MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

RAPPORT DU PROJET

Option : **Réseaux et Systèmes**

THEME : PROPOSITION D'UNE TECHNIQUE DE MIGRATION DANS LE CLOUD COMPUTING

Etudiants : Rabé Labo & Moussa Salifou Chapiou

Encadrante : Dr. Filali Fatima Zohra

Année Universitaire 2020-2021

Moussa Salifou

Rabé Labo,

Résumé

Pouvoir migrer une VM d'un hôte physique à un autre s'avère être quelque chose d'incontournable dans les data centers. Cela permet de délimiter clairement le matériel du logiciel, de faciliter la gestion des pannes, de répartir la charge de travail et surtout de pouvoir organiser la maintenance des machines. Lorsqu'un hôte est surchargé et qu'il n'est plus capable de répondre à la demande, il est nécessaire de migrer l'état de la machine virtuelle vers un hôte plus puissant, ou moins surchargé. L'état de la machine virtuelle inclut la mémoire volatile et non-volatile, l'état des CPUs virtuels, l'état des périphériques connectés ainsi que l'état des connexions actives. Il existe principalement deux types de migration : la migration à froid, qui nécessite la mise hors tension de la machine virtuelle lors de l'opération de migration pendant laquelle la machine virtuelle est inaccessible. La migration à chaud ou en direct, pour laquelle la machine virtuelle n'est pas mise hors tension et reste donc accessible. Notre travail consiste à étudier les techniques de migration existantes et proposer une technique permettant d'optimiser la migration à chaud dans un environnement de Cloud computing.

Mots Clés : *cloud computing, Datacenter, Virtualisation, Machine Virtuelle, Migration, Serveur, Allocation des Ressources.*

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

Abréviations

VM	Virtuelle Machine
PM	Physical Machine
DC	Data center
QoS	Qualité de Service (Quality of Service)
OS	Système d'exploitation (Operating System).
SLA	Contrat du niveau de service (Service Level Agreement).
VMM	Virtual Machine Manager
SQL	Server Query Language
CPU	Central Process Unit
IBM	International Business Machine

Introduction Générale

Le Cloud Computing est l'une des rares technologies de notre époque qui se veut être modélisable et offrir un service [1] dans le sens de la même mécanique technologique plutôt qu'un produit. Dans l'absolu, l'utilisateur des services cloud, ignore tout de l'emplacement physique ainsi que la configuration du serveur qui livre ces services. Généralement ces services sont de l'ordre de calcul, logiciel, stockage ou accès aux données. Pour assurer ce service délicat, on met en œuvre ce qu'on appelle des data centers, autrement dit des centres de données virtualisés à l'intérieur desquels sont installés des serveurs, équipements réseaux et des systèmes de réfroidissement bien sûr, car qui dit serveur dit système de réfroidissement. Cependant ces systèmes dont le rôle est de réfroidir sont des consommateurs d'énergie par excellence. Ils bouffent des quantités faramineuses d'énergie si l'on désire offrir un service à la hauteur des attentes de grands clients[2]. Cet enjeu de surconsommation d'énergie entraine des coûts supplémentaires à l'égard des prestataires de service ce qui n'épargne guère les utilisateurs en terme de coûts à leur tour. On se rendra compte qu'on contribue significativement dans l'émission du gaz carbone responsable de la pollution de la planète dont le sujet est au cœur des questions scientifiques et technologiques de notre époque. Il convient de le souligner que le problème de la consommation d'énergie que font face les data centers émergeait d'abord de l'inutilisation d'une grosse part des ressources physiques, s'ensuit de la surabondance des ressources physiques présentes dans les data centers. Une étude révèle que les serveurs gaspillent environ 70 % de leurs ressources même à l'état de ralenti [3]. C'est dans ce contexte d'extrême préoccupations que la Virtualisation s'annonce comme solution miracle pour permettre au cloud de résoudre le problème substantiel de l'inefficacité énergétique en partant du principe qu'il faille créer plusieurs instances de machines virtuelles VMs dans un même serveur à la fois. D'autre part, on réduit en utilisant la migration à chaud, le nombre de serveurs physiques sur lesquels on migre les VMs en tenant compte des demandes de ces machines virtuelles en terme de ressources comme la RAM, le stockage, ou CPU. De manière optimale un serveur sans VM est mis à l'état d'arrêt pour économiser de l'énergie. En revanche,

les fournisseurs du cloud se doivent de suivre les injonctions du SLA (service lève agreement) dans le cadre du respect de Qos (quality of service) vis-à-vis de leurs clients. Les environnements de cloud travaillent pour trouver le bon équilibre entre les deux.

C'est au milieu de cette problématique que notre nos travaux de recherche se focalisent dans l'optique de trouver des solutions de migration avantageuse pour les data centers. Notre challenge va être comment migrer en sureté et en minimisant la consommation d'énergie sans empiéter sur les performances. Objectivement contribuer au final à la théorie de Green Computing (Informatique Verte).Dans le sillage de ce travail,il est question d'un travail de fond sur les approches et stratégies de migration dans les data centers. Une proposition sera faite à cet effet. Pour ébaucher cette solution nous avons allons mettre en avant la technique de la migration à chaud des machines virtuelles afin de l'utiliser comme technique de base et piedestale dans la réalisation de notre propre solution.

Organisation du travail

Ce mini-project est structuré en chapitre binaire et conclusion binaire dont la dernière servira de conclusion générale.Nous avons comme suit :

- **Chapitre 1 :** Dans ce partie majeure de notre mini-projet, nous avons porté un éclairage sur les concepts de base autour de la virtualisation. Nous avons decris les spécificités de ces concepts ainsi que leurs rôles déterminants dans le cloud.Nous avons évoqué le problème de surconsommation dont font face les data centers. Par la suite nous avons présenté les solutions et stratégies existantes pour surmonter un tant soit peu ces problèmes.
- **Chapitre 2 :** Ce chapitre met l'accent essentiellement sur les techniques de migration de machines virtuelles, la virtualisation et ses caractéristiques, autrement dit les avantages et inconvénients. Nous y avons cité égalemnt quelques travaux scientifiques des chercheurs qui ont contribué aux recherches sur cette thématique dans leurs propositions de techniques et solutions de migrations de VMs.
- **Chapitre 3 :** Ce présent chapitre concerne la partie conception de la solution que nous proposée. Nous y avons fait mention des approches que nous avons proposées dans le cadre de la mise en oeuvre d'une solution de migration. Nous avons défini deux politiques de sélection de VM, une politique d'allocation. Plus loin, nous avons décris le modèle de notre système. Ensuite avons discuté des métriques que nous avons jugées convénable de permettre à évaluer nos propositions, en l'occurence les métriques de réduction du nombre

de migrations de VMs, de consommation d'énergie et de pourcentage de violation de SLAs.

- **Chapitre 4 :** Dans cette partie qui met fin au travail de recherche que nous effectué, il en découle de l'analyse et discutions des résultats que nous avons obtenu suite aux séries de simulations de nos algorithmes dont il est question d'étudier les performances. C'est ainsi que sont rapportés les résultats et les explications de telles expérimentations.
- **Conclusion gnérale :** Nous avons bouclé ce projet avec une conclusion générale,qui de manière retrospective nous rappelle les étapes passées qui ont conduit aux résultats globaux des expériences ménées. Dans conclusion nous avons fait le point sur les spécificités de nos politiques proposées et nous avons dégagé les perspectives que nous projetons d'étudier dans les travaux futurs.

Chapitre 1

Concepts de Virtualisation

1.1 Introduction

La virtualisation est la technologie mère qui a porté haut le Cloud Computing et lui a donnée toute son essence. Elle a établi une utilisation rationnalisée et optimisée des ressources matérielles dans l'optique de faire tourner en parallèle plusieurs systèmes dits virtuels utilisant la même ressource physique tout en y appliquant une couche permettant d'abstraire le matériel physique. Le pionnier indiscutable de la technologie de virtualisation est IBM, qui initia depuis les années 1960 le concept de virtualisation. Des avancées qui leur ont permis d'aboutir à la présentation du tout premier hyperviseur [4]. La virtualisation est une technique qui offre la possibilité de simultanéité de fonctionnements à plusieurs systèmes d'exploitation, distincts dans un même environnement ou machine physique PM. A l'image du système linux, la virtualisation gagne du terrain au sein des serveurs modernes et est présente quasi partout dans les data centers modernes. Ceci s'explique notamment par les avantages qu'elle offre, la tolérance aux pannes, le partage, la portabilité et la réduction des coûts [5]. Avec la virtualisation, on a la notion de partage des différentes ressources physiques entre les machines virtuelles qui fonctionnent en parallèle les unes avec les autres de manière indépendantes dans l'environnement virtualisé. La migration de VM est l'une des prouesses techniques la plus élégante inventée avec la virtualisation du fait qu'elle permet de migrer les machines virtuelles ainsi que leurs environnement d'exécution vers des data centers, serveurs ou vers des machines physiques et ce avec une transparence qui caractérise la fonction. Cette fonction de migration, joue un rôle déterminant dans la gestion des ressources et applications dans un système virtualisable[6]. Nous faisons le point sur la virtualisation et ses principes dans ce chapitre pour amorcer dans la foulée les différentes

techniques de migrations et ses apports pour l'optimisation.

1.1.1 Virtualisation

C'est la virtualisation qu'est le moteur sous-jacent du Cloud Computing, secondée par le web 2.0 et l'accès disponible de la bande passante à travers l'internet. Il y a un double avantage ici d'abord le fait de pouvoir exécuter plusieurs systèmes d'exploitation sur des machines virtuelles distinctes, qui quant à elles se trouvent sur la même machine physique appelée Serveur, ensuite le deuxième avantage est lié aux coûts qui sont beaucoup plus réduits du fait de la gestion rationnelle du matériel qui caractérise la virtualisation. Par-delà, la consommation d'énergie également baisse. Il convient de rappeler que le géant de technologie IBM, dans l'optique d'inter échange avec tous les serveurs de l'entreprise, ont inventé la virtualisation que nous connaissons aujourd'hui. Les utilisateurs finaux qui sont sur les machines virtuelles ont l'illusion d'être en contact direct à la vraie machine physique(virtualisée). Sous l'effet d'abstraction on crée une instance de la machine physique, qui sera jointe dans l'environnement virtualisé. Avec les machines virtuelles crées on surmonte le problème de coût élevé de matériel car, les VMs ont la capacité de partager les ressources matérielles. D'autre part un gain de productivité pour l'entreprise car les utilisateurs se partagent la même machine physique dans le cadre de leurs besoins respectifs. L'ensemble de ce processus se déroule dans la transparence pour permettre aux utilisateurs ou application d'interagir avec les ressources, voir les détails de l'implémentation. Nous avons sur le serveur physique, un système d'exploitation principal installé dit « système hôte » qui va héberger d'autres systèmes. Un hyperviseur ou logiciel qui permet de faire la virtualisation se situe donc entre les deux couches matérielles. Il est installé au niveau du système principal. Avec l'hyperviseur on peut créer d'autres environnements indépendants sur lesquels on peut installer des systèmes dits « invités ». Les systèmes d'exploitation traditionnels n'avaient pas la capacité à bien traiter le problème de l'isolement, c'est tout l'opposé de la virtualisation dans un contexte de cloud. Elle permet l'exécution d'un logiciel "exécutable" sur le matériel, d'être exécuté sur une machine virtuelle. Elle utilise l'encapsulation des services cloud dans les machines virtuelles [7]. Encore une fois, la virtualisation permet de créer des objets virtuels jusqu'à créer des ressources virtuelles dans un but d'optimisation de coûts.

1.1.2 Hyperviseur

Un hyperviseur ou moniteur de machine virtuelle (VMM) est une « couche logicielle qui virtualise toutes les ressources d'une machine physique, et prenant ainsi en charge l'exécution de machines multi-virtuelles (VM) » [8]. En 1974, Il a été défini un certain nombre de propriétés s'appliquant sur VMM [9] dont entre autre l'efficacité, équivalence et contrôle des ressources. On en compte deux types de VMM, type I et type II[10], voir Figure 1.1. Un hyperviseur de type I gère toutes les VMs et se lance d'emblée sur le matériel ayant le plus haut privilège. Ce type d'hyperviseurs, sont qualifiés de natifs et supportent toutes les versions de VMs classiques. VMware ESX, Hyper-V et Xen [11] en sont des exemples de type I. Les hyperviseurs de type II, sont appelés hébergés(hosted). Ils fonctionnent sur un système d'exploitation avec l'anneau du système d'exploitation hôte et gèrent les VMs invitées. VMware Server et VirtualBox en sont un exemple. Les hyperviseurs de type I gèrent la mémoire, le processeur les entrées-Sorties E/S grâce à leurs composants et admettent un planificateur de VM pour la gestion de CPU pour chaque instance. En revanche les hyperviseurs de type II, reposent sur le planificateur natif du système d'exploitation pour dire que chaque VM en exécution est un processus du système d'exploitation.

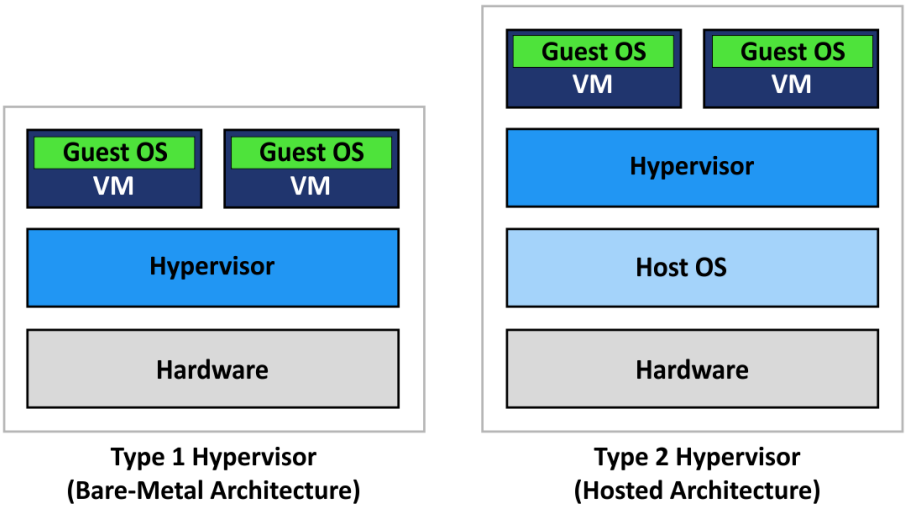

Figure 1.1 – Hyperviseur type 1 et type 2

[12]

1.1.3 Machine Virtuelle

L'intérêt premier de disposer des machines virtuelles est pour exercer le partage des ressources matérielles aux coûts exorbitants. L'idée est que toutes les machines se partagent les ressources matérielles qui leur sont communes. Parallèlement une machine virtuelle n'est que la forme virtuelle de la machine physique, en ce sens où les mêmes caractéristiques de la machine physique sont retrouvées dans la machine virtuelle sous forme des fichiers et caractéristiques virtuelles. La machine virtuelle héberge en son sein un système d'exploitation qui permet aux applications installées sur la machine virtuelle de fonctionner comme si elles étaient installées sur la machine physique grâce aux ressources impérieusement configurées. On avait dit qu'une VM est une structure de fichiers, les fichiers les plus importants sont :

- Fichier de configuration : dans lequel on retrouve les différentes ressources accessibles à la VM ainsi que l'ensemble de matériel virtuel qui constitue la machine virtuelle (CPU, réseau, stockage, mémoire etc.).
- Les fichiers de disque virtuel.
- Fichiers logs.

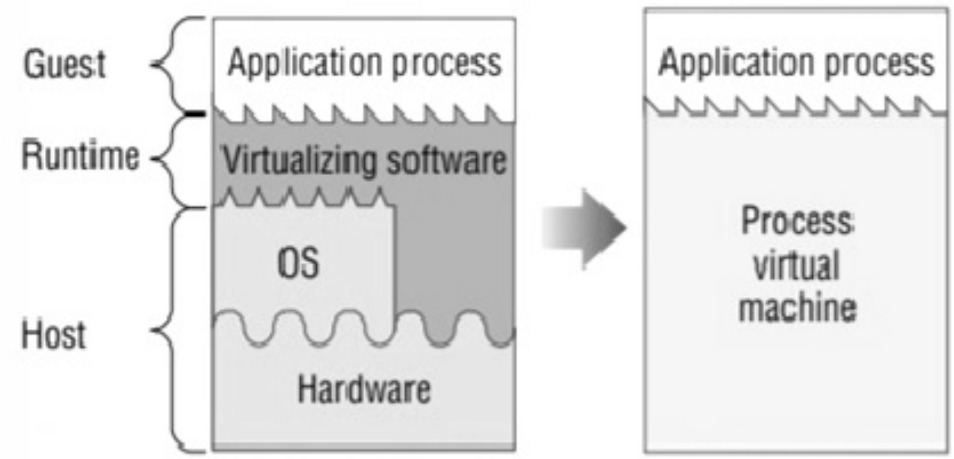

Figure 1.2 – Machine Virtuelle

[13]

1.1.4 Data Center

Un data center en langage de cloud computing ou centre de traitements de données en langage courant, est une loge physique à l'intérieur de laquelle sont truffés beaucoup d'équipements formant le rouage du système d'information d'une entité. On énumère au sein des data centers

des ordinateurs centraux, serveurs rangés dans des baies, équipement réseau, baie de stockage, équipement télécom etc.[14]. Le site data center représente le rouage de l'entité organisationnelle, de ce fait il est impératif qu'il soit ultra protégé contre toute sorte de menace. A titre d'exemple on en cite ceux du pentagone, NASA, SpaceX, Facebook etc. le nombre de data centers disséminés à travers le monde dépasse les 23 000 data centers[15] .

Figure 1.3 – centre de données(Data center)

[16]

1.2 Les techniques de Virtualisation

Savoir maîtriser les techniques de virtualisation de base est avant tout incontournable pour pouvoir utiliser la virtualisation dans un environnement Cloud. Parmi les techniques les plus utilisées de virtualisation figurent celles-ci : l'émulation, la virtualisation complète, la paravirtualisation[17].

1.2.1 Emulation

L'émulation est la forme de virtualisation la plus ancienne et la plus basique. Le logiciel responsable de la virtualisation fournit au système invité toutes ressources virtuelles complètes (carte vidéo, réseau, CPU, interface). L'architecture du système invité peut être distincte de celle de l'hôte. Pour que les instructions CPU du système invité soient comprises par le CPU

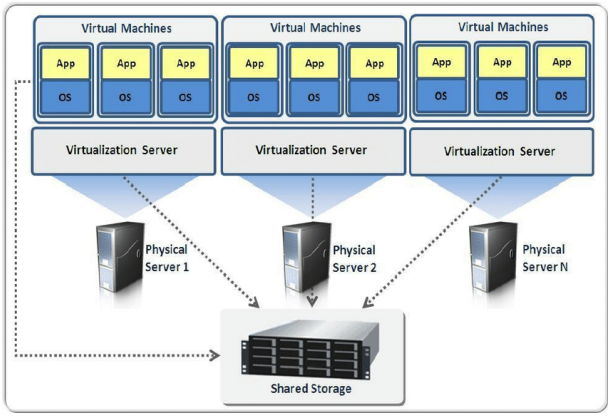

Figure 1.4 – Concept de la Virtualisation

[18]

de l'hyperviseur, il faut qu'elles soient converties. C'est ainsi que les systèmes invités font face à l'illusion d'accéder directement au matériel. (Voir Figure 1.5). Grâce à cette approche, on atteint l'isolement des systèmes invités les uns des autres. Exemple : virtualbox est l'émulateur classique.

