

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

**Faculté des Mathématiques et de
l'Informatique**

Département d'Informatique

N° : 24



**DOMAINE : Mathématiques
Informatique**

et

FILIERE : Informatique

OPTION : RTIC

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par: AKKA ALI BEY / BOUREZG ABDELMADJID

Intitulé

**Ordonnancement des tâches dans
Le cloud computing**

Soutenu devant le jury composé de :

Attir Azzedine

Université de M'sila

Président

Chalabi Baya

Université de M'sila

Rapporteur

Guesmia Salah

Université de M'sila

Examineur

Année universitaire: 2022 / 2023

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ، هَذَا مِنْ فَضْلِ رَبِّي

Table des matières

INTRODUCTION GÉNÉRALE.....	14
GÉNÉRALITÉ SUR LE CLOUD COMPUTING	
I.1. Aperçu.....	16
I.2. Définition générale du Cloud computing.....	16
I.3. Évolution du Cloud Computing.....	17
I.4. Notions de base.....	18
I.4.1. Le centre des données « Datacenter ».....	18
I.4.2. Une machine virtuelle « VM ».....	18
I.4.3. Une tâche « Task ».....	19
I.4.5. Le hôte « Host ».....	19
I.5. L'architecture de référence de déploiement du Cloud computing.....	19
I.5.1. Fournisseur de cloud « Provider ».....	20
I.5.2. Opérateur de cloud « Carrier ».....	20
I.5.3. Auditeur de cloud « Auditor ».....	20
I.5.4. Consommateur de cloud « Consumer ».....	20
I.5.5. Courtier du cloud « Broker ».....	20
I.6. Les modèles de déploiement du cloud computing.....	21
I.6.1. Cloud public.....	22
I.6.2. Cloud privé.....	23
I.6.3. Cloud hybride.....	24
I.6.4. Cloud communautaire.....	25
I.7. Les modèles de services du cloud computing.....	26
I.7.1. IaaS « Infrastructure as a Services ».....	26
I.7.1.1. Les caractéristiques de Infrastructure as a Services «IaaS».....	27
I.7.2. PaaS « Platform as a Services ».....	27
I.7.2.1. Les caractéristiques de Platform as a Services «PaaS».....	28
I.7.3. SaaS « Software as a Services»	28
I.7.3.1. Les caractéristiques de Software as a Services « SaaS ».....	28
I.8. Les caractéristiques du cloud computing.....	29
I.9. Accord de niveau de service «Service Level Agreement».....	30
I.10. Les technologies émergente du cloud computing.....	31
I.11. Architecture du cloud computing.....	32
I.12. Les composants de l'infrastructure du cloud computing.....	33
I.13. La virtualisation dans le cloud computing.....	35
I.13.1. Définition de la virtualisation.....	35
I.13.2. Hyperviseur.....	36
I.13.2.1. Définition de l'Hyperviseur.....	36
I.13.2.2 . Types d'Hyperviseur.....	36
I.13.2.2.1 Hyperviseur de type 1 ou "bare-metal".....	37
I.13.2.2.2 Hyperviseur de type 2 ou "hosted".....	37
I.13.3. Techniques de la virtualisation.....	38
I.13.3.1 La virtualisation complète.....	38
I.13.3.2 La virtualisation par émulation.....	39
I.13.3.3 La virtualisation para-virtualisation.....	40
I.13.4. Les formes de virtualisation.....	41
I.14. Avantages et inconvénients du Cloud computing.....	43
I.14.1. Avantages du cloud computing.....	43
I.14.2. Inconvénients du cloud computing.....	43
I.15. Conclusion.....	44

ORDONNANCEMENT DES TÂCHES DANS CLOUD COMPUTING

II.1. Aperçu.....	46
II.2. Ordonnancement des tâches dans cloud computing.....	46
II.3. Notions de bases.....	48
II.3.1. Job.....	48
II.3.2. Cloudlet.....	48
II.3.3. Ressources.....	48
II.4. Les contraintes de l'ordonnancement des tâches.....	49
II.5. Le rôle de l'ordonnanceur dans le cloud computing.....	49
II.6. Types d'ordonnancement dans Cloud computing.....	50
II.7. Les Algorithmes d'ordonnancement des tâches.....	51
II.7.1. Types de catégories d'algorithmes d'ordonnancement des tâches.....	51
II.7.2. Définitions des catégories d'algorithmes d'ordonnancement des tâches.....	52
II.7.3. Les principaux algorithmes d'ordonnancement.....	53
II.7.3.1. First-Come-First-Served.....	53
II.7.3.2. Shortest-Job-First.....	53
II.7.3.3. Round Robin (RR).....	54
II.7.3.4. Min-min.....	54
II.7.3.5. Max-min.....	55
II.8. Conclusion.....	55

CONCEPTION ET ANALYSE

III.1. Aperçu.....	57
III.2. Introduction.....	57
III.3. Conception du travail.....	58
III.3.1. La création des machines virtuelles dans le Datacenter.....	58
III.3.1.1 Premier cas.....	58
III.3.1.2 Deuxième cas.....	60
III.3.2. Soumission des cloudlets.....	61
III.3.5. Cycle de vie d'un cloudlet.....	67
III.3.6. Diagramme de séquence.....	68
III.4. Modèle proposé.....	69
III.4.1. Introduction.....	69
III.4.2. Classification des machines virtuelles.....	69
III.4.2.1. Algorithme de classification des machines virtuelles.....	70
III.4.3. Classification des cloudlets.....	72
III.4.3.1 Algorithme de classification des cloudlets.....	73
III.4.4. Pseudocodes des méthodes de classification.....	75
III.4.4.1 Machines virtuelles.....	75
III.4.4.2 Cloudlets.....	76
III.5. Conclusion.....	76

IMPLÉMENTATION ET RÉALISATION

IV.1. Aperçu.....	78
IV.2. Introduction.....	78
IV.3. Simulation de Cloud computing.....	79
IV.4. Les outils de développement.....	80
IV.4.1. Présentation du langage de programmation (JAVA).....	80
IV.4.2. Présentation de l'environnement de développement IDE ECLIPSE.....	80
IV.4.2.1. Avantages de l'environnement de développement IDE ECLIPSE.....	80

IV.4.3. Présentation de l'outils de simulation Cloudsim.....	81
IV.4.3.1. Les Avantages d'utilisation de Cloudsim.....	81
IV.4.3.2. Architecture de Cloudsim.....	82
IV.4.3.3 La structure des packages et les classes de Cloudsim.....	83
IV.4.3.4. Package "org.cloudbus.cloudsim".....	83
IV.4.3.5. Package "org.cloudbus.cloudsim.core".....	83
IV.4.3.6. Package "org.cloudbus.cloudsim.schedulers".....	84
IV.4.4. Scénario basique de simulation.....	85
IV.4.5. Différentes étapes du cycle de vie de CloudSim.....	86
IV.5.Réalisation et tests.....	87
IV.5.1. Outils de travail.....	87
IV.5.2. Paramètres de simulation.....	87
IV.5.2.1. Paramètres de simulation des machines virtuelles.....	87
IV.5.2.2. Paramètres des PEs.....	87
IV.5.2.3. Paramètres de simulation des cloudlets.....	88
IV.5.2.4. Paramètres des hôtes.....	88
IV.5.3. Résultats de la simulation.....	89
IV.6 Conclusion.....	93

Index des figures et des diagrammes

I.1 Figure de diverses technologies propres à l'ère de l'information [2].....	14
I.2 Figure de architecture de référence de déploiement du Cloud computing [4].....	16
I.3 Figure des quatre modèles de déploiement du cloud.....	18
I.4 Figure de modèle de déploiement d'un cloud public.....	19
I.5 Figure de modèle de déploiement d'un cloud privé.....	20
I.6 Figure de modèle de déploiement d'un cloud hybride.....	21
I.7 Figure de modèle de déploiement d'un cloud communautaire.....	22
I.8 Figure de les différentes couches des services du cloud computing. [9].....	23
I.9 Figure des caractéristiques du cloud computing.....	26
I.10 Figure des divers technologies émergente du cloud computing.....	28
I.11 Figure de L'architecture du cloud computing.....	29
I.12 Figure des composants de l'infrastructure du cloud computing.....	30
I.13 Figure de différence entre une architecture traditionnelle et une architecture virtuelle.....	32
I.14 Figure de fonctionnement théorique de l'hyperviseur.....	33
I.15 Figure de représentation de l'Hyperviseur de type 1.....	34
I.16 Figure de représentation de l'Hyperviseur de type 2.....	35
I.17 Figure de représentation de la virtualisation complète.....	36
I.18 Figure de représentation de la virtualisation par émulation.....	37
I.19 Figure de représentation de la virtualisation para-virtualisation.....	38
I.20 Figure de représentation montrant des formes de virtualisation en cours d'utilisation dans une infrastructure de virtualisation.....	39
II.21 Figure de modèle d'ordonnancement de tâches dans le cloud computing.....	44
II.22 Diagramme de divers catégories d'algorithmes d'ordonnancement [32].....	48
III.23 Diagramme de création des machines virtuelles dans le datacenter.....	55
III.24 Diagramme de création des machines virtuelles dans le Datacenter "CAS 2".....	57
III.25 Diagramme de soumission des cloudlets.....	59
III.26 Diagramme de processus de mise à jour des machines virtuelles.....	61
III.27 Diagramme de processus de retour des résultats.....	63
III.28 Diagramme de cycle de vie d'un cloudlet.....	64
III.29 Diagramme de séquence.....	65
III.30 Diagramme de l'algorithme de classification des machines virtuelles (VM).....	68
III.31 Diagramme de l'algorithme de classification des cloudlets.....	70
IV.32 Diagramme de divers types de simulateurs dans cloud computing [37].....	76
IV.33 Figure de L'architecture de simulateur Cloudsim [39].....	79
IV.34 Diagramme des principales classes dans Cloudsim [41].....	81
IV.35 Diagramme de scénario principal de la simulation CloudSim.....	82
IV.36 Figure des différentes étapes du cycle de vie de CloudSim.....	83
IV.37 Figure des résultats de la simulation.....	87
IV.38 Figure des résultats de la simulation (Temps d'exécution / Cloudlets).....	87
IV.39 Figure de capture d'écran de quelques tests (1).....	88
IV.40 Figure de capture d'écran de quelques tests (2).....	89
IV.41 Figure de capture d'écran de quelques tests (3).....	89

Index des tableaux

III Pseudocodes de la méthode des VM.....	72
III Pseudocodes de la méthode des Cloudlets.....	73
IV Tableau des paramètres de simulation des machines virtuelles.....	84
IV Tableau des paramètres des PEs.....	84
IV Tableau des paramètres de simulation des cloudlets.....	85
IV Tableau des paramètres des hôtes.....	85
IV Résultats de la simulation entre différents algorithmes.....	86

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

Le Cloud Computing, également appelé informatique en nuage, est une technologie qui permet d'accéder à distance à des ressources informatiques via Internet, sans avoir à les posséder localement. Les services de Cloud Computing sont proposés par des fournisseurs tels qu'Amazon, Microsoft, Google et IBM.

Le Cloud Computing offre une multitude d'avantages aux utilisateurs en leur permettant d'accéder à des services informatiques à la demande. Il peut être utilisé pour héberger des applications, stocker des données et fournir des services de traitement tout cela en fonction de leurs besoins, sans avoir à investir dans des infrastructures coûteuses ou à gérer des serveurs localement.

Cependant, le Cloud Computing soulève également des préoccupations en matière de confidentialité ainsi que de disponibilité des services en cas de pannes. Ces problèmes doivent être pris en compte lors de la mise en place d'une infrastructure de Cloud Computing.

Dans ce travail, nous proposons un algorithme utilisant des méthodes dans le but d'optimiser le fonctionnement d'ordonnancement des tâches qui est de faire classification à travers des plusieurs méthodes pour les machines virtuelles et les cloudlets afin de minimiser le temps d'exécution des tâches.

Ce mémoire est divisé en quatre principaux chapitres. Dans le premier chapitre intitulé Généralités sur le Cloud Computing, nous présentons le Cloud Computing, son évolution, ses caractéristiques, ses modèles de déploiement et de services ainsi que la virtualisation.

Ensuite, nous abordons dans le deuxième chapitre l'ordonnancement des tâches dans le Cloud en définissant le concept et les algorithmes utilisés dans ce domaine.

Dans le troisième chapitre intitulé Analyses et conception, nous effectuons une analyse précise de notre algorithme.

Enfin, nous concluons avec un dernier chapitre qui résume la réalisation de notre travail, en incluant des tests de comparaison dans le but de minimiser le temps d'exécution.

CHAPITRE I

GÉNÉRALITÉS SUR LE CLOUD COMPUTING

I.1. Aperçu

Le Cloud Computing est un domaine de l'informatique en constante évolution, qui a transformé la manière dont les entreprises gèrent leurs données et leurs services informatiques. Ce chapitre se concentre sur l'évaluation pertinente concernant le Cloud Computing et se concentre sur deux domaines clés : la technologie du Cloud Computing et la virtualisation.

La première section de ce chapitre fournit une définition détaillée du Cloud Computing, en mettant en évidence les caractéristiques clés telles que les modèles de déploiement, l'architecture et les différents services offerts.

Aussi la technologie de virtualisation qui est le fondement du Cloud Computing et permet de créer des environnements informatiques virtuels, ce qui rend la gestion des ressources plus efficace et économique. Dans cette section, nous examinons en détail les différentes approches de la virtualisation, ainsi que les types d'hyperviseurs utilisés dans le Cloud Computing.

I.2. Définition générale du Cloud computing

Le cloud computing, également connu sous le nom d'informatique en nuage, est un modèle de prestation de services informatiques via Internet. Il permet aux utilisateurs d'accéder à des ressources informatiques, telles que des serveurs, des applications et des bases de données, à la demande, sans avoir à les gérer eux-mêmes.

Les services de cloud computing sont généralement proposés par des fournisseurs tiers qui gèrent l'infrastructure, les logiciels et les mises à jour, offrant ainsi aux utilisateurs une solution clé en main. Cette approche permet aux utilisateurs de se concentrer pleinement sur leur activité principale, sans se soucier des aspects techniques de l'infrastructure informatique.

Le cloud computing offre une grande flexibilité aux utilisateurs, leur permettant d'ajuster rapidement et facilement leurs ressources informatiques en fonction de leurs besoins. Les services de cloud computing sont souvent facturés à l'utilisation, ce qui permet aux entreprises de maîtriser leurs coûts et d'éviter les investissements initiaux importants liés à l'acquisition et à la gestion de leur propre infrastructure informatique [1].

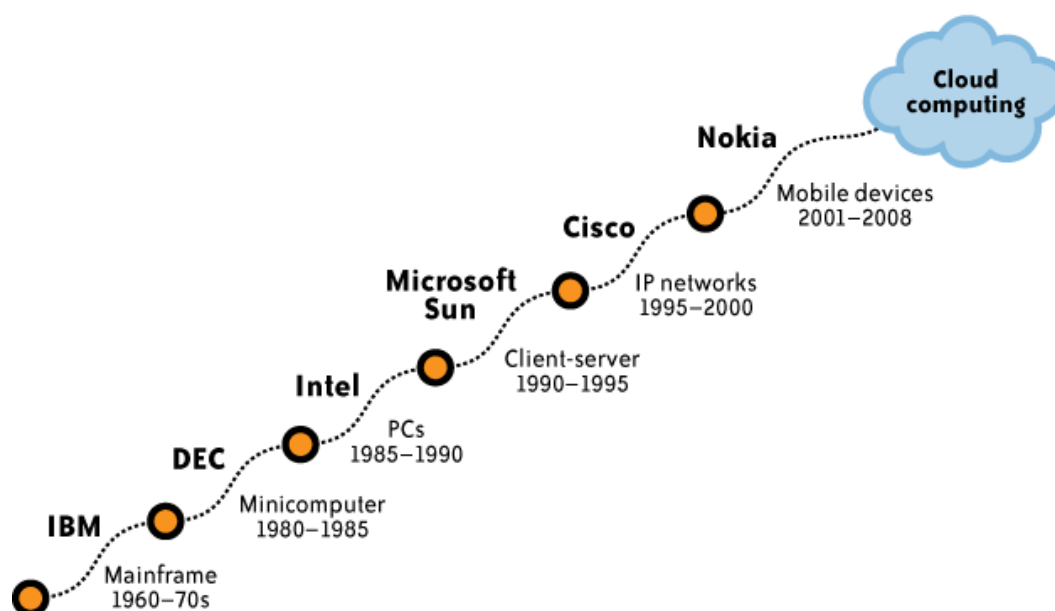
I.3. Évolution du Cloud Computing

Le cloud computing a évolué au fil des ans pour répondre aux besoins changeants des entreprises et des utilisateurs. Au début, le cloud computing était principalement utilisé pour fournir de l'espace de stockage en ligne les utilisateurs pouvaient accéder à ces services à partir de leurs terminaux via des connexions Internet, Au cours des années 2000, le cloud computing s'est développé pour inclure des services d'infrastructure tels que les serveurs virtuels et les réseaux cela a permis aux entreprises de réduire considérablement leurs coûts informatiques [8].

Au fil du temps il s'est étendu pour inclure des services plus sophistiqués tels que l'analyse de données, l'intelligence artificielle, la réalité virtuelle et augmentée, et l'Internet des objets.

Le cloud computing a également permis aux entreprises de réduire leurs coûts et d'améliorer leur efficacité en leur permettant de se concentrer sur leurs activités principales, plutôt que de gérer des infrastructures informatiques coûteuses et complexes en interne [2].

Aujourd'hui, le cloud computing est devenu un élément clé de l'infrastructure informatique de nombreuses entreprises, la figure présentée ci-dessous offre une illustration des diverses technologies propres à l'ère de l'information :



I.1 Figure de diverses technologies propres à l'ère de l'information [2]

I.4. Notions de base

Le cloud computing englobe plusieurs notions de base qui sont importantes à comprendre pour tirer le meilleur parti de cette technologie, voici quelques éléments de base qui nous allons utiliser dans notre travail :

I.4.1. Le centre des données « Datacenter »

Le centre des données aussi appelé Datacenter, ce dernier est une installation physique où sont stockées et gérées les données et les applications informatiques des clients du service cloud.

Le datacenter joue un rôle clé dans le traitement des requêtes des clients du service Cloud. Il utilise ses propres machines virtuelles pour effectuer les tâches demandées de manière séquentielle ou parallèle, tout en assurant le partage efficace des ressources [3]. Les DataCenters sont souvent situés dans des endroits stratégiques pour assurer une connectivité réseau fiable et une disponibilité continue des services cloud. Ils sont conçus pour offrir une haute autonomie, une sécurité robuste, une redondance des systèmes et des performances optimales.

I.4.2. Une machine virtuelle « VM »

Une machine virtuelle est une entité logicielle qui s'exécute sur une infrastructure matérielle physique et partage ses ressources avec d'autres machines virtuelles. Elle est créée en utilisant une technologie de virtualisation qui permet d'exécuter plusieurs instances d'un système informatique sur un seul ordinateur physique.

Une machine virtuelle est considérée comme une réplique d'un ordinateur physique, avec ses propres ressources telles que le processeur, la mémoire, le stockage et les interfaces réseau. Elle peut être créée, modifiée, sauvegardée et supprimée rapidement et facilement en fonction des besoins de l'utilisateur, sans nécessiter de matériel supplémentaire [14].

Leurs utilisation est primordial pour l'exécution des applications et des systèmes d'exploitation isolés les uns des autres, ce qui permet de maximiser l'utilisation des ressources des datacenter et d'améliorer la sécurité et la fiabilité.

I.4.3. Une tâche « Task »

Une tâche fait référence à une unité de travail à exécuter par le Datacenter. celle-ci peut inclure une ou plusieurs étapes de traitement de données ou de calcul, et peut être exécutée sur une machine virtuelle. Elle est définie par diverses caractéristiques, notamment sa taille, sa durée d'exécution, ses entrées et ses sorties. Les tâches peuvent être de simples opérations de calcul à des traitements plus complexes nécessitant une grande quantité de données et des ressources de calcul importantes.

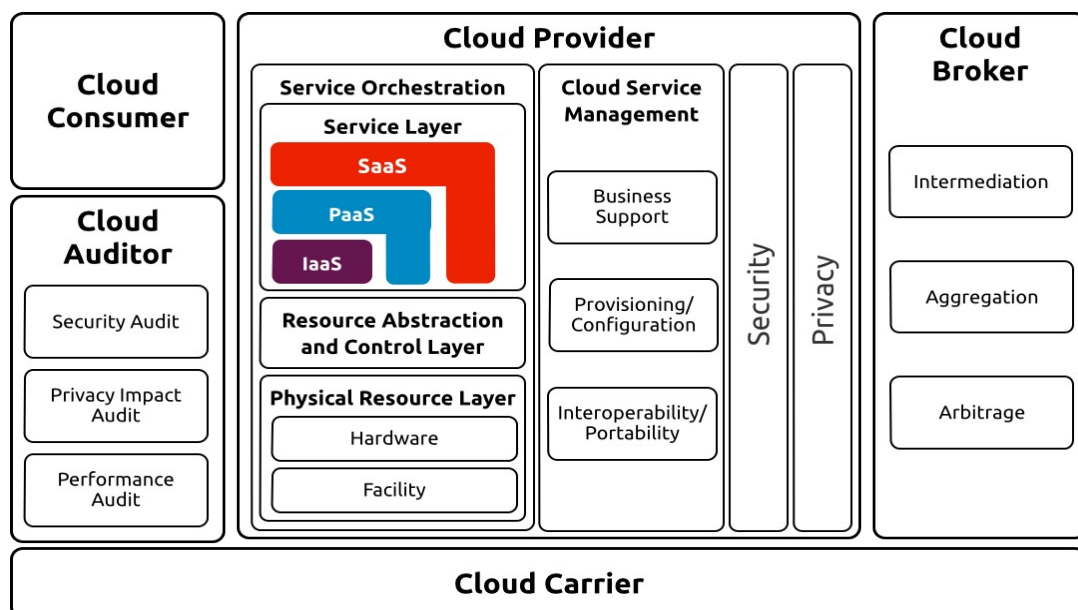
I.4.5. Le hôte « Host »

Le hôte est un serveur physique qui héberge plusieurs machines virtuelles. Il est responsable de la gestion de ces machines virtuelles, notamment de leur allocation de ressources telles que de la puissance de calcul, de la mémoire, de l'espace de stockage et des interfaces réseau pour héberger des applications et des services en ligne.

Les hôtes sont conçus pour fonctionner de manière autonome, de sorte qu'ils peuvent continuer à exécuter les machines virtuelles même en cas de défaillance d'autres composants du Datacenter.

I.5. L'architecture de référence de déploiement du Cloud computing

L'architecture de Cloud computing est constituée de cinq éléments indépendants mais interconnectés, qui travaillent de manière coordonnée pour offrir des services à la demande. Leur collaboration permet de répondre aux besoins des utilisateurs en termes de services Cloud



I.2 Figure de architecture de référence de déploiement du Cloud computing [4]

I.5.1. Fournisseur de cloud « Provider »

Une organisation qui fournit des services de cloud computing est appelée un fournisseur de cloud. Ce dernier a une gestion complète de l'infrastructure informatique et est responsable de la gestion des interruptions techniques, qu'elles soient prévues ou non [4].

Le fournisseur de cloud s'assure également que les niveaux de service convenus sont atteints, en respectant les accords de niveau de service « SLA » [5].

I.5.2. Opérateur de cloud « Carrier »

Un opérateur de Cloud est une entreprise qui joue un rôle d'intermédiaire entre les fournisseurs de services Cloud et les utilisateurs finaux, [4] Ils sont chargés de fournir une connectivité et un accès au cloud aux utilisateurs finaux via les réseaux de télécommunication et divers dispositifs d'accès [5].

I.5.3. Auditeur de cloud « Auditor »

Le rôle de l'auditeur cloud est d'effectuer une évaluation indépendante des services cloud offerts par les fournisseurs de cloud. L'auditeur peut effectuer des contrôles tels que la sécurité, la confidentialité, les performances, les licences, les opérations afin de détecter d'éventuels écarts par rapport aux normes opérationnelles et de protection des données [5].

I.5.4. Consommateur de cloud « Consumer »

Le consommateur Cloud est l'acteur clé de l'utilisation du cloud computing. Il peut s'agir d'une personne ou d'une organisation ayant établi une relation commerciale [5] avec un fournisseur de services cloud et utilisant ses services .

I.5.5. Courtier du cloud « Broker »

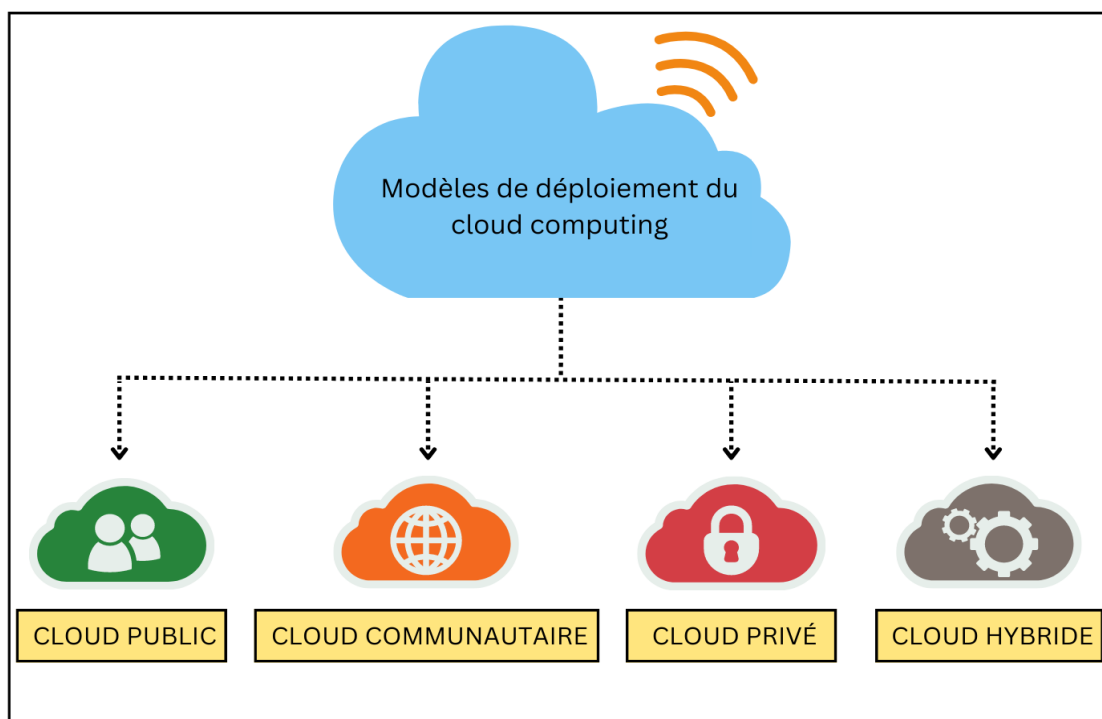
Une organisation appelée « Broker » contrôle la manière dont le cloud est utilisés, exécutés et fournis et négocie les contrats [5] entre les fournisseurs et les utilisateurs de cloud, le broker offre un point d'accès unique pour contrôler de nombreux services dans le cloud ainsi que faire l'orchestration des ressources.

