

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE Dr. TAHAR MOULAY SAIDA
FACULTE : TECHNOLOGIE
DEPARTEMENT : INFORMATIQUE



MÉMOIRE DE MASTER

Réseaux Informatiques et Systèmes Répartis

THÈME :

Optimisation d'Ordonnancement de Workflow Scientifique basé sur Méta-Heuristiques (GA,PSO) Sur le Cloud Computing

Présenté par :

RAHMOUNI Meriem

BAHLOUL Ahmed

Encadré par :

Dr. KOUIDRI Siham

Année Universitaire 2020-2021

Remerciment

Avant Tout, nous remercions le dieu qui nous avons aidé et nous donnons la patience pour terminer cette année scolaire que le monde traverse dans les circonstances de Corona.

Nous tenons également à remercier Dr.KOUIDRI Siham pour son soutien et ses conseils, ainsi qu'elle pour nous avons fourni une énergie positive pour mener à bien nous travail nous avoir fait l'honneur de nous encadrer durant la réalisation de ce mémoire.

Nous sommes également reconnaissants pour le temps précieux qu'elle nous a accordé, nous avons beaucoup appris à ses côtés et nous lui adressons nos gratitudes pour tout cela.

Nos plus vifs remerciements aussi à tous les membres du jury qui ont bien consacré du temps à expertiser ce modeste travail.

Enfin je n'oublions pas tous nos enseignant de cette année, car grâce à eux, nous avons fut la meilleure année et ce mémoire est le fruit de ce qu'ils nous ont présenté.

Merci à tous

...

Dédicace

Avant tout, je remercie mon Dieu qui m'a donné le succès et le courage de mener à bien ce travail.

*Je le dédie aussi à mon père et ma mère, qui ont toujours été heureux de ma réussite, je demande à Dieu d'avoir pitié d'eux, ils sont la raison de ma réussite dans ma vie, malgré leur absence, je sens leur présence 'ALLAH YARHAMHOM'.
A tous mes frères, Ali, Hadj, Snossi, khelifa et surtout krimo et hocine ses épouses et ses enfants.*

A tous mes sœurs Yamina, Azza, OM ELkhir, Karima, Nadjet ses épouses et ses enfants

A tous

*Je n'oublie pas non plus mon mari MILOUD qui me soutient..
Et surtout ma belle-sœur FATMA et mon oncle TAYEB et je n'oublier pas
MASSOUDA et oncle CHEIKH.
a toute la promotion d'informatique 2021*

*Rahmouni Meriem
Et merci*

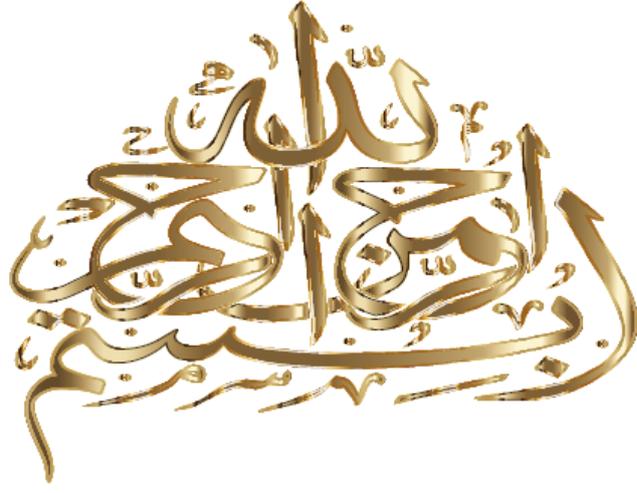
....

Dédicace

*Je dédie ce travail fruit, de plusieurs années de réflexions
A ma très chère mère et mon joyau père qui m'ont aidé par
Leurs invocations ainsi, que leurs bénédictions et leur soutien Durant toute ma vie.
A nos frères et Soeurs
A tous les familles : BAHLOUL
A tous mes Amis et spécialement à (Rahmouni Meriem ,Hocine, Mustapha,
abderazak , Guti, Redoine)
A tous l'ensemble des étudiants du département D'informatique et spécialement la
promotions LMD
Master 2020/2021*

*Bahlol Ahmed
Et merci*

....



«وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا»

سورة الإسراء

Résumé

Le Cloud Computing est une technologie de calcul et de stockage utilisé par des grandes entreprises qui utilise Workflow Scientifiques qui comprennent des centaines de tâches de calcul interconnectées selon différents modèles de dépendance. Les tâches de workflow nécessitent généralement de gros fichiers de données d'entrée et / ou exécutent un nombre extraordinaire d'instructions.

Récemment, le cloud computing s'est développé rapidement en raison de l'utilisation d'équipements avancés, de la technologie de virtualisation. De nos jours, le problème de planification des tâches est le sujet de recherche actuelle en cloud computing, La planification du flux de travail scientifique est un problème compliqué, c'est-à-dire un problème insoluble en temps polynomial avec les ressources actuelles.

- l'objectif de ce mémoire est de proposer une stratégie d'ordonnancement de workflow scientifique optimal (en terme de temps de réponse et de coût) par de méta heuristique GA et PSO.

Mots clés : Algorithme génétique, Méta-heuristique, PSO, Cloud Computing, simulation de nuage, Workflow Scientifique, Ordonnancement.

Abstract

Cloud computing is a computing and storage technology used by large companies that uses Workflow Scientists which include hundreds of interconnected computational tasks according to different dependency models. Workflow tasks typically require large input data files and / or perform an extraordinary number of instructions.

Recently, cloud computing has developed rapidly due to the use of advanced equipment, virtualization technology. Nowadays, the task scheduling problem is the subject of current research in cloud computing, Scientific workflow planning is a complicated problem, that is, an unsolvable problem in polynomial time with current resource. the objective of this thesis is to propose an optimal scientific workflow scheduling strategy (in terms of response time and cost) by GA and PSO meta heuristics.