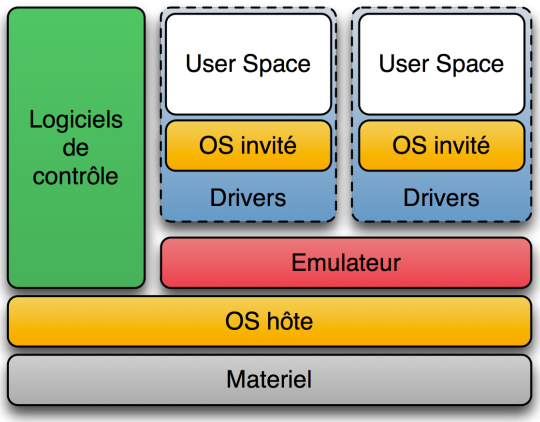

Figure 1.5 – : émulation

[19]

1.2.2 Virtualisation Complète

Là également, dans le cadre de la virtualisation complète[20]on va abstraire en entier un matériel virtuel au système invité. Le système invité présent ignore absolument son état virtuel dans l'environnement et s'exécute comme étant une machine physique. Le système hôte aussi bien qu'invités sont en phase c'est-à-dire qu'il doivent être compatibles les uns avec les autres à l'égard

de leur architecture matérielle. Cette solution offre également une forme d'isolation entre les machines virtuelles qui s'exécutent en parallèle sur les serveurs et notons le bien sans nécessiter aucune modification au préalable. C'est l'approche la plus appliquée dans les systèmes complexes. Les hyperviseurs qui implémentent cette technologie sont les plus connus : Virtualbox, VMware Workstation.

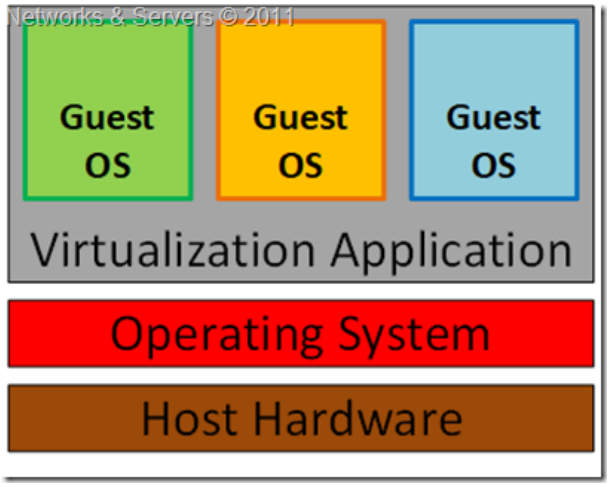

Figure 1.6 – virtualisation complète

[21]

1.2.3 Paravirtualisation

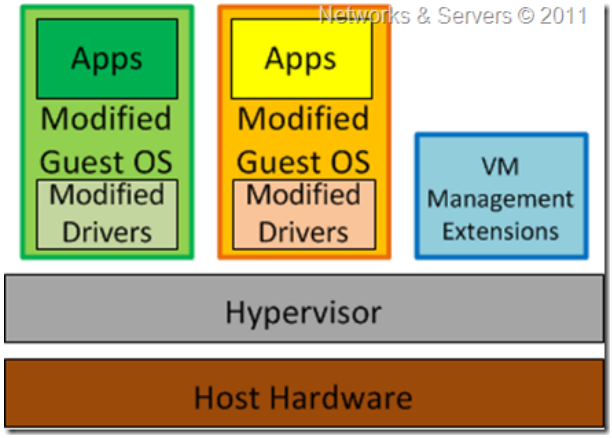

Figure 1.7 – Paravavirtualisation

[22]

Xen [1] est à l'origine du concept de paravirtualisation. Avec l'accessibilité à des logiciels libres permettant ainsi de personnaliser et d'adapter à son système le noyau linux, solaris ou BSD.

1. http ;//www. xenprojet..org / consulté le 24/02/2021

Hyper-V de Microsoft est un concurrent dans le domaine. Un compilateur est utilisé pour établir la paravirtualisation avec les systèmes d'exploitation invités. Le rôle du compilateur est de s'occuper des instructions non virtualisables pour les faire correspondre par des hyperapels (comme sur la Figure 1.7[22] . L'accès aux ressources est plus rapide en paravirtualisation. Cette technique nous permet d'économiser des ressources. L'unique modification sur l'OS invité permet d'améliorer les performances. Les hyperappels modifient parfois, le noyau du système invité. Dans les plateformes open source, c'est la paravitualisation qui est mise en avant. Il y a des exemples de produits disponibles qui implémentent l'architecture de paravirtualisation [23] comme Xen, VMwareESX. En paravirtualisation, on gère les instructions pendant la phase de compilation contrairement à la virtualisation complète. Le défi de la paravirtualisation est la compatibilité dans la mesure où elle prend en charge un système non modifié. Un autre défi est lié au coût faramineux de la gestion des systèmes d'exploitation paravirtualisés du fait qu'ils conservent les modifications sur le noyau d'OS. Néanmoins la paravirtualisation est plus optimale par rapport à la virtualisation complète et plus facile à mettre en œuvre.

Type	**methode**	***hôte* modif.**	**usage**
Paravirtualisation	*hyperappel par le noyau hôte*	oui	même charge de travail et OS
Virtualisation Complète	*traduction d'in-structures*	non	en cas d'abstraction complète
Assistéé par le matériel	*traduction à l'aide dumatériel*	non	même usage que la virt.Complète avec compatibilité de CPU
Hybride	*traductions et* changemnt de pilôtes	seulement le pilôte	pour installer de nouveaux pilôtes

Table 1.1 – comparaison des types de virtualisation

1.2.4 Avantages de la Virtualisation

Nous faisons mention de quelques avantages propres à la virtualisation[24].

- Diminution du nombre de matériels et la facilité de maintenance des périphériques car plusieurs systèmes d'exploitation peuvent fonctionner sur un seul serveur unique.
- Les taches de migration et de déploiement des machines virtuelles d'une machine physique à une autre
- Avec un nombre de serveur physique réduit, cela diminue significativement la consommation d'électricité et de climatisation.
- **Déploiement facile et rapide de nouveau serveur :** En quelque minute le déploiement est effectué au lieu de plusieurs jours comme c'était le cas jadis avec les systèmes non virtualisés.
- **Surveillance des serveurs plus avancée :** grâce aux outils virtuelles présents dans les machines virtuelles.
- **La portabilité :** grâce aux techniques de migration.
- **L'administration des machines virtuelles moins fastidieuse :** par rapport à l'administration classique des serveurs ce qui dispense les administrateurs des charges d'administration.
- Meilleure compatibilité des logiciels avec les systèmes d'exploitation

1.2.5 Inconvénients

Bien qu'elle présente assez d'avantages, il n'en demeure pas moins qu'elle recèle quelques inconvénients, comme on peut les voir ci-après.

- **Coût important :** il faudra miser énormément pour disposer d'un serveur physique de haute performance pour garantir de la virtualisation à la hauteur des besoins.
- **Pannes généralisées :** quand le serveur physique s'écroule, toutes les instances des machines virtuelles aussi se perdent. Il faudra tout reprendre.
- **Vulnérabilité généralisée :** si l'hyperviseur est sensible et en brèche, le problème va se répercuter sur toutes les machines virtuelles.

1.3 Conclusion

A travers l'analyse que nous avions effectuée des data centers dans l'environnement cloud tout au long de ce chapitre nous a permis de dégager une conclusion sur l'impact des data centers dans la consommation d'énergie et par ricochet la contribution des data centers à l'émission du C02. Nous y avons fait le point sur les composants du data centers, qui sont responsables directement de la forte consommation d'énergie. Il a été abordé superficiellement la virtualisation comme technologie sous-tendant le cloud. Beaucoup d'entreprises, en l'occurrence les géants du net se consacrent à la recherche des meilleures techniques d'optimisation d'énergie.

Ensuite dans le chapitre qui va suivre, nous allons parler de la migration et de la virtualisation, une des techniques utilisées pour diminuer la consommation d'énergie dans les data centers.

Chapitre 2

Migration

2.1 Introduction

Le mécanisme qui permet de transférer une entité d'un serveur physique hôte ou d'un emplacement vers un autre, s'appelle la migration. L'entité mise en migration peut s'agir d'une machine virtuelle, un disque virtualisé, processus ou même des objets de base de données. Cette migration est l'instrument fondamental de gestion d'application et de ressources dans les environnements titanesques comme le cas du cloud. La vocation de la migration tient à la réduction de la consommation d'énergie, équilibrage de charge ou encore la tolérance aux pannes. Grâce à l'hyperviseur qui rend l'isolation du système d'exploitation du matériel possible, pour permettre à plusieurs instances de fonctionner indépendamment les unes des autres avec leurs applications respectives, la migration valorise largement cette virtualisation des VMs. Les composants de la machine tels que le CPU, la mémoire ou les entrées/sorties sont transférés ensemble lors de la migration.

2.2 Techniques de migration des machines virtuelles

Il existe principalement deux types de migration [25] qui sont : la **migration à chaud** (Live Migration) et la **migration à froid** (Non live Migration).

Comparaison		
Tech de Migration de VM	**Avantages**	**Inconvénient**
Non-Live(Migration à froid)	facile à implementer.Type de migration pour déplacer toutes les données en même temps	temps d'arrêt important, ce qui est un inconvénient majeur[26]
Live Migration(Migration à chaud	migrer la VM sans interruption système d'exploitation invité pendant le processus[27]	provoque une défaillance du système, duplique la transmission et consomme beaucoup de temps. La machine virtuelle a un temps d'arrêt lors du transfert

Table 2.1 – comparaison entre les techniques de migration de VMs

2.2.1 Migration à froid (stop and copy Migration)

Une migration à froid est une technique simple à mettre en pratique où on éteint la VM sur la machine source pour transférer sa mémoire et son exécution vers une machine destination. A la fin du transfert des pages mémoires vers la machine destination, on enchaine à l'activation de la VM sur la machine destination. Il convient de préciser que la machine virtuelle n'est activée qu'après le transfert complet des pages mémoires. Il est possible avec ce type de migration de transférer les disques d'un data center vers un autre. Par contre il n'est avisé que les machines virtuelles soient sur un support de stockage partagé. Le processus [28] de migration à froid se déroule de cette manière :

- Tous les fichiers de configuration, le fichier NVRAM, les disques de la VM, les fichiers journaux sont intégralement migrés de la machine hôte à la machine destination.
- La machine virtuelle est activée sur le nouvel hôte destination.
- Après l'opération de migration, l'ancienne instance de la machine virtuelle existante sur

l'hôte source est détruite.

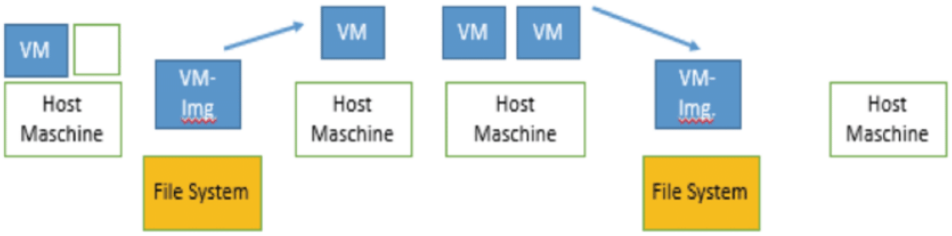

Figure 2.1 – la migration à froid

[29]

2.2.2 Migration à chaud (Live Migration)

La migration à chaud dite migration live est la plus intéressante dans la mesure où elle permet de migrer efficacement des systèmes d'exploitations complets avec leurs applications d'une machine physique à une autre éloignées des data centers et des clusters [30]. C'est la technologie la plus optimale à bien des égards en ce sens qu'elle favorise une faible consommation d'énergie, la maintenance de système, gestion des pannes et l'équilibrage de charges. La particularité de la migration à chaud est que le processus de migration des VMs d'une machine physique à une autre n'interrompt pas les autres VMS. D'où son avantage extraordinaire dans plusieurs cas de figures [31] comme suit :

- Lorsqu'un nœud physique est défaillant, la machine virtuelle peut être transférée vers un nœud physique sain.
- Lorsqu'un nœud physique est surpeuplé de VMs, certaines d'entre ces VMs sont migrées vers d'autres nœuds pour justement équilibrer les charges.
- Lorsque des VMs sont au mode repos dans un nœud physique, elles doivent être migrées vers d'autres entités physiques pour optimiser les ressources.

On distingue deux types de migration à chaud, tous basés sur le processus de la copie de la mémoire : **Post-Copy** et **Pré-Copy**.

- **Live Migration « Pré-Copy »** : VMware de même que Xen utilisent cette technique avec l'hyperviseur. Une copie de VM est entièrement déplacée vers la machine destination au cours de la première itération ensuite, génère d'autres itérations pour transférer les pages mémoires modifiées. Les itérations ne s'arrêtent qu'au moment où assez de pages mémoires eussent été copiées pour permettre à la machine virtuelle de fonctionner sur

l'hôte destination. Comme le montre la Figure 2.2, la migration type Post-Copy se déroule suivant les étapes suivantes :

- **Etape 0 : Prémigration.** Une machine virtuelle est en exécution sur une machine physique donnée ''A''.
- **Etape 1 : Réservation.** Un appel de migration est envoyé par un hôte physique source à un hôte destination et attend sa confirmation de disponibilité de ressources ainsi, l'hôte de destination réserve les ressources disponibles.
- **Etape 2 : Copie itérative de pages mémoire.** Durant la toute première itération, toutes les pages mémoire de la VM source sont copiées vers l'hôte de la machine destination précisément à l'emplacement mémoire alloué pour cette VM. Par cette première itération, les prochaines qui vont suivre, prendront en compte que les pages mémoires modifiées (dirty pages) pendant la phase de transfert précèdent.
- **Etape 3 :Stop and Copy** : pour permettre la copie des pages mémoires modifiées, la machine virtuelle est suspendue. Les registres CPU, l'état des périphériques sont également transférés sur la machine de destination. Une fois le transfert achevé, la machine virtuelle cible qui se trouve à présent sur la machine destination devient une copie parfaite de la machine virtuelle source.
- **Etape 4 : Redirection :** C'est l'étape de confirmation par l'hôte destination de la réception de l'image de la VM à l'hôte source.
- **Etape 5 : Activation :** Maintenant la machine virtuelle migrée peut être exécuter sur l'hôte destination.

■ **Live Migration « Post-Copy »** : Dans ce type migration, la VM n'est pas mise à l'arrêt mais elle est suspendue dans la machine source, son état d'exécution (pages mémoires indispensables pour lancer la VM dans la machine destination, CPU, registres) est copié vers la machine physique de destination. Ce qui est intéressant est que la machine virtuelle encours de transfert vers la machine physique peut d'ores et déjà commencer son exécution sur la machine cible sans attendre la fin de la copie des pages mémoires. Le seul inconvénient de cette technique est l'apparition de défauts de pages très souvent manifestés dans la machine destination. Lorsque les pages mémoires de la VM à migrer ne sont pas disponibles sur notre machine destination cela nous conduit à ce problème de défauts de page.

Note : La technique de pre-copy est meilleure que la méthode post-copy dans la mesure où

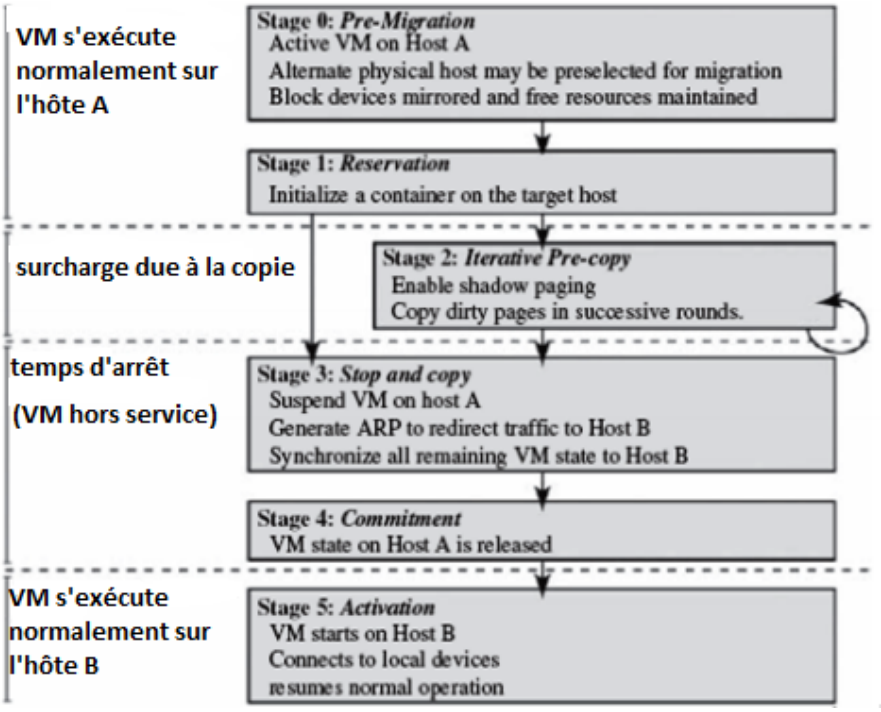

Figure 2.2 – live migration

[32]

elle permet la récupération d'une VM en cas de panne grave sur la machine destination car une copie mémoire avec l'état du CPU de la VM est préservée dans un emplacement sûr au niveau de la machine source. Pour connaitre les performances de la technique à chaud de migration de VM, on utilise des métriques [31] comme :

- **Préparation :** : étape de préparation de l'environnement pour la machine virtuelle, la machine destination prépare les ressources nécessaires pour la ''VM invitée ''. D'autres configurations sont appliquées depuis la machine source.
- **Temps d'arrêt :** moment de mise à l'arrêt de la machine VM sur la machine source.
- **Temps résumé :** : la machine virtuelle s'exécute sur la machine physique destination avec le même état que celui de la machine source.
- **Temps total de migration :** :c'est le temps total qu'il faut pour réaliser toutes ces étapes précitées.