I.6. Les modèles de déploiement du cloud computing

Les modèles de déploiement du cloud computing font référence aux différents types d'architectures et d'environnements dans lesquels les services de cloud computing sont proposés et mis en œuvre.

Il existe quatre modèles de déploiement du cloud, chacun offrant des avantages et des inconvénients en termes de coûts, de sécurité, de contrôle et de flexibilité [6].

- Le cloud public.
- Le cloud privé
- Le cloud hybride
- Le cloud communautaire



I.3 Figure des quatre modèles de déploiement du cloud

Ces modèles de déploiement diffèrent en fonction du type d'implémentation, du type d'hébergement et des personnes qui y ont accès. Tous les modèles de déploiement Cloud sont basés sur le même principe de virtualisation [3] mais diffèrent en termes de :

- Emplacement
- Capacité de stockage,
- Accessibilité

Il faut donc connaître les caractéristiques de chaque environnement.

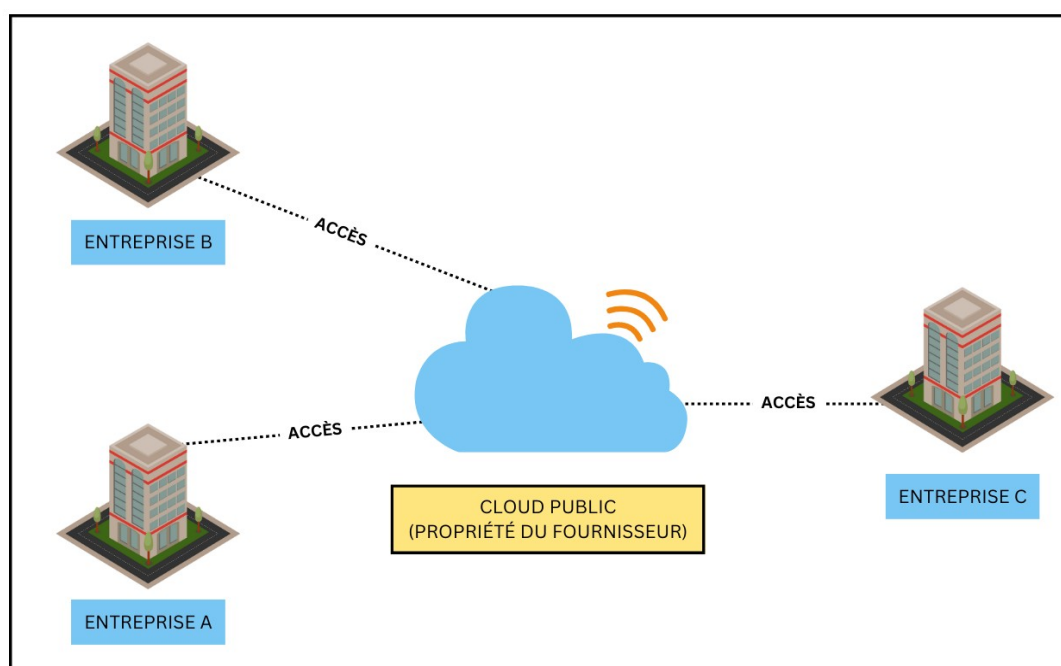
I.6.1. Cloud public

Le cloud public est un modèle de déploiement dans lequel les fournisseurs de services cloud proposent leurs ressources informatiques « tels que les serveurs, le stockage, les bases de données, les applications » aux clients via Internet.

Dans ce modèle, les ressources du fournisseur de cloud sont partagées entre plusieurs clients [1], ce qui permet une utilisation plus efficace des ressources et réduit les coûts pour chaque client.

Les entreprises qui optent pour le cloud public sont souvent des petites entreprises ou des start-ups qui n'ont pas les ressources nécessaires pour construire leur propre infrastructure informatique.

Les avantages du cloud public incluent la flexibilité, la scalabilité et les coûts réduits, ce qui permet aux entreprises de se concentrer sur leur activité principale plutôt que sur la gestion de leur infrastructure informatique [8].



I.4 Figure de modèle de déploiement d'un cloud public.

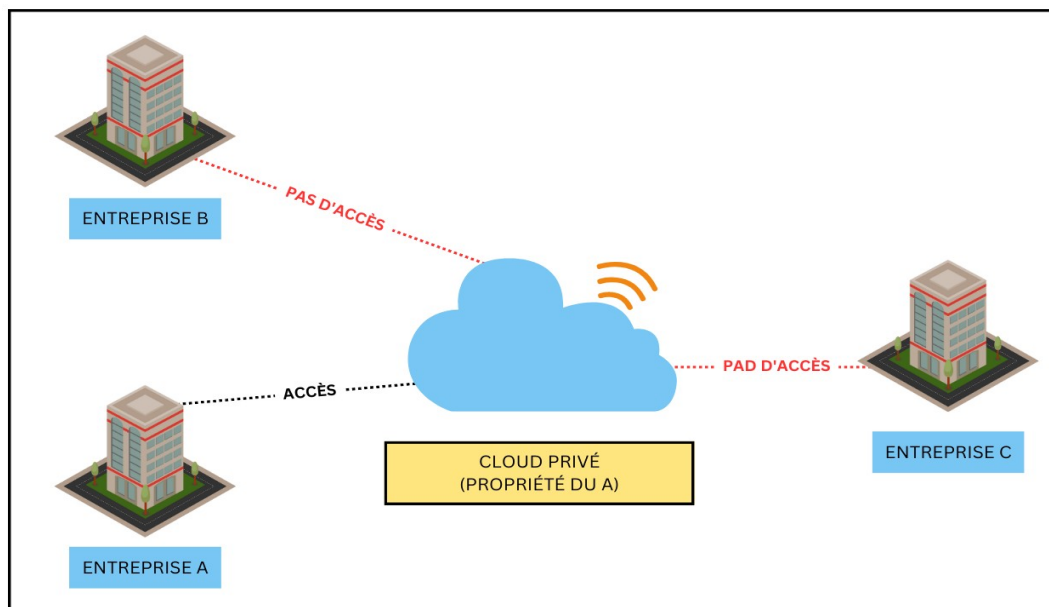
Cependant, le cloud public peut présenter des inconvénients, notamment en termes de sécurité, de manque de contrôle sur l'infrastructure, et de disponibilité limitée des ressources [2].

I.6.2. Cloud privé

Le cloud privé est un modèle de déploiement dans lequel une entreprise ou une organisation construit et exploite son propre environnement de cloud computing pour une utilisation interne.

Cela signifie que les ressources informatiques telles que les serveurs, le stockage, les bases de données et les applications ne sont pas partagées.

Les avantages du cloud privé incluent un niveau de sécurité plus élevé et un plus grand contrôle sur l'infrastructure [6].



I.5 Figure de modèle de déploiement d'un cloud privé

Néanmoins, la mise en place d'un cloud privé peut entraîner des coûts élevés, et son infrastructure peut être moins flexible et moins évolutive [2] que celle d'un cloud public.

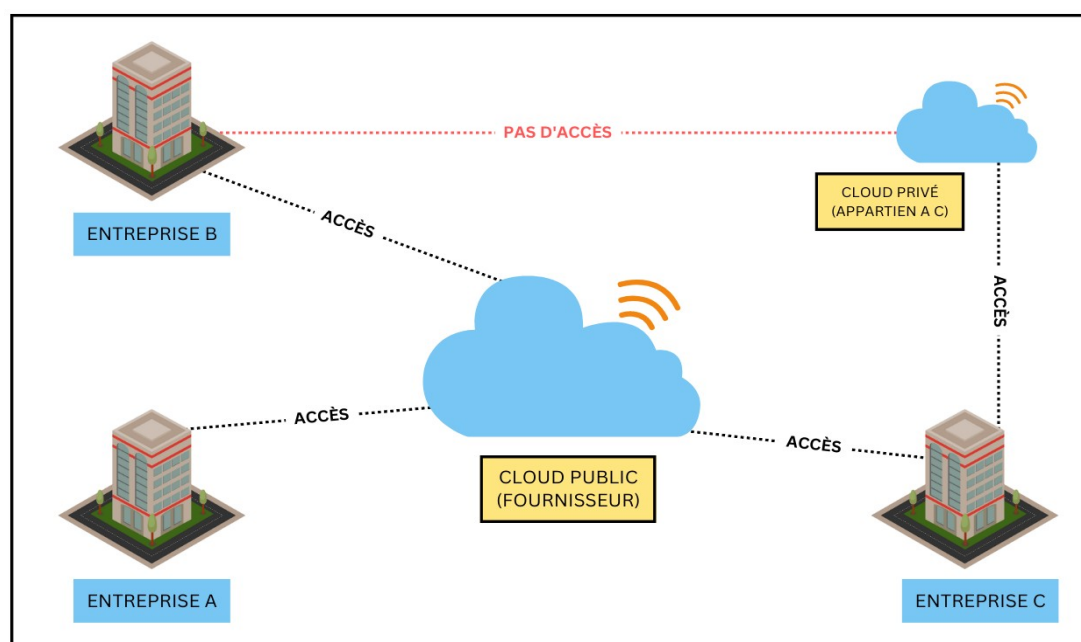
I.6.3. Cloud hybride

Le cloud hybride est un autre modèle de déploiement qui combine [6] à la fois des services de

- Cloud public
- Cloud privé

Tout cela pour permettre aux entreprises de tirer parti des avantages des deux modèles de déploiement.

Les entreprises peuvent ainsi déployer des applications critiques sur un cloud privé, tandis que les applications moins critiques peuvent être hébergées sur un cloud public.



I.6 Figure de modèle de déploiement d'un cloud hybride

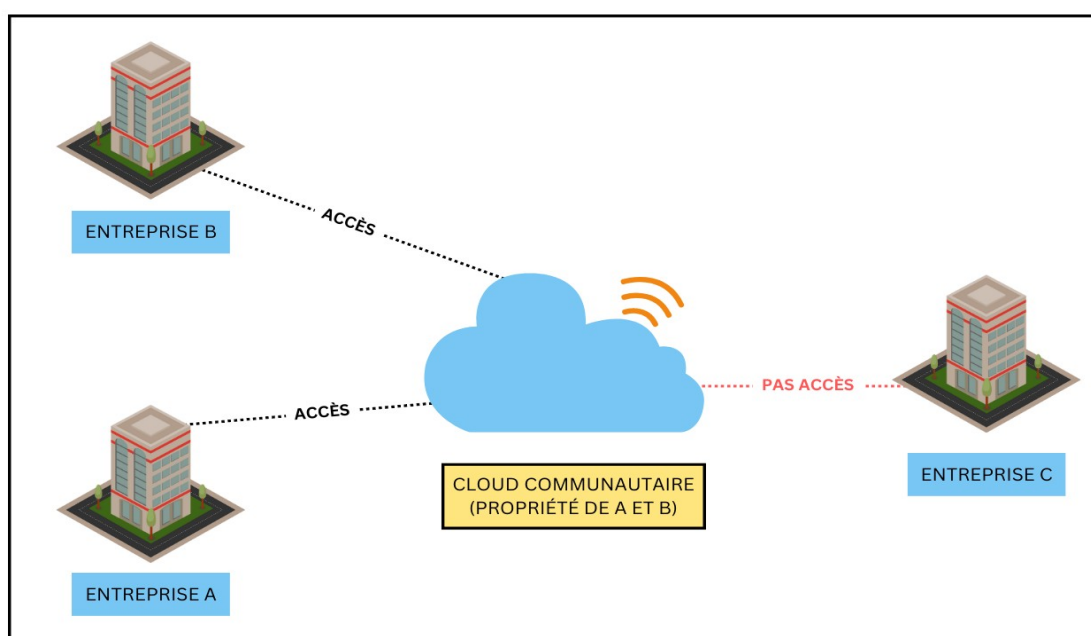
Le cloud hybride offre une flexibilité et une scalabilité ainsi qu'une réduction des coûts. Les entreprises peuvent conserver le contrôle de leurs données sensibles et de leur infrastructure tout en profitant des avantages du cloud public[8].

Cependant, les désavantages résident dans la complexité [2] et la nécessité de gérer deux environnements de cloud distincts de manière simultanée.

I.6.4. Cloud communautaire

Le cloud communautaire est un modèle de déploiement dans lequel les ressources informatiques sont partagées par un groupe d'utilisateurs [6] ayant des besoins et des objectifs communs.

Ce modèle de déploiement permet aux membres de la communauté de partager leurs ressources informatiques, telles que les serveurs, le stockage, les applications afin de répondre aux besoins spécifiques de la communauté, tout en offrant une sécurité par rapport au cloud public.



I.7 Figure de modèle de déploiement d'un cloud communautaire

Le cloud communautaire est souvent utilisé par les organisations qui travaillent avec des données sensibles.

Les avantages du cloud communautaire sont la flexibilité, la scalabilité, les coûts partagés et la sécurité, aussi la possibilité de partager les ressources informatiques avec d'autres organisations ayant des intérêts communs [8].

Les inconvénients sont la complexité et l'accès à distance peut être limité ainsi que les coûts de maintenances constantes [2].

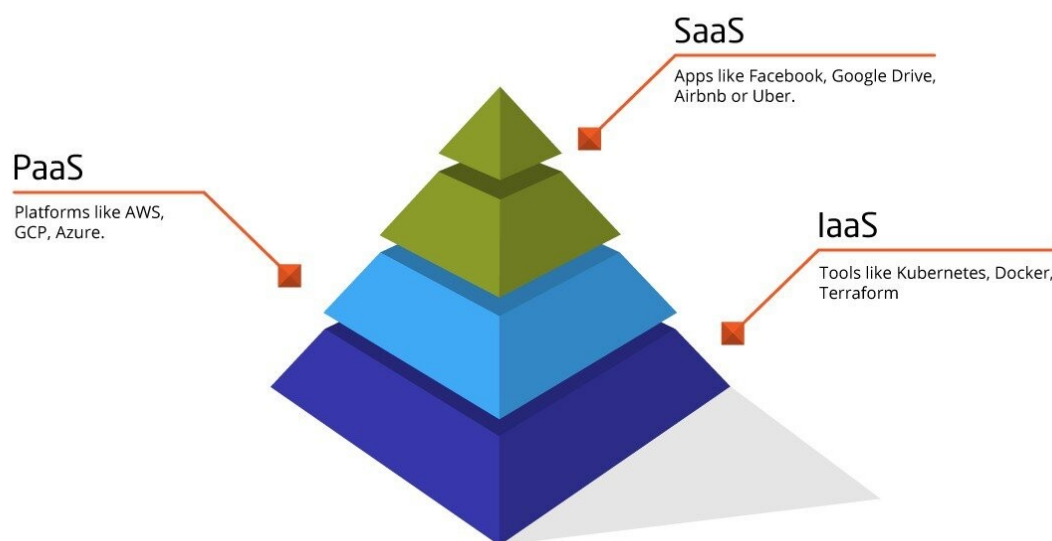
I.7. Les modèles de services du cloud computing

Les modèles de services sont des modèles de références sur lesquels le Cloud computing est basé, un modèle de service représente les modèles de références sur lesquels le Cloud computing est basé.

Ces services sont reliés entre eux de manière à ce que les services les plus bas servent de base sur laquelle les autres services sont fondés [9]. Ils sont catégorisés en trois modèles de services de base.

- Infrastructure en tant que service «IaaS»
- Plate-forme en tant que service «PaaS»
- Logiciel en tant que service «SaaS»

Ci-dessous est une représentation graphique des différentes couches de services proposés par les fournisseurs de cloud computing. Elle est souvent exhiber en forme de pyramide :



I.8 Figure de les différentes couches des services du cloud computing. [9]

I.7.1. IaaS « Infrastructure as a Services »

IaaS est un modèle de cloud computing dans lequel un fournisseur de services cloud fournit une infrastructure informatique à la demande, telle que des serveurs, des réseaux, des stockages et des systèmes d'exploitation.

Les clients peuvent alors utiliser cette infrastructure pour exécuter leurs propres applications et services. Dans ce type de service, le consommateur peut acquérir de la puissance de calcul, des capacités de stockage, des réseaux et d'autres ressources informatiques qu'il peut exploiter et ne payant que pour ce qu'ils utilisent réellement.

IaaS permet une grande flexibilité pour les entreprises qui cherchent à augmenter ou réduire rapidement leurs capacités informatiques en fonction de la demande.

I.7.1.1. Les caractéristiques de Infrastructure as a Services «IaaS»

Les principales caractéristiques de l'Infrastructure en tant que service «IaaS» sont les suivantes :

- **Évolutivité et flexibilité:** Ce service permet aux entreprises de faire évoluer leur infrastructure en fonction de leurs besoins en ressources informatiques tel que la capacités de stockage, de calcul et de réseau flexibles qui peuvent être adaptées en fonction de la demande.
- **Réduction des coûts:** l'IaaS permet aux entreprises de réduire leurs coûts budgétaire en évitant d'investir dans une infrastructure réel qui est coûteuses et en bénéficiant d'une gestion centralisée [5].
- **Accès à distance:** Les ressources informatiques sont accessibles à distance via Internet, ce qui permet une utilisation efficace des ressources [5] et une meilleure collaboration entre les membres de l'entreprise.

I.7.2. PaaS « Platform as a Services »

Le PaaS est un autre modèle dans lequel un fournisseur de services cloud fournit une plateforme complète pour les développeurs, y compris l'infrastructure informatique nécessaire tel que

- Outils de développement.
- Bases de données et des serveurs.
- Langages de programmation et des bibliothèques de code.

Les fournisseurs de services PaaS fournissent une plateforme flexible, évolutive et sécurisée sur laquelle les développeurs peuvent créer et déployer des applications [8].

I.7.2.1. Les caractéristiques de Platform as a Services «PaaS»

Les principales caractéristiques de plateforme en tant que service «PaaS» sont les suivantes :

- **Personnalisation:** Offre une possibilité de personnaliser l'environnement de développement en fonction des besoins spécifiques de l'entreprise ou du projet.
- **Automatisation:** Gestion automatisée de la plateforme par le fournisseur de services PaaS, ce qui permet aux développeurs [5] de se concentrer sur le développement d'applications plutôt que sur la gestion de l'infrastructure.
- **Gestion optimale:** Une gestion efficace des ressources pour optimiser les coûts et la performance des applications, ainsi que la collaboration est simplifiée entre les développeurs, car la plateforme permet de travailler en temps réel sur le même projet.

I.7.3. SaaS « Software as a Services»

SaaS est le dernier modèle dans le quel un fournisseur de services cloud fournit un logiciel ou une application prête à l'emploi via Internet.

Les clients peuvent accéder à l'application via un navigateur Web ou une interface sans avoir besoin d'installer de logiciel sur l'ordinateur local et utiliser les fonctionnalités offertes par l'application.

I.7.3.1. Les caractéristiques de Software as a Services « SaaS »

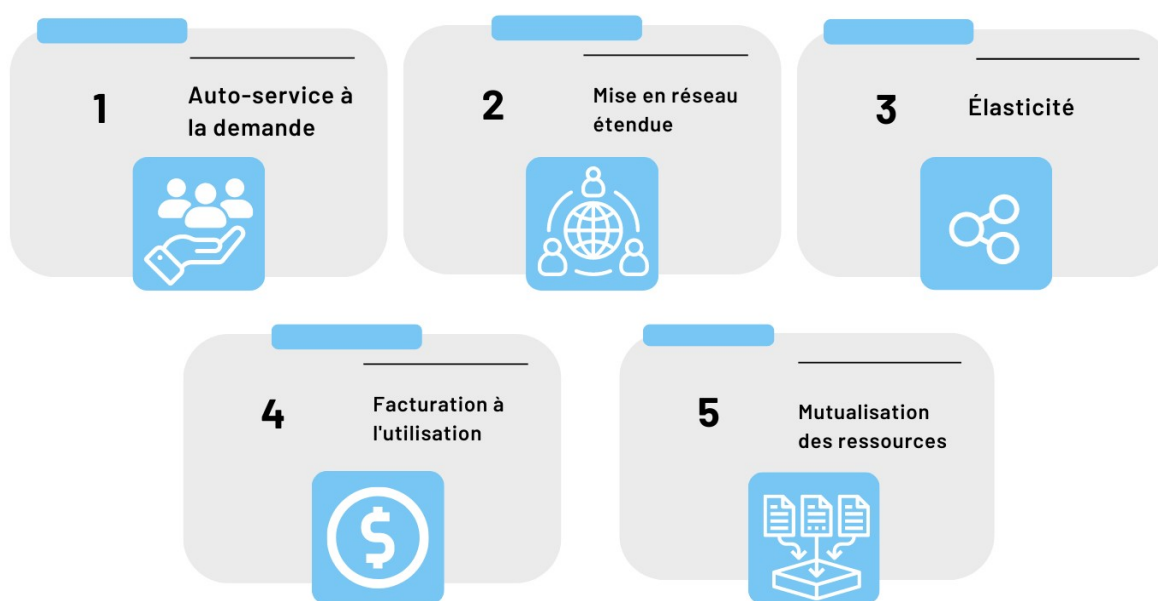
Les principales caractéristiques de logiciel en tant que service «SaaS» sont les suivantes :

- **Accès facile et flexible:** Le SaaS permet aux utilisateurs finaux d'accéder aux applications depuis n'importe où et à tout moment, avec une simple connexion Internet.
- **Maintenance simplifiée :** Les utilisateurs n'ont pas besoin d'installer de logiciel sur leur ordinateur local, ce qui simplifie la maintenance des applications.
- **Mises à jour transparentes :** Le SaaS gère les mises à jour et les mises à niveau des applications, ce qui permet aux utilisateurs de toujours disposer des dernières fonctionnalités sans avoir à se soucier de la mise à jour des logiciels [5].

I.8. Les caractéristiques du cloud computing

Au cours des dernières années, le Cloud computing a complètement révolutionné l'industrie des services informatiques.

Cette technologie a introduit un ensemble de caractéristiques qui ont considérablement amélioré l'utilisation et la fourniture de services informatiques.



I.9 Figure des caractéristiques du cloud computing.

Le cloud computing englobe un ensemble de caractéristiques qui ont grandement amélioré la façon dont les services informatiques sont fournis, ce qui ont poussé son utilisation dans de nombreux secteurs d'activité et qui ont permis aux entreprises et aux particuliers de tirer pleinement parti de ses avantages.

Les principales caractéristiques de cloud computing sont les suivantes :

- L'auto-service à la demande: Les utilisateurs peuvent provisionner et gérer eux-mêmes leurs ressources informatiques en fonction de leurs besoins [14], tout cela sans avoir à passer par un tiers. Les ressources sont accessibles via une interface en libre-service.
- La mise en réseau étendue: Les services cloud sont accessibles depuis n'importe où dans le monde via Internet. Les utilisateurs peuvent accéder aux ressources à partir de différents périphériques et depuis différents emplacements.
- L'élasticité: L'élasticité permet une allocation et une libération des ressources en fonction des besoins de l'utilisateur [10]. Cette flexibilité favorise une utilisation optimale des ressources, car elles peuvent être assignées aux tâches qui en ont besoin.
- La facturation à l'utilisation: Les ressources cloud sont facturées en fonction de l'utilisation réelle. [8] Les utilisateurs ne paient que pour ce qu'ils utilisent, ce qui permet une utilisation plus rentable des ressources.
- La mutualisation des ressources : Les ressources cloud sont partagées entre plusieurs utilisateurs. Cela permet une utilisation plus efficace des ressources, car elles sont allouées là où elles sont nécessaires [8].

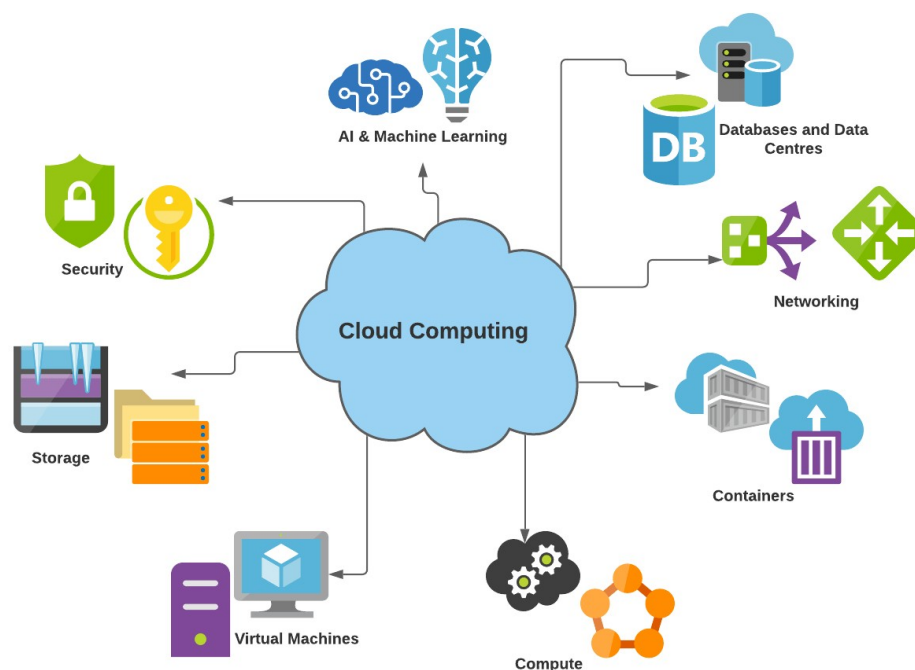
I.9. Accord de niveau de service «Service Level Agreement»

Un Service-Level Agreement « SLA » ou Accord de Niveau de Service en français, est un contrat qui définit les niveaux de service garantis par un fournisseur de services Cloud à ses clients. Le SLA définit les engagements du fournisseur de services en matière de disponibilité, de performance, de temps de réponse, de sécurité, de sauvegarde et de restauration de données, ainsi que de support technique [25].

Le SLA est un élément Important du service Cloud, car il permet aux clients de comprendre les niveaux de service garantis par le fournisseur et de mesurer sa performance par rapport aux engagements pris. Il permet également de définir les recours disponibles en cas de non-respect des engagements.

I.10. Les technologies émergente du cloud computing

Le cloud computing est alimenté par plusieurs technologies émergente qui travaillent de manière coopératif pour offrir une large gamme de services cloud de haute qualité, répondant ainsi aux besoins variés des consommateurs.



I.10 Figure des divers technologies émergente du cloud computing.

