

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. Tahar Moulay SAIDA

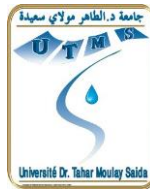
جامعة د الطاهر مولاي سعيدة

Faculté : Technologie

كلية التكنولوجيا

Département : Informatique

قسم : الإعلام الآلي



## MEMOIRE DE MASTER

Option : Sécurité Informatique et Cryptographie

# THEME

Optimisation de ressources dans le Cloud

Computing

Présenté par :

- Didaoui Abdelhak
- Chetti Elhadj Bachir

Encadré par :

✓ Mme Meddah

2015-2016

# *DEDICACE*

A peine nous venons de terminer la rédaction du mémoire de fin de cycle de Master, je voudrais très vite le dédier avec une immense joie, un grand honneur :

A mes très chers parents en signe de ma profonde et affectueuse reconnaissance pour leur amour sans mesure, tous les sacrifices, les soutiens, les tolérances et les encouragements qu'ils ont bien voulu consentir pour moi. Tous les mots restent faibles pour leur exprimer mes sentiments et qu'ils acceptent seulement ces lignes en guise de témoignage

A mes très chères sœurs « asmaa, amina, halima »

A mes amies « Brahim, mohamed, said »

A mon cher binôme BACHIR que je le remercie beaucoup.

A toute la famille chacun à son nom

Et à nos professeurs qui nous ont beaucoup aidés et qui ont fait en sorte que ce travail voie le jour.

Enfin à tous mes ami(e)s et connaissances.

DIDAOUI ABDELHAK

# *DEDICACE*

Je dédie ce travail

A mes chers parents pour leurs soutiens et leurs patiences.

A mes frères : « Ayoub , ilyas , mohamed said » et Mes sœur : « meryam , chifaa , waffa »

A toute ma grande famille.

A tous mes amies : « abd elhak & toufik »

A toute personnes qui m'a aidé un jour de près ou de loin à réussir jusqu'ici.

Chetti elhadj elBachir

# REMERCIEMENT

Ce travail a été le labeur d'une année et n'aurait probablement jamais été mené à terme sans le soutien d'un grand nombre de personnes que nous remercions vivement et très sincèrement à remercier.

Nous commencerons d'abord par remercier notre professeur **Mme Meddah** que nous avons eu l'honneur de rencontrer durant notre passage de l'université

« **Dr MOULAY TAHAR** »

Nous la remercions pour son aide précieuse, la qualité de ses conseils et pour avoir toujours été là pour nous encourager, nous soutenir, et nous avoir permis de reprendre confiance en nous lorsque nous avons eu une décision importante à prendre à un moment critique de ce travail.

Nous remercions aussi Notre chère professeur **Mr Adjir** qui était un peu malchanceux avec nous pour compléter le mémoire précédent pour des raisons impérieuses de notre part.

Il y a également d'autres personnes à qui nous sommes profondément reconnaissants pour avoir su nous écouter et nous redonner de l'énergie dans les moments les plus critiques notre chère professeur **Mr Amine Abd elmalek**, notre collègue **Rahmani amine**, Notre chère professeur ce que nous aimons beaucoup **Fahci Mahmoud**, **Mlle kabli** et **Mlle Yahlali** ...ect

Ne pouvant malheureusement pas citer toutes les personnes que nous avons rencontré durant notre travail et qui ont contribué d'une façon ou d'une autre, de près ou de loin à l'aboutissement de ce mémoire nous les remercions tous d'avoir été là à cet instant précis où nous les avons rencontrées et où ils nous ont apporté cette aide qui a sûrement contribué à aller au bout de cette aventure .

## Table des matières

<b>Introduction général .....</b>	<b>6</b>
 <b>Chapitre I : Cloud Computing</b>	
<b>I.1 Introduction.....</b>	<b>8</b>
<b>I.2 Cloud Computing.....</b>	<b>8</b>
I.2.1. Définition .....	8
I.2.2. Caractéristiques d'un service Cloud .....	8
I.2.2.1. Infrastructure partagée.....	8
I.2.2.2. Provisionnements dynamiques.....	8
I.2.2.3. Accès via le web.....	8
I.2.2.4. Gestions par mesures .....	9
I.2.3. Types de Cloud .....	9
I.2.3.1. Cloud privé .....	9
I.2.3.2. Cloud communautaire .....	9
I.2.3.3. Cloud publique.....	9
I.2.3.4. Cloud hybride .....	9
I.2.4. L'architecture de cloud .....	9
I.2.4.1. Software as a service.....	10
I.2.4.1.1. caractéristique d'un SaaS.....	10
I.2.4.2. Plate-forme as a service.....	11
I.2.4.2.1. caractéristique d'un PaaS.....	12
I.2.4.3. Infrastructure as a service.....	12
I.2.4.2.1. caractéristique d'un IaaS.....	12
I.2.5. Storage as a service StaaS.....	13
I.2.5.1. Fonctionnement du StaaS.....	13
I.2.5.2. Pourquoi le stockage sur le nuage est important ? .....	13
I.2.6. Avantages de cloud.....	14
I.2.6. Inconvénients de cloud.....	14
<b>I.3. La sécurité dans le Cloud Computing.....</b>	<b>15</b>
I.3.1. Les risques sécuritaires de l'informatique a nuage.....	15
I.3.1.1. Perte de contrôle sur les données.....	15

I.3.1.2.Manque de séparation et désolation des données .....	15
I.3.1.3. Non-respect des réglementations et risque d'accès des étrangers .....	15
I.3.1.4.Captivité.....	16
I.3.1.5.perte de données.....	16
I.3.1.6.Utilisation malveillante.....	16
I.3.2.L'aspect sécuritaire dans le cloud computing.....	16
I.3.2.1.sécurité de contrôle.....	17
I.3.2.2.sécurité des données .....	18
I.3.2.3.sécurité des infrastructures.....	19
I.4.Le marché du cloud computing .....	20
<b>I.5.Conclusion.....</b>	<b>22</b>

## **Chapitre II : QoS et enjeu énergétique**

<b>II.1.Introduction .....</b>	<b>23</b>
<b>II.2.Qualité de service .....</b>	<b>23</b>
II.2.1. Définition .....	23
<b>II.3.Catégories .....</b>	<b>24</b>
II.3.1. Performance .....	25
II.3.1.1.Temp d'exécution.....	25
II.3.1.2.Latence débit.....	25
II.3.1.3.Temps de réponse.....	25
II.3.2. Sûreté de fonctionnement .....	26
II.3.2.1.Précision .....	26
II.3.2.2.Fiabilité .....	26
II.3.2.3.Disponibilité.....	26
II.3.2.4.Capacité.....	27
II.3.2.5.Extensibilité .....	27
II.3.2.6.Réactivité.....	28
II.3.2.7.Dynamisme.....	28
II.3.2.8.Robustesse .....	28
II.3.2.9.Stabilité.....	29
II.3.2.10.Tolérance aux fautes .....	29
II.3.2.11.Agilité.....	29
II.3.2.12.Durabilité.....	30

II.3.2.13.Assurances .....	31
II.3.2. 14.Convivialité.....	31
II.3.2.15.Personnalisation .....	31
II.3.2. 16.Mise a jour automatique .....	32
II.3.3.Sécurité données .....	32
II.3.3.1.Authentification.....	32
II.3.3. 2.Autorisation.....	33
II.3.3.3.Intégrité .....	33
II.3.2. 4.Confidentialité .....	34
II.3.3.5.Responsabilisation .....	34
II.3.3. 6.Traçabilité .....	35
II.3.3.7.Cryptage .....	35
II.3.2. 8.Isolation .....	36
II.3.3.9.Cycle de vie des données .....	36
II.3.3. 10.Non-répudiation .....	36
II.3.4.Coûts.....	36
II.3.4.1. Coût de service.....	36
II.3.4. 2. Coût d'énergie .....	37
II.3.4.3.Empreinte carbone .....	37
II.4. Evaluation de l'empreinte environnemental .....	37
II.4.1.Consommation électrique .....	38
II.4.2.Coefficient d'énergie renouvelable .....	38
II.5. Technique pour minimiser l'empreinte environnemental .....	38
II.5.1.Matériel .....	39
II.5.1.1. Refroidissement .....	39
II.5.1. 2. La consommation des composants .....	39
II.5.2.Virtualisation .....	40
<b>II.5. Conclusion .....</b>	<b>43</b>
 <b>Chapitre III : Gestion d'énergie dans le Cloud Computing</b>	
<b>III.1.Introductin.....</b>	<b>44</b>
III.2.Modèle de gestion d'énergie .....	44
III.3.Etat d'une ressource .....	44
III.3.1.Opération de variation de vitesse d'exécution .....	46



III.3.2.Variation de vitesse de façon native .....	46
III.3.3.Regroupement des traitements .....	46
III.4.Gestion d'énergie sur quelques composants d'un serveur .....	47
III.4.1. Le CPU.....	47
III.4.2. Le Disque .....	51
III.4.3. Le périphérique réseau .....	54
III.4.4. La Mémoire .....	55
III.4.5. Gestion d'énergie d'autres composants dans un IaaS .....	58
III.5.L'optimisation est une vertu écologique ? .....	58
III.5.1. Comment peut-on optimiser la consommation d'énergie des Data Center ?.....	59
<b>III.6.Conclusion</b> .....	60

## **Chapitre IV: Outils de simulation**

<b>IV.1.Introduction</b> .....	61
IV.2.Pour quoi la simulation.....	61
IV.3.Outils de simulation .....	62
IV.3.1. SIMGRID.....	62
IV.3.2. GroudSim.....	62
IV.3.3. GSSIM.....	62
IV.3.4. CloudSim.....	63
IV.3.5. GreenCloud.....	63
IV.4.Caractéristiques du GreenCloud .....	64
IV.5.Architecture du GreenCloud .....	65
<b>IV.6.Conclusion</b> .....	66

## **Chapitre V: Implémentation**

<b>V.1.Introduction</b> .....	67
V.2. Environnement de développement .....	67
V.2.1Qu'est-ce que Eclipse ?.....	67
V.3.Introduction au simulateur GreenCloud .....	68
V.4.Installation de GreenCloud .....	68
V.4.1.Installation sur une machine virtuelle (VM).....	68
V.4.2.Installation sur un système réel.....	68

V.5.Exemple de Simulation.....	69
V.6.Changement des paramètres de Cloud .....	69
V.6.1.Interface de changement des paramètres.....	75
V.7. Résultats et expérimentations .....	76
V.7.1. 1 <sup>ère</sup> expérimentation : les topologies de centre e données.....	76
V.7.2. 2 <sup>ème</sup> expérimentation : scheduling.....	77
V.7.3. 3 <sup>ème</sup> expérimentation : Augmentation du Nombre Vms.....	78
<b>V.8. conclusion.....</b>	<b>79</b>
Conclusion général .....	80
Bibliographie.....	81
Liste des figures .....	86
Liste des Tableaux .....	87
Liste des abréviation .....	88

Introduction générale

## ***Introduction générale***

Le Cloud Computing est devenu un sujet de recherche extrêmement intéressant. Selon la définition du National Institute of Standards and Technology (NIST), le Cloud Computing est l'accès via un réseau de télécommunications, à la demande et en libre service, à des ressources informatiques partagées et configurables. Il s'agit donc d'une délocalisation de l'infrastructure informatique. Par ailleurs, la prolifération du Cloud computing a abouti à la création de data center à grande échelle dans le monde entier contenant des milliers de nœuds. Cependant, les data center du Cloud consomment d'énormes quantités d'énergie électrique entraînant ainsi des coûts d'exploitation élevés .

Cette consommation fait estimée au niveau mondial à 50% entre 2005 et 2010, et depuis 2010 elle représente entre 1.1% et 1.5% de l'utilisation de l'électricité mondiale. Par conséquent, les politiques gouvernementales à travers le monde poussent les fournisseurs de data center à réduire la consommation d'énergie et à devenir plus « green » à l'égard de l'environnement.

Pour résoudre le problème de la forte consommation d'énergie, il est nécessaire d'éliminer les inefficacités dans la façon dont l'électricité est livrée aux ressources informatiques, et la façon dont ces ressources sont utilisés pour servir les workloads d'applications. Cela peut être réalisé par l'amélioration des infrastructures physiques des data center et les algorithmes d'allocation et de gestion des ressources. D'ailleurs, les avancées récentes dans la conception du Data center ont abouti à une augmentation significative de l'efficacité de l'infrastructure. Une source de gaspillage d'énergie réside dans l'utilisation inefficace des ressources informatiques. La plus part du temps, les serveurs utilisent 10 à 50% de leur capacité totale, conduisant à des dépenses supplémentaires de sur-approvisionnement, et donc un cout total de l'acquisition (TCA) supplémentaire.

Même complètement inactifs Continuent de consommer jusqu'à 70% de leur puissance, ce qui apporte des défis de gestion éco-efficace de la consommation d'énergie.

C'est dans ce contexte que l'objectif principal de notre projet est défini. Comment peut-on optimiser l'énergie consommée dans le cloud computing.

Afin d'atteindre cet objectif, un certain nombre de buts doivent être réalisés :  
Comprendre l'importance du Cloud Computing, son architecture, ses types et les questions liées au Cloud Computing en termes de gestion de data center .

Pour attendre notre sujet :

- ✓ Une Introduction générale
- ✓ Le *premier chapitre* est consacré au Cloud Computing, il décrit entre autres ses caractéristiques, son architectures, ses éléments constitutifs, ses modèles de déploiement ainsi que ses services.
- ✓ Le *deuxième chapitre* présente les différentes catégories de QoS en mettant l'accent sur les coûts énergétiques, on cite ainsi les grandes approches qui visent à la consommation d'énergie .
- ✓ Le *troisième chapitre* propose la gestion d'énergie dans le cloud computing, en évoquant l'état d'une ressource ..
- ✓ Le *quatrième chapitre* présente le simulateur « Green cloud », ses fonctionnalités, ses caractéristiques, son architecture.
- ✓ Le *cinquième chapitre* expose les résultats obtenus de nos expérimentations.
- ✓ La Conclusion générale dans la quelle on discute les résultats et perspectives.

# Chapitre 1

## Cloud Computing

## **I.1 Introduction :**

Le besoin actuel des entreprises à la gestion de leurs données ne cessent de croître au niveau de volume de données à traiter, pour cela un nouveau concept est apparu face à ce développement et au cout élevé du marché des supercalculateurs et des produits informatiques en générale, c'est le « **CLOUD COMPUTING** ».

## **I.2 Cloud Computing :**

### **I.2.1 définition :**

Le Cloud Computing [A1] sert à utiliser les mémoires et les capacités de calcul des serveurs répartis à travers le monde afin de fournir un service de calcul, de stockage ou d'autres services via le réseau d'une manière dont le client n'a aucun contrôle sur ses données lors de leur existence dans le Cloud alors qu'il n'est autorisé qui à accéder d'une manière évolutive à ces services via les outils web (les navigateurs web en générale), cette limitation de contrôle pose des problèmes et des critiques .

### **I.2.2. caractéristiques d'un service Cloud :**

Un service Cloud possède des caractéristiques distinctives aux autres architectures :

#### **I.2.2.1.infrastructure partagée :**

Passer à la virtualisation signifie de rendre le partage des ressources possible, c'est le cas du Cloud qui sert à exploiter le maximum des infrastructures des utilisateurs.

#### **I.2.2.2.provisionnementnements dynamiques :**

La gestion des ressources dans un Cloud sert à allouer dynamiquement les ressources aux utilisateurs selon le besoin, bien sûr, cette gestion a besoin d'un maintien de haut niveau afin d'assurer l'allocation sécurisée et en temps réel.

#### **I.2.2.3.accès via le web :**

L'accès vers le Cloud est fait via le web et ses protocoles, alors que cet accès est indépendant du matériel utilisé (ordinateur, lap top, téléphone mobile...etc.).

#### **I.2.2.4.gestions par mesures :**

Le Cloud utilise la notion de mesure afin de contrôler et optimiser ses services ainsi ces mesures aident à la facturation et la à fourniture des rapports aux utilisateurs afin de les informer sur le cout des service qu'ils avaient consommé.

### **I.2.3. types de Cloud :**

#### **I.2.3.1.Cloud privé :**

Ce type est fourni pour un seul client (entreprise ou organisation) dont les opérations à exécuter peuvent être exécutées de la maison, de l'intérieur de l'entreprise ou bien de n'importe quel endroit selon des permissions bien définies.

#### **I.2.3.2.Cloud communautaire :**

Est une extension des Cloud privé, son but est de gérer les projets de collaboration entre les entreprises alors que les services ne sont fournis qu'à ses entreprises, l'avantage de ce type réside dans le cout de financement qui sera divisé entre les clients.

#### **I.2.3.3.Cloud publique :**

C'est le contraire du Cloud privé et le moins cher, ce type sert à fournir ses services pour n'importe qui alors que l'accès aux services est de n'importe où et par n'importe qui.

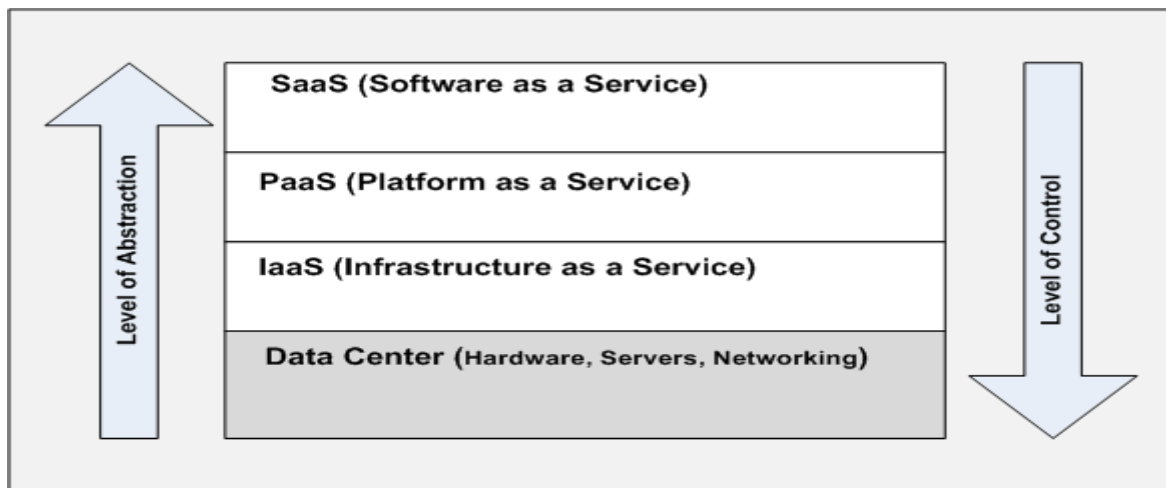
#### **I.2.3.4.Cloud hybride :**

C'est un ensemble de Cloud de différents types (publique et privé) reliés entre eux. Ce type de Cloud sert à gérer la transmission des données et/ou les applications d'un Cloud à l'autre, il est la meilleure solution pour qu'une entreprise puisse profiter des avantages des Cloud pour gérer ses données confidentielles et publiques.

### **I.2.4.l'architecture de Cloud :**

Un service Cloud est construit en trois (03) couches de bases dont chacune représente un reflet d'un composant de l'informatique comme le montre le schéma suivant :





**Figure I.1 :** les couches Cloud Computing

#### **I.2.4.1. Software as a Service: (SaaS)**

Cette couche sert à fournir les logiciels et les applications comme étant des services exécutables sur le web, cette fonctionnalité s'intéresse à l'informatique de haut niveau, autrement dit, les services de ce type sont des logiciels simples tels que les logiciels d'édition de texte sous forme des services dans le web.

L'avantage de ce type est qu'un client n'a besoin ni d'installer ni de licence et clé d'activation des applications.

##### **I.2.4.1.1. Caractéristiques d'un SaaS :**

- Accès via le web vers les applications commerciales.
- La gestion centrale des applications.
- Les applications sont délivrées sous forme multicast (un vers plusieurs).
- L'utilisateur n'a pas besoin de faire des mises à jour et des patches sur les applications
- Les API doivent permettre l'intégration des différentes parties des applications.

Les services SaaS ne sont pas toujours la solution optimale, l'utilisation des applications en nuage dépend des besoins des clients :

**❖ Cas ou on a besoin d'utiliser le SaaS :**

- Les applications qui ont un rôle d'interaction entre l'entreprise et l'extérieure
- Les applications qui ont besoin d'être accédées via le web ou un autre périphérique tel que le smartphone.
- Des applications qui ont besoin d'être utilisées dans un temps limité.
- Des applications telles que les applications de facturation utilisées une fois par mois.