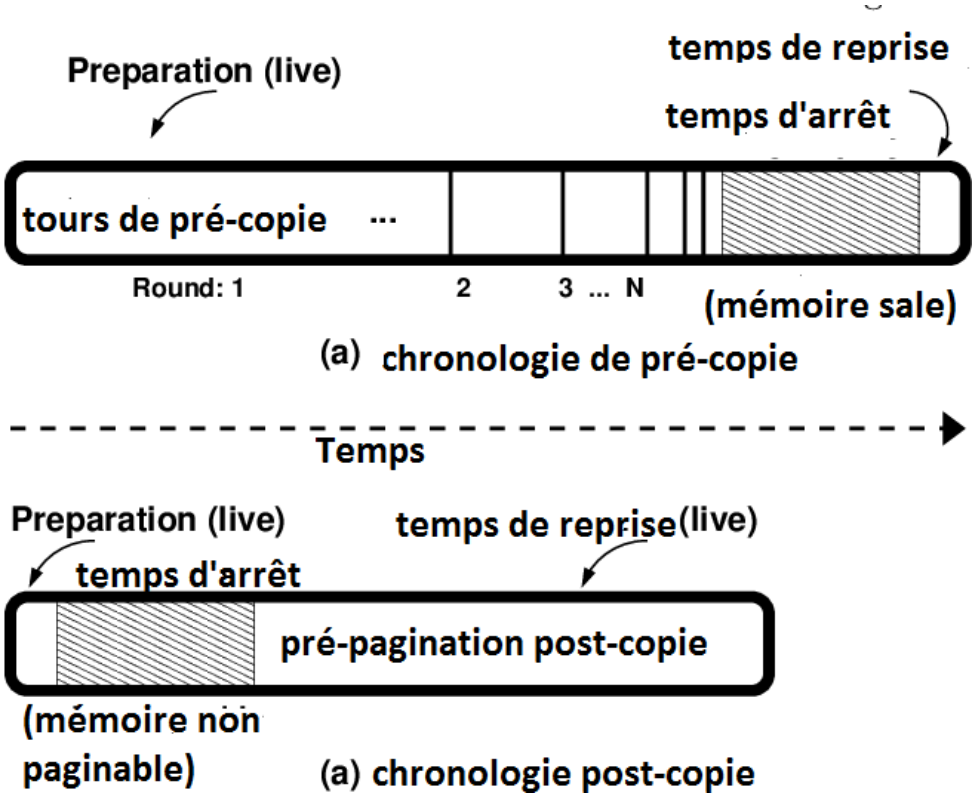

Figure 2.3 – post-copy live migration

[33]

2.2.2.1 Comparaison entre la migration de processus et la migration de VMs

La différente la plus frappante entre les deux les types est au niveau de l'entité à migrer. Pour migrer un processus, il suffit de copier l'image mémoire du processus en exécution pour la transplanter dans la machine de destination alors que pour le cas de la machine virtuelle l'opération est accentuée et lourde. Il faudra transférer l'image de la VM complète, une opération couteuse en temps et en mémoire [7]. La migration de processus est plus aisée par rapport à la migration de VM en vue de la synchronisation de la mémoire. Pourtant, avec la migration les applications de la VM peuvent être exécutées sans soucis car l'ensemble des bibliothèques et autres fichiers indispensables au bon fonctionnement de ces applications va migrer avec les VMs. Tandis qu'ailleurs avec la migration des processus [34] des limitations s'imposent au niveau du traitement de la mémoire partagée. Avec la migration des processus on ne peut pas utiliser l'espace mémoire d'un processus sur un nœud lors le processus en quelque part en exécution dans un autre hôte. Du coup sur l'hôte destination, un espace mémoire provenant d'un nœud source peut être inutilisable. De toute évidence la migration des machines virtuelles est plus productives et facile à mettre en œuvre que la migration des processus.

tableau comparatif

Comparaison		
Entité	**Processus**	**Machine Virtuelle**
Synchronisation du contenu de mémoires	Réduite	Grande
Gestion du contenu descripteur	Compliqué	Plus facile
Continuité des connexion au réseau	Plus acile	Plus facile
Gestion des threads	Difficile	Facile
Vérification de l'environnement utilisateur	obligatoire	initule

Table 2.2 – comparaison VM migration versus processus migration

2.3 Avantages de la migration

Comme avantage on recense [35] :

- **Tolérance aux pannes :** : la machine virtuelle sous cet avantage peut continuer son activité même si une partie du système dysfonctionne. Cette technique prévoit les pannes pendant la migration.
- **Equilibrage de charge :** : la répartition de la charge de travail entre les serveurs.
- Réduction de la consommation d'énergie
- Adaptation automatique à l'évolution de la charge de travail.
- Elasticité et mise à l'échelle
- Possibilité de faire de la maintenance et d'entretiens sans interruption de services
- Mise à jour et mise à niveau transparente aux utilisateurs.

2.4 Inconvénients de la migration

- Problème de réseaux pendant l'opération de la migration
- Conflits d'adressages
- Problèmes de communication entre Vms après migration
- Problème de routage

Comparaison		
Tech de Migration de VM	**Avantages**	**Inconvénients**
Pré-copie(pre-copy)	pré-copie est la technique de migration Live typique la plus utilisée pour migrer l'état mémoire d'une VM[35]	La transmission en double surcharge d'une page mémoire prolonge encore le processus de migration.L'arrêt et la copy phase de Stop and Copy provoque un court temps d'arrêt court de VM pendant le transfert de page sale de la mémoire et l'état du processeur.
Post-Copie(Post-Copy)	post-copie peut réduire le temps total de migration, comme chaque fois la page mémoire est transférée uniquement une fois, donc réduction du temps d'arrêt	Crée un défaut de page au hôte de destination lorsque la machine virtuelle tente de récupérer les pages, qui ne sont pas envoyés.Il subit une dégradation de service en raison de défauts de page, qui doivent être résolus sur le réseau par l'hôte source[36].temps d'arrêt très court (milliseconde) lors de la migration du CPU et de l'état de la machine

Table 2.3 – comparaison entre les techniques de migration de VMs

2.5 Etat de l'art sur les techniques de migration des VMs

2.5.1 Classification des techniques de migration des VMs dans le cloud Computing

Les approches et les techniques de migration des machines virtuelles sont classées en deux grandes catégories : une catégorie qui orientée vers la recherche de l'amélioration du processus standard de migration et la seconde catégorie conserve l'état du processus pour améliorer les performances définies par les fournisseurs de cloud ou établies par les utilisateurs. (Dans notre cas, nous traiterons l'amélioration de processus).

Les travaux et recherches récents sur les techniques de migration sont nombreux, cependant nous fait le point sur ceux de :

A.Self-migration of Operating Systems : Jacob Gorm Hansen & Eric Jul [37], ces auteurs, proposèrent deux prototypes permettant non seulement de faire la migration d'application mais aussi celle des systèmes d'exploitation sur lesquels s'exécutent les applications. L'un des prototypes concerne une nouvelle approche qu'ils appellent self-migration pour les systèmes d'exploitation. Ayant fait le constat de la popularité de clusters pour des ordinateurs commodes pour des calculs scientifiques, ou les processus sont centralisés. La migration de processus n'a pas pu résoudre les problèmes de planification dans les systèmes distribués. Dans leur travail, Jaco et Jul ont étudié la faisabilité de changer l'unité de migration d'un processus unique vers l'ensemble du système d'exploitation. Ils estiment que la migration de OS serait une solution viable au problème de l'ordonnancement dynamique dans les réseaux à grande échelle et résout les problèmes de la sécurité. Pour étudier cette approche, ils ont mis en œuvre deux prototypes, l'un basé sur L4 micokernel, et un basé sur l'hyperviseur Xen. Leur première implémentation est basée sur NomadBIOS [38], qui est un environnement hôte capable d'exécuter plusieurs instances linux de manière concurrente avec la possibilité de migrer ces instances entre hôtes sans interruption de service. Ceci est le moins qu'on puisse dire de la migration live. NomadBIOS est au-dessus de L4 microkernel[39], Il contient une pile TCP / IP pour accepter les migrations entrantes et pour la migrer des systèmes d'exploitation invités. Pour réduire les temps d'arrêt, NomadBIOS se sert de la migration « pre-copy » [40]. Avec les migrations traversant des sous-réseaux, le système d'exploitation invité modifie son adresse IP, sinon, il utilise une couche d'indirection IP telle que IP Mobile [41]. La migration d'un OS avec plusieurs processus actifs, sur un réseau de 100 mégabits/s, provoque un temps d'arrêt moins d'un dixième de seconde. Le deuxième prototype

qu'ils ont utilisé est basé sur l'hyperviseur Xen[42]. Ils proposèrent d'appliquer la migration del'OS invité sans l'intervention d'un hyperviseur, en laissant le système d'exploitation invité se migrer contrairement à l'approche de l'équipe Xen qui est basée sur la migration pilotée par l'hôte. L'approche « self-migration » de ces auteurs présente plus d'avantages par rapport à la migration host-driven migration à savoir :

La sécurité : l'OS utilisera sa propre pile TCP et la base de confiance n'a besoin que de fournir une abstraction de réseau filtrée.

Comptabilité et performance : Les coûts liés au réseau et CPU pour effectuer la migration sont attribués à l'OS invité plutôt mais pas à l'hôte, ce qui simplifie la comptabilité.

Flexibilité et la portabilité. Le principal inconvénient de self-migration est le fait qu'elle nécessite une réimplémentation pour chaque type de système d'exploitation invité. Les auteurs ont effectué leur migration "self-migration" sur le port Xen existant de Linux 2.4, connu sous le nom de XenoLinux, étendu avec des fonctionnalités lui permettant de transférer une copie de son état complet sur un autre hôte physique, pendant que les opérations normales se poursuivent. La technique utilisée est une synthèse de l'algorithme de migration pre-copy avec la technique utilisée pour le point de contrôle dans les systèmes persistants. En général, la migration peut être effectuée en révoquant tous les mappages inscriptibles, en copiant tous les états et en renvoyant toutes les pages sales, d'une manière resend-on-write, jusqu'à ce que l'ensemble de travail ait été identifié, et les pages sales restantes peuvent être transférées après que le point de contrôle a été suspendu sur l'hôte d'origine. Alors que, dans un système self-migranting(auto-migrant), le transfert de l'état final est impossible, car le transfert lui-même touche l'intégrité de cet état, qui ne serait dans ce cas l'état final. Au lieu de cela, les auteurs décidaient de combiner les deux approches, en effectuant un renvoi en écriture jusqu'à l'identification du jeu de travail suivi de la copie sur écriture. Les auteurs ont pu effectuer une self-migration live sur un système d'exploitation de qualité comme Linux, tout en conservant les performances natives fournies par l'hyperviseur Xen, ce qui leur permet de minimiser la base de calcul de confiance, le code privilégié installé sur chaque hôte dans un système. L'utilisation d'un hyperviseur, avec une interface simple et de bonnes performances et l'isolement de sécurité, nous protège des utilisateurs malveillants, et en minimisant la base de confiance, leur offre l'avantage de créer une barrière plus élevée contre l'introduction de logiciels malveillants hôtes dans un système distribué. Leurs travaux futurs porteront sur la base de calcul fiable minimisée, ainsi que sur les mécanismes de sécurité et de comptabilité pour les systèmes décentralisés. Cost-Minimizing Dynamic Migration

of Content Distribution Services into Hybrid Clouds[43] : Dans le cadre de ce travail, les auteurs ont soulevé, le problème de la migration de service en préambule. Ils ont considéré une application de distribution de contenu typique, qui fournit une collection de contenus (fichiers), notée ensemble M, aux utilisateurs répartis sur plusieurs régions géographiques. Il y a un cluster de serveur sur site (ou cloud privé) qui appartient au fournisseur de l'application de distribution de contenu, qui stocke les copies originales de tout le contenu. Le fournisseur de l'application peut migrer les deux composants de service dans le cloud public : le contenu peut être répliqué dans des serveurs de stockage dans le cloud, dans le même temps les requêtes vont être adressées aux services Web installés sur VM des serveurs. Le framework que nos auteurs ont utilisé, sert à minimiser le coût opérationnel récurrent dans le système de distribution de contenu. D'autre part, leur framework applique les contraintes QoS prédéfinie. L'optimisation poursuivie par l'algorithme dynamique qu'ils ont appliqué minimise la moyenne temporelle du coût opérationnel tout en garantissant la qualité de service. Les auteurs ont utilisé un réseau de distribution de contenu privé ou public traditionnel (CDN), pour plus d'agilité et de réduction des coûts. Les auteurs se sont inspirés de la théorie d'optimisation de Lyapunov ,pour leur conception où la minimisation des coûts et la garantie du temps de réponse sont atteintes simultanément par une planification efficace de la migration de contenu et les requêtes sont dispatchées entre les datacenters. Ils ont ensuite développé un framework d'optimisation pour caractériser le problème de migration de service de distribution de contenu optimal. Ils ont utilisé des contraintes pour limiter le délai aller-retour moyen par demande avec une valeur qu'ils ont nommée . les auteurs ont travaillé sur un algorithme de contrôle dynamique utilisant les techniques d'optimisation Lya-punov, qui résout le problème de migration optimale et limite le délai de mise en file d'attente pour chaque demande. Ils ont à la fin analysé la garantie de performance fournie par leur algorithme dynamique. Ensuite ils ont suggéré de consulter leur rapport technique pour plus de preuves sur les théorèmes. Ils prévoient d'étendre leur framework de services de distribution de contenu spécifiques avec exigences détaillées, telles que les services de vidéo à la demande ou applications de réseaux sociaux, dans leur travail futur.

B.Live migration of virtual machines Migrating Memory pre-copy approach : Christopher Clark, Keir Fraser[35] ont implémenté un support de migration haute performance pour Xen. Ils s'attaquent à la migration des systèmes d'exploitation actifs hébergeant des services en direct, en minimisant les temps d'arrêt pendant lesquels les services sont entièrement indisponibles et prendre en compte le temps total de migration. Leur solution répond à toutes ces préoccupations, autorisant par exemple un OS exécutant le SPECweb de migrer sur deux hôtes

physiques avec seulement 210 ms d'indisponibilité. Ils sont parvenus en utilisant une approche pre-copy dans laquelle les pages de mémoire sont copiées de manière itérative à partir de la machine source vers l'hôte de destination, le tout sans jamais s'arrêter l'exécution de la machine virtuelle en cours de migration. Cela ne fait pas la migration dans toute la zone, et n'inclut pas non plus la prise en charge de la migration des périphériques de bloc locaux. Généraliser le transfert de mémoire en trois phases **(Push phase, Stop-and-copy, Pull phase)** pour respecter les compromis entre les exigences. Dans leur article, ils ont choisi la migration pré-copie, équilibre ces préoccupations en combinant une phase de poussée itérative limitée avec une phase stop-and-copy. Enfin, une préoccupation supplémentaire cruciale pour la migration live c'est bien l'impact sur les services actifs. Les auteurs voudraient qu'avec les ressources réseau, qu'un OS migré maintienne toutes les connexions réseau ouvertes sans compter sur les mécanismes de transfert sur l'hôte d'origine (qui peut être fermé vers le bas suite à la migration). Une VM en cours de migration inclura tous les états du protocole et garde son adresse IP avec elle. Lors de la migration live d'un système d'exploitation l'influence la plus significative sur les performances du service est la surcharge de transférer de manière cohérente l'image mémoire de la machine virtuelle. Pour tracer le comportement du jeu de travail inscriptible d'un certain nombre de charges de travail représentatives, ils ont utilisé les tableaux de page de Xen afin de dresser les statistiques de saleté sur toutes les pages utilisées par un OS particulier en cours d'exécution. Les auteurs ont exécutés ces benchmarks SPEC CINT2000, Linux kernel compile, OSDB OLTP utilisant Post-greSQL et SPECweb99 et Apache. Ils ont observé que les données de trace acquises permettent d'estimer l'efficacité de la migration itérative pré-copie pour différentes charges de travail.

C.Live virtual machines migration techniques . Energy Efficient Migration Techniques : P. Getzi et Dr. J. Sharmila[44] : nous dressent un sondage sur les techniques de migration de machine virtuelles. Tout d'abord, les auteurs de cet article présentent la technique à faible consommation d'énergie (Energy Efficient Migration Techniques). Ils montrent que les serveurs nécessitent environ 70% de leur consommation d'énergie même s'ils sont utilisés à leur capacité minimale. Ensuite ils ont abordé les techniques de migration d'équilibrage de charge. L'équilibrage de charge aide à minimiser la consommation de ressources, mise en œuvre du basculement, amélioration de l'évolutivité, évitant goulots d'étranglement et un excès d'approvisionnement des ressources, etc. la technique suivante est la Technique de migration tolérantes aux pannes, Cette technique fait migrer le machine virtuelle d'un serveur physique vers un autre hôte physique basé sur la prédiction des occurrences de panne. Selon ces auteurs, La technologie

actuelle clé pour un fonctionnement économe en énergie des serveurs est la migration live.

D. module de prédiction de l'état des nœuds. Construction et reconfiguration sensibles aux pannes de machines virtuelles distribuées pour un calcul à haute disponibilité : Sung Fu[45] : pour sa contribution réalisa un système qui permette d'augmenter la capacité et la disponibilité des serveurs physiques dans le but rendre la virtualisation plus performante au sein du cloud computing. Grâce à un module des machines virtuelles qui s'occupe de l'état des nœuds, le module de prédiction d'échec se saisit des informations de l'historique de données. Avec l'état de serveur on peut déterminer la durée de vie du serveur physique. Lorsque le module de prédiction d'échec envoie les prédictions, le coordinateur de VM déplace la machine virtuelle de ce serveur vers le serveur le plus proche dont l'état de santé est satisfaisant.

E. quantification des ressources avec un poids. Quantification du déséquilibre de charge sur les serveurs d'entreprise virtualisés : Emmanuel et David[46] leur approche quantifie un poids aux différentes les ressources utilisées par les serveurs. L'auteur souligne que l'hyperviseur planifie les ressources qu'utilisent les machines virtuelles. La charge du serveur virtuel (VSL) s'obtient en calculant les ressources utilisées au niveau de l'hyperviseur. Cette métrique est calculée pour chaque serveur et en fonction de ces valeurs VSL, la migration des VMs s'effectue d'un hôte fatigué vers un serveur moins surchargé.

F.Energy efficient Allocation of virtual machines La proposition des auteurs[47] de cet article est composée de trois stratégies, Minimization of Migration(MM) notamment pour la première,l'approche choisit un nombre minimum de VMS pour le migrer vers un hôte sous utilisé, ainsi une réduction de surcharge de migration s'accompagne. L'approche étudie l'utilisation du CPUs des VMS pour les classer par ordre décroissant, après cela un processus est mis en place pour trouver les VMs adéquates à la migration. Les auteurs ont implémenté un deuxième algorithme appelé Highest Potential Growth(HPG) cette approche vise à réduire l'utilisation des machines virtuelles et dimunier les violations SLA.Pour ce faire elle prend les VMs avec la valeur CPU la plus minimale afin de les migrer. La troisième en lice est Random Choice(RC) qui migre les VMs selon la valeur d'une variable discrète. La variable est uniformemment distribué. L'indexation de de la liste des VMs allouées à un hôte est dûe à cette valeur de la variable discrète.

G.Designing and Evaluating and Energy Efficient Cloud : Le travail de ces auteurs consistait de déterminer le coût[48] de l'énergie qu'une machine virtuelle peut générer par rapport au coût pendant l'exécution d'une tâche, lorsque la VM n'a pas de tâche, ainsi que le coût de

la migration. Ils n'ont pas manqué d'utiliser la virtualisation car il s'agit de système distribué à grande échelle. La proposition de ces auteurs est un système qu'ils ont appelé OGC(Open Green Cloud). L'interêt d'une telle architecture est de permettre de tourner à l'état "off" les resources comme stockage, de calcul ou des réseaux. Ils ont utilisé deux ordonnancement à savoir l'ordonnacement "Round-Robin" et "déséquiliibré". Ils ont effectué des expériences pour évaluer la quantité d'énergie consommée par les noeuds individuels du cloud. Quatre scénarios étaient nécéssaires pour chaque ordonnancement. Cependant le scénario de base n'utilise aucune technique de réduction d'énergie tandis que le scénarios "équilibré" utilise la migration. les noeuds inutilisés sont etteint dans le troisième scénarios alors que pour le scénarios "green", ils ont deécidé de bloquer les noeuds ensuite ils applique la migration pour l'équilibrage des charges entre les noeuds du cloud. Ils ont terminé les scénarios par une comparaison et c'est le scénarios "green" qui afficha les meilleurs résultats en terme de réduction d'énergie.

2.6 Conclusion

A l'issue de ce chapitre, il convient de retenir que la virtualisation a largement propulsé le cloud computing dans toutes ses ambitions. Les composantes fondamentales qui sous-tendent cette technologie sont les machines virtuelles. Ces machines hébergent des ressources virtuelles et des systèmes d'exploitation nécessaires permettant aux utilisateurs de travailler comme s'ils travaillaient sur des machines physiques. Dans le sillage des concepts de base de virtualisation que nous avions développé plus haut, il résulte que les techniques et les technologies de virtualisation concourent à la diminution de la surconsommation d'énergie et mettent un accent particulier dans l'amélioration des services.

Conclusion Générale

Le cloud Computing est une technologie qui conquiert le monde informatique au cours de ces dernières decennies.Le cloud revolutionne la fourniture de service informatique aux clients potentiels de façon decentralisée suivant un modèle pay-as-go qui stipule une consommation rationelle selon la volonté du client final. La demande se fait de plus en plus croissante pour les services cloud, ce qui augmente de façon logique la capacité d'exploitation des infrastructures des data centers pour la satisaction des clients. Cette augmentation a pour corollaire le problème de la gestion des ressources et la surconsommation dans les environnements de cloud. Dans l'optique d'amélioration de l'utilisation et de l'affectation de ressources, les leaders de la technologies cloud travaillent d'arche-pied sur les techniques telles que : la virtualisation et la migration de machines virtuelles que nous avons exploré dans le cadre de notre travail de recherche. Hormis les techniques précitées, nous esquiessons la technique de consolidation qui est permise dans la virtualisation. La virtualisation est la pierre angulaire du cloud dans dans la mesure où elle permet de créer plusieurs instances de VMs sur un même serveur physique, dont les avantages sont connus pour l'amélioration de l'utilisation des rssources. Elle est garante de la technique de migration à chaud, laquelle est au coeur de notre travail.

