie en termes de millions d'instructions par seconde (MIPS). Tous les PE sous la même machine ont la même cote MIPS.

**Storage:** Une interface qui définit la fonctionnalité souhaitée d'un système de stockage dans un nuage de données. Les classes qui implémentent cette interface doivent simuler les caractéristiques des différents systèmes de stockage en définissant la capacité du stockage et le taux de transfert maximum.

**BwProvisionerSimple :** c'est une classe qui met en œuvre une politique simple d'allocation des meilleurs efforts : s'il y a bw disponible à la demande, elle alloue ; autrement, elle échoue.

**RamProvisionerSimple:** est une extension de RamProvisioner qui utilise une politique de déploiement optimal pour allouer de la mémoire à une machine virtuelle.

- **Les classes créées:**

**MyDataCenter:** cette classe hérite à la fois de NetworkDatacenter et possède les caractéristiques du PowerDatacenter son rôle est de calculer l'énergie du Data Center et des hôtes.

**PowerSwitch:** c'est une classe qui hérite d'EdgeSwitch et qui a comme attribut l'objet PowerHost

**DCell :** cette classe crée le Data Center, les hôtes, les Vms et Le broker ainsi que les Cloudlets elle ouvre le fichier brite, configure le réseau pour la topologie DCell et lance la simulation.

**BCube :** cette classe crée toutes les entités nécessaires pour la simulation afin d'ouvrir fichier brite et configurer le réseau pour la topologie BCube.

**Fat tree:** cette classe crée le Data Center, les hôtes, les Vms et Le broker ainsi que les Cloudlets. Elle ouvre aussi le fichier brite, configure le réseau pour la topologie Fat-tree et lance la simulation.

**VL2:** cette classe crée toutes les entités nécessaires pour la simulation afin d'ouvrir fichier brite et configurer le réseau pour la topologie VL2.

**Graphe:** cette classe sert à dessiner le graphe approprié à la topologie.

**Connexion:** cette classe fait la connexion entre le serveur de BDD Wamp (PHP My admin) et le NetBeans

**RemplirBDD:** cette classe sert à remplir la BDD et récupérer les valeurs de cette dernière.

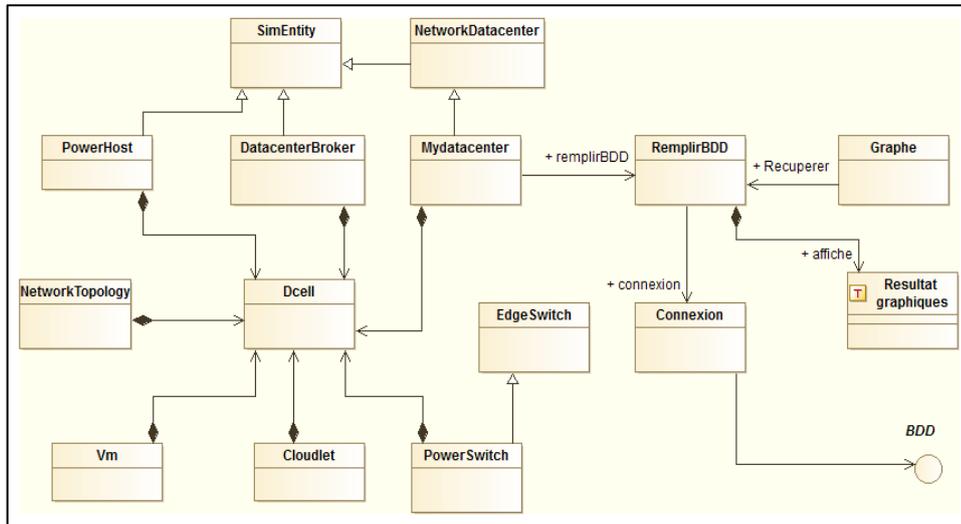


Fig. 3.8 Schéma d'interaction entre les classes (DCell).

### 3.3. Persistance

- **BDD MySQL du serveur wamp2.0 :**



**WampServer** est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

données.

