

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة عين تموشنت بلحاج بوشعيب  
Université -Ain Temouchent- Belhadj Bouchaib  
Faculté des Sciences et de Technologie  
Département des Mathématiques et Informatique



Projet de Fin d'Etudes  
Pour l'obtention du diplôme de Master en : Informatique.  
Domaine : Mathématique et Informatique  
Filière : Informatique  
Spécialité : Réseaux et ingénierie des données

Thème

**Etude comparative des algorithmes dédiés à l'ordonnancement  
dans un environnement de Cloud Computing**

Présenté par :

- 1) Melle MEGROUSSE Marwa.
- 2) Melle YEBBAL Sarra.

Devant le jury composé de :

Dr MEDEDJEL	MCA	UAT.B.B (Ain Temouchent)	<b>Président</b>
Dr BOUCHAKOUR	MCA	UAT.B.B (Ain Temouchent)	<b>Examineur</b>
Dr BOUAFIA	MAA	UAT.B.B (Ain Temouchent)	<b>Encadreur</b>

**Année Universitaire : 2022/2023**

## REMERCIEMENT

*Après avoir rendu grâce a Dieu le tous puissant et le miséricordieux nous tenons absolument a remercier tous ceux qui nous ont soutenu, contribuer et participer a la rédaction de ce document ainsi qu'a la réalisation et le bon fonctionnement de notre travail. Il s'agit plus particulièrement de :*

*Notre encadreur **Monsieur BOUAFIA Zoheir** a qui on exprime notre profonde reconnaissance pour sa disponibilité, sa rigueur scientifique et son sens d'écoute et d'échange.*

*Nous tenon tout d'abord à remercier les membres du jury pour leur présence, pour leur lecture attentive de notre thèse ainsi que pour les remarques qu'ils m'adresseront lors de cette soutenance afin d'améliorer notre travail.*

*Un vif remerciement a notre source de motivation qui n'ont jamais cessé de nous guider tout au long de notre vie qui ont tout sacrifié pour notre bien-être a nos chers parents que Dieu les préserve et leur accorde une longue vie jusqu'à ce qu'ils nous voient répondre a leur attentes.*

# DEDICACES

*A ma très chère mère*

*Quoi que je fasse ou que je dise, je ne serai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.*

*A mon très cher père*

*Tu as toujours été à mes côtés pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection.*

*A mes très chers frères Mohamed Ali et Nadir et ma jolie sœur Loubna*

*A ma chère binôme Sarra « Wassa »*

*A ma chère amie yousra*

*Puisse Dieu vous donne santé, bonheur, courage et surtout réussite*

**MARWA.**

# DEDICACES

*Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,*

*A la mémoire de ma grande mère C'est à la personne la plus idéale dans ce monde, que je le dédie C est vrai quelle n'est pas avec moi, mais, elle reste toujours la plus présente.*

*A mes chers parents*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être*

*Maman, Papa je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*A mon très cher petit frère Mohamed Yassine tu es le meilleur frère que l'on puisse demander. Et je me rends compte que j'ai vraiment de la chance.*

*A mes tantes Farida, Souhila merci d'être les meilleure et les plus cool tantes de la planète Merci à tous d'avoir été présent à mes côtés.*

*A toute ma famille merci pour toute votre aide et votre soutien continu Merci pour la faveur que vous m'avez faite.*

*A ma meilleure amie Asma, je veux te dire merci d'être là dans les bons et les moins bons moments. Merci d'être dans ma vie. Merci de m'avoir aidé lorsque j'avais besoin d'aide.*

*A ma chère binôme Marwa, mon amie qui a eu la patience de me supporter durant ce mémoire, et qui m'a soutenu, merci ma chère*

*À tous mes amis que j'aime (khadidja, Chourouke, Chaima, Malek, Chouchou, Amina, Ferial, Zaho).*

*Je vous aime tous*

**SARRA**

## Résumé

Le Cloud Computing présente une technologie prometteuse qui facilite l'exécution des applications de divers domaines. Il fournit des services flexibles et évolutifs à la demande des utilisateurs, et offre un accès transparent aux applications à grande consommation de ressources. Ces avantages ne sont possibles que par l'utilisation d'une technique d'ordonnancement des tâches efficace.

Dans ce travail, nous proposons une étude comparative entre les algorithmes dédiés à l'ordonnancement des tâches avec le simulateur CloudSim et selon plusieurs performances comme Makespan, coût, consommation d'énergie et fiabilité.

Mots clés : Cloud Computing, Ordonnancement des tâches, CloudSim, Performances

## **Abstract**

The Cloud Computing presents a promising technology that facilitates the execution of applications from various domains. It provides flexible scalable services based on user demand and offers seamless access to resources- intensive applications. These advantages are only possible through the use of an effective task scheduling technique.

In this work, we propose a comparative study between task scheduling algorithms using the CloudSim simulator, considering several performance metrics such as Makespan, cost, energy consumption, and reliability.

Key words: Cloud Computing, Task scheduling, CloudSim, Performances

## ملخص:

الحوسبة السحابية تقدم تقنية واعدة تسهل تنفيذ التطبيقات من مجالات مختلفة. إنها توفر خدمات مرنة و قابلة للتوسع بناء على طلب المستخدمين و توفر وصولا سلسا إلى التطبيقات التي تستهلك موارد كبيرة. هذه المزايا ليست ممكنة إلا من خلال استخدام تقنية فعالة لجدولة المهام. في هذا العمل، نقترح دراسة مقارنة بين الخوارزميات المخصصة لجدولة المهام باستخدام محاكي CloudSim ، مع النظر في عدة مقاييس أداء مثل (Makespan) ، التكلفة، استهلاك الطاقة ، و الموثوقية.

الكلمات المفتاحية: الحوسبة السحابية ، جدولة المهام ، CloudSim، الأداءات.

# Table des matières

**INTRODUCTION GENERALE .....1**

## **CHAPITRE I :**

### **Généralités sur le Cloud Computing**

**Introduction :.....4**

**I. Historique :.....4**

**II. Définitions :.....5**

**III. Les caractéristiques du Cloud Computing :.....6**

    III.1 Libre-service à la demande : .....6

    III.2 Large accès au réseau : .....6

    III.3 Mise en commun des ressources : .....6

    III.4 Élasticité rapide: .....7

    III.5 Service mesuré : .....7

**IV. Les éléments nécessaires du Cloud Computing :.....7**

    IV.1 La virtualisation: .....7

        IV.1.1 Définition : .....8

        IV.1.2 Le principe de la virtualisation : .....8

        IV.1.3 Les techniques de la virtualisation : .....13

        IV.1.4 Les types de virtualisation : .....16

        IV.1.5 Les avantages de la virtualisation : .....18

    IV.2 Data center : .....19

**V. Les services de Cloud Computing :.....21**

    V.1 IaaS (infrastructure as a service) : .....21

    V.2 PaaS (plateforme as a service) : .....22

    V.3 SaaS (software as a service): .....23

**VI. Les modèles de déploiement dans le Cloud Computing:.....24**

    VI.1 Cloud privé : .....24

        VI.1.1 Les caractéristiques de Cloud privé : .....24

        VI.1.2 Les avantages de Cloud privé: .....24

        VI.1.3 Les inconvénients de Cloud privé : .....24

    VI.2 Cloud public : .....25

        VI.2.1 Les caractéristiques de Cloud public : .....25



## Table des matières

VI.2.2	Les avantages de Cloud public:.....	26
VI.2.3	Les inconvénients de Cloud public:.....	26
VI.3	Cloud hybride : .....	27
VI.3.1	Les caractéristiques de Cloud hybride : .....	27
VI.3.2	Les avantages de Cloud hybride : .....	27
VI.3.3	Les inconvénients de Cloud hybride : .....	27
VI.4	Cloud communautaire: .....	28
VI.4.1	Les avantages du Cloud communautaire : .....	28
VI.4.2	Les inconvénients du Cloud communautaire : .....	28
<b>VII.</b>	<b>Les avantages et les inconvénients de Cloud Computing :.....</b>	<b>29</b>
<b>I.</b>	<b>Ordonnancement :.....</b>	<b>32</b>
I.1	Définition : .....	32
<b>II.</b>	<b>L'ordonnancement dans un environnement de Cloud Computing :.....</b>	<b>33</b>
II.1	Le rôle d'ordonnanceur : .....	34
II.2	Les techniques d'ordonnancement : .....	34
II.2.1	Ordonnancement statique : (hors ligne) :.....	35
II.2.2	Ordonnancement dynamique : .....	35
II.2.3	Ordonnancement centralisé \ décentralisé :.....	36
II.3	Migration des machines virtuelles :.....	36
II.3.1	Les processus de migration : .....	37
II.3.2	Les approches de migration :.....	38
II.3.3	L'approche basée sur la consommation énergétique :.....	39
II.4	Les algorithmes d'ordonnancement des tâches dans les Cloud Computing :.....	39
II.4.1	Classification des algorithmes d'ordonnancement :.....	40
II.4.2	Les heuristiques et les métas heuristiques: .....	40
II.4.3	Les algorithmes d'ordonnancement classique (de bases) des tâches : .....	41
II.4.4	Ordonnancement basée sur algorithme Génétique simple : .....	48
II.4.5	Comparaison entre les différents algorithmes d'ordonnancement des tâches :.....	52
<b>I.</b>	<b>Environnement de travail : .....</b>	<b>57</b>
I.1	Environnement matériel : .....	57
I.2	Environnement de développement: .....	57
I.2.1	Langage de programmation Java : .....	57
I.2.2	L'environnement de développement Eclipse : .....	58
I.2.3	Le Simulateur CloudSim :.....	58

## Table des matières

---

---

<b>II. Les critères de performances :</b> .....	<b>67</b>
II.1 Temps total d'exécution (Makespan) : .....	67
II.2 Le coût : .....	68
II.3 La consommation d'énergie :.....	68
II.4 La fiabilité :.....	69
<b>III. Implémentation :</b> .....	<b>69</b>
III.1 Environnement de simulation: .....	71
III.1.1 Paramètres de l'environnement : .....	71
III.2 Résultats et discussion: .....	73
III.2.1 Discussions :.....	75
III.2.2 Résultat d'énergie : .....	76
III.2.3 Discussion: .....	77
III.2.4 Résultat de coût : .....	78
III.2.5 Discussion : .....	78
III.2.6 Résultat de La fiabilité : .....	79
III.2.7 Discussion : .....	80
III.2.8 Synthèse : .....	80
<b>I. Pour les articles :</b> .....	<b>84</b>
<b>II. Pour les mémoires :</b> .....	<b>84</b>
<b>III. Pour les sites web :</b> .....	<b>85</b>

### La Liste des abréviations

<b>Abréviation</b>	<b>Détail</b>
<b>CC</b>	<b>Cloud Computing</b>
<b>NIST</b>	<b>National Institute of Standards and Technology</b>
<b>RAM</b>	<b>Mémoire vive(Random Access Memory)</b>
<b>CPU</b>	<b>Central Processing Unit</b>
<b>DAS</b>	<b>Direct Attached Storage</b>
<b>NAS</b>	<b>Network Attached Storage</b>
<b>SAN</b>	<b>Storage Area Network</b>
<b>VMNIC</b>	<b>Virtual Network Interface Card</b>
<b>LAN</b>	<b>Local Area Network</b>
<b>VLAN</b>	<b>Virtual Local Area Network</b>
<b>IaaS</b>	<b>Infrastructure as a Service</b>
<b>PaaS</b>	<b>Software as a Service</b>
<b>VPN</b>	<b>Virtual Private Network</b>
<b>AWS</b>	<b>Amazon Web Service</b>
<b>VM</b>	<b>Machine Virtuelle</b>
<b>PM</b>	<b>Machine Physique</b>
<b>QoS</b>	<b>Qualité of Service</b>
<b>RR</b>	<b>Round Robin</b>
<b>PSO</b>	<b>Optimisation par essais Particulaires</b>
<b>FCFS</b>	<b>First Come First Served</b>
<b>SJF</b>	<b>Short Job First</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>IDE</b>	<b>Integrated Development Environment</b>
<b>SCSI</b>	<b>Small Computer System Interface</b>
<b>IP</b>	<b>Internet Protocol Address</b>

### Listes des figures

Figure I.1:Le Cloud Computing [5].....	6
Figure I.2: Les caractéristiques de Cloud Computing [14].....	7
Figure I.3: Les services de Cloud Computing. [28].....	21
Figure I.4: Modèle classique et modèle Cloud [25]. ....	23
Figure I.5: Mode de déploiement de Cloud privé [25]. ....	25
Figure I.6: Mode de déploiement de Cloud public [25] .....	27
Figure I.7: Mode de déploiement de Cloud hybride [25].....	28
Figure I.8: Mode de déploiement de Cloud communautaire [25].....	29
Figure II.1:Les techniques d'ordonnancement [16] .....	35
Figure II.2:Migration des machines virtuelles [7].....	37
Figure II.3:Migration a chaud [40] .....	38
Figure II.4:Le processus de l'algorithme aléatoire [12] .....	41
Figure II.5:Le processus d'algorithme RR [12] .....	43
Figure II.6:Processus d'algorithme SJF. [32].....	46
Figure III.1:Architecture de CloudSim. [26] .....	60
Figure III.2:Les packages de bases de CloudSim.....	63
Figure III.3:Le package CloudSim-3.0.3/Exemples. ....	63
Figure III.4: Le package org.cloudbus.cludsim. ....	64
Figure III.5: Le package org.cloudbus.cludsim.lists. ....	64
Figure III.6: Le package org.cloudbus.cludsim. Core. ....	65
Figure III.7 : Le package org.cloudbus.cloudsim.distributions.....	65
Figure III.8: Le package org.cloudbus.cloudsim.provisioners. ....	66
Figure III.9: Le package org.cloudbus.cloudsim.power. ....	67
Figure III.10: Fenêtre principale de l'application. ....	70
Figure III.11: Fenêtre de configuration de l'algorithme génétique.....	70
Figure III.12: Fenêtre d'affichage des résultats. ....	71
Figure III.13: Histogramme du résultat de Makespan. ....	74
Figure III.14: Histogramme de résultat de consommation d'énergie. ....	76
Figure III.15: Histogramme de résultat de coût. ....	78
Figure III.15: Histogramme de résultat de fiabilité. ....	80

### Liste des tableaux

Tableau I.1:les avantages et les incovénients de cloud computing .....	29
Tableau II.1 CompaTableau raison entre les différents algorithmes d'ordonnancement des tâches ....	53
Tableau III.1: les caractéristiques des Datacenters. ....	72
Tableau III.2:les paramètres des machines physiques. ....	72
Tableau III.3 : les paramètres des machines virtuelles.....	72
Tableau III.4:les paramètres des tâches.....	73
Tableau III.5:Résultat du Makespan. ....	74
Tableau III.6:Résultat de consommation d'énergie. ....	76
Tableau III.7:Résultat de coût.....	79
Tableau III.8:Résultat de fiabilité.....	79

# **Introduction générale**

# Introduction générale :

---

---

---

## *Introduction générale :*

---

Le Cloud Computing est une technologie en pleine croissance qui permet aux utilisateurs d'accéder à des ressources informatiques à distance, via internet. Il offre de nombreux avantages par rapport aux méthodes informatiques traditionnelles, il permet une plus grande flexibilité et une évolutivité facile, ce que signifie que les entreprises peuvent facilement ajouter ou supprimer des ressources en fonction de leurs besoins, ces ressources peuvent inclure des serveurs, des bases de données, des logiciels et des applications.

Le Cloud Computing peut également être moins coûteux que les méthodes informatiques traditionnelles car les utilisateurs (clients) ne paient que pour les ressources qu'ils utilisent, enfin il permet un accès facile aux données et aux applications à partir de n'importe où dans le monde, ce qui améliore la collaboration et la productivité.

La théorie de l'ordonnancement des tâches dans le Cloud Computing est un domaine de recherche important qui vise à optimiser la façon dont les tâches sont planifiées et exécutées dans l'environnement de Cloud Computing, en étudiant l'ordonnancement des tâches, les ingénieurs peuvent identifier les algorithmes et les stratégies les plus efficaces pour différents systèmes et environnements.

L'intérêt d'étudier le Cloud Computing et l'ordonnancement des tâches est pour permettre aux ingénieurs de concevoir des systèmes informatiques plus efficaces, plus rentables et plus évolutifs.

Dans le contexte du Cloud Computing le problème de l'ordonnancement des tâches est de trouver le meilleur plan d'exécution pour les tâches à exécuter sur les ressources du Cloud Computing, des algorithmes d'ordonnancement des tâches sont utilisés pour résoudre ce problème.

L'objectif de notre projet de fin d'étude est de proposer une étude comparative entre quelques algorithmes d'ordonnancement des tâches en fonction de plusieurs performances à savoir le Makespan, le coût, la fiabilité et la consommation d'énergie.

Le rapport est structuré de la manière suivante :

## **Introduction générale :**

---

---

Au premier chapitre, nous allons aborder les notions de base du Cloud Computing et son concept générale.

Le deuxième chapitre sera dédié à l'ordonnancement des tâches ainsi qu'aux techniques et algorithmes d'ordonnancement utilisés dans le Cloud Computing.

Dans le chapitre 3, nous allons donner une présentation détaillée des algorithmes d'ordonnancement utilisés dans notre projet ainsi que les critères de performances, nous allons ensuite présenter les résultats obtenus.

Nous achevons ce rapport en présentant une conclusion générale dans laquelle nous précisons les principaux résultats de notre étude, ainsi que les limites de Cloud Computing et l'ordonnancement des tâches en donnant une nouvelle perspective.



**CHAPITRE I :**  
**Généralités sur le**  
**Cloud Computing**

## **Introduction :**

Dans le but de soutenir un nombre important d'utilisateurs et leur permettre d'accéder à leurs données et applications à partir de n'importe où dans le monde avec une grande accessibilité, le Cloud Computing a été inventé. Cette invention est l'une des terminologies techniques les plus utilisées dans le monde.

Le Cloud Computing est arrivé au cours de ces dernières années comme une solution universelle utilisée par différents types d'utilisateurs. Le concept général du Cloud Computing est l'exploitation de la puissance de stockage ou de calcul des serveurs informatiques à distance, à travers un réseau (Internet). Ces serveurs sont alors loués sur demande, selon des critères techniques précis tels que la puissance et la bande passante.

Dans ce chapitre nous mettrons en évidence les notions générales du Cloud Computing à savoir la définition, les caractéristiques, les services fournis et les composants et on termine par quelques avantages et inconvénients.

### **I. Historique :**

Le Cloud Computing existant déjà sous différents noms comme l'externalisation ou l'hébergement sur serveur.

- Les origines et la naissance de Cloud Computing est commencé en 1961 où le concept de « l'informatique en tant que service » est apparu dans cette année, lorsque le professeur John McCarthy a suggéré que la technologie d'ordinateur en temps partagé « time-sharing » pourrait conduire à un avenir où la puissance de calcul et même les applications spécifiques pourraient être vendues comme un service public.
- Dans les années 1980 avant l'émergence du Cloud Computing, il y avait client/serveur qui est essentiellement un stockage centralisé dans lequel toutes les applications, les données sont hébergées côté serveur.
- La première personne à employer l'expression de Cloud Computing fut le professeur Ramnath Chellappa de l'université du Texas à Austin en 1997.
- En 1999, la première fourniture des applications aux utilisateurs à l'aide d'un simple site web. Les applications ont été livrées aux entreprises via internet, et ainsi le rêve de l'informatique vendue utilitaire s'est réalisé. [1]

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

- Dans les années 2000, l'émergence de solutions de type Cloud Computing (CC) engendre une profonde transformation de l'industrie informatique et des pratiques organisationnelles, comme individuelles. [21]
- En 2002, Amazon a lancé Amazon Web Services, fournissant des services tels que le stockage, le calcul et même l'intelligence humaine, et en 2009, Google Apps a également commencé à fournir des applications d'entreprise Computing. [1]

### **II. Définitions :**

Le Cloud Computing est un modèle de prestation de services informatiques qui permet aux utilisateurs d'accéder à des ressources informatique à distance via internet, ces ressources peuvent inclure des serveurs, des application, des plateformes de développement et des services de stockage de donnée, le Cloud Computing est une informatique distribuée qui partage les données d'une manière transparente entre les utilisateurs dans un modèle de paiement à l'utilisation [14].

Le concept générale du Cloud Computing est super simple : on clique, on paye et on obtient quasiment instantanément.