Les techniques de migrations des machines virtuelles sont nombreuses, beaucoup de travaux de recherche sont sur cette voie dans le but d'améliorer davantage les performances des serveurs dans les data centers en particuliers et dans cloud en général. Nous avons exploré plusieurs approches qui existent dans la littérature afin de nous guider vers la proposition de la nôtre que nous allons implémenter dans le cadre de notre projet de fin d'étude.

Chapitre 3

Conception

3.1 Introduction

Le problème de la consommation des ressources physiques,de latence, d'énergie est omniprésent dans les environnements cloud. D'innombrables travaux de recherche ont proposé des solutions plus ou moins efficaces pour réduire l'enjeu de la surconsommation de ces ressources et le temps d'indisponibilité. La plupart de ces travaux portent essentiellement sur la réduction d'énergie consommée. Dans le cadre de notre travail nous allons proposer les stratégies suivantes (MiMcHybrid,MmtMiMcHybrid,EBFDHybrid) qui sont en phase et corrélées aux différents problèmes soulevés dans la littérature basées sur la phase de selection et d'allocation en ce qui concerne la dernière.

Dans ce présent chapitre nous présentons notre modèle dans son architecture ainsi que quelques métriques qui nous permettent de tester les performances de ces approches. Par la suite nous dégagerons l'implémentation du modèle y compris les algorithmes et des graphes pour une lecture plus claire.

3.2 Architecture du modèle proposé

le modèle en question est un environnement Iaas, représenté par un data center à grande échelle. Le principe de son data center consiste à heberger un nombre conséquent d'hôtes hétérogènes dans lesquels vont s'exécuter des machines virtuelles. Gâce au virtual machine monitor, de multiples VMs peuvent être allouées au niveau de chaque hôte physique. Au-delà chaque hôte et VM se caractérisent par des métriques de performance de CPU illustrées en terme de millions

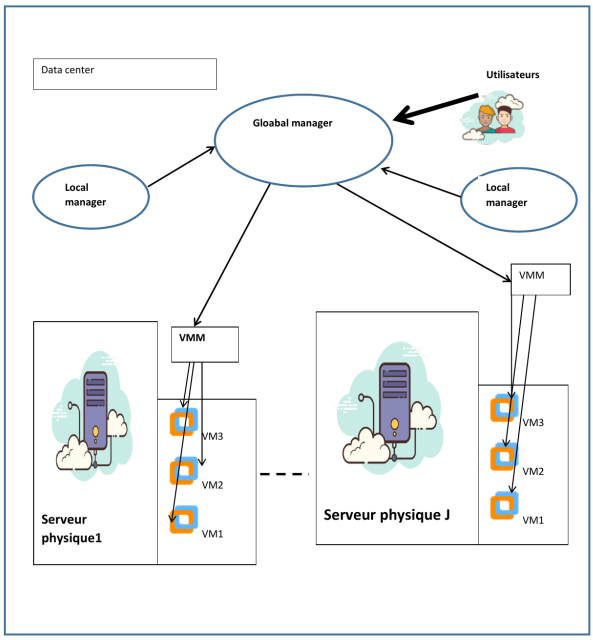

Figure 3.1 – Modèle du système

d'instructions par séconde(MIPS), la capacité de la RAM ainsi que la bande passante du réseau. la Figure 3.1 ci-dessus nous montre la réprésentation du modèle.

l'architecture de ce modèle comporte comme composants principaux Global Manager et Local Manager. Les requêtes des utilisateurs sont soumises pour la fourniture en VMs hétérogènes. Au sein d'un noeud physique réside le local manager qui sert de branche au virtual machine monitor(VMM). Les local Managers sont chargés de veiller à la supervisation de l'utilisation des CPU par les noeuds qui les hebergent.Ils gèrent aussi le redimensionnement des VMS en fonction des ressources dont elles ont besoin ou utilisées. Ils décident aussi de la VM à migrer et du moment qu'il faille faire la migration. Tant dis que les global Managers sont à l'intérieur d'un noeud gloabal ou noeud maître pour collecter les informations stockées au niveau des Local Managers afin d'avoir une vue générale de l'utilisation des ressources. C'est le global Manager qui ordonne l'optimisation pour le placement de VM. Il n'en demeure pas moins que les VMMs s'occuppent du rédimensionnement, migration de VMs et le changement de niveau d'énergie des noeuds. Les différents composants sont :

1. **Data center :** C'est un site physique constitué de machines physiques. Il est influencé

par sa bande passante, le stockage, processeurs, capacité de la mémoire.

2. **Broker :** C'est une entité jouant le rôle intermediare de gérer les VMs dans plusieurs Data centers ainsi que le routage du trafic vers les data centers appropriés. Il se charge également de sélectionner le data center optimal pour les requêtes des clients dont il prend en charge avec les data centers.
3. **Physical Node :** C'est la machine qui est physiquement dans l'enceinte des data centers et qui de facto doit contenir les VMs. L'hôte physique se caractérise par la vitesse du CPU et le nombre de cœurs, nombre de requêtes envoyé, le nombre de VMs, capacité de stockage.
4. **VMs :**Une VM est une instance ou environnement d'exécution sur une même machine physique pour permettre aux utilisateurs d'utiliser ces VMs comme des vraies machines physiques.
5. **VM Manager :** qui garantit l'utilisation des ressources et la disponibilité des machines virtuelles. C'est grâce au VMM que se fait l'affectation de VM d'une machine physique à une autre.
6. **Local Manager :** Ils sont presents dans tous les nœuds comme un module du VMM. Ils aident à rédimensionner les VMs par rapport au besoin, ils définissent la politique de migration de VM. Le suivi du processeur d'un nœud est d'odre de VM Managers.
7. **Global Manager :** sert de satellite pour un nœud maitre, ensuite il a besoin du gestionnaire local pour récueillir des informations pour avoir le contrôle sur la gestion des ressources.

3.3 Les métriques utilisées

Les métriques que nous avons choisies vont nous permettre d'évaluer l'éfficacité de notre approche et un rapport avec les autres algorithmes qui va déboucher sur une comparaison d'éfficacité. Entre autre nous avons : la consommation d'énergie par les data centers, le pourcentage de violation des SLAs,compte du nombre maximum de VMs migrées et le dégré d'équilibrage des charges des VMs migrées[49]

3.3.1 Consommation d'énergie

Pour calculer la consommation d'énergie par un serveur à l'instant t pour un placement P nous pouvons nous appuyer sur les formules (3.3.1) suivantes [3, 50].

(1)

$$Ex(P,t) = k * Emax + (1-k) * Emax + Ux(P,t)$$

où Emax réprésente la consommmation d'énergie d'un serveur à 100% de son utilisation, k est un coefficient statique de l'énergie qui est égal au niveau d'énergie consommée par un processeur inactif. Sachant qu'un processeur au repos consomme environ 70% de l'energie consommée par rapport à son utilisation de 100%. Dans le cadre de ce test k est defini à 70%. Nous considerons que Ux(P,t) est l'utilisation courante du CPU par un serveur à l'instant t, ce qui est en rapport linéaire avec la consommation d'énergie. Nous pouvons calculer la consommation de l'énergie totale par tous les serveurs dans l'intervalle de temps t1 et t2 en posant l'équation (3.3.1) suivante[47].

(2)

$$Energy(P,t1,t2) = \sum_{x=1}^{J} \int_{t1}^{t2} Ex\,(P,t)$$

3.3.2 Nombre de VMs migrées

Plus le nombre des VMs migrées est élevé plus les performances se dégradent et la charge du réseau augmente.En appliquant cette équation(3.3.2) nous pouvons calculer le nombre de migrations effectuées pendant un certain intervalle de temps[51].

(3)

$$Migrations(P,t1,t2) = \sum_{x=1}^{J} \int_{t1}^{t2} Migi\,(P)$$

Avec P répresentant le placement courant des VMs, J étant le nombre d'hôtes, Migi le nombre d'hôtes J entre l'intervalle t1 et t2 pour le placement P.

3.3.3 Violation des SLAs

EN cloud, il est fondamental de mettre l'accent sur la qualité de service sous l'application de SLA. Le SLA qui peut se présenter comme une forme de contraintes parmi lesquelles le debit minimum ou le temps de réponse maximum fourni par le système deployé. Nous allons definir de metrique dans le cadre de notre simulation pour estimer le niveau de violation de SLA étant donné que les caractéristiques changent d'une application à une autre comme cela a été dit précédemment. ll est avéré que dès que les requêtes pour les performances de CPU dépassent la capacité réelle du CPU, une violation de SLA s'ensuit entre le fournisseur de service cloud avec ses clients. l'équation[52] suivante va permettre de calculer la violation du SLA

(4)

$$SLAViolation(SLAV) = \frac{1}{J}\sum_{x=1}^{J}\frac{Tsx}{Tax} * \frac{1}{N}\sum_{i=1}^{N}\frac{Cdi}{Cri}$$

Avec J nombre d'hôtes, Tsx temps total d'utilisation d'un hôte x à 100% tandis que Tax est la durée de vie(le temps durant lequel un hôte est actif) d'un hôte x. N etant le nombre de VMs, Cdi c'est l'utilisation du CPU par la VMi lors de toutes les migrations. Cette valeur estimée à 10% . Cri est le total de CPU requis par la VMi. Lorsqque l'utilisation d'un hôte atteint 100%, les performances des applications sont limitées par l'hôte.

3.3.4 Comptage du Nombre Maximum de VMs Migrées

Lorque le nombre d'occurence de VM migrée impacte sur la Violation de la VM et la dégradation des performances.Aucours d'un intervalle de temps donné nous pouvons calculer le nombre maximum d'occurence de VMs migrées[49] grâce à l'équation suivante :

(5)

$$Occurence\ migration(P, t1, t2) = \max(\int_{t1}^{t2} MigVM1\,(P), \int_{t1}^{t2} MigVM2\,(P), \int_{t1}^{t2} MigiVMn\,(P))$$

où P répresente l'emplacement courant de la VM, MigVMn(P) détermine le nombre de migrations de VM n entre t1 et t2 pour le placement P.

3.3.5 degré d'équilibrage des charges du nombre de VMs migrées

Ceci contribue à la réduction du problème de la selection biaisée des VMs. De facto contribue à réduire le pourcentage de violation des SLAs. Pour calculer le degré d'équilibrage des charges on utilise la variance du nombre de VMs migrées[49] comme suit :

(6)

$$Degré\ d'équilibrage\ des\ charges = \sqrt{\frac{1}{N} * \sum_{i=1}^{N} (mi - \overline{m})^2}$$

(7)

$$\overline{m} = \frac{1}{N} * \sum_{i=1}^{N} mi$$

Avec mi répresentant le nombre de fois de migration de VMi, N nombre de VMs et enfin $\overline{m}$ *la moyenne du nombre de VMs migrées.*

3.3.6 Coût de la Migration à chaud des VMs

La migration à chaud dégrade sensiblement des performances des applications s'exécutant sur les VMs. Voorlys et al[53]ont realisé une etude afin d'évaluer la valeur de l'impact de cette dégradation. Leur étude a prouvé que la dégradation de performances des performances et le temps d'arrêt downtime sont relatifs au comportement de l'application lancée sur la VM. Autrement dit cela depend du nombre des pages memoires mises à jour par l'application lors de son exécution. Les applications Web ont un pourcentage moyen de 10% de degradation de performance et de temps d'arrêt par rappport à l'utilisation de CPU. A cette allure aucune migration n'est exempte de violation de SLA. C'est pourquoi il serait primordial de minimiser le nombre de migrations de VM. Sachant que le temps de migration des VMs en Live depend

de la taille de mémoire utilisée par la VM et la bande passante. Selon la formule suivante [3]la dégration des performances peut être déduite à partir de :

(8)

$$Tmj = \frac{Mj}{Bj}$$

(9)

$$Udj = .1 * \int_{t0}^{t0+tmj} Uj(t)\, dt$$

Avec Udj comme dégradation totale de performance par VMj, t0 temps de debut de la migrationn, Tmj temps necessaire pour parachever la migration, Uj(t) l'utilisation de CPU par la VMj , la VMj une quantité mémoire Mj enfin Bj est la bande passante du réseau disponible.

3.4 Stratégies de sélection de VMs proposée

Les enjeux du cloud computing sont axés sur la consommation d'énergie dans les data centers. Pour surmonter les problèmes liés à cette surconsommation d'énergie et en vue de promouvoir la croissance de cloud à travers le monde, il convient de trouver les voies et moyens permettant d'optimiser la consommation d'énergie pour satisfaire un certain seuil d'exigences de la qualité de service(Qos) en conformité avec le contrat SLA(Service Level Agreement) qui regit les services des prestateurs vis-à-vis des clients[54].

Nous allons précisement dans cette partie nous focaliser sur deux politiques afin de réduire le nombre de migration de VMs, la consommation d'énergie etc. Cela est rendu possible grâce à la virtualisation et la migration live En général, le processus de migration Live est gourmand en ressources surtout en CPU et ressources réseaux.

Il a l'inconvénient de garder les VMs indisponibles pendant la migration entre l'hôte source et destination.Notre stratégie vise à reduire la violation de SLA de manière considérable pour toutes les VMs sans empiéter sur les pourcentages des autres métriques. La solution que nous avons proposée consiste en :

- Minimum VM Migrated Count Hybrid(MiMcHybrid) : La politique de l'algorithme consiste à sélectionner la VM à partir d'un hôte le moins surchargé vers un hôte moins surchargé en tenant compte du nombre minimun d'occurence de VMs migrées.
- Migration Time Minimum VM Migrated Count Hybrid(MmtMiMcHybrid) :L'algorithme va sélectionner les VMs selon la capacité minimun de leurs RAM pour réduire le temps[55] de migration. Cela permet également d'ordonner les VMs pour qu'à la fin la sélection se fasse sur le nombre minimum de fois qu'il y a eu de migrations de VMs.

Le modèle à presenter est le suivant : Un data center constitué d'un groupe de machines physiques (PMs). Tout hôte physique peut contenir m VMs distinctes(hétérogènes). Ces entités distinctes disposent de nombreuses resssources dont entre autre : l'utilisation de CPU, la capacité de la RAM,MIPS,les seuils inférieurs et supérieurs d'utilisation de ressources. L'objectif visé est de réduire la violation SLA en minimisant le nombre de VMs à migrer, pour cela on selectionne un nombre réduit de machines virtuelles. Cette stratégie est adaptée pour diminuer au maximum la consommation d'énergie. Ceci revient à booster le taux d'utilisation des ressources physiques. Nos trois approches se decrivent ici :

- Minimum VM Migrated Count Hybrid(MiMcHybrid).
- Migration Time Minimum VM Migrated Count Hybrid(MmtMiMcHybrid).

Ces approaches s'appliquent sur la phase de sélection. Avec le diagramme d'activité nous expliquérons la démarche de nos propositions. **Phase de sélection :** il s'agit de savoir quelle VM migrer selon la stratégie que nous avons proposée. **Phase d'allocation :** c'est la phase de placement de la machine virtuelle sur un hôte physique en utilisant un algorithme de placement trouvé dans la litterature. Le diagramme d'activité va servir à décrire le fonctionnement de notre politique de migration proposée.

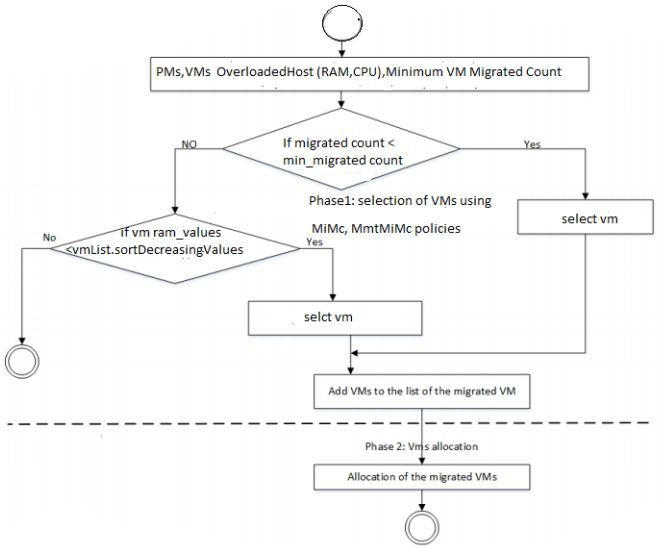

Figure 3.2 – Diagramme d'activité de la stratégie proposée

3.5 Phase de Sélection

Avec cette phase nous pouvons déterminer quelle VM migrer et à quelle moment la migrer dans l'objectif de minimiser à une grande échelle l'énergie consommée. C'est pourquoi cette phase se caractérise par l'identification des VMs surchargées du fait de l'utilisation des ressources physiques.A cela s'ajoute également la détermination de la quantité de la RAM utilisée.

3.5.1 Algorithme de MiMcHybrid

Dans le pseudocode qui va suivre, au cas ou l'utilisation du CPU de la PM est au dessus de la valeur du CPU fixée , l'algorithme tentera de migrer pour garder les hôtes dans la norme des valeurs du CPU. Si le comptage minimum de migration de VMs des hôtes surchargés s'avère en dessous de la valeur du comptage initial, cela permettra de choisir les meilleures VMs. IL va prendre en compte les VMS dont l'utiilisation du CPU excède celle des autres VMS du même hôte. Ceci contribue à minimiser le nombre de migration. La fin du pseudocode traite de la migration de toutes les VMS lorsque le CPU n'est pas utilisé par la machine physique dans la valeur minimale fixée. Ensuite la PM est mise à l'arrêt pour économiser de l'énergie. Les valeurs seuils ont été fixées dans ces approches.

3.5.2 Algorithme et diagramme d'activité de MmtMiMcHybrid

Pour ce cas de figure, le pseudocode va sélectionner les VMs selon la valeur de la RAM, cela réduit le temps donc de migration. L'algorithme va ordonner les VMS dans l'ordre croissant.Puis il va prendre en considération les VMs avec le comptage minimum du nombre de migrations de VMs. Dans la séconde partie, l'algorithme vérifie si l'utilisation du CPU ou de la RAM est inférieure à la norme définie.Si la condition est vérifiée, une migration de toutes les VMS est programmée de la PM vers d'autres hôtes physiques. Une mise hors tension prévaut si les PMs ne possèdent pas de VMs, ceci dans un but de dimunier la consommation d'énergie de ces serveurs.

3.6 Phase d'Allocation

3.6.1 Principe de l'algorithme Fest Fit DEcreasing (BFD)

En informatique comme à l'instar des autres secteur ; on fait face au problème d'optimisation, tel que par exemple le rangement des fichiers dans un emplacement informatique en tenant en compte dela capacité de stockage. A titre d'exemple nous prenons un cas des sacs tous identiques les uns des autres « m » et un ensemble d'objets « n ». Pour tout objet i, il existe un poids « p max ». La question est de connaitre combien de nombre minimum de sacs a-t-on besoin pour remplir nos objets ? C'est en effet tout le cœur du problème de Bin packing auquel nous nous confrontons. Des techniques permettent de parvenir à des solutions plutôt satisfaisantes dans une large mesure à l'exemple de Next Fit Heuristic (BP-NF) ;First Fit DEcreasing (FFD) ou Best Fit DEcreasing(BFD) etc[56]. Ces algorithmnes sont à adapter selon le besoin de sa propre solution car ils sont pas optimaux. Ils aident à obtenir un compromis entre un bon ratio de la qualité de la solution avec le temps de calcul. En ce qui nous concerne nous avons opté pour la version EBFD en le modifiant à notre contexte.