La BDD cloud : notre BDD contient 15 tables

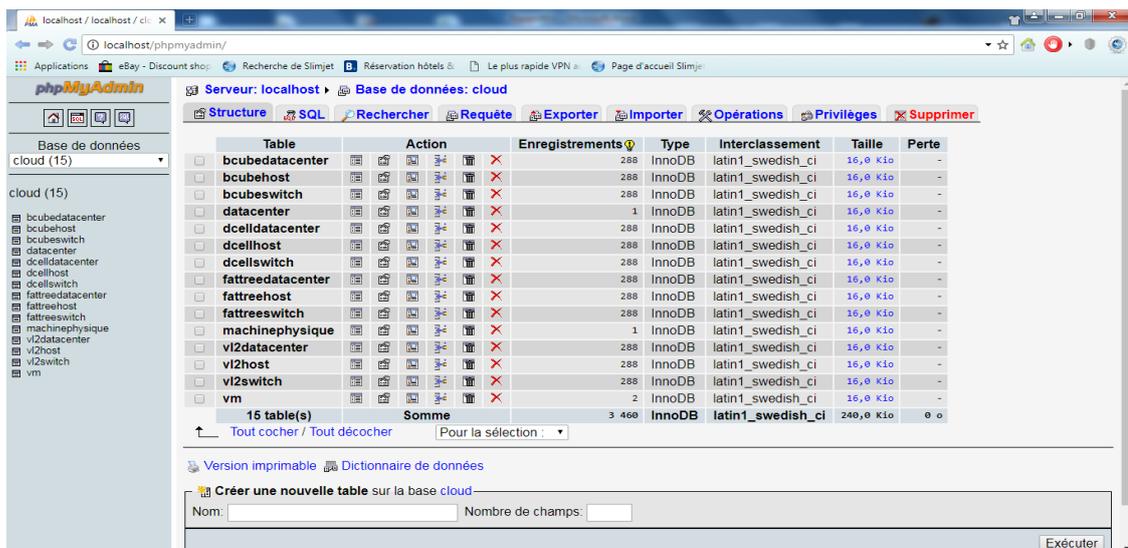


Fig. 3.9 La base de données de notre application

**Les fichiers brite:** on a créé 4 fichiers Brite pour les 4 topologies on précisant les 100 nœuds avec leurs caractéristiques ainsi que les liens entre eux.

Le fichier de sortie au format BRITE contient trois sections:

**Informations de modèle:** informations sur la topologie contenue dans le fichier. Inclut le nombre de nœuds et d'arêtes, ainsi que des informations spécifiques au modèle utilisé pour générer la topologie.

**Nœuds:** pour chaque nœud graphique, une ligne écrite dans le fichier de sortie

**Edge:** pour chaque lien graphique, une ligne écrite dans le fichier de sortie:

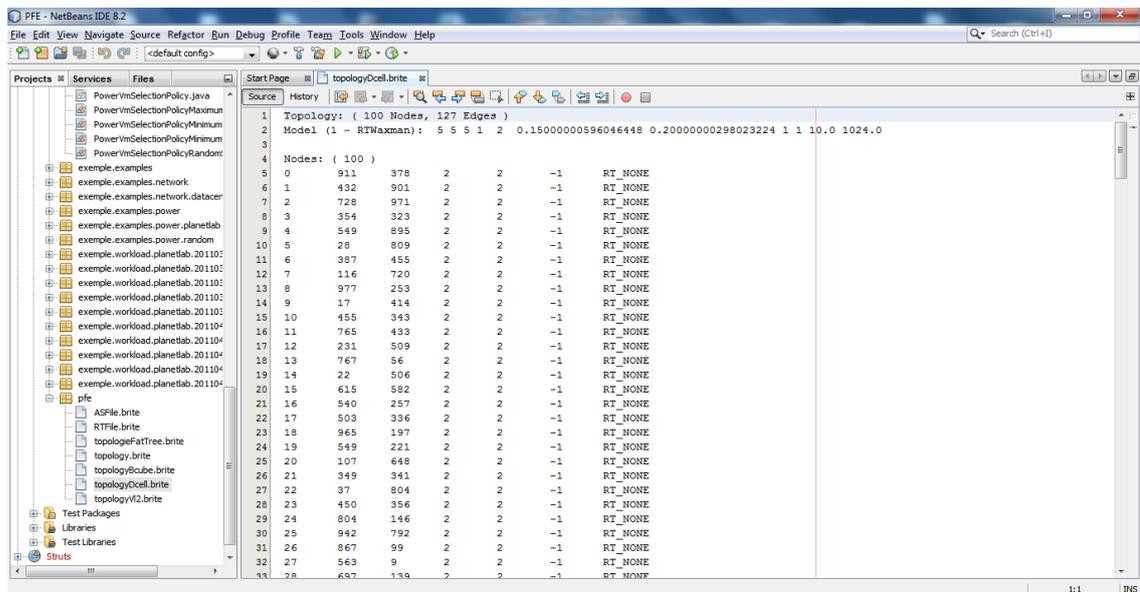


Fig. 3.10 Un aperçu sur le fichier brite

#### 4. Résultats obtenues

- Les résultats sont réalisés par l'outil jfreechart 1.0.19.
- Les graphes montrent la variation de l'énergie consommée par les switches et les machines physiques ainsi que l'énergie globale du Data center en fonction du temps
- L'énergie de chaque entité est calculé chaque 300 secondes et le calcul est terminé à 86400 secondes c'est-à-dire une simulation de 24 heures.