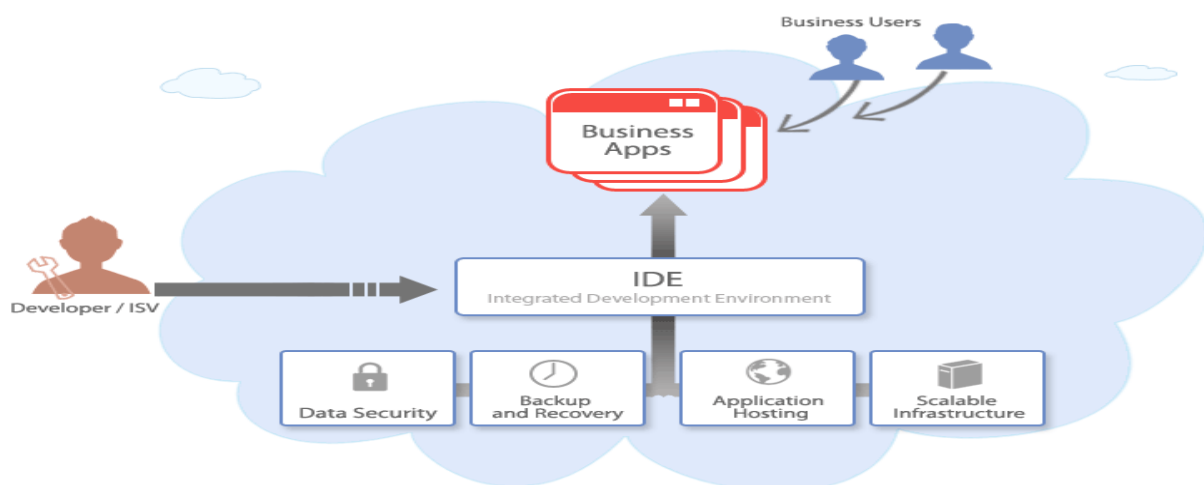
**❖ Cas ou on ne doit pas utiliser le SaaS :**

- Des applications qui exigent un calcul rapide en temps réel
- Des applications qui utilisent des données internes dont la réglementation les empêche d'être externalisées.
- Des applications dont l'existence d'une solution locale remplit tous les besoins de l'entreprise.

**I.2.4.2 plateforme as a service :(PaaS)**

C'est une extension de SaaS en bas niveau, autrement dit, le PaaS est un service qui vise une catégorie spéciale des clients, ce sont les développeurs dont ce service permet de fournir une plateforme de développement des applications en ligne ainsi que la possibilité de les tester.

Le positionnement de service PaaS est illustré par le schéma suivant :



**Figure I.2 : l'emplacement d'un service PaaS**

#### **I.2.4.2.1.Caractéristiques de PaaS :**

- Service de développement, teste, déploiement, et maintien des applications dans le même environnement.
- Création des interfaces d'utilisateur basées sur le web permet la gestion des différents scénarios.
- Un service multi-locataire dont il permet à plusieurs utilisateurs d'utiliser le même environnement simultanément.
- Construit une évolutivité des logiciels développés, y compris l'équilibrage de charge et le basculement
- Intégration des services web et les bases de données via un support standards.
- Un support de développement collaboratif entre les développeurs.

Comme le SaaS, le PaaS ne présente pas une solution optimale dans tous les cas,

Les cas suivants exigent de ne pas utiliser un environnement de développement en nuage :

- Cas dont des approches et langages propriétaires peuvent impacter le développement.
- Cas dont la performance des applications requière la personnalisation du matériel à utiliser ...etc.

#### **I.2.4.3.infrastructure as a service :(IaaS)**

L'IaaS représente le moyen de gestion des infrastructures Cloud et des ressources tels que les serveurs de calcul et de stockage...etc. Ces services offrent un avantage important sur le coût, l'IaaS peut être considéré comme le reflet de virtualisation des hyper viseur dans l'informatique en nuage. L'IaaS sert à allouer des ressources distantes aux clients en se basant sur le mécanisme de virtualisation.

##### **I.2.4.3.1.Caractéristiques d'IaaS :**

- Le déploiement des ressources comme étant des services.
- Permet l'allocation dynamique des ressources.
- Une allocation économique basée sur le concept de « **PAY WHAT YOU USE** ».
- Permet à plusieurs utilisateurs d'utiliser la même ressource simultanément grâce au concept d'hyper viseur.

L'IaaS représente une solution optimale pour les organisations des capitaux limitées qui ne peuvent pas investir sur le matériel ainsi que les entreprises qui évoluent rapidement alors que leurs besoins de matériels croissent aussi. L'IaaS pose un problème dans certains cas tels que :

- Cas dont les règles d'organisations mettent l'externalisation des données difficile.
- Cas dont l'organisation possède déjà des ressources de hautes capacités et n'a pas besoins d'autres ressources.

### **I.2.5.Storage as a Service StaaS:**

L'accroissement actuel de volume de données à stocker dans les entreprises cause des problèmes concernant l'espace de stockage ceci à pousser à chercher une solution plus au moins économique, cette solution représente précisément ce qu'on appelle « **STORAGE AS A SERVICE** », le Cloud de stockage est une extension des services Cloud de type Infrastructure as a service, le StaaS est un service permettant le stockage des données sur un serveur géant à distant via le web.

#### **I.2.5.1.fonctionnement du StaaS:**

Le service StaaS permet à un client de connecter à un serveur via une interface fournie par le web, cette interface permet l'échange des données et des documents, alors que le client peut envoyer n'importe quel document au serveur, le mécanisme de stockage chez le serveur est basé sur des concepts importants de l'informatique notamment l'indexation automatique qui permet la recherche d'information.

Le StaaS peut être considéré comme un support de stockage intelligent qui permet le stockage, la récupération ainsi que le traitement en ligne des documents.

#### **I.2.5.2.pourquoi le stockage sur le nuage est important ?:**

L'utilisation des services de stockage sur le nuage n'est pas toujours dû au débordement de taille des supports physiques existant, ce service peut être aussi considéré comme une solution de sauvegarde et de récupération des données en cas d'une panne de système ou de perte de support de stockage, le mécanisme de sauvegarde est aussi implémenté à l'intérieur du nuage, la gestion de ce dernier sert à stocker plusieurs copies des mêmes données dans des différents emplacements à travers le monde.

Un autre avantage important de StaaS est la facilité de partage de données entre les clients car le service de stockage est un service de partage multicast des données.

### **I.2.6. avantages de Cloud :**

Le Cloud et malgré ses inconvénients donne lieu à des avantages importants qui résident dans le coût d'utilisation qui a été diminué car l'utilisateur d'un Cloud n'a pas besoin d'installer les logiciels sur son PC (un navigateur web suffit) ce qu'il conduit à des avantages de moindre de maintenance et un cout de licence moins puis que les applications sont externalisées.

Les services Cloud sont accessibles de n'importe quel équipement qui dispose d'un accès à internet alors que les applications d'une telle entreprise ne sont accessibles que de l'intérieure de l'entreprise.

Un autre avantage qui fait la différence entre un Cloud et un réseau informatique car le Cloud fournit la possibilité d'accéder de n'importe où via le web et de faire des calculs en utilisant les ressources de l'entreprise et non pas de l'équipement utilisé pour l'accès contrairement à l'informatique réseau qui sert à partager les applications dans l'entreprise et non pas de l'hors, cet avantage présente la solution optimale des projets de collaboration entre les entreprises du monde à travers le monde, c'est ce qu'on appelle la multi-sociétés, qui n'est pas garantie par l'informatique réseau et même l'utilisation de cette dernière dans les projets de collaboration pose un problème de cout surtout au niveau de sécurité dont l'utilisation des mécanismes tels que les VPNs est couteuse en coté finance et implémentation (un VPN occupe toute une ligne de connexion).

### **I.2.7 inconvénients de Cloud Computing :**

Le Cloud présente des inconvénients majeurs que l'on peut les résumer dans les points suivants :

- Les lieux de stockage et de traitement posent un problème car les clients n'ont aucun contrôle sur les ressources utilisées, donc le stockage peut être dans d'autres pays ce qu'il touche les réglementations et les lois du pays en matière de protection de données.

- Un autre inconvénient réside dans l'utilisation de l'internet et ceci pose le problème de besoin de connexion de haute qualité et les problèmes de connexion telles que les coupures.
- L'inconvénient le plus important est l'insécurité et la sensibilité aux attaques exploitant les failles des protocoles web.

### **I.3.1a sécurité dans le Cloud Computing:**

L'un des problèmes majeurs qu'un distributeur d'un service Cloud doit prendre en considération est la sécurité des données qui utilisent ces services [A2] :

#### **I.3.1. les risques sécuritaires de l'informatique en nuage :**

On peut distinguer six (06) catégories des risques qu'un utilisateur peut les constater dans l'informatique en nuage :

##### **I.3.1.1.perte de contrôle sur les données :**

La gestion du Cloud est basée sur la centralisation ce qu'il empêche le client de prendre le contrôle sur ses données lors de leur existence dans le nuage, cela peut conduire à un état de l'insécurité des données et d'absence de confiance entre le client et le serveur, cette délocalisation des données est due à l'interconnexion mondiale et l'utilisation des mémoires virtuelles.

##### **I.3.1.2.manque de séparation et d'isolation des données :**

L'informatique en nuage permet à plusieurs utilisateurs d'utiliser le même nuage simultanément ceci crée un environnement approprié pour qu'un attaquant puisse distribuer son attaque sur plusieurs clients.

##### **I.3.1.3.non-respect des réglementations et risque d'accès des étrangers :**

Le stockage dans le nuage est fait sur plusieurs supports dans le monde entier, cette délocalisation pose un problème concernant le respect de la confidentialité ainsi que les règles de gestion des données des clients, un autre problème réside dans la possibilité de stockage dans des pays qui ne peuvent pas assurer un bon service en matière de protection des données, et de qualité de connexion.

#### **I.3.1.4.captivité :**

La technologie et l'infrastructure utilisées par le prestataire des données jouent un rôle très important dans le système de nuage car un simple problème de faiblesse d'interopérabilité et de portabilité des services Cloud peuvent causer un problème de captivité des données, ce problème peut être touché lors de migration des données pour un autre prestataire.

#### **I.3.1.5.perte de données :**

Le Cloud est basé sur le web et utilise un ensemble d'équipements physiques et son architecture ne dépasse pas les mécanismes de bases de l'informatique, ce qui le rend une cible des attaques de cybercriminalité qui exploitent les faiblesses des protocoles du web et causent une perte d'information d'où la nécessité de mettre en œuvre un mécanisme de sauvegarde et un autre mécanisme de correction d'erreur.

#### **I.3.1.6.utilisation malveillante :**

La politique de sécurité du nuage ne vérifie pas le principe de conception ouverte, dans ce cadre, les utilisateurs d'un nuage ne peuvent pas connaître la gestion de confidentialité et des droits d'accès ceci peut conduire à un état d'utilisation malveillante et abusive des données.

### **I.3.2 l'aspect sécuritaire dans le Cloud Computing :**

Les fournisseurs de nuage ne cessent de développer et de chercher toute solution pour assurer la sécurité de leurs services.

En 2009, le groupe « **CLOUD SECURITY ALLIANCE** » a défini un modèle standard pour gérer la sécurité dans un nuage, ce modèle définit les exigences essentielles à prendre en considération afin de garantir une sécurité acceptable auprès des fournisseurs des nuages :

### I.3.2.1.sécurité de contrôle :

Le tableau suivant mis au point par le groupe BURTON en juillet 2009, définit les droits de contrôle sur une architecture normale et les services de l'informatique en nuage :

	<b>Informatique</b>	<b>Hébergeur</b>	<b>IaaS</b>	<b>PaaS</b>	<b>SaaS</b>
<b>Données</b>	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Partage de contrôle entre l'entreprise et le fournisseur
<b>Applications</b>	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Partage de contrôle entre l'entreprise et le fournisseur	Contrôlé par le fournisseur de service
<b>Machine virtuelle</b>	Contrôlé par l'entreprise	Contrôlé par l'entreprise	Partage de contrôle entre l'entreprise et le fournisseur	Partage de contrôle entre l'entreprise et le fournisseur	Contrôlé par le fournisseur de service
<b>Serveur</b>	Contrôlé par l'entreprise	Partage de contrôle entre l'entreprise et le fournisseur	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service
<b>Stockage</b>	Contrôlé par l'entreprise	Partage de contrôle entre l'entreprise et le fournisseur	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service
<b>Réseau</b>	Partage de contrôle entre l'entreprise et le fournisseur	Partage de contrôle entre l'entreprise et le fournisseur	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service	Contrôlé par le fournisseur de service

**Tableau I.1 : modèle de contrôle dans différents services**



### **I.3.2.2.sécurité des données :**

Les données occupent l'espace le plus important car elles sont l'objectif de transmission entre client, fournisseur de nuage et les prestataires :

#### **a. La confidentialité des données :**

Le client doit connaître ce qui est préférable d'être en nuage et ce qui ne doit pas être, les données à l'intérieur de nuage doivent être contrôlées selon leurs niveaux de confidentialité alors qu'un tel service doit posséder la possibilité qu'un client décide de supprimer ces données ainsi que les transmissions des données doivent être contrôlées par des permissions et des autorisations.

#### **b. Le chiffrement des données :**

La cryptographie joue un rôle essentiel dans les relations et les transmissions entre le fournisseur et les clients, alors il est obligatoire de contrôler les clés de chiffrement et de déchiffrement ainsi que la transmission, logiquement le client doit prendre le contrôle total sur ses clés de chiffrement.

#### **c. Le mécanisme de journalisation :**

Dans le cadre de la sécurité informatique, le mécanisme de journalisation représente un support important de l'information, aussi dans l'informatique en nuage ce mécanisme est un concept important car il représente un support d'aide à la facturation d'optimisation des services et d'évaluation de leur niveau de sécurité.

#### **d. Lieu de stockage :**

Une mesure de sécurité est implémentée dans le nuage, elle sert d'informer uniquement le fournisseur de services de l'endroit de stockage des données, ce mécanisme a pour but d'empêcher les attaquants de violer les informations qu'ils veulent.

#### **e. L'intégrité des données :**

Le transfert des données entre le client et le nuage peut être dévoilé par plusieurs causes telles que le bruit de réseau, l'interaction d'un tiers (un attaquant)...etc.

Pour cela l'implémentation des mécanismes assurant l'intégrité des données est une exigence sur tous les niveaux.

### I.3.2.3.sécurité des infrastructures :

La sécurité dans le nuage ne s'intéresse pas seulement aux données des clients, les infrastructures sont aussi concernées par le processus de sécurité :

#### a. Sécurité physique :

Grâce à la virtualisation, le Cloud sert à naviguer le maximum par le concept de partage de ressources pour cela des mécanismes de protection sont obligatoires afin d'assurer la bonne utilisation et la plus sécuritaire, les mécanismes de protection des équipements doit être fait en transparence par rapport au client dont le fournisseur doit bien choisir où et comment déployer son architecture.

#### b. Audit :

L'audit et la journalisation ne doivent pas être appliqués sur les données seulement, l'application des mécanismes d'audit sur les infrastructures aide à l'évaluation des états de services et leur sécurité.

#### c. Teste de sécurité :

L'évaluation périodique de niveau de sécurité des équipements est un aspect important qui aide le fournisseur à choisir les meilleures solutions pour le déploiement des services ainsi que pour le choix de ce qui doit être mis à jour et ce qui est déjà à jour.

Le tableau suivant indique les différents mécanismes de sécurité et leur importance dans les différents types de services selon le type de Cloud :

	CLOUD PUBLIQUE			CLOUD PRIVE OU COMMUNAUTAIRE			CLOUD HYBRIDE		
	IaaS	PaaS	SaaS	IaaS	PaaS	SaaS	IaaS	PaaS	SaaS
IDENTIFICATION ET AUTHENTIFICATION	O	O	F	O	O	F	F	O	F
AUTORISATION	O	O	O	F	O	F	F	O	F
CONFIDENTIALITE	F	O	F	F	O	O	F	O	F
INTEGRITE	O	O	F	F	O	O	O	O	O
GOUVERNANCE	F	O	F	F	O	F	F	F	F
DISPONIBILITE	O	F	O	O	O	O	F	F	F

Remarque : 'O' pour obligatoire et 'F' pour facultatif

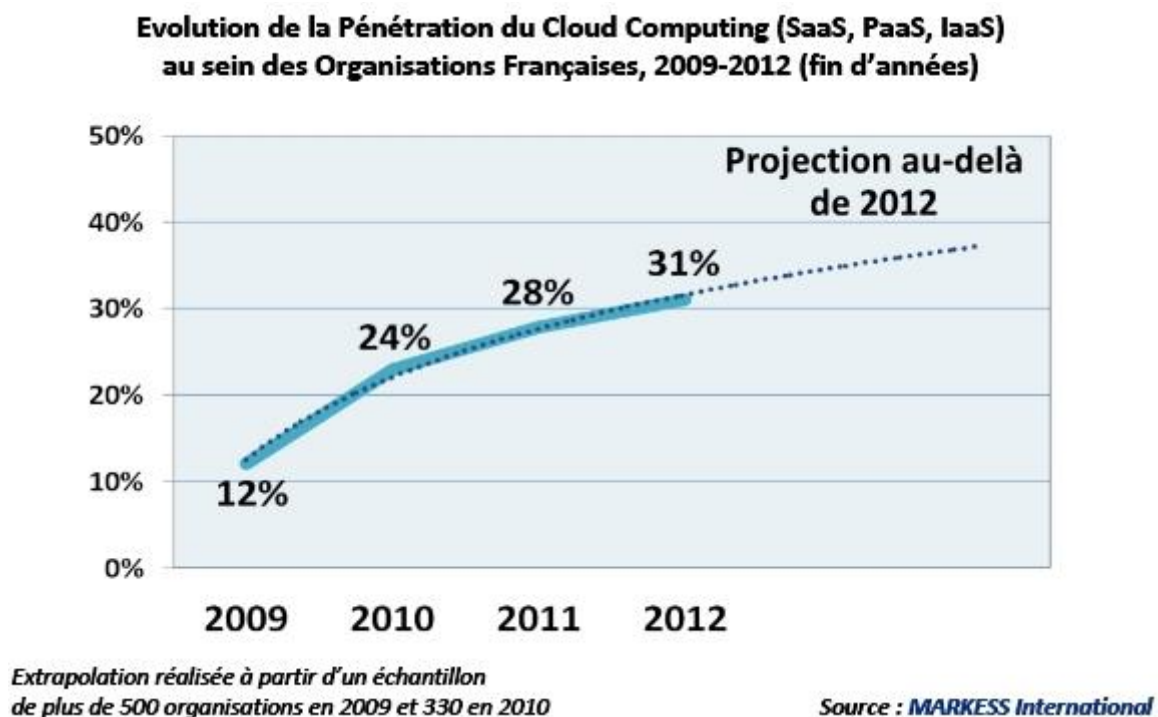
**Tableau I.2 :** modèle de sécurité pour Cloud Computing

## I.4. Le marché du Cloud Computing :

L'arrivée du concept de l'informatique en nuage en 2009 a fait une révolution dans la technologie de l'information, avec un croisement de 25% dans une seule année, les investissements dans ce domaine ont touché les 68 milliards de dollars en 2012, les 149 milliards de dollars en 2014 selon une étude faite par le cabinet Gartner sous le nom « **FORECAST : PUBLIC CLOUD SERVICES, WORLDWIDE AND REGIONS, INDUSTRY SECTORS, 2009-2014** ».

Le marché du Cloud affiche un taux de croissance de 30% par année, les experts de domaine prévoient un marché de 270 milliards de dollars d'ici 2020.

Le schéma suivant indique l'évolution d'utilisation des services de l'informatique en nuage en France depuis 2009 :



**Figure I.3:** l'évolution d'utilisation des services de l'informatique en nuage  
en France depuis 2009

Le marché du Cloud est contrôlé par des sociétés géantes telles que Google et Amazon qui représentent les acteurs principaux du Cloud, ces acteurs sont divisés en trois (03) catégories :

**a. Les éditeurs :**

Les éditeurs sont des acteurs qui serent à fournir une technologie permettant l'hébergement des services en nuage, leurs relations dans le Cloud sont faites avec les fournisseurs de services, cependant le marché des éditeurs connaît un état faible car la plus part des fournisseurs de services Cloud sont des éditeurs de leurs propres produits.

Actuellement, la société VMware fait partie des grands éditeurs des Cloud avec son produit publié en 2010 sous le nom « **vCloud Datacenter Service** », Microsoft est aussi un grand éditeur de Cloud avec son produit « **Windows Azure Appliance** » publié en 2011.