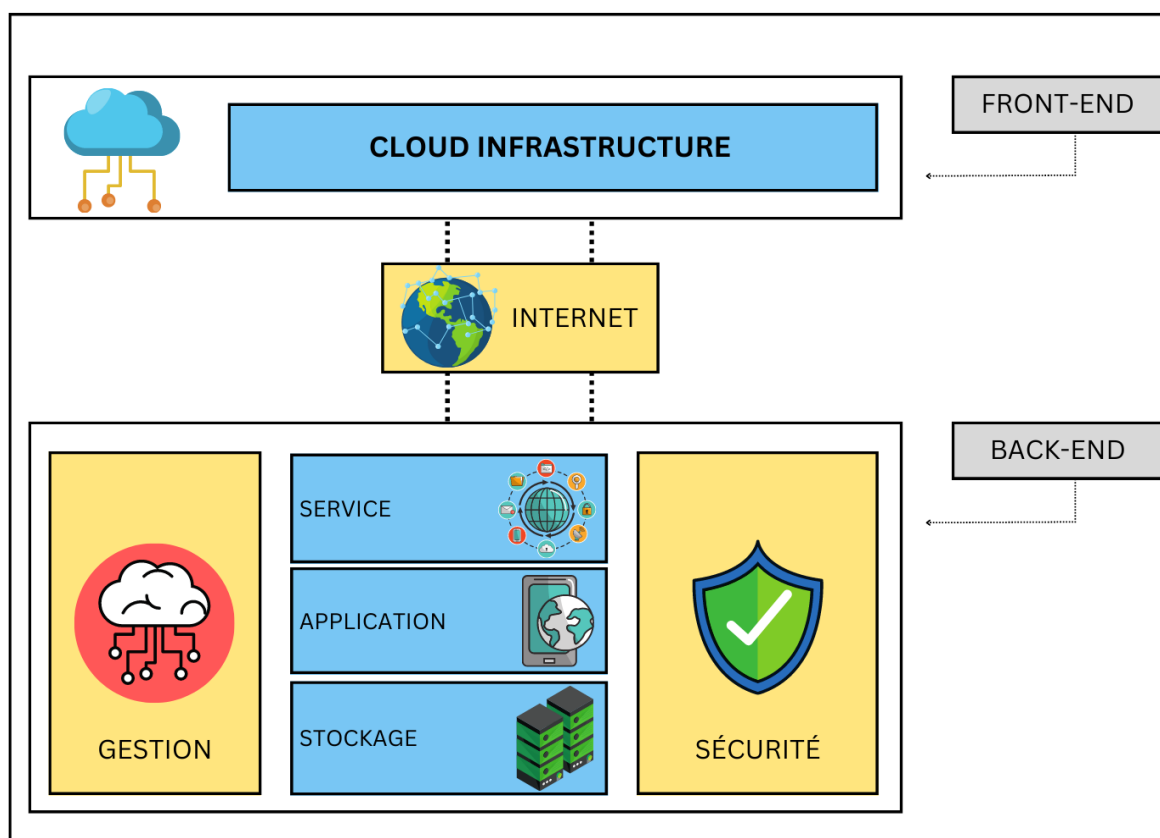
- Intelligence artificielle «IA» et apprentissage automatique «machine learning» : Le cloud computing offre des capacités d'IA et de machine learning qui permettent d'exploiter et d'analyser des données pour obtenir des informations importante et des modèles prédictifs.
- Les conteneurs: Ils permettent d'emballer des applications et leurs dépendances en une seule unité portable, ce qui facilite leur déploiement et leur gestion dans un environnement cloud.
- Edge computing: L'edge computing déplace le traitement des données et des applications plus près des utilisateurs finaux, en les rapprochant des périphériques ou des points de collecte de données. Cela réduit la latence et améliore les performances pour les applications nécessitant une réponse rapide [34]

- Les systèmes de stockage distribué: Ils permettent de stocker des données sur plusieurs serveurs, ce qui augmente la disponibilité et la résilience des données, ainsi que les performances [11].
- Les systèmes de gestion de bases de données Cloud «Database-as-a-Service, DBaaS»: Les DBaaS permettent de fournir des bases de données Cloud gérées, qui peuvent être facilement mises à l'échelle en fonction des besoins [26].
- Les technologies de conteneurisation des applications « Container-as-a-Service, CaaS»: Elles permettent de fournir des plates-formes de développement et de déploiement pour les applications conteneurisées, telles que Kubernetes [27].

I.11. Architecture du cloud computing

L'architecture de cloud computing se divise en deux parties, à savoir le front-end et le back-end.

Le front-end et le back-end communiquent par l'intermédiaire d'un réseau ou de l'internet. Une représentation schématique de l'architecture de l'informatique en nuage est présentée ci-dessous :



I.11 Figure de L'architecture du cloud computing

- Front-end : Le front-end de l'architecture cloud représente la partie avec laquelle les utilisateurs finaux interagissent.

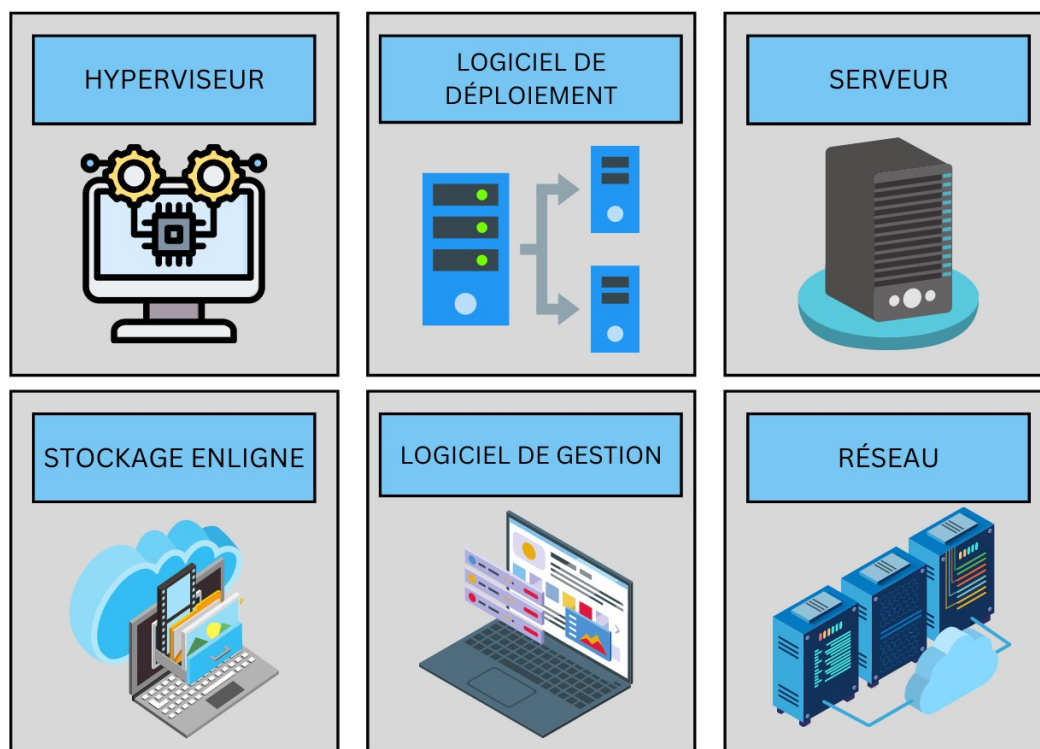
Il comprend l'interface utilisateur graphique « GUI » ou l'interface utilisateur basée sur le web « UI », les applications ou les services accessibles aux utilisateurs finaux, son rôle est de fournir une interface permettant aux utilisateurs d'interagir avec les services cloud [12].

- Back-end : Le back-end de l'architecture cloud fait référence à l'infrastructure et aux composants qui soutiennent les opérations du cloud computing. Il comprend les serveurs, les systèmes de stockage, les bases de données, les logiciels de virtualisation, les réseaux, les systèmes de sécurité, I

Il est responsable de la fourniture des services demandés par le front-end. Il traite les demandes des utilisateurs, exécute les calculs, stocke les données et assure la gestion globale du cloud [12].

I.12. Les composants de l'infrastructure du cloud computing

Le cloud computing est une infrastructure complexe qui comprend plusieurs composants. Ces composants travaillent ensemble pour fournir des services de haute qualité aux consommateurs, ils sont présentés comme ci-dessous :



I.12 Figure des composants de l'infrastructure du cloud computing.

- Hyperviseur: C'est un logiciel qui permet la création et la gestion des machines virtuelles « VM » [13], l'un des objectifs principaux de l'hyperviseur est de diviser et d'allouer les ressources de manière efficace pour permettre une utilisation optimale de l'infrastructure de cloud computing.
- Logiciel de gestion : Est responsable de la gestion et de la surveillance des opérations du cloud en utilisant diverses stratégies pour améliorer les performances [13], Le logiciel de gestion peut effectuer des tâches telles que l'audit de conformité pour s'assurer que les normes de sécurité sont respectées.
- Logiciel de déploiement : Ce composant permet de déployer les applications et les services sur la plateforme de cloud [13]. Il s'occupe de l'installation, de la configuration et du déploiement des applications et des services dans l'environnement cloud.
- Réseau: Le réseau assure la connectivité entre les différents éléments de l'infrastructure. celui-ci permet la transmission de données entre les serveurs, les dispositifs de stockage et les clients via Internet.
- Stockage en ligne : Ceci permet aux utilisateurs de stocker leurs données dans le cloud plutôt que sur un disque dur local.
Le stockage en ligne est géré par les fournisseurs en utilisant des centres de données massifs pour stocker les données des utilisateurs [13]. ces derniers peuvent accéder à leurs données stockées en ligne à partir de n'importe quel endroit et à tout moment, en utilisant une connexion Internet.
- Serveur : Un serveur dans l'infrastructure de cloud computing est un système informatique qui fournit des services à d'autres ordinateurs ou dispositifs connectés via un réseau. Dans le cloud computing, les serveurs sont utilisés pour stocker et traiter les données et les applications.
Les serveurs sont généralement regroupés dans des centres de données où ils sont gérés et surveillés pour assurer leur disponibilité et leur performance.

I.13. La virtualisation dans le cloud computing

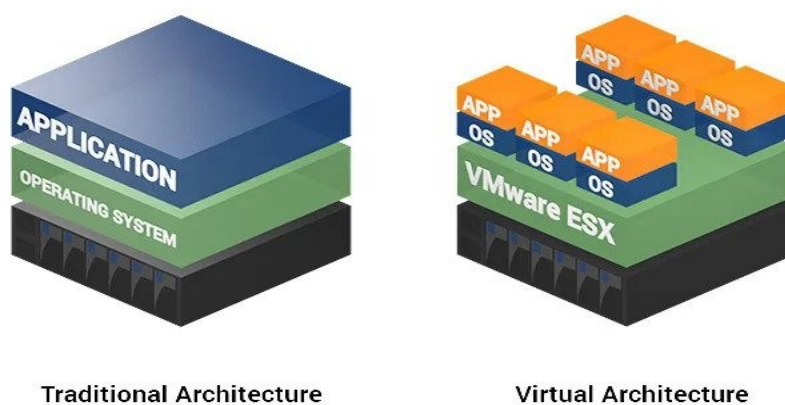
I.13.1. Définition de la virtualisation

La virtualisation est une technique qui permet de créer des environnements informatiques virtuels, indépendants du matériel physique, à l'aide de logiciels spécialisés appelés hyperviseurs. L'hyperviseur est installé sur un serveur physique et permet de créer des machines virtuelles « VM » qui peuvent être exécutées indépendamment les unes des autres sur le même serveur physique.

Chaque machine virtuelle peut exécuter un système d'exploitation et des applications comme si elle était une machine physique distincte, et dispose de ses propres ressources, telles que le processeur, la mémoire, le stockage et les interfaces réseau. [7] La virtualisation permet ainsi de maximiser l'utilisation des ressources informatiques, en permettant à plusieurs machines virtuelles d'être exécutées sur un seul serveur physique, ce qui réduit les coûts et augmente l'efficacité.

Cette technologie présente également d'autres avantages, tels que la flexibilité, la portabilité et la réduction des temps d'arrêt en cas de défaillance matérielle car ils peuvent être facilement être déplacées d'un serveur à un autre.

La virtualisation est utilisée dans de nombreuses applications, notamment dans les environnements Cloud, où elle permet de fournir des services Cloud hautement évolutifs et efficaces. Les fournisseurs utilisent la virtualisation pour créer des infrastructures Cloud dynamiques et flexibles, capables de répondre aux besoins changeants des clients en matière de ressources informatiques [14].



I.13 Figure de difference entre une architecture traditionnel et une architecture virtuel.

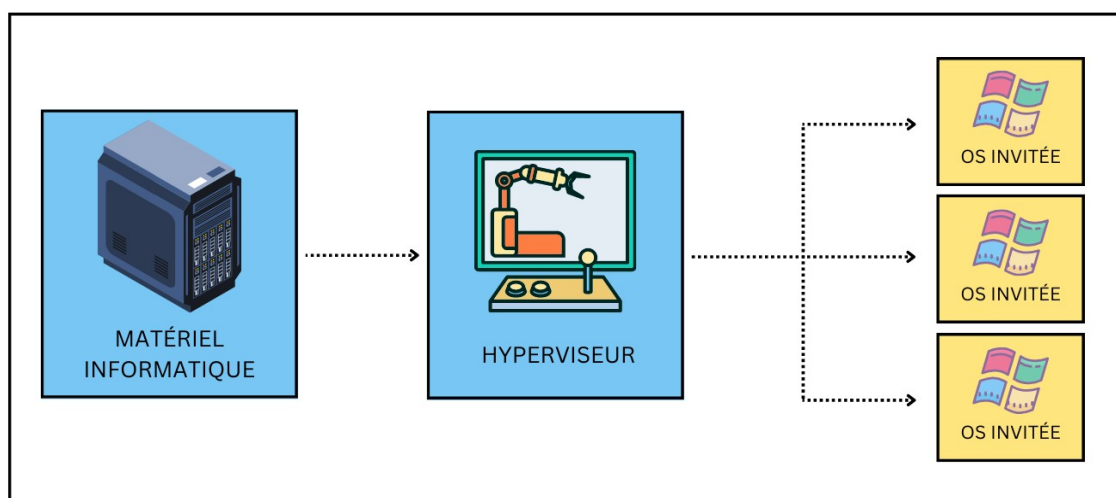
I.13.2. Hyperviseur

I.13.2.1. Définition de l'Hyperviseur

Un hyperviseur, également appelé moniteur de machine virtuelle, est un logiciel qui permet de créer et de gérer des machines virtuelles sur un serveur physique.

Les hyperviseurs permettent aux entreprises de consolider leurs ressources informatiques, d'utiliser efficacement leur matériel.

Dans la figure suivante représente le fonctionnement théorique de l'hyperviseur:



I.14 Figure de fonctionnement théorique de l'hyperviseur.

L'hyperviseur est installé directement sur le matériel et agit comme une couche de virtualisation. Il gère les ressources matérielles, telles que le processeur, la mémoire et les périphériques, et les partage entre les machines virtuelles [17].

Il permet aussi de gérer les communications entre les différents OS invités et le matériel informatique et fournit également des mécanismes de sécurité et d'isolation pour éviter les conflits entre les machines virtuelles et garantir leur fonctionnement indépendant [15] .

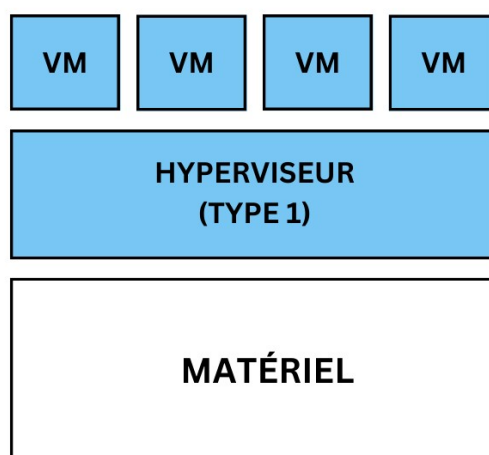
I.13.2.2 . Types d'Hyperviseur

Les hyperviseurs sont des composants essentiels de la virtualisation dans ce contexte, il est important de noter qu'il existe deux catégories d'hyperviseurs

I.13.2.2.1 Hyperviseur de type 1 ou "bare-metal"

Il s'agit d'un hyperviseur qui s'exécute directement sur le matériel du serveur physique. Les machines virtuelles s'exécutent sur cet hyperviseur, qui gère l'accès aux ressources matérielles telles que le processeur, la mémoire et les disques durs.

Ce type d'hyperviseurs sont souvent utilisés dans les centres de données pour les charges de travail critiques, telles que les bases de données et les applications d'entreprise, qui nécessitent une performance et une sécurité maximales [16].

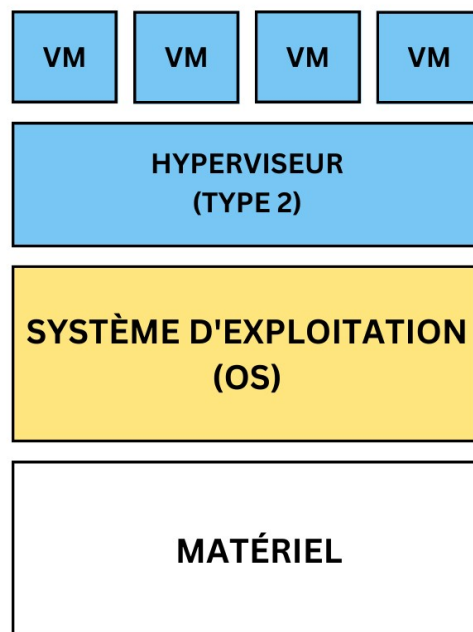


I.15 Figure de représentation de l'Hyperviseur de type 1.

I.13.2.2.2 Hyperviseur de type 2 ou "hosted"

Cela concerne un hyperviseur qui s'exécute sur un système d'exploitation hôte, tel que Windows ou Linux. Les machines virtuelles s'exécutent sur l'hyperviseur, qui gère l'accès aux ressources matérielles via l'interface logicielle fournie par le système d'exploitation hôte [15].

Les type 2 sont souvent utilisés pour la virtualisation de poste de travail et la virtualisation de test et développement. Ils sont également utilisés pour la virtualisation de petits environnements d'entreprise, où les performances et la sécurité ne sont pas des préoccupations majeures [16].



I.16 Figure de représentation de l'Hyperviseur de type 2.

I.13.3. Techniques de la virtualisation

Les techniques de virtualisation doivent être choisies avec soin pour s'adapter aux différentes composantes technologiques de l'infrastructure informatique. Il existe trois principaux types d'architectures de virtualisation qui sont les suivantes :

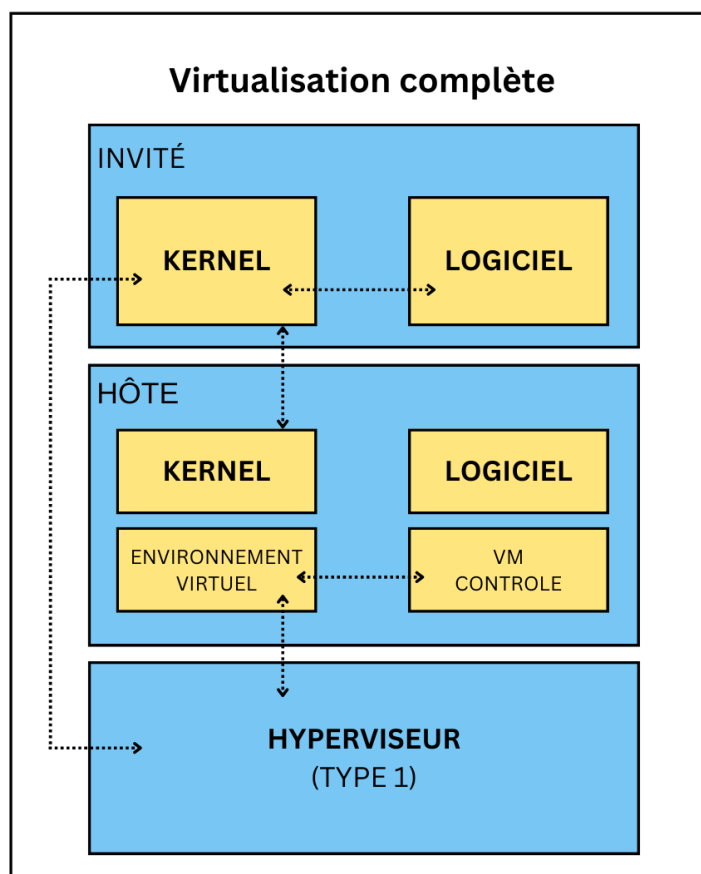
- Virtualisation complète.
- Virtualisation par émulation.
- Virtualisation Para-virtualisation [18].

chacune de ses types de virtualisation offre des avantages et des limites spécifiques en termes de performance, de flexibilité et de sécurité.

I.13.3.1 La virtualisation complète

La virtualisation complète consiste en un système d'exploitation qui exécute un logiciel appelé hyperviseur. Ce dernier permet l'exécution de plusieurs machines virtuelles sur la machine physique en gérant l'allocation des ressources telles que la mémoire et le CPU.

La virtualisation complète offre de nombreux avantages, notamment l'isolement complet des machines virtuelles pour une sécurité accrue, et une meilleure utilisation des ressources matérielles grâce à la gestion centralisée de l'hyperviseur [19].

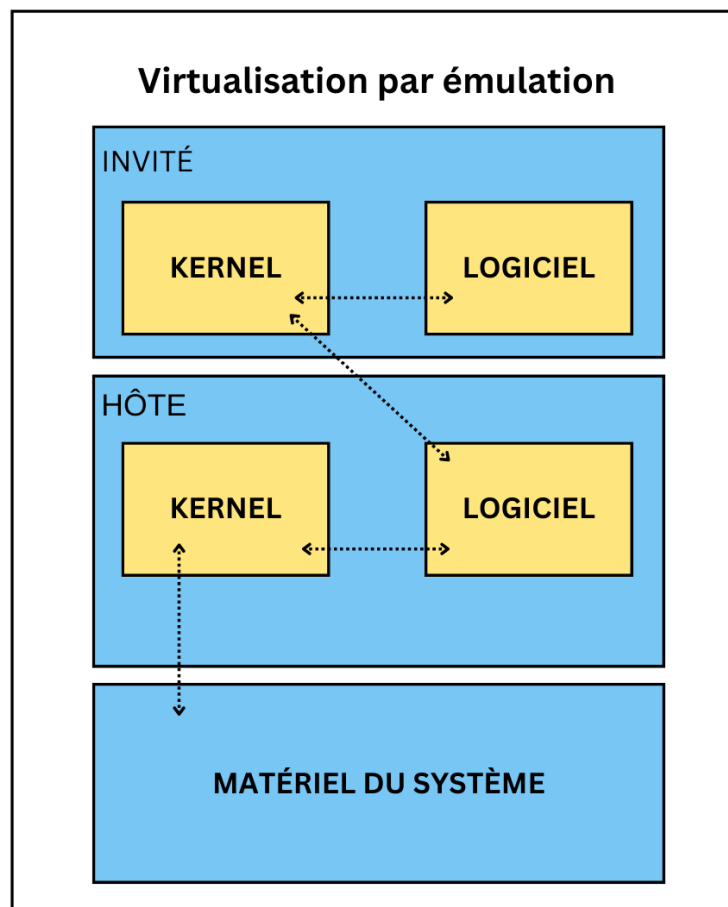


I.17 Figure de représentation de la virtualisation complète.

I.13.3.2 La virtualisation par émulation

La virtualisation par émulation consiste à émuler un environnement matériel complet pour chaque machine virtuelle. Dans ce cas, le logiciel d'émulation permet à chaque système d'exploitation invité d'exécuter ses propres pilotes de périphériques et d'accéder aux ressources matérielles de manière isolée [18].

L'émulation permet de faire fonctionner des systèmes d'exploitation différents et de faire tourner des applications qui nécessitent un environnement matériel spécifique. Cependant, cette technique de virtualisation consomme plus de ressources matérielles que la virtualisation complète, car elle nécessite l'émulation de l'environnement matériel pour chaque machine virtuelle [19].



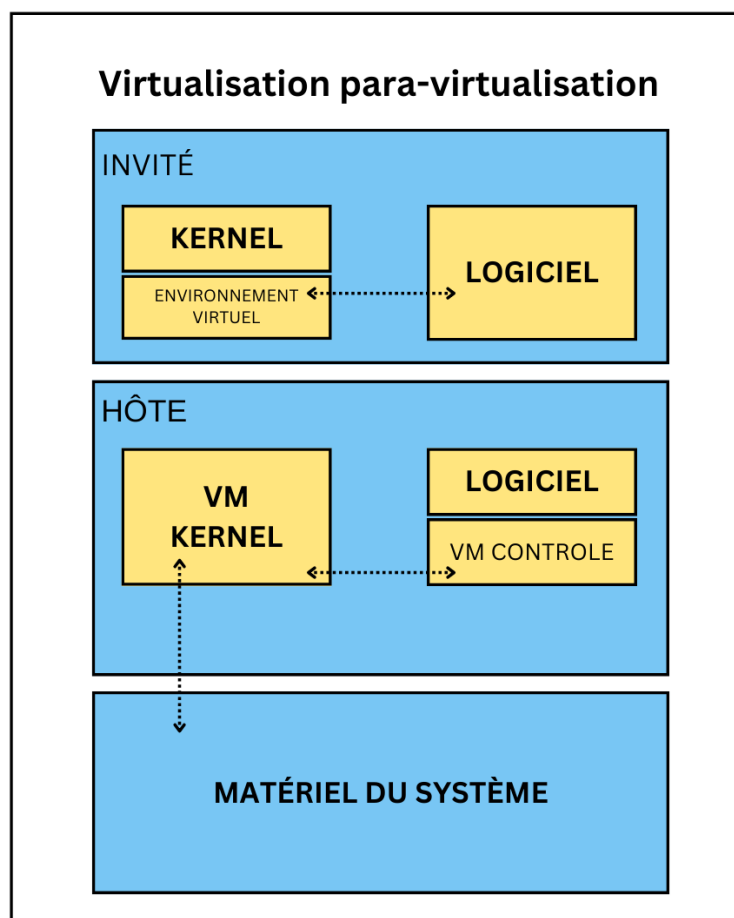
I.18 Figure de représentation de la virtualisation par émulation

I.13.3.3 La virtualisation para-virtualisation

Une para-virtualisation consiste à modifier le système d'exploitation invité afin qu'il puisse accéder directement aux ressources matérielles partagées avec l'hôte.

Cette forme de virtualisation nécessite une coordination entre le système d'exploitation invité et l'hyperviseur pour garantir l'isolation entre les machines virtuelles [19].

La para-virtualisation permet d'obtenir de meilleures performances que la virtualisation par émulation, mais elle nécessite des modifications au niveau du système d'exploitation invité [18].