L'énergie du Data Center est calculée par la formule suivante:

$$\text{Energie Data Center} = \text{Energie des Hosts} + \text{Energie des Switches}$$

Pour montrer les performances de chaque topologie, nous avons utilisé quatre scénarios où chacun implémente une topologie.

La topologie	DCell
<p>Fig. 3.11 Host graph</p>	
<p>Fig. 3.12 Switch graph</p>	
<p>Fig. 3.13 Data Center graph</p>	

Tab 3.1 Tableau de résultats de DCell

La topologie	BCube
<p>Fig. 3.14 Host graph</p>	
<p>Fig. 3.15 Switch graph</p>	
<p>Fig. 3.16 Data Center graph</p>	

Tab. 3.2 Tableau de résultats de BCube

la topologie	Fat-tree
<p>Fig. 3.17 Host graph</p>	
<p>Fig. 3.18 Switch graph</p>	
<p>Fig. 3.19 Data Center graph</p>	

Tab. 3.3 Tableau de résultats de Fat-tree

La topologie	VL2
<p>Fig. 3.20 Host graph</p>	
<p>Fig. 3.21 Switch graph</p>	
<p>Fig. 3.22 Data Center graph</p>	
<p>Tabl. 3.4 Tableau de résultats de VL2</p>	

## 5. Comparaison

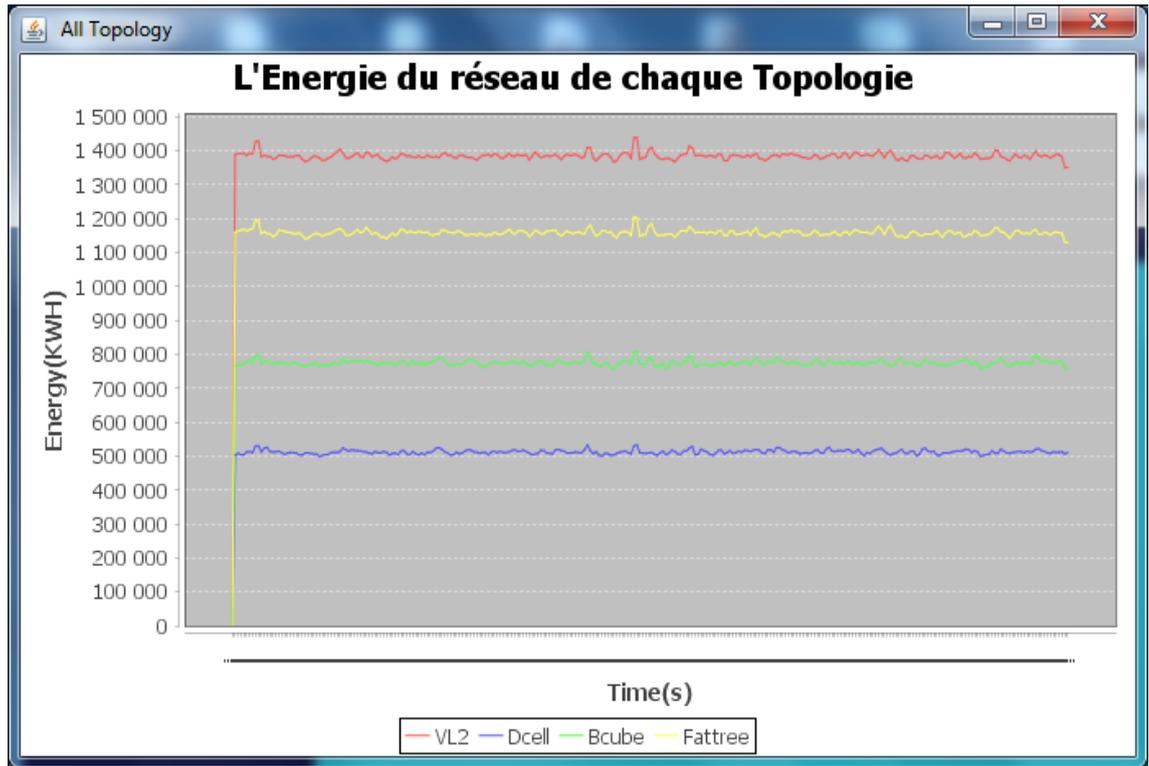


Fig. 3.23 graphe des quatre topologies

Les graphes montrent la différence dans la consommation d'énergie entre les différentes topologies. Donc on peut dire que la topologie du réseau a une influence non négligeable sur la consommation d'énergie dans le Cloud Computing. Ces graphes montrent que la topologie DCell est la plus efficace en termes de consommation d'énergie. Ces résultats s'expliquent par le fait que la topologie DCell utilise moins de switches que les autres topologies.