**b. Les fournisseurs :**

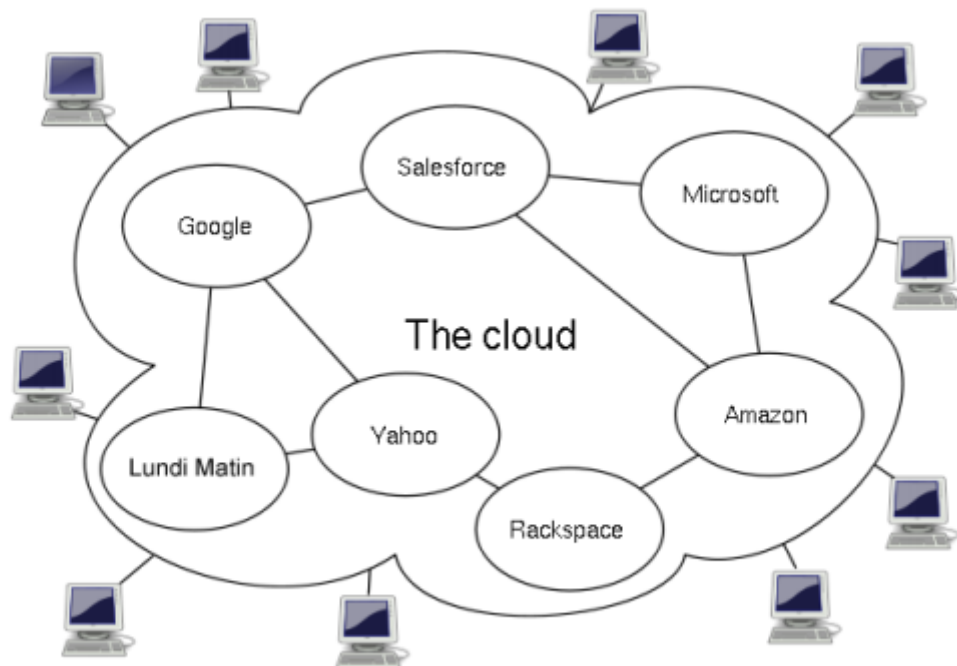
Ces acteurs sont les fournisseurs directes des services de l'informatique en nuage, ces fournisseurs sert à utiliser leur propres ressources afin d'assurer une infrastructure des services de Cloud.

La société Microsoft joue un rôle important dans le marché du Cloud avec ses produits de PaaS « **Windows Azure** », et de SaaS « **office 365** » ainsi que Google avec son SaaS « **Google App** » et son PaaS « **Google App Engine** » ainsi que la société Amazon et d'autres.

**c. Les pure PLAYERS :**

Sont des éditeurs connus par « **les joueurs purs** », ils fournissent des services plus professionnels que les éditeurs cités précédemment, notamment la société Sales forces qui représente le PURE PLAYER le plus grand des société avec ses services sous le nom de « **entreprise-facile.com** » qui sert à fournir des services aux entrepreneurs tels que les services de facturation, livraison, activité commerciale...etc. donc ces joueurs sont plus professionnels comme Mr. François Tonic dit dans son livre blanc « **Le Cloud Computing** » : « *lorsque l'on s'attaque frontalement à un géant comme SAP sur des progiciels, difficile d'imaginer un combat équitable. Sur de petits projets ou des projets précis dans une grande entreprise, le pure player à sa place. Mais l'éditeur traditionnel, quand il a vu la menace, a réagi soit en tissant des alliances avec le pure player, soit en commercialisant sa propre solution en ligne* ».

Le schéma suivant indique les principaux acteurs dans le domaine de l'info-nuagique :



**Figure I.4:** les principaux acteurs de Cloud Computing

### **I.5.conclusion :**

Le terme Cloud Computing ou bien Informatique en nuage représente un ensemble de services permettant de réaliser certaines fonctionnalités en commençant par le niveau le plus bas concernant l'infrastructure physiques tels que les serveurs de stockage et en arrivant au niveau le plus haut concernant les application et les logiciels.

Le Cloud Computing est alors la solution la plus économique en matière de cout de matériel et financière car il permet de représenter n'importe quel composant informatique (matériel ou logiciel) sous forme d'un service simple dans le web grâce au mécanisme de virtualisation.

Le Cloud et malgré les critiques qu'il a vu depuis sa création à cause des problème sécuritaire en générale il a fait un succès qui ne cesse de croitre d'un jour à l'autre grâce à ses services de qualités.

# Chapitre II

## QoS et enjeu énergétique

## **II.1.Introduction**

Le Cloud computing marque une nouvelle avancée vers l'infrastructure informatique dématérialisée. Le Cloud fournit des ressources informatiques, logicielles ou matérielles, accessibles à distance, en tant que service. L'adoption de ce modèle soulève un certain nombre de défis, notamment au sujet de la qualité de service (QoS) des services fournis

## **II.2.Qalité de Service**

### **II.2.1.Définition :**

La qualité de service (QDS) ou qualité of service (**QoS**) est la capacité à véhiculer dans de bonnes conditions un type de trafic donné, en termes de disponibilité, débit, délais de transmission, décide, taux de perte de paquets. La qualité de service est un concept de gestion qui a pour but d'optimiser les ressources d'un réseau (en management du système d'information) ou d'un processus (en logistique) et de garantir de bonnes performances aux applications critiques pour l'organisation. La qualité de service permet d'offrir aux utilisateurs des débits et des temps de réponse différenciés par applications (ou activités) suivant les protocoles mis en œuvre au niveau de la structure. Elle permet ainsi aux fournisseurs de services (départements réseaux des entreprises, opérateurs) de s'engager formellement auprès de leurs clients sur les caractéristiques de transport des données applicatives sur leurs infrastructures.

**[B1]**

### II.3.Catégories

Les fournisseurs de services Cloud peuvent proposer différentes solutions aux entreprises voulant se tourner vers le Cloud Computing. On peut facilement comprendre que chaque solution propose différents niveaux de performance en termes de fonctionnalités, de temps de réponse ou de précision. C'est alors aux entreprises de comparer celles-ci, de comprendre comment leurs applications peuvent fonctionner sur chacune d'entre elles et enfin de trouver quelle solution est la mieux adaptée à leurs besoins.[B2] [B3]

Le contrat liant l'utilisateur d'un Cloud à son fournisseur de service est appelé **SLA** (Service Level Agreement). Un **SLA** décrit le niveau de prestation, conditions d'utilisation et d'exécution, les responsabilités incombant à chaque partie (utilisateur/fournisseur), puis des informations plus précises de performance appelées **SLO** (Service Level Objective) que le fournisseur de service se promet de respecter pour garantir une certaine qualité de service à son utilisateur. L'expression de la Qualité de Service (**QoS**) dans un environnement de Cloud Computing s'exprime avec des paramètres de plus ou moins haut niveau.

Les catégories de QoS au sein d'un Cloud sont classés comme suit :

- ❖ Performance (Performance).
- ❖ Sûreté de fonctionnement (Dependability).
- ❖ Sécurité Données (Security Data).
- ❖ Coûts (Cost)

Tout d'abord, deux types des paramètres sont à distinguer :

- **Fonctionnel**

Une exigence fonctionnelle permet de caractériser un comportement d'un système. Ces comportements peuvent s'exprimer sous la forme de services, de tâches ou bien de fonctions que le système est censé fournir. Il est important de pouvoir distinguer les fonctionnalités de base, qui peuvent être communes à plusieurs systèmes, et les fonctionnalités les plus spécifiques de chacun qui permettent de les différencier mais aussi de les mettre en concurrence.



- **Non-Fonctionnelle**

Une exigence non-fonctionnelle est une exigence qui spécifie des critères qui peuvent être utilisés pour juger le fonctionnement d'un système, plutôt que ses comportements spécifiques. Les exigences non fonctionnelles peuvent aussi être appelées « qualités », « contraintes », « attributs de qualité », « objectifs de qualité » et « exigences non-comportementales ».

Dans le cadre du Cloud Computing, la problématique la plus importante aux yeux de l'utilisateur est de pouvoir sélectionner le service qui lui convient le mieux en fonction de ses besoins.

Du côté des fournisseurs de service, il est important de pouvoir démontrer que leurs offres de services sont intéressantes et de pouvoir assurer le bon fonctionnement de leur plateforme. Pour cela, il est nécessaire de définir des paramètres globaux de **QoS** afin de pouvoir comparer, analyser et classer le niveau de **QoS** des différents fournisseurs de services.

### **II.3.1. Performance (Performance) :**

#### **II.3.1.1.Temps d'exécution (Execution Time) :**

Le temps d'exécution dépend de la complexité de la requête à exécuter (en nombre d'instructions) et de la capacité de la machine virtuelle dans laquelle elle est traitée.[B4]

#### **II.3.1.2.Latence Débit (Latency Throughput) :**

La latence représente le temps nécessaire à la requête utilisateur pour arriver jusqu'au service concerné. Cela dépend uniquement de la performance et de l'état du réseau au moment où la requête utilisateur est envoyée au fournisseur de service.

#### **II.3.1.3.Temps de réponse (Reponse Time) :**

L'efficacité d'un service peut être mesurée en terme de temps de réponse, noté « **Trep** », ou autrement dit le temps nécessaire à la mise à disposition d'un service à un utilisateur, ou également le temps écoulé entre l'envoi d'une requête utilisateur et le moment où celui-ci reçoit la réponse du service. Le temps de réponse dépend de

plusieurs sous-facteurs comme le temps de réponse moyen, le temps de réponse maximum garanti par le fournisseur de ce service ou bien le pourcentage de chance que ce temps soit dépassé (échec du temps de réponse).

### **II.3.2. Sureté de fonctionnement (Dependability) :**

#### **II.3.2.1. Précision (Accuracy) :**

La précision d'un service est définie comme une fonction de deux paramètres :

- ❖ La fréquence de précision, notée  $\phi_s$ .
- ❖ La valeur de pertinence, notée  $TS$ .

Elle mesure le degré de proximité des résultats par rapports aux valeurs attendues par l'utilisateur, et est notée  $A_c = f(\phi_s, TS)$ .

#### **II.3.2.2. Fiabilité (Reliability) :**

La fiabilité, représente la capacité d'un service à remplir ses fonctions requises sous certaines conditions et pendant un intervalle de temps donné. La mesure globale d'un service est lié au nombre de fois (par jour, semaine, mois ou année) que le service n'est plus capable de remplir son travail.

La fiabilité reflète la capacité d'un service à fonctionner sans aucune perte pendant un temps donné sous des contraintes données. Cette caractéristique est donc basée sur la « durée de fonctionnement avant une défaillance » ou « durée de fonctionnement entre pannes du service », ainsi que sur la base de connaissance des défaillances des utilisateurs. [B5]

#### **II.3.2.3. Disponibilité (Availability) :**

La disponibilité, est le pourcentage de temps pendant lequel un utilisateur peut accéder à un service. Cette disponibilité est la probabilité que le système soit en état de fonctionnement et est donc lié à son taux de panne et densité de panne.

#### **II.3.2.4.Capacité (Capacity) :**

La capacité est définie par la limite maximale du nombre de requêtes (par seconde) qu'un service est capable de traiter. En termes de calcul, cela dépend de la performance des machines virtuelles utilisées par ce service et des machines physiques sur lesquelles tournent ces machines virtuelles.

#### **II.3.2.5.Extensibilité (Scalability) :**

Au sein d'une plateforme Cloud, il est difficile d'estimer le nombre d'utilisateurs à un instant donné car il n'est ni prédéfini, ni constant. Sans connaître ces informations, il est donc impossible de prévoir les ressources nécessaires au fonctionnement d'un service. L'arrivée d'utilisateurs et leur intensité d'utilisation peuvent varier très rapidement et brusquement d'un instant à l'autre. Il peut donc être très difficile de satisfaire tous ces utilisateurs en même temps.

L'extensibilité [B6] (aussi appelé « passage à l'échelle ») représente la capacité d'augmenter la capacité de calcul et donc la capacité du système à traiter un grand nombre de requêtes utilisateurs dans un intervalle de temps donné. Elle est directement liée à la performance et s'évalue en faisant augmenter le nombre d'utilisateurs (i.e. le nombre de requêtes aux services par secondes) tout en vérifiant si les caractéristiques de performance décrites dans le contrat **SLA** sont respectées.

L'extensibilité est donc une propriété plus que souhaitable pour un fournisseur de services Clouds.

Cela permet de garantir la stabilité des services en cas de forte charge (croissance du nombre de services demandés ou du nombre de requêtes des utilisateurs) de façon fluide et transparente pour les utilisateurs.

Cela dit, il est dans l'intérêt du fournisseur de service de ne pas sur-dimensionner la capacité de calcul de ses machines et également de ne pas les utiliser à 100% de leur capacité pour des raisons évidentes de consommation d'énergie, et ainsi rester dans une zone d'extensibilité acceptable.

### **II.3.2.6.Réactivité (Reactivity) :**

La réactivité ou (élasticité), est la capacité d'un service à passer à l'échelle durant un pic de fréquentation. Cela est défini par deux variables : le temps nécessaire pour augmenter ou réduire le service et la variation de capacité maximum du service.

La capacité de traitement maximum du service est assimilée au nombre d'unités de calcul et à leurs capacités maximum respectives pouvant leur être allouées durant le pique de fréquentation.

En d'autres termes, c'est la rapidité d'un système à augmenter son degré de performance tout en restant dans sa zone d'extensibilité acceptable. Si un système réagit trop lentement à un pic d'utilisation alors les besoins des utilisateurs ne seront plus respectés et le système sera pris en défaut.

Un système qui réagit vite arrivera à traiter les besoins des utilisateurs en temps voulu, mais il est également important de ne pas surestimer la reconfiguration à faire afin de ne pas se retrouver en zone d'extensibilité surdimensionnée.

### **II.3.2.7.Dynamisme (Dynamism) :**

Le Dynamisme ou « Capacité latente », noté Dyn, représente la capacité de calcul disponible sans allumer de nouvelles machines physiques. Cela amène le fournisseur de Cloud à ne pas consolider de manière abusive les machines virtuelles sur les machines physiques, et ainsi à garder une certaine capacité disponible en cas de pic de charge.

Le dynamisme peut être évalué comme la moyenne de la capacité de CPU libre de toutes les machines physiques en cours d'utilisation. Si les machines physiques ont le **DVFS** activé, alors le dynamisme est calculé en fonction de la capacité maximale du processeur et de la capacité actuelle utilisée.

### **II.3.2.8.Robustesse (Robustness) :**

La robustesse, notée Rob, peut être interprétée comme la probabilité d'un service d'être affecté par une défaillance d'un composant du Cloud. Du point de vue du fournisseur de services, cela revient à comptabiliser en moyenne le nombre de services

pouvant être potentiellement touchés par une défaillance d'une machine physique. Plus le nombre de services mis en défaut est faible, plus le fournisseur de Cloud peut assurer à ses utilisateurs un haut niveau de robustesse. La robustesse peut simplement être représentée comme le nombre moyen de services en cours d'utilisation sur les machines physiques.

#### **II.3.2.9.Stabilité (Stability) :**

La stabilité d'un service est définie par les variations de performance de celui-ci. Par exemple, pour un service de stockage, c'est la variance du temps moyen de lecture et d'écriture. Pour un service de calcul, la stabilité est définie par le calcul de l'éloignement par rapport à la performance spécifiée dans le contrat **SLA**.

#### **II.3.2.10.Tolérance aux fautes (Fault Tolerance) :**

La résistance [B7] aux fautes d'un service,représente sa capacité à rester accessible et en bon état de fonctionnement lorsque des anomalies interviennent. Ces anomalies peuvent être :

- ✓ Taux d'erreur d'une machine physique, noté  $T_m$
- ✓ Taux d'erreur sur le réseau, noté  $T_n$
- ✓ Taux d'erreur logiciel : paramètres d'entrée invalides, incomplets ou contradictoires, noté  $T_l$ .

Un service Cloud doit être en mesure d'assurer son bon fonctionnement face à ces différents cas, ce qui définit son niveau de tolérance aux fautes.

#### **II.3.2.11.Agilité (Agility) :**

Sous le terme agilité, deux différents types de caractéristiques peuvent être définis.

Le premier correspond à la possibilité de variation de la disponibilité des machines physiques mises à disposition des utilisateurs du Cloud

le second correspond à la facilité avec laquelle un fournisseur de services accepte ou non de renégocier les termes du contrat le liant à l'utilisateur.

Dans le première cas , le principal avantage d'un environnement Cloud est qu'il intègre un degré d'agilité des services qui le composent. C'est à dire que l'environnement considéré peut rapidement évoluer et se développer sans engendrer des couts insurmontables. L'agilité est mesurée comme un taux de variation de plusieurs métriques, qui démontrent la rapidité avec laquelle le système peut intégrer de nouvelles fonctionnalités. Quand on parle d'agilité de services Cloud on veut pouvoir comprendre à quel point le service auquel on s'intéresse est élastique, portable, adaptable et flexible. Dans ce cas, les variations possibles du système, définissant son agilité sont les suivantes :

- ❖ Nombre de ressources allouées à un utilisateur.
- ❖ Flexibilité de la capacité (calcul, mémoire, ...) des machines virtuelles.
- Utilisation d'un service supplémentaire.

Dans le deuxième cas (redéfinition des valeurs limites définies au départ dans le SLA), les paramètres à prendre en compte sont les suivants :

- ❖ Augmentation du temps de location des machines virtuelles.
- ❖ Augmentation du nombre de machines mises à disposition de l'utilisateur.
- ❖ Augmentation de la capacité (calcul, mémoire, ...) maximum autorisée.

### **II.3.2.12.Durabilité (Sustainability) :**

La durabilité peut être définie en terme de cycle de vie du service ou bien en terme d'impact environnemental des services utilisés. On peut donc diviser ce paramètre en deux sous-attributs : durabilité de service et durabilité environnementale.

- ✓ La durabilité de service peut être définie par le nombre de composants d'un service réutilisables sans aucun changement indépendamment de l'évolution des exigences de l'utilisateur.

En d'autres termes, on peut dire qu'un service est très durable s'il possède plus de caractéristiques, utilisables par l'utilisateur, que ce qui lui est nécessaire.

Dans ce cas, si les exigences de l'utilisateur évoluent (augmentation du nombre de caractéristiques demandées) alors il pourra tout de même continuer à utiliser ce même service, simplement en utilisant plus de caractéristiques de celui-ci. Ceci évite à l'utilisateur de chercher un autre service correspondant à ses critères.

- ✓ La durabilité d'un service est définie par le rapport du « nombre de caractéristiques fournit par le Service » sur le « nombre de caractéristiques demandés par l'utilisateur ».

La durabilité environnementale peut être appréciée comme le bilan carbone du service. Le calcul du bilan carbone est très complexe et dépend de beaucoup de paramètres, c'est pourquoi on préfère souvent s'attacher au calcul du **PUE**.

#### **II.3.2.13.Assurance (Insurance) :**

Le terme d'assurance dans le cadre du Cloud Computing donne une information sur l'engagement qu'il prend à garantir le bon fonctionnement de ces services. Dans le cas contraire, des procédures de dédommagement peuvent être envisagées comme dans tout contrat d'assurance. Ce paramètre indique donc aussi la probabilité qu'un fournisseur de services Cloud respecte les données du contrat **SLA** et cela n'est pas sans effet sur le point de vue que les entreprises peuvent se faire du Cloud Computing.

En effet, les entreprises cherchent à développer leurs activités et à fournir de meilleurs services à leurs clients. Les paramètres de fiabilité, de résistance et de stabilité sont considérés avec une grande importance par les entreprises avant de basculer de leurs applications traditionnelles vers des services de type Cloud.

#### **II.3.2.14.Convivialité (Usability) :**

Pour une utilisation rapide de services Cloud, la convivialité joue un rôle très important. Plus les services Cloud seront faciles à prendre en main plus une entreprise aura tendance à passer le cap du Cloud Computing. La convivialité d'un service Cloud intègre plusieurs paramètres comme la facilité d'accès, d'installation et d'utilisation.[B8]

#### **II.3.2.15.Personnalisation (Customization) :**

La personnalisation représente le pouvoir des services de Cloud à s'adapter aux différents types d'utilisateurs. En d'autres termes, chaque utilisateur doit avoir la possibilité de choisir et de sauvegarder ses propres réglages (apparence, logo, police d'écriture,...) et fichiers de configuration. Ainsi le service utilisé s'adapte à chaque utilisateur.

### **II.3.2.16. Mise à jour automatique (Automatic Update) :**

Un atout majeur d'un fournisseur de service **SaaS** est de pouvoir mettre à jour ses services régulièrement. La plateforme doit donc fonctionner avec les dernières innovations en terme de service afin d'en faire directement profiter ses utilisateurs.

Du point de vue utilisateur, un des atouts majeurs des mises à jour automatiques, sans compter qu'elles se font de manière totalement transparente à ses yeux, est que s'il travaille en collaboration avec une autre personne utilisant cette même application, alors il n'a plus à se soucier des problèmes de compatibilité qui sont les premières sources d'erreurs dans de telles situations.

### **II.3.3.Sécurité Données (Security Data) :**

Bien que la performance des services Cloud soit un paramètre très important, la sécurité mise en place pour l'accès, l'utilisation des services ainsi que le stockage des données est également une des préoccupations majeures des utilisateurs.

La protection des données est un aspect incontournable pour toutes les entreprises. Faire héberger ses données « à l'extérieur » est un point critique ce qui demande aux fournisseurs de services Cloud de mettre en place des politiques de sécurité très performantes. La confidentialité des données est tout autant indispensable : l'utilisation de techniques de preuve de stockage permet de vérifier que le Cloud ne subit pas d'attaque de type « déni de service » et que toutes les données des clients sont intactes. Les données dynamiques peuvent également être vérifiées avec une approche similaire.