Mais selon les recherches il existe plusieurs définitions sur le Cloud Computing, il peut être un concept déroutant et abstrait. Pour diminuer cette confusion, la national institue of standard and technologie(NIST) a proposé une définition de Cloud Computing comme suit « Un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de services. Ce modèle de Cloud est composé de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement. » [4]

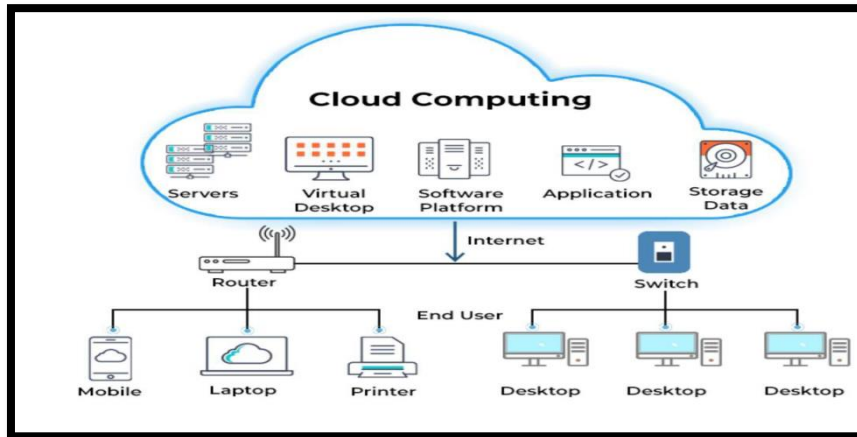


Figure I.1:Le Cloud Computing [5]

### III. Les caractéristiques du Cloud Computing :

Pour mieux comprendre les concepts de base de Cloud Computing nous avons pris de définition de la national institute of standard and technologie(NIST) les cinq caractéristiques essentielles:

#### III.1 Libre-service à la demande :

Signifier que l'utilisateur peut faire l'acquisition de service par lui même sans l'interaction humain ou passer par un service informatique. [4]

#### III.2 Large accès au réseau :

Le service Cloud doit être accessible ou bien disponible sur le réseau (Internet) et depuis n'importe quel emplacement en appliquant des mécanismes et des protocoles standards [5]

#### III.3 Mise en commun des ressources :

Les ressources matérielles du Cloud sont partagées entre les différents utilisateurs de service à l'aide d'une infrastructure homogène, donc un même client va utiliser les mêmes accès réseau, les mêmes systèmes de stockage avec d'autres clients. Le tout étant rendu transparent grâce à des technologies de virtualisation [5].

# CHAPITRE I : Généralités sur le Cloud Computing

---

## III.4 Élasticité rapide:

L'élasticité désigne la flexibilité du service Cloud pour évoluer automatiquement à la hausse ou à la baisse pour répondre aux besoins de l'utilisateur. [4]

Le Cloud fournit rapidement des ressources dans n'importe quelle quantité à n'importe quel moment grâce à l'illusion de ressources informatiques.

Illimitées et disponibles à la demande.

## III.5 Service mesuré :

Il est possible de surveiller et de contrôler toutes les ressources allouées pour mesurer leur consommation par type de service.

Ce modèle assure aux utilisateurs une plus grande transparence et un meilleur contrôle sur leurs coûts liés au nuage. [4]

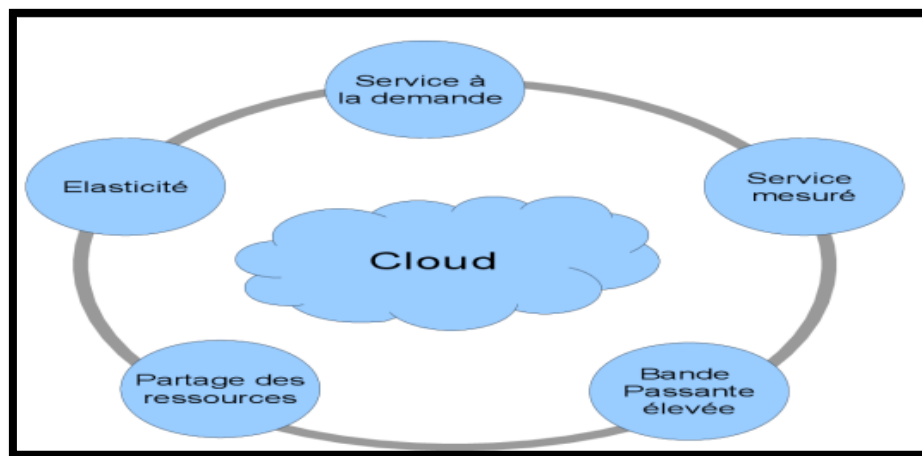


Figure I.2: Les caractéristiques de Cloud Computing [14].

## IV. Les éléments nécessaires du Cloud Computing :

### IV.1 La virtualisation:

Le nombre de serveurs dans le monde a augmenté d'une manière considérable cependant ces serveurs sont mal exploités ils ont souvent une charge moyenne de 20% et ceci engendre bien sur un coût d'exploitation très élevé ce gaspillage d'énergie a suscité de recours à des solutions notamment la virtualisation.

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

On va voir dans cette partie que cette dernière améliore considérablement l'efficacité et la disponibilité des ressources des applications.

### **IV.1.1 Définition :**

La virtualisation est l'ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation et/ou plusieurs applications, séparément les uns des autres comme s'ils fonctionnaient sur des machines physiques distinctes cela signifie que nous allons avoir une machine physique appelée les machines hôte sur laquelle seront exécutées les machines virtuelles.[6]

La virtualisation de serveurs extrait le logiciel serveur de la couche matérielle selon un modèle invité (système d'exploitation virtualisé) /hôte (système d'exploitation installé sur le matériel), permettant ainsi l'exécution de plusieurs serveurs virtuels sur une même machine physique.

Pour profiter de la virtualisation il suffit d'ajouter dans la machine un logiciel de virtualisation permettant d'ajouter une couche de virtualisation appelée le hyperviseur.

### **IV.1.2 Le principe de la virtualisation :**

Le principe de la virtualisation repose sur trois pivots majeurs que sont le système hôte, l'hyperviseur et le système invité. La combinaison de l'ensemble de ces ingrédients permet de créer une virtualisation.

Un système d'exploitation: qui est appelé également système hôte, est installé. Il représente l'OS principal pour accueillir les autres systèmes d'exploitation.

Un logiciel de virtualisation dénommé hyperviseur est alors installé sur le système hôte son rôle :

- Pouvoir créer des environnements sur lesquels d'autres systèmes d'exploitation seront hébergés.[25]
- Assure le contrôle du processeur et des ressources de la machine hôte.[25]
- Alloue à chaque VM (machine virtuelle) les ressources dont elle a besoin.[25]
- S'assure que ces VM n'interfèrent pas l'une avec l'autre.[25]

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

Ces derniers sont appelés systèmes invités. Chaque environnement, appelé machine virtuelle, fonctionne de manière indépendante.

### **IV.1.2.1 L'hyperviseur :**

Pour pouvoir utiliser un système d'exploitation vous avez besoin de l'installer sur une machine que ce soit un serveur où un simple desktop, certes vous pouvez installer plusieurs système d'exploitation sur une seule machine mais vous ne pouvez démarrer qu'un seul à la fois aussi si vous voulez faire fonctionner plusieurs système d'exploitation à la fois vous aurez donc besoin de plusieurs machines physique, ce qui trop cher donc la solution c'est l'hyperviseur.

Il s'agit d'un programme qui permet de faire fonctionner plusieurs systèmes d'exploitation sur une seule machine. Il y a deux types d'hyperviseur.

Dans cette partie, ce que nous vous proposons, c'est de faire expliquer ce qu'est un hyperviseur de type 1 et de type 2 et de vous donner quelques exemples pour maîtriser cette notion de base en terme de virtualisation.

- **Hyperviseur de type 1 :**

Un hyperviseur de type 1 est un système installé directement sur la machine que ce soit un serveur où un ordinateur, il faut considérer que c'est le système d'exploitation principale c'est ta dire il agit comme un système d'exploitation, est au plus près de la couche matérielle.[6]

En fin de compte, quand on parle d'hyperviseur de type 1, on parle d'hyperviseur natif c'est un hyperviseur qui sera beaucoup plus performant puisqu'il est directement en contact avec le matériel. Ce qui veut dire que quand vous démarrez le vôtre serveur, et bien vous démarrez directement sur la couche logicielle de l'hyperviseur qui lui ensuite va démarrer les machines virtuelles, donc c'est bien l'hyperviseur qui gère les machines virtuelles.[6]

Le système d'exploitation a besoin des ressources matérielle pour fonctionner donc l'hyperviseur partagent les ressources matérielles de la machine physique entre les machines virtuelles et il leur donne les ressources matérielles dont ils ont besoin comme si chaque système d'exploitation à ses propres ressources matérielles.

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

Le gros avantage de ce type d'hyperviseur, c'est qu'il permet d'allouer la quasi-totalité de ressources disponible aux machines virtuelles.

### **Exemples d'hyperviseur de type 1 :**

Concrètement, en entreprise on déploie des hyperviseurs de type 1 que l'on va venir installer sur un ou plusieurs serveurs, que l'on peut ensuite montrer éventuellement en cluster ce qui va permettre d'utiliser des fonctionnalités avancées notamment en terme d'équilibrage de charge.

Nous allons vous donner quelques exemples d'hyperviseur de type 1 :

-VMware ESXI et la suite VSphere.[6]

-Promo x VE qui est une solution open source basée sur LINUX KVM.[6]

-CitrixXen Server.[6]

-Microsoft avec sa solution Hyper-V.[6]

- **Hyperviseur de type 2 :**

L'hyperviseur de type 2, que l'on appelle aussi hyperviseur hébergé, est un logiciel qui s'installe et s'exécute sur un système d'exploitation déjà en place. Et puis s'installe les machines virtuelles, là où c'est différent c'est qu'il y a la couche système d'exploitation de la machine entre l'hyperviseur et le matériel donc forcément cela va impacter les performances puisqu'il faut interagir avec l'os puis après le matériel.[6]

L'os invité s'exécute donc en troisième niveau au-dessus de matériel.

Il faut également savoir qu'un hyperviseur de type 2 aura moins de ressources disponibles pour les machines virtuelles parce que il y aura des ressources consommées par le système hôte en lui-même.[25]

L'avantage principal de ce type d'hyperviseur est la possibilité d'installer et d'exécuter autant d'hyperviseur que l'on désire sur notre système hôte.[6]

### **Exemples d'hyperviseur de type 2 :**



## **CHAPITRE I : Généralités sur le Cloud Computing**

---

Pour finir nous allons vous montrer quelques exemples d'hyperviseur de type 2 on a :

-Oracle virtuel Box qui est une solution très intéressante et gratuite.[6]

-VM wareworkstation (payant) qui coute à peu près 200 euro.[6]

-VM ware workstation player (gratuite). [6]

-VM ware Fusion (Mac OS).[6]

### ***IV.1.2.2 La machine virtuelle :***

La machine virtuelle est un conteneur de logiciel isolé permettant d'exécuter simultanément plusieurs systèmes d'exploitation sur un seul ordinateur.

Les machines virtuelles sont aussi des logiciels d'émulation capables de simuler un ordinateur et également capable d'effectuer des tâches telles que l'exécution d'application et programme.  
[6]

Grâce à la virtualisation chaque application et système d'exploitation se trouve dans une machine virtuelle il est donc possible d'exécuter plusieurs machines virtuelles simultanément sur un même ordinateur physique.

Les VM réalisent son propre système d'exploitation et dispose d'équipements semblables à une machine physique tels que mémoire RAM, CPU disque dur et carte réseau.

### **Pourquoi avoir recourt à une machine virtuelle ?**

Pour obtenir plusieurs systèmes d'exploitation sur votre serveur il était nécessaire d'avoir un autre parce qu'un serveur ne pouvait supporter qu'un seul système d'exploitation. Maintenant certaines entreprises pouvaient avoir jusqu'à plusieurs centaines de serveurs physiques.[17]

Géniale avec la virtualisation et la création de machine virtuelle ce temps-là est révolu, l'optimisation des équipements et la maintenance devient plus simple.[17]

### **Les caractéristiques de la machine virtuelle :**

- **Sécurité :**

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

Chaque machine virtuelle est isolée des autres y compris du système hôte, cela permet de limiter les risques de propagation en cas d'intrusion de type malware.

Isoler certaines parties du système afin de limiter les dégâts.[17]

- **Partitionnements :**

Exécuter plusieurs systèmes d'exploitation sur une machine physique.

Partager les ressources entre les machines virtuelles.

- **Sauvegarde :**

Enregistrement de l'état complet des différentes machines virtuelles dans des fichiers.

Durant le fonctionnement de votre VM, il est possible d'obtenir une sauvegarde grâce à un instantané de la VM et ses données. En cas d'incident, cet instantané permet de remettre la VM dans son état précédent. Les sauvegardes sont facilitées, comme la reprise après incident.

- **Flexibilité des ressources :**

Les VM installées sur un hôte physique sont capable de consommer des quantités inégale de ressources, et il est facile de déplacer, copier les VM entre des serveurs hôtes pour optimiser l'exploitation des ressources matérielles.[17]

- **Indépendance vis-à-vis du matérielle :**

Réserver n'importe quelle machine virtuelle vers n'importe quel serveur physique.[6]

Les avantages de la machine virtuelle :

- **Avantage économique :** grâce à une seule infrastructure regroupant l'exécution de plusieurs environnements virtuels, vous réduisez son encombrement ainsi que les coûts d'électricité et de maintenance.[32]
- **Avantage sur la rapidité :** mettre en action une machine virtuelle est d'une grande rapidité et simplicité. Cela permet un gain de temps dans le processus d'exécution des scénarios de développement et de test.[32]

## CHAPITRE I : Généralités sur le Cloud Computing

---

---

- **Avantage sur la sociabilité** : les machines virtuelles vous permettent de faire en sorte que vos applications soient à l'échelle plus facilement grâce à une charge de travail répartie entre plusieurs machines virtuelles.[32]

### IV.1.3 Les techniques de la virtualisation :

#### IV.1.3.1 Le cloisonnement (isolation) :

Le cloisonnement vise à isoler chaque processus dans un conteneur dont il est théoriquement impossible de sortir

L'isolateur est un logiciel permettant d'isoler l'exécution des applications dans ce qui sont appelés des contextes ou bien zones d'exécution différents.[15]

Les programmes et les applications de chaque contexte ne sont pas capables de communiquer qu'avec les processus associés à leur propre contexte.

Un processus isolé de la sorte ne saura pas quels autres processus l'exécutant sur le même système et n'aura qu'une vision limitée de son environnement.

Dans cette technologie l'espace noyau est le même il est partagé entre les différents contextes, il est possible de partitionner un seul serveur en plusieurs contextes.

Le but principal de l'isolation est d'améliorer la sécurité du système d'exploitation et des applications, cette technique est performante mais les environnements virtualisés ne sont pas complètement isolés

Une des solutions les plus avancées pour isolation est Open VZ. [15]

#### IV.1.3.2 Virtualisation complètes :

La virtualisation complétée est l'émulation du système complet permettant d'exécuter le système d'exploitation invité directement sur la machine virtuelle.

Elle dite complète lorsque le système invité n'a pas conscience d'être virtualisé c'est à dire le système ne se rend pas compte qu'il est en cours d'exécution dans un environnement virtualité.[22]

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

Ce type de virtualisation est plus simple à mettre en place et est le plus pratique.

Le principe est que l'hyperviseur émule un environnement matériel sur chaque machine virtuelle c'est à dire l'hyperviseur crée un environnement virtuel complet simulant littéralement un nouvel ordinateur complet, avec du "faux matériel.

L'hyperviseur gère l'ensemble des requêtes des machines virtuelles ce qui permet aux machines virtuelles de fonctionner sans aucune modification de leur noyau.

Ce type de virtualisation ne permet de virtualiser que des systèmes d'exploitation prévus pour la même architecture matérielle que le processeur physique de l'ordinateur hôte.

### **Quelques hyperviseurs de virtualisation complète :**

-Virtual Box

-GNOME Machines

-Logiciels de virtualisation de VMware : VMware Player, VMware Workstation

-Parallels Desktop 4 for Windows & Linux

-KVM

### **IV.1.3.3 La paravirtualisation :**

La paravirtualisation est une technique de virtualisation qui fournit une interface vers des machines virtuelles similaires à leur matériel sous-jacent.

La para virtualisation permet à plusieurs systèmes d'exploitation différents de s'exécuter sur un même ensemble de matériel en utilisant efficacement des ressources telles que des processeurs et de la mémoire.

La paravirtualisation est la virtualisation complète sont assez proches. Le cœur du para virtualisation est un hyperviseur qui fournit et gère une interface qui permet à plusieurs systèmes hôte d'accéder de manière concurrente aux ressources.[6]

L'hyperviseur capture les appels système de l'invité et les transmet au matériel.[6]

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

Avec la para virtualisation le système invité doit avoir conscience qu'il tourne dans un environnement virtuel ce qui implique d'employer un noyau modifié.

Avec cette technique il doit avoir des modifications au niveau de système d'exploitation hôte et au niveau du système d'exploitation invité. [6]

La paravirtualisation présente de nombreux avantages de performances significatifs et son efficacité offre une meilleure mise à l'échelle.

### **Quelques hyperviseurs de paravirtualisation :**

Xen : de Citrix

KVM : projet hyperviseur intégré dans le noyau linux

ESX/ESXi : hyperviseur leader de VMware.

### **IV.1.3.4 Virtualisation assisté par le matériel :**

Virtualisation assistée par matériel est une technologie qui simule l'environnement matériel d'un ordinateur implique l'intégration de logiciels de machines virtuelles dans le composant matériel d'un serveur.[6]

La virtualisation assistée par le matériel est en réalité une extension du principe de virtualisation complète.[6]

Dans ce type de virtualisation l'hyperviseur fonctionne en mode Racine (RootMode) avec le niveau de contrôle le plus élevé.[6]

La principale modification qui est apportée est l'implémentation des instructions dans les processeurs : AMD-V chez AMD et Intel. [6]

Ces instructions permettent aux VMs de gérer leurs propres interruptions et donc changement de contexte.[6]

## CHAPITRE I : Généralités sur le Cloud Computing

---

---

### IV.1.4 Les types de virtualisation :

#### IV.1.4.1 Virtualisation des serveurs :

La virtualisation des serveurs, ou « server virtualisation », est le processus qui consiste à diviser un serveur physique en plusieurs serveurs virtuels uniques et isolés au moyen d'une application logicielle.[32]

La virtualisation des serveurs est le premier type de virtualisation rencontré. Cette technique implique le regroupement de plusieurs serveurs virtuels en un seul **serveur physique**.

Chacune des machines virtuelles agit d'une manière indépendante et isolé et exécutant ses propre système d'exploitation et applications.

Les avantages de la virtualisation des serveurs :

- Amélioration des performances des applications. [32]
- Accélération du déploiement des charges de travail. [32]
- Optimiser les performances d'un serveur. [32]

#### IV.1.4.2 La virtualisation de stockage :

La virtualisation de stockage est le processus de regroupement du stockage physique à partir de plusieurs périphériques de stockage réseau afin qu'il ressemble à un seul périphérique de stockage.

Le processus consiste à extraire et à couvrir les fonctions internes d'un périphérique de stockage à partir de l'application hôte, des serveurs hôtes ou d'un réseau général afin de faciliter la gestion du stockage indépendante de l'application et du réseau.

Il existe trois principales approches de stockage :

- DAS (Direct Attached Storage)
- NAS (Network Attached Storage)
- SAN (Storage Area Network)

**Le DAS (Direct Attached Storage):** Cela désigne tout simplement le cas où un périphérique de stockage est relié directement à une machine. C'est le type d'accès le plus rapide, tant en

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

termes de débit que de latence, mais il implique que chaque machine dispose de son propre stockage.[30]

**Un NAS (Network Attached Storage) :** offre un emplacement centralisé pour stocker des données sur un réseau a pour fonction de sécuriser, sauvegarder, partager et faciliter l'accès à vos fichiers depuis plusieurs appareils multimédias reliés à un même réseau. [31]Le contenu du NAS est alors visible comme un simple dossier partagé en local ou à distance. Pour stocker ces données, le serveur Nas doit être équipé d'un support de stockage, le disque dur la plupart du temps.

**Le SAN (Storage Area Network) :** est un réseau physique en fibre optique, dont le but est de permettre la mise en relation de serveurs avec des baies de disques Communément, pour parler de SAN, on utilise les termes "**baie SAN**" et "**baie de stockage**". La baie SAN se présente comme un espace de stockage partagé entre plusieurs serveurs, simultanément. Elle est constituée de disques physiques et il peut s'agir de disques mécaniques ou de disques SSD,

Les principales différences entre DAS, NAS et SAN sont les coûts, l'évolutivité et la façon dont le stockage est partagé. Les trois systèmes utilisent également des mécanismes de stockage différents : le DAS utilise principalement le stockage sur disque dur avec des secteurs, le NAS utilise des fichiers partagés et le SAN utilise le stockage par blocs. [30]

Différentes technologies sont également utilisées pour la transmission des données. DAS utilise IDE/SCSI, NAS utilise TCP/IP et Ethernet, et SAN utilise Fibre Channel et IP.

On peut dire aussi que les méthodes de stockage SAN et NAS permettent de gérer le stockage de manière centralisée et de le partager entre plusieurs hôtes (serveurs). Toutefois, les NAS sont basés sur Ethernet, tandis que les SAN peuvent aussi bien utiliser Ethernet que Fibre Channel.[30]

### **IV.1.4.3 La virtualisation du réseau :**

La virtualisation de réseau combine plusieurs réseaux physiques en un réseau logiciel virtuel, ou encore diviser un réseau physique en plusieurs réseaux virtuels indépendants et distincts.

La virtualisation de réseau découple les services réseau du matériel sous-jacent et permet d'effectuer le provisionnement virtuel de tout un réseau.

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

La virtualisation de réseau permet également de procéder à l'automatisation de nombre de tâches administratives, ce qui réduit ainsi le nombre d'erreurs manuelles et le délai de provisionnement. Elle renforce la productivité et l'efficacité du réseau.