Dans ce qui suit, nous montrons le pourcentage d'éléments réseau pour faire fonctionner un total de 100 équipements pour chaque topologie.

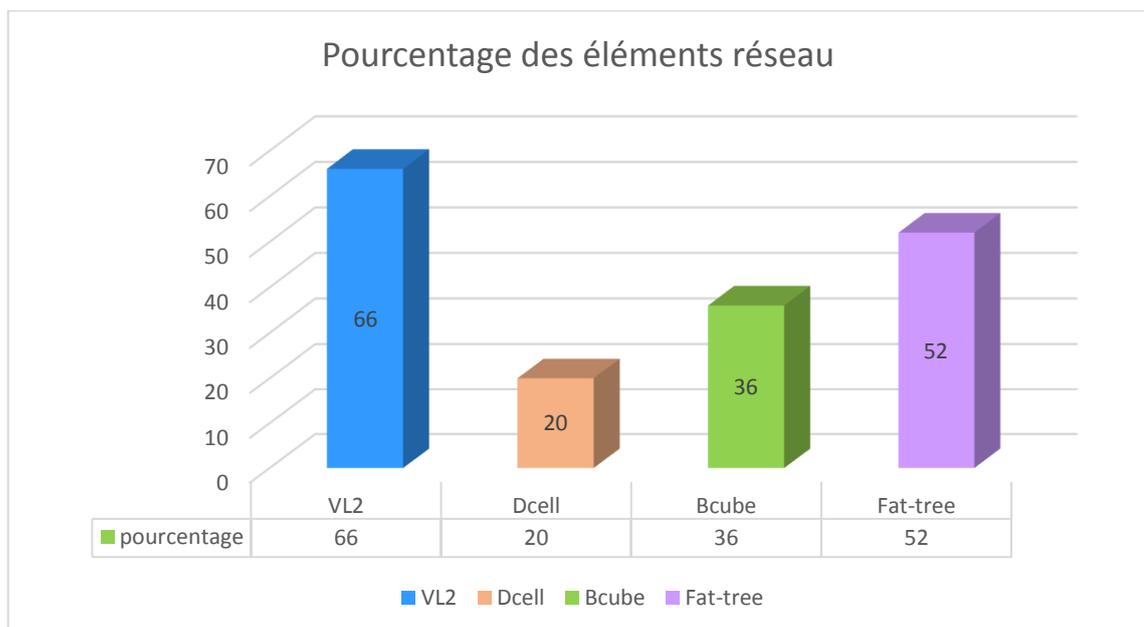


Fig. 3.24 graphe de pourcentage d'éléments réseau pour chaque topologie

Ce graphe confirme et montre le lien évident entre le nombre d'équipement réseau et la consommation énergétique du réseau.

## 6. Conclusion

Dans ce chapitre nous avons présenté notre application de simulation. Nous avons commencé par l'environnement de développement NetBeans puis le simulateur CloudSim, après nous sommes passées à la présentation des interfaces graphiques et les classes utilisées sans oublier la BDD.

Finalement, nous avons montré les résultats obtenus après une série de simulations dans le but de les comparés et découvrir qu'elle est la meilleure topologie et prouver l'idée théorique que la topologie à une influence importante sur la consommation d'énergie dans le cloud.

## Conclusion générale

**L**e Cloud Computing est la technologie qui a fait une évolution majeure dans le monde informatique, il a pris naissance en 1990 par les ASP (les fournisseurs de services d'applications), cette idée a progressé avec l'apparition des nouvelles solutions IT tel que la mise du logiciel sur le web, l'apparition d'Ébay, Amazon et l'arrivée de Google qui ont tous accéléré le processus.

Pouvoir utiliser des ressources informatiques sans les posséder est le rôle du Cloud, il met sur des serveurs localisés à distance des applications, des plateformes et des infrastructures pour permettre à l'utilisateur de les exploiter depuis n'importe où via Internet.

Selon les services offerts par les fournisseurs du Cloud Computing on distingue 3 types essentiels du Cloud IaaS, PaaS et SaaS. Le cloud peut aussi être privé, publique, communautaire ou hybride selon le type des utilisateurs.

Plusieurs solutions ont été proposées pour réduire la consommation d'énergie tel que le « bain d'huile » d'Asperitas, des logiciels qui optimisent le fonctionnement des centres de données, Les nouveaux serveurs de 2CRSI, la migration et La virtualisation avec ses différentes techniques émulation, Paravirtualisation et Hyper viseur.

Le choix de la topologie du réseau interne c'est-à-dire les liens entre les serveurs du Data Center représente aussi une solution importante qui réduit la consommation d'énergie. Chaque topologie classique en étoile, en bus, en anneau, en arbre et maillé ou plus complexe comme DCell, BCube, FiConn, VL2, DPillar, Fat-tree sont tous caractérisées par ses machines physiques et un nombre précis d'équipements réseaux. L'énergie totale des équipements réseaux diffère d'une topologie à une autre.