Keywords : Genetic algorithm, Meta-heuristic, PSO, Cloud Computing, cloud simulation, Scientific Workflow, Scheduling,

ملخص:

الحوسبة السحابية هي تقنية للحوسبة والتخزين تستخدمها الشركات الكبيرة التي تستخدم سير العمل العلمي والتي تشمل المئات من المهام الحسابية المترابطة وفقاً لنماذج التبعية المختلفة. تتطلب مهام سير العمل عادةً ملفات بيانات إدخال كبيرة و / أو تنفيذ عدد غير عادي من التعليمات.

- في الآونة الأخيرة، تطورت الحوسبة السحابية بسرعة بسبب استخدام المعدات المتقدمة وتكنولوجيا المحاكاة الافتراضية. في الوقت الحاضر، تعد مشكلة جدولة المهام موضوع البحث الحالي في الحوسبة السحابية، كما يعد تخطيط سير العمل العلمي مشكلة صعبة ، أي مشكلة غير قابلة للحل في وقت متعدد الحدود مع الموارد الحالية. الهدف من هذه الطروحة هو اقتراح استراتيجية جدولة سير العمل العلمية المثلى (من حيث وقت الاستجابة والتكلفة) من خلال الاستدلال الفوقي الخوارزمية الجينية وتحسين سرب الجسيمات

الكلمات المفتاحية: الخوارزمية الجينية، الاستدلال الفوقي، تحسين سرب الجسيمات، الحوسبة السحابية ، المحاكاة السحابية، سير العمل العلمي، جدولة المهام.

Table des matières

Table des matières	11
Table des figures	13
Liste des tableaux	14
Listes des abréviations	15
Introduction générale	16
1 Etat de l'art sur le Cloud Computing	19
1.1 Introduction	20
1.2 Cloud Computing	20
1.2.1 Introduction	20
1.2.2 Définition	20
1.2.3 Caractéristiques essentielles	21
1.2.4 Les technologies de cloud computing	22
1.2.4.1 Grid Computing	22
1.2.4.2 Informatique utilitaire	22
1.2.4.3 La virtualisation	23
1.2.5 Modèles de déploiement	24
1.2.5.1 Cloud privé	24
1.2.5.2 Cloud communautaire	24
1.2.5.3 Cloud public	24
1.2.5.4 Le Cloud hybride	24
1.2.6 Les Services de Cloud Computing	25

1.2.6.1	Infrastructure as a Service (IaaS)	25
1.2.6.2	Platform as a Service (PaaS)	26
1.2.6.3	Software as a Service (SaaS)	26
1.2.7	Avantages et Inconvénients des services	27
1.2.8	Acteurs du cloud computing	28
1.2.8.1	Cloud Provider	28
1.2.8.2	Cloud Consumer	28
1.2.8.3	Cloud Carrier ou Network Provider	28
1.2.8.4	Cloud Broker	29
1.2.8.5	Cloud Auditor	29
1.2.9	Exemples connus de Cloud Computing	29
1.3	Conclusion	30
2	L'ordonnement des workflows scientifiques dans le cloud computing	31
2.1	Introduction	32
2.2	les algorithmes d'ordonnement des tâches	32
2.2.1	Ordonnement des tâches indépendantes	33
2.2.1.1	Heuristiques d'ordonnement en ligne	33
2.2.1.2	Heuristiques d'ordonnement par lot (batch scheduling)	33
2.2.2	Ordonnement des tâches dépendantes	33
2.2.2.1	Les algorithmes d'ordonnement des listes	33
2.2.2.2	Les algorithmes d'ordonnement de clustering	33
2.2.2.3	Les algorithmes d'ordonnement de duplication des tâches	34
2.2.3	Critères de planification dans le Cloud Computing	35
2.2.3.1	Temps	35
2.2.3.2	Coût	35
2.2.3.3	Fiabilité	36
2.2.3.4	Sécurité	36
2.3	Conclusion	36
3	Approches Proposés	37
3.1	Introduction	38

3.2	Représentation de workflows scientifique	38
3.3	Description de l'adaptation Algorithme Génétique	39
3.4	Opérateurs de l'algorithme génétique	40
3.4.1	Population Initiale	40
3.4.2	Représentation d'un chromosome	41
3.4.3	Opérateurs de sélection et de croisement	41
3.4.4	Opérateur de mutation	43
3.5	Description de la fonction de fitness	43
3.5.1	Temps d'execution estime T_{execi}	44
3.5.2	Estimation du temps d'accès aux données $T_{Data_accessi}$	45
3.5.3	Temps de libération des ressources $T_{release}$	45
3.6	description du Coût(\$)	45
3.7	Description de la fonction Globale	46
3.8	Description de l'adaptation de l'algorithme PSO	46
3.8.1	particule	48
3.8.2	Essaim	48
3.9	Conclusion	49
4	Expérimentation et évaluation	50
4.1	Introduction	51
4.2	Langage et environnement de travail	51
4.2.1	Langage de programmation JAVA	51
4.2.2	Environnement de travail Netbeans	52
4.2.3	Simulateur CloudSim	52
4.2.3.1	Définition	52
4.2.3.2	Caractéristiques de CloudSim	53
4.2.3.3	Architecture CloudSim	53
4.2.3.4	Différente classe de CloudSim	54
4.3	Interface principale	56
4.3.1	Configuration de simulation	56
4.3.1.1	le Cloud	56
4.3.1.2	algorithme génétique	60
4.3.1.3	PSO	61
4.3.1.4	la représentation graphique	61

4.3.1.5	comparaison entre deux Aproches	61
4.3.2	Résultats expérimentaux	61
4.4	Conclusion	65
	Conclusion générale	66
	Bibliographie	67