Ces techniques fonctionnent avec un très faible temps de calcul tout en transférant peu de données entre les clients et le serveur.[B9]

#### **II.3.3.1.Authentification (Authentication) :**

Les moyens traditionnels d'authentification ne sont plus d'actualité dans une infrastructure de Cloud et c'est pourquoi de plus en plus de fournisseurs Cloud sont à la recherche d'un service d'authentification performant, appelé Authentification-As-A-Service (**AAAS**). Cela afin d'accroître la sécurité et gérer l'authentification des utilisateurs plus facilement.

L'authentification forte peut aujourd'hui être assurée partout où un mot de passe



est utilisé par l'utilisation des normes industrielle tels que **RADIUS** (Remote Authentication Dial In User Service) et **SAML** (Security Assertion Markup Language) et la disponibilité d'**API** pour d'autres applications.[B10] L'utilisation de jeton d'authentification, méthode appelée « Cloud Token » permet aux utilisateurs de ne pas perdre leur jeton lorsqu'ils décident de migrer de plateforme en plateforme. De plus une automatisation de la gestion d'authentification permet de réduire considérablement le cout de gestion et d'administration.

### **II.3.3.2.Autorisation (Authorization) :**

Les moyennes et grandes entreprises ont généralement des besoins spécifiques de caractéristiques d'autorisation d'accès aux données pour leurs utilisateurs de Clouds. C'est à dire l'attribution de privilèges, ou droits, aux utilisateurs en fonction de leurs rôles et de leurs besoins. Dans certains cas, une application d'entreprise peut exiger un « contrôle d'accès à base de rôles » (**RBAC**),[B11] dans laquelle les autorisations sont structurées de manière à répondre aux exigences des rôles fonctionnels de l'entreprise.

A ce jour, la mise en place des autorisations au sein d'un service Cloud et la gestion des capacités sont faibles, et ne peuvent pas être gérées avec une granularité satisfaisante. La plupart des fournisseurs de services Clouds prennent au moins en compte deux rôles (privilèges) : administrateur et utilisateur. Cela permet d'attribuer des droits d'administration sur le service afin de pouvoir gérer les profils utilisateurs, les politiques d'accès au service ou d'autoriser ou non les connexions au service suivant leur provenance.

### **II.3.3.3.Intégrité (Integrity) :**

La capacité de conserver l'intégrité des données des utilisateurs est d'assurer que le contenu de leurs données ne soit pas modifié à leur insu. Une manière de vérifier l'intégrité d'un contenu est de comparer l'état actuel des données par rapport à un état connu dans le passé sans qu'il y ait eu d'intervention de l'utilisateur entre les deux moments.

### **II.3.3.4. Confidentialité (Confidentiality) :**

Plusieurs types de problèmes tels que les bogues logiciels, erreurs venant de l'opérateur et attaques externes peuvent compromettre la confidentialité des données applicatives des Clouds, et de ce fait les rendre vulnérables face aux actions malveillantes de certains utilisateurs dont l'accès au Cloud aurait dû être interdit. Des techniques de protection contre les attaques visant à détruire la confidentialité des données peuvent également aider les utilisateurs à vérifier la cohérence et la mise à jour de leurs données. [B12]

### **II.3.3.5. Responsabilisation (Accountability) :**

Le terme Accountability [B12] peut être associé au terme monitoring et être traduit par « Responsabilisation », « Vérification » ou « Contrôle »

Au sein d'un Cloud lorsque l'on parle d'accountability, il est d'usage de parler de « chaîne de vérification » ou de « chaîne de contrôle », de différents paramètres, que le fournisseur de service met en place au sein de son Cloud. Les fournisseurs de services implémentent donc des mécanismes de vérification, essentiellement destinés à assurer un niveau de sécurité sur les données des utilisateurs. Cela dans l'optique de pouvoir leur fournir un suivi et un contrôle de leurs données stockées dans le Cloud.

Une chaîne de contrôle d'un fournisseur de Cloud permet de garantir l'accessibilité aux outils suivants :

1-Des outils permettant de donner aux utilisateurs un contrôle et une vision de la façon dont leurs données sont utilisés, l'assurance que leurs données soient traitées et protégées de la manière dont ils le souhaitent.

2-Des outils permettant aux utilisateurs de faire des choix sur la façon dont les fournisseurs de services Cloud peuvent utiliser et protéger leurs données, d'être informés sur les risques, les conséquences et la mise en place de ces choix.

3-Des outils pour vérifier que les attentes en terme de sécurité des utilisateurs soient respectées, et qu'il n'y ait pas d'entrave aux règles définies dans les **SLA**.

4-Des outils d'explication et d'aide pour informer l'utilisateur sur la façon dont est effectuée la surveillance et le contrôle de leurs données, par le fournisseur de service d'un point de vue éthique.

### **II.3.3.6.Traçabilité (Traceability) :**

D'une manière générale la traçabilité est définie comme suit : « La traçabilité désigne la situation où l'on dispose de l'information nécessaire et suffisante pour connaître (éventuellement de façon rétrospective) la composition d'un produit tout au long de sa chaîne de production et de distribution. »

Au sein d'un Cloud, la traçabilité de l'utilisation des services permet à l'utilisateur de savoir précisément comment le service est utilisé. Toute information relative aux interventions effectuées concernant de près ou de loin le ou les services utilisés doivent pouvoir être mise à la disposition de chaque utilisateur. En effet, l'impossibilité de connaître l'état de fonctionnement, ou la localisation des données est un réel frein à l'utilisation de ces services Clouds. Il est donc important d'informer les utilisateurs, à travers un journal de logs (traces écrites), quels types d'actions automatisées ou humaines ont été effectuées, et où sont stockées leurs données.

Une plateforme de traçabilité permet de répondre à ces problèmes de non-transparence, et donc de sauvegarder une chaîne d'événements indiquant par exemple les opérations humaines, les transferts de fichiers, et l'activité des processus automatiques ainsi que des informations provenant des systèmes connexes tels que les systèmes d'authentification et de gestion des équipements.

### **II.3.3.7.Cryptage (Encryption) :**

Le cryptage [B13] est un procédé utilisé pour protéger le transit ou le stockage d'informations. Il s'agit de la conversion des données en cryptogrammes, qui ainsi ne peuvent être lus que par les personnes propriétaires de ces données ou autorisées à en lire le contenu. Le chiffrement est utilisé pour protéger les informations sensibles stockées et utilisées dans les réseaux, appareils mobiles ou sans fil.

Dans un Cloud, les algorithmes de chiffrement sont utilisés pour protéger les données sortantes, afin que les informations ne soient pas vulnérables une fois qu'elles sont à l'extérieur de l'entreprise de laquelle elles appartiennent. Le chiffrement est couramment utilisé pour s'assurer de répondre aux réglementations des entreprises.

### **II.3.3.8.Isolation (Isolation) :**

Dans les Clouds, un défi est de permettre aux utilisateurs authentifiés de consulter leurs propres données depuis l'extérieur de ce Cloud tout en empêchant un utilisateur malveillant d'exploiter une faille du prestataire de services lui permettant d'accéder aux données des utilisateurs et rendant par la même occasion l'ensemble du système vulnérable. Pour éviter ce problème, la mise en place d'un service de journalisation (log) sécurisé peut aider à isoler les données des différents utilisateurs.

### **II.3.3.9.Cycle de vie des données (Data life cycle) :**

Oracle définit le cycle de vie des données [B14] par la fréquence d'accès à celles-ci. Plus le temps passe, plus la fréquence d'accès aux données diminue, menant finalement à un archivage de celles-ci. Le cycle de vie comprend trois états : actif, moins actif et état historique ou archivé. La Cloud Security Alliance a proposé un concept de cycle de vie de sécurité des données, qui résume dernière en six étapes :

- Production.
- Conservation.
- Utilisation.
- Partage.
- Archivage.
- Destruction.

### **II.3.3.10.Non-Répudiation (Non-Repudiation) :**

La non-répudiation implique le fait que le contrat ne peut être remis en cause par aucune des deux parties (utilisateur/fournisseur).

## **II.3.4. Coûts (Cost) :**

### **II.3.4.1.Coût de service (Service Cost) :**

La première question qui surgit, notamment au sein d'une entreprise, avant de se tourner vers le Cloud Computing est : « cela est-il rentable ou non ». La rentabilité et le coût sont clairement des attributs qui ont une très grande importance aux yeux des utilisateurs (particulier ou entreprise) de Cloud. Le coût tend à être un paramètre facilement quantifiable, mais il est important de pouvoir l'exprimer avec des caractéristiques pertinentes par rapport à l'organisation à laquelle on s'adresse.

### **II.3.4.2.Coût d'énergie (Energy Cost) :**

Le coût d'énergie d'un équipement, exprimé en kWh, représente l'énergie nécessaire pour le faire fonctionner sur une période de temps donnée. Il est en fonction de la puissance (en Watt) délivrée par cet équipement (machine physique par exemple) et de la durée d'utilisation (en Heure). Concernant la puissance délivrée par une machine physique, plusieurs paramètres rentrent en jeux, notamment le taux d'utilisation du CPU. L'utilisation du modèle affine, bien que assez simpliste, a fait ses preuves depuis longtemps et est utilisé dans de nombreux travaux de recherche.

### **II.3.4.3.Empreinte carbone (Carbon Cost) :**

L'empreinte Carbone [B15] représente l'impact que peut avoir un système sur l'environnement en tenant compte du volume de dioxyde de carbone (CO<sub>2</sub>) émis par combustion d'énergies fossiles. Dans les Clouds, les valeurs et les coûts qui lui sont associés peuvent varier en fonction des fournisseurs de services, car de nombreux paramètres rentrent en jeux. Selon la « Open Data Center Alliance » , ces variations peuvent être dues à :

- ❖ L'efficacité énergétique du matériel utilisé
- ❖ La configuration et les paramètres du matériel et des logiciels utilisés
- ❖ Le degré de virtualisation et de l'efficacité de sa gestion
- ❖ Les conditions environnementales (par exemple, climat nordique vs climat méditerranéen)
- ❖ L'efficacité de l'infrastructure du Data Center et de son hébergement (i.e., efficacité énergétique ou **PUE**)
- ❖ Source d'électricité utilisée : charbon, nucléaire, générateurs locaux, etc...
- ❖ La « compensation carbone » Carbon offsetting (engagement à diminuer les émissions de carbone)
- ❖ Les dispositions nationales ou régionales régulant les taxes liées aux émissions de carbone.

## **II.4.Evaluation de l'empreinte environnementale :**

Il existe différentes méthodes pour évaluer l'empreinte environnementale du cloud computing chacune associée à une métrique, nous présentons ici les plus utilisées :

### II.4.1. Consommation électrique :

Le PUE (Power Usage Effectiveness) est une métrique largement utilisée pour calculer la consommation électrique, il s'agit du rapport entre la consommation énergétique totale de l'infrastructure et la consommation des équipements informatiques (processeurs, mémoires, stockage) :

$$PUE = \frac{\text{Consommation totale}}{\text{Consommation informatique}}$$

Cette méthode largement utilisée ne doit cependant pas servir à comparer deux data center car la façon de prendre ces mesures n'est pas standardisée et peut varier selon ce qui est pris en compte, en particulier pour les bâtiments qui ne sont pas dédiés aux centres de données (ressource énergétiques partagées). Elle sert uniquement à prendre deux mesures dans le temps pour estimer l'impact d'une modification.

### II.4.2. Coefficient d'énergie renouvelable :

Pour déterminer l'impact environnemental d'un centre de données, on utilise le coefficient d'énergies renouvelables :

$$CER = \frac{\text{Quantité d'énergie renouvelable}}{\text{Quantité totale d'énergie utilisée}}$$

C'est cette métrique qui est utilisée depuis 2012 pour estimer l'impact environnemental des grilles de calcul.

## II.5. Techniques pour minimiser l'empreinte environnementale :

Il a été montré que le cloud computing avait une empreinte environnementale importante. Il existe cependant de nombreuses techniques permettant de la réduire, ces techniques concernent l'aspect logiciel ainsi que l'aspect matériel.

### **I.5.1.Matériel :**

#### **II.5.1.1.Refroidissement**

Le refroidissement des centres de données représente plus de 45 % de leur consommation d'énergie. Le refroidissement à air est actuellement le plus répandu, cependant cette technique est très peu efficace car le refroidissement par air conditionné a tendance à trop refroidir la pièce et pas assez les composants. De plus l'air projeté de bas en haut dans une armoire refroidira efficacement le bas et moins efficacement le haut. une autre méthode existe, le Watercooling « refroidissement à eau ». Ainsi les composants sont directement refroidis par des liquides ce qui réduire la consommation énergétique consacrée au refroidissement de 53 % à 44 % par rapport au refroidissement à air.

Cependant, le refroidissement à eau pose un problème, en effet une fois l'eau chauffée par le centre de données, celle-ci ne peut être reversée dans la nature, elle doit d'abord être refroidie, cela ne fait que déplacer le problème. Une solution est de trouver une utilité à cette source de chaleur. Par exemple, IBM a produit un supercalculateur dont l'eau du refroidissement est réutilisée pour chauffer le bâtiment. Cette solution permet d'avoir une capacité de calcul de 6 Teraflops pour une consommation électrique équivalente à 450 Mégaflops. Ce système permet de réduire de 40% la consommation électrique du refroidissement par rapport à un refroidissement à eau, l'empreinte environnementale du bâtiment est réduite de 80 %.

Le problème du refroidissement peut être envisagé dès la construction des bâtiments qui abritent les serveurs. En effet, à l'origine les data center ont été construits comme des bâtiments résidentiels classiques. Ils ne sont pas faits pour dissiper la chaleur mais pour la conserver à l'intérieur. Il faudrait qu'en hiver les murs laissent entrer le froid extérieur, et qu'en été ils ne laissent pas entrer la chaleur.

#### **II.5.1.2.La consommation des composants**

Au niveau des processeurs, plusieurs techniques existent pour réduire leur consommation en énergie :

- ❖ Adapter les techniques de réductions de consommation des processeurs mobiles aux processeurs des serveurs.

- ❖ Utiliser le clock gating qui permet d'éteindre l'horloge d'un bloc logique quand il n'est pas actif. En effet la consommation d'un centre de données qui ne travaille pas reste égale à 60 % de sa consommation maximal.
- ❖ Utiliser le Dynamic voltage scaling qui permet d'augmenter ou réduire le voltage d'un composant selon les circonstances.

Pour le stockage des données, l'utilisation de **SSD** permet de réduire le besoin en refroidissement ainsi que les déchets électroniques grâce à l'absence de pièces mécaniques réduisant les pannes.

### II.5.2.Virtualisation :

Les outils pouvant permettre de réduire la consommation d'énergie d'une ou d'un ensemble de machines physiques sont des solutions qui s'appliquent à différents niveaux (machine physique, machine virtuelle, réseau, ...). Toutes ces solutions ont pour but de réduire globalement la consommation d'un data center. En effet, chaque amélioration (diminution de la consommation) de chacune des machines physiques composant un data center amène au final à une réduction de la consommation globale non négligeable étant donné le grand nombre de machines physiques qui compose un data center.

La première solution concerne **l'extinction de machines physiques** lorsque celles-ci sont trop peu utilisées ou pas du tout utilisées, rendant leur rendement très médiocre. Bien que très simple et intuitive, cette méthode requiert un processus de déplacement des machines virtuelles s'exécutant sur la ou les machines physiques concernées afin de pouvoir les éteindre sans détruire les services qu'elles hébergeaient. Le procédé inverse : réallumage d'une ou de plusieurs machines physiques est évidemment également utilisé si toutes les machines physiques déjà allumées sont surchargées et que de nouvelles machines virtuelles doivent être allouées.

Il est souvent nécessaire de pouvoir déplacer des machines virtuelles entre différentes machines physiques. Ce processus est appelé « **migrations de machines virtuelles** », il permet de déplacer une machine virtuelle, et tout son environnement (données stockées en mémoire), d'une machine physique à une autre en fonction des



terminaisons d'exécutions d'autres machines virtuelles. Ces transferts ne sont pas sans frais pour la consommation d'énergie car ils nécessitent un certain temps durant lequel les machines virtuelles en migration consomment des ressources sur les machines physiques source et destination. Il est donc très important de le faire à bon escient pour éviter des sur coûts inutiles. Le fait de déplacer des machines virtuelles permet :

- Libérer de la place sur une ou des machines physiques et ainsi les éteindre complètement lorsque celles-ci sont vides
- A l'inverse, rassembler le plus possible de machines virtuelles sur une machine physique (technique de consolidation) pour utiliser au maximum ses capacités et ainsi améliorer son rendement.

Enfin, le **DVFS (Dynamic Voltage and Frequency Scaling)** permet de changer dynamiquement la fréquence des CPUs des machines physiques en fonction de leurs taux d'utilisation. L'idée clé de la mise en œuvre du **DVFS** est de réduire la fréquence (GHz) du CPU et de la tension (V) pendant des périodes de faible utilisation du processeur. Une réduction de la fréquence et de la tension engendre inévitablement une diminution de la puissance délivrée par CPU en raison de la nature des circuits **CMOS** (Complementary Metal Oxide Semiconductor) aujourd'hui. En particulier, la puissance dissipée par les circuits **CMOS** est composée de deux parties :

statique et dynamique. La partie statique est entre autre due aux courants de fuite du composant, dont la diminution est possible lors de la conception des circuits **CMOS**. D'autre part, la partie dominante, dite dynamique, due à la charge et la décharge des composants du **CMOS**. Cette consommation dynamique est communément approximée par l'équation suivante :

$$P = C \times f \times V^2$$

P-state	Fréquence	Voltage	Puissance
P0	1.6 GHZ	1.484 V	25 Watts
P1	1.4 GHZ	1.420 V	17 Watts
P2	1.2 GHZ	1.276 V	13 Watts
P3	1.0 GHZ	1.164 V	10 Watts
P4	800 GHZ	1.036 V	8 Watts
P5	600 GHZ	0.956 V	6 Watts

**Table II.1** – Présentation des P-states et leur couple fréquence/voltage ainsi que leur puissance associée, Pour le processeur Intel Pentium M 1.6GHz .

Où  $C$  est la capacité de commutation « switching factor »,  $f$  la fréquence de commutation, et  $V$  la tension d'alimentation.

Selon cette équation, une réduction linéaire de tension  $V$  implique une réduction quadratique de puissance fournie par le composant. Cependant, une diminution de la tension  $V$  implique également une réduction de la vitesse de commutation des transistors composant le **CMOS**, ce qui conduit inévitablement à une baisse de la fréquence maximale théoriquement possible du processeur. Par conséquent, pour assurer le bon fonctionnement du CPU, sa fréquence doit être abaissée proportionnellement à la baisse de tension appliquée. C'est ainsi que l'utilisation du **DVFS** est associée à des couples, appelés P-states et C-states : les P-states sont associés à des états actifs du CPU (exemple en Tableau précédent), alors que les C-states sont associés à des états inactifs (« Idle » en anglais) du CPU. Il est également à noter qu'un abaissement de la tension de fonctionnement du CPU diminue également les courants de fuite des transistors, participant aussi également ainsi au gain de puissance consommée. Le temps de transition, très rapide, entre ces différents couples de fonctionnement fréquence/voltage est de l'ordre de la dizaine de microsecondes.[B16]

Ainsi, l'utilisation de ces différents couples tension/fréquence des CPUs, en fonction de la capacité de traitement nécessaire à un instant  $t$ , diminue la rapidité de calcul du CPU comme illustré en Tableau précédent (diminution de sa fréquence  $f$  de fonctionnement), mais permet ainsi de réduire la puissance ( $W$ ) qu'il délivre (abaissement de sa tension  $V$  de fonctionnement). L'intérêt de l'utilisation du **DVFS** réside donc dans le fait d'utiliser ses changements de fréquence CPU à bon escient. Ainsi, de multiples études ont déjà été menées dans ce domaine. Certaines s'intéressent à trouver, pour des applications découpées en différentes phases d'utilisation des ressources, une fréquence optimale à adopter pour les phases d'utilisation du CPU. Cette approche est utilisée en se basant sur des traces existantes d'intervalles d'utilisation CPU ou de données, et utilisant ainsi des algorithmes de prédiction d'utilisation. Ce sont donc ces premières études qui ont utilisées le DVFS avec une approche de prédiction de charge CPU essayant ainsi d'utiliser la fréquence optimale à chaque instant. D'autres approches se basent sur l'exécution de tâches dont les dates de fin d'exécution sont connues à l'avance. Ces approches plus simplistes, permettent donc de déduire les fréquences optimales d'exécution de chaque tâche afin de respecter au mieux leur date de fin d'exécution. Ces approches permettent d'obtenir

un excellent compromis entre la performance (temps d'exécution) et la réduction de consommation d'énergie obtenue.