L'étude comparative que nous avons fait entre les quatre topologies DCell, BCube, Fat-tree et VL2 a confirmé l'influence non négligeable de la topologie réseau et a montré que DCell est la meilleure topologie car elle consomme moins d'énergie.

Souhaitant que notre travail qui a été réalisé avec 4 topologies soit réalisé avec d'autres topologies pour que les ingénieurs d'infrastructure réseau du Cloud puissent choisir la meilleure adaptée et améliorer le rendu énergétique du cloud.

## Référence Bibliographique

- [1] : Jean-Louis CAIR - Willy MUNSCH. Objectif Cloud. Une démarche pratique orientée services. Edition ENI. France : juillet 2014, 313 p
- [2] : <http://www.cloud-entreprise.info/historique-cloud-computing/> (consulté le 14/02/2019)27/3/19
- [3] : <https://missarte.wordpress.com/lhistoire-du-cloud-computing/> (consulté le 14/02/2019)
- [4] : <https://openclassrooms.com/fr/courses/4810836-decouvrez-le-cloud-avec-amazon-web-services/4810843-quest-ce-que-le-cloud> (consulté le 06/03/2019)
- [5] : [https://lexpansion.lexpress.fr/high-tech/le-cloud-computing-explique-aux-nuls\\_1384009.html](https://lexpansion.lexpress.fr/high-tech/le-cloud-computing-explique-aux-nuls_1384009.html) (consulté le 21/02/2019)
- [6] : <https://www.editions-eni.fr/open /mediabook.aspx?idR= c22b8320e91a957a931456986534e17f> (consulté le 06/03/2019)
- [7] : <http://le-cloud.net/> (consulté le 14/02/19)
- [8] : [https://fr.wikipedia.org/wiki/Cloud\\_computing](https://fr.wikipedia.org/wiki/Cloud_computing) (consulté le 21/02/2019)
- [9] : <https://www.culture-informatique.net/cest-quoi-le-cloud/> (consulté le 23/02/2019)
- [10] : <https://www.lebigdata.fr/definition-cloud-computing> (consulté le 23/02/2019)
- [11]: Meslem Yamina – Debbas Soumia Ismahèn. Ordonnancement et réplication de données dans le Cloud Computing. Saida : Université Dr. Tahar Moulay, 2016-2017, 67p
- [12] : <https://blog.3li.com/cloud-les-modeles-de-deploiement/> (consulté le 23/02/2019)
- [13] : <https://www.scalair.fr/blog/quels-sont-les-principaux-fournisseurs-de-cloud> (consulté le 23/02/2019)
- [14] : <https://www.lebigdata.fr/definition-cloud-computing> (consulté le 22/02/19)
- [15] : <https://fr.slideshare.net/cedricmora/prsentation-eurocloud-france-cloud-computing-en-france-cdric-mora> (consulté le 27/03/2019)
- [16] : <https://azure.microsoft.com/fr-fr/overview/what-is-cloud-computing/> (consulté le 23/02/19)
- [17] : Jean-François CARPENTIER. La sécurité informatique dans la petite entreprise. Etat de l'art et bonnes pratiques. 3<sup>ème</sup> édition : ENI, France, Janvier 2016, 444p
- [18] : <http://www.tradeflock.com/2017/12/16/iaas-cloud-iaas-definition/> (consulté le 05/03/2019)
- [19] : <https://azure.microsoft.com/fr-fr/overview/what-is-paas/> (consulté le 23/02/2019)
- [20] : <http://meship.com/Blog/2012/05/29/paas-next-cloud-battleground/> (consulté le 05/03/2019)
- [21] : <https://azure.microsoft.com/fr-fr/overview/what-is-saas/> (consulté le 24/02/2019)
- [22] : <https://www.lebigdata.fr/definition-saas> (consulté le 24/02/2019)
- [23]: <https://serverless-stack.com/chapters/fr/what-is-serverless.html> (consulté le 24/02/2019)
- [24]: <https://blogs.oracle.com/imc/data-warehouse-in-the-cloud-dwaas> (consulté le 24/02/2019)