Table des figures

1.1	Description du Cloud Computing	21
1.2	la virtualisation de Cloud Computing	23
1.3	Les modèles de déploiement dans le cloud computing	25
1.4	Les services de cloud computing	27
1.5	les acteurs de cloud computing	28
3.1	Exemple d'une instance simple de workflow scientifique	39
3.2	organigramme d'un algorithme génétique	40
3.3	Description du chromosome	41
3.4	Exemple du « Cut Crossover » pour l'ordonnancement de workflow.	42
3.5	Exemple du « Application de l'opérateur mutation de swap simple	43
3.6	Paramètres d'optimisation	44
3.7	Organigramme algorithme PSO	47
3.8	Déplacement d'une particule	49
4.1	Architecture de CloudSim	54
4.2	Data Center de Microsoft	55
4.3	interface principale	57
4.4	Configuration du data centers	58
4.5	Configuration des machines virtuels	58
4.6	Configuration des cloudlets	59
4.7	Configuration des données	60
4.8	algorithme génétique	60
4.9	interface principale avec les caractéristique décrit ci-dessus	62
4.10	présentation graphique de la simulation de l'algorithme génétique	62

4.11	présentation graphique de la simulation de PSO	63
4.12	la Représentation graphique	63
4.13	présentation graphique de la simulation de la comparaison (Response time) temps de reponse et le coût entre deux Approches	64
4.14	comparaison selon le coût entre deux Approches	64

Liste des tableaux

1.1	Les avantages et Les inconvénients des services	27
-----	---	----

Listes des abréviations

CDDL Common development and distribution licence

CIS Cloud Information Services

DAG Directed Acyclic graph

EC2 Elastic Compute Cloud2

EDI Environnement de développement intégré

FCFS First come First Served

GOOGLE Global Organization of Oriented Group Language of Earth

IAAS Infrastructure as a Service

IBM International Business Machines

NET Network

NIST National Institute of Standards and Technology

OS Operating System

PAAS Platform as a Service

PSO Particle Swarm Optimization

QOS Quality of Service

RR Round Robin

SAAS Software as a Service

SJF Shortest-Job First

SLA Service Level Agreements

SSH Secure Shell

VM Virtual Machines

Introduction Générale

Contexte

Le cloud computing présente une technologie prometteuse qui facilite l'exécution des applications scientifiques et commerciales. Il fournit des services flexibles et évolutifs, à la demande des utilisateurs, via un modèle de paiement à l'usage. Généralement, il peut fournir trois types de services : SaaS (Software as a Service) , PaaS (Platform as a Service) , IaaS (Infrastructure as a Service) . Ces services sont offerts avec différents niveaux de qualité de service afin de répondre aux besoins spécifiques de différents utilisateurs.

Bien que de nombreux services de cloud computing ont des fonctionnalités similaires (par exemple des services de calculs, services de stockages, services réseaux, etc.), ils diffèrent les uns des autres par leurs QoS (Quality of Service) non-fonctionnelles, telles que le temps, le coût, la disponibilité du service, la consommation d'énergie, etc.

Ces paramètres de QoS peuvent être définis et proposés par différents contrats de service, SLA (Service Level Agreements) . Un SLA spécifie les exigences négociées en termes de ressources, les QoS, les attentes minimales et les obligations qui existent entre les utilisateurs et les fournisseurs de cloud. Pour les fournisseurs des services du cloud, appliquer de tels contrats SLA est impératif. Le manque de tels accords peut entraîner l'éloignement des utilisateurs d'un fournisseur cloud et compromettra la croissance de ce dernier. [1]

Le cloud computing est l'une des technologies émergentes dont l'utilisation est devenue omniprésente, tant dans la vie privée, que dans la vie professionnelle. Le cloud Infrastructure as a Service (IaaS), est la catégorie du cloud qui propose une Infrastructure matérielle (i.e. CPU, stockage, réseau), et logicielle (i.e. technologies de

virtualisation), sous la forme d'un seul service. Les ressources offertes par ce service sont des ressources virtualisées, à savoir des machines virtuelles VM (Virtual Machines) pour offrir des services attractifs et à moindre coût constitue un grand défi pour les fournisseurs de cloud. [2]

Le Cloud Computing a révolutionné l'informatique, libérant les entreprises des grands investissements en capital informatique et leur permettant de se connecter à des ressources informatiques extrêmement puissantes sur le réseau. [3]

Le recours aux infrastructures de type Cloud se justifie par deux motivations essentielles :

- Nécessité de disposer de puissance de calcul élevée pour répondre aux besoins des applications qui nécessitent un calcul élevé de haute performance.
- L'utilisation d'ensemble de données très large.

Problématiques et motivations :

L'ordonnancement de workflows dans un cloud est un problème difficile. Ce problème est d'autant plus difficile lorsqu'il y a plusieurs facteurs à prendre en compte, à savoir : (1) les différentes contraintes et exigences de QoS des utilisateurs (par exemple le temps de réponse du service, le coût des services et autres); (2) l'hétérogénéité, la dynamique et l'élasticité des services de cloud disponibles; (3) les diverses possibilités de combiner ces services pour exécuter les tâches de workflows; (4) le transfert de gros volumes de données (les ressources peuvent être géographiquement distantes les unes des autres, ceci peut poser des problèmes aux workflows qui comportent une énorme quantité de données) et autres. Cependant, le problème d'ordonnancement de workflows est vu comme un problème d'optimisation combinatoire, où il est impossible de trouver la solution globale optimale en utilisant des algorithmes ou des règles simples. Il est bien connu comme un problème NP-complet, et dépend de la taille du problème à résoudre. [1]

L'ordonnancement est un aspect fondamental dans le cadre du Cloud Computing car il permet la parallélisation des applications et l'amélioration de leurs performances. Autrement dit, un problème d'ordonnancement consiste à organiser l'exécution d'un ensemble de tâches, de données et de ressources. Les algorithmes d'ordonnancement de tâches souffrent de plusieurs insuffisances dont les plus significatives se déclinent selon les points suivants :

- ils ne prennent pas en compte qu'aux cas qu'une tâche, pour les applications gourmandes en données, peuvent nécessiter plusieurs ensembles de données. Cependant, lorsque ces tâches sont exécutées dans différents centres de données du Cloud, le transfert de données deviendrait inévitable. [4][5][6]
- ils ne prennent en compte qu'un seul critère d'optimisation à savoir le temps d'exécution. Il serait plus approprié de prendre en compte le coût généré pour chaque application.