## **II.6.Conclusion**

Dans ce chapitre nous avons détaillé les différents points qui concernent la qualité de services dans le cloud computing en mettant l'accent sur la consommation d'énergie, et ainsi nous avons invoqué les deux grandes approches de la réduction de celle-ci.

# Chapitre III

## Gestion d'énergie dans le cloud computing

### III.1 Introduction:

Du fait de son mode de consommation de ressources IT “à la demande”, le cloud computing est un secteur très concurrentiel, car les clients ne sont pas liés par des engagements à long terme et peuvent donc, assez facilement, changer d'opérateur. Ainsi, pour demeurer attractifs, ces derniers doivent régulièrement, d'une part, améliorer leur offre de service, et d'autre part, maintenir des tarifs compétitifs. Or, sur ce dernier point, la consommation d'énergie des Datacenter est un défi primordial puisque l'on estime que 70% du coût de fonctionnement d'un serveur provient de l'énergie nécessaire pour l'alimenter ... et le refroidir.

C'est donc en cherchant à réduire la consommation énergétique de leurs Datacenter que les opérateurs de cloud computing allient, selon un cercle vertueux, baisse des coûts et réduction d'émission de CO<sup>2</sup>.

### III.2. Modèle de gestion d'énergie

Afin d'avoir une consommation d'énergie modérée et fiable, le fournisseur est tenu de définir et de mettre en place des politiques de gestion d'énergie efficaces.

Dans cette partie, nous définissons un modèle de gestion d'énergie. Il s'agit d'un découpage “sémantique” des politiques de gestion dynamique d'énergie les plus souvent utilisées. Dans ce modèle, nous nous basons sur un couple formé d'une ressource et des opérations applicables sur celle-ci. La ressource dans notre contexte représente tout équipement électronique en production au sein d'un **IaaS** quel que soit son état. Les opérations quant à elles représentent toute action permettant de faire varier la vitesse d'exécution de cette ressource.

### III.3. Etat d'une ressource

Afin de mieux comprendre le principe de variation de vitesse d'exécution d'une ressource, nous définissons quelques concepts clés, relatifs à l'état de celle-ci, que nous utiliserons par la suite.

A l'image du fonctionnement standard d'un processus, la majorité des ressources informatiques subissent, au cours de leur exécution, des changements d'état que nous illustrons sous forme d'un diagramme d'état-

transition (Figure III.1).

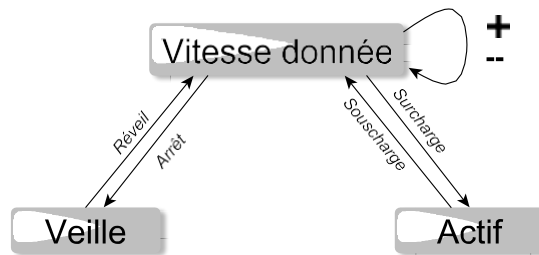


Fig III.1. Diagramme de changement d'état d'une ressource

Une ressource peut ainsi être dans un état : actif, veille ou de vitesse donnée :

- **Etat actif** : la ressource est en cours d'exécution. Par défaut, elle fonctionne au meilleur de ses performances et consomme par conséquent l'énergie qui lui est nécessaire (selon les normes du constructeur).
- **Etat de veille** : la ressource est "endormie". Elle nécessite un temps d'activation relativement long, du au "réveil" de la ressource, en cas de nouvelle requête. La ressource n'est ni en fonctionnement ni en attente active d'une requête. Cet état représente l'état le plus économique en énergie.
- **Etat de vitesse donnée** : la ressource est en cours d'exécution et sa vitesse est susceptible de varier (situation semblable à un ralentissement ou à une accélération). Elle peut ainsi fonctionner à différentes vitesses d'exécution et possède de ce fait un niveau de performance variable (fonction de sa vitesse courante). Ces changements sont soit le résultat d'une opération de configuration déclenchée par un acteur externe soit l'objet d'une politique implantée dans le périphérique. Peu importe l'origine de cet état, la ressource consomme une quantité d'énergie en général proportionnelle au niveau de performance d'exécution.

Comme exemple de politique, nous observons qu'au terme de l'exécution d'une opération et passe un délai d'attente initialement prédéfini, la ressource peut passer de l'état actif à un état de vitesse inférieure puis de veille successivement. Elle ne revient à l'état actif qu'à l'arrivée de nouvelles

opérations à exécuter.

Les opérations de variation de vitesse que nous définissons dans notre modèle sont essentiellement construites autour de ce diagramme d'état.

### III.3.1. Opérations de variation de vitesse d'exécution

Dans le modèle défini précédemment, les opérations de variation de vitesse d'une ressource sont possibles soit nativement de par la conception de la ressource, soit sont simulées à travers le regroupement des traitements qu'effectue la ressource.

### III.3.2. Variation de vitesse de façon native

Ce type d'opérations tire parti du fait que le périphérique/composant, de par sa conception, est capable d'être dans l'un des états actif, veille ou vitesse donnée. Ainsi, en se basant sur le taux d'utilisation du périphérique/composant, les changements d'états (nativement intégrés dans le périphérique/composant) permettent de réduire la consommation énergétique de la ressource en l'arrêtant (état de veille) ou en réduisant sa vitesse d'exécution (état de vitesse donnée) lorsqu'elle est inactive ou sous utilisée respectivement. [C1] L'état de vitesse actif étant utilisé quand le périphérique/composant est surcharge.

### III.3.3. Regroupement des traitements

Le changement de vitesse d'exécution de certains périphériques/composants en fonction de leur charge n'existe pas toujours de façon native.

De ce fait, plusieurs approches ont entrepris de l'émuler de façon logicielle grâce au regroupement (temporel ou spatial) des traitements :

- Lors d'un regroupement temporel des traitements, le périphérique/composant est mis en état de veille pendant toute une période de temps dédiée à l'accumulation (au **buffering**) des opérations à traiter. A l'échéance d'un délai (fixe ou variable), le

périphérique/composant passe à l'état actif et traite l'ensemble des requêtes en attente.

- Le regroupement spatial s'applique sur un ensemble de périphériques sur lesquels la charge est répartie.

Ces mécanismes permettent de simuler l'exécution du périphérique/composant à une vitesse donnée.

Cette vitesse représente, en théorie, la vitesse moyenne d'exécution du périphérique/composant pendant toute la période de traitement. Ces simulations permettent ainsi de maximiser le rendement des périphériques/composants en terme de consommation énergétique.

Il serait logique d'affirmer que les solutions de gestion d'énergie consistent toutes en une variation dynamique de vitesse d'exécution des périphériques/composants en fonction de la charge courante de ceux-ci.

Plusieurs projets se sont intéressés à la gestion d'énergie dans un **IaaS** en proposant des politiques pour la plupart de ses équipements électroniques. Toutefois, dans le cas des serveurs, les politiques proposées sont davantage focalisées sur ses principaux composants : le CPU, le disque, les périphériques réseaux et la mémoire.

### **III.4. Gestion d'énergie sur quelques composants d'un serveur**

Dans le modèle de gestion d'énergie, chaque composant ayant des caractéristiques de performance et de consommation d'énergie différentes, il est généralement approprié d'avoir une stratégie distincte de gestion d'énergie pour chacun d'eux.

Pour chaque composant nous évoquons éventuellement ses caractéristiques matérielles particulières, puis nous présentons quelques exemples de politiques proposées.



### III.4.1.Le CPU

Dans un **IaaS**, les performances d'exécution des **VMs** et le taux de consolidation de celles-ci sont liées aux caractéristiques architecturales du processeur les exécutant. L'architecture actuelle des processeurs permet d'atteindre des performances de calcul assez élevées, ce néanmoins au prix d'une forte consommation énergétique de l'ordre d'une centaine de Watts.[C2] Plusieurs travaux de recherches ont contribué à définir quelques politiques de gestion dynamique d'énergie.

#### A.Variation de vitesse de façon native

En règle générale, les politiques d'économie d'énergie applicable sur le CPU porte sur le changement d'état. Les approches proposées se composent de la mise du CPU en état de veille ou en état de vitesse donnée.

La mise en veille est traditionnellement privilégiée en période d'inactivité du CPU parce qu'elle abaisse la consommation d'énergie du CPU au minimum. Pour réduire davantage l'énergie consommée en absence d'exécution, les nouvelles architectures de CPU possèdent un nouvel état nommé "état de sommeil (sleep)". Ce dernier se décline sur plusieurs niveaux progressifs : sleep, deep sleep et éventuellement deeper sleep. L'état de "sommeil" se caractérise par la perte des données du cache, une consommation d'énergie nulle, mais un temps relativement (fonction de l'état courant) long pour revenir en état actif. Plus l'état de "sommeil" est profond, plus le temps de retour à l'état actif est long. Ainsi, en cas d'inactivité prolongée du CPU, celui-ci passe en état de veille et peut ensuite passer successivement par chacun de ces états de sommeil. Ce changement d'état permet ainsi de réduire la consommation d'énergie du CPU.

Le passage du CPU en état de vitesse donnée pour réduire la consommation d'énergie du CPU s'effectue à travers les variations de fréquence d'exécution du CPU en fonction de leur charge. Autrefois, un CPU en exécution fonctionnait toujours à sa performance maximale indépendamment de sa charge. L'énergie qu'il consommait était gaspillée en cas de faible charge. Pour remédier à cette situation, les constructeurs d'équipements électroniques ont intégré une nouvelle technologie nommée « Dynamic Voltage and Frequency Scaling (**DVFS**) » dans les nouveaux processeurs.

Le **DVFS** est une technologie qui permet de modifier dynamiquement la

fréquence (et la tension) d'un CPU en fonction de sa charge .[C3] Lorsque le CPU est faiblement chargé, sa fréquence (et sa tension) est graduellement abaissée de manière à ne pas altérer la latence d'exécution du CPU.

Ce dernier fonctionne à sa fréquence (et sa tension) maximale (état actif) lorsqu'il est fortement sollicité ou possède une charge supérieure à un seuil défini (statiquement ou dynamiquement).

L'économie d'énergie observable lors de l'utilisation du **DVFS** se base sur le rapport existant entre la tension et la fréquence. En effet, la puissance totale d'un CPU ( $P_t$ ) peut être exprimée comme la somme de sa puissance dynamique ( $P_d$ ) et sa puissance statique ( $P_s$ ) :

$$P_t = P_d + P_s$$

La puissance dynamique est proportionnelle à la "capacité"  $C$  invariable, à la fréquence ( $f$ ) et la tension ( $V$ ) au carré [C4]:

$$P_d = C * f * V^2$$

La puissance statique représente la puissance au repos. Par conséquent, en réduisant la fréquence du CPU, ce qui conduit également à réduire la tension, la puissance du système est globalement réduite .[C5]

## B. Regroupement des traitements :

Cette section présente quelques approches basées sur le regroupement des traitements. Grâce au regroupement temporel, l'exécution des applications par le CPU peut être différée. C'est le cas de l'approche proposée par Elnozahy et al. et connue sous le nom de **Request batching** .[C6] Le regroupement spatial utilise les possibilités de migration (la consolidation) d'applications entre systèmes ou d'équilibrage de charge (**load balancing**).

Le "**request batching**" est un mécanisme qui consiste à regrouper les requêtes en mémoire pendant une période prédéfinie (nommée **batching timeout**) et à les traiter une fois le délai atteint.

Avant l'expiration du délai, le CPU est inactif et est mis en état de veille. Le

CPU possède dans cet état une consommation d'énergie presque nulle. A l'expiration du délai, le CPU passe à l'état actif et fonctionne à performance maximale pour le traitement des requêtes accumulées.

Les serveurs étant utilisés à environ 30% de leur performance, cette configuration permet d'avoir de longues périodes d'inactivité et favorise ainsi la réduction de la consommation d'énergie pendant chacune de ces périodes. Le regroupement temporel des traitements permet de réduire au mieux la consommation d'énergie du CPU.

Toutefois à l'opposé du **DVFS**, cette approche n'est avantageuse que si, pendant les périodes d'activités, le CPU possède une charge garantissant que la consommation d'énergie du CPU à sa vitesse maximale est "proportionnelle" à sa charge.

Afin de bénéficier des atouts du **DVFS** et du regroupement des traitements, une nouvelle approche les utilise indépendamment en fonction de la charge du système. Ainsi, en absence de requêtes à traiter, le système est mis en état de veille jusqu'à ce qu'il soit réveillé par l'expiration du délai de regroupement (batching timeout). Grâce au **DVFS**, le CPU s'exécute successivement à différents niveaux de fréquence en fonction de sa charge courante.

Utilisé de la sorte, le système réalise beaucoup plus d'économie d'énergie en garantissant une utilisation efficace du CPU.

Le regroupement spatial des traitements se sert des atouts de migration d'applications qu'intègrent les systèmes d'exploitation. Particulièrement, Srikantaiah et al.[C7] exploitent les fonctionnalités de migration et de consolidation de VMs qu'apportent les technologies de virtualisation. Dans leur proposition, les VMs sont dynamiquement migrées entre les machines physiques en fonction de leurs caractéristiques, de leur charge courante et des performances de la machine physique. L'objectif idéal recherché par ces consolidations est de regrouper les VMs sur un nombre réduit de machines physiques pouvant les contenir. Ainsi, les machines non utilisées sont mises en état de veille ou en état de "sommeil" ce qui favorise la baisse de l'énergie consommée par le système. Celles en cours d'utilisation, grâce au **DVFS**, fonctionnent à un niveau de fréquence correspondant à la charge courante de leurs CPU. En cas d'augmentation de charge d'un CPU, les machines physiques non exploitées sont successivement mises en état actif et les

VMs sont migrées sur ces dernières afin d'offrir une bonne **QoS**.

Les machines en cours d'exécution ne sont pas source de gaspillage d'énergie ; et l'arrêt des systèmes non utilisés permet d'accroître davantage la baisse de la consommation d'énergie.

De même, le **"load balancing"** implémente entre les CPUs d'un même ordinateur physique permet de regrouper les traitements et simuler ainsi une vitesse donnée pour l'ensemble des CPUs.

### III.4.2. Le disque

La consommation d'énergie des disques représente en moyenne 25 à 35% de la consommation totale d'un centre d'hébergement. Fort de ce constat, de nombreuses approches ont été étudiées afin de réduire la consommation d'énergie des disques.

#### A. Variation de vitesse de façon native

En générale, les politiques de gestion d'énergie sont basées sur la définition d'une durée et d'un ensemble d'actions à effectuer en cas de dépassement de celle-ci.

La valeur de cette durée est soit fixe, soit variable (en fonction des tranches horaires par exemple) soit adaptable (peut par exemple dépendre de la charge de travail).

L'une des premières techniques de gestion d'énergie pour disque présentée par Douglass et al.[C8] Adopte ce principe en définissant une durée d'inactivité. Cette approche consiste à faire transiter le disque en état de veille (Spin down) si cette durée d'inactivité prédéfinie est atteinte. La transition vers l'état actif d'un disque se produit dès l'arrivée d'une opération d'E/S. Cependant, ce réveil coûte généralement cher et amène une détérioration significative des performances du disque. Le compromis entre performance et économie d'énergie est donc, dans ce cas, étroitement lié à la valeur de la durée adoptée. Plus le disque passe en état de veille (parce que la valeur de la durée choisie est trop petite), plus le système sera pénalisé en terme de performance. Moins le disque passe en état de

veille, moins l'économie d'énergie est possible. Le challenge de cette approche est le choix de la meilleure valeur de durée d'inactivité.

La transition vers l'état de veille ne survient qu'en cas d'inactivité prolongée du disque. Malheureusement, au regard du flux de données dans un IaaS, la période de latence entre 2 requêtes successives d'E/S ne permet pas fréquemment le passage en mode de veille des disques. De ce fait, les équipes de "Gurumurthi et al.". Et de "Carrera et al.". [C9] Proposent une nouvelle politique de gestion de la consommation énergétique des disques nommée « **Dynamic Rotations Per Minute** » (**DRPM**) par la première équipe et "multi-speed approche" par la seconde. Cette approche consiste à ajuster dynamiquement la vitesse de rotation du disque en fonction du volume des requêtes. En période d'inactivité, la vitesse de rotation est nulle (état de veille).

## B. Regroupement des traitements

Le but de ces regroupements est une utilisation efficiente des disques. Cette efficience est liée au rapport entre le taux de requêtes traitées et la consommation d'énergie du périphérique. Grâce au regroupement temporel, les demandes de E/S sur disque sont regroupées et subissent un traitement différé. Grâce au regroupement spatial, les données sont regroupées sur des disques en fonction de leur fréquence d'accès ainsi que de leurs données voisines (en terme d'adressage).

Le regroupement temporel des traitements distingue ces deux (2) types d'opérations : la lecture et l'écriture sur disque.

Pour la lecture, "Weissel et al.". Propose de regrouper et de reporter le traitement des demandes de lecture autant que les applications présentent la flexibilité requise pour le faire. [C10] A cet effet, "Weissel" introduit dans les **OS** une interface logicielle utilisée par les applications pour spécifier un délai maximal d'attente des demandes. Cette interface permet aussi d'activer un drapeau d'annulation pour des demandes d'E/S. Ainsi, quand le disque est en état de veille, l'**OS** peut :

- ✓ soit annuler la demande de lecture si le drapeau d'annulation a été activé

par l'application ;

- ✓ soit différer le traitement des demandes de lecture jusqu'à expiration du délai d'attente.

Le principe est pratiquement identique quand il s'agit des demandes d'écriture sur disque. Les blocs de données modifiées sont conservés en mémoire cache tant que le disque est en état de veille. D'un côté, ces données sont directement sauvegardées sur disque lorsque ces derniers deviennent actifs (**write-back with eager update (WBEU)**). D'un autre côté, les données modifiées sont temporairement sauvegardées dans un fichier log en mémoire (ou sur n'importe quel périphérique de stockage persistant) pendant toute la période de veille du disque. Elles sont ensuite sauvegardées sur disque au moment où celui-ci devient de nouveau actif (**write-through with deferred update (WTDU)**).

Dans chacun des cas de regroupement, les demandes sont traitées par lot, ce qui augmente les périodes d'inactivité des disques. Pendant ces phases d'inactivité, les opérations de gestion d'énergie applicables de façon native sur le disque permettent de réduire sa consommation d'énergie.

L'exploitation du regroupement spatial permet de regrouper les données sur les disques en fonction de leur taux d'utilisation. "Colarelli et al".[C11] définit une nouvelle technologie de configuration de disque nommée **MAID(Massive Array of Idle Disks)** et propose d'organiser les disques en "disque de données (DD)" et "disque cache (DC)". Les DD servent à la sauvegarde des données. Les DC servent à la réplication des données récemment utilisées et de celles qui leur sont contiguës. Les DC sont utilisées en permanence parce que toute opération de lecture y est exécutée. De plus, les DC servent de mémoire tampon pour les opérations d'écritures. Les données modifiées sont directement sauvegardées sur les DC. L'écriture sur le DD est différée et réalisée lors de son passage en état actif [C12]. Ainsi, pendant une durée relativement longue, il est possible d'appliquer les politiques de gestion d'énergie ci-dessus citées sur les DD et réaliser de ce fait une réduction de la consommation d'énergie des disques.

Cette idée a été étendue aux architectures **Redundant Array of Independent (or inexpensive) Disks (RAID)**. “Verma et al”. Propose **sample-replicate-consolidate mapping (SRCMap)**, une approche basée sur la consolidation des données par réplication/copie entre disques. **SRCMap** représente une technique d’optimisation de l’utilisation des disques d’un cluster en copiant uniquement les données actives (donnée la plus souvent utilisées). Pour chaque disque, **SRCMap** identifie les blocs de données actives qu’il copie sur d’autres disques. La petite taille des copies favorise la création de multiples copies sur un ou plusieurs disques ou inversement permet à un disque d’accueillir les copies de plusieurs disques. **SRCMap** s’assure de regrouper l’ensemble des données et les répliques sur un nombre minimum de disques. Les disques non utilisés sont de ce fait mis en état de veille et toute requête d’E/S est dirigée vers l’un des disques actifs contenant la copie de la donnée souhaitée. La réduction du nombre de disques actifs permet de baisser la consommation énergétique des disques. Les opérations d’écriture sont différées et des espaces sont prévus sur chaque disque et sont utilisés comme mémoire tampon.[C13]

### III.4.3.Le périphérique réseau

Des mesures d’équipements réseaux d’entreprises révèlent une utilisation de moins de 30%. Ils sont donc pour la plupart du temps inactifs mais leur consommation énergétique reste aussi importante. Les techniques de gestion d’énergie appliquées à ces équipements (notamment à une carte réseau) ont pour but de réduire l’énergie gaspillée lorsque le réseau est inactif.