- [25]: <https://searchdatamanagement.techtarget.com/definition/data-warehouse-as-a-service-DWaaS> (consulté le 24/02/2019)
- [26]: <https://www.intervision.com/solutions-services/orig-cloud-services/desktop-as-a-service/> (consulté le 24/02/2019)
- [27]: <https://www.dreamstime.com/stock-illustration-cloud-storage-computing-as-service-concept-data-devices-stored-image53013851> (consulté le 27/03/2019)
- [28]: <https://www.lebigdata.fr/top-10-vendeurs-cloud-computing-2017> (consulté le 06/03/2019)
- [29]: <https://missarte.wordpress.com/les-fournisseurs/> (consulté le 06/03/2019)
- [30]: [https://fr.wikipedia.org/wiki/Cloud\\_computing#Avantages](https://fr.wikipedia.org/wiki/Cloud_computing#Avantages) (consulté le 25/02/2019)
- [31]: <https://www.lebigdata.fr/raas-cloud-robotics-definition> (consulté le 15/03/2019)
- [32]: <https://www.outerplaces.com/science/item/16889-humanoid-robot-sophia-united-nations-speech-hanson-robotics> (consulté le 15/03/2019)
- [33]: [https://fr.wikipedia.org/wiki/Cloud\\_computing#Avantages](https://fr.wikipedia.org/wiki/Cloud_computing#Avantages) (consulté le 15/02/2019)
- [34]: <https://www.lebigdata.fr/definition-data-center-centre-donnees> (consulté le 20/03/2019)
- [35]: Dad Djouhra. Optimisation des performances des Data Centers des Cloud sous contrainte d'énergie consommée. Oran : Université Ahmed Benbela, 2017-2016, 116 p
- [36]: <http://hebergement-et-infrastructure.fr/actualites-et-innovations/8-6-millions-de-datacenters-dans-le-monde-en-2017> (consulté le 20/03/2019)
- [37]: Bedjaoui Mohammed Arslane – Khat Menouar. Étude de la consommation d'énergie dans le Cloud Computing. Tlemcen : Université Abou Bekr Belkaid, 2017/2018, 67 p
- [38]: <https://www.planetoscope.com/electronique/230-energie-consommee-par-les-data-centers.html> (consulté le 19/03/2019)
- [39]: <https://info-digitale.com/avantages-de-virtualisation/> (consulté le 20/03/2019)
- [40]: [https://docs.vmware.com/fr/VMware-vSphere/6.7/com.vmware.vsphere.vm\\_admin.doc/GUID-FE2B516E-7366-4978-B75C-64BF0AC676EB.html](https://docs.vmware.com/fr/VMware-vSphere/6.7/com.vmware.vsphere.vm_admin.doc/GUID-FE2B516E-7366-4978-B75C-64BF0AC676EB.html) (consulté le 21/03/2019)
- [41]: [https://fr.wikipedia.org/wiki/Migration\\_de\\_machines\\_virtuelles](https://fr.wikipedia.org/wiki/Migration_de_machines_virtuelles) (consulté le 25/03/2019)
- [42]: <https://www.iifa.fr/reseaux> (consulté le 03/04/2019)
- [43]: <https://www.commentcamarche.net/contents/512-topologie-des-reseaux> (consulté le 26/03/2019)
- [44]: [https://fr.wikipedia.org/wiki/Topologie\\_de\\_r%C3%A9seau](https://fr.wikipedia.org/wiki/Topologie_de_r%C3%A9seau) (consulté le 26/03/2019)
- [45]: <https://www.commentcamarche.net/contents/512-topologie-des-reseaux#topologie-en-bus> (consulté le 26/03/2019)
- [46]: <http://www.ense.eu/cablage-reseau-structure/> (consulté le 26/03/2019)
- [47]: <https://sites.google.com/site/technologisti/les-topologies/topologie-physique/topologie-etoile-etendue> (consulté le 26/03/2019)
- [48]: [https://www.memoireonline.com/03/15/8967/m\\_Conception-et-mise-en-place-dune-plateforme-de-securisation-par-synthese-et-reconnaissance-biom20.html](https://www.memoireonline.com/03/15/8967/m_Conception-et-mise-en-place-dune-plateforme-de-securisation-par-synthese-et-reconnaissance-biom20.html) (consulté le 03/04/2019)
- [49]: <https://www.vulgarisation-informatique.com/topologie-reseau.php> (consulté le 26/03/2019)