Les composants de base de la virtualisation du réseau sont :

- Carte réseau virtuelle(VNIC)
- Switch(VSwitch)

Exemple de virtualisation de réseau :

Les LAN virtuels (VLAN) sont un exemple de virtualisation de réseau. Un VLAN est une sous-section d'un réseau local (LAN) créée avec un logiciel qui combine les terminaux du réseau en un groupe, quel que soit leur emplacement physique. Les VLAN améliorent la vitesse et les performances des réseaux surchargés et simplifient les opérations de modification ou d'ajout effectuées sur le réseau.

### **IV.1.4.4 La virtualisation applications:**

La virtualisation d'application consiste à encapsuler dans un même package l'application et son environnement système de manière imperméable au système d'exploitation sur lequel l'application s'exécute.

La virtualisation d'applications consiste à publier des applications individuelles à partir d'un serveur, l'accès étant fourni soit via le protocole d'affichage à distance, soit via la distribution d'applications à la demande.

Dans ce type de virtualisation il est nécessaire d'ajouter une couche supplémentaire entre un programme donné et le système d'exploitation le but de cette couche est intercepté toutes les opérations d'accès ou de modification de fichiers.

### **IV.1.5 Les avantages de la virtualisation :**

Les entreprises qui possèdent la technologie de la virtualisation sont améliorées grâce aux avantages de cette technologie :

- Le nombre de machines est diminué réduisant considérablement les coûts de l'infrastructure physique.



## CHAPITRE I : Généralités sur le Cloud Computing

---

---

- Une sécurité optimale.
- De meilleures performances
- Chaque machine virtuelle est isolée des autres y compris du système hôte. Cela permet de limiter les risques de propagation en cas d'intrusion de type malware.
- Une flexibilité dans l'utilisation des différents systèmes d'exploitation. Il est possible de travailler sous plusieurs environnements tels que Windows ou encore Linux sur un même poste.

### IV.2 Data center :

Les musiques qu'on écoute, streaming, les vidéos qu'on regarde sur youtube les sites web qu'on visite et les documents qu'on stocke dans le Cloud. Tout cela est hébergé sur les serveurs qui sont eux-mêmes hébergés dans les Datacenter sur lequel une quarantaine de personnes travaillent quotidiennement pour gérer plus de 100000 serveurs Datacenter.

L'objectif d'un data center est d'héberger les sites web, les applications et les données pour les rendre accessibles aux utilisateurs et c'est dans ce bâtiments sécurisé que sont gérés les équipements informatique qui permettent de stocker, analyser et traiter les données.

Il faut savoir que pour assurer efficacement son rôle d'hébergement et de mise a disposition des données et des applications le Datacenter s'appuie sur différents technique qui garantissent la continuité d'activité et la tolérance aux pannes cela commence par :

- La gestion de l'énergie pour alimenter les infrastructures informatique.
- Les dispositions de refroidissement pour éviter aux serveurs toute surchauffe
- Les techniques de redondance en cas de panne d'un des équipements
- L'aspect sécurité qui est aussi un point essentiel des Datacenter.

Dans cette partie on va voir comment sont organise les équipements informatique a l'intérieur du Datacenter.

- salle serveur où les équipements informatiques sont empilés les uns sur les autres dans ce qu'on appelle des racks ou étagères en français et ces racks sont alignés en rangées. Sur ces racks en plus des serveurs on trouve aussi une multitude d'équipements informatique tels que :

## CHAPITRE I : Généralités sur le Cloud Computing

---

---

Des baies de stockage, des équipements réseaux comme des routeurs, des Switch, des load, des balancer, des proxys ainsi que des équipements de sécurité comme les firewalls.

- Serveur : c'est un peu un super ordinateur comme des ventilos, des barrettes de mémoire ram, des processeurs mais tout ça évidemment en beaucoup plus performant car c'est un équipement informatique qui traite énormément de données de manière automatique et simultanées et qui en plus tourne 24 h sur 24h.Évidemment pour cela il faut beaucoup d'énergie et la gestion de cette énergie est un enjeu majeur du Datacenter.
- l'onduleur : son rôle est de stabiliser le courant en effet pour fonctionner de façon optimale les serveurs ont besoin de courant stable c'est ta dire sans risque de micro coupures, de surtension ou de baisse de tension les serveurs et l'ensemble des équipements du data center sont de ce fait alimentés par le courant des onduleurs, le système d'onduleur est pris en charge par des batteries pendant une panne de courant de secteur.
- Le système de refroidissement : son rôle est de refroidir les équipements informatique pour éviter qu'il ne surchauffe et tombe en panne, ce refroidissement peut se faire par air l'eau ou par système de climatisation.

Microsoft, Amazon et Google contrôlent plus de la moitié des 600 plus grands data center mondiaux.

## V. Les services de Cloud Computing :

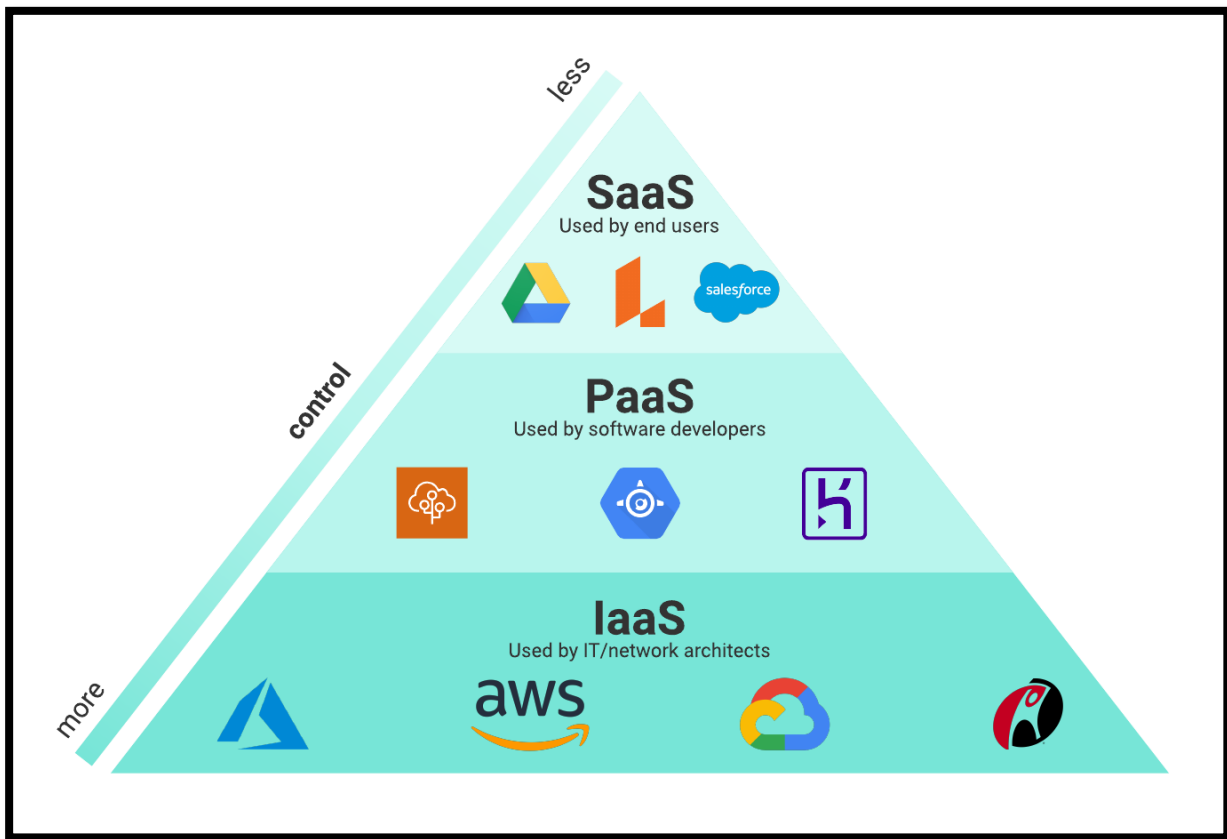


Figure I.3: les services de Cloud Computing. [28]

### V.1 IaaS (infrastructure as a service) :

C'est un système qui permet d'offrir des ressources informatiques au sein d'un environnement virtuel par le biais d'internet ou autrement dit une manière de mettre à disposition des ressources informatiques en ligne en les hébergeant sur un environnement virtuel comme un Cloud de cette manière les fournisseurs tiers hébergent les serveurs et les ressources d'autres sociétés telles que le stockage, des hardware, software et la bande passante...ect[28]

Le principal avantage qu'offre l'IaaS à scalabilité des plateformes proposent effectivement des solutions très adaptables qui s'ajustent au mieux aux besoins des clients.

Un autre avantage notable de l'IaaS est son accès pratique et rapide quelque soit l'endroit où se trouve le client dès lors qu'il bénéficie d'une connexion internet sécurisé il peut accéder au service parmi les autres à tous de l'infrastructure à ce service on trouve aussi la sécurité de

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

données, l'automatisation des tâches administratives sur demande un éventuel scaling dynamique ou encore possible virtualisation bureau. [28]

### **V.2 PaaS (plateforme as a service) :**

C'est un modèle de Cloud Computing dont lequel fournisseur tiers fournit des outils matériels et logiciels généralement ceux nécessaires au développement d'applications aux utilisateurs via internet.

Le PaaS libère les développeurs de l'installation de matériel et de logiciel internes ou de l'exécution d'une nouvelle application. Cependant, il ne remplace généralement pas l'ensemble de l'infrastructure informatique d'une entreprise. Au lieu de cela il intègre généralement divers composants d'infrastructure Cloud sous-jacents, tels que les systèmes d'exploitation, des serveurs, des bases de données, des logiciels, des réseaux des équipements et des services de stockage tous détenus, exploités, configurés et entretenus par le fournisseur de services et non pas par l'entreprise. Celle-ci fournit simplement les données et les applications. Le PaaS fonctionne généralement sur une base d'utilisateur, ce qui le rend abordable pour les projets mais certains fournisseurs facturent des frais mensuels fixes, les offres PaaS sont généralement utilisées pour le développement d'application mobiles et multiplateforme, ainsi que pour les outils DevOps.[28]

#### **Les avantages du PaaS :**

- Simplicité de la commodité pour les utilisateurs.
- Réduction la gestion de l'infrastructure.
- Automatisation ou élimination des tâches de maintenance.
- Suppression des complexités de l'équilibrage de charge, de la mise à l'échelle et de la distribution des services dépendants.
- Permission aux organisations d'adopter de nouvelles offres et technologies sans remanier les processus métiers.

#### **Mais il y'a aussi des risques :**

- la disponibilité des services est une préoccupation car les interruptions de service des fournisseurs peuvent gravement affecter les clients et la productivité.

# CHAPITRE I : Généralités sur le Cloud Computing

## V.3 SaaS (software as a service):

Se traduit en français logiciel en tant que service : c'est en fait une forme spécifique de modèle de distribution par le biais de laquelle un fournisseur héberge des applications et les propose à ses clients.

### Le principe avantage de SaaS :

- En premier lieu cela leur permet de ne plus avoir à installer ou lancer des applications sur leur propre ordinateur et cela représente donc une forte réduction des coûts puisque les sociétés n'ont plus besoin d'acquérir certains matériels de maintenir leurs licences....etc.

La figure suivante désigne les principaux composants des services de Cloud Computing et leurs niveaux de responsabilités qui appartiennent au fournisseur et à l'entreprise.

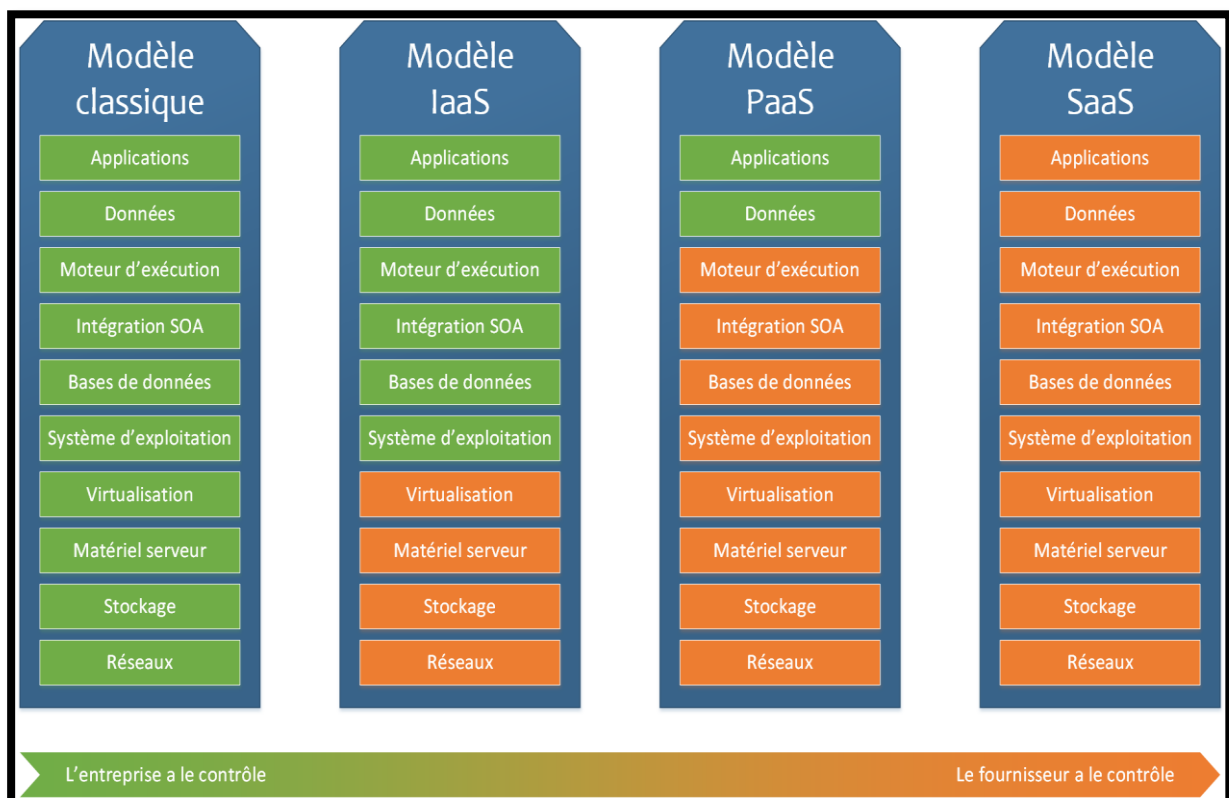


Figure I.4: modèle classique et modèle Cloud [25].

### VI. Les modèles de déploiement dans le Cloud Computing:

#### VI.1 Cloud privé :

C'est une infrastructure ou bien un ensemble de services et de ressources entièrement dédiée à une entreprise unique, pouvant être gérée en interne par l'entreprise elle-même dans ce cas on l'appelle <<le Cloud privé interne >> où gérée par un tiers loué par l'entreprise et dans celui-ci on l'appelle <<le Cloud privé externe>>, et il accessible par les réseaux sécurisés du type VPN (Virtual Private Network).

##### VI.1.1 Les caractéristiques de Cloud privé :

- L'accès est réservé à un client.
- L'architecture est conçue pour répondre aux besoins et aux objectifs d'une seule entreprise.
- Capacité d'hébergement des données sur les serveurs physiques de l'entreprise ou dans un Datacenter externe.
- Accessibilité via internet, grâce à une connexion cryptée et sécurisée.
- Accessibilité en local ou a distance depuis ordinateurs, Smartphones et tablettes.

##### VI.1.2 Les avantages de Cloud privé:

- Les tarifs sont fixes, quelle que soit la qualité de données que l'on transfère.
- Protection et confidentialité.
- Un contrôle complet des données et de l'infrastructure.
- Réponse exacte aux besoins de l'entreprise.
- Utilisation plus efficace des ressources informatique de l'entreprise.

##### VI.1.3 Les inconvénients de Cloud privé :

- Frais généraux de gestion : le département informatique doit prendre la responsabilité et gérer le Cloud privé.
- Aucune réduction des coûts : l'exploitation d'un Cloud privé entraîne des coûts personnels, d'administration et de maintenance comparable à ceux des centres de données traditionnels.

**Exemple:** Eucalyptus, OpenNebula, OpenStack.

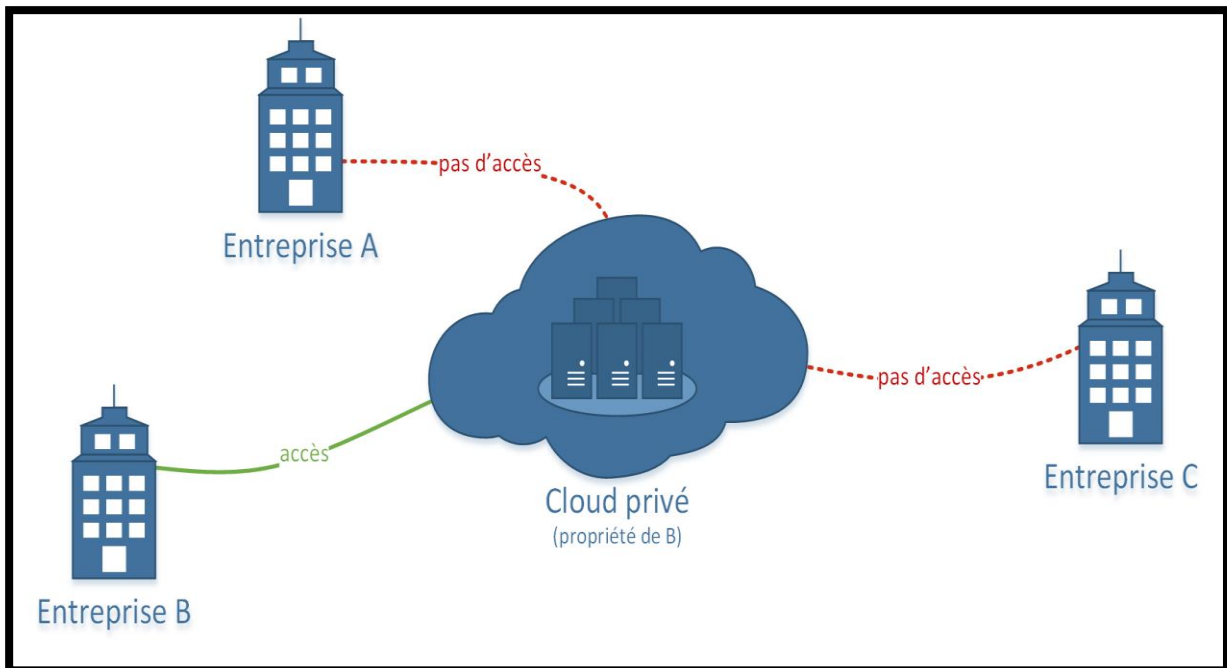


Figure I.5: mode de déploiement de Cloud privé [25].

### VI.2 Cloud public :

Les ressources informatiques de l'entreprise sont stockées sur un serveur mutualisé, autrement dit partagé entre plusieurs clients, et accessibles par internet. Ces serveurs sont partitionnés de manière à interdire les fuites de données.

#### VI.2.1 Les caractéristiques de Cloud public :

- Coûts adaptés aux besoins : chaque client reçoit son propre accès pour son compte Cloud
- Possibilité de réserver des services individuels en fonction des besoins.
- Interface utilisateur basée sur le Web.
- Les logiciels souscrits à partir de Cloud public sont accessibles via cette interface utilisateur.
- **Évolutivité** : si un trafic élevé exige de manière excessive les performances d'une application Web, on pourra éviter les interruptions dues à une surcharge en augmentant les ressources.
- **Efficacité** : le fournisseur Cloud gère les processus dans les plus brefs délais.

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

- **Fiabilité :** les fournisseurs s'occupent de la maintenance de l'infrastructure informatique et remplacent les appareils défectueux. Le matériel redondant prévient les pannes.

### **VI.2.2 Les avantages de Cloud public:**

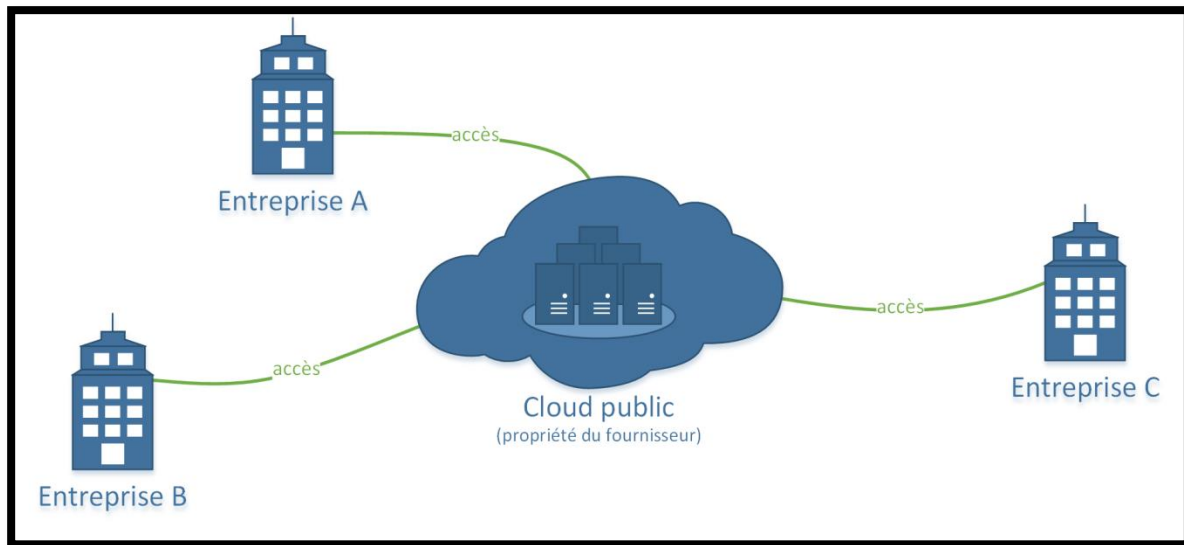
- L'enregistrement et la mise à l'échelle sont efficaces, le paiement est basé sur le principe de service et selon le service, également sur une base d'abonnement.
- Les investissements dans du matériel informatique appartenant à l'entreprise sont très rentables pour les Lean clients.
- Des experts surveillent les fermes de serveurs 24 heures sur 24, s'occupant de l'entretien et faisant le plein de matériel.