3.6.2 Algorithme Energy-Aware Best Fit Decrreasing (EBFDHybrid)

Afin de permettre le placement de VM, dans une PM, il est nécessaire que l'algorithme vériifie que la machine est prête à recevoir la VM, autrement dit a assez de resssources. Ensuite vérifier que les utilisations du CPU sont en dessous des valeurs fixées. ENfin vérifier que la consommation d'énergie de la machine physique par rapport aux autres PMs est la plus petite. Cet algorithme permet de placer les VMs sélectionnées par les algorithmes de sélection,Il place les VMs dans

les hôtes physiques pour trouver le meilleur emplacement pour ces VMs.

Algorithme 1 : Minimum VM Migrated Count Hybrid(MiMcHybrid)

Entrée: OverloadedHost, VMList,hostList

Sortie: VMs_toomigrate

```
1:  Min_migrated_count ← Max
2:  Selected_VM ← none
3:  VMList ← OverloadedHost.getVMList() ;
4:  Pour chaque h dans hostList faire
5:  hostUtilCPU ← h.utilCPU() ;
6:  tant que hostUtilCPU > h.plafondCPU() faire
7:      bestVMUtilCPU ← h. BestVMUtilCPU() ;
8:      bestVM ← Null ;
9:  pour chaque VM dans VMList faire
10:     si VM.utilCPU = bestVMUtilCPU alors
11:         Migrated_count ← VM.getMigrated_count
12:         si migrated_count < Min_migrated_count alors
13:             Min_migrated_count ← Migrated_count
14:             bestVM ← VM ;
15:             selectedVM ← bestVM ;
16:             break ;
17:             hostUtilCPU ← hostUtilCPU() - bestVM.UtilCPU() ;
18:             VMs_tomigrate.add(bestVM) ;
19:             VMList.remove(VM) ;
20:             si hostUtilCPU < h.plancherCPU() alors
21:                 VMs_tomigrate.add(h.getVMList()) ;
22:                 VMList.remove(h.getVMList()) ;
23:                 h.Offstate() ;
24:
    retourne VM_tomigrate ;
```

Algorithme 2 : Minimum Migration Time Minimum VM Migrated Count Hybrid(MmtMiMcHybrid)

Entrée: OverloadedHost, Vms_ram_Val,VMList

Sortie: VMs_toomigrate

Min_migrated_count ← *Max*

bestVmUtil ← *Max*

Selected_VM ← *none*

VMList ← *OverloadedHost.getVMList() ;*

VMList.sortDecreasing_vms_ram_Val() ;

pour chaque *h* **dans***hostList* **faire**

si *hostUtilRAM* > *h.plafondRAM* **alors**

bestVMUtilRAM ← *h. bestVMUtilRAM() ;*

bestVM ← *Null ;*

pour (*i* = 1, *i* <4, *i*++) **faire**

vmList2[i] ← *VMList[i]*

pour chaque *vm* **dans***vmList2* **faire**

Migrated_count ← *VM.getMigrated_count ;*

si *migrated_count* < *Min_migrated_count* **alors**

Min_migrated_count ← *migrated_count ;*

si *VM.utilRAM* = *bestvmUtilRAM* **alors**

bestVM ← *vm ;*

selectedVM ← *bestVM ;*

break ;

hostUtilCPU ← *hostUtilCPU() - bestVM.UtilCPU() ;*

hostUtilRAM ← *hostUtilRAM() - bestVM.UtilRAM() ;*

VMs_tomigrate.add(h.getVMList()) ;

VMList.remove(h.getVMList()) ;

si *hostUtilCPU* < *h.plancherCPU()orhostUtilRAM* < *h.plancherRAM()* **alors**

VMs_tomigrate.add(h.getVMList()) ;

VMList.remove(h.getVMList()) ;

h.Offstate() ;

retourne *VM_tomigrate ;*

Algorithme 3 : Energy Aware Modified Best Fit Decreasing

h

Entrée: hostList,vmList,CPU_Thrl, CPU_Thrm, CPU_Thrh, upper_RAM_utilThr

Sortie: allocatedHost

vmList.sortDecreasingByCpuUtilization( ) ;

pour chaque *vm* **dans***vmList* **faire**

minPow ← Max ;

allocatedHost ← Null ;

pour chaque *host* **dans***hostList* **faire**

si *(isHostHeavilyLoaded(host))* **alors**

HeavilyLoadedHosts.add(host) ;

Fin si ;

Fin si ;

vmMig ← getVmsFromHeavilyLoadedHosts(HeavilyLoadedHosts) ;

HeavilyLoadedHosts.clear( ) ;

vmMig.clear() ;

pour chaque *host* **dans***hostList* **faire**

si *(host est prêt pour accueillir vm)* **alors**

ratio ← Get_CPU_resUtil_RatioApresAlloc(host,vm) ;

si *((ratio < CPU_Thrl))|| ratio > CPU_Thrm) et (RAM_resUtil_ RatioApresAlloc < up*

alors

energyConsom ← getPowerApresVM(host, vm) ;

si *(energyConm<minPow)* **alors**

allocatedHost ← host ;

minPow ← energyConsom ;

Fin si ;

Fin pour ;

Fin pour ;

retourne *allocatedHost(vm dans PM) ;*

Conclusion

Nous sommes à la conclusion de ce chapitre, grâce aux approches proposées,il s'agira de gérer la consommation d'énergie des data centers et contribuer à réduire l'émission du CO2 dans les perspectives.La virtualisation et la migration à chaud viennent à l'appui afin d'atteindre cet objectif. Les approches sont basées sur la selection et l'allocation respectivement pour MiMcHybrid,MmtMiMcHybrid et EBFDHybrid. La première approche s'appuit sur l'utilisation du CPU, le comptage minimum de VMs migrées, la séconde approche vrifie les deux ressources physiques à savoir la RAM et le CPU mais également avec la prise en conssideration du comptage minimum de VMs migrées pour appliquer la sélection des VMs. Nous allons consacrer le prochain chapitre à l'implementation et interprétations des résultats des simulations que nous allons effectuer sous le simulateur CloudSim.

Chapitre 4

Implémentation et Interprétations

4.1 Introduction

Les propositions que nous avions donné permettrons idéalement de réduire le nombre de migration des Vms, les violations du SLA, la consommation denergie. Pour les autres métriques citées dans le chapitre prcedent, nous n'allons pas les performances de nos algorithmes sur ces métriques pour des raisons d'encombrement.Nous avons utilisé le simulateur CloudSim pour tester, discuter les résultats de notre solution et commparer avec d'autres politiques qui existent déjà dans la litterature. Dans ce dernier chapitre nous présentons java, comme lengage utilisé, l'IDE Netbeans comme environnement de dévéloppement choisi dans le cadre de notre travail. Pour joindre l'utile à l'agréable nous avons opté pour le simulateur Cloudsim, dont nous allons présenter l'architecture. Nous allons également présenter ses classes fondamentales et celles que nous avons crées.Enfin nous allons expliquer la configuration ncessaire au lancement de la simulation et interpreter les resultats obtenus.

4.2 Environnement de développement

Nous avons effectué l'ensemble des tests sur une machine avec les caractéristiques suivantes :

- Une machine avec un processeur Intel(R) Core(TM) i5-4200M CPU, une vitesse de @ 2.50GHz, 2 coeurs.Une mémoire Ram de 8 GB. Cloudsim tourne sur windows 10 64bits.
- Le simulateur Cloudsim version 3.03 développé en Java.
- Langage de programmation Java.
- L'IDE netbeans.

4.2.1 Langage de Programmation Java

Java a été crée pour être à la fois langage de programmation orientée objet et environnment d'exécution portable. Des employées de Sun Microsystems sont à l'initiative du langage. en l'occurence James Gosling et Patrick Naughton avec l'appui de Bill Joy en 1982 mais dont dont la présentation au grand public a été un 23 mais 1995 au SunWorld(Ivan 2010). le langage java a une spécificité de portabilité des programmes développés sous ce langage sur toutes les plateformes telles que : Microsoft windows, Mac OS, Linux. C'est la plateforme qui assure la portabilité des applications développées en java(ivan devosa 2010).

La plateforme garantissant aux applications java d'être portables sur tous systèmes d'exploitations est la JRE(java runtime environnement) qui existent pour tous ces systèmes d'exploitations. Cette portabilité est dûe à la JVM(java virtual machine), c'est le programme qui interprète le code java pour le convrtir en code natif. JRE admet aussi une bibliothèque standard capable de sont développés tous les programmes java.Cette portabilité qui a fait l'angouement de java dans les architectures client/serveur en facilitant la migration entre serveurs , chose compliquée pour les grands systèmes(IPETI 2005). Aujourd'hui java est omniprésent et devenu incontournable dans le monde de la programmation,on peut noter les caractéristiques qui ont hissé au succès java sont entre autres(Ivan 2010).

- L'independance de toute plate-forme : le code peut s'excuter peu importe la machine. Encore mieux , un programme java peut s'exécuter sur toute plateforme munie d'une JVM(java virtual machine).
- La portabilité de java permet d'avoir une simulation distribuée sans necessiter une recompilation de code pour les differents systèmes.
- Le code est organisé en plusieurs classes dont chacune traite une partie differente.
- Java assure la gestion de la mémoire
- Java aussi multitâche c'est-à-dire qu'il gère les threds.

4.2.2 Environnement de dveloppement Netbeans IDE

Netbeans est un environnement de de développement intégré moderne lancé par Sun et placé en Open Source dans les années 2000.C'est un IDE multi-langages, supportant java,C,Python, C++, javascript,XML,Ruby,PHP etc. Il est composé d'un éditeur en couleur,projets multi-langages, refactoring, éditeur d'interface graphique,et de page web comme tout environnement

de développement moderne. Il est disponible sur le site officiel url : https://netbeans.apache.org\T1\guilsinglrightdownload. Il existe des versions pour tous les systèmes d'exploitation et pour toute autre version de OS indpendante(en intégrant une machine virtuelle java). Sa compatibilité esst avérée avec les nouvelles technologies.

4.2.3 Le Simulateur CloudSim

CloudSim est un simulateur orienté recherche et développement dans le domaine de cloud computing. Le choix orté sur ce simulateur s'explique du fait que contrairement à d'autres outils de simulation, il supporte plusieurs fonctionalités[57].

- supporte la modélisation et la simulation d'environnements à grande échelle du cloud computing, y compris les data centers.
- modlisation et simulation des serveurs virtuels sous la politique d'approvisionnement des ressources de la machine physique en machine virtuelle.
- modélisation et simulation des ressources de calcul permettant la gestion d'energie.
- modélisation et simulation des topologies de réseau et de applications de transmission de messages.
- modélisation et simulation des clouds différés.
- supportte l'insertion dynamique des élements de simulation, l'arrêt et la reprise de la simulation.
- supporte les politiques définies par l'utilisateur pour l'allocation des VMs dans des hôtes et l'allocation des ressources de l'hôte dans des Vms.
- CloudSim prend en charge la simulation des approches permettant la rduction de la consommation d'energie des data centers. Ainsi que la simulation des applications de service avec une charge de travail dynamique a té incorporée.
- une plate-forme autonome capable de modéliser le cloud computing, des Brokers, des data centers,intervient dans l'ordonnancement et dans les politiques d'allocations des ressources.
- un support pour la simulation des connexions réseau entre les élements du système de simulation.
- la disponibilité d'un outil de virtualisation permettant de gérer et créer facilement des services. Les services virtualisés également sont hebergs dans les noeuds des data centers.

- prise en charge de la répartition des coeurs de trraitements aux services virtualisés. Ainsi qu'une commutation entre l'allocation en espace partagé et en temps partagé.
- simulation des spéfications, de la création et de la destruction des machines virtuelles.

4.2.4 Architecture de CloudSim

Le moteur de simulation utilisé dans CloudSim est SimJava.Il possède plusieurs fonctionnalités basiques dont les files d'attente,le traitement des évenements,la création des entités du cloud(Service,Hôtes,Data center, Broker,Vms),la communication entre les composants, la gestion d'hôrloge de simulation. SimJava applique la simulation à évenements discrets.La figure ci-après montre une architecture multicouche de CloudSim ainsi que ses composnts(Figure 4.1). Néanmoins la couche du moteur SimJava n'existe plus dans la version actuelles dans un souci de prise en charge des opérations impossibles à réaliser avec SimJava[57].

La couche CloudSim supporte la modélisation et la simulation des data centers. Au-delà, elle permet la prise en charge de la simulation des interfaces de gestion des machines virtuelles, de stockage, de mémoire et la bande passante. Cette couche ne s'arrête pas qu'à ça, elle assure également l'affectation des Vms vers les hôtes, la gestion de l'excution des applications ainsi que la surveillance de l'état du système dynamique.

Pour founir les entités nécessaires aux hôtes telles que (nombre de machines, leur spécification...), les applications(nombre de tâches), Vms,nombre d'utilisateurs,types d'applications avec les politiques d'ordonnancement du Broker, nous avons besoin du code utilisateurappelé (user code) et qui est situé au niveau de la couche supérieur de CloudSim. Pour un développeur d'applications Cloud, générer des activités est abyssal.Parmi les inombrables activités qui peuvent en sortir on peut citer [58] : des scnarios de disponibilité des modèles clouds,effectuer des tests robustes basés sur les configurations prises en charge par CloudSim qui sont personnalisables, il y a aussi des techniques d'approvisionnement des applications, encore une fois personnalisables à travers le Cloud.

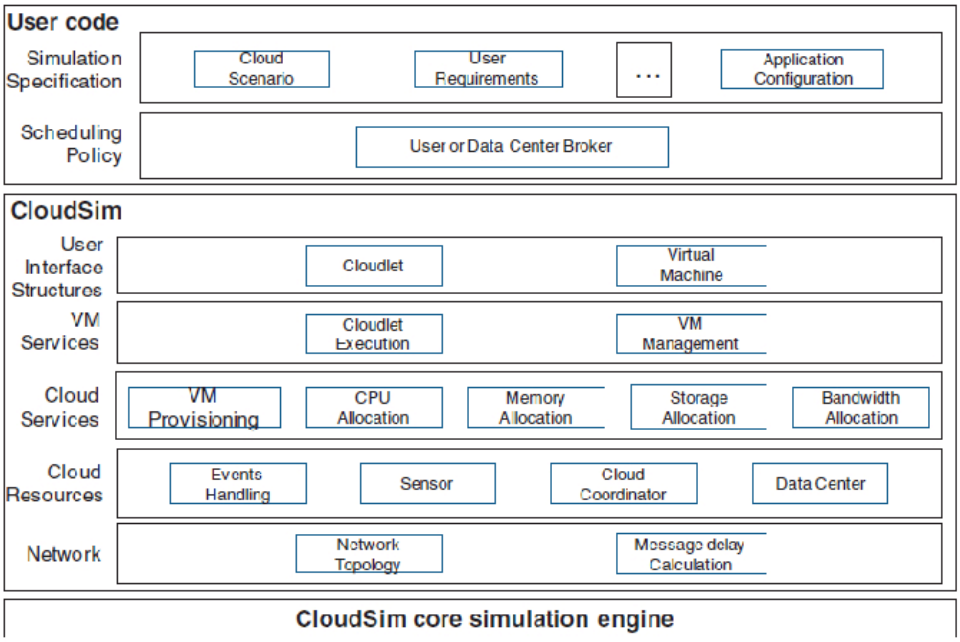

Figure 4.1 – Architecture de CloudSim

[59]

4.2.5 Les classes de CloudSim

Nous distinguons principalement deux catégories de classes pour le simulateur CloudSim : Les Classes qui modélisent les entités comme le data center, le Broker,etc. De l'autre côté nous avons les classes qui modélisent les politiques d'allocation. Nous présentons les classes de base de CloudSim et les classes du package Power particulièrement avec lesquelles nous avons travaillé. Nous avons integré d'autres classes qui s'inscrivent dans le cadre de notre solution pour la simulation. La figure (Figure 4.2) explique les classes en détails.

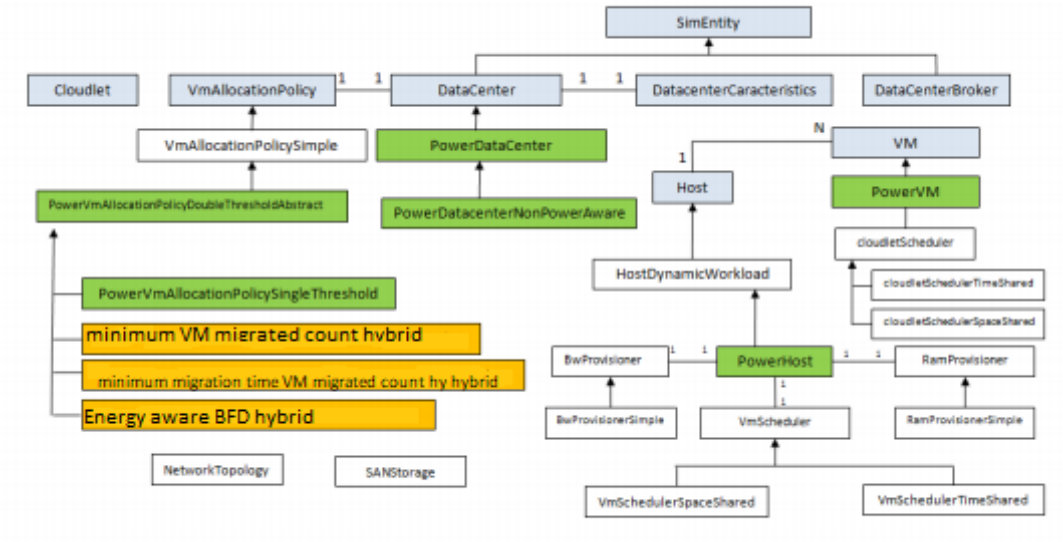

Figure 4.2 – Diagramme de Classe de la Conception de CloudSim [59]

(A) **Cloudlet :** pour un cloud standard, il existe des services dédiés aux appplications du cloud parmi lesquelles il y a(la livraison, réseaux sociaux, workflow d'affaires). ces services sont modélisés par la classe cloudlet. Le simulateur interprète les besoins informatiques d'une application pour definir sa complexité. pour un service d'application donné, correspond un cycle de vie pour lequel la taille d'instruction et du flux de transfert de données est fixée Il est loisible d'etendre la classe cloudlet pour un besoin de modélisation de performances et bien d'autres paramètres de composition pour les application, à l'exemple des transactions dans les applications orientées base de données[58].

(B) **Datacenter :** Il est évident que les grands fournisseurs du cloud tels ques (Amazon,Azure,App Engine) fournissent des services au niveau des infrastruces matérielles(ou infrastructure de base), cependant la classe Datacenter modélise les services au niveau de l'architecutre matériellle. Cette classe à le merite de brouiller un certain nombre d'hôtes jugés homogènes ou héntérogène selon la configuration de base (mémoire, noyeau, capacité de stockage. Nous y avvons également la classe PowerDatacenter.java qui se touve le package le package Power hérite de la classe Dataccenter.java. Pour la simulation, elle permet de simuler des data centers en gardant un oeil sur l'energie consommée.

(C) **DatacenterCharacteristics :** on fait appel à cette classe pour obtenir les données liées à la configuration des ressources des data centers[58]. Ces informations peuvent concer-

ner le système d'exploitation , la bande passante, la liste des hôtes, le coût par seconde d'utilisation des ressources,le stockage, le coût pour la mémoire, VMM de ces ressources.

(D) **DatacenterBroker :** C'est une classe qui sert d'intermédiaire entre les prestataires du cloud et les utilisateurs en tenant compte des conditions de QoS des utilisateurs. Il est responsable aussi du delpoiement des services differents clouds. Les utilisateurs adressent leurs requêtes aux au Broker, qui grâce à CIS(Cloud Information Services) peut contacter les prestataires du service idoine pour le compte du client afin de lui fournir les ressources dont il a besoin, et qui reponde au QoS[58].

(E) **Host :** Le serveur proprement dit est modelisé par cette classe avec ses composantes de ressources physiques. Il peut s'agir du serveur de stockage ou de calcul. La plupart des informations relatives à la bande passante, aux politiques d'allocation de mémoire, à la politique d'allocation pour le partage de la puissance de traitement, le type de coeurs de traitement(pour representer une machine multicore), la quantité de mémoire et de stockage sont encapsulées par la classe Host. Pour traiter l'energie la classe PowerHost est mise en avant pour la simulation.