- [50]: Fatima Ezzahra Choukairy. Optimisation de la consommation d'énergie dans un environnement Cloud. Canada : Université Laval, 2018, 83 p
- [51]: <https://www.vulgarisation-informatique.com/topologie-reseau.php> (consulté le 26/03/2019)
- [52]: <https://blogchinmaya.blogspot.com/2017/04/what-is-fat-tree-and-how-to-construct.html> (consulté le 03/03/2019)
- [53]: Rodrigo S. Couto, Miguel Elias M. Camper, Luis Henrique M. K. Costa. A reliability analysis of Data Center Topologies. Rio de Janeiro: Université Federal, 6 p
- [54]: <https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/08-fattree.pdf> (consulté le 03/04/2019)
- [55]: <https://www.vulgarisation-informatique.com/topologie-reseau.php> (consulté le 03/04/2019)
- [56]: <https://www.supinfo.com/articles/single/7300-differentes-topologies-reseaux> (consulté le 03/04/2019)
- [57]: <https://docs.oracle.com/cd/E19284-01/819-4439/acrjn/index.html> (consulté le 27/04/2019)
- [58]: Alejandro Erickson, Iain A. Stewart, Javier Navaridas, Abbas E. Kiasari. The stellar transformation: from interconnection networks to datacenter networks. <https://arxiv.org/pdf/1510.02181.pdf> (consulté le 04/05/2019)
- [59]: [https://www.researchgate.net/figure/A-FiConn-2-with-n-4-The-FiConn-2-is-composed-of-4-FiConn-1-s-and-each-FiConn-1-is\\_fig3\\_224500462](https://www.researchgate.net/figure/A-FiConn-2-with-n-4-The-FiConn-2-is-composed-of-4-FiConn-1-s-and-each-FiConn-1-is_fig3_224500462) (consulté le 04/05/2019)
- [60]: [https://www.researchgate.net/publication/221092104\\_DPillar\\_Scalable\\_Dual-Port\\_Server\\_Interconnection\\_for\\_Data\\_Center\\_Networks](https://www.researchgate.net/publication/221092104_DPillar_Scalable_Dual-Port_Server_Interconnection_for_Data_Center_Networks) (consulté le 27/04/2019)
- [61]: [https://www.researchgate.net/figure/shows-a-DPillar-network-built-from-8-port-switches-There-are-two-server-columns-in-this\\_fig7\\_221092104](https://www.researchgate.net/figure/shows-a-DPillar-network-built-from-8-port-switches-There-are-two-server-columns-in-this_fig7_221092104) thèse de DPillar (consulté le 04/05/2019)
- [62]: <https://cacm.acm.org/magazines/2011/3/105326-vl2-a-scalable-and-flexible-data-center-network/abstract> (consulté le 04/05/2019)
- [63]: <https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/09-vl2.pdf> (consulté le 20/05/2019)
- [64]: <https://www.ivation.fr/contrat-de-maintenance-le-sla-informatique/> (consulté le 10/06/2019)
- [65]: <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203473-sla-service-level-agreement-definition-traduction/> (consulté le 10/06/2019)
- [66]: <http://www-igm.univ-mlv.fr/~forax/ens/java-avance/cours/pdf/old/I-%20Le%20langage%20Java.pdf> (consulté le 23/05/2019)
- [67]: <https://www.futura-sciences.com/tech/definitions/internet-java-485/> (consulté le 23/05/2019)
- [68]: <https://graal.ens-lyon.fr/~ecaron/m2rts/2015/blogbazm/> (consulté le 23/05/2019)

# Annexe

1. **CERN** : L'un des environnements informatiques scientifiques les plus exigeants du monde, et le lieu de naissance du World Wide Web
2. **Solutions IT** : désignent l'usage des ordinateurs, du stockage, des réseaux et des appareils, des infrastructures et des processus pour créer, traiter, stocker, sécuriser et échanger toutes sortes de données électroniques.
3. **Navigateur Mosaic** : est un navigateur Web développé à partir de fin 1992 au centre de recherches américain NCSA (National Center for Supercomputing Applications). C'est le navigateur qui a rendu le World Wide Web populaire.
4. **Navigateur Netscape** : était un navigateur web qui a dominé le marché au milieu des années 1990. Celui-ci a régressé face à la concurrence d'Internet Explorer. Il était édité par la société Netscape\_
5. **Palme PDA** : c'est un ordinateur de poche composé d'un processeur, de mémoire vive, d'un écran tactile et de fonctionnalités réseau dans un boîtier compact de petite taille.
6. **Webmail**: est une des deux façons de consulter son courrier électronique.
7. **Magic Quadrant**: est une série de rapports d'études de marché publiés par le cabinet de conseil en informatique Gartner.
8. **Gartner**: est une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées
9. **AWS Management Portal for vCenter**: il permet de gérer vos ressources AWS via VMware Vcenter.
10. **Enhanced Networking**: mise en réseau améliorée
11. **Azure**: c'est un ensemble de services Cloud destinés à aider notre organisation à relever les défis auxquels elle est confrontée.
12. **Hyper-V** : connu sous le nom de Windows Server Virtualisation, est un système de virtualisation basé sur un hyperviseur 64 bits de la version Windows Serveur 2008
13. **Azure Stack** : c'est un nouveau service qui apporte les modèles de déploiement d'application et de ressources d'Azure
14. **Cloud Google Compute Engine** : est une infrastructure en tant que service IaaS permettant aux utilisateurs de lancer des instances de machines virtuelles
15. **Apollo Global Management** : est un fonds de capital-investissement américain.
16. **multi-tenant** : ou multi-entité, c'est une architecture où une seule instance d'application va servir plusieurs clients
17. **Intergiciels** : est un logiciel tiers qui crée un réseau d'échange d'information entre plusieurs applications informatiques
18. **CAPEX**: il désigne les dépenses d'investissement d'une entreprise capitalisées au bilan
19. **Interconnectivité** : connecter des réseaux distincts.
20. **Humanoïde** : signifie ressemblant à l'humain.
21. **Energie grise**: est la quantité d'énergie consommée lors du cycle de vie d'un matériau ou d'un produit, elle est en effet cachée, indirecte, au contraire de celle qu'on connaît