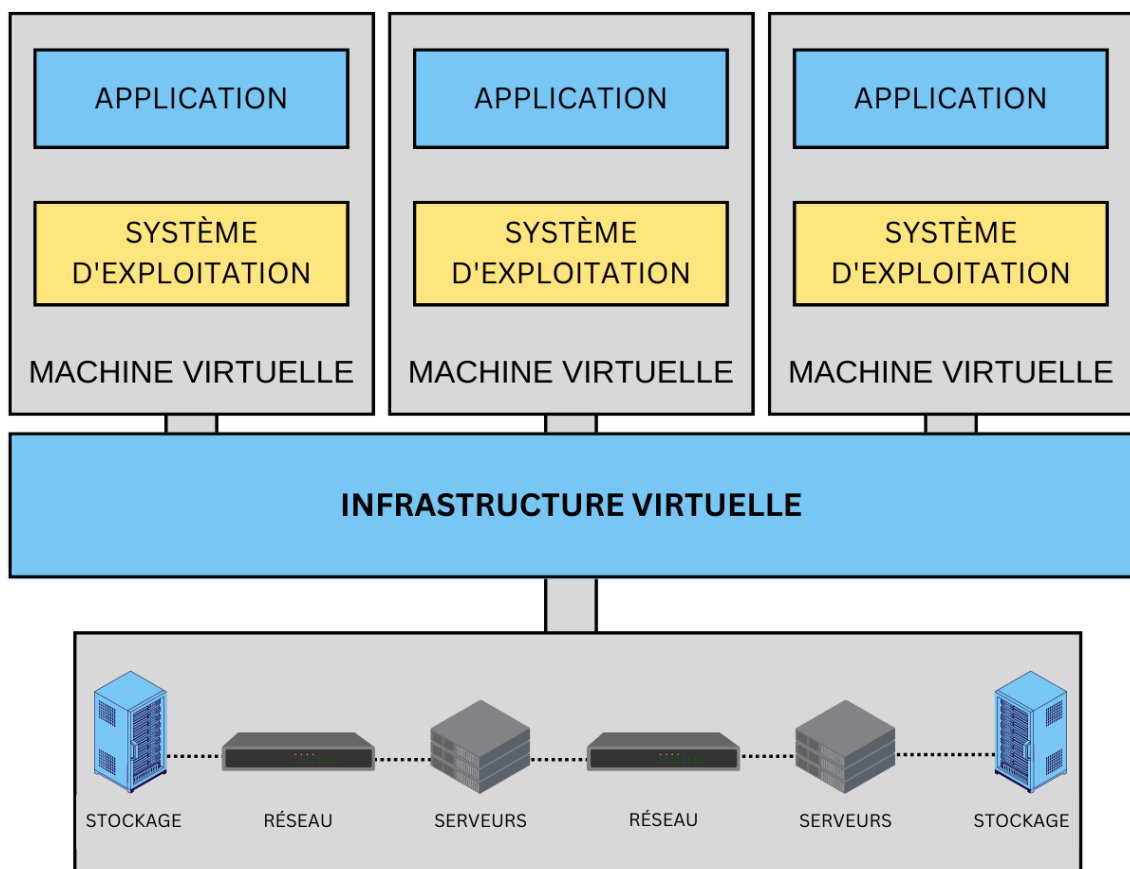
I.19 Figure de représentation de la virtualisation para-virtualisation.

I.13.4. Les formes de virtualisation

Les principales formes de virtualisation sont les suivantes :

- **Virtualisation matérielle** : Cette forme utilise des processeurs équipés de fonctionnalités de virtualisation pour exécuter plusieurs machines virtuelles sur une seule machine physique. Les processeurs modernes, tels que ceux d'Intel et d'AMD, incluent des extensions matérielles spéciales qui permettent de créer des environnements d'exécution isolés pour chaque machine virtuelle [20].
- **Virtualisation logicielle** : Est une pratique qui utilise un hyperviseur logiciel qui s'exécute sur un système d'exploitation hôte pour créer et gérer des machines virtuelles[21]. L'hyperviseur gère l'allocation des ressources et fournit une interface de programmation pour les applications [20].

- Virtualisation de la mémoire : Cette forme de virtualisation permet à plusieurs machines virtuelles d'accéder à la même mémoire physique en utilisant des techniques de partitionnement et de gestion de la mémoire. Elle permet d'optimiser l'utilisation de la mémoire en partageant une mémoire identiques entre les machines virtuelles [20].
- Virtualisation des données : Cette méthode permet de créer des copies virtuelles de données stockées sur des serveurs de stockage physiques. Elle permet de réduire les coûts de stockage et de gérer plus facilement les données en les centralisant [20].
- Virtualisation du réseau : Ce model de virtualisation permet de créer des réseaux virtuels qui peuvent être utilisés pour isoler le trafic entre les machines virtuelles et pour fournir une connectivité réseau aux machines virtuelles [20].



I.20 Figure de représentation montrant des formes de virtualisation en cours d'utilisation dans une infrastructure de virtualisation.

I.14. Avantages et inconvénients du Cloud computing

I.14.1. Avantages du cloud computing

Le cloud computing présente plusieurs avantages pour les entreprises et les particuliers, notamment :

- **Économies de coûts** : Le cloud computing donne la permission aux entreprises de réduire leurs coûts en évitant d'avoir à investir dans des infrastructures matérielles et logicielles coûteuses.
Les utilisateurs peuvent simplement louer des ressources cloud en fonction de leurs besoins, ce qui peut être beaucoup moins cher que l'achat de matériel et de logiciels.
- **Accès facile et rapide aux données** : Les services cloud permet aux utilisateurs d'accéder facilement à leurs données à partir de n'importe où et à tout moment, à condition qu'ils aient une connexion Internet. Les fichiers sont stockés sur des serveurs distants, ce qui signifie que les utilisateurs peuvent y accéder facilement.
- **Évolutivité** : Le cloud computing offre aux entreprises la possibilité de faire évoluer rapidement leur utilisation des ressources en fonction de leurs besoins, sans nécessiter d'investissement supplémentaire dans des infrastructures. Les fournisseurs de services cloud proposent des solutions évolutives qui peuvent être ajustées en fonction de la croissance de l'entreprise.
- **Sauvegarde et récupération des données** : Le cloud computing permet de sauvegarder facilement des données et de les récupérer rapidement en cas de perte de données. Les fournisseurs de services cloud proposent des sauvegardes régulières pour protéger les données de leurs clients.

I.14.2. Inconvénients du cloud computing

Le cloud computing présente également des inconvénients et des risques pour les entreprises et les particuliers, notamment :

- **Sécurité des données** : Les données stockées dans le cloud peuvent être vulnérables aux cyberattaques et aux violations de sécurité [22]. Les entreprises doivent être conscientes des risques de sécurité associés au stockage de données dans le cloud et prendre des mesures pour protéger leurs données.

- Coûts supplémentaires : Bien que le cloud computing puisse offrir des économies de coûts à court terme, les entreprises peuvent finir par payer plus cher à long terme en raison des coûts supplémentaires liés à la migration des données, à l'intégration de systèmes, à la formation du personnel et aux coûts de maintenance [22].
- Perte de données : En cas de panne du fournisseur de services cloud, les données stockées dans le cloud peuvent être perdues de manière permanente. Les entreprises doivent mettre en place des plans de sauvegarde et de récupération des données pour minimiser les risques de perte de données.
- Dépendance au fournisseur de services cloud : Les entreprises qui utilisent le cloud computing dépendent des fournisseurs de services cloud pour la fourniture et la gestion de leurs services informatiques [22]. Si le fournisseur de services cloud rencontre des problèmes techniques ou financiers, cela peut affecter les opérations de l'entreprise.

I.15. Conclusion

Dans ce chapitre, nous avons dressé un panorama complet du cloud computing, en examinant son évolution historique, ses caractéristiques essentielles, ainsi que les différents modèles de déploiement et de services qu'il propose.

Nous avons également abordé le domaine de la virtualisation, qui est une technique fondamentale pour la mise en place d'infrastructures de cloud computing efficaces et flexibles, ainsi que les avantages et les risques du cloud computing.

Dans le prochain chapitre de ce travail, nous allons explorer plus en détail le fonctionnement du cloud computing du point de vue du fournisseur de services, en nous concentrant sur la planification et l'ordonnancement des tâches qui permet en planifiant l'exécution des tâches de manière optimale de répondre aux demandes des utilisateurs de manière efficace.

CHAPITRE II

ORDONNANCEMENT DES TÂCHES DANS LE CLOUD COMPUTING

II.1. Aperçu

Le Cloud Computing est une solution qui offre la possibilité d'accéder sur demande à des ressources et services partagés, proposés sous forme de services via un réseau et accessibles via Internet. Cette approche est devenue une méthode à la fois intéressante et avantageuse pour transformer l'informatique dans son ensemble.

Le Cloud Computing est devenu une méthode intéressante et avantageuse pour transformer l'informatique dans son ensemble. Dans cette approche, les utilisateurs envoient leurs tâches pour qu'elles soient exécutées dès que possible dans un ensemble de ressources partagées disponibles. Chaque centre de données de fournisseur de Cloud dispose d'un composant qui supervise et gère la planification des tâches sur différentes machines virtuelles.

L'ordonnancement des tâches est un aspect essentiel du cloud computing, qui consiste à planifier et à exécuter efficacement les tâches informatiques dans un environnement distribué et hautement virtualisé,

Dans ce contexte les défis de l'ordonnancement des tâches incluent la prise en compte des contraintes de ressources, la minimisation du temps d'exécution et l'optimisation des coûts.

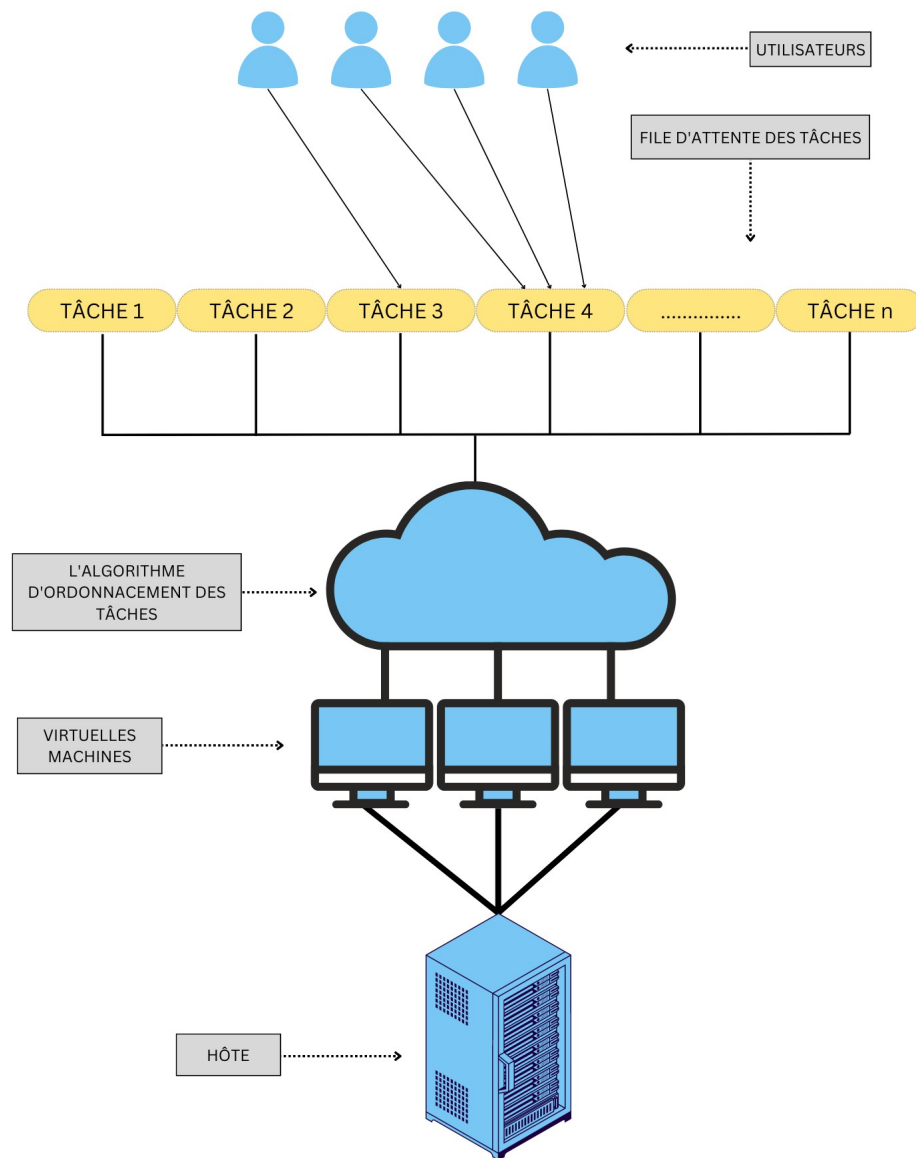
Le présent chapitre traite des fondements et des principes de l'ordonnancement, les définitions des algorithmes d'ordonnancement qui existent ainsi que les en se concentrant sur son utilisation dans le cadre spécifique du cloud computing.

II.2. Ordonnancement des tâches dans cloud computing

Le problème de l'ordonnancement des tâches consiste à attribuer efficacement les tâches informatiques à des ressources disponibles, telles que des processeurs ou des serveurs, dans le but d'optimiser les performances du système et de maximiser l'utilisation des ressources et de faire la minimisation du temps d'exécution [28] .

Cela peut être un défi dans les environnements de cloud computing de grande échelle, où de nombreuses tâches doivent être exécutées simultanément sur des ressources partagées.

Les stratégies d'ordonnancement doivent donc prendre en compte divers facteurs tels que la disponibilité des ressources, les contraintes de temps, les priorités des tâches, tout en s'assurant que les performances du système sont satisfaisantes et que les utilisateurs obtiennent les résultats attendus.



II.21 Figure de modèle d'ordonnement de tâches dans le cloud computing

Dans cette représentation on met en évidence un modèle de flux des tâches, de leur origine dans la file d'attente, à travers l'ordonnement et l'exécution sur les machines virtuelles, jusqu'à leur traitement final sur le hôte [29].

- L'utilisateur représente la source des tâches à exécuter.
- La file d'attente des tâches est l'endroit où les tâches sont stockées en attendant d'exécution.
- Les tâches numérotées de 1 à n représentent les différentes tâches à exécuter.
- L'algorithme d'ordonnement des tâches représente le mécanisme utilisé pour décider de l'ordre dans lequel les tâches seront exécutées.
- L'hôte est l'infrastructure de calcul où les machines virtuelles sont exécutées et où les tâches sont traitées.

II.3. Notions de bases

II.3.1. Job

Un « Job » peut être défini comme un ensemble de tâches informatiques qui doivent être effectuées simultanément ou de manière coordonnée pour atteindre un objectif donné.

Les jobs sont souvent utilisés pour automatiser des processus complexes qui nécessitent plusieurs étapes pour être achevés [33].

II.3.2. Cloudlet

Un «cloudlets» est une petite infrastructure et une extension du cloud computing, dans le contexte de l'ordonnancement des tâches, un cloudlet peut être considéré comme une unité de traitement de tâches. Il peut être utilisé pour offrir des services de traitement de tâches de manière distribuée, ce qui permet de répartir la charge de travail entre différents cloudlets.

II.3.3. Ressources

Une ressource est un élément qui est utilisé pour effectuer une tâche ou un travail donné. Dans le contexte de l'informatique et du cloud computing, les ressources peuvent être des éléments matériels tels que des processeurs, des mémoires, des disques durs, des cartes réseaux [33].

Dans le domaine de l'ordonnancement des tâches, il existe deux types de ressources :

- Les ressources renouvelables : Elles sont disponibles en quantité constante tout au long de l'exécution des tâches et peuvent être allouées à plusieurs tâches de manière répétée. Les ressources renouvelables comprennent les machines, les processeurs, les mémoires, les disques de stockage.
- Les ressources consommables : Elles ne sont pas disponibles en quantité constante tout au long de l'exécution des tâches et sont utilisées une seule fois. Une fois qu'une ressource consommable a été allouée à une tâche, elle n'est plus disponible pour les autres tâches. Les exemples de ressources consommables comprennent les matières premières, l'énergie, les batteries.

II.4. Les contraintes de l'ordonnancement des tâches

Dans un contexte d'ordonnancement des tâches, les contraintes peuvent être des restrictions imposées sur les tâches qui doivent être exécutées, sur les ressources qui doivent être utilisées ou sur les délais qui doivent être respectés et peuvent varier en fonction du contexte et des objectifs de l'ordonnancement. Les principales contraintes sont [30] :

- Les contraintes de temps : Les tâches peuvent avoir des dates d'échéance ou des délais à respecter. L'ordonnancement doit donc tenir compte de ces contraintes de temps pour s'assurer que les tâches sont exécutées dans les délais requis.
- Les contraintes de dépendance : Les tâches peuvent dépendre les unes des autres, ce qui signifie que certaines tâches ne peuvent pas commencer tant que d'autres n'ont pas été achevées. L'ordonnancement doit donc prendre en compte ces dépendances pour éviter les retards et les blocages.
- Les contraintes de ressources : Les tâches peuvent nécessiter différentes ressources pour être exécutées, telles que des machines, du personnel, des matières premières. L'ordonnancement doit donc tenir compte de ces contraintes de ressources pour s'assurer que les ressources nécessaires sont disponibles au moment où les tâches en ont besoin.
- Les contraintes de coûts : L'ordonnancement peut également être contraint par des considérations de coûts, telles que les coûts de main-d'œuvre, les coûts de production, les coûts de stockage. L'ordonnancement doit donc prendre en compte ces contraintes de coûts pour optimiser l'utilisation des ressources disponibles et minimiser les coûts globaux.

II.5. Le rôle de l'ordonnanceur dans le cloud computing

L'ordonnanceur de tâches «également appelé "scheduler" en anglais» joue un rôle essentiel dans le cloud computing en coordonnant et en gérant la distribution des ressources informatiques entre les différents utilisateurs et applications. Plus précisément, l'ordonnanceur de tâches est responsable de l'allocation des ressources de calcul «PEs» [31], de stockage et de réseau nécessaires pour exécuter les tâches des utilisateurs de manière efficace. Il doit également s'assurer que les ressources sont utilisées de manière optimale afin d'optimiser les performances du système tout en respectant les contraintes de coûts et de temps de traitement, l'ordonnanceur de tâches peut être utilisé pour [31] :

- Planifiez les tâches en fonction de la priorité et du calendrier : L'ordonnanceur peut planifier l'exécution des tâches en fonction des niveaux de priorité et des délais afin que les tâches importantes soient exécutées en premier et que les délais soient respectés.
- Équilibre de la charge de travail : L'ordonnanceur peut répartir la charge de travail uniformément sur différents serveurs et nœuds de traitement pour éviter surcharge et optimiser l'utilisation des ressources.
- Optimisation des coûts : L'ordonnanceur peut être configuré pour utiliser les ressources disponibles les moins chères pour exécuter les tâches, ce qui permet de réduire les coûts de traitement.

II.6. Types d'ordonnancement dans Cloud computing

Les principales types d'ordonnancement dans le cloud computing sont les suivantes [35] :

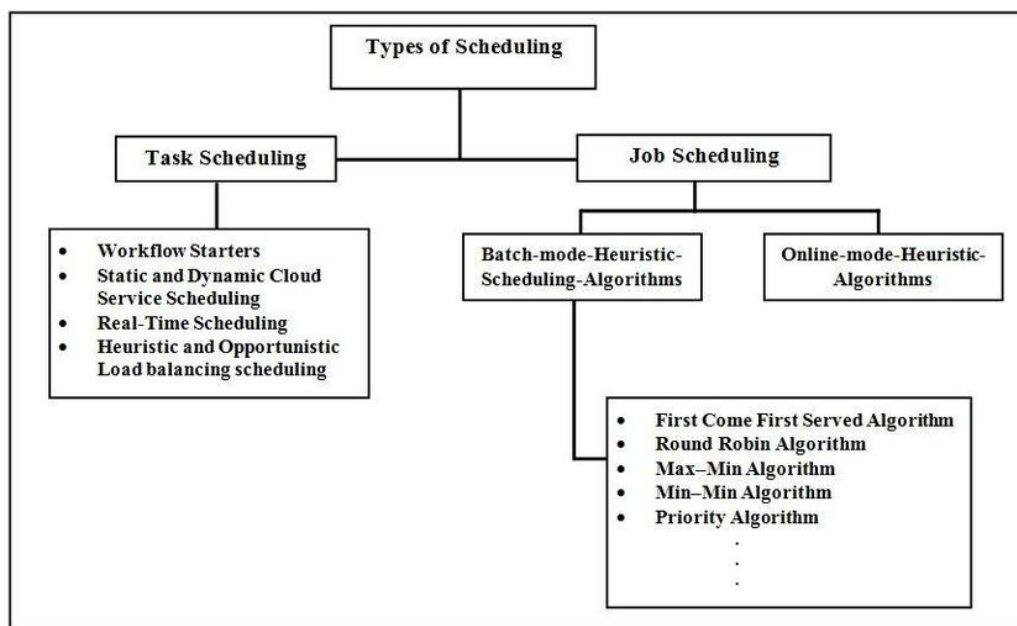
- Ordonnancement de Job: Généralement, impliquent un ensemble de tâches indépendantes. La plupart des méthodes commencent par évaluer l'importance des tâches dans leur ensemble en fonction de critères tels que le temps d'attente, la date limite, le moment d'arrivée, le temps d'exécution, afin de déterminer la priorité de chaque tâche et de trouver la meilleure correspondance possible.
- Ordonnancement de workflows: Consiste en un ensemble de tâches qui dépendent les unes des autres et sont représentées par un graphe acyclique dirigé, où chaque nœud correspond à une tâche et chaque arête représente la communication entre les tâches qui en dépendent. Une tâche ne peut être exécutée que lorsque toutes les tâches parentes ont été achevées.
- Ordonnancement des VM: Les fournisseurs de services de Cloud créent des groupes de machines virtuelles dans une réserve de ressources, puis chaque VM doit être assignée à un hôte. en tenant compte de différentes contraintes telles que les exigences de service, la disponibilité des ressources, la sécurité.
- Ordonnancement de stockage: Cela fait référence à la gestion d'un ensemble de blocs de données stockés de différents types dans le cloud computing, tels que des images, des textes ou des sons. En raison de la grande capacité de ces données une méthode de traitement est nécessaire pour stocker et récupérer les données. qit implique la création d'une partition des données en fonction des ressources de stockage disponibles.

II.7. Les Algorithmes d'ordonnancement des tâches

II.7.1. Types de catégories d'algorithmes d'ordonnancement des tâches

Les algorithmes d'ordonnancement peuvent être divisés en deux catégories principales : l'ordonnancement des tâches «task scheduling» et l'ordonnancement des jobs «job scheduling».

- L'ordonnancement des tâches « Task Scheduling » : Consiste à ordonner et à allouer des ressources pour l'exécution des tâches individuelles. Les tâches peuvent être exécutées sur un seul processeur ou sur plusieurs processeurs. L'objectif de l'ordonnancement des tâches est de minimiser le temps d'exécution total et de maximiser l'utilisation des ressources.
- L'ordonnancement des jobs « Job Scheduling » : Permet d'ordonner et à allouer des ressources pour l'exécution d'un ensemble de tâches qui sont liées entre elles et qui doivent être exécutées dans un ordre précis. Les jobs peuvent être exécutés sur un seul processeur ou sur plusieurs processeurs. L'objectif de l'ordonnancement des jobs est de respecter les délais et les contraintes temporelles associées à chaque job, tout en maximisant l'utilisation des ressources.



II.22 Diagramme de divers catégories d'algorithmes d'ordonnancement [32]

II.7.2. Définitions des catégories d'algorithmes d'ordonnancement des tâches

- Ordonnancement statique et dynamique : L'ordonnancement statique pré-planifie les tâches à l'avance, en supposant que toutes les ressources et informations sur les tâches sont disponibles avant l'exécution [31]. Cependant, l'ordonnancement dynamique permet aux tâches d'être disponibles pour l'ordonnanceur au fil du temps, sans qu'il soit possible de déterminer à l'avance leur temps d'exécution. L'exécution dynamique de la tâche peut être affectée par des pannes de ressources ou des interruptions de tâches de priorité supérieure en mode préemptif [36].
- Ordonnancement centralisé, hiérarchique et distribué : L'ordonnancement centralisé est caractérisé par un contrôle central sur les ressources et une connaissance globale du système, tandis que l'ordonnancement hiérarchique permet la coordination de plusieurs ordonnancements à différents niveaux[31]. Dans l'ordonnancement distribué, il n'y a pas d'entité centrale qui contrôle les ressources, les décisions d'ordonnancement sont partagées par plusieurs ordonnanceurs distribués.
- Ordonnancement préemptif et non préemptif : L'ordonnancement préemptif permet à une tâche d'être interrompue pendant son exécution, et peut être déplacée vers une autre ressource. Dans l'ordonnancement non préemptif, les ressources ne sont pas réallouées jusqu'à ce que la tâche en cours d'exécution soit terminée [36].
- Ordonnancement immédiat et en mode batch : L'ordonnancement immédiat ou en ligne planifie une tâche dès son arrivée, sans attendre un intervalle de temps ultérieur [36]. L'ordonnancement en mode batch ou hors ligne regroupe les tâches en groupes pour les résoudre sur des intervalles de temps successifs [31].
- Ordonnancement indépendant et ordonnancement de workflow: Dans l'ordonnancement indépendant, les tâches peuvent être exécutées indépendamment les unes des autres. Dans l'ordonnancement de workflow, les tâches sont interdépendantes, c'est-à-dire qu'il y a un ordre de précedence entre les tâches [36], ce qui signifie qu'une tâche ne peut pas être exécutée tant que toutes ses tâches parentes ne sont pas terminées. Les workflows sont représentés par des graphes acycliques la minimisation du temps d'exécution dirigés où chaque nœud représente une tâche et les connexions entre les nœuds représentent les dépendances entre les tâches [31].

II.7.3. Les principaux algorithmes d'ordonnancement

Il existe de nombreux algorithmes d'ordonnancement utilisés dans la planification des tâches ou des processus. Voici quelques-uns des principaux algorithmes d'ordonnancement :

II.7.3.1. First-Come-First-Served

First-Come-First-Served «FCFS» est l'un des algorithmes les plus simples d'ordonnancement des tâches dans le cloud computing. Il fonctionne en traitant les tâches dans l'ordre dans lequel elles sont soumises, sans prendre en compte leur priorité ou leur temps d'exécution.

Cet algorithme peut être utilisé dans des situations où les tâches ont des exigences de temps d'exécution similaires et où la priorité n'est pas un facteur important [36].

Les avantages de l'algorithme FCFS :

- Facile à mettre en œuvre,
- Pas de complexité d'algorithme.
- Pas de temps de latence.

Les inconvénients de l'algorithme FCFS :

- Manque de considération de la priorité et du temps d'exécution, peut causer un temps d'attente élevé pour les tâches de longue durée.

II.7.3.2. Shortest-Job-First

Shortest-Job-First « SJF » est un algorithme d'ordonnancement des tâches dans le cloud computing qui ordonne les tâches en fonction de leur durée d'exécution prévue. Les tâches les plus courtes sont exécutées en premier, suivies des tâches plus longues

L'algorithme SJF peut être utilisé dans des situations où la durée d'exécution des tâches est relativement prévisible et où il est important de réduire le temps d'attente et de latence des tâches [36].