(F) **SimEntity :** C'est une classe qui intervient pour envoyer des messages aux entités, pour recevoir des messages et dans le même temps gère les évenements. Cette classe est à la base abstraite et joue le rôle de l'entité de simulation. De ce fait, toutes les autres entités doivent etendre cette classe et redefinir ses méthodes abstraites : startEntity(),processEvent(),shutdownEntity(). Ce sont ces trois principales méthodes qui sont redefinies. Avec la redefinition, ces méthodes definissent respectivement l'action d'initialisation de l'entité, le traitement des évenements et la destruction de l'entité[58].

(G) **VM :** EN fait, c'est la machine physique qui gère et heberge l'instance de machine virtuelle(VM) que cette classe représente. Toutes les charactéristiques des VMs comme :mémoire accessible,processeur,capacité de stockage, les politiques d'approvisionnement sont stockés dans les composants. Et chaque composant peut librement accéder à ces composants qui stockent ces données[58]. Etant donné la question de la réduction d'energie dans les data centers est vivrante pour la plupart des travaux, la classe PowerVm.java a été développé. Elle hérite donc de la classe VM.java qui sauvegarde l'historique de l'utilisation du CPU. Les politiques de slection et d'allocation se servent de cette donnée pour la réutiliser dans le coprps de leurs méthodes.

(H) **VmAllocationPolicy :** Les machines hôtes,dans le data centre, ont pour vocation d'accueillir les machines virtuelles, cette classe simule cet approvisionnement des hôtes en

machines virtuelles.En se qui concerne nos propositions, nous avons travaillé avec les méthodes de sélection et d'allocation pour les trois trois approches que nous avions proposées et qui sont implémentées dnas les classes suivantes : PowerVmSelectionPolicyMinimumVmMigratedCountHybrid.java,PowerVmSelectionPolicyMinimum MigrationTimeMinimumVmMigratedCountHybrid.java, EnergyAwareBestFitDecreasingVmAllocationPolicyHybrid.java pour la politique de sélection et de d'allocation respectivement MiMcHybrid,MmtMiMcHybrid,EBFDHybrid. Les deux premières classes héritent de la classe PowerVmSelectionPolicy .La classe d'allocation hérite de la classe PowrVmAllocationPolicyMigrationdAbstract.java héritant à son tour de PowerVmAllocationPolicy.java.

4.3 Expérimentations et résultats

4.3.1 Configuration

A. **Configuration des Data Centers(DC) :** Pour configurer un data center sous CloudSim, il faut remplir les informations suivantes : le nom du Data Center, son architecture, le gestionnaire des VMs(VMM), les différents coûts(mémoire, traitement,stockage,bande passante), l'intervalle d'ordonnacement ainsi que le système d'exploitation utilisé par le data center.

B. **Configuration des Machines Physiques(PM) :** La création d'une PM nécessite la configuration suivante : le nombre d'hôtes, nombre de processeurs,la capacité de la Ram, le nombre des MIPS,la bande passante et la capacité de stockage. PPour notre part les hôtes configurés sont hétérogènes avec differentes valeurs de MIPS et de la RAM.

C. **Configuration des Machines Virtuelles(VMs) :** Il faudra renseigner les champs suivants pour pouvoir configurer les VMs : le nombre de VMs, le nombre de processeurs, la RAM pour chaque VM,la bande passante et le stockage, le nombre de MIPS.En résumé pour configurer les cloudlets, il faudra préciser la taille du cloudlet, le nombre de processeurs, le nombre de cloudlet. Comme il a été dit précedemment pour notre simulation, les machines virtuelles sont hétérogènes et les cloudlets égalent les VMs en nombre.

D. **Lancement de la simulation :** pour lancer la simulation,après la configuration, une classe contient une méthode main avec les paramètres de simulation nécéssaires à la simulation. Elle contient le type de workload, la politique de selection de VM choisie, la politique d'allocation.

Dans la section suivante, nous allons montrer les résultats expérientaux que nous avions obtenus suite aux différentes simulations effectuées.

4.4 Résultats Expérimentaux

Pour tester nos propositions, nous avons selectionné trois métriques à savoir :la consommation d'energie du Data Center,le nombre de migrations de VMs et la violation des SLAs avec la contrainte d'energie. En revanche les autres métrques n'étant pas pris en compte dans le cadre de l'évaluation. Nous avons jugé utile de présenter la simulation de deux approches déjà presentes dansCloudSim version afin de mésurer l'efficacité de nos approches par rapport aux autres.La première politique est Non Power Aware Policy(NPA). La particularité de cette approche est qu'elle se deroule sans aucune condition d'optimisation et fait fonctionner toutes les machines physiques en utilisant la totalité des ressources CPU. Elle est de loin l'approche ayant une utilisation maximale de puissance[60]. La politique suivante est Single Threshold Policy. La spécificité de l'approche est de fixer un seuil d'utilisation consacré aux hôtes avec une valeur suprieur afin de placer les machines virtuelles dans ces hôtes, dans le même temps maintenir l'utilisation totale de CPU à une valeur inférieure à celle du seuil supéreieur fixé.L'idée mène à la preservation des ressources libres ce qui permet de prévenir les violations SLAs émanant de la consolidation de VMs dans le cas ou les machines virtuelles sollicitent énormement de ressources[3]. L'intérêt d'avoir choisi cette approche ST[61] se justie du fait que les chercheurs qui ont procédé à plusieurs expériences avec différents valeurs du seuil d'utilisation pour nous fournir les résultats de l'évaluation. Ces resultats sont présentés dans la figure(Figure 4.3). D'après les expériences, les résultats prouvent une dimunition de la consommation d'energie significativement en comparaison avec les politiques NPA,DVFS de 77% et 53%, respectivement, et avec 5,4% dess violations de SLA. Plus loin, les resultats montrent une augmentation du pourcentage de violation de SLAs pendant que le seuil d'utilisation reduit la consommation d'energie lorsque sa valeur est élevée.

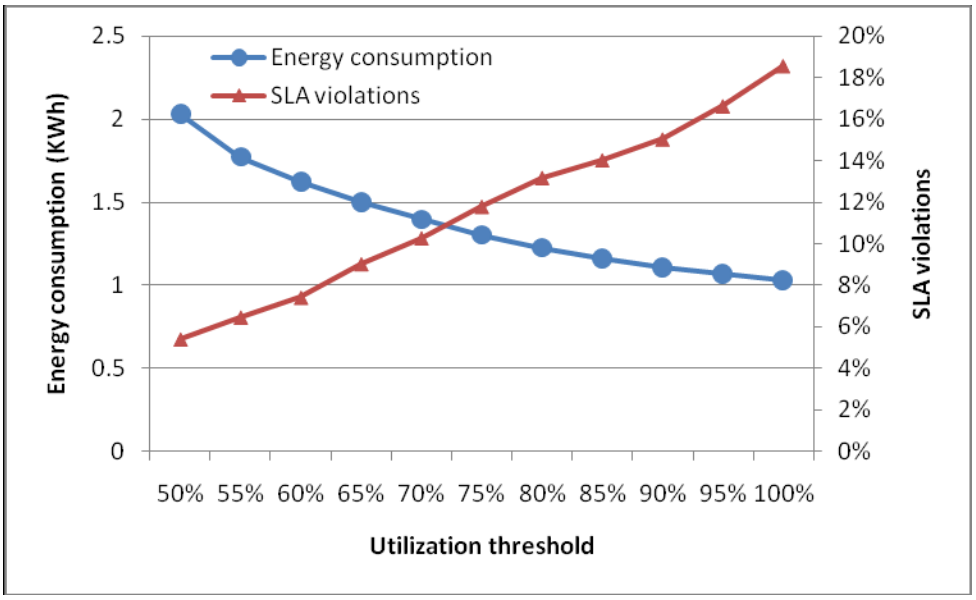

Figure 4.3 – Consommation d'energie et violation de SLA poour ST [61]

4.4.1 Paramètre de Simulation

Dans le cadre général de notre simulation, nous avons configuré un data center conténant des hôtes hétérogènes sous CloudSim version 3.0.3.Le data center héberge 800 noeuds physiques hétérogènes,dont la moitié est constituée de HP ProLiant ML110 G4 (Intel [Xeon 3040 1860 MHz, 2 cores], 4GB) et l'autre moitié est constituée de HP ProLiant ML110 G5 (Intel [Xeon 3075 2660 MHz, 2 cores], 4GB).Il y a 2500 VMs de quatre types(High-CPU Medium Instance(2500 MIPS, 0.85 GB) ;Extra Large Instance (2000 MIPS, , 3.75 GB) ;Small Instance (1000 , 1.7 GB) et Micro Instance (500 MIPS, 0.633 GB)) Les caractéristiques de ces VMs correspondent d'égal à égal au type d'instance de VMs de Amazon EC2. La simulation s'est faite avec les approches proposées de selection de VM sur le workload Random.Dans cette configuration, chaque hôte possède 2 processeurs avec une vitesse qui varie en MIPS(1860,2660,3000), (4GB,6GB,8GB) de mémoire Ram, 1TB de stockage,1GB/s de bande passante. Le broker dans son rôle fait varier un nombre de machines virtuelles hétérogènes.Toute VM possède un processeur avec une vitesse variant en MIPS(2500,2000,1000,500), (870,1740,1740,613) de mémoire Ram,100 Mb/s de bande passante,1GB de stockage. Parmi les paramètres choisis du seuil d'utilisation du CPU de la politique PowerVmAllocationPolicyMigrationThreshold, nous avons pris 0.9 comme seuil. Pour notre approche c'est-à-dire EBFDHybrid, nous avons fixé le seuil à 1.2. Nous avons poursuivi les évaluations avec les autres politiques d'allocation comme PowerVmAllocationPolicyMi-

grationLocalRegressionRobust pour tester nos deux politiques de selections à savoir MiMcHybrid,MmtHybrid. les résultats sont représentés grâce aux graphes et tableaux plus en détails dans la section suivante.

4.4.2 Première Expérience : Un nombre élevé de PMs et VMs

Dans cette première expérience, nous avons configuré le data center avec 100 hôtes physiques et 300 accompagnés de 300 machines virtuelles de différents types. Le Broker utilise un pas de 50 pour faire varier le nombre de VMs de 50 à 300.

4.4.2.1 Consommation d'energie

Après avoir méné la simulation et recueilli lesrésultats des tests, la consommation d'energie s'avère moindre avec nos approches MiMcHybrid,MmtMiMcHybrid par rapport aux approches existentes dans la litterature, en l'occurence MC(Maximum Correleation),MU(minimum utilisation). D'après l'analyse des resultats obtenus comme peut l'indiquer la (Figure 4.4), nous constatons que les politiques que nous avons implémenté, ont un ratio de consommation d'energie relativement faible par rapport aux autres, ce qui nous ramène à dire que nos approches convergent vers l'objectif de l'informatique verte (Green Computing) c'es-à-dire produisent moins de chaleur et economisent de l'energie.

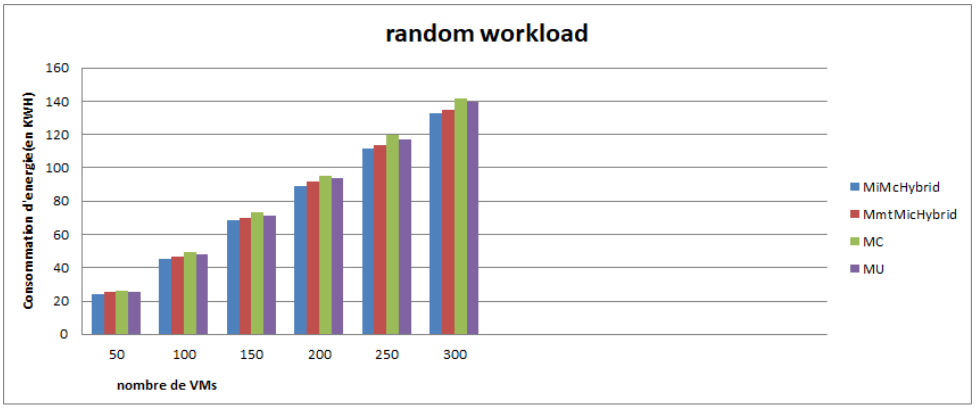

Figure 4.4 – Consommation d'energie

En effet, MiMcHybrid montre une consommation d'energie plus faible que MmtMiMcHybrid. Cette valeur a tendance a restée constante au fil des phases de simulation, nous remarquons que la consommation d'energie decuple avec l'augmentation du nombre de machines virtuelles dans les

hôtes physiques,l'allure est ascendante.Nous avons réalisé cette expérience avec notre politique d'allocation de machines virtuelle basée sur la version hybridée de l'algorithme Energy Aware Best Fit Decreasingque nous avons nommée EBFDHybrid, qui a des traits communs avec la politique PowerVMAllocationPolicyMigrationStaticThreshold. MiMicHybrid a une consommation d'energie inferieure à MmtMiMcHybrid en fonction de l'augmentation du nombre de VMs, les performances de MiMc sont depassent legerement celles de MC,MU avec les mêmes données. Nous y avons obtenu ces resultats avec les charges de travail random(random workload), les tests dans le package power.random de cloudsim.

Nombre de VMs	50	100	150	200	250	300
MiMcHybrid	24,43	45,58	68,28	89,19	111,45	132,71
MmtMiMcHybrid	25,30	46,77	69,61	91,68	113,62	135,12
MC	26,37	49,10	73,08	95,38	119,66	141,54
MU	25,39	48,23	71,59	93,89	117,18	139,64

Table 4.1 – consommation d'energie

D'après les enregistrements de ce tableau qui contient les valeurs des quantités d'energie consommée par les differents approches, nous observons l'approche de selection de VM MiMcHybrid reduit la consommation d'energie rapport à toutes les autres approches avec un nombre de VMs à 50. L'energie consommée est de l'ordre de 24,43 KW, tandis que les autres vont au minimum 25 KWh. Plus on augmente le nombre de VMs plus la consommation augmente legerement pour toutes les autres approches car il y a une surutilisation des ressources des PMs. L'ecart sur la réduction de la consommation d'energie entre MiMcHybrid et MU s'observe avec beaucoup d'acuité lorsqu'on atteint le seuil de 300 VMs, notre approche reduit avec environ 7 de moins par rapport à MU et 9 par rapport à MC.

4.4.2.2 Nombre de migrations de VMs

En ce qui concerne cette métrique, au vu des résultats des simulations mis en évidence, il en ressort que nos approches ont largement pris le dessus dans la réduction du nombre de migration de VMs. Nos approches ont significativement dimunié le nombre de migration comme il est visible dans la (Figure 4.5) et le tableau(Table 4.2). Avec l'augmentation du nombre de VMs dans les hôtes, le nombre de migration de VMs monte egalment toute fois, nos approches présentent des

meilleurs résultats par rapport à MC et MU avec "random workload" comme charge de charge. Les resultats sont presques dans le même ordre de grandeur qu'il s'agisse du workload random ou "20110303" avec comme politique d'allocation EBFDHybrid.Cependant ces resultats changent lorssque nous appliquons la politique d'allocation Local Regression Robust (LRR) comme on pourra le voir plutard dans une autre section.

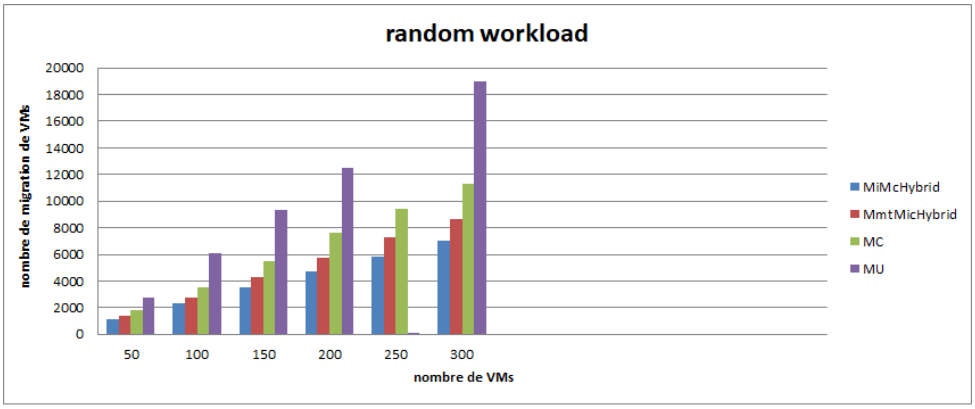

Figure 4.5 – nombre de migrations de VMs

D'après les resultats à l'issue de l'expérience, l'approche MiMcHybrid est celle qui rduit le plus le nombre de mgration tandis que l'approche MU admet une migration agressive de VMs du fait d'une surutilisation des ressources physiques de la Ram et du CPU definies. MiMcHybrid migre moins par rapport aux autres approches, s'ensuit de MmtMiMcHybrid. L'approche MC admet moins de migrations comparée à MU dont la barre surplombe toutes les autres progressivement que le nombre de VMs augmente.MU atteint son pic avec le nombre maximal de VMs lancées dans les hôtes.

Nombre de VMs	50	100	150	200	250	300
MiMcHybrid	1142	2302	3491	4721	5822	17035
MmtMiMcHybrid	1433	2802	4272	5731	7283	8633
MC	1798	3589	5489	7604	9395	11262
MU	2729	6056	9296	12492	16305	19008

Table 4.2 – nombre de migrations de VMs

Nous remarquons que lorsque le nombre de VMs est à 50, l'approche MiMcHybrid réduit le nombre de migration de plus de moitié par rapport à l'approche MU. Cependant la différence par

rapport à MmtMiMcHybrid est très nette, et plus nette encore comparée à l'approche MC. Au fur et à mesure que le nombre de VMs augmente, MU triple quatiment le nombre de migrations par rapport à notre approche MiMcHybrid. Cela dit, les autres également ont un nombre de migrations de VMs plus élevé tout au long de la simulation par rapport à MiMcHybrid. Il convient de rappeler que le workload utilisé est random dans le cadre de cette simulation.

4.4.2.3 Violation de SLAs

Suite aux simulations effectuées, les resultats concernant la métrique violation des SLAs comme l'illustre la (Figure 4.6) et le (Table 4.3), l'approche MiMcHybrid semble être celle qui présente plus de violation des SLAs par rapport aux autres. En debut de simulation, avec 50 VMs, nous constatons plus de violation des SLAs de la part de l'approche MiMcHybrid sur MmtMiMcHybrid.

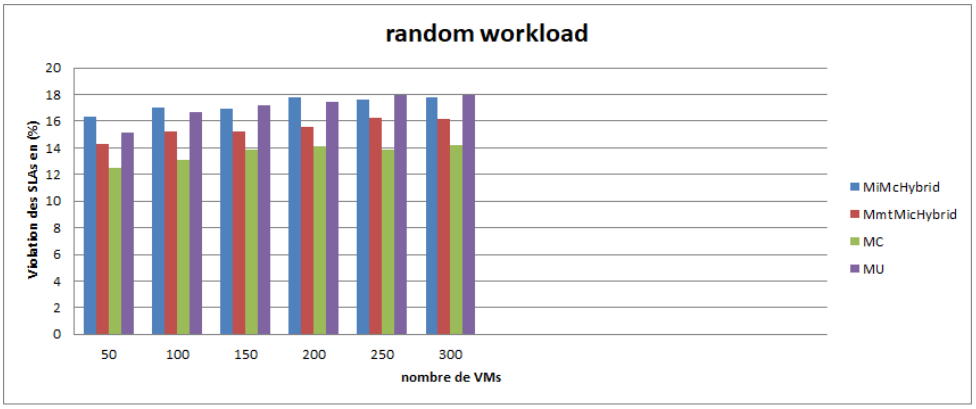

Figure 4.6 – Violation de SLAs

L'approche MC est celle qui engrange le moins de violation de SLA de façon significative. Notre approche MiMcHybrid varie en fonction de l'augmentation du nombre de VMs, la valeur se stablise plus ou moins lorsqu'on atteint la barre de 150 VMs tandis que les barres des autres approches tendent à monter progressivement jusqu'à depasser celle de MiMcHybrid, qui elle baisse de manière négligeable. Tout au long de la simulation, la violation de SLA de MC est restée inferieure à toutes les autres. MU et MmtMiMcHybrid ont des valeurs approximatives vers la fin. Cela s'explique par la vérification de l'utilisation du CPU et de la Ram. Plus il y a de migrations plus il y a le risque de violation de SLA.