### **VI.2.3 Les inconvénients de Cloud public:**

- Etant donné que la connexion se fait via l'internet public, le Cloud public est généralement considéré comme plus vulnérable aux fuites de sécurité.
- D'autres clients peuvent être en mesure d'affecter l'accessibilité ou les performances de vos services Cloud en utilisant la même machine physique.
- Certains fournisseurs restreignent la liberté de choix.

**Example:** Microsoft Azure, Amazon Web Service (AWS).





**Figure I.6:** mode de déploiement de Cloud public [25]

### VI.3 Cloud hybride :

Un Cloud hybride associe des environnements de Cloud public et de Cloud privé en leur permettant de se communiquer les données et application.

#### VI.3.1 Les caractéristiques de Cloud hybride :

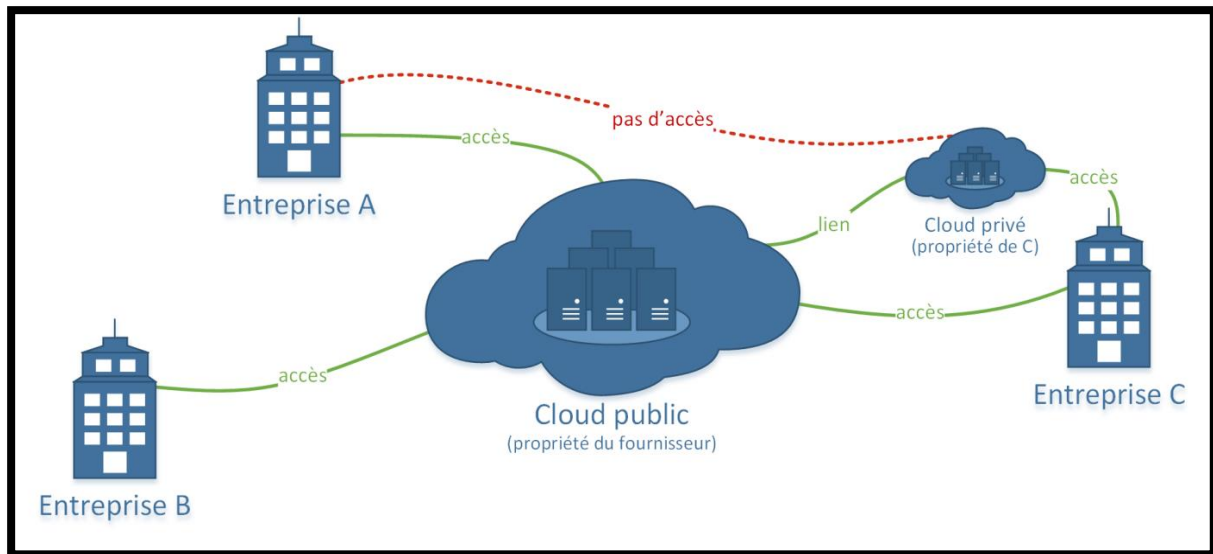
- Caractéristique de Cloud privé + Cloud public.
- Une partie des données et des applications se trouve dans les locaux de l'entreprise et une autre sur les serveurs d'un prestataire dédié.

#### VI.3.2 Les avantages de Cloud hybride :

- Flexible et évolutif.
- Economie de ressources.
- Peu coûteux par comparaison.
- Sécurité de données sensibles et des applications critiques.

#### VI.3.3 Les inconvénients de Cloud hybride :

- Effort supplémentaire.
- La sécurité peut uniquement être garantie par des règles claires.



**Figure I.7:** mode de déploiement de Cloud hybride [25]

### VI.4 Cloud communautaire:

Un Cloud communautaire désigne hybride d'un Cloud privé, c'est une plateforme d'hébergement multi-locataire.

Les Clouds communautaires sont relativement plus récents que les Clouds privés, publics ou hybrides. Le travail en ligne s'étant répandu pendant la pandémie, les Clouds communautaires ont gagné en popularité.

#### VI.4.1 Les avantages du Cloud communautaire :

- L'infrastructure est partagée entre plusieurs entreprises.
- La réalisation des économies.
- Les données restent en partie segmenté sauf dans les zones où l'accès partagé est convenu et configuré.

#### VI.4.2 Les inconvénients du Cloud communautaire :

- Le mode de fonctionnement de Cloud communautaire pose des problèmes de sécurité.
- d'isolation des données malgré le cloisonnement.
- Les risques d'attaques existent surtout lorsqu'on multiplie le nombre d'utilisateur.

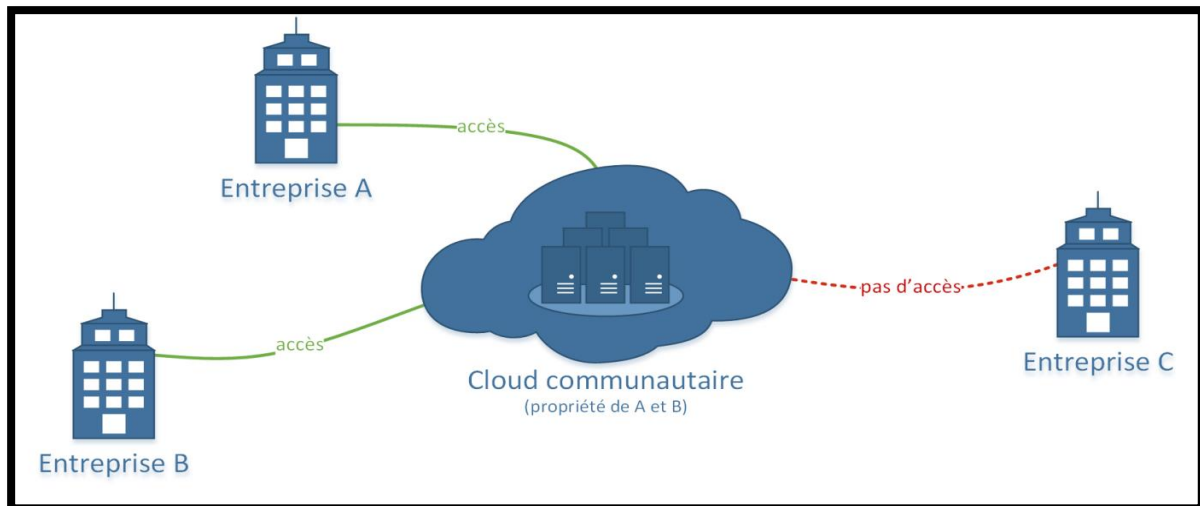


Figure I.8: mode de déploiement de Cloud communautaire [25]

## VII. Les avantages et les inconvénients de Cloud Computing :

Les avantages	Les inconvénients
<ul style="list-style-type: none"> <li>-Gain de temps sur la maintenance.</li> <li>-Réduction des coûts. [27]</li> <li>-Une plus grande flexibilité. [27]</li> <li>-Une meilleure collaboration. [27]</li> <li>-Facturation a la consommation. [1]</li> <li>-Service d'une grande disponibilité. [1]</li> </ul>	<ul style="list-style-type: none"> <li>-Le prix : en fonction des besoins, un modèle Cloud peut revenir plus cher à la longue qu'une infrastructure on permise. [13]</li> <li>-Le Cloud n'est pas à l'abri des hackers.</li> <li>-Les fonctionnalités peuvent être limitées.</li> <li>-les données stockées sur le Cloud ne vous n'appartiennent plus totalement à l'utilisateur. [13]</li> </ul>

Tableau.VII-1.1:les avantages et les inconvénients de cloud computing

## **CHAPITRE I : Généralités sur le Cloud Computing**

---

---

### **Conclusion :**

Dans ce chapitre, nous avons présenté des notions fondamentales, Les modèles de déploiement, les services et les caractéristiques du Cloud Computing ainsi que ces avantages et ces inconvénients.

Dans le chapitre suivant, nous allons entamer un sujet très intéressant qui est l'ordonnancement dans le Cloud Computing, dans lequel nous allons étudier quelques algorithmes d'ordonnancement des tâches.

**CHAPITRE II:**  
**LES ALGORITHMES**  
**ORDONNANCEMENT DES TÂCHES**  
**DANS UN ENVIRONNEMENT DU**  
**CLOUD COMPUTING**

### **Introduction :**

Le Cloud Computing est la fourniture de service informatique qui utilise de nombreuse technologie ayant besoin de plusieurs élévations et implémentation, de normes précises pour éviter les risques et améliorer le fonctionnement du Cloud.

Un des vrais challenges pour les gestionnaires dans ce type de technologie est l'ordonnancement des tâches qui est souvent considéré comme une solution pour gérer les ressources de calculs et assurer leurs performances dans les systèmes distribués à grande échelle tel que l'environnement Cloud Computing.

L'ordonnancement consiste à ordonner un ensemble d'opérations en satisfaisant un ensemble des contraintes de performance.

Il existe plusieurs types d'algorithmes d'ordonnancement. Dans ce qui suivra on verra comment ces algorithmes fonctionne et comment sont-ils utilisés des tâches et de comparer ces algorithmes en fonction de plusieurs critères

### **I. Ordonnancement :**

#### **I.1 Définition :**

L'ordonnancement c'est le fait de décider dans quel ordre on réalisera une suite de tâches ou de prioriser les différentes opérations. C'est un processus important pour garantir que les ressources soient utilisées de manière efficace et que les tâches sont effectuées dans les délais impartis. Les fournisseurs de service de Cloud utilisent des techniques d'ordonnancement des tâches pour gérer les ressources de manière efficace, en répartissant les tâches entre les différents serveurs et en utilisant des algorithmes d'ordonnancement des tâches.

L'ordonnancement des tâches doit prendre en compte les critères de performance pour garantir que les tâches sont effectuées de manière efficace, les contraintes peuvent inclure le délai, la disponibilité de ressource, le coût, la fiabilité, consommation d'énergie, le temps d'exécution et le temps de latence.

Un problème d'ordonnancement est caractérisé par un ensemble de tâches et un ensemble de ressources.

### ➤ **Tâche :**

C'est une entité élémentaire de travail localisée dans le temps par une date de début et de fin, dont la réalisation est caractérisée par une durée.

L'analyse d'une tâche consiste à comprendre les objectifs des utilisateurs et de comprendre comment ils passent des objectifs aux tâches puis aux actions. [11]

Il existe deux types de tâches :

- Tâches dépendantes : partagent d'autres ressources ou sont reliées par des contraintes de précedence et ont un type d'ordre à suivre. [11]
- Tâche indépendante : les informations sont connues à l'avance comme le temps d'exécution sur les différentes ressources, donc dans ce cas les tâches s'exécutent en parallèle, indépendamment les unes des autres et ne nécessitent aucune communication autres elle. [11]

### ➤ **Ressource :**

La ressource est un moyen technique disponible en quantité limitée, il est utilisé pour la réalisation d'une tâche [11].

Les ressources peuvent être des machines virtuelles, des machines physiques, des serveurs, des CPU et des RAM.

## II. **L'ordonnancement dans un environnement de Cloud Computing :**

L'ordonnancement est un enjeu essentiel qui influe sur les fonctionnements du Cloud Computing. Il est réparti sur deux niveaux, niveau utilisateur et niveau système.

Au niveau de l'utilisateur, l'organisation traite les problèmes des prestations de services entre les fournisseurs et les clients.

Au niveau système c'est la gestion des ressources dans les centres de données (Datacenters) le centre de données se compose de plusieurs machines physiques, des millions de tâches des utilisateurs sont reçues, l'attribution de ces tâches aux machines physiques se fait au niveau

du centres de données. Cette affectation d'ordonnancement joue un rôle significatif sur les performances du centre de données.

L'ordonnancement des tâches dans le Cloud Computing consiste à placer et ordonner de manière correcte les tâches sur les machines. L'objectif général de l'ordonnancement des tâches est la minimisation de temps de complétion (Makespan), maximiser l'utilisation des ressources, respecter les délais et de réduire les coûts.

### **II.1 Le rôle d'ordonnanceur :**

Un ordonnanceur (Scheduler) devra trouver la machine la plus appropriée pour traiter les tâches qui lui sont soumises. Les ordonnanceurs peuvent aller du plus simple (allocation de type round-robin) au plus compliqué (ordonnancement à base de priorités). [2]

Les ordonnanceurs déterminent le taux de charge de chaque ressource afin de bien ordonner les prochaines tâches. Les ordonnanceurs supervisent le déroulement d'une tâche jusqu'à son achèvement, et si la tâche est interrompue de manière inattendue l'ordonnanceur la relance pour assurer qu'elle est terminée avec succès, et si la tâche se trouve dans une boucle infinie d'exécution, l'ordonnanceur la termine prématurément pour éviter que les ressources ne soient gaspillées. [2]

### **II.2 Les techniques d'ordonnancement :**

Il existe plusieurs techniques pour ordonner les tâches d'une application en prenant en considération le coût des ressources utilisées.



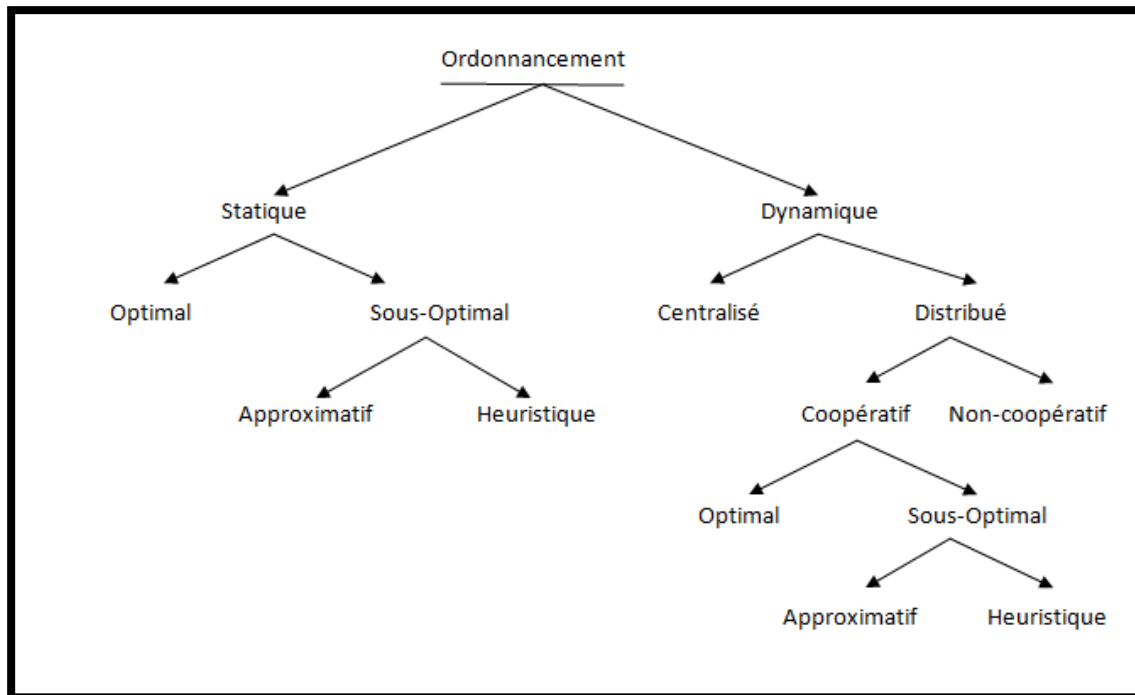


Figure II.1:les techniques d'ordonnancement [16]

### II.2.1 Ordonnancement statique : (hors ligne) :

Avant le lancement de l'application l'ordonnancement statique met en place une séquence fixe d'exécution des tâches à partir de tous les paramètres de celle-ci, les caractéristiques des tâches sont connues, et sa date d'exécution d'une tâche est fixe et toujours la même. [14]

### II.2.2 Ordonnancement dynamique :

L'ordonnanceur établit une séquence d'exécution des tâches dynamiques au cours de la vie de l'application. [13]

Le nombre de tâche peut changer au fil du temps. [13]

Les machines virtuelles peuvent être migrées sur les nouveaux nœuds de calcul dans le Cloud.

- **Préemptif** : la tâche active peut être interrompue à tout moment et le processeur alloué à une autre tâche
- **non préemptif** : la tâche est exécutée jusqu'à sa terminaison.

- **Optimal** : l'ordonnancement minimise les coûts d'exécution ou bien trouve toujours une faisabilité
- **Heuristique** : un algorithme est dit heuristique s'il cherche une faisabilité d'ordonnancement en utilisant une fonction empirique.

### II.2.3 Ordonnancement centralisé \ décentralisé :

L'ordonnancement centralisé assure le rendement et l'efficacité et garantie une surveillance facile des ressources, dans ce type d'ordonnancement les décisions sont prises dans un nœud central et a l'avantage de proposer un planning, l'avantage majeur d'ordonnancement centralisé est la tolérance aux évolutivités et aux panne. [14]

Dans l'ordonnancement décentralisé les décisions sont prises au pied de chaque machine sans avoir à considérer la situation des autres files d'attente donc ne permet pas de faire une estimation. [14]

### II.3 Migration des machines virtuelles :

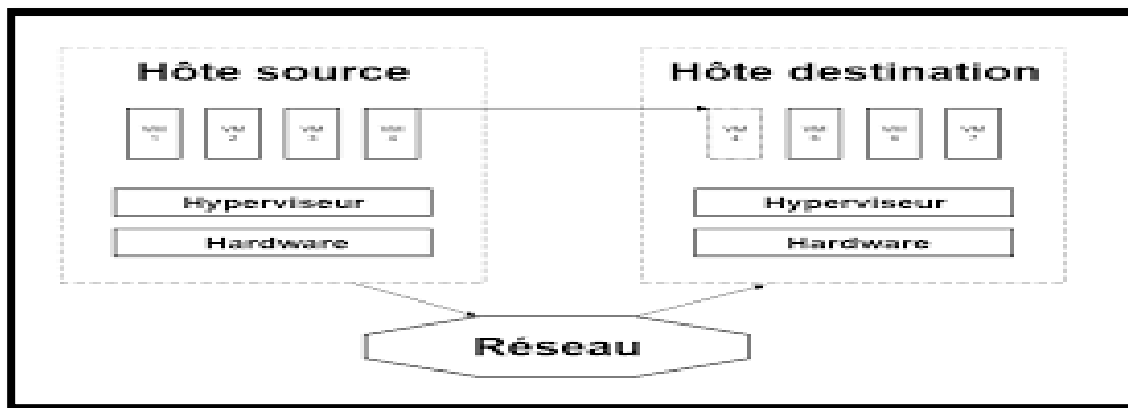
La migration des machines virtuelles consiste à déplacer les machines virtuelles d'un hôte physique (PM) vers une autre ou déplacer la machine virtuelle vers une autre ressource de calcul. [18]

La migration des machines virtuelles entre les différentes machines physiques est devenue indispensable dans les centres de données, les besoins de renforcement et de distribution de charge ont un fort bénéfice pour la migration des VMs et aussi les capacités d'hébergement des serveurs deviennent avec le temps très élevées. [18]

L'objectif principale de la migration est d'augmenter la performance des applications, réduire la consommation énergétique des centres de données, Les applications qui tournent sur le serveur doivent être transférées à un autre avant que le serveur ne soit éteint c'est pourquoi les chercheurs utilisent le principe de machine virtuelle.

Pour mieux comprendre l'approche de migration : Le principe de migration des machines virtuelles ou le placement des VM est proche de celui du remplissage d'un sac à dos qui peut contenir une quantité des machines virtuelles.