Les avantages de l'algorithme SJF :

- Temps d'exécution plus court pour les tâches de courte durée
- Amélioration de l'utilisation des ressources
- Réduction des temps d'attente et de latence.

Les inconvénients de l'algorithme SJF :

- Difficulté de prévoir le temps d'exécution des tâches et favoritisme de la tâche de courte durée.

II.7.3.3. Round Robin (RR)

Round Robin est un algorithme qui divise le temps d'exécution du processeur en segments de temps égaux. Chaque tâche est traitée pendant un quantum et si elle n'est pas terminée, elle est mise en attente et la tâche suivante est traitée. Ce processus est répété jusqu'à ce que toutes les tâches soient terminées [31].

L'algorithme Round Robin peut être utilisé dans des situations où les tâches ont des exigences de temps d'exécution similaires et où il est important de garantir un temps d'exécution équitable pour toutes les tâches [36].

Les avantages de l'algorithme Round Robin :

- Temps de réponse rapide pour les tâches de courte durée.
- Partage équitable des ressources.
- Réduction des temps d'attente et de latence.

Les inconvénients de l'algorithme Round Robin :

- Risque de gaspillage de ressources pour les tâches de longue durée.
- Risque de fragmentation des ressources.
- Risque de surcharge pour les tâches de longue durée.

II.7.3.4. Min-min

L'algorithme de Minmin vise à minimiser le temps total d'exécution de toutes les tâches en optimisant le temps d'exécution de chaque tâche individuelle. Il sélectionne la tâche ayant le temps d'exécution minimal dans chaque étape de l'ordonnancement.

L'algorithme calcule le temps d'exécution minimal pour chaque tâche en utilisant une méthode qui consiste à trouver la tâche ayant le temps d'exécution minimal [36] et à la retirer de la liste des tâches restantes, puis à recalculer les temps d'exécution pour les tâches restantes. Cette étape se répète jusqu'à ce qu'il ne reste plus de tâches [31].

Les avantages de l'algorithme Min-min :

- Permet de minimiser le temps total d'exécution de toutes les tâches.
- Ne prend pas en compte les contraintes de dépendance entre les tâches.

Les inconvénients de l'algorithme Min-min :

- L'algorithme très sensible aux estimations de temps d'exécution.
- Retards importants dans l'exécution des tâches dû aux erreurs d'estimation de temps

II.7.3.1.5 Max-min

L'algorithme Maxmin est un algorithme d'ordonnancement de tâches dans le cloud computing qui vise à maximiser l'utilisation des ressources en donnant la priorité aux tâches les plus longues à exécuter. Cet algorithme sélectionne la tâche ayant le temps d'exécution maximal dans chaque étape de l'ordonnancement [31]

L'algorithme calcule le temps d'exécution maximal pour chaque tâche en utilisant une méthode qui consiste à trouver la tâche ayant le temps d'exécution maximal et à la retirer de la liste des tâches restantes, [36] puis à recalculer les temps d'exécution pour les tâches restantes. Cette étape se répète jusqu'à ce qu'il ne reste plus de tâches.

Les avantages de l'algorithme Max-min [35] :

- L'algorithme est efficace pour maximiser l'utilisation des ressources
- Minimiser les temps d'attente des tâches les plus longues.
- Ne prend en compte les contraintes de dépendance entre les tâches.

Les inconvénients de l'algorithme Max-min :

- Utilisation inéquitable des ressources si les temps d'exécution des tâches sont très différents.
- Les tâches les plus longues peuvent monopoliser les ressources, ce qui peut entraîner des temps d'attente élevés pour les tâches plus courtes.

II.8.Conclusion

En conclusion, l'ordonnancement des tâches est un domaine essentiel dans les systèmes informatiques. Les algorithmes d'ordonnancement permettent de gérer efficacement l'exécution des tâches en attribuant les ressources disponibles de manière optimale.

Il existe différentes catégories d'algorithmes, chacune ayant ses propres caractéristiques et avantages. Les contraintes de l'ordonnancement des tâches, telles que les dépendances entre les tâches, les contraintes de ressources et les contraintes de temps, doivent être prises en compte pour garantir un fonctionnement efficace du système

CHAPITRE III

CONCEPTION ET ANALYSE

III.1. Aperçu

L'ordonnancement des tâches est un aspect important des applications cloud à grande échelle. La recherche traite des problèmes rencontrés dans un environnement réel est extrêmement complexe qui échappent au contrôle des développeurs.

Comme mentionné précédemment, l'ordonnancement des tâches dans le cloud computing est un défi majeur qui nécessite une allocation efficace des ressources tout en minimisant le temps d'exécution. Pour résoudre ce problème, une optimisation a été proposée, impliquant l'utilisation d'un algorithme d'ordonnancement conçu pour traiter et améliorer ces deux objectifs.

Ce chapitre se concentre sur la présentation et l'explication de l'algorithme proposé pour l'ordonnancement des tâches dans le cloud computing, en passant par la description et l'explication du modèle proposé et des techniques d'optimisation utilisées.

III.2. Introduction

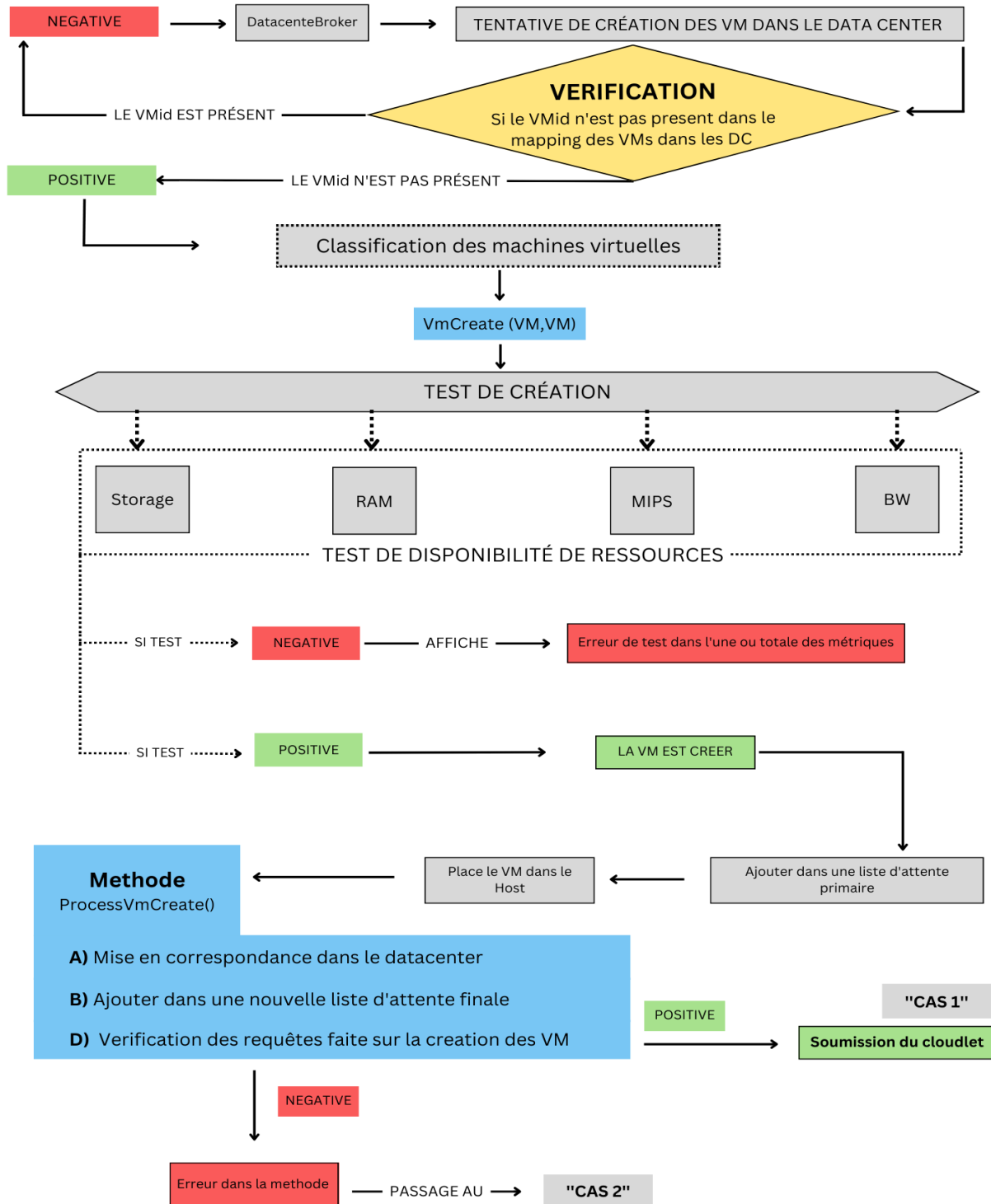
Après avoir achevé la phase initiale qui a consisté à établir les définitions essentielles du cloud computing et à développer et sélectionner des algorithmes d'ordonnancement des tâches adaptés à ce domaine, nous entrons maintenant dans la phase essentiel de l'analyse et de la conception de l'algorithme proposé.

Pour commencer, nous présenterons le conception de l'algorithme que nous avons utilisés afin de la création de ce projet. Ensuite, nous passerons au modèle proposé de l'algorithme en examinant de manière détaillée le travail fait sur cet algorithme.

III.3. Conception du travail

III.3.1. La création des machines virtuelles dans le Datacenter

III.3.1.1 Premier cas



III.23 Diagramme de création des machines virtuelles dans le datacenter

Cette phase est spécifiquement conçu pour illustrer une fonctionnalité particulière. Elle permet de créer des scénarios où plusieurs utilisateurs demandent des ressources aux fournisseurs de services cloud. Les machines virtuelles sont déployées et des tâches sont exécutées sur ces machines virtuelles simulées.

La première étape de l'algorithme est la tentative de création des VM dans le datacenter. Pour chaque VM créée, il vérifie si elle existe déjà dans le mapping du datacenter avec comme indicateur le Vmid, si ce n'est pas le cas il entame la classification des VM dans le datacenter afin de les trier, Ensuite arrive la création des VM avec la méthode «**VmCreate(VM VM)**» .

La méthode VmCreate à pour but de faire :

- La vérification des ressources
- L'Allocation de ressources

Pour allouer les ressources des VM «*RAM, MIPS, bande passante, stockage*», l'algorithme effectue un test de vérification des ressources nécessaires qui sont disponibles dans l'hôte «*la vérification des limites de quota du système*» .

Si le test de la vérification est positif, la VM est créée. Si le cas contraire qui est négatif en raison de ressources insuffisantes, un message d'erreur est renvoyé.

Une fois que la VM a été créée en utilisant la méthode «**VmCreate(VM VM)**» , la méthode «**ProcessVMCreate()**» est appelée pour effectuer les actions nécessaires liées à la création de la VM.

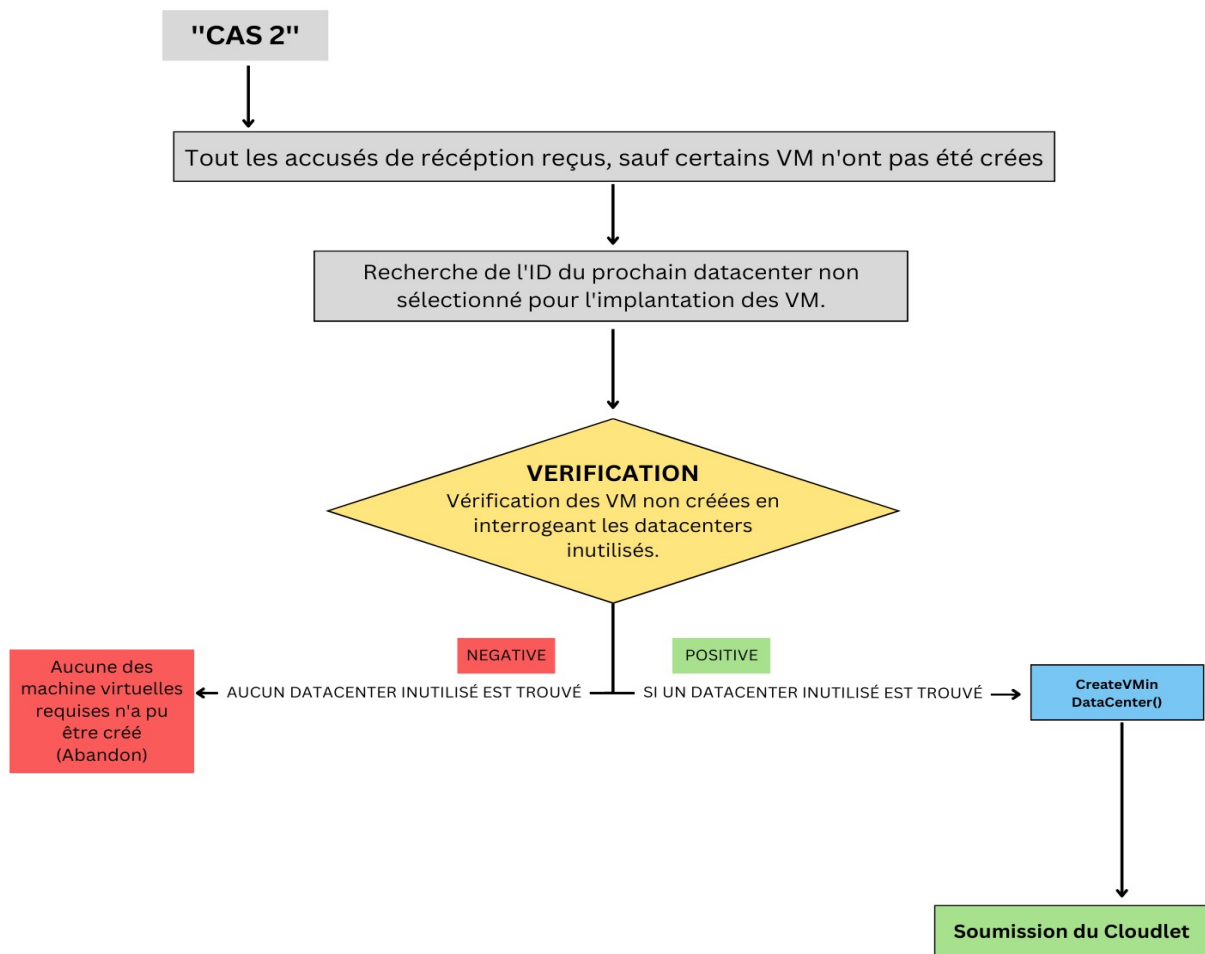
Les VM sont ajoutées à une liste d'attente primaire. Ensuite l'algorithme procède au placement des VM dans les hôtes en utilisant la méthode.

Cette méthode effectue les étapes suivantes :

- A) Mise en correspondance dans le datacenter: Cela implique de trouver un emplacement approprié pour la VM nouvellement créée.
- B) Ajout des VM à la liste d'attente finale: Une fois la mise en correspondance effectuée la méthode ajoute la VM à la liste d'attente finale pour être déployée.
- C) Vérification des requêtes de création des VM : La méthode vérifie les requêtes de création des VM pour s'assurer qu'elles sont valides et conformes aux politiques du système.

Après ces étapes, si la dernière étape qui est la vérification des requêtes est positive, le cloudlet est soumis pour l'exécution. Sinon vient le passage au deuxième cas.

III.3.1.2 Deuxième cas



III.24 Diagramme de création des machines virtuelles dans le Datacenter "CAS 2"

Le deuxième cas se produit lorsque des acquittements sont reçus, mais certaines VM n'ont pas été créées avec succès. Pour résoudre ce problème, l'algorithme propose une solution consistant à rechercher l'ID du prochain datacenter qui n'a pas été sélectionné pour l'implantation des VM.

Ensuite, l'algorithme passe à l'étape de vérification des VM qui n'ont pas été créées en interrogeant les datacenters qui ne sont pas utilisés. L'objectif est de trouver un datacenter inutilisé qui peut accueillir les VM manquantes.

Une fois que l'étape de vérification est positive qui signifie que l'algorithme a identifié un datacenter qui n'est pas utilisé et qui peut accueillir les VM manquantes, il appelle la méthode «*CreateVMinDataCenter()*» pour effectuer le processus de création dans ce

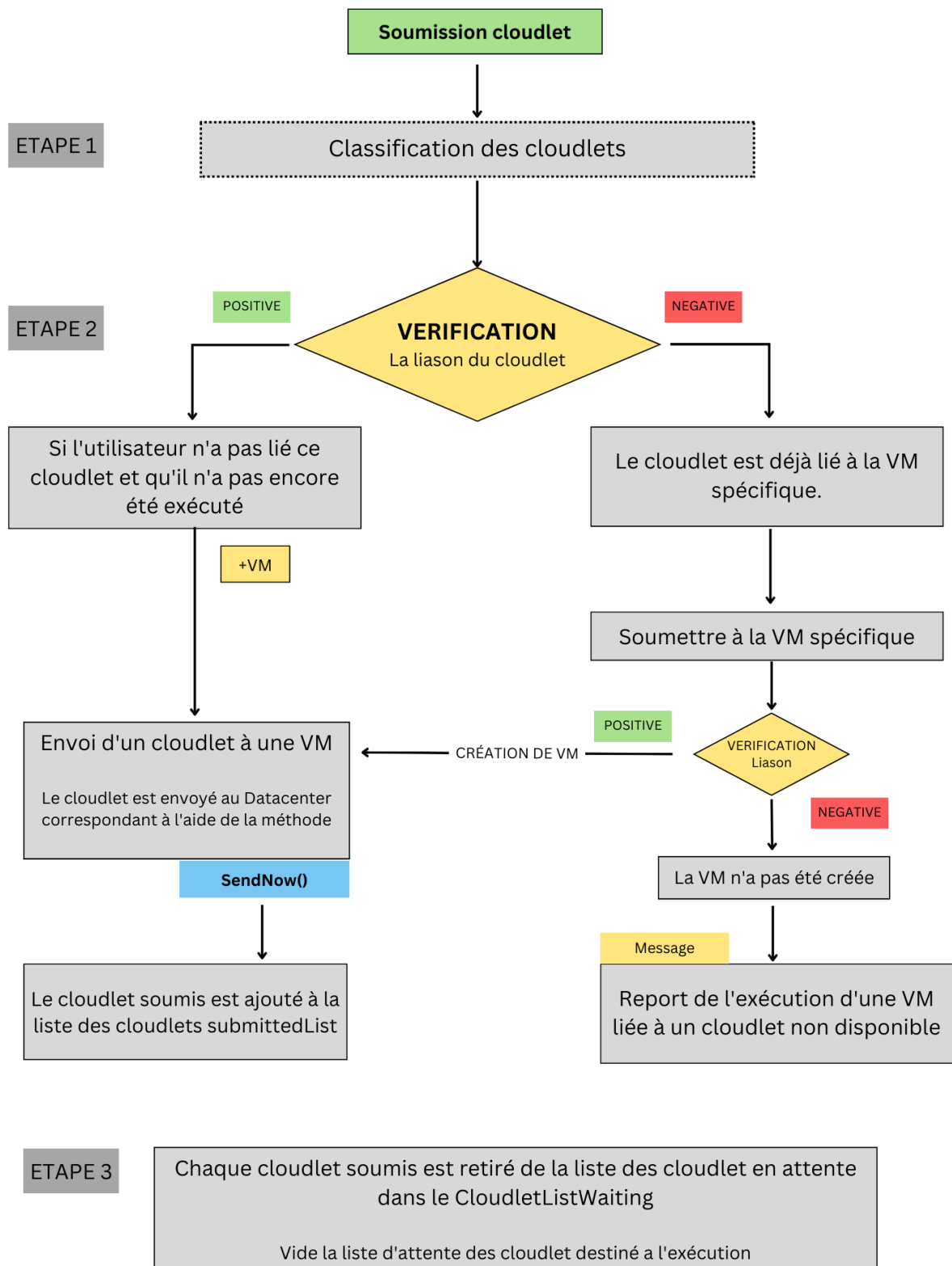
Datacenter précis alors une soumission du cloudlet est effectuée. Cela signifie que les tâches ou les processus prévus pour les VM peuvent être exécutés.

Dans le cas contraire, le résultat de la vérification est négative et qu'aucune des VM requises n'a pu être créée, un message d'erreur est renvoyé, indiquant que « Aucune des machines virtuelles requises n'a pu être créée (Abandon) » .

III.3.2. Soumission des cloudlets

Lorsqu'il s'agit de l'exécution de tâches dans un environnement de cloud computing, la Soumission des cloudlets joue un rôle essentiel car elle fait référence au processus par lequel les tâches ou les travaux, appelés cloudlets, sont envoyés et préparés pour être exécutés dans le cloud en utilisant ses ressources de calcul.

Soumission des cloudlets peut être divisée en trois étapes précises qui sont responsables de l'envoi d'une liste de cloudlets vers le centre de données pour leur traitement comme le montre le schema dans la page suivante :



III.25 Diagramme de soumission des cloudlets

Étape 1:

La première étape consiste à effectuer la classification et le tri des cloudlets selon un calcul de moyenne d'un ensemble de métriques qui sera expliqué prochainement.

Étape 2 :

Avant d'envoyer les cloudlets pour leur traitement, il est courant de procéder à une vérification des liaisons entre les cloudlets et les machines virtuelles (VM) dans lesquelles ils doivent être exécutés. Cette vérification vise à s'assurer que chaque cloudlet est correctement associé à la VM appropriée.

La vérification des liaisons des cloudlets dans les VMs permet de garantir que chaque tâche est affectée à la machine virtuelle qui convient le mieux pour son exécution.

Si la vérification est positive, cela signifie que l'utilisateur n'a pas lié ce cloudlet et qu'il n'a pas encore été exécuté, ce qui indique qu'il n'est pas lié à une machine virtuelle. Pour remédier à cela, le cloudlet est envoyé à une VM correspondante au centre de données à l'aide de la méthode «*SendNow()*».

La méthode "SendNow" est responsable de :

- L'envoi immédiat du cloudlet à la VM spécifiée.
- Décider de la meilleure VM à utiliser pour l'exécution du cloudlet.

En utilisant la méthode «*SendNow()*», Le cloudlet soumis est ajouté à la liste des cloudlets submittedList.

Si la vérification des liaisons des cloudlets est négative, cela signifie en effet que le cloudlet est déjà lié à une machine virtuelle spécifique. Dans cette situation, l'algorithme doit effectuer une vérification supplémentaire pour déterminer si la VM associée existe toujours ou non.

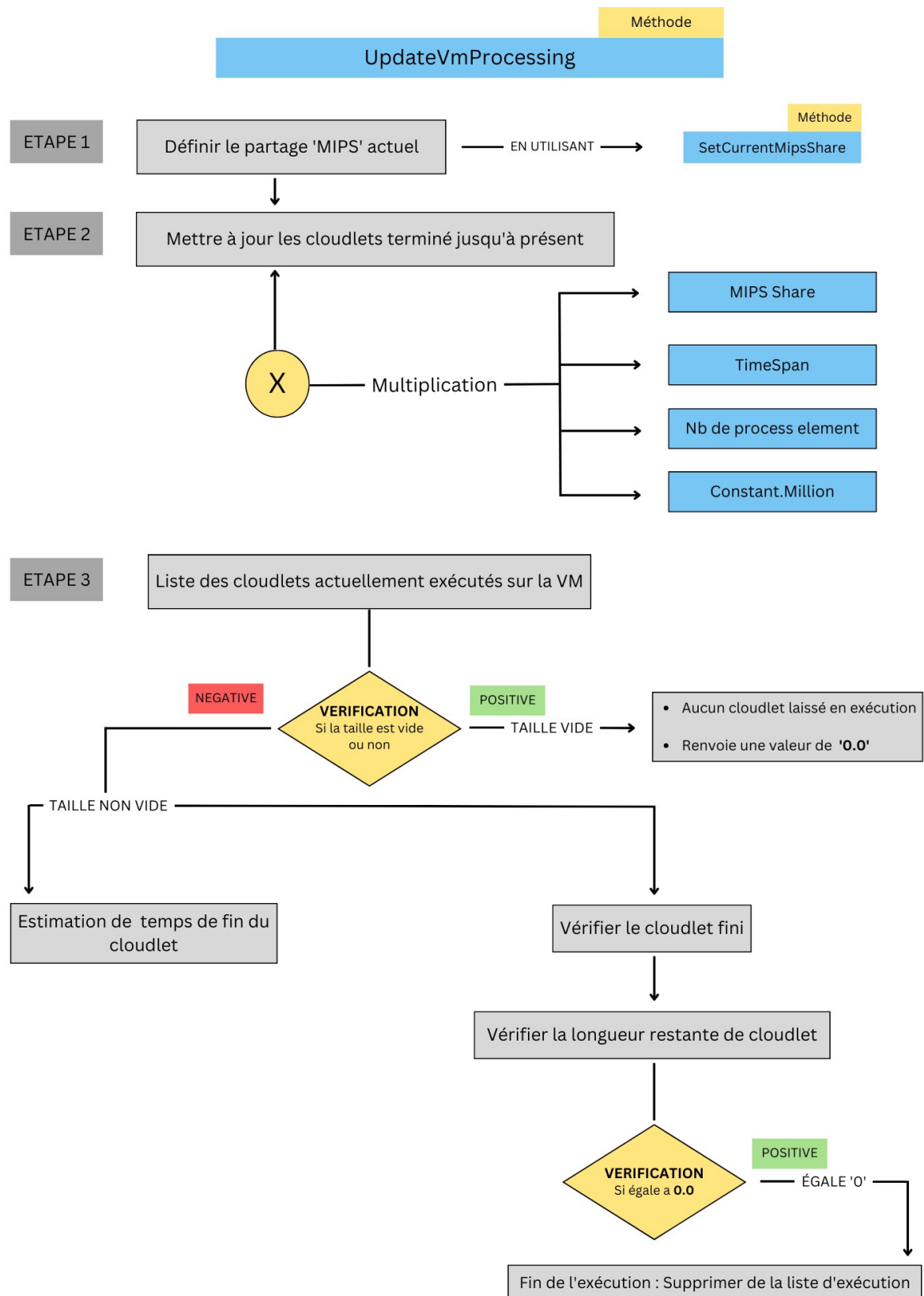
Si la vérification est positive et que la VM liée au cloudlet existe toujours, alors le cloudlet peut être envoyé à cette VM pour son exécution conformément à la planification initiale.

Cependant, si la vérification est négative et que la VM associée au cloudlet n'existe plus, cela signifie que la VM n'est pas disponible. Dans ce cas un report de l'exécution d'une VM liée à un cloudlet non disponible est envoyé.

Étape 3 :

Dans la troisième étape, lorsque chaque cloudlet soumis est envoyé pour exécution, il est retiré de la liste d'attente des cloudlets à exécuter. Cela signifie que le cloudlet n'est plus considéré comme étant en attente d'exécution et ne figure plus dans la liste «*CloudletListWaiting*».

III.3.3. Processus de mise à jour des machines virtuelles



III.26 Diagramme de processus de mise à jour des machines virtuelles

La méthode *"UpdateVMProcessing()"* est une méthode utilisée dans les systèmes de cloud computing pour mettre à jour le traitement des machines virtuelles «VM».