Comme c'est indiqué dans le(Table 4.3),l'approche MC a reduit la violation de SLA de l'ordre

Nombre de VMs	50	100	150	200	250	300
MiMcHybrid	16,31	17	16,93	17,82	17,58	17,78
MmtMiMcHybrid	14,25	15,26	15,26	15,53	16,28	16,16
MC	12,47	13,13	13,88	14,11	13,82	14,20
MU	15,13	16,66	17,23	17,46	17,94	17,97

Table 4.3 – Violation de SLAs des differentes approches

de 4% par rapport à MiMcHybrid, 2% de plus que MmtMiMcHybrid et depasse bien l'approche MC dans cette métrique. En augmentant le nombre de VMs, la barre des autres approches haussent legèrement. Nous pouvons noter ainsi tout de même que notre approche MmtMiMcHybrid depasse MU en terme de réduction de violation de SLA sur toute la ligne, de ce fait elle est préeminente dans cette métrique. Elle est en tête egalment par rapport à MiMcHybrid, qu'elle depasse legèrement. L'ecart entre les approches en terme de violation de SLAs n'est pas très considerable à une echelle du nombre de VMs relativement grand sachant que le workload utilisé est random et la politique d'allocation est EBFDHybrid

4.4.3 Deuxième Expérience : Un nombre réduit de PMs et VMs

Dans le processus de notre simulation, nous aons dans le cadre cette deuxième expérience considéré 10 machines physiques et 40 machines virtuelles hétérogènes.Le nombre de VMs passe de 10 à 40 sur un pas de 10 grâce au Broker.

4.4.3.1 Consommation d'energie

Il est observable qu'au vu des resultats des simulation, la reduction de la consommation d'energie est ressentie avec l'approche MiMcHybrid qui reduit la consommation par rapport à MMtMiMcHybrid. Avec 10 VMs la consommation d'energie des autres approches est identique.

C'est à partir de 30 VMs qu'une legère difference s'apperçoit entre ces dernières. Cependant l'approche MiMcHybrid a reduit de justesse la consommation d'energie par rapport aux autres. Cette reduction est dûe au fait que l'approche controle l'utilisation des ressources physiques, en cas de violation de ces ressources les VMs sont migrées ce qui allège la PM et reduit le besoin en energie. Le workloadd utilisé etatit random et la politique d'allocation EBFDHybrid.

Tout comme avec la précedente expérience, celle-ci montre que la réduction de la consommation

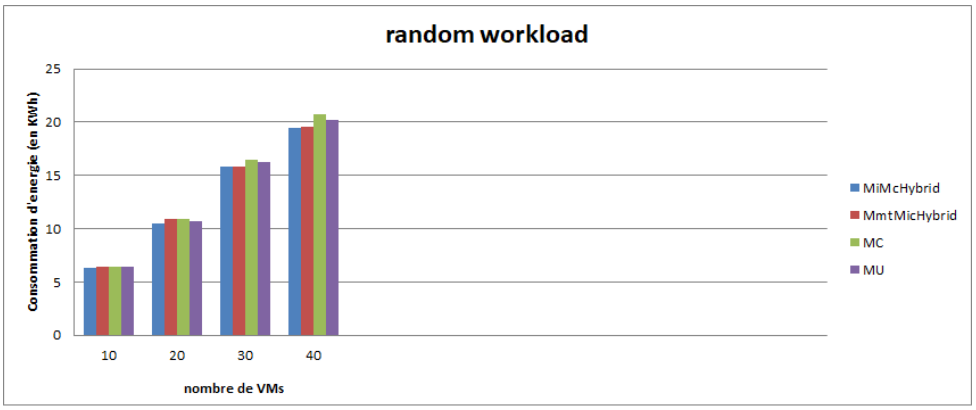

Figure 4.7 – Consommation d'energie

Nombre de VMs	10	20	30	40
MiMcHybrid	6,33	10,52	15,82	19,46
MmtMiMcHybrid	6,40	10,88	15,87	19,58
MC	6,42	10,91	16,42	20,75
MU	6,42	10,74	16,30	20,26

Table 4.4 – consommation d'energie

d'energie vient de l'approche MiMcHybrid, qui depasse les autres approches d'une valeur minime.

4.4.3.2 Nombre de migrations de VMs

L'atout de nos approches resulte dans la reduction du nombre de migration de VMs. Comme le montre la(Figure 4.8) ainsi que le (Table 4.5), le nombre de migrations de VMs est autant petit que le nombre de VMs dans le cadre de cette deuxième expérience avec le nombre de VMs réduit. Le nombre de migrations de VMs des differentes approches est petit proportionnellement au nombre de VMs utlisées pour la simulation. Ce faisant, nos approches ont reduit le nombre de VMs de manière considrable comparées aux approches MU et MC. L'approche MiMcHybrid est celle qui a le plus reduit le nombre de migration.

En nous réferant aux resultats des simulations prsentés dans le tableau cidessous, nous observons que l'approche MiMcHybrid a reduit le nombre de migration de VMs de 172 par rapport à MU et de 43 en comparaison avec MC. EN revanche, avec une première simulation de 10 VMs au total, MmtMiMcHybrid a présenté des meilleures rsultats que l'approche MiMcHybrid soit un

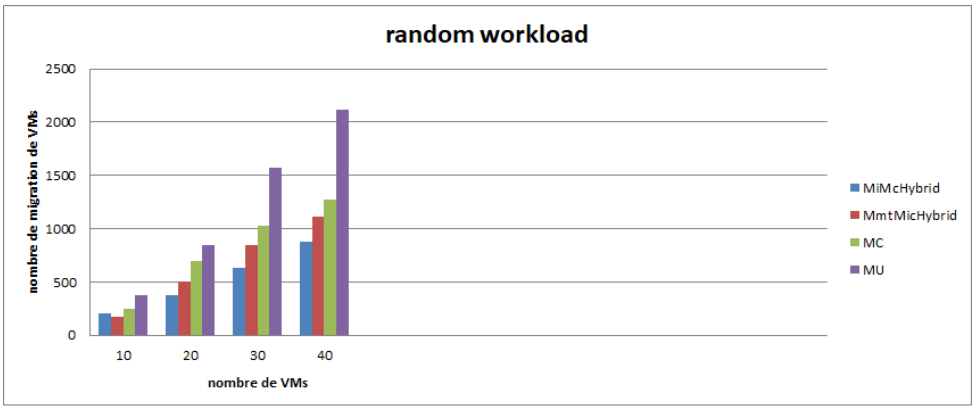

Figure 4.8 – nombre de migrations de VMs

écart d'au moins 25.A partir de 20 VMs à plus dans les hôtes, l'approche MiMcHybrid surpasse toutes les autres dans la réduction du nombre de migrations avec une nette performance qui est visible à travers le graphe. MU a été l'approche qui a migré le plus de VMs par rapport aux autres.En sommes nous pouvons dire que nos approches ont reduit le nombre de migrations plus que quelconque autre approche dans cette expérience.

Nombre de VMs	50	100	150	200
MiMcHybrid	203	375	638	878
MmtMiMcHybrid	178	503	849	1117
MC	246	694	1030	1274
MU	375	847	1569	2120

Table 4.5 – nombre de migrations de VMs

4.4.3.3 Violation de SLAs

Au debut de la simulation, pour un nombre de VMs minimal c'est-à-ire envion 10 VMs jusqu'à 20VMs, nos approches ont revélé des réductions de violations de SLAs, ce qui leur permet de l'emporter fâce à MU et MC. En revance avec l'augmentation du nombre de VMs dans les hôtes, cela a inversement dimunié la violation des SLAs pour les approches MC et MU au detriment des notres. Lorsqu'une approche pose des contraintes d'utilisation des ressources CPU et RAM, cela a des repercutions sur le nombre de migration de VMs d'où une dimunition des violations des SLAs comme c'est lecas avec ces approches.

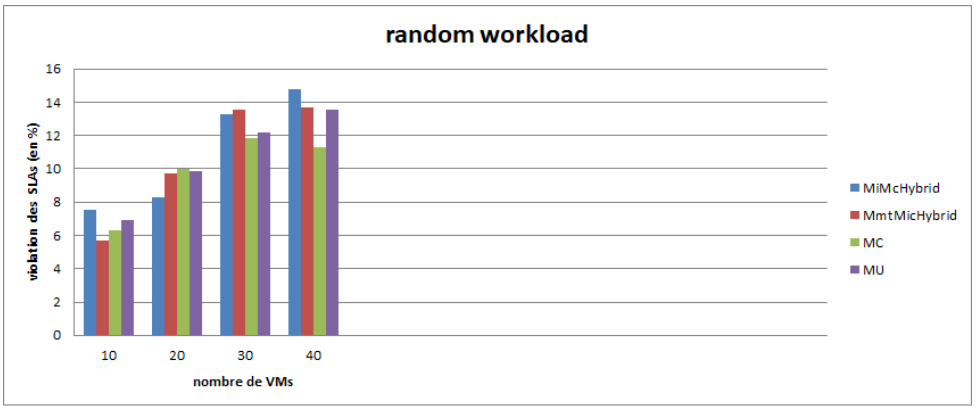

Figure 4.9 – violation de SLAs

Note : Comme nous l'avions annocé précedemment, la politique d'allocation utilisée était EBFDHybrid qui a servi de cadre à la serie de simulation que nous avons méné. Cependant nous n'avons pas manqué de tester nos approches avec une autre politique d'allocation nommée Local Regression Robust (LRR). D'après une simple comparaison faite entre cette approche et la montre donne des meilleurs en faveur de notre approche qui devance de loin cette approche Local Regression Robust (LRR) en terme de reduction d'energie. Avec tous nos algorithmes de selection et ceux présents dans la litterature en l'occurence MU et MC, les métriques sont beaucoup moins réduites en utilisant Local Regression Robust (LRR) qu'en utilisant notre politique d'allocation. Les résultats obtenus suite aux différentes simulations sont prsentés dans les figures et les que nous allons présenter.Il convient de noter que le workload est toujours random workload. Nous avons utilisé 100 hôtes et pour 300 VMs dans cette expérience avec LRR.

Il faudra remarquer qu'avec les résultats du graphe, l'approche LRR produit beaucoup plus d'energie que notre approche. Le pic depasse la barre des 160KWh avec les algorithmes de selection en utilisant LRR,comme l'illustre la (Figure 4.13) tandis que avec notre approche d'allocation la consommation d'energie est beaucoup plus faible avec le même workload comme il est montré dans la (Figure 4.4).

pareillement les resultats sont à-propos du nombre de migration de VMs connaissent la même évolution avec la politique LRR qui a tendance permettre plus de migration avec les quatres approches de selection utilisées. Nous constatons que le nombre de migration de VMs est plus élevé avec LRR dont le pic est de plus de 27.000 VMs

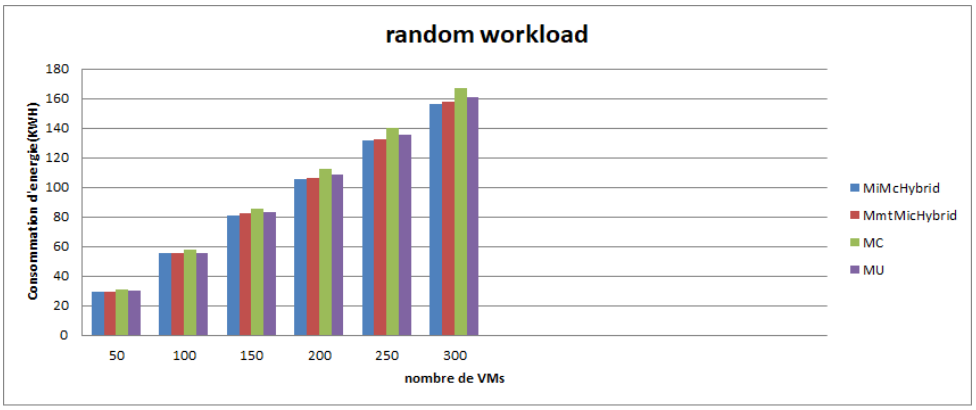

Figure 4.10 – Consommation d'energie avec l'approche LRR

migrées avec l'approche de sélection MU. Le pic atteint par une de nos approches est d'environ 13.000 nombre de migrations de VMs. Alors que pour les mêmes algorithmes avec notre approche d'allocation, la valeur jamais atteinte est 8633 et 70335 respectivement pour MmtMiMcHybrid et MiMcHybrid. Les (Figure 4.5) et (Figure 4.11) dont beaucoup plus de détails. Dans la foullée, nous allons présenter

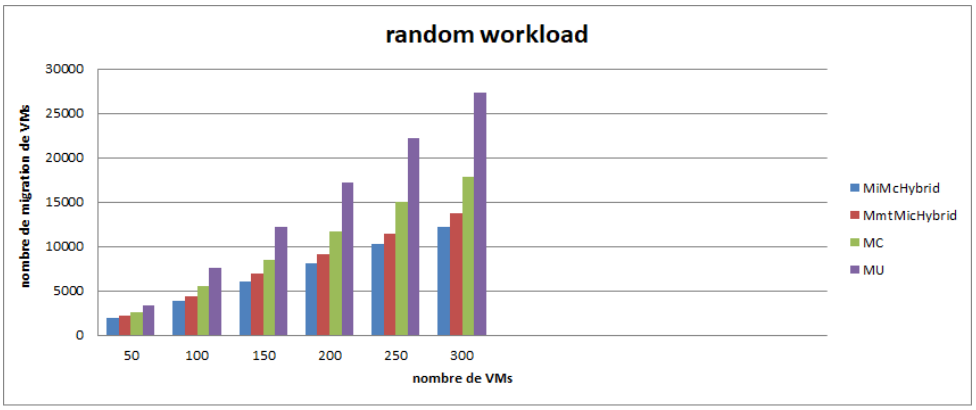

Figure 4.11 – Nombre de migrations de VMs avec l'approche LRR

les résultats avec la métriqueviolation de SLAs, qui quant à elle a dérangé à la règle de connaitre plus de résultats satisfaisants avec l'application de la politique LRR par rapport à notre politique d'allocation la (Figure 4.12) et la (Figure 4.6) présentent les comparaison de manière plus nette et concise. Cependant il est à préciser que la politique d'allocation PowerVmAllocationPolicyMigrationStaticThreshold présente des meilleures résultats avec bon nombre de métriques et cela dit, dépasse parfois notre politique d'alloaction dans la réduction de violation de SLA. Ce qui nous

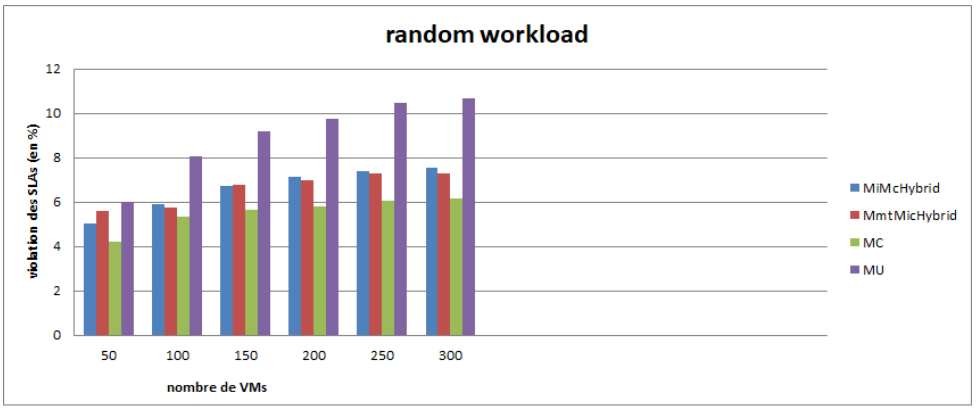

Figure 4.12 – violation de SLAs avec l'approche LRR

a aménè à tenter une troisième et dernière expérience avec commme politique d'allocation LRR et nos principaux algorithmes de selection de VMs, mais pour cette fois-ci nous avons changé de workload pour prendre en compte le workload "20110303".

4.4.4 Troisième Expérience : Un nombre réduit de PMs et VMs

Dans cette expérience ultime nous y avons mis en oeuvre 40 hôtes physiques et un nombre de VMs aussi 40 VMs hétérogènes. Le Broker se charge de les faire varier entre 10 jusqu'à 40 avec un pas toujours égal à 10.Tout au long de la simulation nous avons travaillé avec LRR.

4.4.4.1 Consommation d'energie

A la lecture des résultats recoltés, il est tout à fin évident que nos approches de selctions ainsi que MC et MC ont un taux de consommation d'energie plus aigus avec LRR comme politique d'allocation. ainsi avec la(Figure 4.13 nous assistons à une observation dans laquelle,les aproches ont presque la même consommation au debut de la simulation avec 10 VMS.

Par contre plus on augmente le nombre de VMS plus lentement augmente la difference de reduction d'energie remportée par nos approches de selection. Les résultats sont observables dans le Table 4.6. Difficile de savoir si le workload a influé sur les résultats, ce qui est sûr la politique d'allocation y est pour beaucoup bien qu'il

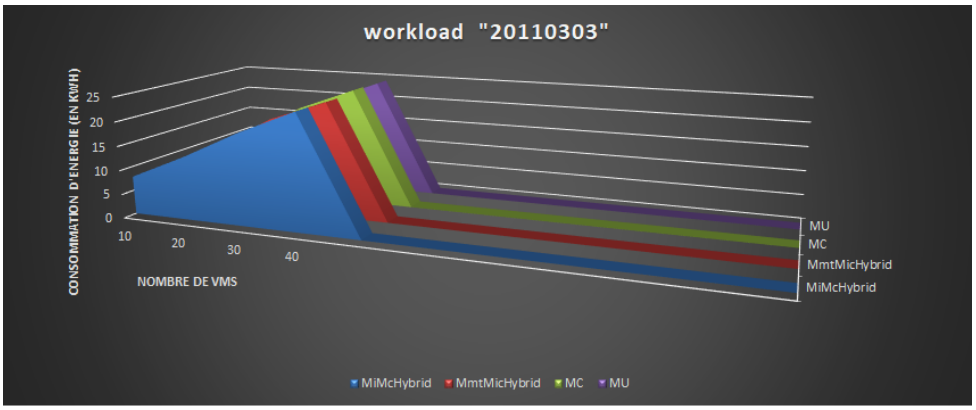

Figure 4.13 – Consommation d'energie

faudra noter que 20 hôtes ont ét utiliséé au lieu de 10 comme c'était le cas pour l'expérience 2.

Nombre de VMs	10	20	30	40
MiMcHybrid	7,88	13,11	18,93	23,98
MmtMiMcHybrid	7,87	12,99	19,49	23,78
MC	7,92	13,08	19,79	24,39
MU	7,73	13,2	19,44	24,14

Table 4.6 – consommation d'energie

4.4.4.2 Nombre de migration de VMs

La même remarque peut se vérifier avec cette métrique, nous remarquons déjà qu'avec la(Figure 4.14), la réduction du nombre de migration est plus prometeuse avec l'approche MiMcHybrid qui prend le dessus sur les autres approches. Plus on augmente de VMs plus les approches MC et MU voient le nombre de migration augmenter avec l'allure de chacune de leur courbe qui monte pendant la courbe de MiMcHybrid baisse vers la fin. Celle de MiMcHybrid lui emboîte les pas alors que l'allure de la courbe de MC essaie de joindre celle de MU, qui finit par générer le plus de migration.