Le défaut majeur de la migration est la nécessité d'arrêter ou de mettre en pose la VM s'exécutent pendant la durée de son déplacement sachant que les applications et les services qui s'exécutent sur les VMs sont sévères et critiques il est vraiment difficile de les arrêter même pendant une courte période, Le problème est donc lorsqu'une VM est suspendue, les liaisons réseaux actives sont arrêtées et la durée d'interruption essentiel à sa migration est élevée pour permettre la prise des liaisons. C'est pour cela que les chercheurs ont créé un autre processus de migration, une nouvelle stratégie connue sur le nom « à chaud ». [18]



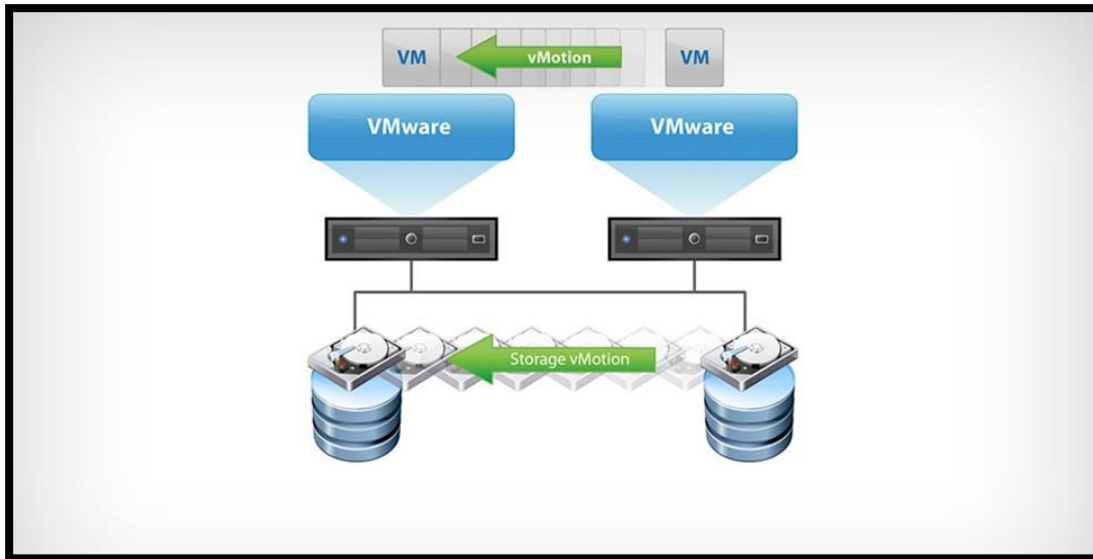
**Figure II.2:** migration des machines virtuelles [7]

### II.3.1 Les processus de migration :

#### II.3.1.1 Migration à chaud :

Cette stratégie consiste à déplacer une VM en cours d'exécution depuis sa machine physique vers une autre en assurant une période d'interruption minimale de la machine virtuelle, le coût important imposé par la migration à chaud et la diminution temporaire de performance, la perte de qualité de service est représentée par une faible durée d'interruption de la VM et aussi la réduction de la bande passante réseau alloué pendant la durée de migration. [8]

On retrouve deux algorithmes parmi les différents approches de la migration à chaud qui sont l'algorithme pré-copier et post-copier. [8]



**Figure II.3:**migration a chaud [40]

### ***II.3.1.2 Migration à froid :***

La migration à chaud est venue pour pallier les problèmes de la migration à froid. [7]

### **II.3.2 Les approches de migration :**

En générale, il existe deux approches de migration des machines virtuelles :

- Approche basée sur la qualité de service.
- Approche basé sur la consommation énergétique.

#### ***II.3.2.1 L'approche basée sur la qualité de service :***

La qualité de service désigne le degré de performance et de disponibilité qu'offrent les applications et l'infrastructure qui l'héberge, la qualité de service est un critère très important aux utilisateurs du Cloud Computing qui attendent la délivrance des caractéristiques annoncées par les fournisseurs. Ces derniers ont besoin de trouver un meilleur compromis entre les degrés de la qualité de service et le coût opérationnel, mais ce n'est pas facile de trouver le bon compromis qui est une décision difficile car elle indique la « service level Agréments (SLAs) » qui spécifie les objectifs de la QoS (qualité de service) et les pénalités économique. [8]

### II.3.3 L'approche basée sur la consommation énergétique :

L'approche est très utilisée pour diminuer le coût énergétique quand il est très élevé dans les data centres donc les machines physiques qui ont un très petit nombre des machines virtuelles pourront être éteinte. Cette approche contient 3 étapes habituellement :

#### 1. La détection de la surcharge de l'hôte :

Pour décider quand un certain hôte/serveur est surchargé dans cette phase la technique de l'ordonnancement doit avancer un seuil limite. Cette dernière appelée seuil chaud et dans le cas où cette limite est dépassée certaines des machines virtuelles de la machine physique doivent être migrées vers d'autres hôtes. [8]

#### 2. La détection de la charge réduite de l'hôte :

Quand un certain serveur est sous-utilisé c'est à dire il atteint sous le seuil chaud, le scénario est juste contraire de celui de l'hôte surchargé, alors l'objectif est l'identification de ce serveur et de migrer tous ses VM vers d'autre hôte actif donc l'hôte sous- utilisé soit libre et peut être éteint dans le but de économiser l'énergie. [8]

#### 3. La sélection de la machine virtuelle et la migration :

Les machines virtuelles sont sélectionnées soit des hôtes sous utilisés ou des hôtes surchargé pour la migration. Ainsi les machines virtuelles qui sont sélectionnées dans les phases précédentes sont placées dans autres machines physiques en fonction des critères de mappage adéquats. [8]

### II.4 Les algorithmes d'ordonnancement des tâches dans les Cloud Computing :

Il existe plusieurs algorithmes d'ordonnancement qui servent à planifier et ordonner les tâches dans le Cloud Computing en choisissant la meilleure ressource pour l'exécution des tâches ou pour allouer les machines à celle-ci.

Le choix de l'algorithme d'ordonnancement dépend des besoins spécifiques de chaque application et des contraintes de performance.

## **CHAPITRE II : LES ALGORITHMES ORDONNANCEMENT DES TÂCHES**

---

Il existe plusieurs algorithmes d'ordonnement dans le Cloud Computing tel que l'algorithme Round Robin, l'algorithme de priorité, l'algorithme de l'ordonnement de file d'attente, les algorithmes génétiques.

### **II.4.1 Classification des algorithmes d'ordonnement :**

Il existe deux modes d'ordonnement des tâches

#### **II.4.1.1 Espace partagé :**

L'allocation des ressources demandée est faite d'une façon dédiée à la charge du travail exigé pour l'exécution, et ne seront libérées qu'une fois terminées. [25]

#### **II.4.1.2 Temps partagé :**

Dans ce type d'ordonnement, les ressources demandées seraient partagées sur plusieurs charges de travail. [25]

Le mode temps partagé est une méthode d'ordonnement des tâches qui permet à chaque tâche d'utiliser le processeur pendant un temps limité avant de passer à la tâche suivante.

### **II.4.2 Les heuristiques et les métas heuristiques:**

Avant d'aborder les algorithmes d'ordonnement des tâches et leur fonctionnement, nous devons tout d'abord comprendre les méthodes heuristiques et les méthodes méta heuristiques.

#### **II.4.2.1 Les méthodes heuristiques:**

Sont des méthodes de résolution de problèmes qui utilisent des règles approximatives pour trouver une solution, généralement on n'obtient pas la solution optimale mais une solution approchée. [30]

#### **II.4.2.2 Les métas heuristiques:**

- Sont des algorithmes d'optimisation combinant plusieurs approches heuristiques. Ceux-là utilisent deux approches principales pour résoudre un problème.
- La première est nommée « approche à population » désignant les algorithmes qui traitent plusieurs solutions à la fois, elles maintiennent et améliorent plusieurs solutions candidates

## CHAPITRE II : LES ALGORITHMES ORDONNANCEMENT DES TÂCHES

en même temps par exemple l'algorithme génétique (GA), l'algorithme de fourmis(ACO) et l'algorithme d'optimisation par essais particuliers(PSO). [30]

- La seconde est nommée « approche de trajectoire » représentant les algorithmes qui font évoluer une fonction objectif unique à chaque itération par exemple : recherche Tabou. [30]

### II.4.3 Les algorithmes d'ordonnancement classique (de bases) des tâches :

Le principe fondamental des algorithmes d'ordonnancement des tâches dans l'environnement Cloud est de répartir la charge sur les processeurs et maximiser l'utilisation des ressources tout en minimisant le temps total d'exécution (Makespan). Par ce principe, ils essayent de trouver le meilleur mappage entre les tâches et les machines virtuelles. Autrement dit leur objectif est de minimiser le traitement total et les temps d'attente associés à l'ordonnancement afin de maximiser le débit du système, assurer et préserver les contraintes de qualité de service (Qos).

#### II.4.3.1 Algorithme aléatoire :

Ce type d'algorithme affecte aléatoirement les tâches aux machines virtuelles, l'algorithme ne considère pas si la machine virtuelle est sous charge lourde ou faible. Si dans le premier cas la machine virtuelle est sous charge lourde, le travail nécessite un long temps d'attente avant d'acquies le service, mais ne nécessite aucun surcoût ou prétraitement c'est pour cela sa complexité est faible.

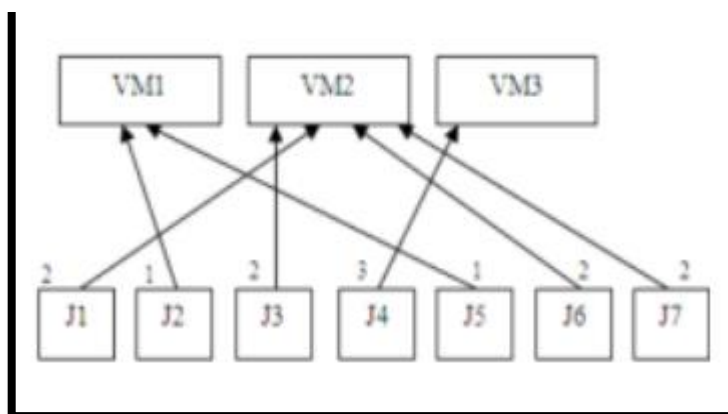


Figure II.4:le processus de l'algorithme aléatoire [12]

**Les entrées :** cloudletlist : la liste des tâches, VML : la liste de VM.

**Les sorties :** mapper chaque tâches à une machine virtuelle

---

**Les étapes :**

1.  $Nbcl \leftarrow \text{cloudletlist.size()};$
2.  $NbVM \leftarrow \text{VML.size()};$
3.  $i \leftarrow 0;$
4. **pour**  $j \leftarrow 0, j=Nbcl$  **alors**
5.      $cl \leftarrow \text{cloudletlist.get}(j);$
6.      $i \leftarrow \text{random()} * (NbVM - 1);$
7.      $v \leftarrow \text{VML.get}(i);$
8.      $\text{stagein} \leftarrow \text{TempsDeTransfert}(cl, v, \text{in});$
9.      $\text{stageout} \leftarrow \text{TempsDeTransfert}(cl, v, \text{out});$
10.     $\text{exec} \leftarrow \text{TempsDeExecution}(cl, v);$
11.    **Si**  $(cl\_AT + \text{stagein} + \text{stageout} + \text{exec} + v.RT \leq cl.DL)$  **alors**
12.        EnvoyerLeTravail( $cl, v$ );
13.        Mise-a-Jour( $v$ );
14.    **FinSi**
15.    **Sinon**
16.        Déposer( $cl$ );
17.        EchouéLeTravail;
18.    **FinSinon**
19. **FinPour**

Algorithme Aléatoire [12]



## CHAPITRE II : LES ALGORITHMES ORDONNANCEMENT DES TÂCHES

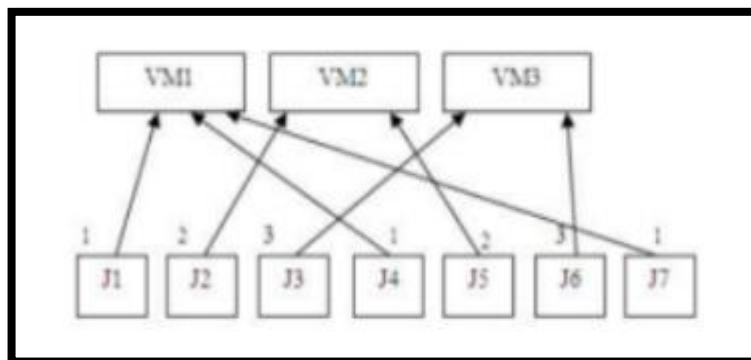
### II.4.3.2 FCFS (First Come First Served):

Le premier arrivé est le premier servi (First come First served). Ce type d'ordonnancement est utilisé dans un système à ordonnancement non préemptif ou sans réquisition. C'est -à-dire qu'il affecte les tâches automatiquement aux machines virtuelles (VM) dans l'ordre de leur arrivée.

### II.4.3.3 Round-robin :

L'algorithme Round Robin est un algorithme d'ordonnancement préemptif basé sur une tranche de temps et des cycles d'exécution. Chaque tâche dans la file d'attente prête pour la machine virtuelle a une chance équitable d'être exécutée à chaque tour et il se trouve que cette politique améliore le temps de réponse des tâches. Ici, un ensemble de tâches est alloué à un ensemble de VM dans un ordre cyclique.

Le défi associé à l'algorithme RR est l'estimation du bon quantum de temps pour toutes les itérations.



**Figure II.5:**le processus d'algorithme RR [12]

**Les entrées :** cloudletlist : la liste des tâches, VML : la liste de VM.

**Les sorties :** mapper chaque tâches à une machine virtuelle

**Les étapes :**

---

1.  $Nbcl \leftarrow cloudletlist.size()$  ;
2.  $NbVM \leftarrow VML.size()$  ;
3.  $i \leftarrow 0$  ;
4. **pour**  $j \leftarrow 0, j=Nbcl$  **alors**
5.      $cl \leftarrow cloudletlist.get(j)$  ;
6.      $i \leftarrow (i+1) \bmod NbVM$  ;
7.      $v \leftarrow VML.get(i)$  ;
8.      $stagein \leftarrow TempsDeTransfert(cl, v, in)$  ;
9.      $stageout \leftarrow TempsDeTransfert(cl, v, out)$  ;
10.     $exec \leftarrow TempsDeExecution(cl, v)$  ;
11.    **Si**  $(cl\_AT + stagein + stageout + exec + v.RT \leq cl.DL)$  **alors**
12.        EnvoyerLeTravail( $cl, v$ );
13.        Mise-a-Jour( $v$ );
14.    **FinSi**
15.    **Sinon**
16.        Déposer( $cl$ );
17.        EchouéLeTravail;
18.    **FinSinon**
19. **FinPour**

Algorithme Round Robin [12]

### II.4.3.4 SJF (*short Job First*) :

L'algorithme SJF est un algorithme classique d'ordonnement de tâche, il vous permet de choisir la tâche qui a la plus courte durée d'exécution parmi les tâches qui sont en attente pour les exécuter.

- Nous avons une liste de tâches sont prêtes à l'exécution avec le temps d'exécution de chacune. [6]
- La tâche ayant le temps d'exécution le plus court, sera sélectionné par l'algorithme à chaque instant. [6]
- La tâche sélectionnée dans la phase précédente est exécutée jusqu'à son achèvement. [6]
- Quand une nouvelle tâche arrive, l'algorithme doit comparer son temps d'exécution avec la tâche qui est en cours d'exécution. Si le temps d'exécution de la nouvelle tâche est plus court, Cette dernière prend la place de la tâche en cours d'exécution. [6]
- Lorsque la tâche est terminée, l'algorithme sélectionne la tâche ayant le plus court temps d'exécution. [6]

Le principe de cet algorithme est que les tâche les plus courtes ont le temps d'attente le plus court donc l'algorithme doit les traiter en priorité, ceci permet la réduction du temps moyen d'attente des tâches et l'amélioration de la performance du système.

Mais l'inconvénient majeur de SJF est la privation (la famine) pour les tâches longues, qui peuvent être constamment repoussées du profit des tâches plus courtes.

Pour régler ce défaut, les chercheurs améliorent le fonctionnement de ce algorithme pour qu'il permette l'interromptions de l'exécution d'une tâche longue pour exécuter une tâche plus courte en attente.

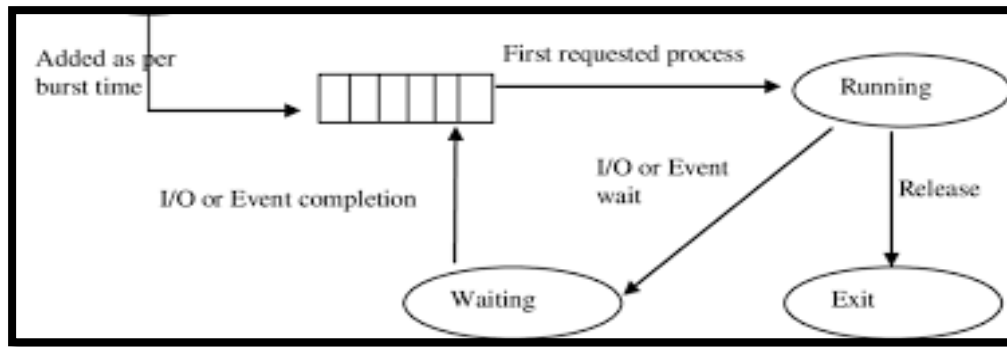


Figure II.6:Processus d'algorithme SJF. [32]

### II.4.3.5 Min-Min :

Cet algorithme se déroule en deux phases :

Phase 1 : Calculer les temps d'exécution de chaque tâche sur chaque machine, puis la machine qui traite les tâches en un minimum temps d'exécution sera sélectionnée.

Phase2 : Au milieu de toutes les tâches du méta tâche, la tâche qui a un temps d'exécution maximal est sélectionnée et est affectée à la machine. La tâche est supprimée de la liste Meta tâche et la procédure continue jusqu'à ce que Meta la liste des tâches est vide.

### II.4.3.6 Max-Min :

C'est le même principe que l'algorithme min-min sauf :

Phase 1 : Calcul d'abord le temps d'exécution de chaque tâche sur machine, puis pour chacun d'elle choisir la machine qui traite les tâches en un minimum de temps.

Phase 2 : Parmi toutes les tâches de la méta-tâche, la tâche avec un temps d'exécution maximal et sélectionnée et est affectée à la machine. La tâche est supprimée de la liste de Meta tâches et la procédure continue jusqu'à ce que Meta la liste des tâches est vide.

**Les entrées :** cloudletlist : la liste des tâches, VML : la liste de VM.

**Les sorties :** mapper chaque tâches à une machine virtuelle

**Les étapes :**

1. Initialisation
2.  $Nbcl \leftarrow cloudletlist.size()$ ;
3.  $NbVM \leftarrow VML.size()$ ;
4.  $index \leftarrow 0$  ;
5. **pour**  $j \leftarrow 0, j=Nbcl$  **alors**
6.      $cl \leftarrow cloudletlist.get(j)$  ;
7.      $min \leftarrow +\infty$
8.     **pour**  $i \leftarrow 0, i=NbVM$  **alors**
9.          $v \leftarrow VML.get(i)$  ;
10.         **Si**  $min > (v.getready() + cl.getlength()/v.speed)$  **alors**
11.              $min \leftarrow v.getready() + cl.getlength()/v.speed$ ;
12.              $index \leftarrow i$ ;
13.         **Finsi**
14.     **Finpour**
15.      $v \leftarrow VML.get(index)$ ;
16.      $stagein \leftarrow TempsDeTransfert(cl, v, in)$  ;
17.      $stageout \leftarrow TempsDeTransfert(cl, v, out)$ ;
18.      $exec \leftarrow TempsDeExecution(cl, v)$ ;
19.     **Si**  $(cl\_AT + stagein + stageout + exec + v.RT \leq cl.DL)$  **alors**
20.         EnvoyerLeTravail( $cl, v$ );
21.         Mise-a-Jour( $v$ );
22.     **FinSi**
23.     **Sinon**
24.         Déposer( $cl$ );
25.         EchouéLeTravail;
26.     **FinSinon**
27. **FinPour**

Algorithme Max-Mi

### II.4.4 Ordonnancement basé sur algorithme Génétique simple :

L'algorithme génétique dans l'ordonnancement des tâches est une technique d'optimisation. Il vise à trouver une solution efficace pour l'affectation des tâches aux machines virtuelles dans un environnement de Cloud Computing (VM)

Il fonctionne généralement selon les étapes suivantes :

1. **Initialisation** : Création d'une population initiale de chromosomes représentant différentes combinaisons d'attribution des tâches aux machines virtuelles.
2. **Évaluation de la fitness** : Chaque chromosome de la population est évalué en fonction d'une fonction de fitness qui mesure sa qualité en termes de critères spécifiques tels que le temps d'exécution, la consommation d'énergie, la latence, etc
3. **Sélection** : Une ou plusieurs méthodes de sélection sont utilisées pour choisir les chromosomes les plus performants de la population, qui serviront de parents pour la reproduction
4. **Croisement (Crossover)** : Les chromosomes sélectionnés sont combinés pour créer de nouveaux chromosomes en utilisant des techniques de recombinaison génétique. Cela permet de mélanger les caractéristiques des parents et d'explorer de nouvelles solutions potentiellement meilleures.
5. **Mutation** : Des opérations de mutation sont appliquées à certains chromosomes de la population pour introduire une exploration aléatoire et éviter une convergence prématurée vers un optimum local.
6. **Évaluation de la fitness des descendants** : Les nouveaux chromosomes obtenus après le croisement et la mutation sont évalués en fonction de la fonction de fitness pour déterminer leur qualité.
7. **Remplacement** : Les descendants remplacent certains membres de la population existante en fonction d'une stratégie de remplacement prédéfinie, généralement basée sur la fitness
8. **Répétition** : Les étapes 3 à 7 sont répétées pendant un certain nombre d'itérations ou jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint, tel que le nombre maximal d'itérations ou une convergence satisfaisante vers une solution optimale

**Les entrées :** M : taille de la population, Ub, Lb : domaine (borne supérieure, borne inférieure), D : dimension, C1, C2 : coefficients d'apprentissage personnel et global, MaxIt : nombre maximal d'itérations, Cloudletlist : liste des tâches, VML : liste des machines virtuelles

**Les sorties :** mapper chaque tâches à une machine virtuelle

**Les étapes :**

1. Initialisation de la population ;
2. Détermination du meilleur chromosome personnel et global ;
3. **Pour** i = 0 jusqu'à MaxIt, **faire**

4. **Pour** chaque chromosome de la population, **faire**

5. Générer des valeurs aléatoires r1, r2, w  $\in [0,1]$  ;

6. **Pour** chaque gène j du chromosome, **faire**

7. Calculer la vitesse  $V_{i,j(t+1)}$  selon l'équation :

$$V_{i,j(t+1)} = V_{i,j(t)} + c1 * r1 * (PBEST_{i,j} - X_{i,j(t)}) + C2 * r2 * (GBEST_j - X_{i,j(t)}) ;$$

8. Mettre à jour la position  $X_{i,j(t+1)}$  selon l'équation :

$$X_{i,j(t+1)} = X_{i,j(t)} + V_{i,j(t+1)} ;$$

9. **Si**  $X_{i,j(t+1)} > Ub$ , **alors**

10. Réduire la position à la borne supérieure :

$$X_{i,j(t+1)} = Ub ;$$

**FinSi**

11. **Si**  $X_{i,j(t+1)} < Lb$ , **alors**

12. Augmenter la position à la borne inférieure :

l'algorithme génétique [12]

### *II.4.4.1 L'optimisation par essais particulières (Particle Swarm Optimization) :*

L'optimisation par essais particulières est une méthode d'optimisation qui s'inspire du comportement des essaims d'oiseaux ou d'insectes.