Son objectif principal est de mettre à jour la progression et estimer le prochain temps d'événement pour le cloudlet de la machine virtuelle. Elle est appelée périodiquement lorsque de nouveaux cloudlets sont soumis ou que des cloudlets existants sont terminés, l'algorithme de cette méthode se divise en trois étapes :

Étape 1:

On commence par définir la part actuelle de MIPS «*millions d'instructions par seconde*» pour la machine virtuelle «VM». Cela est réalisé en utilisant la méthode «*SetCurrentMipsShare*». Lorsqu'on appelle cette méthode, elle permet de définir la part actuelle de MIPS de la VM en utilisant la valeur fournie en paramètre, appelée «*mips share*» Ainsi la VM est informée de la quantité de MIPS qui lui a été allouée pour le traitement des tâches.

Étape 2:

La deuxième étape consiste à mettre à jour la progression de chaque cloudlet dans la liste d'exécution, en tenant compte des cloudlets qui ont déjà été terminés.

La mise à jour de la progression est calculée en multipliant la capacité «*mips share*» de la VM par la durée «*time span*» et le nombre d'éléments de traitement «*PEs*» du cloudlet.

Le résultat de cette multiplication est utilisé pour mettre à jour l'objet cloudlet avec sa progression terminée.

Étape 3:

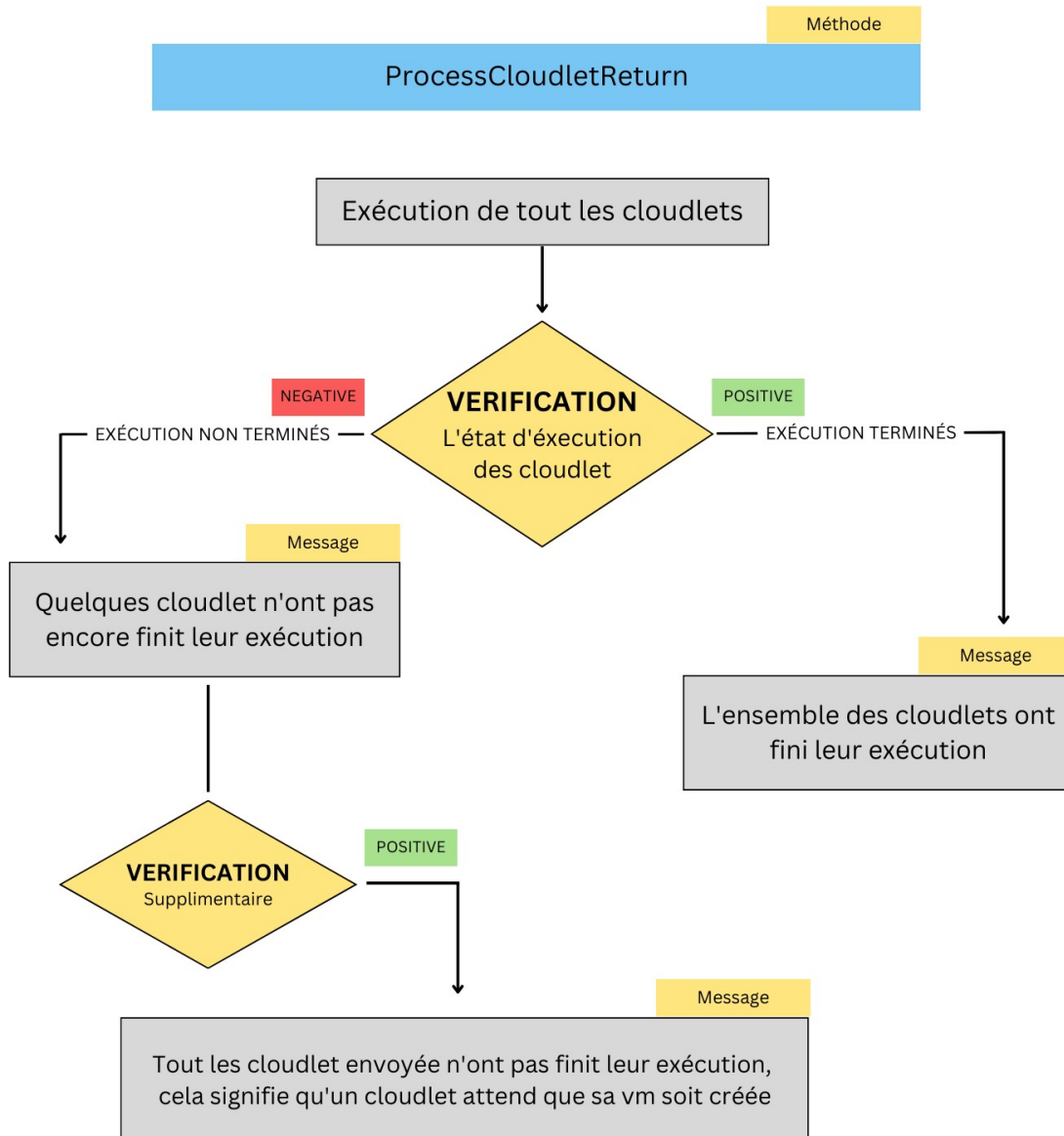
Enfin, dans la troisième étape, l'algorithme vérifie si la taille de la liste des cloudlets exécutés actuellement sur la VM est vide ou non.

Si la taille de la liste est vide, cela signifie qu'aucun cloudlet n'est actuellement en cours d'exécution sur la VM. Dans ce cas, l'algorithme définit le temps précédent «*previous time*» à l'heure actuelle et retourne 0.0 , indiquant qu'il n'y a pas d'événements à venir.

Si la taille de la liste n'est pas vide, cela signifie qu'il y a des cloudlets en cours d'exécution. L'algorithme vérifie si certains cloudlets ont terminé leur exécution en vérifiant le temps restant «*remaining*» pour chaque cloudlet. Si le temps restant est égal à 0, cela signifie que le cloudlet a terminé son exécution et il est ensuite supprimé de la liste d'exécution.

III.3.4. Processus de retour des résultats

Après avoir effectué l'exécution du cloudlet, le processus de retour entre en jeu. L'objectif de cet méthode est de déterminer quels cloudlets ont terminé leur exécution. Dans ce diagramme, nous allons voir comment se déroule le processus de retour des cloudlets, appelé «*ProcessCloudletReturn*».



III.27 Diagramme de processus de retour des résultats

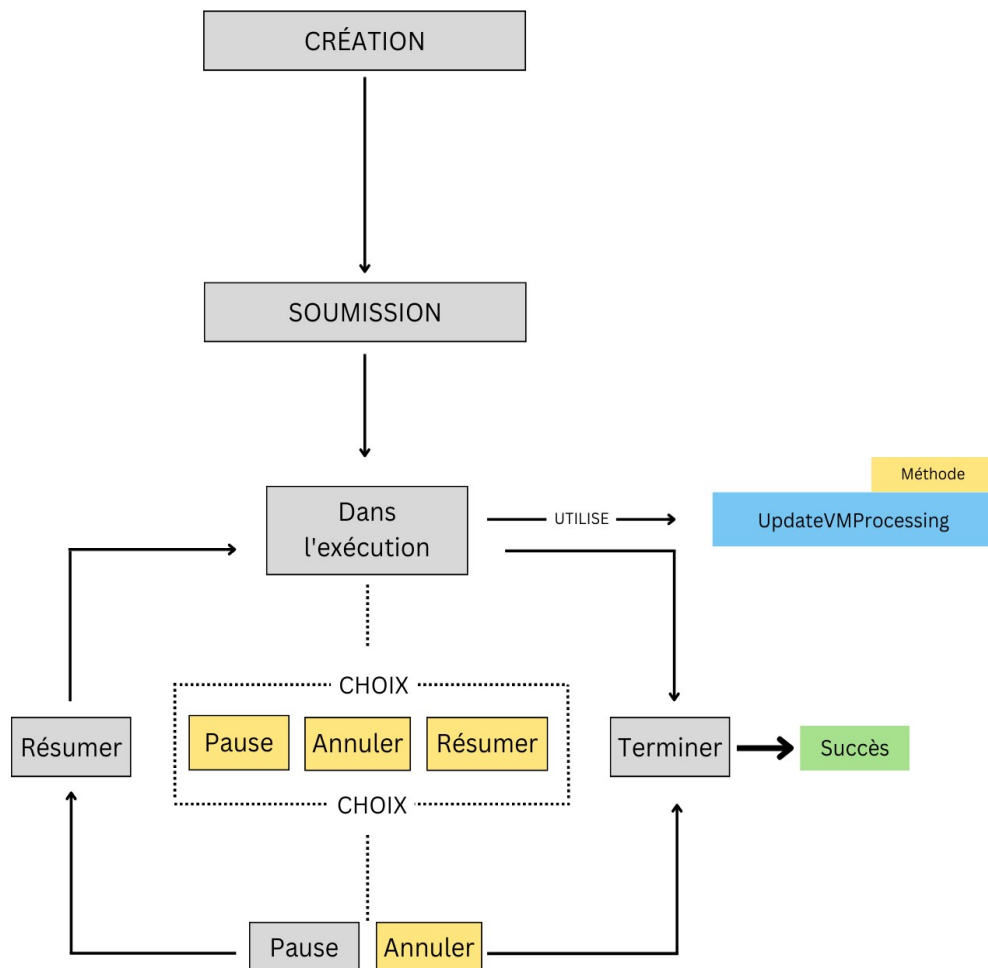
L'algorithme examine les cloudlets exécutés afin de faire une vérification pour déterminer lesquels ont terminés leur exécution. Si cette vérification est positive, il affiche un message indiquant que tous les cloudlets ont terminé leur exécution.

Dans le cas où la vérification indique que certains cloudlets n'ont pas terminé leur exécution, cela implique qu'il y a des cloudlets en attente de leur exécution complète. L'algorithme effectue alors une vérification supplémentaire pour confirmer que ces cloudlets n'ont pas encore terminé leur exécution.

Si cette vérification confirme que les cloudlets sont toujours en cours d'exécution, l'algorithme envoie un message indiquant que tous les cloudlets envoyés n'ont pas encore terminé leur exécution.

III.3.5. Cycle de vie d'un cloudlet

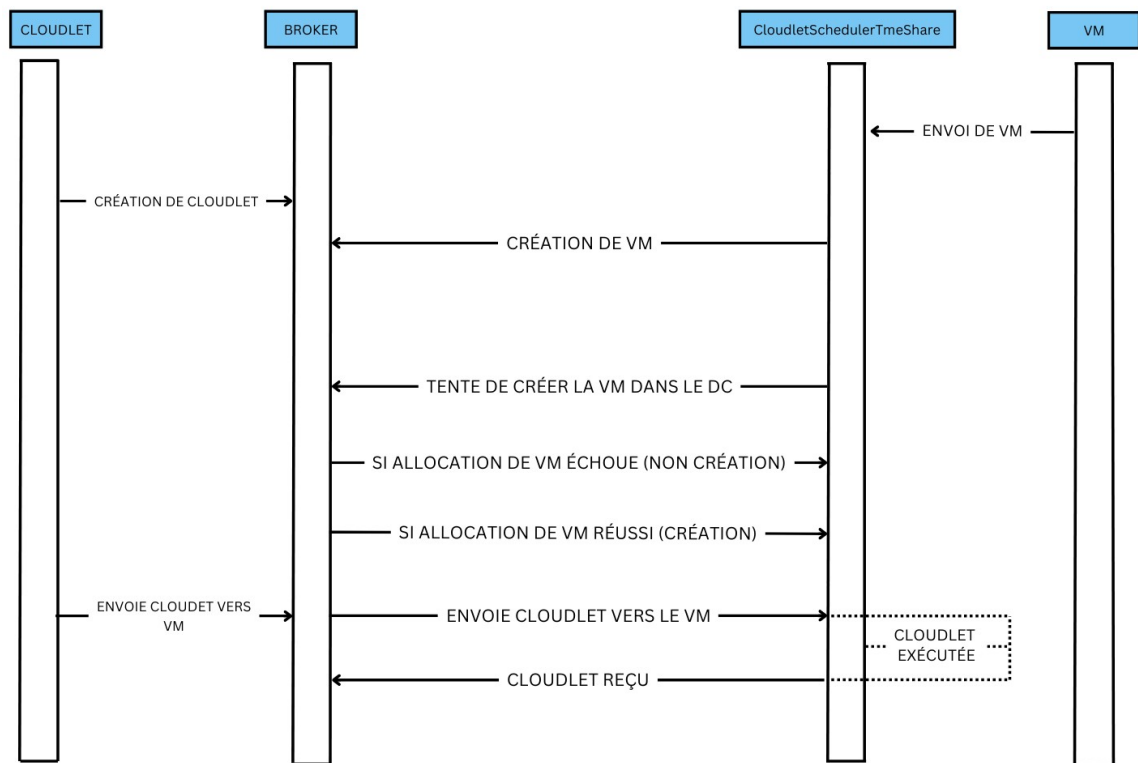
Ces étapes représentent un cycle de vie typique d'un cloudlet :



III.28 Diagramme de cycle de vie d'un cloudlet

- Création : Cette étape consiste à créer le cloudlet avec ses caractéristiques telles que « filesize, length...»
- Soumission du cloudlets : On appelle cette étape avant l'exécution, car on attribue des ressources spécifiques « Allocation des ressources» au cloudlet afin de l'exécuter.
- Exécution : Il s'agit de la dernière étape où le cloudlet commence l'opération d'exécution. L'algorithme contient des méthodes appelées « *updateVMProcessing* » qui fournissent des indicateurs « progression » sur l'état de chaque cloudlet pendant l'exécution. Pendant l'exécution, le cloudlet peut passer par plusieurs états tels que la pause ou résumer et l'annulation, en fonction de l'état de l'exécution.

III.3.6. Diagramme de séquence



III.29 Diagramme de séquence

Ce diagramme démontre la relation et les échanges entre les VM « machines virtuelles », les cloudlets et le broker.

Le broker est responsable de la création des VM et des cloudlets. Ensuite, il envoie la VM au « *CloudletSchedulerTimeShare* ». Le broker tente ensuite de créer des VM dans le datacenter, en respectant la condition d'allocation de VM.

Cette condition permet d'allouer les ressources physiques de chaque VM. De l'autre côté, le cloudlet est envoyé à la VM via le broker, qui effectue une ordonnance du cloudlet vers sa VM spécifique.

Une fois que cela est fait, le cloudlet est envoyé vers la VM. Après avoir effectué et terminé les tâches du cloudlet, celui-ci envoie un message de "cloudlet Reçu".

III.4. Modèle proposé

III.4.1. Introduction

Afin de minimiser le temps d'exécution, il est essentiel de maîtriser et connaître les caractéristiques des machines virtuelles « VM » et des cloudlets.

Nous avons choisi cet algorithme qui permet de faire un tri des VM ainsi que les cloudlets en fonction de leurs caractéristiques afin de planifier et d'ordonnancer chaque cloudlet dans la VM appropriée.

Tout cela contribue à minimiser le temps d'exécution des tâches et à utiliser efficacement les ressources.

Cette comparaison s'effectue en calculant la moyenne des métriques de chaque VM ainsi que chaque cloudlets afin de déterminer leur performance élevée et basse. Cela nous permet de mieux optimiser leur utilisation.

En utilisant cet algorithme, nous espérons améliorer l'efficacité globale du système et optimiser les performances des tâches exécutées.

III.4.2. Classification des machines virtuelles.

La classification des machines virtuelles « VM » à travers ces métriques peut être effectuée pour évaluer et comparer différentes VM en fonction de leurs caractéristiques et de leurs performances. Les métriques utilisées sont les suivantes:

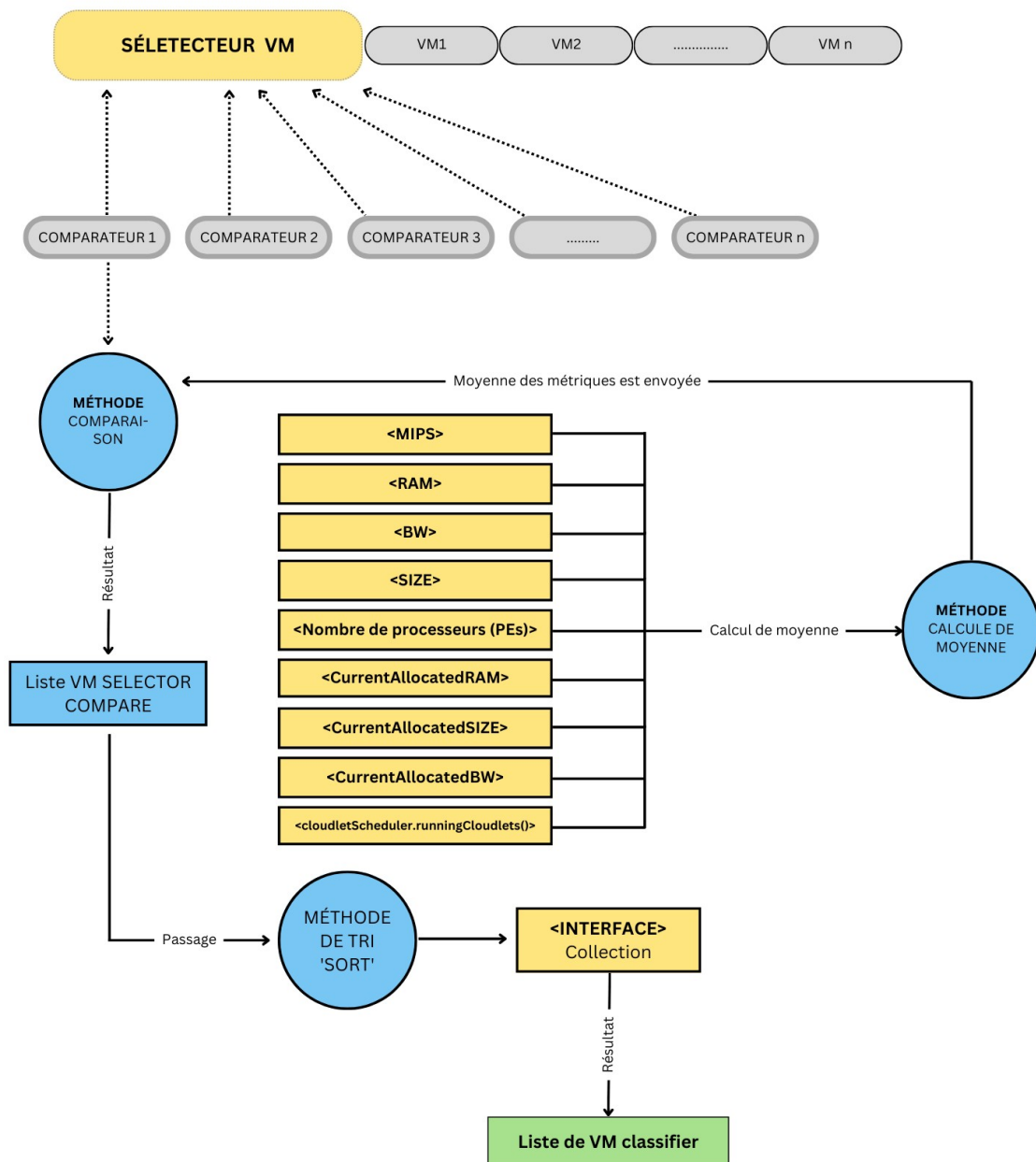
- MIPS « Million Instructions Par Seconde » : Mesure la capacité de traitement de la VM en termes d'instructions exécutées par seconde. Une valeur plus élevée indique une meilleure performance.
- Nombre de processeurs PEs « Processing Elements » : Indique le nombre de cœurs de processeur alloués à la VM. Un nombre plus élevé de PEs peut permettre une puissance de calcul donc une exécution de tâches plus efficace.

- Mémoire allouée « RAM» : La quantité de RAM allouée à une VM est essentiel pour les performances générales. Une plus grande quantité de RAM permet à la VM de stocker plus de données en mémoire, réduisant ainsi la nécessité d'accéder fréquemment au stockage de données.
- CloudletScheduler.runningCloudlets() : Nombre actuel de cloudlets en cours d'exécution dans la VM multiplié par 2. Un nombre plus élevé peut indiquer une utilisation intensive des ressources de la VM.
- Bande passante allouée « BW» : La bande passante allouée à une VM est importante pour les communications réseau. Une puissante bande passante permet une transmission plus accéléré des données entre la VM et les autres systèmes.
- Taille de la VM : Fait référence à l'espace de stockage qu'elle occupe. Une plus grande taille de VM peut indiquer qu'elle est capable de prendre en charge des applications ou des systèmes plus complexes.

Une VM avec un grand nombre d'unités de ces métrique peut être considérée comme hautement performante et adaptée aux charges de travail exigeantes. D'autre part, une VM avec des ressources plus limitées peut être classée comme étant plus adaptée à des charges de travail moins intensives.

III.4.2.1. Algorithme de classification des machines virtuelles.

Ce diagramme représente les étape à suivre afin de réaliser une classification des machines virtuelles :



III.30 Diagramme de l'algorithme de classification des machines virtuelles (VM)

Étape 1 : Tout d'abord, l'algorithme sélectionne un groupe de machines virtuelles « VM » à partir de la liste des VM existant dans le datacenter. On appelle cette liste les sélecteurs de VM.

Étape 2 : Dans cette étape, le rôle du comparateur entre en jeu. Ce dernier appelle sa méthode nommée « COMPARE » qui effectue le travail de comparaison. Cette comparaison est réalisée en après l'utilisation d'une autre méthode qui calcule la moyenne des métriques pour chaque VM.

La moyenne est calculée en utilisant les métriques suivantes :

- MIPS « millions d'instructions par seconde » .
- SIZE « Taille » .
- Le nombre de processeurs « PEs » .
- Mémoire RAM.
- Bande passante.
- CloudletScheduler.runningCloudlet.
- CurrentAllocatedRAM.
- CurrentAllocatedSIZE.
- CurrentAllocatedBW.

Après avoir terminé, il appelle la méthode «*Compare(VM1, VM2, Vmn)*» les résultats sont ensuite placés dans une liste nommée «*VM SELECTOR COMPARE*» Cette liste est également envoyée dans une méthode de tri «*methodeOfcollectionclass()*».

Étape 3 : Dans cette dernière étape, le résultat du tri effectué à l'étape précédente est collecté, puis envoyé dans l'interface. Ensuite, il est renvoyé à nouveau afin d'obtenir une classification prête des VM.

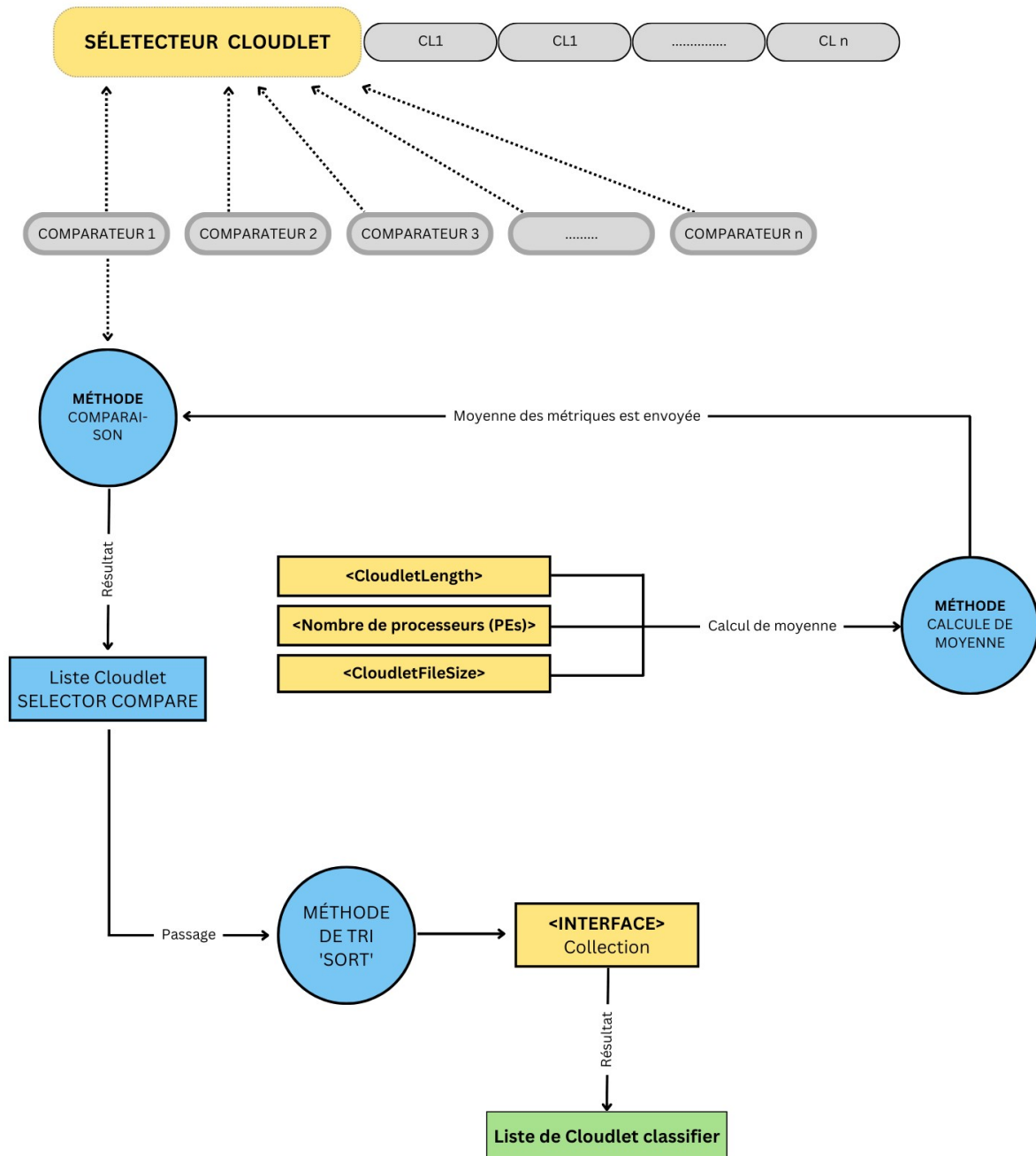
III.4.3. Classification des cloudlets

La classification des cloudlets au moyen de ces métriques permet d'évaluer et de comparer différentes instances de cloudlets en fonction de leurs caractéristiques et de leurs performances. Les métriques utilisées comprennent :

- CloudletLength : La longueur d'un cloudlet, c'est-à-dire la quantité de travail ou de tâches qu'il contient, la charge de travail ou de la complexité des opérations à exécuter dans un cloudlet.
- Nombre de processeurs « PEs » .
- CloudletFileSize : Représente la taille des fichiers associés à un cloudlet. Elle mesure la quantité de données qui doivent être transférées ou manipulées lors de l'exécution du cloudlet.