Là aussi nous obtenons avec MiMcHybrid une réduction de cette métrique avec une

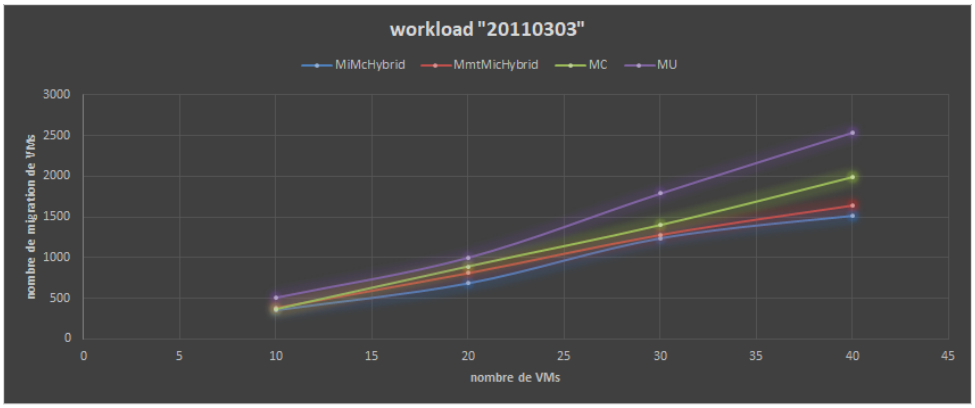

Figure 4.14 – nombre de migration de VMs

Nombre de VMs	50	100	150	200
MiMcHybrid	353	686	1242	1520
MmtMiMcHybrid	386	813	1283	1644
MC	369	892	13990	1986
MU	504	998	1791	2544

Table 4.7 – nombre de migrations de VMs

difference de 151 sur MU lorque le nombre de VMs mis est 10. La réduction est très significative lorsqu'on passe de 10 à 20 VMs jusqu'à 40 VMs. Nos approches ont réduit le nombre de migrations plus que toutes les autres.

4.4.4.3 Violation de SLA

En dépit d'utiliser un real workload trace, nous avons bouclé nos tests sur cette métrique,avec "20110303". Les resultats obtenus suite à la serie de simulation effectuée, montrent globalement que les valeurs des violations des SLAs sont significativement plus faibles. la (Figure 4.15) et le (Table 4.8) nous en disent plus clairement.

Comme le montre letableau,MiMcHybrid nous reduit la violation SLA plus que MU quand il s'agit de 10 VMs dans les hôtes. La plus faible valeur nous vient de l'approche MC. En revanche à partir de 20 VMs,MU commence à reduire la violation plus que toutes autres approches. Les approches connaissent tout de même des variations plus ou moins conséquentes en terme de violation de SLA jusqu'à

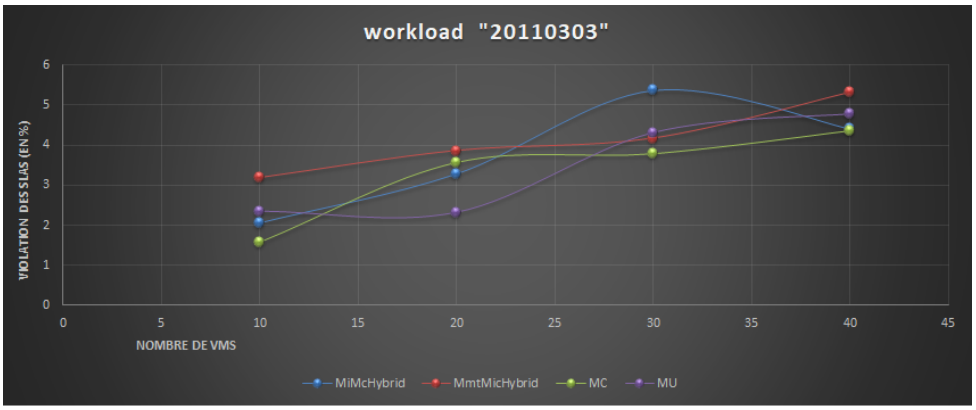

Figure 4.15 – Violation de SLAS

40 VMs, la courbe de MiMcHybrid baisse sa valeur. Cela dit plus le nombre de VMs est grand plus ça va nous exonérer de la violation de SLA dans ce cas précis d'application de l'approche LRR.

Nombre de VMs	10	20	30	40
MiMcHybrid	2,05	3,28	5,37	4,41
MmtMiMcHybrid	3,18	3,86	4,17	5,32
MC	1,58	3,57	3,79	4,36
MU	2,35	3,32	4,31	4,78

Table 4.8 – Violation de SLA

4.5 Conclusion

Le choix du simulateur CloudSim s'avère satisfaisant dans la mésure où, il nous a été possible de pouvoir y integrer nos algorithmes afin de les tester et comparer avec d'autres. CloudSim permet de faire plus que, notamment, vous pouvez etendre les classes, rajouter d'autres fonctions, créer des composants et ressources don vous avez besoin ou bien importer. Globalement nous pouvons dire que nos politiques MiMcHybrid,MMtMiMcHybrid,EBFDHybrid ont des meilleures performances vis-à-vis de plusieurs dont entre autres la consommation d'energie, nombre de migrations de VMs par rapport aux approches comme MU ou MC qui sont déjà presentes

dans la littérature. Néanmoins lorqu'il s'agit de la métrique violation de SLAs, nos approches se heurtent à une contre-performance d'un ordre minime,quoique.

Durant la phase de la première expérience avec un nombre de hôtes élevé et un nombre de machines virtuelles aussi élevé, notre approche MiMcHybrid a su minimiser la consommation d'energie, le nombre de migrations de VMs ainsi que les violations des SLAs. L'approche MmtMiMcHybrid à son tour a su améliorer ces métriques dans toutes les expériences. Avec un nombre limité de VMs, l'approche a aussi prouvé son efficacit par rapport à MU et MC, cependant elle reste derrière MiMcHybrid. Une consommation d'energie est plus accentuée par le data center avec les autres approches alors qu'en utilisant nos approches, il y a moins de gaspillage.

Nos approches étant caractérisées par l'utilisation du compte minimum de migration et l'otimisation d'energie surclassent les approches MU et MC en terme de réduction d'energie. Nous pouvons clairement affirmer qu'un des objectif a etteint à savoir minimiser la consommation des datacenter en energie pour permettre d'éviter le gaspillage d'energie par les systèmes de refroidissement. En somme nos approches feront figuration des approches de "Green Computing approach".

4.6 Conclusion Générale

Au cours de ces décenies, la part du Cloud Computing constitue une nouvelle ère pour le monde Informatique. Avec un principe plutôt satisfaisant du côté des bénéficiaires de ce service car les services sont fournis aux clients de manière décentralisée et suivant le modèle "pay-as-you-go" de consommation de ressources. Pour des raisons d'angouement et de demande de ce type de service, les fournisseurs se sont tournés vers l'extension des capacités des infrastructures ce qui entraine comme conséquence logique la consommation d'energie accrue, de même que les coûts ont décuplés. La question de la consommation d'energie et le management des ressources sont dévenues centrales dans le monde du Cloud Computing. Les centres de données de Cloud Computing étant des systèmes distribués, par conséquent gourmands en energie, consomme une quantité faramineuse d'energie électrique. Ceci a pour corrolaire l'émission phénoménale du dioxyde de carbone (CO2) et des répercutions sur la rentabilité des propriétaires su CLoud.Pour agir sur ces pro-

blématiques, les fournisseurs des services Cloud mettent en oeuvre des techniques pour optimiser l'utilisation des ressources et la consommation d'energie. Des techniques que nous pouvons citer : La virtualisation, la migration et la consolidation. La technique de la consolidation de VMs répresente un bouclier contre la consommation d'energie et améliore l'utilisation des ressources d'où son inclusion dans la technologie de virtualisation. LA virtualisation est en fait une manière de créer plusieurs instances de machines virtuelles dans un seul hôte physique d'où l'intérêt que lui portent les fournisseurs de Cloud afin d'augmenter leurs renvenus.La virtualisation est également la matrice de la migration à chaud, technique de transfert de machines virtuelles entre serveurs physiques. Il est essentiel de veiller sur les resources physiques telles ques CPU ou la RAM avant d'allouer des ressources est crucial aujourd'hui dans le monde du Cloud.

Afin de contribuer à optimiser la consommation d'energie, nous avons proposé trois politiques dans le cadre de notre travail. Ces approches sont MiMcHybrid, MmtHybrid et EBFDHybrid respectveement les deux premeières assurent la selection et la troisième est utilisée pour faire l'allocation des VMs. Elles sont basés sur l'utilisation de compte minimum de migrations de VMs, sur l'utilisation du CPU et la capacité de la RAM. Dans la réalisation de cet objectif de minimisation de la consommation d'energie, nos approches ont prouvé leur efficacité en réduisant également la quntité du CO2 émise, reduction de la chaleur produite par les serveurs.

Dans le cadre de notre simulation, il a été question d'étudier le comportement de nos propres approches et compaparer les résultats avec les autres approches. Les simulations effectuées sur CloudSim à travers une série d'expériences ont produits des résultats concluants. Les métriques mises en avant sont la consommation d'energie, le nombre de migrations, la violation SLAs.

Sur l'ensemble des expériences effectués, nous avons constaté que nos approches sont devant les approches MU et MC en terme d'amélioration de ces trois métriques précitées. Nos approches sont montré un côté rassurant envue de figurer parmi les approches de "Green Computing" au terme de nos travaux.

Dans les travaux futurs nous projetons de prendre en consideration la bande passante comme ressource clé pour la phase de sélection de nos algorithmes et aussi contrôler les données transférées avec des paramètres de sécurité au cours de la

migration des VMs. Nous projetons d'affiner nos stratégies en introduisant les contraintes SLAs, avec les pénalités de ses violations afin que notre propsition surpasse les autres dans cette performance. Nous essayerons également d'améliorer ces approches toutes confondues.Nous allons intégrer d'autres paramètres pour la métrique consommation d'énergie comme l'utilisation la bande passante, le système de réfroidissment pour obtenir des meilleurs performances plus tard. ENfin notre ultime objectif sera d'implémenter notre approche dans une plate-forme de cloud comme Open Stack pour plus de pertinence.

Bibliographie

[1] **Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms : Vision, hype, and reality for delivering computing as the 5th utility.** ***Future Generation computer systems*****, 25(6) :599–616, 2009.**

[2] **Richa Sinha, Nidhi Purohit, and Hiteshi Diwanji. Power aware live migration for data centers in cloud using dynamic threshold. 2011.**

[3] **Anton Beloglazov and Rajkumar Buyya. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers.** ***MGC@ Middleware*****, 4(10.1145) :1890799–1890803, 2010.**

[4] **Bob DuCharme.** ***The Operating System Handbook or, Fake Your Way Through Minis and Mainframes.*** **IBM, 2001.**

[5] **Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization.** ***ACM SIGOPS Operating Systems Review*****, 37(5) :164–177, 2003. doi : 10.1145/1165389.945462.**

[6] **Yangyang Wu and Ming Zhao. Performance modeling of virtual machine live migration. In** ***2011 IEEE 4th International Conference on Cloud Computing*****, pages 492–499. IEEE, 2011.**

[7] **Gabor Kecskemeti, Péter Kacsuk, Thierry Delaitre, and Gabor Terstyanszky. Virtual appliances : a way to provide automatic service deployment. In** ***Remote Instrumentation and Virtual Laboratories*****, pages 67–77. Springer, 2010.**

[8] **Andrew Whitaker, Marianne Shaw, Steven D Gribble, et al. Denali : Light-**

weight virtual machines for distributed and networked applications. Technical report, Technical Report 02-02-01, University of Washington, 2002.

[9] Gerald J Popek and Robert P Goldberg. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7) :412–421, 1974.

[10] virtualisation des systèmes IBM, version 2. http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/eicay/eicay.pdf(2005). Accedé le : 2021-02-23.

[11] Xen and the Art of Virtualization, 2003 sosp. https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/xen/architecture.html. Accedé le : 2021-02-24.

[12] Type 1 and type 2 hypervisor,nakivo. https://www.nakivo.com/blog/hyper-v-virtualbox-one-choose-infrastructure/type-1-and-type-2-hypervisor/. Accedé le : 2021-02-24.

[13] what is virtual machine. https://www.elprocus.com/virtual-machine/. Accedé le : 2021-02-23.

[14] what is a datacenter for cloud. https://www.wikipedia.com/wiki/datacenter/. Accedé le : 2021-02-23.

[15] Jinkyun Cho, Joonyoung Yang, Changkeun Lee, and Jinyoung Lee. Development of an energy evaluation and design tool for dedicated cooling systems of data centers : Sensing data center cooling energy efficiency. *Energy and Buildings*, 96 :357–372, 2015.

[16] Inside Google's massive datacenter. https://www.datacenterknowledge.com/. Accedé le : 2021-02-24.

[17] D. M and Kannan. P. *A Study On Virtualization Techniques And Challenges In Cloud Computing*, volume v3. International Journal of Scientific and Technology Research, 11 edition, 2014.

[18] concept of virtualization,researchgate. https://www.researchgate.net/figure/illustration-of-the-concept-of-Virtualization-7_fig1_269636339/, . Accedé le : 2021-02-23.

[19] emulation,wikipedia. https://fr.wikipedia.org/wiki/Fichier:Diagramme_ArchiEmulateur.png. Accedé le : 2021-02-21.

[20] la virtualisation,k. lefevre. https://fr.wikipedia.org/wiki/Fichier:Diagramme_ArchiEmulateur.png, . Accedé le : 2021-02-21.

[21] full virtualization. https://networksandservers.blogspot.com/2011/11/full-virtualization-explained.html,. Accedé le : 2021-02-21.

[22] la paravirtualisation,blogspot. https://networksandservers.blogspot.com/2011/11/full-virtualization-explained.html, . Accedé le : 2021-02-21.

[23] la paravirtualisation,k. hwang. http://www.itpro.fr/a/para-virtualisation, . Accedé le : 2021-02-21.

[24] les avantages de la virtualisation,f.beland. http:www.agentsolo.com/ca/fr/blog/fbsc/les-avantages-de-la-virtualisation. Accedé le : 2021-02-21.

[25] L. MALLU and E.R. *live migration of virtual machines in cloud Environnement a : survey*, volume v8. International Journal of Computer Applications, 9 edition, 2014.

[26] Gulshan Soni and Mala Kalra. Comparative study of live virtual machine migration techniques in cloud. *International Journal of Computer Applications*, 84(14), 2013.

[27] S Liyanage, S Khaddaj, and J Francik. Virtual machine migration strategy in cloud computing. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 147–150. IEEE, 2015.

[28] optimization,gurobi. www.gurobi.com. Accedé le : 2021-02-21.

[29] cold migration. https://www.researchgate.net/figure/VM-Cold-Migration-Prozess-eigene-Darstellun. Accedé le : 2021-02-21.

[30] P.D Patel M. Karamta M. D. Bhavsar. *Live virtual machine techniques in cloud computings : A surver," International Journal of Computer Applications*, 86 (16), 2014.

[31] **Ompriya Kale L.J.I.E.T Ahmedabad. Virtual machine migration techniques in cloud environment : A survey.** ***International Journal for Science and Development*****, 8(1) :1635–1638, 2013.**

[32] **live migration. https://www.semanticscholar.org/paper/Live-Migration-of-Virtual-Machines-in-Cloud-Agarwal-Raina/. Accedé le : 2021-02-21.**

[33] **post-copy migration. \T1\textquotedblright,https://www.semanticscholar.org/paper/Post-copy-live-migration-of-virtual-machines-Hines-Deshpande/. Accedé le : 2021-02-21.**

[34] **D MARSHALL. The rise of virtual systems and virtualization (on-line), virtualizationsdefrag. com editors, 2009[cit. 2011-06-07].**

[35] **Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines in : Proceedings of the 2nd conference on symposium on networked systems design & implementation-volume 2, 273–286.** ***USENIX Association*****, 2005.**

[36] **Aidan Shribman and Benoit Hudzia. Pre-copy and post-copy vm live migration for memory intensive applications. In** ***European Conference on Parallel Processing*****, pages 539–547. Springer, 2012.**

[37] **Jacob Gorm Hansen and Eric Jul. Self-migration of operating systems. In** ***Proceedings of the 11th workshop on ACM SIGOPS European workshop*****, pages 23–es, 2004.**

[38] **Jacob G Hansen and Asger K Henriksen.** ***Nomadic operating systems*****. PhD thesis, Citeseer, 2002.**

[39] **Jochen Liedtke. On micro-kernel construction.** ***ACM SIGOPS Operating Systems Review*****, 29(5) :237–250, 1995.**

[40] **Marvin M Theimer, Keith A Lantz, and David R Cheriton. Preemptable remote execution facilities for the v-system.** ***ACM SIGOPS Operating Systems Review*****, 19(5) :2–12, 1985.**

[41] **Charles E Perkins and A Myles. Mobile ip, ieee communications magazine. *doi*, 10(35.592101) :84–99, 1997.**

[42] **Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *ACM SIGOPS operating systems review*, 37(5) :164–177, 2003.**

[43] **Xuanjia Qiu, Hongxing Li, Chuan Wu, Zongpeng Li, and Francis CM Lau. Cost-minimizing dynamic migration of content distribution services into hybrid clouds. *IEEE Transactions on Parallel and Distributed Systems*, 26(12) :3330–3345, 2014.**

[44] **P Getzi Jeba Leelipushpam and J Sharmila. Live vm migration techniques in cloud environment—a survey. In *2013 IEEE Conference on Information & Communication Technologies*, pages 408–413. IEEE, 2013.**

[45] **Song Fu. Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing. In *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 372–379. IEEE, 2009.**

[46] **Emmanuel Arzuaga and David R Kaeli. Quantifying load imbalance on virtualized enterprise servers. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, pages 235–242, 2010.**

[47] **Anton Beloglazov and Rajkumar Buyya. Energy efficient allocation of virtual machines in cloud data centers. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pages 577–578. IEEE, 2010.**

[48] **Laurent Lefèvre and Anne-Cécile Orgerie. Designing and evaluating an energy efficient cloud. *The Journal of Supercomputing*, 51(3) :352–373, 2010.**

[49] **Suhib Bani Melhem, Anjali Agarwal, Nishith Goel, and Marzia Zaman. Minimizing biased vm selection in live vm migration. In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pages 1–7. IEEE, 2017.**

[50] **Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power provisioning**

for a warehouse-sized computer. *ACM SIGARCH computer architecture news*, 35(2) :13–23, 2007.

[51] Saad Mustafa, Babar Nazir, Amir Hayat, Sajjad A Madani, et al. Resource management in cloud computing : Taxonomy, prospects, and challenges. *Computers & Electrical Engineering*, 47 :186–203, 2015.

[52] Anton Beloglazov. *Energy-efficient management of virtual machines in data centers for cloud computing.* PhD thesis, 2013.

[53] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in clouds : A performance evaluation. In *IEEE International Conference on Cloud Computing*, pages 254–265. Springer, 2009.

[54] Awada Uchechukwu, Keqiu Li, Yanming Shen, et al. Energy consumption in cloud computing data centers. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 3(3) :31–48, 2014.

[55] Anton Beloglazov and Rajkumar Buyya. Openstack neat : a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds. *Concurrency and Computation : Practice and Experience*, 27(5) :1310–1333, 2015.

[56] Jangha Kang and Sungsoo Park. Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147(2) :365–372, 2003.

[57] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. Cloudsim : a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software : Practice and experience*, 41(1) :23–50, 2011.

[58] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N Calheiros. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit : Challenges and opportunities. In *2009 international conference on high performance computing & simulation*, pages 1–11. IEEE, 2009.

[59] The architecture of CloudSim. \T1\textquotedblright,https://www.

researchgate.net/figure/The-Architecture-of-CloudSim-25_fig1_277722890/. Accedé le : 2021-05-25.

[60] Dimple Maheshwari, Purnima Gandhi, and Richa Sinha. Energy efficient threshold based approach for migration at cloud data center. *Int. J. Eng. Res. Technol.(IJERT)*, 1(10), 2012.

[61] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5) :755–768, 2012.