#### a) Variation de vitesse de façon native

La première approche de gestion de la consommation d’énergie de carte réseau tire avantage des longues périodes d’inactivité régulièrement observées dans le fonctionnement d’une carte réseau. Ainsi, pendant les périodes d’inactivité, la carte réseau est mise en état de veille. “Nedevschi et al”.[C14] ont effectué un travail initial d’exploration portant sur la réduction de la consommation d’énergie des réseaux sans dégradation de performance de celui-ci. Par analogie au CPU, la

carte réseau intègre un support matériel permettant le changement d'état. La consommation d'énergie est réduite au minimum ou est presque nulle. Toutefois, en présence d'un trafic réseau (faible ou dense), la carte réseau fonctionne au mieux de ses performances et consomme ainsi la même quantité d'énergie ; ce qui conduit à nouveau à un gaspillage d'énergie.

Pour remédier à cette situation, "Nordman et Christensen" [C15] ont introduit le concept **d'Adaptive Link rate (ALR)** comme un moyen de changer dynamiquement le débit de traitement d'un réseau full-duplex en fonction de la charge de trafic qu'il traite. Sur ce principe, plusieurs approches ont été élaborées.[C16]

Notamment, "Gunaratne et al".[C16] ont implémenté une approche nommée **Adaptive Link rate (ALR)**. Le fonctionnement d'une carte réseau à faible débit étant plus économique (sur le plan de l'énergie) qu'une exécution à haut débit, l'approche **ALR** consiste à faire varier dynamiquement le débit d'exécution de la carte réseau en fonction de la charge du trafic reçu. Ce contrôle et cette variation de débit réduit ainsi la consommation énergétique globale de la carte réseau. Bien que la technique **ALR** soit efficace en terme d'économie d'énergie quelle que soit la charge du trafic, l'économie réalisée est inférieure au passage de la carte réseau en état de veille.

"Ananthanarayanan et al". Ont proposé une approche d'économie d'énergie utilisant conjointement le passage en état de veille et l'**ALR** qu'ils ont appliqué sur les Switchs d'un réseau local.

## **b) Regroupement des traitements**

Pour la mise en place du regroupement temporel, des travaux de recherches se sont également intéressés à faire du **buffering** au niveau des périphériques réseaux. Notamment, "Klausmeier et al". Ont proposé un procédé et un appareil pour mettre en tampon des informations dans un réseau numérique.

Afin d'utiliser le regroupement spatial, la technique de **NIC teaming** (généralement traduit par couplage de carte réseaux en français) a été largement adoptée. Le **NIC teaming** consiste à faire fonctionner 2 ou plusieurs cartes



réseaux ensemble afin d'améliorer la bande passante du serveur et de fournir une redondance réseau. "WeiLiu et al". ont ainsi proposé une solution d'économie d'énergie en se basant sur ce principe. Dans leur approche, toute carte réseau du couplage est capable de recevoir du trafic réseau. Chaque contrôleur de carte réseau est également adapté et configuré pour recevoir des informations associées à la charge du trafic. Cette configuration lui permet également de notifier les cartes réseaux en état de veille afin de les rendre active. Ainsi, en absence de trafic ou quand la charge du trafic est inférieure à un seuil prédéfini, un sous ensemble de cartes réseaux sont mises en état de veille. Elles sont successivement mises en état actif en fonction du trafic reçu et ce dernier est équitablement distribué entre les cartes réseaux actives.

### III.4.4. La mémoire

La gestion de la consommation d'énergie de la mémoire revêt actuellement une très grande importance et fait l'objet de plusieurs travaux de recherches.[C17]

#### 1) Variation de vitesse de façon native

Le modèle de gestion d'énergie de la mémoire le plus répandu suit les spécifications de **Rambus Dynamic Random Access Memory (RDRAM)**. [C18]

Il est basé sur le principe d'une mémoire capable de fonctionner dans de multiples niveaux de puissance. Dans un système de mémoire **RDRAM**, chaque mémoire peut être réglée de façon indépendante à une puissance apparente appropriée à un état : actif, veille, nap, ou power down.[C19]

Dans l'état actif, toutes les parties de la mémoire sont actives. Il représente le seul état dans lequel les opérations de lecture et d'écriture sont réalisables. L'état actif constitue également l'état le plus consommateur en énergie. Dans les autres états, l'ensemble des données est préservé et seules certaines parties du circuit mémoire sont activées. Ces états sont appelés états de faible consommation. La consommation d'énergie de la mémoire décroît au fur et à mesure qu'elle est passée à l'état de faible consommation inférieur. Toutefois, le passage d'un état de faible consommation à l'état actif est généralement coûteux en terme

d'énergie et de délai de réponse.[C20]

La mémoire passe en état de veille au terme de toute exécution de requête si la file d'attente de l'ordonnanceur est vide. Elle peut également passer d'un état de faible consommation à un autre suite à l'expiration d'un seuil d'inactivité. Le retour à l'état actif survient à l'arrivée d'une requête à traiter.[C21]

Les travaux antérieurs ont essentiellement porté sur l'utilisation des états de faible consommation pendant les périodes d'inactivité entre les requêtes de mémoire pour économiser l'énergie. Avec l'avènement des nouvelles mémoires de type **DDR (Double Data Rate)** et la possibilité qu'elles ont de fonctionner à des vitesses différentes en fonction de leur charge courante ; le **DVFS** mémoire a été adopté par plusieurs projets afin de réduire la consommation d'énergie des mémoires.[C22]

“Deng et al”. Propose une approche nommée **MemScale** basée sur le changement dynamique de fréquence de la mémoire. Pratiquement, **MemScale** met en œuvre la variation dynamique de fréquence et de tension du contrôleur mémoire en fonction des données relatives à l'état du système « taille de la file d'attente de l'ordonnanceur, nombre total de défaut de cache » que lui retourne l'algorithme de monitoring qu'il intègre. A chaque quantum, **MemScale** monitor le système et sélectionne une fréquence qui minimise l'énergie globale du système mémoire et la dégradation de performance des applications. La mémoire fonctionne ainsi à basse fréquence quand elle est sous chargée et au meilleur de ses performances dans le cas contraire.

## 2) Regroupement des traitements

Les approches précédentes, appliquées sur les barrettes mémoire de façon individuelle, permettent d'économiser de l'énergie en effectuant des transitions entre l'état actif et les états de faible consommation. Toutefois, au regard du coût (énergie et délai d'exécution) qu'engendrent ces transactions, ces approches ne sont véritablement efficaces que dans le cas d'une période d'inactivité suffisamment longue.[C23]

Cependant, les applications consommatrices de mémoire ne font généralement face qu'à de très courtes périodes d'inactivité. Par conséquent, le mécanisme de transition d'état, en vue de réduire la consommation d'énergie, nécessite

l'exploitation du regroupement spatial des traitements. Ce dernier concerne le groupement de pages permettant de disposer de plus longues périodes d'inactivités.[C24]

“Wu et al.”. proposent un nouveau modèle de mémoire nommée **RAMZzz** intégrant les techniques de gestion d'énergie. **RAMZzz** effectue la migration des pages en fonction des différentes fréquences d'accès à celles-ci. **RAMZzz** définit 2 catégories de mémoire : “hot” et “cold” et regroupe les pages sur l'une ou l'autre catégorie de mémoire. La première “hot” contient les données les plus souvent consultées et possède de très courtes périodes d'inactivité. La seconde “cold” contient les pages les moins consultées et a de ce fait, des périodes relativement longues d'inactivité. Ces périodes sont utilisées pour les transitions d'état en vue d'économiser la consommation d'énergie.

En cours de fonctionnement, les pages sont dynamiquement placées sur les mémoires en fonction de leur taux d'utilisation.

Ce regroupement permet de consolider les petites périodes d'inactivité en longues périodes et d'appliquer ainsi les politiques de transaction d'état. D'un autre côté, les mémoires “hot” sont exécutées à une performance donnée en fonction de leur charge ; ce qui favorise également l'économie d'énergie.

Le regroupement temporel dans le cas de la mémoire est étroitement lié au regroupement temporel du CPU. En effet, pendant que le CPU est inactif, la file d'attente de l'ordonnanceur étant vide la mémoire est également mise en veille.

### III.4.5. Gestion d'énergie d'autres composants dans un IaaS

A un niveau de gestion plus macroscopique, les opérations de variation de vitesse de façon native ou par émulation sont également applicables à un serveur physique en tant qu'entité. à tout autre équipement réseaux (Switch ou routeur) ou aux autres équipements utiles pour le refroidissement/ventilation d'un centre d'hébergement.[C25]

### III.5. l'optimisation est une vertu écologique ?

Tandis que le thème du cloud computing envahit les discours de la planète IT, une autre transformation de l'économie « high tech » est en cours. Avec une évolution vers des activités éco-responsables qui limitent la consommation de ressources naturelles.

Les vertus écologiques du cloud computing repose sur 3 piliers :

- la virtualisation,
- la mutualisation
- **L'optimisation**

Dans notre mémoire nous nous intéressons à l'optimisation de la consommation d'énergie dans les centres des données (cloud computing).

#### III.5.1. Comment peut-on optimiser la consommation énergétique des Datacenter ?

##### 1) Réduire la consommation des composants

- CPU, mémoire : diminuer la fréquence d'utilisation, augmenter la finesse de gravure, ...
- disques durs : utilisation de la technologie **SSD**, réduction de la vitesse de rotation des disques magnétiques, ...

##### 2) Réduire l'énergie nécessaire au refroidissement des composants

- améliorer l'efficacité des systèmes de climatisation : localisation au plus près des composants, utilisation de fluides au lieu de l'air, ...
- augmenter les limites de température de fonctionnement,
- utiliser l'air extérieur (free cooling) efficace dès que la température descend sous les 13°

##### 3) Améliorer l'efficacité énergétique globale des datacenters

Il s'agit globalement de minimiser la consommation d'énergie qui concerne pas directement les composants actifs du datacenter que sont les serveurs, la climatisation, le stockage, ...

Cette efficacité est mesurée par l'indicateur **PUE** (Power Usage Effectiveness) qui est le ratio entre l'énergie consommée par les composants IT et l'énergie globale consommée par le Datacenter.

Souvent supérieur à 2 il y a encore trois ans, ce ratio est descendu sous les 1,2 en 2011 pour les Datacenter les plus efficaces. Virtualisation, mutualisation, optimisation de la consommation énergétique des Datacenter permettent, à ressources IT comparables, de réduire significativement la consommation d'énergie et les émissions de gaz à effet de serre associées.

### **III.6.Conclusion**

Dans ce chapitre, nous avons présenté un modèle de gestion d'énergie ainsi que quelques implémentations pour certains périphériques (CPU, disque, carte réseau et mémoire) d'un serveur.

Le modèle de gestion d'énergie défini met en exergue le fait que les solutions actuelles de gestion d'énergie sont basées sur le changement d'état d'exécution des périphériques. Ce changement d'état peut être mis en œuvre de façon native via des fonctionnalités intrinsèques à certains périphériques. Et peut également être mis en œuvre par simulation via des regroupements temporel ou spatial des traitements qu'effectue le périphérique.

# Chapitre IV

## Outils de simulation

## **IV.1.Introduction**

Un des grands problèmes de développement d'applications sur le Cloud est l'environnement dans lequel nous pouvons reproduire les tests. Faire des tests sur un environnement Cloud réel comme Amazon EC2, pourrait être très coûteux. Mais avec la simulation, les développeurs peuvent tester et optimiser leurs applications Cloud sans avoir à payer quoique ce soit, et même les clients peuvent tester les performances de leurs services de façon reproductible, sur un environnement contrôlable et gratuit, avant de mettre les applications sur l'environnement Cloud réel.

Un simulateur Cloud est un outil qui permet de manipuler des Data center avec leurs serveurs de base et tout le hardware nécessaire à leur bon fonctionnement (CPUs, Disque dur, RAM etc.).

Afin de simuler la création et distribution des tâches sur des machines virtuelles. Dans notre travail, nous utilisons le simulateur Greencloud.

## **IV.2.Pourquoi la simulation ?**

Dans un système distribué, il existe des enjeux à résoudre tels que la gestion des ressources et l'ordonnancement des applications, ces tâches sont compliquées et il n'existe pas une solution optimale pour répondre à ces problématiques. D'autre part, dans l'environnement d'un système distribué comme Cloud, il est difficile d'effectuer les différents scénarios avec différents nombres de ressources et utilisateurs afin d'évaluer la performance des algorithmes du partage de charge, Broker, gestion des ressources...etc. Lorsqu'on veut évaluer les scénarios de manière répétable et contrôlable, cela est parfois impossible car on aura le problème du coût et de la gestion.

Afin de résoudre cette problématique, les chercheurs utilisent les simulateurs pour effectuer leurs scénarios avant de les appliquer au sein d'un système distribué réel. Dans ce qui suit, on va présenter l'un des outils de simulation le plus connu et utilisé dans le domaine de simulation du Cloud.

### IV.3.Outils de simulation

Il existe plusieurs outils de simulation de systèmes distribués. Les plus connus sont SIMGRID, GroudSim , GSSIM, Cloudsim et GreenCloud.

**IV.3.1.SIMGRID [D1]** est un simulateur qui fournit des fonctionnalités de base pour la simulation d'applications distribuées dans des environnements hétérogènes distribués. Son utilisation est bien adaptée pour l'évaluation de l'heuristique, le prototypage, le développement et l'amélioration des applications de grilles.

**IV.3.2.GroudSim [D2]** Offre des outils de simulation d'événement pour le Cloud Computing.

Les principaux domaines de la simulation sont les transferts de fichiers, le coût des ressources et le calcul des coûts d'exploitation. Il peut être utilisé avec un package contenant des informations sur la probabilité d'une panne matérielle.

Dans ce cas, le module d'événements peut détecter des erreurs se produisant sur la machine ou sur le réseau et de lancer un processus de reconfiguration des entités concernées.

**IV.3.3.GSSIM [D3][D4]** Est un simulateur qui permet l'étude des politiques d'ordonnancement avec une description flexible de l'architecture et les interactions entre les modules.

Les formats standards sont acceptés pour la description des flux de travail, tels que **Standard Workload Format (SWF)** et **Grid Workload Format (GWF)**. Un fichier XML peut être défini pour inclure des informations plus détaillées (ex les contraintes de temps). GSSIM permet la virtualisation et intègre également des modèles de l'énergie et une représentation précise de l'utilisation des ressources.



**IV.3.4.CloudSim** est une nouvelle structure de simulation généralisée et extensible qui permet la modélisation des environnements hétérogènes, la simulation et l'expérimentation de Cloud émergeant des infrastructures de calcul et des services d'application.

Le simulateur CloudSim couvre la plupart des fonctionnalités ayant lieu dans un centre de traitement des données en détail. Ceci inclut :

- ❖ Simulation de la définition de matériel de centre de traitement des données (Data Center) en termes de machines physiques composées de processeurs, de dispositifs de stockage, de mémoire et de largeur de bande interne.
- ❖ Simulation des spécifications, de la création et de la destruction de machine virtuelle.
- ❖ Gestion des machines virtuelles, allocation des ressources physiques de matériel pour le fonctionnement des machines virtuelles basées sur différentes politiques (par exemple en temps partagé et espace partagé).
- ❖ Simulation de l'exécution des programmes de l'utilisateur ou des demandes (Cloudlet/Gridlet) sur les machines virtuelles.

**IV.3.5.GreenCloud [D5]** Est une extension du simulateur réseau **NS2**. **[D6]** Il fournit une modélisation détaillée et l'analyse de l'énergie consommée par les éléments du réseau : serveurs, routeurs, et les liens entre eux.

En outre, il analyse la distribution de la charge dans un réseau ainsi que les communications avec une grande précision.

En terme d'énergie, GreenCloud définit trois types d'énergie : calcul (CPU), communications et centre de calcul physique (systèmes de refroidissement), et inclut deux méthodes pour la réduction de l'énergie : **DVS (Dynamic Voltage Scaling)** pour diminuer la tension des commutateurs et **DNS (Dynamic Network Shutdown)** qui permet d'arrêter les commutateurs lorsque cela est possible.

## IV.4.Caractéristiques du GreenCloud

**GreenCloud** [D7] fournit un environnement de simulation, implémenté en C++, dédié à l'analyse de la consommation d'énergie dans les centres de calcul. Il est principalement destiné aux études expérimentales sur les modes de communication et l'évaluation détaillée de la consommation d'énergie, des architectures actuelles mais aussi des futures architectures des data center.

GreenCloud est un simulateur au niveau paquet réseau, ce qui signifie qu'à chaque fois qu'un message doit être transmis, c'est une structure de paquets qui doit être allouée en mémoire puis exécutée. Bien que cette approche améliore significativement la précision de la simulation, cela augmente le temps global de simulation et conduit à des difficultés sur de très grandes instances (passage à l'échelle difficile).

GreenCloud se focalise sur la modélisation détaillée des aspects de communication des données dans les réseaux des data center. Etant basé sur la plate-forme **NS2** implémentant de bout en bout le protocole TCP/IP, cela permet de capturer la dynamique des protocoles de communication tels qu'IP, TCP, UDP, etc.

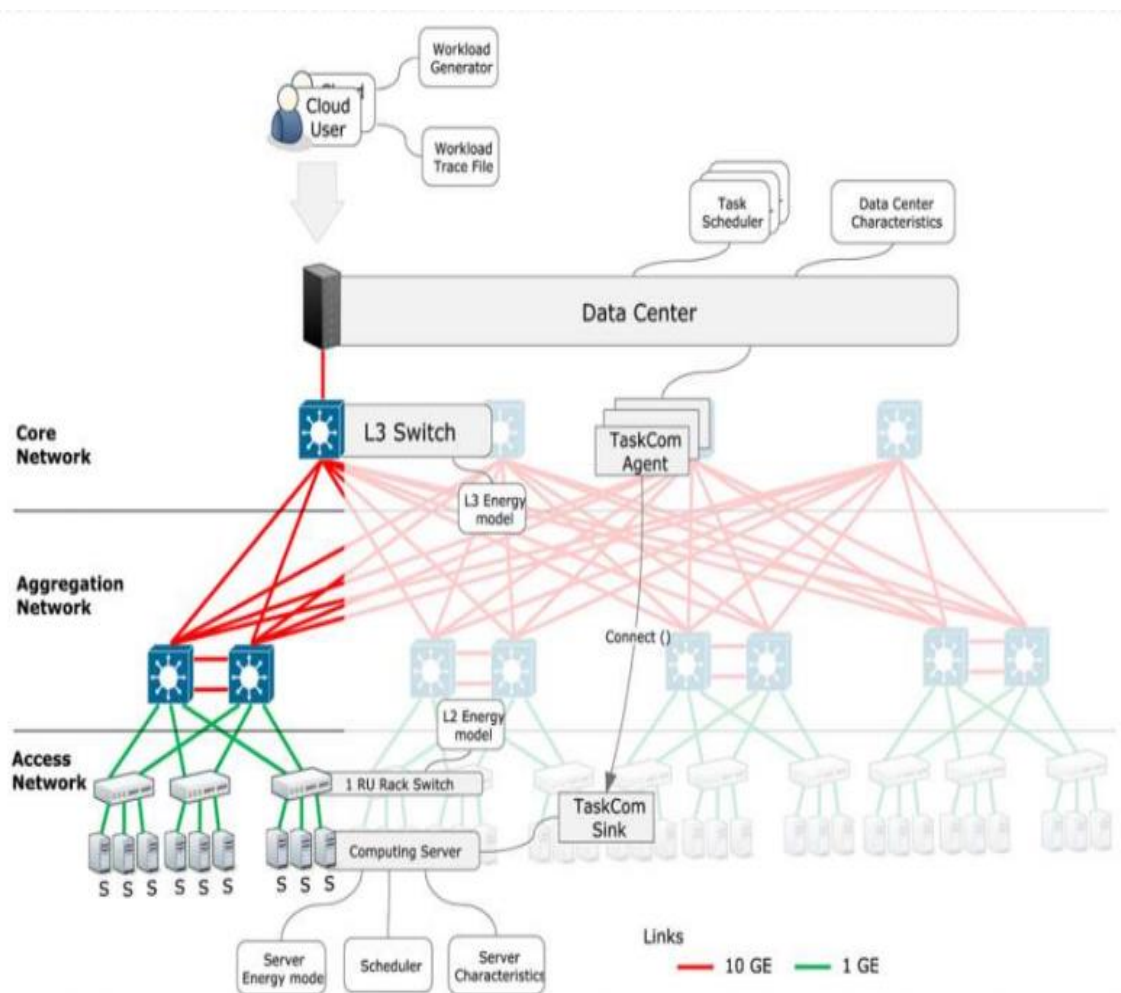
Ainsi, chaque message envoyé entre deux éléments, est fragmenté en un nombre de paquets dont la taille est limitée par le réseau **MTU**. Puis, lors de leur routage à l'intérieur du réseau du Data center ils peuvent être soumis à un certain nombre de problèmes typiques tels que des erreurs sur les liens ou de la congestion dans les routeurs réseau donnant lieu à des pertes de paquets. Pour être considéré comme correctement terminé, chaque objet de la charge de travail doit être correctement exécuté par les principaux composants de calcul et de communication du simulateur. Le composant de calcul définit le nombre d'opérations qui doivent être exécutées avant une date limite de terminaison (dit deadline). L'inclusion de deadlines vise à introduire des contraintes de **QoS** spécifiées dans un **SLA**.