22. **Greenpeace** : c'est une organisation indépendante des Etats, des pouvoirs politiques et économiques.
23. **Backup** : la sauvegarde des données
24. **Xen**: un logiciel libre de virtualisation, plus précisément un hyperviseur de machine virtuelle.
25. **Solaris** : un système d'exploitation Unix propriétaire développé par Sun Microsystems.
26. **Hyperappels** : une fonctionnalité qui permet d'utiliser les fonctions ou l'interface d'un logiciel à travers l'interface d'un ou plusieurs autres logiciels.
27. **Token ring** : anneau à jeton en français, est une topologie de réseau associée à un protocole de réseau locale qui fonctionne sur la couche « liaison » du modèle OSI
28. **Plug-and-Play** : qui signifie connecter et jouer, est une technologie qui permet aux SE de reconnaître rapidement et automatiquement les périphériques compatibles avec cette technologie dès le branchement.

## ملخص

الحوسبة السحابية هي تكنولوجيا توفر العديد من الخدمات. لها مزايا وعيوب. العيب الأول الذي يمكننا التحدث عنه هو استهلاك الطاقة ذو الأهمية القصوى في مراكز البيانات نظرا لتكلفته المرتفعة للغاية بسبب الخوادم ونظام التبريد وكذا العناصر المكونة لشبكة الحوسبة السحابية. دراستنا تهتم أساسا بالاستهلاك الطاقوي لشبكة الحوسبة السحابية وبشكل أكثر دقة هيكل (طوبولوجيا) هذه الأخيرة.

الطوبولوجيا تمثل عاملا مهما في تقليل استهلاك الطاقة عن طريق اختيار الهيكل الأكثر كفاءة. اختيار الطوبولوجيا المستخدمة أمر ضروري. يمكن أن تكون هذه الأخيرة من النوع الأساسي مثل ناقل الهيكل، نجمة ... إلخ، أو من النوع الأكثر تقدماً مثل DCCell، BCube، FiConn ... إلخ.

الغرض من هذه الدراسة هو تسليط الضوء على تأثير الطوبولوجيا في استهلاك الطاقة للحوسبة السحابية، وكذلك تحديد أفضل بنية من بين البنيات المدروسة.

الكلمات المفتاحية: الحوسبة السحابية، استهلاك الطاقة، الطوبولوجيا، البنية الأساسية.

## Résumé

Le cloud computing est une technologie qui offre plusieurs services. Il a des avantages et des inconvénients. Le premier inconvénient dont nous pouvons parler, est la consommation d'énergie qui a une importance primordiale dans les Data Centers, vu que son coût qui est très élevé à cause de ses serveurs et son système de refroidissement ainsi que les éléments de l'infrastructure réseau. Notre étude s'intéresse principalement à l'aspect réseau de la consommation d'énergie dans le Cloud et plus précisément la structure (topologie) de ce dernier.

La topologie représente un facteur important pour réduire la consommation d'énergie en choisissant la plus efficace. Le choix de la topologie utilisée est primordial. Cette dernière peut être de type basic tel que la topologie en bus, étoile...etc., ou de type plus aboutis tel que : DCCell, BCube, FiConn...etc.

Le but de cette étude est de mettre en évidence l'influence de la topologie dans la consommation d'énergie dans le cloud, ainsi que la détermination de la meilleure structure parmi celles étudiées.

Mots clés: cloud computing, consommation d'énergie, topologie, infrastructure.

## Abstract

Cloud computing is a technology that offers several services. It has advantages and disadvantages. The first drawback we can talk about is the energy consumption that is of paramount importance in Data Centers, as its cost is very high because of its servers and cooling system as well as the infrastructure elements network. Our study focuses mainly on the network aspect of the energy consumption in the Cloud and more precisely the structure (topology) of the latter.

Topology is an important factor in reducing energy consumption by choosing the most efficient one. The choice of the topology used is essential. The latter can be of basic type such as topology bus, star ... etc., or more advanced type such as: DCCell, BCube, FiConn ... etc.

The purpose of this study is to highlight the influence of the topology in the energy consumption in the cloud, as well as the determination of the best structure among those studied.

Keywords: Cloud Computing, the energy consumption, topology, infrastructure