Contribution et Organisation du manuscrit :

Le but de ce mémoire est d'étudier les applications scientifiques dans le cloud computing en se basant sur les méta-heuristiques GA et PSO, nous avons étudié le temps d'exécutions de l'application et nous avons étudié aussi le coût (parce que Les opérateurs de cloud IaaS proposent à leurs clients des ressources virtualisées (CPU, stockage et réseau) sous forme de machines virtuelles (VM). L'explosion du marché du cloud les a contraints à optimiser très finement l'utilisation de leurs centres de données afin de proposer des services attractifs à moindre coût. En plus des investissements liés à l'achat des infrastructures et de leur coût d'utilisation). Le travail que nous avons mené dans le cadre de la problématique est résumé dans le présent document qui est structuré en quatre chapitres :

- **Chapitre 1** : présente un état de l'art sur Cloud computing, ses Services, ses types.
- **Chapitre 2** : un état de l'art sur l'ordonnancement de workflows dans le cloud computing.
- **Chapitre 3** : Est la présentation de nos approches proposées.
- **Chapitre 4** : l'implémentation de nos approches, les outils de développement, ainsi que les résultats obtenus.

Chapitre **1**

Etat de l'art sur le Cloud Computing

1.1 Introduction

Le Cloud Computing est un service généralement employé par l'internet que nous utilisons presque tous les jours sans nous rendre compte : de l'envoi des emails au stockage de nos données. Ces services se sont vite développés grâce à l'évolution rapide de l'internet et son installation quasi-totale dans le monde. Ce 1er chapitre est entièrement consacré pour avoir une vue générale sur le Cloud comment sa marche. [7]

1.2 Cloud Computing

1.2.1 Introduction

Depuis la création de l'Internet, elle se développe de façon permanente et rapide .Aujourd'hui le monde de technologie de l'information et de la communication a connu une nouvelle disposition c'est le Cloud Computing, il s'appuie sur le WEB 1.0. Cette technologie permet aux utilisateurs l'exploitation des logiciels en virtuel (enligne) ce qui lui permet de réduire les coûts d'exploitation de ces derniers. Une vaste gamme de services de Cloud Computing est offerte par plusieurs fournisseurs talque Google (Global Organization of Oriented Group Language of Earth), Amazon IBM(International Business Machines).

Bien que le Cloud Computing soit dans sa phase de début néanmoins plusieurs acteurs importants optent leurs propres politiques de créateur qui conduira l'utilisation du Cloud Computing des entreprises désirant investir .[8]

1.2.2 Définition

Le Cloud signifie « nuage » et Computing « informatique », le Cloud Computing est donc l'informatique en nuage pour une traduction littérale anglais français. Plusieurs définitions du Cloud Computing existent :

-Selon NIST(National Institute of Standards and Technology)[9]) : qui définit le Cloud Computing comme étant l'accès via un réseau de télécommunication, à la demande et en libre-service, à des ressources informatiques partagées configurables. par exemple réseaux, serveurs, stockage, applications et services qui peuvent être rapidement mises à disposition des utilisateurs, un des points clés des services Cloud

est qu'ils peuvent être approvisionnés rapidement mais décommissionnés.

-Selon R.Buyya[10] :Le Cloud est un système parallèle et distribué constitué d'un ensemble d'ordinateurs interconnectés et virtualisés, provisionnés de manière dynamique et présentés comme une ou plusieurs ressources informatiques réunies sur la base d'accords de niveau de service établis par négociation entre le fournisseur de services et les consommateurs.

-Selon K.Sheynkman[8] :Les Clouds se sont concentrés sur la fabrication de la couche matérielle en tant que capacité de calcul et de stockage à la demande. Il s'agit d'une première étape importante, mais pour que les entreprises exploitent la puissance du Cloud, une infrastructure d'applications complète doit être facilement configuré, déployée, mise à l'échelle de manière dynamique et gérée dans ces environnements matériels virtualisés.

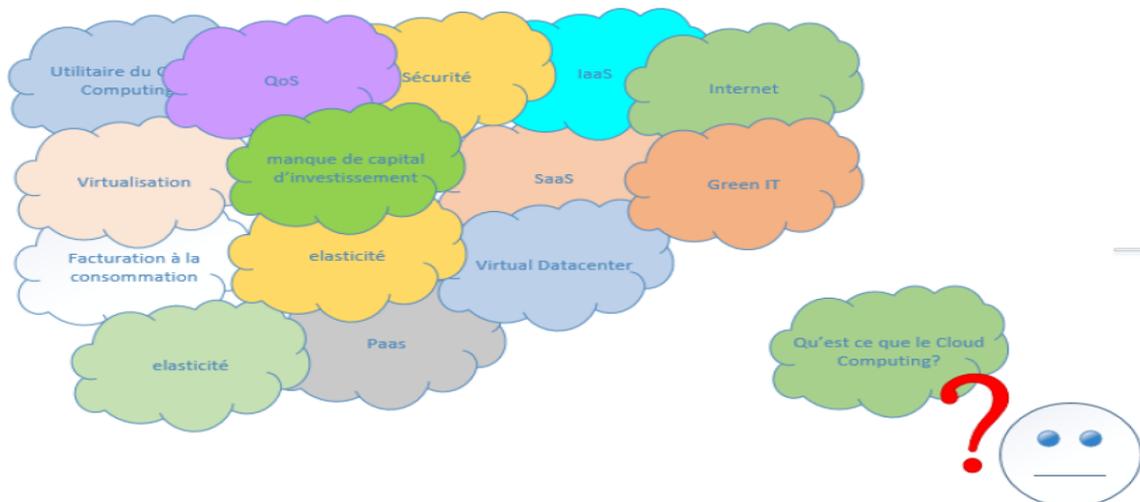


FIGURE 1.1 – Description du Cloud Computing
[11]