## IV.5. Architecture du GreenCloud

Le composant de communication détermine la quantité et la taille des transferts de données qui doivent être effectués avant, pendant et après l'exécution du charge de travail.

Afin de couvrir la majorité des applications de Cloud Computing, GreenCloud définit trois types de charge de travail [D7]:

- ❖ Calcul intensif qui charge considérablement les serveurs de calcul
- ❖ Données intensives qui nécessitent des transferts de données très lourds
- ❖ Une charge de travail équilibré qui vise à modéliser les applications ayant des exigences à la fois de calcul et de transfert de données.



**Figure IV.1** Architecture de GreenCloud.

Les informations sur la consommation d'énergie sont divisées en trois catégories : calcul, communication et infrastructure physique.

La modélisation de la consommation d'énergie associée aux composants de calcul, aux Switches réseau ainsi qu'aux systèmes de refroidissement du data center.

Les modèles énergétiques profitent ainsi de la granularité du simulateur (niveau paquet), ce qui permet de mettre à jour les niveaux de consommation d'énergie à chaque fois qu'un nouveau paquet arrive ou quitte un lien, ou à chaque fois qu'une tâche commence ou termine son exécution. GreenCloud implémente le **DNS** (Dynamic Network Shutdown) comme outil de réduction de consommation réseau.

## **IV.6.Conclusion**

L'enjeu de ce chapitre était de présenter le simulateur Greencloud, utilisé au cours de notre travail, d'introduire ses caractéristiques et d'expliquer les nouveaux outils qui ont dû être intégrés afin qu'il regroupe l'ensemble des fonctionnalités voulues.

# Chapitre V

## Implémentation

## V.1. Introduction

L'objectif de notre projet est de contribuer à l'étude de la consommation d'énergie en proposant des comparaisons entre plusieurs paramètres pour chercher les cas les plus optimales.

## V.2. Environnement de développement

### V.2.1 Qu'est-ce que Eclipse ?

**Eclipse [E1]** est un projet décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Son objectif est de produire et de fournir des outils pour la réalisation de logiciels, englobant des activités de programmation (notamment environnement de développement intégré et « frameworks » et d'AGL recouvrant la modélisation, « la conception ,le testing », la gestion de la configuration, et le (reporting)... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel , c'est l'apparition du framework Eclipse RCP en 2004.

Figurant parmi les grandes réussites de l'Open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs de logiciels et sociétés de services. Les logiciels commerciaux *Lotus Notes 8*, *IBM Lotus Symphony* ou *WebSphere Studio Application Developer* sont notamment basés sur Eclipse.

### V.3.Introduction au simulateur GreenCloud :

GreenCloud est un simulateur de niveau de paquets qui utilise les bibliothèques existantes **Network Simulator 2 (NS2)** dans la gestion de l'énergie des data center ( cloud computing, Il modélise les différentes entités du cloud tels que les serveurs, les Switches, les liens de communication et leur consommation d'énergie.

Il peut être serviable dans le développement de solutions pour la surveillance des ressources d'un répartition, et du scheduling workloads (la planification de la la charge de travail ) pour un nombre d'utilisateurs, il permet d'optimiser les protocoles utilisés pour la communication et de fournir également des solutions pour network switches (les switches réseau).

GreenCloud est développé par l'Université du Luxembourg et publié sous la General Public License (**GPL**).

### V.4.Installation de GreenCloud

L'outil GreenCloud est développé principalement pour les systèmes basés sur debian comme (*Ubuntu, Debian, Linux Mint, etc*). et peut être mieux utilisé avec Ubuntu 12.x.

#### V.4.1.Installation sur une machine virtuelle (VM)

GreenCloud est livré avec une machine virtuelle préconfigurée qui inclut

- ✓ Eclipse debugger ns
- ✓ la modification du code source
- ✓ démarrer / exécuter la simulation.

#### V.4.2.Installation sur un système réel

Voici les instructions pour GreenCloud sur un système réel.

Exécutez les commandes comme indiqué ci-dessous.

-----CODE-----

Décompressez le logiciel en utilisant la commande

```
pradeep@localhost $] tar zxvf greencloud-v2.0.0.tar.gz
pradeep@localhost $] cd greencloud-v2.0.0
pradeep@localhost greencloud-v2.0.0 $] ./configure
pradeep@localhost $] ./install-sh
```

Exécutez le (script) avec cette commande (Cette commande fera apparaître une fenêtre dans un navigateur avec des données de simulation de test)

`pradeep@localhost $] ./run`

-----CODE-----

### V.5.Exemple de Simulation :

Le GreenCloud est livré avec une simulation de test par défaut avec 144 serveurs avec un utilisateur Cloud. Tous les paramètres peuvent être variées et testées les données entrantes dans le fichier Tcl.

Les fichiers Tcl sont situés dans le repertoire ~ GreenCloud / src / scripts /. Il y a beaucoup de scripts qui spécifient la fonctionnalité de l'environnement cloud

Scripts	Fonctionnalité
main.tcl	<ul style="list-style-type: none"> <li>❖ spécifie l'architecture du centre de données</li> <li>❖ spécifie le temps de simulation</li> </ul>
topology.tcl	❖ créer la topologie du réseau
dc.tcl	❖ crée les serveurs et les machines virtuelles
setup_params.tcl	❖ configuration générale comme les serveurs, les switches, les tâches, etc.
user.tcl	❖ définit les utilisateurs et leurs comportements
record.tcl	❖ presentation des resultants
finish.tcl	❖ afficher les statistiques

**Table V.1** les fonctionnalite de GreenCloud scripts

### V.6.Changement des paramètres de Cloud

Les paramètres du data centre peuvent être modifiés en utilisant les fichiers Tcl qui ont été présentées dans le tableau précédent.

Un simple changement est illustré ci-dessus. Deux fichiers (main.tcl et topology.tcl) sont modifiés, approvisionne 40 serveurs et un seul utilisateur de cloud avec une capacité de charge moyenne de 0,3



-----CODE-----

```

#topology.tcl, where the network topology is been set
switch $sim(dc_type) {
"three-tier high-speed" {
    set top(NCore) 2 ;# Number of L3 Switches in the CORE network
    set top(NAggr) [expr 2*$top(NCore)] ;# Number of Switches in
AGGREGATION
    set top(NAccess) 256 ;# Number switches in ACCESS network
    set top(NRackHosts) 3 ;# Number of Hosts on a rack
}
"three-tier debug" {
    set top(NCore) 1 ;# Number of L3 Switches in the CORE network
    set top(NAggr) [expr 2*$top(NCore)] ;# Number of Switches in AGGREGATION
    set top(NAccess) 2 ;# Number switches in ACCESS network per pod
    set top(NRackHosts) 20 ;# Number of Hosts on a rack
}
}
# three-tier
default {
    set top(NCore) 8 ;# Number of L3 Switches in the CORE network
    set top(NAggr) [expr 2*$top(NCore)] ;# Number of Switches in AGGREGATION
    set top(NAccess) 64 ;# Number switches in ACCESS network
    set top(NRackHosts) 3 ;# Number of Hosts on a rack
}
}

# Number of racks is set as 2 * 1
set top(NRacks) [expr $top(NAccess)*$top(NCore)]

# Number of servers is set to 2 * 20 (40 servers)
set top(NServers) [expr $top(NRacks)*$top(NRackHosts)]
.....

```

-----CODE-----

## -----CODE-----

```

#main.tcl, where the simulation information and data center load information is
specified
# Type of DC architecture
set sim(dc_type) "three-tier debug"

# Set the time of simulation end
set sim(end_time) [ expr 60.1 + [lindex $argv 1] ] ;# simualtion length set to 60 s +
deadline of tasks

# Start collecting statistics
set sim(start_time) 0.1

set sim(tot_time) [expr $sim(end_time) - $sim(start_time)]

set sim(linkload_stats) "enabled"

# Set the interval time (in seconds) to make graphs and to create flowmonitor file
set sim(interval) 0.1

# Setting up main simulation parameters
source "setup_params.tcl"

# Get new instance of simulator
set ns [new Simulator]

# Tracing general files (*.nam & *.tr)
set nf [open "../traces/main.nam" w]
set trace [open "../traces/main.tr" w]

# Building data center topology
source "topology.tcl"

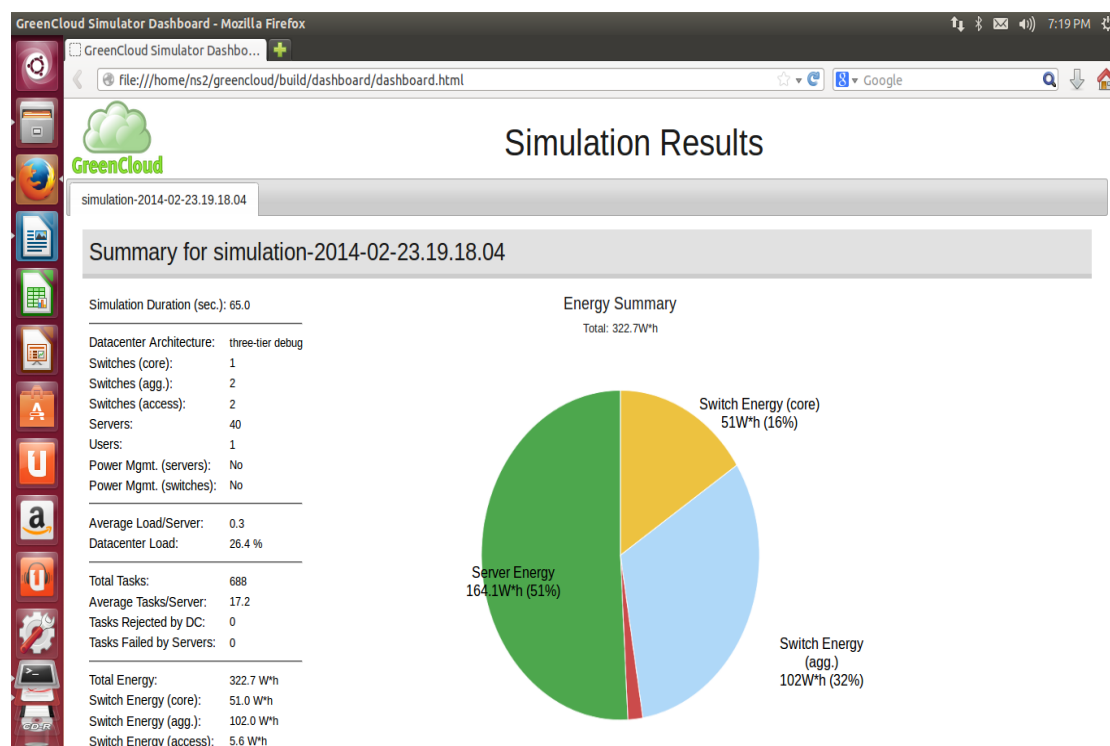
```

## -----CODE-----

Le tableau suivant montre le changement des paramètres précédents

Data center architecture	Three-tier Debug
Core switches	1
Aggregation switches	2
Access switches	2
Number of servers	40
Users	1
Average load/server	0.3
Total tasks	688
Average task/server	17.2
Total energy calculated	322.7 watt hour
Servers energy	164.1 watt hour
Total switches energy	158.6 watt hour

**Table V.2** : résultats de changement de paramètres



**Figure V.1** : Résultats de simulation

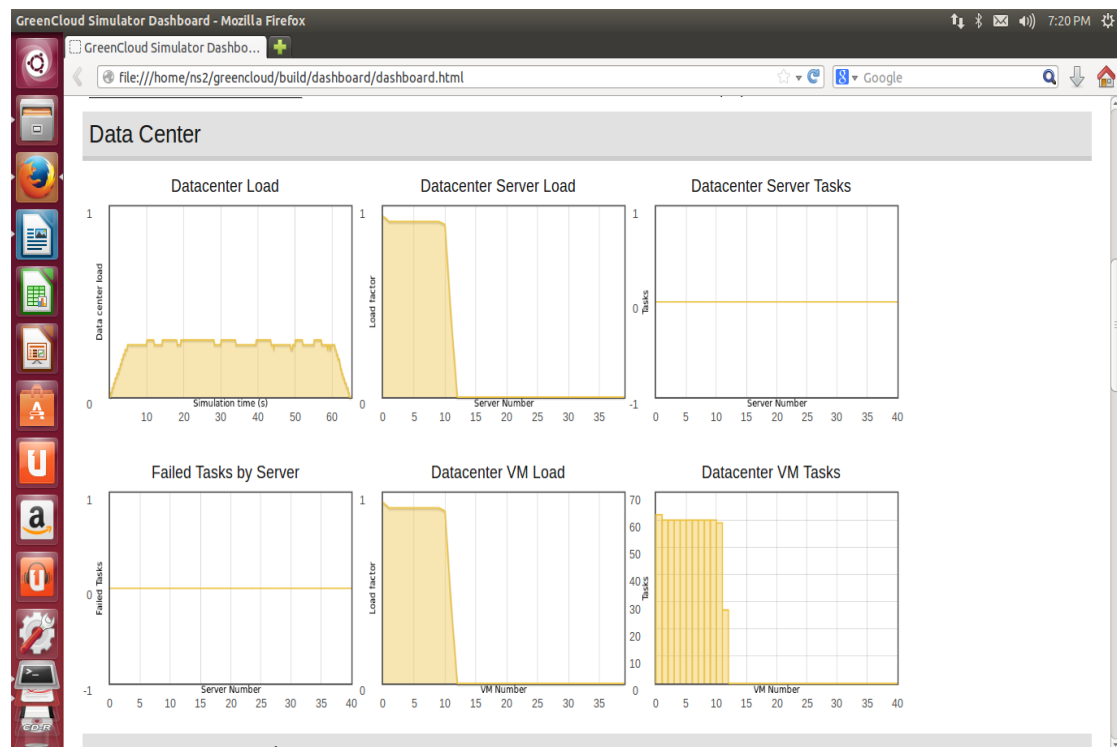


Figure V.2 : Caractéristiques de centre de données

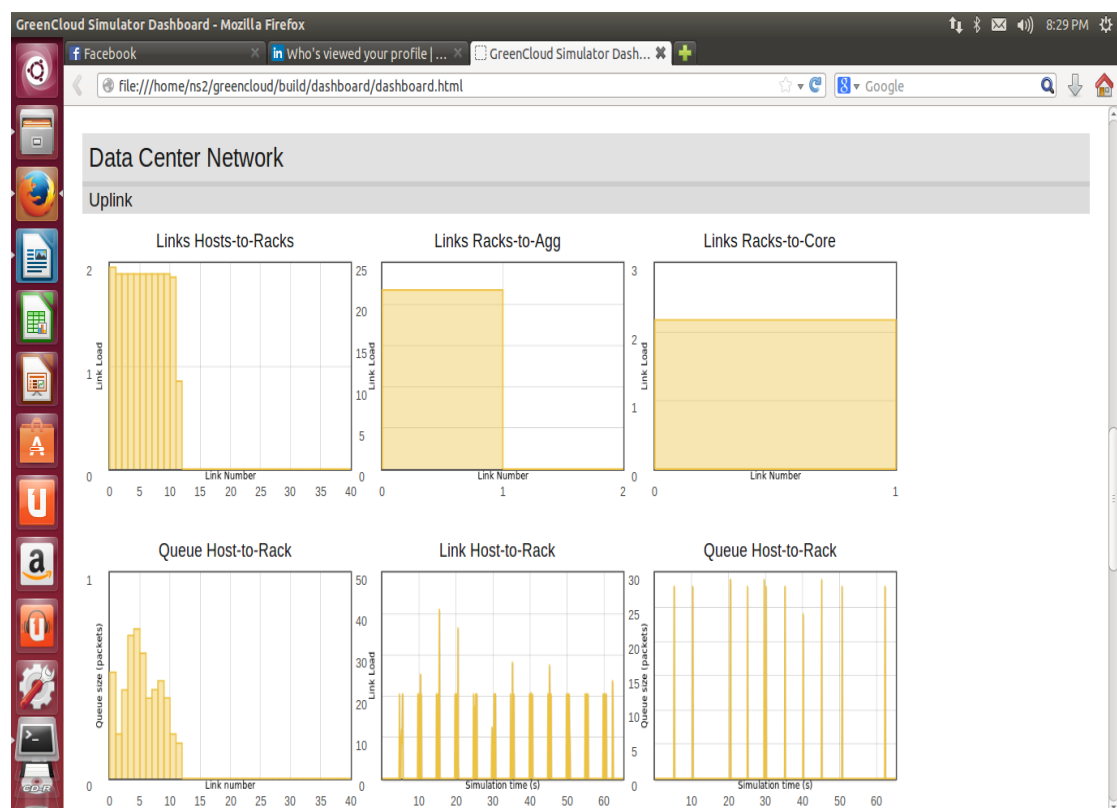
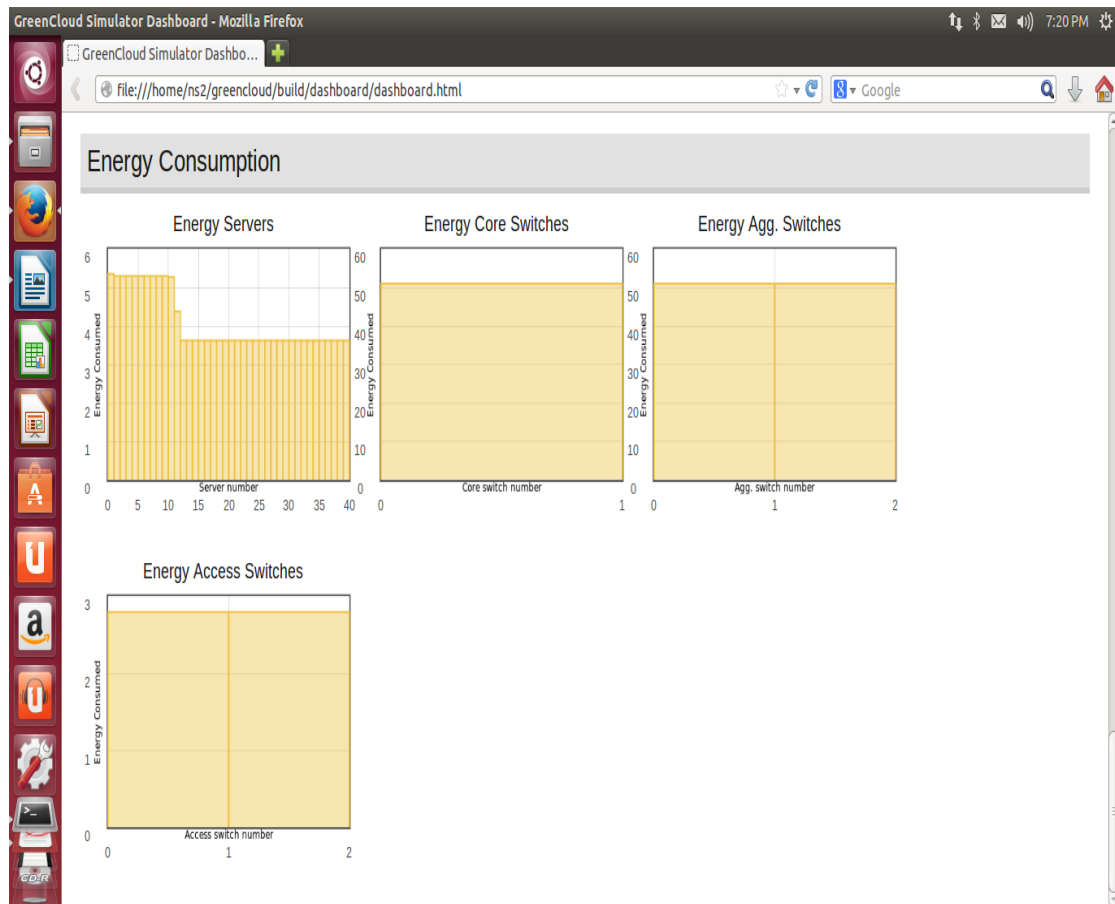


Figure V.3 Caractéristiques de réseau de centre de données



**Figure V.4 :** les détails de la consommation d'énergie

Le graphe dans le navigateur affiche 4 parties :

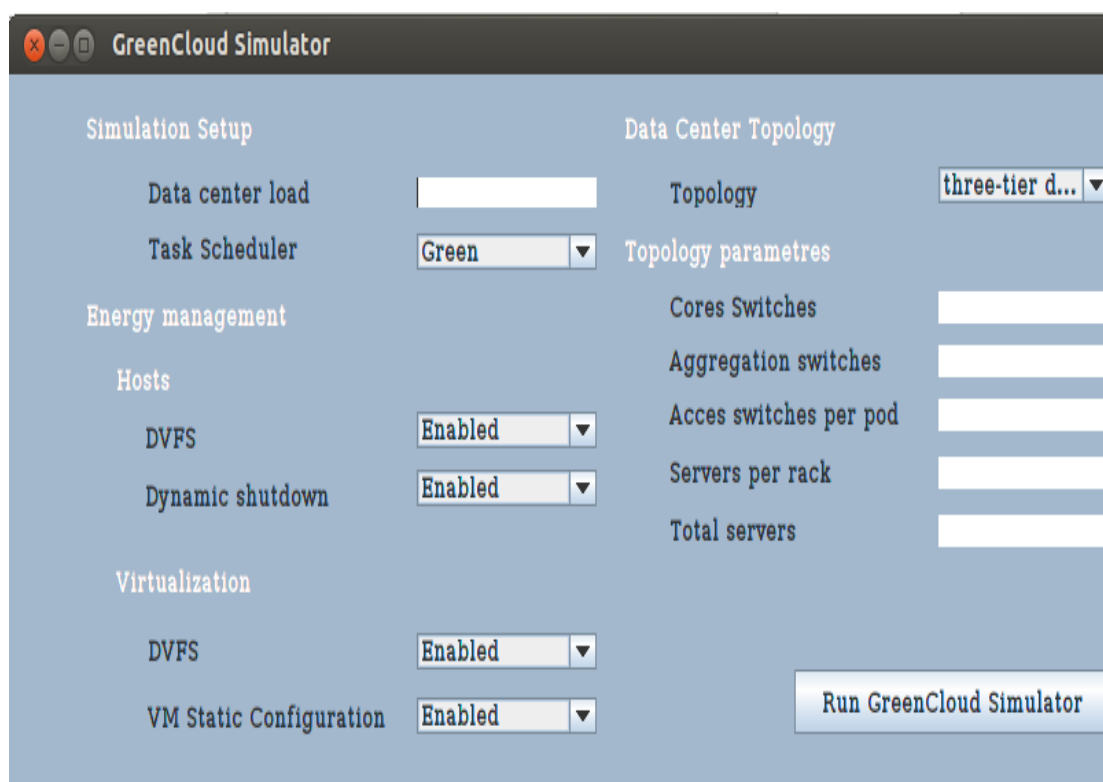
- ✓ Un résumé de la simulation comme indiqué dans le tableau V.2
- ✓ Les caractéristiques du centre de données comme le montre la figure V.2
- ✓ Les caractéristiques du réseau à courant continu comme le montre la figure V.3
- ✓ Les détails la consommation d'énergie comme le montre la figure V.4

### V.6.1.Interface de changement des paramètres

Pour faciliter le changement de paramètres nous proposons une interface graphique qui a le contenu des paramètres suivant :

- ✓ Configuration de simulation
- ✓ Gestion de l'énergie
- ✓ Data Center Topologie.