III.4.3.1 Algorithme de classification des cloudlets

Ce diagramme représente les étapes à suivre afin de réaliser une classification des cloudlets:



III.31 Diagramme de l'algorithme de classification des cloudlets

Étape 1: En premier lieu, l'algorithme effectue une sélection initiale des clodulets à partir de la liste complète des cloudlets présentes dans le datacenter. Cette liste est désignée comme les « sélecteurs de Cloudlet » .

Étape 2 : Au cours de cette étape, le comparateur joue un rôle essentiel. Il invoque la méthode appelée « **COMPARE** » qui se charge de la comparaison. Cette comparaison est réalisée à l'aide d'une méthode qui calcule la moyenne pour chaque cloudlet des métriques suivantes :

- CloudletLength.
- Le nombre de processeurs « PEs».
- CloudletFileSize.

En utilisant ces métriques, l'algorithme évalue les performances des différentes VM et classe les cloudlets en conséquence. Cela permet de regrouper les cloudlets similaires et d'optimiser leur exécution sur les VM correspondantes.

Étape 3 : Durant cette ultime phase, la sélection réalisée lors de l'étape précédente est rassemblée et transmise à l'interface. Par la suite, elle est renvoyée à nouveau pour obtenir une classification définitive des cloudlets prête à l'emploi

III.4.4. Pseudocodes des méthodes de classification

III.4.4.1 Machines virtuelles

```
# Liste des objets VM à trier
inputVMLIST = [...]

# Fonction pour calculer la moyenne des VM : ( calculate_average_vm)
return average = (vm.MIPS + vm.NumberOfPesm + vm.RAM + vm.BW + vm.SIZE +
vm.CurrentAllocatedRAM + vm.CurrentAllocatedSIZE
+ vm.CurrentAllocatedBW+(cloudletScheduler.runningCloudlets()*2)) /10

# Fonction pour compare la moyenne des VM (comparateur) :
def compare_vm(vm1, vm2):
    average1 = calculate_average_vm(vm1) ;
    average2 = calculate_average_vm(vm2) ;

    if average1 < average2:
        return -1
    else average1 > average2:
        return 1
    else
        return 0

# Trier la liste VM en utilisant le comparateur
Collection.sort(VM LIST, comparator)
# La liste de VM triée selon la note moyenne, en ordre décroissant
outputvmlist = Vmlist
```

III.1 Tableau de pseudocodes de la méthode des VM

III.4.4.2 Cloudlets

```
# Liste des objets cloudlets à trier
inputCLOUDLETLIST = [...]

# Fonction pour calculer la moyenne des cloudlets
return average = (cloudlet.NumberOfPes + (cloudlet.cloudletLength*2) +
cloudlet.cloudletFile) / 4

# Fonction pour comparer les cloudlets en utilisant return la fonction de la moyenne
(comparateur) :
def compare_cloudlet(cloudlet1.average, cloudlet2.average):
    average1 = calculate_average_cloudlet(cloudlet1)
    average2 = calculate_average_cloudlet(cloudlet2)

    if average1 < average2:
        return -1
    else average1 > average2:
        return 1
    else:
        return 0

# Trier la liste CloudletList en utilisant le comparateur
Collection.sort(cloudlet LIST, comparator)

# La liste de cloudlets triée selon la note moyenne, en ordre croissant
outputcloudletlist = CloudletList
```

III.2 Tableau de pseudocodes de la méthode des Cloudlets

III.5. Conclusion

Dans ce chapitre, nous avons examiné en détail l'algorithme optimisé que nous avons proposé afin résoudre le problème de minimisation du temps d'exécution et de la charge de travail des cloudlets et des machines virtuelles.

Dans le prochain chapitre, nous allons explorer plus en détail la présentation des outils de développement que nous avons utilisés ainsi que les résultats des tests de notre algorithme avec une évaluation de ces derniers.

CHAPITRE IV

IMPLÉMENTATION ET RÉALISATION

IV.1. Aperçu

Au cours de ce mémoire, Nous avons abordé l'importance cruciale de l'ordonnancement des tâches dans les applications cloud.

Après avoir terminé les phrases d'analyse et de conception, Ce chapitre représente la conclusion de ce mémoire, où nous présentons l'implémentation de l'algorithme et les tests réalisés afin d'évaluer ses performances

En premier temps, nous introduisons les outils employés pour le développement de notre projet. Ensuite, nous procéderons à l'analyse des résultats obtenus à l'issue de l'exécution de notre travail.

IV.2. Introduction

L'ordonnancement des tâches est un aspect crucial des applications cloud. La recherche traite des problèmes rencontrés dans un environnement réel est extrêmement complexe qui échappent au contrôle des développeurs.

C'est pourquoi des simulateurs ont été développés pour permettre une meilleure compréhension et application des avancées dans le domaine des simulateurs de Cloud computing.

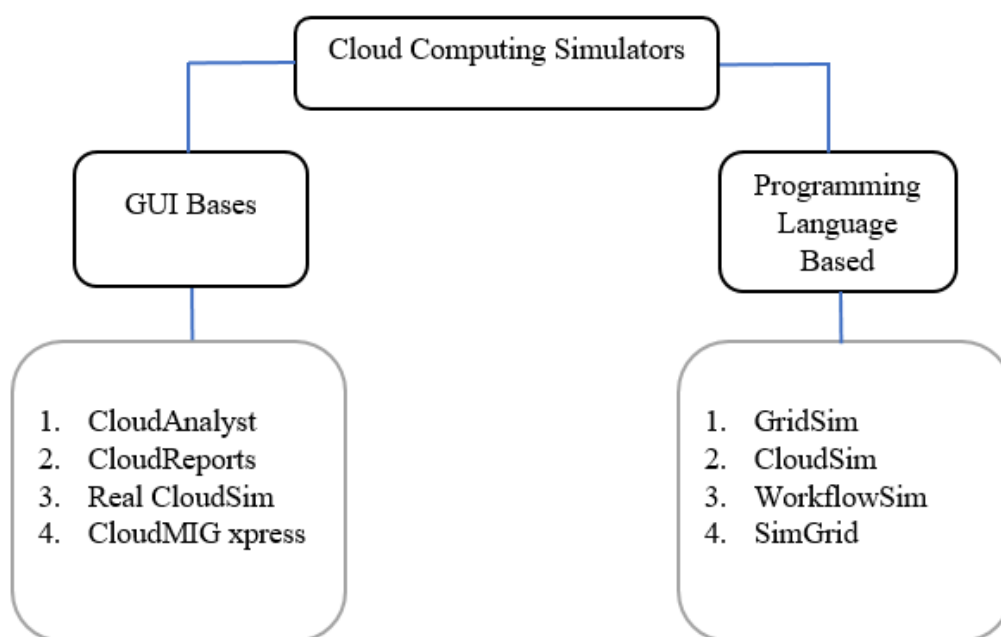
Ces simulateurs fournissent un environnement virtuel permettant d'étudier et d'expérimenter différentes stratégies d'ordonnancement des tâches, afin d'optimiser les performances et l'efficacité des applications cloud.

IV.3. Simulation de Cloud computing

La technologie du Cloud Computing est confrontée à de nombreux problèmes dans divers secteurs, directement ou indirectement liés aux technologies de l'information. Ce domaine connaît une évolution rapide grâce à d'importants acteurs tels que Microsoft, Google et Amazon qui investissent massivement pour obtenir des améliorations.

Afin de mieux comprendre et optimiser ces environnements, divers outils de simulation ont été développés pour modéliser et simuler des environnements de Cloud Computing.

En général, on peut distinguer deux types de simulateurs : les simulateurs dotés d'une interface utilisateur graphique « GUI » et les simulateurs basés sur des langages de programmation tels que Java, Python par exemple [37].



IV.32 Diagramme de divers types de simulateurs dans cloud computing [37]

Dans ce travail, nous avons délibérément opté pour l'utilisation du simulateur CloudSim en raison de sa facilité d'utilisation ainsi que de la disponibilité des ressources et de plus, l'accès à une multitude de ressources en ligne, telles que des tutoriels, des forums de discussion et des exemples de code.

IV.4. Les outils de développement

IV.4.1. Présentation du langage de programmation (JAVA)

Le langage de programmation Java est un langage orienté objet largement utilisé dans le domaine informatique. Sa caractéristique principale est sa portabilité, ce qui signifie qu'il peut être facilement exécuté sur différentes plateformes, telles que Windows, Linux et macOS, sans avoir à être réécrit pour chaque système d'exploitation.

Cette portabilité est rendue possible grâce à la machine virtuelle Java, qui permet à une application Java de s'exécuter sur n'importe quelle plateforme compatible Java, sans nécessiter de modifications de code supplémentaires [38].

IV.4.2. Présentation de l'environnement de développement IDE ECLIPSE

Eclipse est une plateforme de développement logiciel gratuite, open source, hautement personnalisable et flexible qui prend en charge un de larges langages de programmation. Avec une interface utilisateur intuitive [42] et des fonctionnalités avancées telles que la complétion automatique de code, la détection d'erreurs de syntaxe et le débogage,

IV.4.2.1. Les avantages de l'environnement de développement IDE ECLIPSE

- Personnalisation et extensibilité
- Support multiplateforme
- Intégration avec des outils de développement

IV.4.3. Présentation de l'outil de simulation Cloudsim

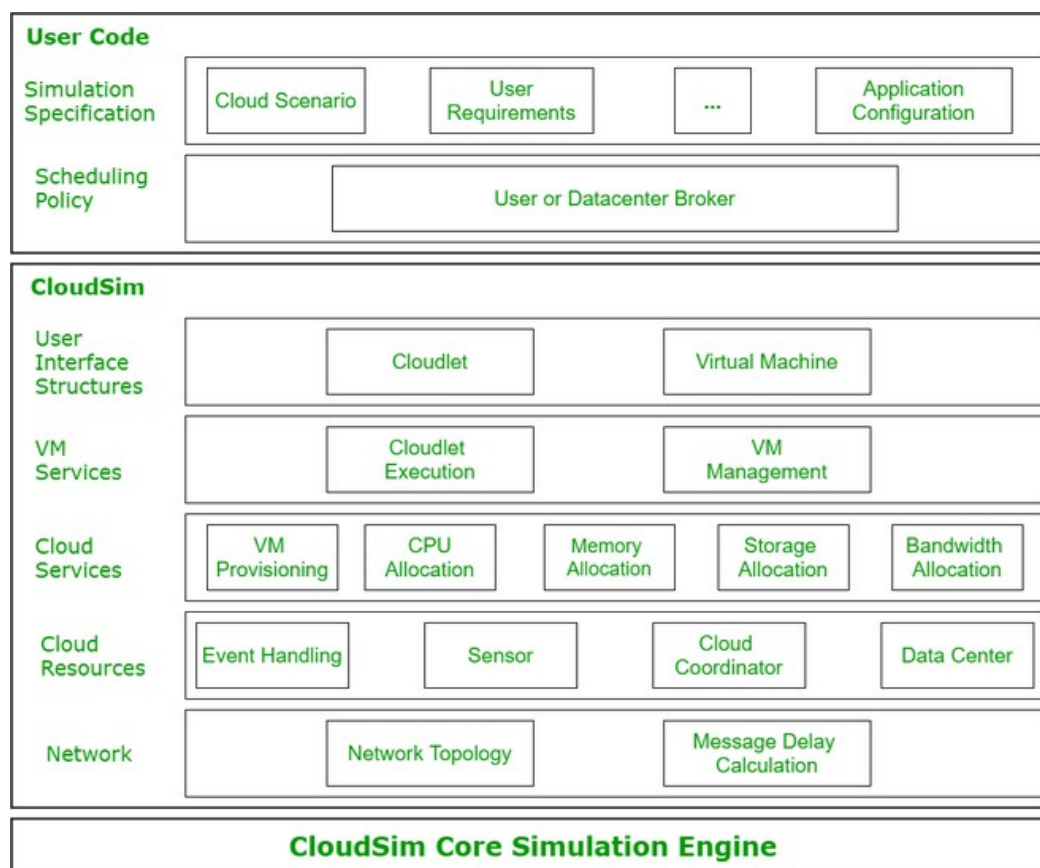
Le framework CloudSim est un outil de simulation de Cloud computing largement utilisé dans la recherche et l'industrie pour évaluer les performances de diverses stratégies d'ordonnancement de tâches dans des environnements Cloud.

CloudSim fournit un ensemble de bibliothèques qui permettent aux développeurs de créer des modèles de Cloud personnalisés en fournissant des composants de l'infrastructure de Cloud tels que les machines virtuelles, les centres de données, les réseaux et les politiques d'ordonnancement [39]. Les modèles créés dans CloudSim peuvent être utilisés pour évaluer la performance des algorithmes de gestion de ressources, pour estimer les coûts d'utilisation du Cloud et pour analyser la consommation des ressources de la plateforme Cloud.

IV.4.3.1. Les avantages d'utilisation de Cloudsim

- L'élasticité des configurations: CloudSim permet de modéliser des configurations significatives, offrant ainsi une flexibilité dans la gestion des ressources cloud.
- Prise en charge des environnements cloud variés : CloudSim est compatible avec différents environnements cloud, ce qui permet de simuler des scénarios dans divers contextes [39].
- Implémentation rapide des applications basées sur le cloud : CloudSim permet de mettre en œuvre des applications basées sur le cloud en un minimum de temps et d'efforts, accélérant ainsi le processus de développement.
- Facilité d'utilisation et personnalisation : CloudSim est convivial et peut être personnalisé selon les besoins spécifiques, offrant ainsi une expérience de simulation agréable et adaptée aux utilisateurs.

IV.4.3.2. Architecture de Cloudsim



IV.33 Figure de L'architecture de simulateur Cloudsim [39]

Le schéma illustré ci-dessus représente l'architecture en couches de l'outil de simulation CloudSim.

- CloudSim CoreSimEngine qui est le moteur de simulation central est responsable de la modélisation et de la simulation d'environnements de centres de données virtualisés basés sur le Cloud. [39] Ses fonctionnalités principales incluent l'ordonnancement et le traitement des événements.
- La couche CloudSim fournit des interfaces de gestion dédiées pour les machines virtuelles, la mémoire, le stockage et la bande passante. Elle gère également d'autres problèmes fondamentaux, tels que l'approvisionnement des hôtes pour les machines virtuelles, la gestion de l'exécution des applications.
- Le code utilisateur «User Code» est la couche personnalisée dans l'architecture de CloudSim où les utilisateurs peuvent écrire leur code pour définir les caractéristiques spécifiques de l'environnement de simulation selon leurs besoins et leurs nouvelles découvertes en recherche[39] .

IV.4.3.3 La structure des packages et les classes de Cloudsim

Dans CloudSim, les packages et les classes sont organisés de manière hiérarchique pour faciliter la gestion et l'organisation du code. Voici certains des packages et des classes principaux dans CloudSim :

IV.4.3.4. Package "org.cloudbus.cloudsim"

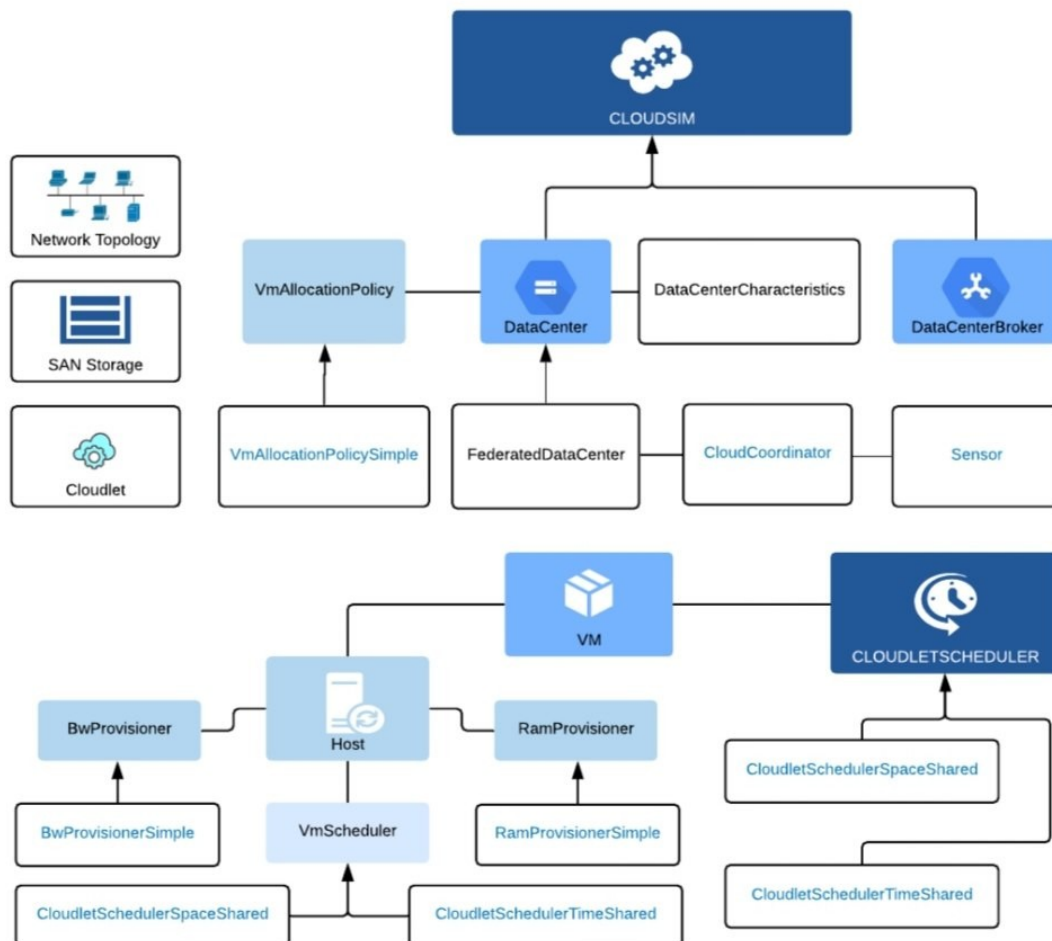
- Classe CloudSim : Classe principale de CloudSim, responsable de l'initialisation et du déroulement de la simulation.
- Classe Datacenter : Représente un centre de données dans le cloud.
- Classe Host : Modélise un hôte physique dans un centre de données.
- Classe VM : Représente une machine virtuelle.
- Classe VmAllocationPolicy : Fait référence à une classe qui définit une politique d'allocation des machines virtuelles «VM» aux hôtes physiques.
- Classe Cloudlet : Modélise une tâche ou une charge de travail exécutée sur une machine virtuelle.
- Classe DatacenterBroker : Cette classe est responsable de la gestion des demandes des utilisateurs, de l'allocation des ressources et de l'ordonnancement des tâches, Elle agit en tant qu'intermédiaire entre les utilisateurs du cloud «Cloud Users» et les ressources du cloud «datacenters, hosts et vms»[42] .

IV.4.3.5. Package "org.cloudbus.cloudsim.core"

- Classe PeProvisioner : Provisionneur de puissance utilisé pour allouer la capacité de traitement d'un hôte à une machine virtuelle.
- Classe RamProvisionerSimple : Provisionneur de mémoire utilisé pour allouer la mémoire d'un hôte à une machine virtuelle [42] .
- Classe BwProvisionerSimple : Provisionneur de bande passante utilisé pour allouer la bande passante d'un hôte à une machine virtuelle.
- CloudletSchedulerTimeShared : Une implémentation de l'ordonnancement des Cloudlets basée sur le partage du temps, où les Cloudlets partagent le temps de traitement de la machine hôte.
- CloudletSchedulerSpaceShared : Une implémentation de l'ordonnancement des Cloudlets basée sur le partage de l'espace, où les Cloudlets reçoivent une allocation spécifique de ressources «telles que les PEs, la mémoire» [42].

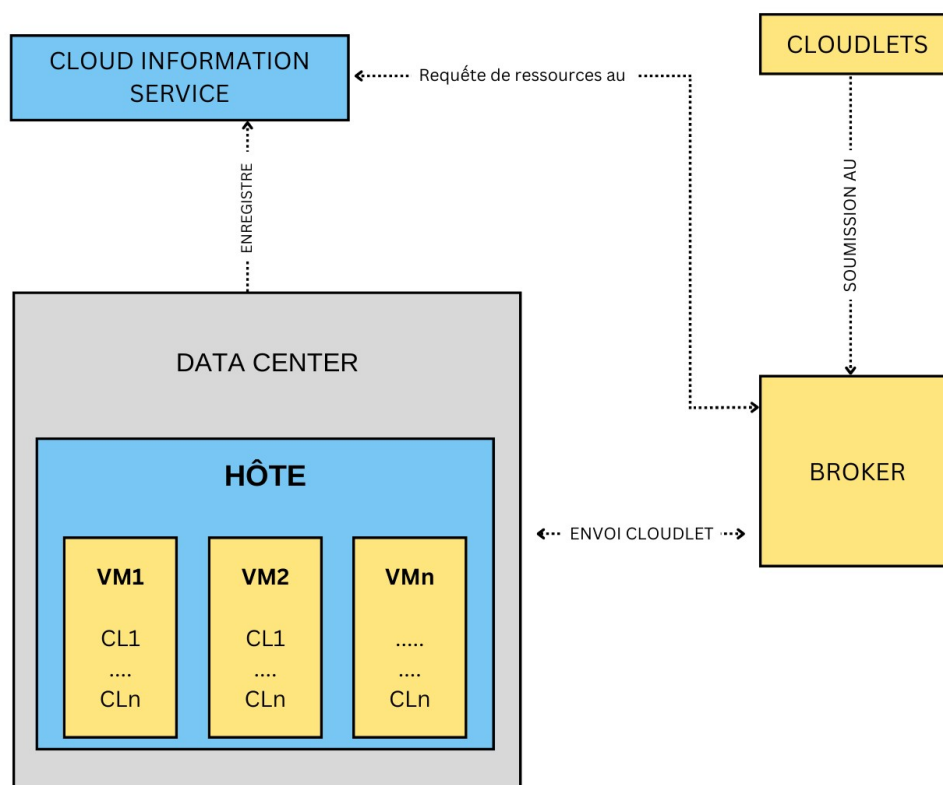
IV.4.3.6. Package "org.cloudbus.cloudsim.schedulers"

- VmScheduler : Classe qui fournit les méthodes et les attributs de base pour implémenter des planificateurs spécifiques.
- Classe VmSchedulerTimeShared : Planificateur de machines virtuelles utilisant une politique de temps partagé.
- Classe VmSchedulerSpaceShared : Planificateur de machines virtuelles utilisant une politique d'allocation d'espace partagé [42] .



1.IV Diagramme des principales classes dans Cloudsim [41]

IV.4.4. Scénario basique de simulation

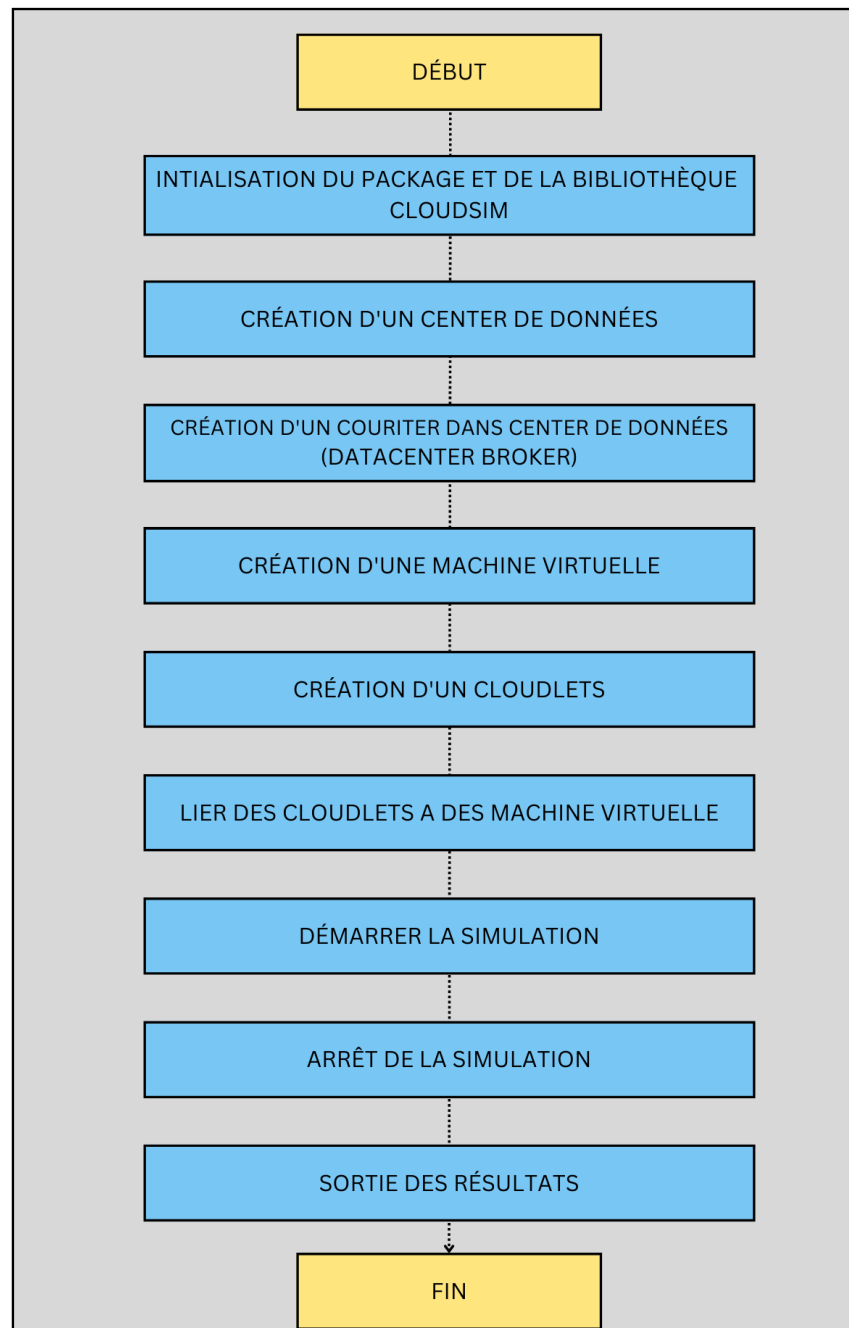


IV .35 Diagramme de scénario principal de la simulation CloudSim

Le scénario principal de la simulation CloudSim est présenté dessus ce scénario peut être décrit en détail pour chaque composant de CloudSim comme suit [41] :

- Centres de données «datacenter» représente l'environnement de cloud computing. C'est là où une liste d'hôtes sera créée avec différentes configurations pour chaque hôte, telles que la bande passante, la mémoire, le stockage et les MIPS.
- Cloudlets : ils agissent comme des demandes d'application « tâches utilisateur entrantes ». Selon l'algorithme proposé, différents paramètres seront pris en compte, tels que la date limite, la durée «Taille » et l'heure d'arrivée. Les cloudlets sont utilisés pour représenter les demandes des utilisateurs.
- Le courtier « Broker » possède des caractéristiques de centre de données «datacenter» qui lui permettent de soumettre des machines virtuelles à l'hôte exact [39].
- Le service d'information sur le cloud « CIS » est responsable de la liste des ressources, de l'indexation ainsi que de la découverte de l'efficacité des centres de données.