1.2.3 Caractéristiques essentielles

Le cloud computing est une nouvelle façon de délivrer les ressources informatiques, et non une nouvelle technologie. Il se caractérise par :[12]

Un accès en libre-service à la demande informatiques en fonction de ses besoins. Les ressources informatiques sont fournies d'une manière entièrement automatisée.[12]

Une mise en commun des ressources les ressources informatiques sont mises à la disposition des clients sur un modèle multi-locataires, avec une attribution dynamique des ressources physiques et virtuelles en fonction de la demande. Le client n'a

généralement ni contrôle, ni connaissance sur l'emplacement des ressources, mais est en mesure de le spécifier à un plus haut niveau d'abstraction (pays, état ou centre de données).[12]

Une élasticité rapide les capacités informatiques mises à disposition du client peuvent être ajustées (augmenter ou diminuer) rapidement (quelques minutes ou quelques secondes) en fonction des besoins et/ou de la charge et de façon automatique dans certains cas.[12]

Un service mesuré en permanence Les ressources consommées sont contrôlées et Communiquées au client et au fournisseur de service de façon transparente. Cela garantit un niveau de disponibilité adapté aux besoins spécifiques des clients.[12]

1.2.4 Les technologies de cloud computing

Le cloud computing utilise des technologies telles que la virtualisation, l'architecture orientée services et les services web. le cloud est souvent confondu avec plusieurs paradigmes informatiques, tels que le Grid computing, l'Utility computing et l'Autonomic computing, dont chacun partage certains aspects avec le cloud computing :

1.2.4.1 Grid Computing

une grille informatique (en anglais, grid) est une infrastructure virtuelle constituée d'un ensemble de ressources informatiques potentiellement partagées, distribuées, hétérogènes, délocalisées et autonomes.[15]

Une grille est en effet une infrastructure, c'est-à-dire des équipements techniques d'ordres matériel et logiciel. Cette infrastructure est qualifiée de virtuelle car les relations entre les entités qui la composent n'existent pas sur le plan matériel mais d'un point de vue logique.

1.2.4.2 Informatique utilitaire

L'informatique utilitaire est basée sur le modèle de paiement à l'utilisation .Il fournit des ressources de calcul à la demande en tant que service ave compteur. Tous les services informatiques gérés, Grid computing, cloud computing sont basés sur le concept de grid computing.[16]

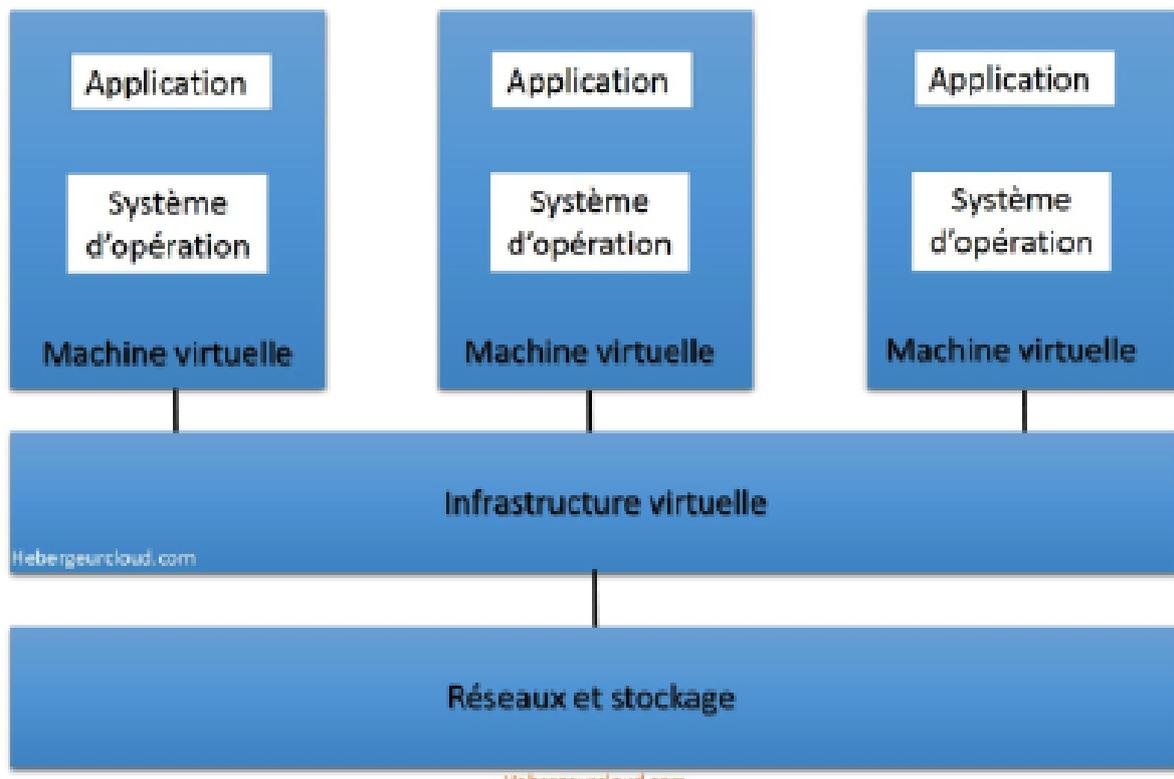


FIGURE 1.2 – la virtualisation de Cloud Computing
[14]

1.2.4.3 La virtualisation

La virtualisation est une technique qui permet de partager une seule instance physique d'une application ou d'une ressource entre plusieurs organisations ou locataires (clients). Il le fait en attribuant un nom logique à une ressource physique et fournir un pointeur sur cette ressource physique lorsqu'elle est demandée.