Cette optimisation est basée sur les « interaction sociales » entre les agents qui sont appelés « des particules » où chacun d'elle est caractérisée par une capacité de mémorisation et de traitement de l'information.

Chaque particule représente une solution possible. À l'aide d'un groupe de particules, cet algorithme recherche la solution optimale dans l'espace de recherche, en tenant compte de chaque mouvement de particule. Cette méthode a été utilisée pour optimiser le temps total de réalisation des tâches assignées aux ressources.

Le principal inconvénient de cet algorithme est qu'il ne tient pas compte du budget de l'utilisateur

Le principal inconvénient de cet algorithme est qu'il ne tient pas compte du budget de l'utilisateur. [3]



**Les entrées :** M : taille de la population, Ub, Lb: domaine (borne supérieur, borne inférieur), D : dimension, C1, C2 : coefficient d'apprentissage personnel et global, MaxIt : nombre maximal d'itération.

Cloudletlist : la liste des tâches, VML : la liste de VM.

**Les sorties :** mapper chaque tâches à une machine virtuelle

**Les étapes :**

1. Initialisation ;
2. Déterminer personnelle et global best;
3. **Pour**  $i \leftarrow 0, i = \text{MaxIt}$  **alors**
4.     ensemble  $r1, r2, w \in [0,1]$  ;
5.     **pour**  $j \leftarrow 0, j = D$  **alors**
6.          $V_{i,j(t+1)} = c1r1(\text{PBEST}_{i,j} - X_{i,j(t)}) + C2r2(\text{GBEST}_j - X_{i,j(t)})$ ;
7.          $X_{i,j(t+1)} \leftarrow X_{i,j(t)} + V_{i,j(t+1)}$  ;
8.         **Si**  $X_{i,j(t+1)} > Ub$  **alors**;
9.              $X_{i,j(t+1)} \leftarrow Ub$ ;
10.         **FinSi**
11.         **Si**  $X_{i,j(i+1)} < Lb$  **alors**;
12.              $X_{i,j(i+1)} \leftarrow Lb$ ;
13.         **FinSi**
14.         **Si**  $X_i \leq \text{PBEST}_j$  **alors**
15.             Mise-a-Jour(PBEST);
16.              $\text{PBEST}_j \leftarrow X_i$ ;
17.         **FinSi**
18.         **Sinon**
19.             Mise-a-Jour(GBEST);
20.         **FinSinon**
21.          $\text{GBEST} \leftarrow \text{argmin}(\text{PBEST})$  ;
22.     **FinPour**
23. **FinPour**

l'Algorithme PSO [10]

## CHAPITRE II : LES ALGORITHMES ORDONNANCEMENT DES TÂCHES

### II.4.5 Comparaison entre les différents algorithmes d'ordonnancement des tâches :

Algorithmes	Méthodologie	Paramètres	Avantage	Inconvénient
Aléatoire	L'algorithme affecte aléatoirement les tâches aux machines virtuelles	///	<ol style="list-style-type: none"> <li>L'algorithme ne nécessite aucun surcoût.</li> <li>Complexité est faible.</li> </ol>	<ol style="list-style-type: none"> <li>L'algorithme ne considère pas si la machine virtuelle soit sous charge lourd.</li> </ol>
Round Robin	L'algorithme fonctionne sur cyclique approche dans laquelle chaque tâche a une chance égale d'être choisie et a une unité de temps tout aussi petite pour exécution	<ol style="list-style-type: none"> <li>Heure d'arrivée</li> <li>Tranche de temps</li> </ol>	<ol style="list-style-type: none"> <li>le temps de réponse est bon.</li> <li>équilibrer la charge de travail entre les différents serveurs ce qui peut améliorer l'efficacité globale du système.</li> <li>garantir un temps de réponse équitable pour toutes les tâches</li> </ol>	<ol style="list-style-type: none"> <li>la préemption entraine l'arrêt du processus une fois la tranche de temps expirée.</li> <li>inefficace pour les tâches qui ont des contraintes de priorité strictes, car il ne prend pas en compte la priorité des tâches lors de l'ordonnancement.</li> </ol>
SJF	Cet algorithme fonctionne sur une stratégie dans laquelle la tâche ayant un temps d'exécution minimum est sectionnée pour toutes les tâches	<ol style="list-style-type: none"> <li>Makespan</li> </ol>	<ol style="list-style-type: none"> <li>Minimiser le temps d'attente des tâches, car il donne la priorité aux tâches les plus coûteuses.</li> <li>optimiser l'utilisation des ressources, car il permet de libérer rapidement les ressources pour les tâches suivantes.</li> </ol>	<ol style="list-style-type: none"> <li>déséquilibre de charge.</li> <li>mauvaise qualité de service.</li> <li>entraîner un temps d'attente élevé pour les tâches longues, car il donne la priorité aux tâches les plus courtes.</li> <li>difficile à mettre en œuvre pour les tâches dont la durée est inconnue.</li> </ol>
FCFS (premier arrivé premier servi)	Cet algorithme gère l'ordonnancement des tâches avec file d'attente FIFO. la tâche qui vient en premier sera exécutée en premier	<ol style="list-style-type: none"> <li>Heure d'arrivée</li> </ol>	<ol style="list-style-type: none"> <li>simple à mettre en œuvre et à comprendre, car il traite toutes les tâches dans l'ordre dans lequel elles sont soumises.</li> <li>équitable, car il traite</li> </ol>	<ol style="list-style-type: none"> <li>l'ordonnancement des tâches est basée sur l'heure d'arrivée ne tient pas compte d'autres critères</li> <li>il peut entraîner un</li> </ol>

## CHAPITRE II : LES ALGORITHMES ORDONNANCEMENT DES TÂCHES

	sur la VM.		toutes les tâches de manière égale, sans donner de priorité à certaines par rapport à d'autres tâches	temps d'attente élevé pour les tâches courtes, car il traite les tâches dans l'ordre dans lequel elles sont soumises, sans tenir compte de leur durée de traitement.  3. il peut être sensible aux variations de charge de travail, car il ne prend pas en compte la charge de travail réelle lors de l'ordonnement.
Algorithme Min-Min	L'algorithme sélectionne la tâche avec temps d'exécution minimale	1. Makespan	1. Meilleur Makespan	1. Déséquilibre de charge. 2. Mauvaise qualité de service.
Algorithme génétique	L'algorithme génétique représente domaine de solution et une fonction appropriée pour estimer le domaine de solution	1. la taille de population. 2. probabilité de crossover. 3 - probabilité de mutation.	1. il peut résoudre les problèmes mathématiques et financiers avec plus de précision. 2. Facile à comprendre les concepts. 3. certaines demandes ont nécessite moins de temps pour le traitement.	1. la solution d'optimisation globale n'est pas atteinte par cet algorithme. 2. Très difficile d'apporter des modifications aux paramètres.

**Tableau II-1** Tableau comparatif entre les différents algorithmes d'ordonnement des tâches

### Synthèse :

Il existe plusieurs algorithmes d'ordonnancement des tâches dans le Cloud Computing, chacun avec ses avantages et ses inconvénients. Nous avons étudié des algorithmes d'ordonnancement de base (l'algorithme aléatoire, FCFS, SJF, RR, min-min, min\_max) et deux algorithmes d'intelligence artificielle (PSO et l'algorithme génétique).

Il existe d'autres algorithmes d'ordonnancement des tâches comme Ant Colony Optimization, Partical Swarm Optimization et plusieurs autres. Ces algorithmes sont conçus pour répondre à des besoins spécifiques en matière de Cloud Computing, tels que la minimisation du temps d'exécution des tâches, la réduction des coûts de traitement, l'amélioration de la qualité de service.

### Conclusion :

Pour conclure, l'ordonnancement des tâches est un processus d'ordonnancement d'affectation des ressources informatique telles que le stockage, la mémoire et la puissance de traitement, pour exécuter des tâches informatiques dans un environnement de Cloud Computing.

Dans cette partie nous avons cité les types d'ordonnancement, les notions générales de ce processus tel que l'ordonnanceur et son rôle, la migration et l'allocation des machines virtuelles.

Dans ce chapitre, nous avons étudié quelques algorithmes d'ordonnancement des tâches comme Round Robin, SJF, FCFS, Min-Min, Max-Min, PSO et algorithme génétique.

Dans le chapitre suivant, nous allons proposer une comparaison entre quelques algorithmes d'ordonnancement des tâches en fonction de plusieurs critères de performances comme le temps d'exécution, le coût, la fiabilité et la consommation d'énergie.

## **CHAPITRE III:**

**Etude comparative entre les  
algorithmes  
d'ordonnancement des tâches.**

### Introduction :

L'objectif de ce chapitre est de décrire notre travail, qui consiste à réaliser une comparaison entre les algorithmes d'ordonnancement des tâches en fonction de plusieurs critères de performance à savoir le Makespan, la consommation d'énergie, le coût et la fiabilité.

Dans la première partie de ce chapitre, nous allons présenter les critères performance ainsi que notre environnement de travail et les outils utilisés pour le développement notre projet surtout le simulateur CloudSim et ses différentes classes.

Dans la deuxième partie, nous allons exposer les configurations de simulation du Cloud Computing et les paramètres initiaux des algorithmes utilisés dans notre projet, ensuite nous allons présenter et analyser les résultats de performance de chaque algorithme.

### I. Environnement de travail :

#### I.1 Environnement matériel :

Ordinateur portable : hp.

Système d'exploitation : Windows 07.

Type de système : système d'exploitation 64bits

Processeur : Intel ® core i3.

Mémoire RAM : 4.00 Go.

#### I.2 Environnement de développement:



##### I.2.1 Langage de programmation Java :

Le langage de programmation Java est un langage orienté objet qui a été développé par Sun Microsystems dans les années 1990.

Il est devenu l'un des langages de programmation les plus populaires, utilisé pour créer des applications pour diverses plates-formes, telles que les ordinateurs de bureau (Java), les téléphones mobiles (Java ME (Micro Edition) et les serveurs (Java EE (Entreprise Edition)).

Une des caractéristiques clés de Java est sa portabilité, ce qui signifie que le code écrit en Java peut être exécuté sur n'importe quelle plate-forme sans avoir besoin d'être modifié.

Java est également connue pour sa sécurité, car il utilise un système de sécurité basé sur des classes appelé "sandbox". Cela permet d'exécuter du code Java potentiellement dangereux dans un environnement isolé, empêchant ainsi tout dommage potentiel à l'ordinateur hôte. En outre, Java dispose d'une vaste bibliothèque de classe standard qui peuvent être utilisées pour développer rapidement des applications.



### I.2.2 L'environnement de développement Eclipse :

L'environnement de travail Eclipse est un environnement de développement intégré (IDE) open source populaire utilisé pour le développement de logiciels Java. Il offre une variété de fonctionnalités, y compris la coloration syntaxique, l'auto complétion et la détection d'erreurs en temps réel. Les développeurs peuvent également personnaliser leur environnement de travail avec des plugins pour répondre à leurs besoins spécifiques.

Eclipse est également connu pour sa communauté active et son support pour de nombreux langages de programmation autres que Java, tels que C ++ et Python. Les développeurs peuvent collaborer sur des projets dans un environnement de travail convivial, ce qui facilite la gestion du code source et la résolution de problèmes.



### I.2.3 Le Simulateur CloudSim :

#### I.2.3.1 Définition :

CloudSim est un cadre de simulation qui permet la modélisation et la simulation d'infrastructures et de services de Cloud Computing.

Il permet aux utilisateurs de créer et d'évaluer différents scénarios pour optimiser les performances et la rentabilité de leurs applications basées sur le Cloud.

CloudSim fournit une gamme de fonctionnalités notamment :

- la prise en charge de plusieurs architectures Cloud.



- la gestion des machines virtuelles
- la modélisation de la charge de travail.
- le provisionnement des ressources.

Avec son interface conviviale et sa documentation complète, c'est un outil essentiel pour les chercheurs et les praticiens dans le domaine du Cloud Computing.

### I.2.3.2 Installation de CloudSim sous Eclipse :

Pour installer le CloudSim dans l'IDE Eclipse nous allons suivre les étapes suivantes :

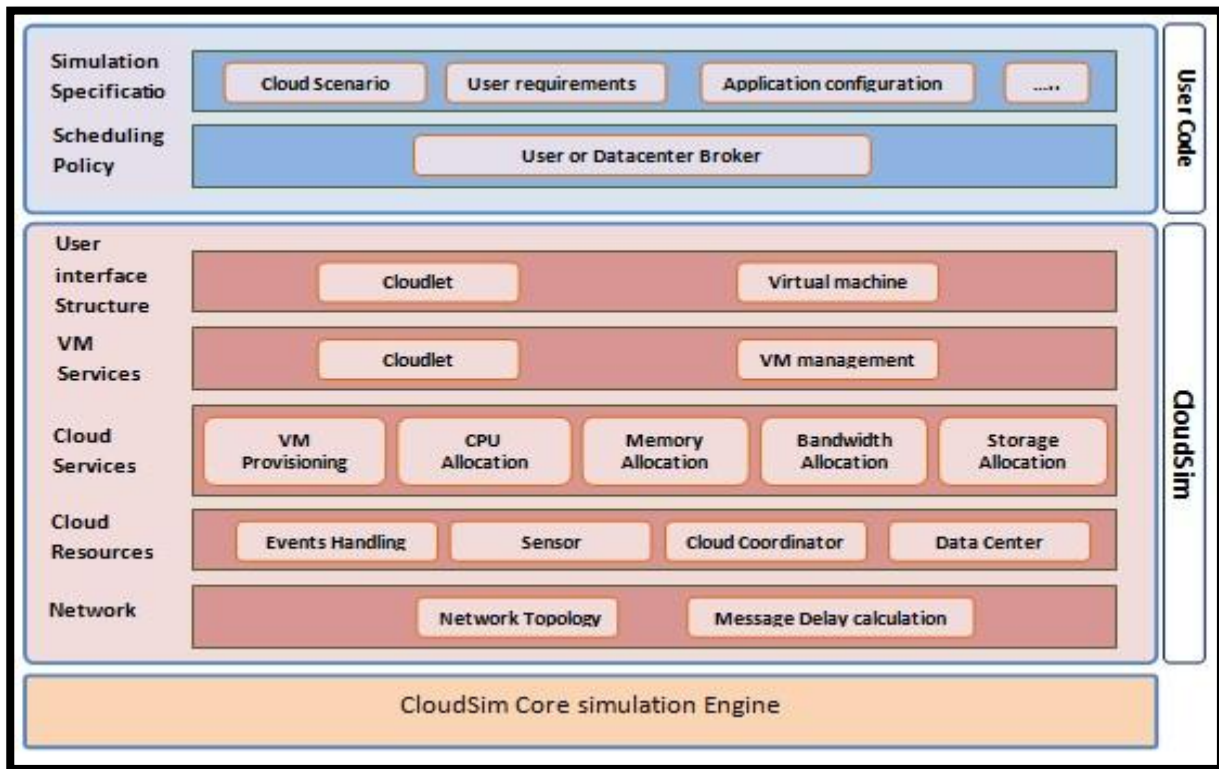
1. télécharger les fichiers suivants :
  - Clousim-3.0.toolkit; <https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3>
  - CloudSim : <http://redrockdigimark.com/apachemirror//commons/math/binaries/commons-math3-3.6.1-bin.zip>
2. La décompression de deux fichiers avant de lancer l'installation.
3. cliquez sur **File>New>** ; une boîte de dialogue s'ouvre Sélectionnez « **Java Projects** »
4. Dans la boîte de dialogue « **NewJava project** », entrer un nom de projet.  
Décocher "**use default location**" et cliquez sur "**Browse**" et accédez au chemin où vous avez extrait le dossier "CloudSim-3.0.3". Cliquez sur **NEXT**.
5. Après avoir cliqué sur Suivant, accéder à l'onglet « **Librairies** ». Vous devez cliquer pour ajouter la bibliothèque « **commons-math3-3.6.1.jar** ». Cliquez sur « **add external JARs** » et accédez au chemin au lequel vous avez Téléchargé et décompressé le fichier « **commons-math3-3.6.1.jar** ». Veuillez l'ajouter Cliquez sur Ouvrir pour l'ajouter à la liste.

### I.2.3.3 L'architecture de CloudSim :

L'architecture de CloudSim est conçue pour offrir une grande flexibilité et une haute performance. Elle se compose de plusieurs modules fonctionnels, tels que le module de gestion des machines virtuelles, le module de gestion des ressources, le module de modélisation de charges de travail, et le module de provisionnement de ressources.

Chaque module est conçu pour être hautement configurable et extensible, ce qui permet aux utilisateurs de personnaliser leur environnement de simulation en fonction de leurs besoins spécifiques. L'architecture de CloudSim est également conçue pour être facile à

utiliser, avec une interface utilisateur intuitive et une documentation complète pour aider les utilisateurs à tirer le meilleur parti de la plateforme. [27]



**Figure III.1:**Architecture de CloudSim. [26]

La figure ci-dessus illustre l'architecture en couche de CloudSim. La couche la plus basse représente le moteur de simulation aux événements discrets SimJava qui implémente les fonctionnalités de base requise de simulation au niveau supérieur, tels que la création du composants du système (les machines virtuelles, Datacenter, broker, hôte, services), la communication entre les composants et la gestion l'horloge de simulation et aussi les files d'attends et le traitement des événements. [26]

Le Cloud prise en charge de la modélisation et la simulation d'environnement de Datacenter, tel que les interfaces de gestion dédiées aux VMS, la mémoire, le stockage et la bande passante, les entités tel que (VMS, Hôte, Datacenter, applications) dans la couche supérieure de la pile de simulation, on retrouve le code de l'utilisateur qui propose de configurer les fonctionnalités liées aux hôtes par exemple le nombre de machine et leur spécification, les politiques d'ordonnancement de Broker, applications comme nombre de tâche et leurs besoins, VM, nombre d'utilisateur). [26]

### I.2.3.4 Les différentes classes de CloudSim :

- **Cloudlet :**

La classe Cloudlet est un élément clé de la plateforme de simulation CloudSim. Elle représente une tâche à exécuter sur une machine virtuelle dans le cadre d'une simulation de Cloud Computing. La classe Cloudlet contient des informations sur les besoins en ressources de la tâche, telles que la quantité de CPU et de mémoire requise, ainsi que la durée de la tâche. Elle permet aux utilisateurs de modéliser avec précision les différents types de charges de travail pour simuler des scénarios réels.

La classe Cloudlet est hautement configurable et peut être personnalisée pour répondre aux besoins spécifiques de l'utilisateur. Elle peut également être combinée avec d'autres classes pour créer des simulations plus complexes. En utilisant la classe Cloudlet, les utilisateurs peuvent simuler différentes stratégies d'ordonnancement de tâches et évaluer l'impact de ces stratégies sur les performances du système de Cloud Computing.

- **Datacenter :**

La classe Datacenter représente un centre de données virtuel dans le cadre d'une simulation de Cloud Computing.

La classe Datacenter contient des informations sur les ressources disponibles, telles que la quantité de CPU, de mémoire et de stockage, ainsi que sur les politiques de gestion des ressources. Elle permet aux utilisateurs de modéliser avec précision différents types de centres de données pour simuler des scénarios réels.

- **DatacenterBroker :**

La classe DatacenterBroker est un élément essentiel de la plateforme de simulation CloudSim. Elle permet de modéliser les courtiers en centres de données virtuels dans le cadre d'une simulation de Cloud Computing. La classe contient des informations sur les ressources disponibles, telles que la quantité de CPU, de mémoire et de stockage, ainsi que sur les politiques de gestion des ressources.