IV.4.5. Différentes étapes du cycle de vie de CloudSim



IV .36 Figure des différentes étapes du cycle de vie de CloudSim

IV.5.Réalisation et tests

IV.5.1. Outils de travail

- Une machine avec un processeur Intel® I5 7th Gen 2.16Ghz.
- 8 Gigas de mémoire vive.
- OS : Win10 Professional 64bits

IV.5.2. Paramètres de simulation

IV.5.2.1. Paramètres de simulation des machines virtuelles

Ce tableau en dessous représente les paramètres des machines virtuelles utilisé lors des tests :

PARAMÈTRE	VALEUR
ID	20
MIPS	1000
PEs	2
RAM	2048
BW	1000
TAILLE (IMAGE SIZE)	1000
VMM	'Xen'

IV .1 Tableau des paramètres de simulation des machines virtuelles

IV.5.2.2. Paramètres des PEs

Les paramètres des Processing Elements (PEs) sont présenter dans le tableau suivant :

ID PEs	MIPS	LISTE PEs
1	1500	1
2	2000	1
3	2500	1
4	1700	1

IV .2 Tableau des paramètres des PEs

IV.5.2.3. Paramètres de simulation des cloudlets

Le tableau ci-dessous présente les différents paramètres spécifiques aux cloudlets qui ont été utilisés lors des tests :

PARAMÈTRES	VALEURS
CLOUDLET ID	40
CLOUDLET LENGTH	1000
CLOUDLET OUTPUT SIZE	300
PEs	2
MODÈLE D'UTILISATION	MODÈLE CPU
MODÈLE D'UTILISATION	MODÈLE RAM
MODÈLE D'UTILISATION	MODÈLE BW

IV .3 Tableau des paramètres de simulation des cloudlets

IV.5.2.4. Paramètres des hôtes

Les paramètres des hôtes sont présenter dans le tableau suivant :

ID HÔTES	RAM	STOCKAGE	BW	LISTE PEs	ID PEs
0	2048	1000 000	1000	1	1
1	2048	1000 000	2000	2	1
3	2048	1000 000	1000	1	4
4	1024	10 000	1000	1	3

IV .4 Tableau des paramètres des hôtes

IV.5.3. Résultats de la simulation

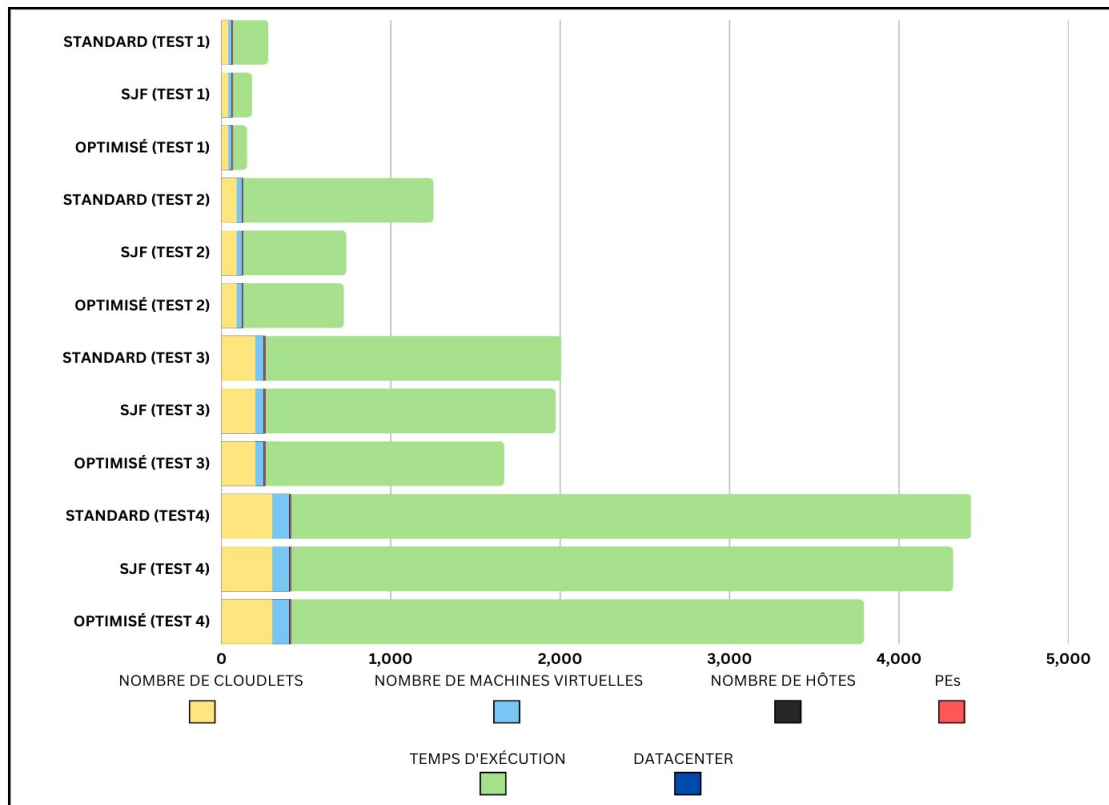
Dans le tableau et le graphique ci-dessous, nous présentons une comparaison entre notre algorithme et deux autres algorithmes , à savoir :

- L'algorithme standard sans les méthodes de classification.
- L'algorithme SJF (Shortest Job First).

Le tableau et le graphique fournissent des données sur cinq tests qui permettent d'évaluer les performances et les résultats obtenus par chaque algorithme.

ALGORITHMES	NOMBRE DE CLOUDLETS	NOMBRE DE MACHINE VIRTUELLES	NOMBRE DE HÔTES	PEs	DATA CENTER	TEMPS D'EXÉCUTION (SECONDES)
STANDARD (TEST1)	40	20	2	4	2	204.91
STANDARD (TEST2)	90	30	2	4	2	1119.75
STANDARD (TEST3)	200	50	3	6	2	1741.81
STANDARD (TEST4)	300	100	4	6	2	4012.02
STANDARD (TEST5)	1000	500	4	6	3	103397.66
SHORT JOB FIRST (TEST1)	40	20	2	4	2	109.33
SHORT JOB FIRST (TEST2)	90	30	2	4	2	605.81
SHORT JOB FIRST (TEST3)	200	50	3	6	2	1708.85
SHORT JOB FIRST (TEST4)	300	100	4	6	2	3903.58
SHORT JOB FIRST (TEST5)	1000	500	4	6	3	52135.01
L'ALGORITHME OPTIMISÉ (TEST1)	40	20	2	4	2	79.96
L'ALGORITHME OPTIMISÉ (TEST2)	90	30	2	4	2	590.96
L'ALGORITHME OPTIMISÉ (TEST3)	200	50	3	6	2	1404.71
L'ALGORITHME OPTIMISÉ (TEST4)	300	100	4	6	2	3377.64
L'ALGORITHME OPTIMISÉ (TEST5)	1000	500	4	6	3	51028.46

Tableau IV .5 Résultats de la simulation entre différents algorithmes



IV.37 Figure des résultats de la simulation

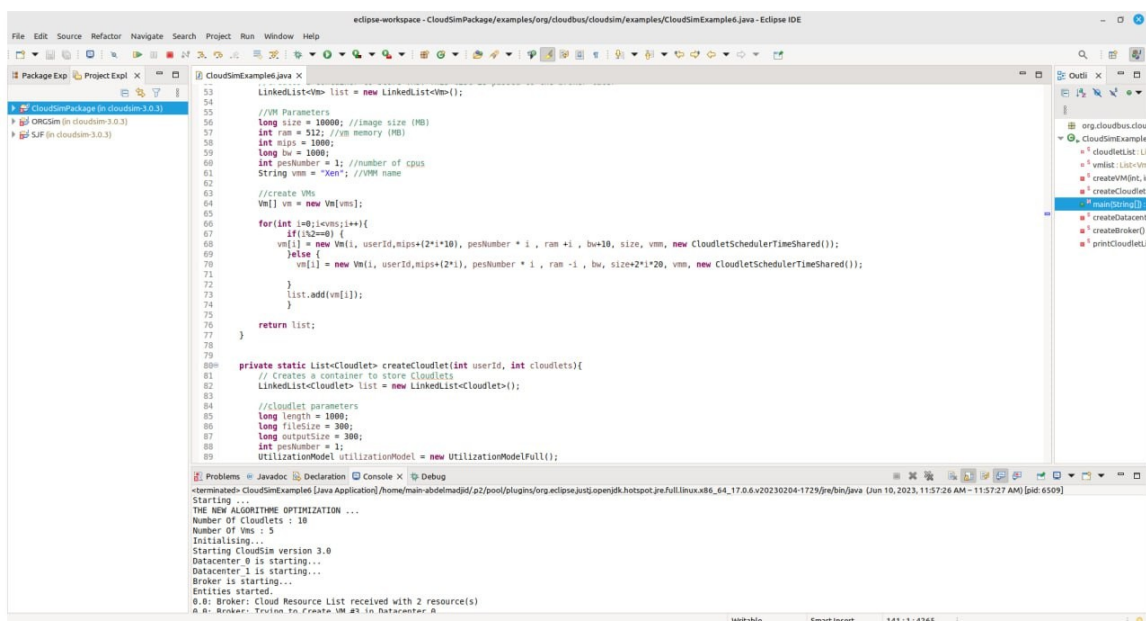
IV .38 Figure des résultats de la simulation (Temps d'exécution / Cloudlets)



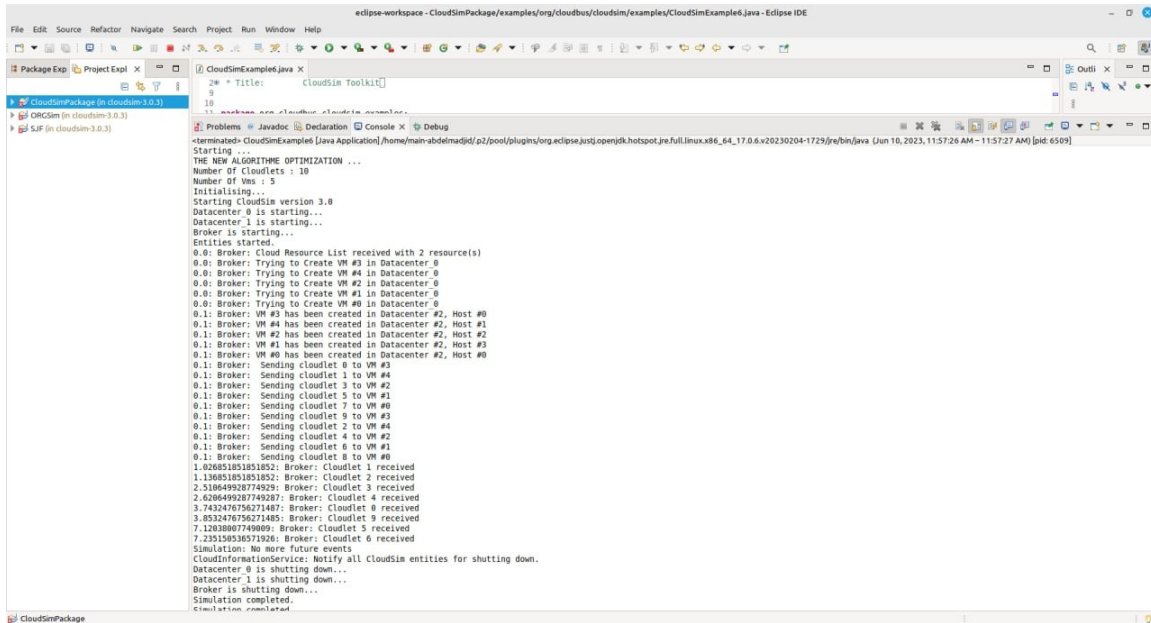
En analysant ces résultats, nous constatons une tendance générale :

- Chaque itération du tableau montre une amélioration « minimisation » du temps d'exécution lorsque le nombre de cloudlets et de machines virtuelles « VM » est augmenté.

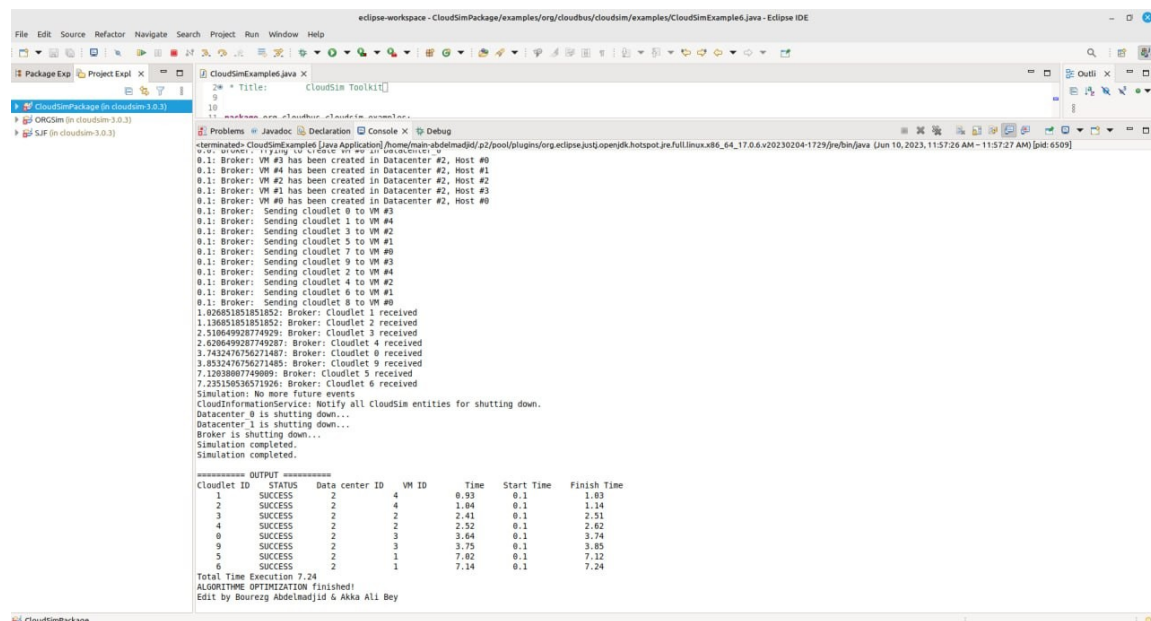
Ces résultats mettent en évidence l'efficacité de notre algorithme optimisé par rapport aux autres approches, ce qui indique que notre approche d'optimisation a permis d'améliorer significativement l'efficacité de l'ordonnancement des tâches. En regroupant les VM et les cloudlets de manière optimale.



IV.39 Figure de capture d'écran de quelques tests (1)



IV.40 Figure de capture d'écran de quelques tests (2)



IV.41 Figure de capture d'écran de quelques tests (3)

IV.6 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre de notre méthode d'optimisation pour l'ordonnancement des tâches dans le cloud computing, qui vise à minimiser le temps d'exécution. Nous avons commencé par présenter les outils utilisés et l'environnement de développement sur lesquels notre solution a été mise en œuvre. Ensuite, nous avons affiché les résultats de notre solution, qui consiste en la classification des machines virtuelles « VM » et des cloudlets, en effectuant une comparaison avec deux autres algorithmes afin d'évaluer son efficacité.

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

Le cloud computing, ou l'informatique en nuage, est devenu l'un des piliers fondamentaux de la technologie moderne. Il offre une infrastructure permettant d'accéder à des services informatiques à la demande via Internet.

Dans l'environnement du cloud computing, l'ordonnancement des tâches est un défi majeur à relever. Il consiste à attribuer et à planifier et ordonnancer les différentes tâches sur les ressources disponibles de manière optimale, l'un des problèmes les plus importants dans ce contexte est d'optimiser le temps d'exécution des tâches et l'utilisation des ressources.

Cette étude s'est concentrée sur l'amélioration de cet ordonnancement et en introduisant des ajouts et des mécanismes. L'objectif était d'obtenir une solution qui permettrait d'utiliser de manière optimale les ressources dans un délai optimal.

Un algorithme d'optimisation a été proposé et appliqué pour améliorer les performances en termes de temps d'exécution et d'utilisation des ressources. Pour évaluer les résultats de cette optimisation, l'algorithme a été mis en œuvre dans un environnement CloudSim, qui simule un environnement cloud réaliste.

Les expériences menées en utilisant cet algorithme amélioré pour planifier un certain nombre de Cloudlets sur un groupe de machines virtuelles (VMs) ont révélé une différence significative en termes de temps d'exécution par rapport à l'algorithme SJF et à l'algorithme standard.

Les systèmes cloud continueront d'être un domaine de recherche où divers problèmes se poseront pendant longtemps. Pour cette raison, il est nécessaire de se concentrer sur les solutions d'optimisation de diverses méthodes dans les systèmes cloud dans leurs futures études

RÉFÉRENCES

- [1] Wikipedia- https://en.wikipedia.org/wiki/Cloud_computing , consulté le : 02/03/2023
- [2] Tim Mather, Subra Kumaraswamy, Shahed Latif, Cloud Security and Privacy
Published by O'Reilly Media, Inc. , September 2009.
- [3] Thomas Erl, Ricardo Puttini, Zaigham Mahmood , Cloud Computing: Concepts,
Technology & Architecture Published by PearsonCloud Computing , May 2013.
- [4] Fang Liu, Jin Tong, Jian Mao , Robert Bohn, John Messina, Lee Badger and ,Dawn Leaf
NIST Cloud Computing Reference Architecture «National Institute of Standards and
Technology» , September 2011.
- [5] Knowledgehut online training -
<https://www.knowledgehut.com/blog/cloud-computing/cloud-computing-reference-model>
consulté le : 05/06/2023
- [6] Atlassian Software company - <https://www.atlassian.com/microservices/cloud-computing>
consulté le : 03/06/2023
- [7] Virtualization 2.0 For Dummies, published by Wiley Publishing, Inc, Indianapolis, Indiana,
2009. <http://andic.co.nz/wp-content/downloads/humour/virtualization-2edition.pdf>
- [8] Blaisethirard blog - <https://blog.blaisethirard.com/tag/cloud-2/> , consulté le : 14/05/2023
- [9] Martin Irvine, “Big Ideas” in Technology and What They Mean: AI to The Cloud
CCTP-607, Communication, Culture & Technology Program, Georgetown University, 2019.
- [10] K. Bessai, Gestion optimale de l'allocation des ressources pour l'exécution des processus
dans le cadre du Cloud, Doctorat, Ecole Doctorale Management Panthéon-Sorbonne
Published 2 December 2014.
- [11] Nutanix Computer software company – <https://www.nutanix.com/fr/info/distributed-storage> , consulté le : 01/06/2023
- [12] SimpleLearn Online courses - <https://www.simplilearn.com/tutorials/cloud-computing-tutorial/cloud-computing-architecture> , consulté le : 05/06/2023.

[12] SimpleLearn Online courses - <https://www.simplilearn.com/tutorials/cloud-computing-tutorial/cloud-computing-architecture> , consulté le : 05/05/2023.

[13] TutorialandExemple <https://www.tutorialandexample.com/cloud-computing-infrastructure> , consulté le : 12/05/2023.

[14] K. Chandrasekaran, Essentials of Cloud Computing Edition 1 , Chapman and Hall, New York Published 2014.

[15] Vmware Cloud computing company

<https://www.vmware.com/topics/glossary/content/hypervisor.html?resource=cat1299087558#cat-1299087558> , consulté le : 01/02/2023

[16] ResellerClub - <https://blog.resellerclub.com/what-is-a-hypervisor-and-how-does-it-work/> consulté le : 04/05/2023

[17] It-connect - <https://www.it-connect.fr/les-types-dhyperviseurs/> , consulté le : 05/05/2023

[18] Vmware Inc. - Understanding Full Virtualization, Paravirtualization, and Hardware

https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf , CA USA, 2007.

[19] Gregory Kipper, Diane Barrett Virtualization and Forensics A Digital Forensic Investigator's Guide to Virtual Environments, Syngress, Published 2010.

[20] Appvizer: the software comparison tool - <https://www.appvizer.fr/magazine/services-informatiques/virtualisation/type-virtualisation> , consulté le : 11/05/2023.

[21] Iconos - <https://www.ionos.fr/digitalguide/serveur/know-how/les-avantages-du-cloud-computing/> consulté le : 08/06/2023.

[22] SavoirPlus - <https://savoir.plus/cloud-computing/avantages-et-inconvenients/> ,consulté le : 08/06/2023.

[23] IBM Technology corporation - <https://www.ibm.com/fr-fr/topics/virtualization> , consulté le : 29/04/2023.

[24] Namrata Bulchandani, Uday Chourasia, Shikha Agrawal, Priyanka Dixit, Alpana Pandey, A Survey On Task Scheduling Algorithms In Cloud Computing, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 01, JANUARY, 2020.

[25] Amazon web service IT service management company - <https://aws.amazon.com/fr/what-is/service-level-agreement/> , consulté le : 05/06/2023.

[26] Oracle Computer software company - <https://www.oracle.com/database/what-is-a-cloud-database/dbaas/> , consulté le : 04/03/2023.

[27] RedHat Software Software company<https://www.redhat.com/en/topics/cloud-computing/what-is-caas> ,consulté le : 02/06/2023.

- [28] Sudheer Mangalampalli ,Ganesh Reddy Karria, Utku Kose, Multi objective trust aware task scheduling algorithm in cloud computing using whale optimization , Journal of King Saud University - Computer and Information Sciences , Volume 35, Issue 2, February 2023, Pages 791-809.
- [29] An-Ning Zhang, Shu-Chuan Chu, Pei-Cheng Song, Hui WangJeng-Shyang Pan Task Scheduling in Cloud Computing Environment Using Advanced Phasmatodea Population Evolution Algorithms, 30 April 2022.
- [30] Tao Hai, Jincheng Zhou, Dayang Jawawi , Dan Wang , Uzoma Oduah , Cresantus Biamba6, Sanjiv Kumar Jain, Task scheduling in cloud environment: optimization, security prioritization and processor selection scheme, Journal of Cloud Computing: Advances, Systems and Applications, 2023.
- [31] Gopika Venu, Vijayanand K , Task Scheduling in Cloud Computing: A Survey International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 8 Issue V May 2020.
- [32] Ahmed Subhi Abdalkafor, Khattab M. Ali Alheeti, Hybrid Approach For Scheduling Applications In Cloud Computing Environment, International Journal of Electrical and Computer Engineering (IJECE), Vol. 9, No. 4, August 2019.

- [33] Saydul Akbar Murad , Abu Jafar Md Muzahid ,Zafril Rizal, Azmi Md Imdadul Hoque , Md Kowsher, A review on job scheduling technique in cloud computing and priority rule based intelligent framework, Journal of King Saud University – Computer and Information Sciences, Volume 34, Issue 6, Part A, June 2022, Pages 2309-2331.
- [34] Microsoft Azure cloud platform - <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/> , consulté le : 03/04/2023.
- [35] Javadoc (Java documentation)
<https://www.javadoc.io/doc/org.cloudsimplus/cloudsim-plus/1.2.6/org.cloudbus/cloudsim/datacenters/class-use/Datacenter.html> ,consulté le : 02/06/2023.
- [36] Rodrigo da Rosa Righi , Scheduling Problems - New Applications and Trends, Universidade do Vale do Rio dos Sinos, Brazil, 2020.
- [37] Cloud Computing CPU Allocation and Scheduling Algorithms using CloudSim Simulator
https://www.researchgate.net/publication/336888721_Cloud_Computing_CPU_Allocation_and_Scheduling_Algorithms_using_CloudSim_Simulator/figures?lo=1
- [38] Scand - <https://scand.com/company/blog/eclipse-ide-for-java-developers/> ,consulté le : 03/02/2023.
- [39] Cloudsimtutorials - <https://www.cloudsimtutorials.online/cloudsim-simulation-toolkit-an-introduction/> ,consulté le : 12/06/2023.

- [41] Dalia Abdulkareem Shafiq, Noor Zaman Jhanjhi, Azween Abdullah, CloudSim 3.0.3 Simulator Step by Step, Experiment Finding, Mars 2021.
<https://www.researchgate.net/publication/349835588>

RÉSUMÉ

La planification des ressources et la gestion du temps sont les deux problèmes les plus importants dans le cloud computing. Une planification efficace des tâches est une priorité opérationnelle pour améliorer l'efficacité du système, la satisfaction des clients et les bénéfices commerciaux dans le cloud computing. Dans ce rapport, nous proposons un algorithme de planification basé à la fois sur l'amélioration de l'algorithme « Short Job First » et sur l'ajout de procédures et de stratégies visant à réduire le temps d'exécution. Nous avons réalisé une expérience en calculant le taux des machines virtuelles « VM » en fonction de leurs caractéristiques et en les classant par ordre décroissant. D'autre part, nous avons calculé le taux des tâches « Cloudsim » et les avons classées par ordre croissant. Pour visualiser et expliquer les résultats de l'algorithme, nous l'avons comparé à l'algorithme "Short Job First" ainsi qu'à l'algorithme utilisé par Cloudsim.

Mots clés : Cloud Computing, Algorithme d'ordonnancement, Short Job First, Machines virtuelles, Temps d'exécution, CloudSim.

خلاصة

تعد جدولة الموارد واستهلاك الوقت أهم مشكلتين في الحوسبة السحابية كما تعد جدولة المهام الناجحة إحدى الأولويات الإجرائية لزيادة كفاءة الطاقة ورضا العملاء والأرباح التجارية في الحوسبة السحابية نقدم في هذا التقرير خوارزمية جدولة تعتمد من جهة على تحسين خوارزمية « Short Job First » ومن جهة أخرى إضافة بعض الإجراءات والإستراتيجيات لتقليل وقت التنفيذ حيث قمنا بإجراء تجربة حساب معدل الأجهزة الافتراضية « VM » على حسب خصائصها وترتيبها تنازلياً، ومن جهة أخرى قمنا بحساب معدل المهام وترتيبها تصاعدياً، تم تجربة الخوارزمية المحسنة باستخدام محاكي « Cloudsim » ، وفرت حل أفضل وذلك بتقليل وقت التنفيذ.

الكلمات الدالة :

CloudSim ,Short Job First ،الحوسبة السحابية ،وقت التنفيذ ،الالات الافتراضية، خوارزمية الجدولة

ABSTRACT

Resource allocation and time management are the two most critical issues in cloud computing. Effective task scheduling is an operational priority to enhance system efficiency, customer satisfaction, and business profitability in cloud computing. In this report, we propose a scheduling algorithm based on both improving the « Short Job First » algorithm and incorporating procedures and strategies to reduce execution time. We conducted an experiment by calculating the rate of virtual machines « Vms » based on their characteristics and arranging them in descending order. Additionally, we calculated the rate of cloudlets' tasks and arranged them in ascending order. The improved algorithm was tested using the "Cloudsim" simulator and provided a better solution by reducing execution time.

Keywords : Cloud Computing, Scheduling Algorithm, Short Job First, Virtual Machines, Execution Time, CloudSim.