L'architecture multi comptes offre un isolement presque total entre les locataires Multiples et par conséquent, les organisations peuvent utiliser et personnaliser l'application, comme si, ils possèdent chacun sa propre instance d'exécution.[13]

1.2.5 Modèles de déploiement

Selon la définition du Cloud Computing donnée par le NIST, il existe quatre modèles de déploiement des services de Cloud, à savoir : Cloud privé, Cloud communautaire, Cloud public et Cloud hybride.[14]

1.2.5.1 Cloud privé

L'infrastructure cloud est exploitée uniquement pour une organisation. Elle peut être gérée par l'entreprise elle-même ou par un tiers et peut exister sur site ou hors site.[14]

1.2.5.2 Cloud communautaire

L'infrastructure cloud est partagée par plusieurs organisations, réunies au sein d'une communauté et partageant des préoccupations spécifiques communes (par exemple, la mission, les exigences de sécurité, des politiques et des considérations de conformité). Elle peut être gérée par les entreprises elles-mêmes ou par un tiers et peut exister sur site ou hors site.[14]

1.2.5.3 Cloud public

L'infrastructure cloud est mise à la disposition du grand public ou d'un grand groupe industriel et appartient à une organisation vendant des services cloud.[14]

1.2.5.4 Le Cloud hybride

L'infrastructure Cloud est composée l'infrastructure cloud est une composition de deux ou plusieurs clouds (privés, communautaires ou publics). Ceux-ci demeurent des entités uniques mais sont connectés par une technologie normalisée ou exclusive qui permet le partage des données et des applications des nuages).[14]

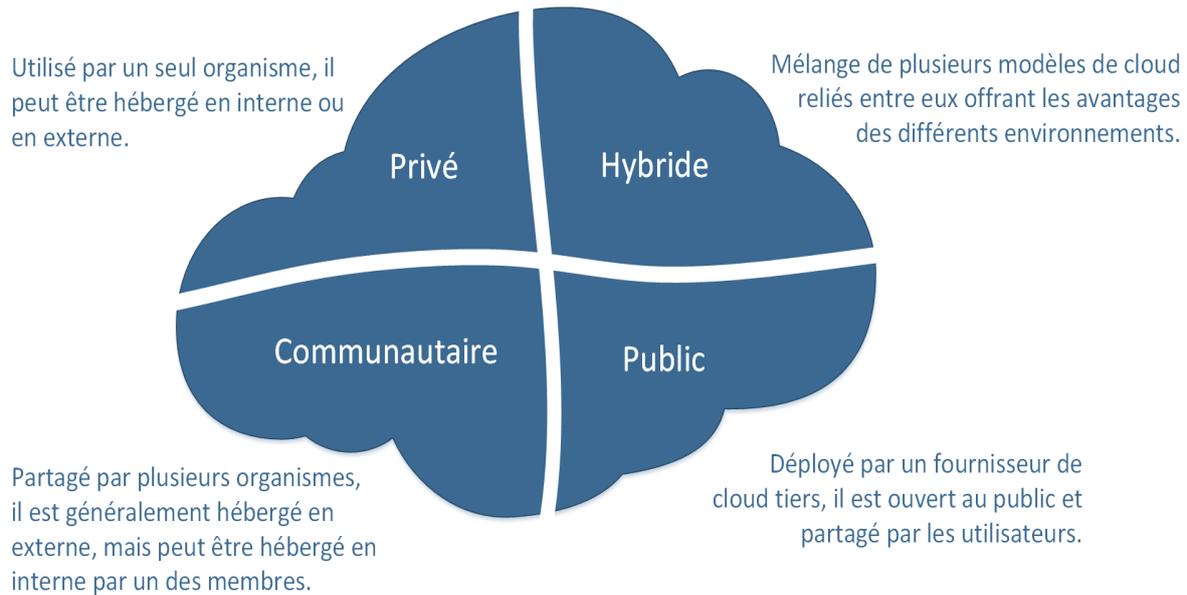


FIGURE 1.3 – Les modèles de déploiement dans le cloud computing
[14]

1.2.6 Les Services de Cloud Computing

Il existe aujourd'hui trois modèles de base qui peuvent être utilisés sur un cloud selon les ressources allouées aux clients. La Figure 1.4 montre ces trois modèles ainsi que le modèle traditionnel et les ressources fournies aux clients dans chaque modèle. On peut ainsi distinguer :[12]

1.2.6.1 Infrastructure as a Service (IaaS)

Le fournisseur met à disposition uniquement des ressources matérielles (machine, réseaux, Disque,datacenter) et le client gère le système et les applications[12].On retrouve dans cette catégorie : Amazon EC2(Elastic Compute Cloud2) comme exemple d'IaaS.

- **Avantage** :grande flexibilité, contrôle total des systèmes (administration à distance par SSH(Secure Shell) ou Remote Desktop...), qui permet d'installer tout type de logiciel métier.
- **Inconvénient** :besoin d'administrateurs système comme pour les solutions de serveurs classiques sur site.

1.2.6.2 Platform as a Service (PaaS)

Le client ne gère et ne contrôle ici que ses applications métier, le reste est délégué au fournisseur de services[12]. Google App Engine⁴ et Windows Azure⁵ sont des exemples de PaaS.

- **Avantage** : le déploiement est automatisé, pas de logiciel supplémentaire à acheter ou à installer.
- **Inconvénient** : limitation à une ou deux technologies (ex. : Python ou Java pour Google AppEngine, .NET(Network) pour Microsoft Azure. . .). Pas de contrôle des machines virtuelles sous-jacentes. Convient uniquement aux applications Web.

1.2.6.3 Software as a Service (SaaS)

C'est l'ultime niveau de transfert vers le fournisseur, le client ne gère plus que ses données métier, il utilise les applications fournies par le fournisseur. Il s'agit d'utiliser le logiciel à la demande[12]. Nous pouvons citer Google Apps⁶ comme exemples de SaaS.

- **Avantage** : plus d'installation, plus de mise à jour (elles sont continuées chez le fournisseur), plus de migration de données etc. Paiement à l'usage. Test de nouveaux logiciels avec facilité.
- **Inconvénient** : Pas de contrôle sur le stockage et la sécurisation des données associées au logiciel. Réactivité des applications Web pas toujours idéale.