En utilisant la classe DatacenterBroker, les utilisateurs peuvent simuler différentes stratégies de gestion des courtiers en centres de données et évaluer l'impact de ces stratégies

sur les performances du système de Cloud Computing. La classe est hautement configurable et peut être personnalisée pour répondre aux besoins spécifiques de l'utilisateur. Elle peut également être combinée avec d'autres classes pour créer des simulations plus complexes.

- **Host :**

La classe host représente une machine physique ou un serveur informatique qui peut exécuter des tâches liées à la gestion des machines virtuelles, il contient deux politiques, une politique définie pour économiser la mémoire et la bande passante et une politique de distribution de processeur élément à des machines virtuelles. Un hôte est connecté à un Datacenter il peut héberger un ou plusieurs VMs.

- **DatacenterCharacteristic :**

Il s'agit de la classe qui comporte les informations sur la configuration des ressources des Datacenter tel que la liste des hôtes, le système d'exploitation, le coût par seconde d'utilisation des ressources.

- **VMAllocationPolicy :**

Elle permet de définir les politiques d'allocation des machines virtuelles dans le cadre d'une simulation de Cloud Computing. La classe **VMAllocationPolicy** contient des informations sur les critères d'allocation des machines virtuelles, tels que la disponibilité des ressources et les priorités des utilisateurs.

Elle permet aux utilisateurs de modéliser avec précision différents scénarios pour simuler des environnements réels.

### **I.2.3.5 Les packages de CloudSim :**

Il existe plusieurs packages disponibles dans CloudSim, chacun offre des fonctionnalités spécifiques pour la simulation de systèmes de Cloud Computing.

Les deux packages de bases de CloudSim sont Le package CloudSim-3.0.3/Exemples et Le package CloudSim-3.0.3/sources

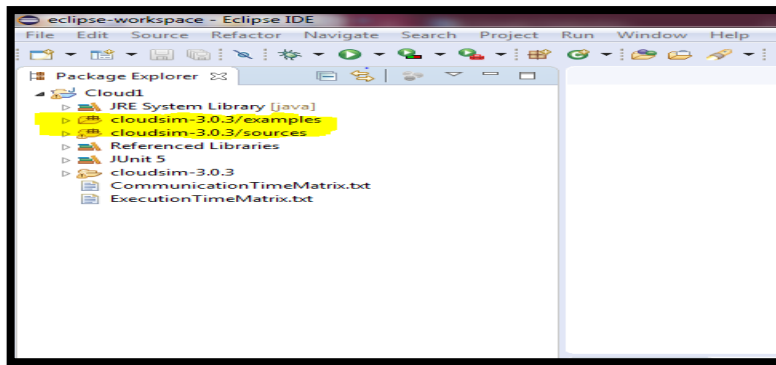


Figure III.2: les packages de bases de CloudSim.

- **Le package CloudSim-3.0.3/Exemples:**

Est un ensemble des exemples implémentés qui couvrent une variété de sujets, tels que la gestion de la mémoire, l'ordonnement des ressources et la gestion des machines virtuelles. Chaque exemple est accompagné d'une documentation détaillée qui explique son fonctionnement et ses paramètres, offrant ainsi une expérience d'apprentissage complète pour les utilisateurs de CloudSim.

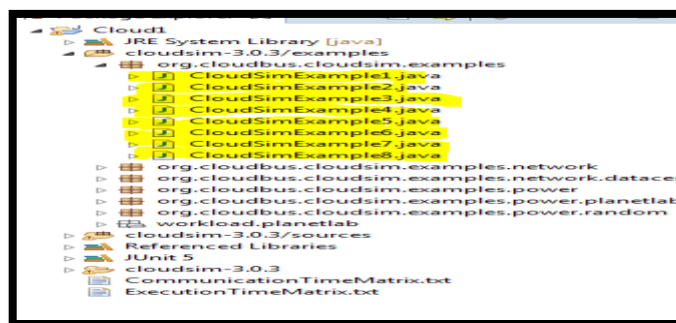


Figure III.3: Le package CloudSim-3.0.3/Exemples.

- **Le package CloudSim-3.0.3/sources :**

Est un outil puissant qui contient des différents sous-packages qui sont appelés à chaque implémentation tels que :

- **Le package org.cloudbus.cludsim :** Le package comprend divers modules permettant la création de machines virtuelles, le déploiement de piles logicielles, la gestion des ressources et l'analyse des mesures de performances (par exemple les classes : Cloudlet, Datacenter, Host CloudletScheduler...)



Figure III.4: Le package org.cloudbus.cludsim.

➤ Le package org.cloudbus.cludsim.lists : Ce package comprend une variété de types de listes, y compris qui sont la liste des Cloudlets, la liste des Hosts, la liste des Vms.... etc

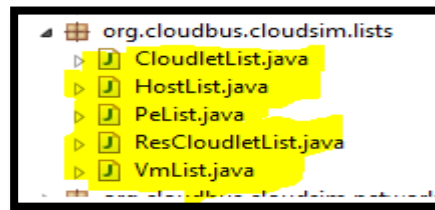
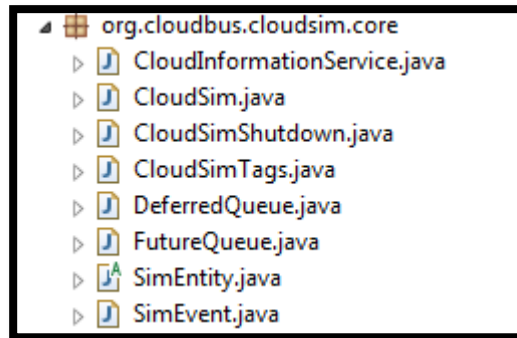


Figure III.5: Le package org.cloudbus.cludsim.lists.

➤ Le package org.cloudbus.cludsim.Core : Ce package est la base de CloudSim, fournissant une gamme de classes et d'interfaces qui forment le cœur des simulations informatiques distribuées. Ce package comprend plusieurs composants clés, tels que CloudSimEntity, CloudSimTag et SimEvent, qui fonctionnent ensemble pour permettre la création et la gestion de systèmes basés sur le Cloud.

Ce package fournit les outils et la flexibilité nécessaires pour créer des simulations Cloud puissantes et efficaces.



**Figure III.6:** Le package org.cloudbus.cludsim. Core.

➤ Le package org.cloudbus.cloudsim.distributions : Ce package contient des classes pour générer des nombres aléatoires à partir de distributions de probabilité spécifiques, telles que des distributions exponentielles, normales et uniformes.

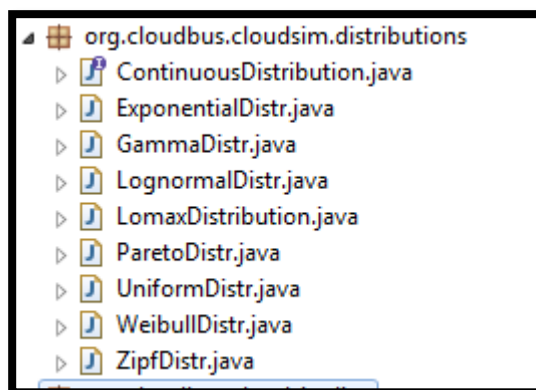
La classe **ExponentialDistribution** génère des nombres aléatoires à partir de la distribution exponentielle.

Cette distribution est couramment utilisée pour modéliser le temps entre les arrivées de requêtes ou de tâches.

La classe **Uniform Distribution** génère des nombres aléatoires à partir de la distribution uniforme.

La classe **NormalDistribution** génère des nombres aléatoires à partir de la distribution normale.

La classe **Gamma Distribution** génère des nombres aléatoires à partir de la distribution gamma.



**Figure III.7 :** Le package org.cloudbus.cloudsim.distributions.

➤ Le package org.cloudbus.cloudsim.provisioners :

Il fournit des classes et des interfaces pour la mise en œuvre des politiques de provisionnement des ressources dans les environnements Cloud Computing.

Le provisionnement fait référence au processus d'allocation de ressources telles que le processeur, la mémoire et le stockage aux machines virtuelles (VM) dans un environnement Cloud. Le package org.cloudbus.cloudsim.provisioners fournit diverses stratégies de provisionnement qui peuvent être utilisées pour allouer des ressources aux machines virtuelles en fonction de différents critères.



**Figure III.8:** Le package org.cloudbus.cloudsim.provisioners.

➤ Le package org.cloudbus.cloudsim.power :

Les principales caractéristiques est sa capacité à simuler le comportement de différents types de systèmes de Cloud Computing dans différentes conditions de charge de travail. Cela permet aux utilisateurs de comprendre comment différents facteurs, tels que le nombre de machines virtuelles ou le type d'application en cours d'exécution, affectent la consommation d'énergie.

Une autre caractéristique importante est la prise en charge par le package de différentes politiques et algorithmes de gestion de l'alimentation. Celles-ci incluent la mise à l'échelle dynamique de la tension et de la fréquence (DVFS), qui ajuste la tension et la fréquence des processeurs en fonction des demandes de charge de travail, et la consolidation, qui combine les charges de travail sur moins de machines physiques pour réduire la consommation d'énergie.



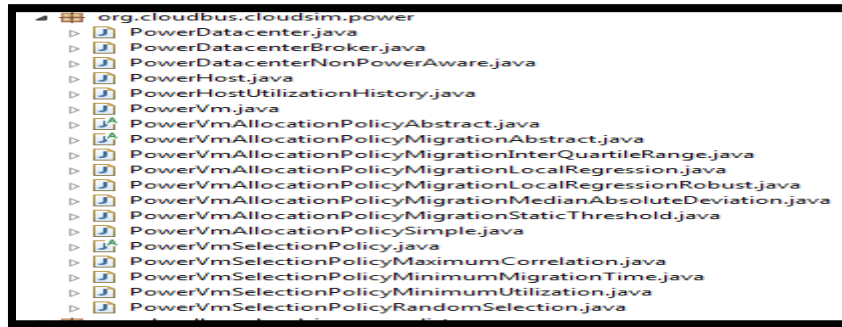


Figure III.9: Le package org.cloudbus.cloudsim.power.

## II. Les critères de performances :

Dans notre projet, nous allons comparer quelques algorithmes d'ordonnancement des tâches en fonction de plusieurs critères de performance à savoir le Makespan, la consommation d'énergie, le coût et la fiabilité.

### II.1 Temps total d'exécution (Makespan) :

Le Makespan est le temps total nécessaire pour effectuer un groupe de tâches qui indique l'heure d'accomplissement de la dernière tâche. La tâche d'ordonnancement consiste à minimiser le Makespan comme la plupart des utilisateurs qui souhaitent une exécution plus rapide de ses applications. [22]

La formule de calcul de Makespan est illustrée dans l'équation :

$$\text{Makespan} = \text{Max}(\text{FinishTime}).$$

(**FinishTime** : le moment où la tâche termine son exécution).

$$\text{FinishTime} = \text{ExTime} + \text{Start Time}.$$

(**ExTime** : le temps exact mis afin d'exécuter la tâche donnée sur la machine virtuelle

Tel qu'il est défini dans l'équation suivante :

$$\text{ExTime} = \text{LT} / \text{VMP}. [22]$$

Où le :

**VMP** est le traitement de puissance de la machine virtuelle (VM).

**LT** est la longueur de la tâche).

(**Start Time**: le moment où la tâche commence son exécution.

### II.2 Le coût :

Le coût total engendré par l'exécution d'une tâche distribuée peut comprendre de nombreux éléments de coût, tels que le coût de calcul et le coût de transfert de données. [24]

Le Coût total d'exécution des tâches est défini par l'équation suivante :

$$\text{Le Coût} = \sum_{i,j=0}^{N,M} \text{CoûtEx}(i, j). [1]$$

Où le :

**N** est le nombre de tâches à exécuter.

**M** est le nombre des machines virtuelles.

**CoûtEx(i, j)** est le coût d'exécution de la tâche *i* par le DC *j*.

$$\text{CoûtEx}(i,j) = \text{ExTime} * \text{PrixU}_{\text{Ex}}. [1]$$

### II.3 La consommation d'énergie :

Le lien entre les contraintes énergétique et l'activité des tâches dans un environnement de Cloud Computing peut être particulièrement fort. [24]

L'intégration énergétique a pour objectif de réduire la consommation en optimisant les échanges de données, le transfert de données et le calcul. [24]

Les problèmes d'ordonnancement en temps réel et la minimisation d'énergie deviennent un caractère essentiel dans la conception des Cloud Computing. [24]

Pour calculer l'énergie consommée par l'ensemble des Datacenter, on applique l'équation suivante :

$$\text{Energie} = \sum_{i,j=0}^{N,M} \text{energieEx}(i, j). [1]$$

N est le nombre de tâche exécuté.

**EnergieEx(i,j)** est l'énergie consommée lors d'une exécution d'une tâche i par le DCj.

$$\text{EnergieEx}(i,j) = \text{ExTime}(i,j) * K [1]$$

Avec

**K** est une quantité énergétique consommée à chaque exécution.

#### II.4 La fiabilité :

La fiabilité d'exécution d'une tâche également abordée. Cet objectif garantit que les ressources sélectionnées dans un ordonnancement vont probablement terminer l'exécution des tâches qui leur sont programmées. [24]

La fiabilité est définie par l'équation suivante :

$$\text{Fiabilité} = e^{-\sum_{i=0}^n \text{Temps d'exécution}(i) * \lambda} . []$$

Où

$\lambda$  est le taux d'echec de la machine qui exécute la tâche .

**n** le nombre de tâche.

### III. Implémentation :

La figure n°10 présente la page d'accueil de notre application nommée « Algocompare » pour remplir le nombre des Datacenter et le nombre des tâches qui sont les principaux paramètres des algorithmes, l'utilisateur doit remplir les informations nécessaires et appuie sur next pour passer vers la deuxième page.



Figure III.10: fenêtre principale de l'application.



Figure III.11: fenêtre de configuration de l'algorithme génétique.

Cette figure illustre les paramètres initiaux de l'algorithme génétique dans laquelle l'utilisateur doit remplir les informations nécessaires suivantes : population\_size, itération\_size et la sélection\_rate et appuie sur next pour passer vers l'étape suivante.



Figure III.12: fenêtre d'affichage des résultats.

Cette fenêtre affiche les résultats de simulations de chaque algorithme.

### III.1 Environnement de simulation:

Dans cette partie, nous allons présenter les configurations de simulation de notre Cloud Computing à savoir les Datacenter, machines physiques, machines virtuelles et les tâches.

#### III.1.1 Paramètres de l'environnement :

Pour notre étude comparative et pour simuler les différents critères de performance, nous avons travaillé avec un ordonnancement statique et des tâches indépendantes, hétérogènes, des machines virtuelles avec les mêmes caractéristiques et des machines physiques homogènes.

## CHAPITRE III: Etude comparative entre les algorithmes d'ordonnancement des tâches

### III.1.1.1 Les caractéristiques des Datacenters :

Nombre DSC	Nombre des hôtes	Système d'architecture	Système opération	Prix CPU (\$/s)	Prix bande passante (\$/s)	Prix RAM (\$/s)	Prix de stockage (\$/s)	Time zone this resource located
20	20	“X86“	“Linux“	3	0.1	0.05	0.1	10

**Tableau III-1:** les caractéristiques des Datacenters.

### III.1.1.2 Les paramètres des machines physiques :

Nombre des hôtes	Mips	CPU	bande passante	RAM	stockage	Politique
20	1000	1	2000000 Mb/s	2048 MB	1000000	Temps partagé

**Tableau III-2:** les paramètres des machines physiques.

### III.1.1.3 Les paramètres des machines virtuelles :

Nombre de machines virtuelles	Mips	CPU	RAM	Bande passante	Taille
20	1000	1	512 MB	1000MB	10000

**Tableau III-3 :** les paramètres des machines virtuelles.

### III.1.1.4 Les paramètres des tâches:

Nombre de tâches	Longueur	CPU	Taille de fiche	Taille de sortie	Mode d'utilisation
200, 400,800 et 1600	Hétérogène 1000 à 2000	1	300	300	Full

**Tableau III-4:**les paramètres des tâches.

### III.2 Résultats et discussion:

Notre travail se base sur l'ordonnancement des tâches statique, en fonction de quatre critères de performance à savoir le Makespan, la consommation d'énergie, le coût et la fiabilité.

Nous allons travailler sur les algorithmes d'ordonnancement des tâches : le SJF, FCFS, RR, Min\_Max et l'algorithme génétique GA.

Nous effectuons également plusieurs simulations, en faisant varier le nombre des tâches (200, 400, 800 et 1600).

Nous présentons les valeurs de la simulation dans un tableau et un histogramme.

Nous comparons les résultats obtenues pour chaque métrique de chaque algorithme.

Enfin, nous discutons et décidons qui est le meilleur algorithme pour l'ordonnancement des tâches dans le Cloud Computing.

Résultat du Makespan :

Nombre des tâches	200	400	800	1600
Min_Max	7302.72	17046.92	15965.30	17123.83
FCFS	8580.16	17646.49	19753.23	22790.73
RR	9033.68	18175.42	20911.80	47888.70
SJF	8041.34	16618.55	19570.75	50481.75
GA	57.77	104.00	213.55	479.86

Tableau III.5: Résultat du Makespan.

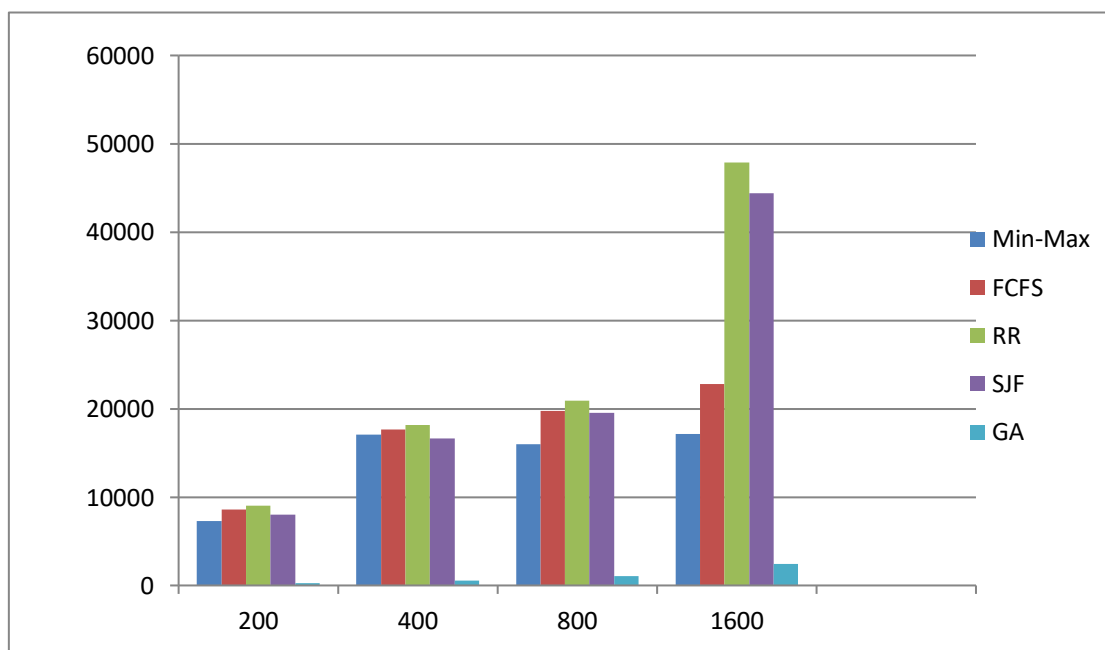


Figure III.13: Histogramme du résultat de Makespan.



### III.2.1 Discussions :

On remarque que lorsque le nombre des tâches augmente, les valeurs de Makespan augmentent au niveau de chaque simulation pour chacun des cinq algorithmes exécutés.

Pour les algorithmes de base : d'après la figure N° 12 les valeurs de Makespan pour l'algorithme FCFS sont élevées car les tâches longues peuvent bloquer les tâches courtes, et si toutes les tâches sont affectées par cet algorithme de manière inefficace à la machines virtuelles cela peut entraîner des temps d'attente supplémentaires, sachant que l'affectation des tâches à des machines virtuelles peut avoir un impact significatif sur le Makespan.

Les valeurs de Makespan pour l'algorithme RR est également élevé car dans certains cas cet algorithme partage le temps de traitement des tâches en segment de durée fixe. Si une tâche nécessite plus de temps que la durée de segment, elle doit être interrompue et reprise plus tard, ce qui peut entraîner des temps d'attente supplémentaires.

Si on compare entre les deux algorithmes précédents pour chaque simulation on remarque que le Makespan de RR est élevé que celui de FCFS, les tâches ont des temps d'exécution différents, alors l'algorithme RR peut entraîner des temps d'attente supplémentaires plus que le FCFS.

Le Makespan de SJF est supérieur à celui de FCFS pour un ensemble de 1600 tâches en raison de la stratégie d'ordonnancement utilisée. SJF ordonne les tâches en fonction de leur durée d'exécution ce qui signifie que les tâches courtes sont exécutées en premier.

On remarque aussi que les valeurs de Makespan de l'algorithme min-max sont aussi augmentées, lorsque le nombre de tâches augmente au niveau de chaque simulation pour chacun des cinq algorithmes.