Voici notre interface dans la figure ci-dessus :



**Figure V.5** : interface de changement de paramètres

## V.7. Résultats et expérimentations :

### V.7.1. 1<sup>ère</sup> expérimentation : les topologies de centre e données

L'objectif de cette expérimentation est de comparer la valeur de la consommation d'énergie entre chaque architecture du réseaux et les autre

Le tableau ci-dessus cite les différents résultats de simulation

Data center architecture	Three-tier debug	Three-tier	Three-tier heterogenous
Core switches	1	8	1
Aggregation switches	2	16	2
Acces switches	3	64	3
Number of servers	144	1536	144
Users	1	1	1
Average load/server	0.3	0.3	0.3
Total tasks	32689	348899	32874
Average task/server	227.0	227.1	228.3
Total energy calculated	301.9 w*h	6314.9 w*h	495.9 w*h
Server energy	138.6 w*h	3541 w*h	332.6 w*h
Total switches energy	163.3 w*h	2773.9 w*h	163.3 w*h

**Table V.3**les différents résultats de simulation de chaque architecture

### Discussion des résultats

Dans notre projet, on se base sur cette expérimentation, on va comparer le déroulement de la consommation d'énergie entre trois déférentes archetecture du réseaux ( topologies network Datacenter ) : **Three-tier debug** , **Three-tier** et **Three-tier heterogenous** on remarque que la meilleure architecture qui donne une bonne optimisation de la consommation d'énergie est la première topologie « **Three-tier debug** » parce quelle travaille avec une minimisation de tous les paramètres, du nombre des cores jusqu'au le nombre des taches , et cette architecture c'est la plus favorable dans « Greencloud » .

### V.7.2. 2<sup>ème</sup> expérimentation : le scheduling

L'objectif de cette expérimentation est de comparer la valeur de la consommation d'énergie entre deux méthodes de tâche « **Green** » et « **RoundRobin** »

Le tableau ci-dessus cite la différence entre ces deux résultats de simulation

Scheduler	Green	RoundRobin
Core switches	1	1
Aggregation switches	2	2
Acces switches	3	3
Number of servers	144	144
Users	1	1
Average load/server	0.3	0.3
Total tasks	32689	32689
Average task/server	227.0	227.0
Total energy calculated	301.9 w*h	481.0 w*h
Servers energy	138.6 w*h	317.7 w*h
Total switches energy	163.3 w*h	163.3 w*h

**Table V.4 :** Comparaison entre deux méthodes de planification « Green »  
« RandRobin »

### Discussion des résultats

Dans le tableau précédent on compare deux méthodes de planification ( scheduling ) différentes pour trouver la méthode plus efficace entre eux.

Nous remarquons que la méthode de la planification « **green** » optimise l'énergie des serveurs plus que la méthode « **RoundRobin** » même en remplissant tous les paramètres avec équivalence.



### V.7.3. 3éme expérimentation : Augmentation du Nombre Vms

L'objectif de ces expérimentations est d'évaluer comment l'augmentation du nombre de virtuels machines peut avoir un effet sur la consommation d'énergie

Vms ( statics virtualisation )	1	0
Core switches	1	1
Aggregation switches	2	2
Acces switches	3	3
Number of servers	144	144
Users	1	1
Average load/server	0.3	0.3
Total tasks	32689	32874
Average task/server	227.0	227.0
Total energy calculated	301.9 w*h	305.9 w*h
Servers energy	138.6 w*h	142.6 w*h
Total switches energy	163.3 w*h	163.3 w*h

**Table V. 5 :** résultats de simulation avec et sans Vms ( statics virtualisation )

### Discussion des résultats

Dans cette partie d'expérimentation, nous faisons une comparaison entre deux état du situations des machine virtuelles quand elles sont activés ou désactivées :

- ✓ **Etat Activé (VMs)** : une machine virtuelle installer sur un Host
- ✓ **Etat Désactivé (VMs)** : inexistence des virtuelles machines sur les Hosts

Nous remarquons que l'absence des virtuelles machines peut avoir un effet négatif sur la consommation d'énergie, et leur présence est nécessaire pour maintenir l'optimisation d'énergie.

## V.8. conclusion

Dans ce chapitre, nous avons montré comment la virtualisation, la topologie de réseau des datacenter et le « **scheduling** » jouent un rôle très important pour la réduction et l'optimisation de la consommation d'énergie dans le Cloud computing.

conclusion générale

## *Conclusion général*

De nos jours, la consommation d'énergie des Data center a des répercussions énormes sur les environnements. Les chercheurs cherchent à trouver des solutions efficaces pour que les centres de données puissent réduire leur consommation d'énergie tout en maintenant la qualité souhaitée des objectifs de service ou de niveau de service.

La Machine virtuelle (VM), une technologie virtuelle qui a été largement appliquée dans les environnements de centres de données en raison de ses caractéristiques énormes telle que la fiabilité, la flexibilité et la facilité de gestion.

Il y a plusieurs facteurs qui ont permis au « Cloud computing » de réduire leur consommation de l'énergie et de carbone. Les organisations peuvent réduire leur consommation d'énergie par les utilisateur en déplaçant leur applications vers le Cloud. Ces économies sont entraînées par la grande efficacité de grande échelle Nuage « data center ».

Ce projet a défini, entre autres, ce qu'est le Cloud Computing, ses différents types de services et ses différentes offres. En- suite, on a procédé à des tests des techniques de réduction de consommation d'énergie à savoir : Les topologies des réseaux des data center , les Etats des VMs ( Activé ou Désactivé ) et Les méthodes de planification des charges de travaille ( scheduling ) qui ont été déployé dans un environnement Cloud simulé, testés à travers des expérimentations simulés pour la bonne gestion de la consommation d'énergie.

Les résultats des expérimentations ont montré comment l'énergie peut être sauvée en utilisant les expériences définies au cours de ce projet pour contrôler sa consommation. On a utilisé le « GreenCloud » qui permet la surveillance complète du système pour vérifier l'efficacité et l'efficience de l'architecture proposée, Les résultats d'évaluation ont montré que nous pouvons économiser jusqu'à 27% de l'énergie lors de l'application des différents types d'architecture modifiant certains composants du « GreenCloud ».

En fin en espérant avoir atteint notre objectif, d'autres perspectives restent ouverts pour tester d'autres méthodes pour une bonne gestion de la consommation de l'énergie qui est le pilier du fonctionnement de tout système à bases des nouvelles technologies et qui a un effet directe sur la vie quotidienne de l'être humain que ce soit sur l'utilisation de ces outils ou sur l'environnement par les gaz à effet de sert que dégagent ces systèmes .

## Bibliographie

- [A1] explications concernant l'informatique en nuage (Cloud Computing) Feldeggweg 1300 Berne – [www.leprepose.ch](http://www.leprepose.ch) – Octobre 2011
- [A2] clusif-cycle de conférences sur Cloud Computing et virtualisation Cloud Computing et sécurité – Pascal SAULIERE ; architecte Microsoft France
- [B1] Yacin Kessaci, Multi-criteria Scheduling on Clouds, Operations Research. Université des Sciences et Technologie de Lille - Lille I, 2013.
- [B2] W.A. Pauley. Cloud Provider Transparency: An Empirical Evaluation. Dans: Security Privacy, IEEE 8.6(2010)
- [B3] Steve Versteeg Saurabh Kumar Garg ET Rajkumar Buyya. SMICloud: A Framework for Comparing and Ranking Cloud Services. Dans: Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA), Melbourne, Australia. 2011
- [B4] Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter Puschner, Jan Staschulat et Per Stenstrom. The worst case execution time problem overview of methods and survey of tools? Dans : ACM Trans. Embed. Comput. Syst. 7.3 (mai 2008), 36 :1–36 :53. doi : 10.1145/1347375.1347389
- [B5] YuanShun Dai, Bo Yang, Jack Dongarra et Gewei Zhang. Cloud service reliability: Modeling and analysis. Dans: The 15th IEEE Pacific Rim International Symposium on Dependable Computing. 2009
- [B6] Juan Caceres, Luis M Vaquero, Luis Roderio-Merino, Alvaro Polo et Juan J Hierro. Service scalability over the cloud. Dans : Handbook of Cloud Computing. Springer, 2010, p. 357–377
- [B7] I.P. Egwuotuoha, Shiping Chen, D. Levy, B. Selic et R. Calvo. Energy Efficient Fault Tolerance for High Performance Computing (HPC) in the Cloud. Dans: Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. 2013, p. 762–769

**[B8]** Pankaj Goyal. Enterprise usability of cloud computing environments : issues and challenges. Dans : Enabling Technologies : Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on. IEEE. 2010, p. 54–59

**[B9]** S. Subashini et V. Kavitha. A survey on security issues in service delivery models of cloud computing. Dans : Journal of Network and Computer Applications 34.1 (2011), p. 1–11. Doi <http://dx.doi.org/10.1016/j.jnca.2010.07.006>

**[B10]** Ahmad-Reza Sadeghi, Thomas Schneider et Marcel Winandy. Token-Based Cloud Computing. Dans : Trust and Trustworthy Computing. Sous la dir. d'Alessandro Acquisti, Sean W. Smith et Ahmad-Reza Sadeghi. T. 6101. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, p. 417–429. doi : 10.1007/978-3-642-13869-0\_30

**[B11]** Ravi S Sandhu, Edward J Coyne, Hal L Feinstein et Charles E Youman. Role-based access control models. Dans : Computer 29.2 (1996), p. 38–47

**[B12]** Ralph C. Merkle. A Digital Signature Based on a Conventional Encryption Function. Dans : A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology. CRYPTO '87. London, UK, UK : Springer-Verlag, 1988, p. 369–378

**[B13]** Roger M Needham et Michael D Schroeder. Using encryption for authentication in large networks of computers. Dans: Communications of the ACM 21.12

**[B14]** An Oracle White paper. L'information Life cycle Management for Business Data. <http://www.oracle.com/us/026964.pdf>. 2007

**[B15]** Thomas Wiedmann et Jan Minx. A definition of carbon footprint. Dans: Ecological economics research trends 2 (2007), p. 55–65

**[B16]** Ranjan Kumar, G.Sahoo Cloud Computing Simulation Using CloudSim. Dans : International Journal of Engineering Trends and Technology (IJETT) –Volume 8 Number 2–Feb 2014

[C1]Benini, L. and de Micheli, G. (2000). System-level power optimization : techniques and tools. ACM Transactions on Design Automation of Electronic Systems.

[C2]Suleiman, D., Ibrahim, M., and Hamarash, I. (2005). Dynamic voltage frequency scaling (dvfs) for microprocessors power and energy reduction. In 4th International Conference on Electrical and Electronics Engineering.

[C3]Zhao, B., Aydin, H., and Zhu, D. (2013). Shared recovery for energy efficiency and reliability enhancements in real-time applications with precedence constraints. ACM Trans. Des. Autom. Electron. Syst.

[C4]Valentini, G. L., Lassonde, W., Khan, S. U., Min-Allah, N., Madani, S. A., Li, J., Zhang, L., Wang, L., Ghani, N., Kolodziej, J., et al. (2013). An overview of energy efficiency techniques in cluster computing systems. Cluster Computing.

[C5]Shuja, J., Madani, S. A., Bilal, K., Hayat, K., Khan, S. U., and Sarwar, S. (2012). Energy-efficient data centers. Computing.

[C6]Elnozahy, M., Kistler, M., and Rajamony, R. (2003). Energy conservation policies for web servers. In Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems.

[C7]Srikantaiah, S., Kansal, A., and Zhao, F. (2008). Energy aware consolidation for cloud computing. In Proceedings of the 2008 conference on Power aware computing and systems.

[C8]Douglass, F., Krishnan, P., and Marsh, B. (1994). Thwarting the power-hungry disk. In Proceedings of the 1994 Winter USENIX Conference.

[C9]Gurumurthi, S., Zhang, J., Sivasubramaniam, A., Kandemir, M., Franke, H., Vijaykrishnan, N., and Irwin, M. J. (2003). Interplay of energy and performance for disk arrays running transaction processing workloads. In IEEE International Symposium on Performance Analysis of Systems and Software.

[C10]Weissel, A., Beutel, B., and Bellosa, F. (2002). Cooperative i/o : A novel i/o semantics for energy-aware applications. In Proceedings of the 5th symposium on Operating systems design and implementation. ACM.

[C11]Colarelli, D. and Grunwald, D. (2002). Massive arrays of idle disks for storage archives. In Proceedings of the 2002 ACM/IEEE conference on Supercomputing.

[C12]Schulz, G. (2008). Maid 2.0 : Energy savings without performance compromises. Greg Schulz, Jan.

[C13]Narayanan, D., Donnelly, A., and Rowstron, A. (2008). Write off-loading : Practical power management for enterprise storage. ACM Transactions on Storage.

[C14]Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D. (2008). Reducing network energy consumption via sleeping and rate-adaptation. In NSDI.

[C15]Nordman, B. and Christensen, K. (2005). Reducing the energy consumption of network devices. Tutorial presented at the July.

[C16]Gunaratne, C., Christensen, K., Nordman, B., and Suen, S. (2008). Reducing the energy consumption of ethernet with adaptive link rate (alr). Computers, IEEE Transactions on.

[C17]Catthoor, F., Greef, E. d., and Suytack, S. (1998). Custom memory management methodology : Exploration of memory organisation for embedded multimedia system design.

[C18]Inc., R. (1999). Rambus dram.

[C19]Delaluz, V., Kandemir, M., Vijaykrishnan, N., Sivasubramaniam, A., and Irwin, M. J. (2001). Hardware and software techniques for controlling dram power modes. Computers, IEEE Transactions on.

[C20]Li, X., Li, Z., Zhou, Y., and Adve, S. (2005). Performance directed energy management for main memory and disks. Transaction on Storage.

[C21]Lebeck, A. R., Fan, X., Zeng, H., and Ellis, C. (2000).Power aware page allocation. ACM SIGPLAN Notices.

[C22]David, H., Fallin, C., Gorbato, E., Hanebutte, U. R., and Mutlu, O. (2011). Memory power management via dynamic voltage/frequency scaling. In Proceedings of the 8th ACM international conference on Autonomic computing.

[C23]Fan, X., Ellis, C., and Lebeck, A. (2001). Memory controller policies for dram power management. In Proceedings of the 2001 international symposium on Low power electronics and design.

[C24]Sudan, K. (2013). DATA PLACEMENT FOR EFFICIENT MAIN MEMORY ACCESS. PhD thesis, The University of Utah.



[C25]Ganesh, L., Weatherspoon, H., Marian, T., and Birman, K. (2013). Integrated approach to data center power management. Computers, IEEE Transactions on.

[D1]S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, Cluster Computing 12 (2009) 1 15.

[D2]Anton Beloglazov et Rajkumar Buyya. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers . Dans : Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. ACM. 2010, p. 4

[D3]Haibo Mi, Huaimin Wang, Gang Yin, Yangfan Zhou, Dianxi Shi et Lin Yuan. Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers . Dans : Services Computing (SCC), 2010 IEEE International Conference on .IEEE. 2010, p. 514 521

[D4]A novel approach for distributed application scheduling based on prediction of communication events, Future Generation Computer Systems 26 (5) (2010) 740 752.

[D5]Santhosh R, Ravichandran, Preemptive scheduling of online real time services with task migration for cloud computing , publisher : IEEE, dans Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference , 271 276, 21-22 Feb. 2013.

[D6]E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power-Aware Computer Systems (2003) 179 197.

[D7]Lenk, A., Klems, M., Nimis, J., Tai, S., and Sandholm, T. (2009). What's inside the cloud ? an architectural map of the cloud landscape. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing.

[E1] [https://fr.wikipedia.org/wiki/Eclipse\\_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet)) Mardi 10 mai 2016

## Liste des figures

<b>I.1</b> : Les couches Cloud .....	10
<b>I.2</b> : L'emplacement d'un service PaaS .....	11
<b>I.3</b> : L'évolution d'utilisation des service de l'informatique en nuage en France depuis 2009 .....	20
<b>I.4</b> : Les principaux acteurs de cloud computing.....	22
<b>III.1</b> : Diagramme de changement d'état d'une ressource.....	45
<b>IV.1</b> : Architecture de GreenCloud .....	65
<b>V.1</b> : Résultats de simulation .....	72
<b>V.2</b> : Caractéristiques de centre de données .....	73
<b>V.3</b> : Caractéristiques réseau de centre de données .....	73
<b>V.4</b> : les détails de la consommation d'énergie .....	74
<b>V.5</b> Interface de changement de paramètres.....	75

## Liste des Tableaux

<b>I.1</b> : Modèle de contrôle dans différents service .....	17
<b>I.2</b> : Modèle de sécurité pour Cloud Compting .....	19
<b>II.1</b> :Présentation des P-states et leur couple Fréquence/Voltage ainsi que leur puissance associée, pour le processeur Intel Pentium M1.6GHz .....	41
<b>V.1</b> : les fonctionnalite de GreenCloud scripts .....	69
<b>V.2</b> : résultats de changement de paramètres .....	72
<b>V.3</b> : les différents résultats de simulation de chaque architecture .....	75
<b>V.4</b> : Comparaison entre deux méthodes de planification « Green » « RandRobin » .	76
<b>V.5</b> : résultats de simulation avec et sans Vms ( statics virtualisation ) .....	77

## Liste des abréviations

**QoS** : Qualite of Service

**SLA** : Service Level Agreement

**SLO** : Service Level Objective

**DVFS** : Dynamic Voltage and Frequency Scaling

**PUE** : Power Usage Effectiveness

**RADIUS** : Remote Authentication Dial In User Service

**SAML** : Security Assertion Markup Language

**API** : Application Programming Interface

**RBAC** : contrôle d'accès à base de rôles

**CER** : Coefficient d'énergie renouvelable

**SSD** : Solid-State Drive

**CMOS** : Complementary Metal Oxide Semiconductor

**DRPM** : Dynamic Rotations Per Minute

**WBEU** : write-back with eager update

**WTDU** : write-through with deferred update

**RAID** : Redundant Array of Independent (or inexpensive) Disks

**SRMap** : sample-replicate-consolidate mapping

**ALR** : Adaptive Link rate

**RDRAM** : Rambus Dynamic Random Access Memory

**DDR** : Double Data Rate

**SWF** : Standard Workload Format

**NS2** : Network Simulator

**DNS** : Dynamic Network Shutdown

**MTU** : maximum transmission unit

**GPL** : General Public License