FIGURE 1.4 – Les services de cloud computing [14]

1.2.7 Avantages et Inconvénients des services

Le tableau(1.1) illustre les services de Cloud Computing qui ont été décrits dans la section précédente tout en montrant les avantages et les inconvénients de chaque service[12].

	Avantages	Inconvénients
SaaS	<ul style="list-style-type: none"> — pas d’installation. — plus de licence. — migration. 	<ul style="list-style-type: none"> — logiciel limité. — sécurité. — Dépendance du prestataire.
Paas	<ul style="list-style-type: none"> — pas d’infrastructure nécessaire. — pas d’installation. — environnement hétérogène. 	<ul style="list-style-type: none"> — limitations des langages. — pas de personnalisation dans la configuration des machines virtuelles.
IaaS	<ul style="list-style-type: none"> — administration. — personnalisation. — flexibilité d’utilisation. 	<ul style="list-style-type: none"> — sécurité. — besoin d’un administrateur système.

TABLE 1.1 – Les avantages et Les inconvénients des services [12]

1.2.8 Acteurs du cloud computing

L'écosystème du Cloud Computing est composé principalement par cinq acteurs majeurs (Cloud Provider, Cloud Consumer, Cloud Carrier, Cloud Broker, Cloud Auditor) :[1]

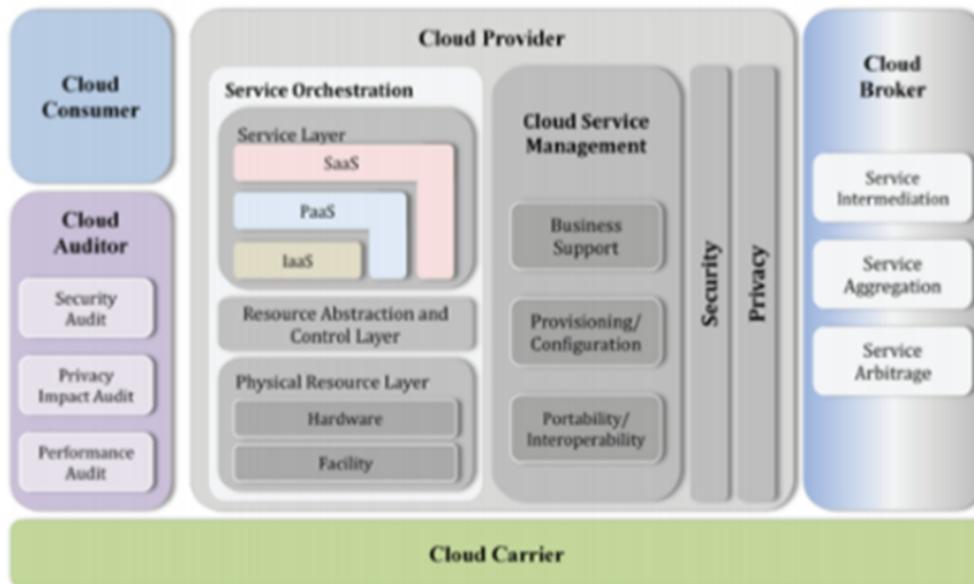


FIGURE 1.5 – les acteurs de cloud computing
[1]

1.2.8.1 Cloud Provider

Le Cloud Provider a comme activité l'allocation, l'orchestration et la gestion des ressources qu'il offre tout en assurant le bon niveau de sécurité.[1]

1.2.8.2 Cloud Consumer

L'utilisateur des ressources Cloud Computing. Cet utilisateur peut être un utilisateur final ou un développeur selon le type du service Cloud alloué. Cet utilisateur peut être une personne, un groupe de personnes, les petites et moyennes entreprises, les multinationales ou les gouvernements.[1]

1.2.8.3 Cloud Carrier ou Network Provider

Le fournisseur de réseau est l'intermédiaire qui assure principalement la connectivité entre les ressources Cloud Computing et la liaison entre les acteurs de l'écosys-

tème Cloud Computing (en particulier entre le Cloud Provider et le Cloud Consumer). Cet utilisateur peut jouer un simple rôle d'acheminements des paquets, comme il peut jouer un rôle plus important en offrant des fonctionnalités avancées dans le réseau. Ces fonctionnalités sont basées sur des SLAs établies avec les autres acteurs de l'écosystème.[1]

1.2.8.4 Cloud Broker

Le courtier Cloud est un intermédiaire qui négocie la relation entre les Cloud Providers et les Cloud Consumers. Il peut offrir de nouveaux services qui simplifient les tâches de gestion du Cloud Consumer. Ce dernier peut demander les ressources Cloud Computing auprès du Cloud Broker au lieu du Cloud Provider directement.[1]

1.2.8.5 Cloud Auditor

L'auditeur Cloud s'occupe de la vérification et l'audition des services Cloud Computing. Il évalue les services offerts par les Cloud Providers, Cloud Carriers et Cloud Brokers du point de vue performances et sécuritaires. Le but principal est de vérifier que les fournisseurs respectent bien les SLAs qu'ils proposent.[1]

1.2.9 Exemples connus de Cloud Computing

Voici quelques exemples de cloud computing : [14]

Google Drive

Google Drive est un pur service Cloud Computing. Il propose un stockage en ligne, et fonctionne avec les applications Cloud Google Docs, Google Sheets et Google Slides. Ce service est accessible depuis un ordinateur, depuis une tablette, ou même depuis un smartphone, au même titre que les applications mobiles Docs et Sheets. La plupart des services Google peuvent d'ailleurs être classés dans la catégorie du Cloud Computing. C'est le cas de Gmail, Google Calendar, et Google Maps par exemple.

Apple iCloud

Principalement utilisé pour le stockage en ligne, le back-up, Apple iCloud est également utile pour la synchronisation des mails, des contacts, ou encore du calendrier. Toutes les données sont disponibles sur iOS(Operating System), ou sur les appareils

Windows depuis le panneau de contrôle iCloud. Apple propose également des versions Cloud de son traitement de texte Pages, de sa feuille de calcul Spreadsheet, et de son logiciel de présentation Keynote pour tous les utilisateurs d'iCloud. La plateforme permet en outre aux possesseurs d'iPhone d'utiliser la fonctionnalité Localiser mon iPhone.