D'après l'histogramme, les valeurs de Makespan de l'algorithme SJF est plus courtes que le FCFS et RR, car les tâches courtes sont traitées en premier, ce qui réduit les temps d'attente pour toutes les tâches et réduit le Makespan.

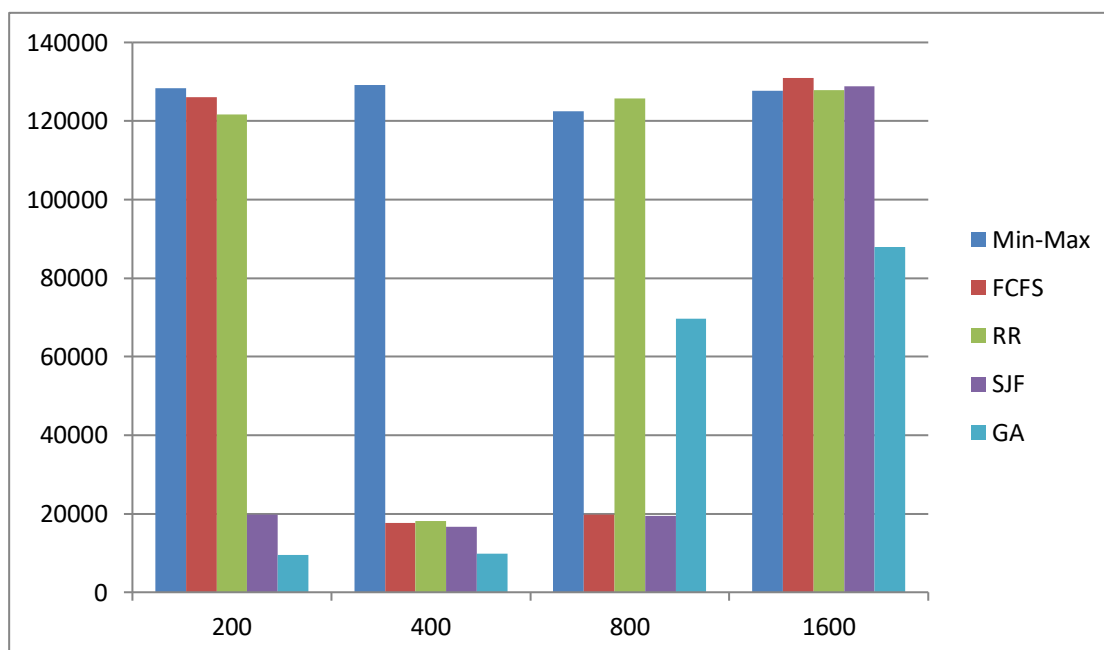
De plus on remarque que l'algorithme génétique GA a généré les minimums de ces résultats par rapport aux autres algorithmes car il est possible que les paramètres de

l'algorithme génétique (population size, nombre d'itération, sélection rate) soient bien adaptés à notre simulation.

**III.2.2 Résultat d'énergie :**

Nombre des tâches	200	400	800	1600
Min_Max	128340.27	129221.14	122508.17	127728.32
FCFS	126006.11	129469.11	129890.41	130984.80
RR	121594.96	124926.48	125749.09	127892.9
SJF	119687.67	121672.6	124192.48	128815.58
GA	4677.75	9564.38	69676.26	87896.29

**Tableau III.6:**Résultat de consommation d'énergie.



**Figure III.14:** Histogramme de résultat de consommation d'énergie.

### III.2.3 Discussion:

Comme la discussion précédente, on remarque que lorsque le nombre de tâche augmente, les valeurs de la consommation d'énergie augmentent au niveau de chaque simulation pour chacun des cinq algorithmes exécutés.

D'après les résultats obtenus : pour les algorithmes de base on remarque que le FCFS a des valeurs de consommation d'énergie plus élevé par rapport aux autres algorithmes (RR, SJF, min\_max). Cet algorithme peut entraîner une utilisation inefficace des ressources et une consommation d'énergie excessive si les tâches ne sont pas affectées de manière appropriée.

le RR peut être plus efficace en termes de consommation d'énergie car il permet une utilisation plus équilibrés des ressources.

L'algorithme génétique GA a généré également les minimums valeurs de ces résultats, car l'algorithme génétique peut être plus efficace en termes de consommation d'énergie que certain autres algorithmes car il utilise des processus de sélection naturelle et de croisement pour trouver des solutions optimales plutôt que de simplement tester toutes les options possibles.

III.2.4 Résultat de coût :

Nombre des tâches	200	400	800	1600
Min_Max	337568.92	34748.85	347491.62	356759.8
FCFS	349400.79	360671.02	363242.08	366429.71
RR	344447.42	348723.00	367066.60	375250.73
SJF	338361.00	341161.41	344705.08	347340.04
GA	1814.62	3409.09	11901.27	47416.29

Tableau III-7: Résultat de coût.

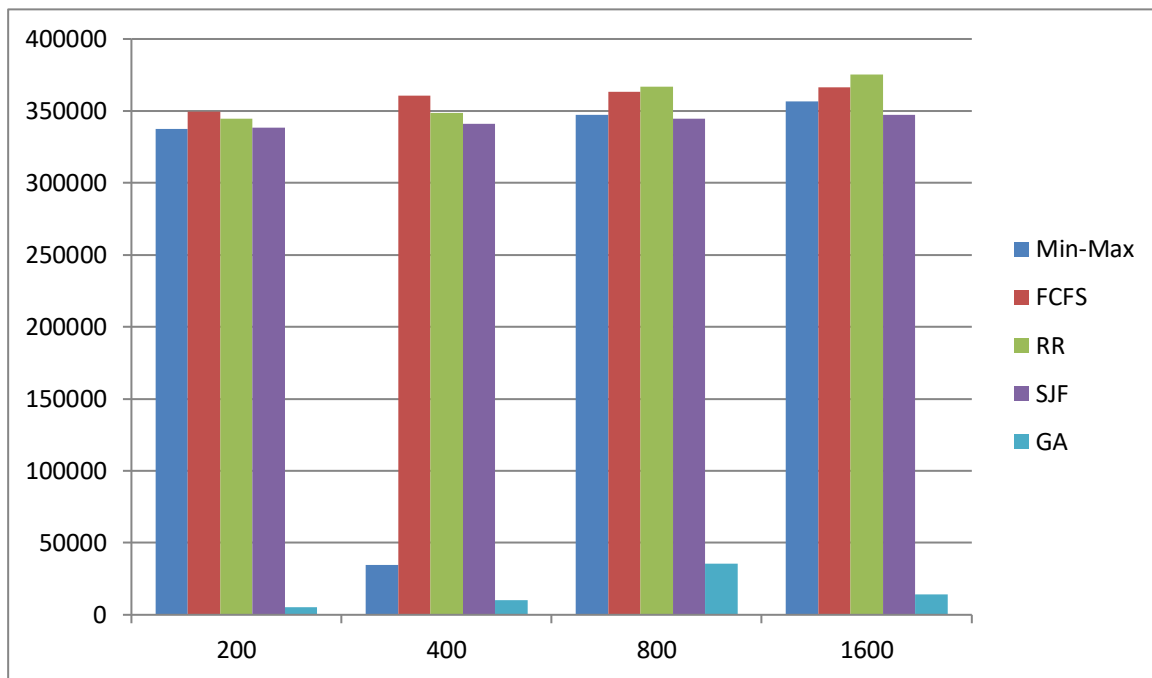


Figure III.15: Histogramme de résultat de coût .

III.2.5 Discussions :

On remarque que lorsque le nombre de tâche augmente, les valeurs de coût générées au niveau de chaque simulation augmentent pour chacun des cinq algorithmes exécutés,

On remarque que les algorithmes FCFS, SJF, min\_max et le FCFS peuvent être très coûteux pour les grands ensembles de tâches, ils peuvent entraîner des temps d'attente élevé pour les tâches.

Le coût de l'algorithme génétique est plus petit car l'utilisation de techniques d'optimisation telles que la sélection de population, la mutation et la recombinaison peut aider à réduire le coût.

#### III.2.6 Résultat de La fiabilité :

Nombre des tâches	200	400	800	1600
min_max	0.9935	0.9943	0.9944	0.9955
FCFS	0.9933	0.9944	0.956	0.9962
RR	0.9903	0.9966	0.9970	0.9987
SJF	0.9894	0.9949	0.9968	0.9983
GA	0.9999	0.9999	0.9999	0.9999

Tableau III.8:Résultat de fiabilité.

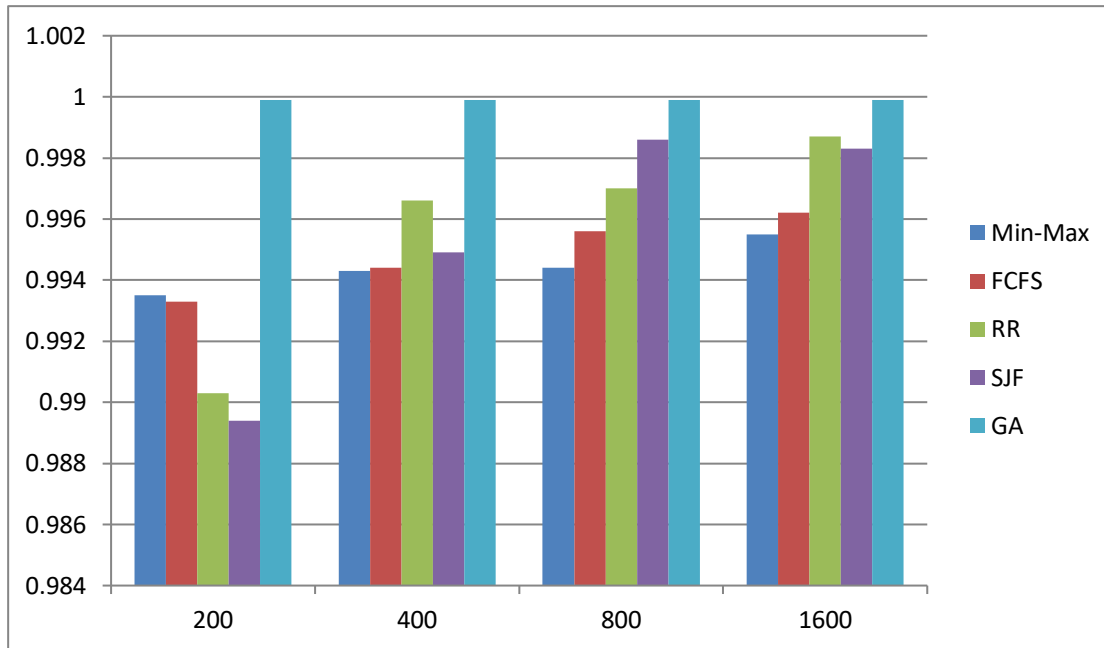


Figure III.16: Histogramme de résultat de fiabilité

### III.2.7 Discussions :

À base des résultats obtenues on remarque que peu importe le nombre de tâche les valeurs de fiabilité générées au niveau de chaque simulation pour chacun des cinq algorithmes sont presque 99%, mais en remarque que l'algorithme GA est le plus fiable par rapport aux autres algorithmes RR, SJF, Min\_Max et le FCFS.

### III.2.8 Synthèse :

D'après les résultats obtenus de notre simulation, c'est intéressant de constater que les résultats de l'algorithme génétique (GA) sont toujours les plus petites en termes de critères de performance (Makespan, consommation d'énergie, le coût et la fiabilité). Cela peut indiquer que l'algorithme génétique est particulièrement bien adapté pour résoudre le problème que nous avons étudié. Cependant, il est important de noter que les résultats de l'algorithme génétique peuvent varier en fonction des paramètres choisis, les caractéristiques des tâches, le type et les caractéristiques d'ordonnancement.

Le Makespan change dans chaque simulation en raison de la variabilité des facteurs (nombre de tâches, leur temps d'exécution, la capacité de traitement de système) qui influencent la performance de l'algorithme d'ordonnancement.

Les tâches hétérogènes, peuvent avoir un impact significatif sur les valeurs des critères de performance, si on a travaillé avec des tâches homogènes les résultats des simulations vont changer.

D'après les résultats de notre simulation, le temps de l'exécution de l'algorithme min\_max été plus élevé par rapport aux autres algorithmes pour chaque simulation.

#### **Conclusion :**

Dans ce chapitre, nous avons présenté les outils utilisés qui nous ont aidés pour faire cette étude comparative à savoir le langage java Eclipse et le simulateur CloudSim.

Nous avons travaillé sur cinq algorithmes d'ordonnancement de tâches tels que RR, SJF, min\_max, FCFS et l'algorithme génétique.

Nous avons évalué les algorithmes en fonction de quatre critères de performances : le Makespan, la consommation d'énergie, le coût et la fiabilité.

Après plusieurs simulations, Les résultats obtenus montrent que l'algorithme génétique améliore quelques critères de performance, mais il est important de noter qu'il est gourmand en temps de calcul.

## Conclusion générale :

L'ordonnancement des tâches est une technique très importante dans le Cloud Computing, elle vise à optimiser la façon dont les tâches sont planifiées et exécutées sur les machines virtuelles.

L'objectif de ce projet est de présenter une étude comparative entre les algorithmes d'ordonnancement des tâches en fonction de plusieurs performances à savoir le Makespan, le coût, la fiabilité et la consommation d'énergie.

Nous avons comparé les algorithmes d'ordonnancement RR, SJF, min\_max, FCFS et l'algorithme génétique à l'aide de simulateur CloudSim.

A travers les différentes simulations réalisées en changeons le nombre des tâches, nous avons constaté une nette amélioration de performance pour l'algorithme génétique.

Suite à réalisation de notre étude, il serait intéressant de proposer des solutions automatiques pour l'ajustement des paramètres initiaux de certains algorithmes. Réduire le temps de calcul l'algorithme génétique à travers le calcul parallèle et distribué. Ajouter d'autres algorithmes d'ordonnancement dans notre étude et évaluer d'autres critères de performance.



# **bibliographie**

## Bibliographie:

---

---

### **I. Pour les articles :**

- [12] MOHIALDEEN I. «Comparative study of scheduling AI-gorithms in cloud computing environment ». Journal of computer science, 9 (2), 2013. p 252- p263.
- [13] RAHMANI. N. 2013. « Ordonnancement dans les systèmes en temps réel » Conception de système temps réel. Cours en ligne. Université d'OTTAWA. 8p.
- [14] VIJAYALAKSHMI A. LEPAKSHI, PRASHANTH C S R. « Une étude sur les algorithmes de planification de tâche dans le Cloud Computing ». Journal of Engineering and Innovative Technology (IJETT), 2(11), 2013. P119-P124.
- [18] VALERIE F. VALERIE FE. SABINE KH, 2018, annales des mines-Gérer et comprendre, Institut mines-télécom, 68p, (N° 133)

### **II. Pour les mémoires :**

- [1] AILAM. M. BAFDEL. M. 2019. « Ordonnancement de tâches dans le Cloud ». Mémoire de master. Université de MOULOU D MAMMERI TIZI-OUZOU. 101p.
- [2] BENSASSIF F. ARRARIA. S. 2020. « Ordonnancement temps réel des tâches indépendantes sur environnements de Cloud Computing ». Mémoire de master. Université d'IBN KHALDOUN TIARET. 84.
- [3] BOUYEBBA M. 2020. « Ordonnancement des commandes client, simulation par PSO ». Mémoire de master. Université de Mohamed Boudiaf Msila. 45p.
- [4] CHAARI T. 2010. « Un algorithme génétique pour l'ordonnancement robuste : application au problème du flow hybride ». Thèse de doctorat. Université de Valenciennes et du Hainaut –Cambrésis. 145p.
- [5] FALLAH. H. 2014. Cloud Computing et sécurité : une architecture organique pour la sûreté de fonctionnement des processus métiers. Mémoire de master. Université de LARBI Ben M'hidi OUM EL BOUAGHI. 143p.
- [6] HAMDANI N, KERROUM S, OUALLOUCHE C. 2018. « Etude et comparaison des failles de sécurité d'OpenStack et OpenNebula ». Mémoire de master. Université de MOULOU D MAMMERI de TIZI-OUZOU. 138p.
- [7] KIARED A. 2019. « Migration de machine virtuelle en temps réel ». Mémoire de master. Université de M'HAMED BOUGARA BOUMERDES. 51p.

## Bibliographie:

---

---

- [8] MANIRAKIZA N. RAKOTONIRINA Z. 2016. « La migration en temps réel des machines virtuelles dans le Cloud Computing ». Mémoire de master. Université de MOULOUDE MAMMERI DE TIZI-OUZOU. 85p.
- [9] MEHDID M, MOUAOUED M. 2022. « Conception d'une plateforme Web de dimensionnement d'un vEPC (virtualized Evolved Packet Core) dédiée pour la 5G ». Mémoire de master. Université Aboubaker Belkaid de TELEMEN. 104p.
- [10]BOUYEBBA M. 2019. « Ordonnement des commandes client, Simulation par PSO». Mémoire de master. Université MOHAMED BOUDIAF-M'SILA.51p.
- [11] ZERROUKI A. ZRROUKI I. 2020. « Ordonnement des workflows scientifique dans un environnement Cloud Computing ». Mémoire de master. Université de TAHAR MOULAY SAIDA.75.
- [34] DERRAR A. AYED ZEDDAM M. 2020. « Optimisation de la QoS dans le Cloud Computing en utilisant les algorithmes génétique ». Mémoire de master. Université BELGADJ BOUCHAIB Ain Temouchent.87p
- [33]NOUMRI M. 2021 . «Introduction à la virtualisation Introduction to virtualization ». Mémoire de master. Université el djelfa.

### III. Pour les sites web :

- [16] «Qu'est-ce que le Cloud Computing ?»,8 juin 2014. [En ligne]. Available: [<https://blog.blaisethirard.com/qu-est-ce-que-le-cloud-computing/>]. [Accès le Avril 2023].
- [17] «Machine virtuelle : comment ca marche et pourquoi l'utiliser» 28 octobre 2019 [En ligne]. Available: [<https://bluebearsit.com/machine-virtuelle-comment-ca-marche/>]. [Accès le mars 2023].
- [19]«Ordonnement»[[http://www.cetice.universite-paris-saclay.fr/aunege/gestion\\_flux/co/quizzsortie.html](http://www.cetice.universite-paris-saclay.fr/aunege/gestion_flux/co/quizzsortie.html)]. [Accès le mars 2023].
- [20] «How to do Virtual machine and Task Scheduling in CloudSim» 17 juin2019 .Available: [<https://www.cloudsimtutorials.online/how-to-do-virtual-machine-and-task-scheduling-in-cloudsim/>] [Accès le mai 2023].
- [21]<https://www.compufirst.com/compufirst-lab/cloud/cloud-avantages-inconvenients/>(indisponible)

## Bibliographie:

---

---

- [22] «Virtualisation de systèmes d'exploitation» 6 mars 2023 . Available: [<https://doc.ubuntu-fr.org/virtualisation#ressources>].
- [23] «comment fonctionne l'équilibrage de charge». 2023. Available: [<https://fr.zevenet.com/blog/how-load-balancing-works/>].
- [24]<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FTa-xonomie-des-algorithmes-dordonancement>(indisponible)
- [25] «Le framework de simulation de l'environnement du Cloud» 20 Juin 2015. Available: [<https://graal.ens-lyon.fr/~ecaron/m2rts/2015/blogbazm/>].
- [26] « Le framework de simulation de l'environnement du Cloud»,21 juin 2019. Available: [[https://www.ionos.fr/digitalguide/serveur/configuration/la-virtualisation/#:~:text=Le%20terme%20de%20virtualisation%20hardware,\(abrégé%20VM%20en%20anglais\)](https://www.ionos.fr/digitalguide/serveur/configuration/la-virtualisation/#:~:text=Le%20terme%20de%20virtualisation%20hardware,(abrégé%20VM%20en%20anglais))].
- [27] « Les avantages du Cloud Computing»,20 Juin 2015». Available: [<https://www.ionos.fr/digitalguide/serveur/know-how/les-avantages-du-cloud-computing/>]
- [28] «comprendre les principe de Cloud Computing». 2023. Available: [<https://www.lucidchart.com/blog/fr/comprendre-les-principes-du-cloud-computing>]
- [29] « Cloud Computing : les inconvénients et secrets inavouables du nuage»,30 mai 2020». Available: [<https://www.lebigdata.fr/secrets-cloud-inconvenients/>].
- [30] « DAS, NAS, SAN, stockage en blocs, fichiers ou objets : qu'est-ce que ça signifie ?»14mai 2020». Available: [<https://www.nextinact.com/article/67145/das-nas-san-stockage-en-blocs-fichiers-ou-objets-quest-ce-que-ca-signifie>].
- [31] « What Is Cloud Computing? Definition, Benefits, Types, and Trends», 9 février 2022». Available: [<https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/>].
- [32]<https://www.vmware.com/fr/topics/glossary/content/servervirtualization.html#:~:text=La%20virtualisation%20des%20serveurs%2C%20ou,ses%20propres%20syst%C3%A8mes%20d'exploitation>.<https://www.vmware.com/fr/topics/glossary/content/server-virtualization.html#:~:text=La%20virtualisation%20des%20serveurs%2C%20ou,ses%20propres%20syst%C3%A8mes%20d'exploitation>(indisponible).