Amazon Cloud Drive

Amazon propose essentiellement le stockage de musique au format MP3 et de photos. Les abonnés Amazon Prime bénéficient d'une capacité de stockage illimitée. Amazon Cloud Drive sert également à stocker le contenu acheté pour Kindle. En résumé, cette plateforme sert à stocker tout le contenu numérique acheté auprès d'Amazon. Des services hybrides comme Box, Dropbox ou SugarSynch se positionnent dans la catégorie Cloud en proposant une version synchronisée sur internet des fichiers stockés. La synchronisation est une pierre angulaire du Cloud Computing, même si l'accès aux fichiers se fait de façon locale. De même, si plusieurs personnes avec des appareils séparés travaillent sur les mêmes données synchronisées, on peut considérer qu'il s'agit de Cloud Computing.

1.3 Conclusion

Dans cette première chapitre, nous avons parlé sur le Cloud Computing, en présentant ses caractéristiques, ses modèles de service (IaaS,PaaS, SaaS),et ses modèles de déploiement, ainsi que ses avantages et inconvénients. le chapitre suivant nous allons présenter l'ordonnancement des workflows scientifiques dans le cloud computing.

Chapitre 2

L'ordonnancement des workflows scientifiques dans le cloud computing

2.1 Introduction

En raison de l'importance des applications de workflows, plusieurs projets de recherche ont été menés pour concevoir des systèmes de gestion de workflows et des algorithmes d'ordonnancement appropriés. Les systèmes de gestion de workflows peuvent être considérés comme un type de service facilitant l'automatisation des applications de business et science sur la grille et le cloud. Ils sont utilisés pour gérer ces applications de façon transparente en masquant l'orchestration et les détails d'intégration spécifiques lors de l'exécution des tâches sur les ressources distribuées de grille ou de cloud.[1]

En vue d'ordonner efficacement les tâches de ces applications sur les environnements de cloud computing, les systèmes de gestion de workflows requièrent des stratégies d'ordonnancement plus élaborées pour répondre aux exigences de QoS (makespan(temps d'exécution), coût et autres)spécifiées dans le SLA, alors l'étude des stratégies d'ordonnancement de workflows devient un enjeu important dans le cloud computing.[1]

2.2 les algorithmes d'ordonnancement des tâches

Avec l'apparition de l'informatique en nuage (ou Cloud computing) [17], plusieurs algorithmes d'ordonnancement ont été adaptés pour ce type de plate-forme. Les auteurs dans[18],[19],[20] étudient le problème d'ordonnancement des tâches dans les Clouds et proposent différentes techniques pratiques mises en œuvre pour résoudre ce problème.

La plupart des algorithmes d'ordonnancement dans les Clouds vise à remplir un ou plusieurs objectifs. Certains objectifs touchent à la rapidité de traitement des tâches et dans les délais[18],[21]. D'autres sont liés à la répartition équitable des ressources entre les tâches [24]. D'autres encore sont relatifs à l'utilisation optimale des ressources, par exemple le temps de réponse et le coût entre les ressources[21], le travail de Bessai.K dans[22] propose une classification des algorithmes existants en trois catégories selon les types de tâches constituant l'application : tâches indépendantes, tâches dépendantes DAG (Directed Acyclic graph)et les DAGs s'exécutant en parallèle.

2.2.1 Ordonnancement des tâches indépendantes

2.2.1.1 Heuristiques d'ordonnancement en ligne

Les tâches qui arrivent dans le système sont ordonnancées et affectées à des ressources de leurs arrivées.[16]

2.2.1.2 Heuristiques d'ordonnancement par lot (batch scheduling)

Les tâches ne sont pas exécutées à leurs arrivées mais regroupées dans des ensembles distincts et les dates de leurs ordonnancements sont déterminées ultérieurement. Les tâches parallèles sont aussi étudiées par des nombreuses heuristiques dynamiques. L'objectif de ces heuristiques est essentiellement l'augmentation du taux d'utilisation des ressources et la minimisation du temps d'attente des tâches .[18]

2.2.2 Ordonnancement des tâches dépendantes

Les applications composées de tâches dépendantes visent à optimiser leurs temps d'exécution en se focalisant sur des applications définies par des tâches liées par des contraintes de précédence (DAGs). L'ordonnancement de graphes de tâches composées des taches séquentielles a été largement étudié et plusieurs algorithmes heuristiques ont été proposés. Ces algorithmes peuvent être classés en trois catégories[18] :

2.2.2.1 Les algorithmes d'ordonnancement des listes

Les algorithmes de cette classe commencent par le calcul de la priorité des tâches constituant l'application, ensuite ces algorithmes consistent à choisir une ressource pour tâches dans l'ordre de priorité tout en minimisant une fonction de récompense préalablement définie (par exemple, la date de fin d'une tâche). Ces d'heuristiques et produisent des résultats similaires.[16]

2.2.2.2 Les algorithmes d'ordonnancement de clustering

La première étape de ces algorithmes est de commencer de regrouper les tâches dans des groupes (cluster). la deuxième étape consiste à affecter ces tâches aux ressources disponibles la contrainte à respecter est que toutes les tâches appartenant à un même groupe soient exécutées par une même ressource.

2.2.2.3 Les algorithmes d'ordonnancement de duplication des tâches

Les algorithmes de duplication sont des algorithmes NP-complet, capable de réduire les fais de communication par l'exécution d'une même tâche sur plusieurs ressources. Il existe plusieurs algorithmes basés sur d'ordonnancement de duplication de tâches basés sur les heuristiques. Le travail de classe les algorithmes d'ordonnancement de duplication de tâches en deux catégories : l'ordonnancement avec duplication partielle et l'ordonnancement avec double duplication.[29]

- Les principaux exemples d'algorithmes de planification sont l'algorithme de planification du premier arrivé, premier service, l'algorithme de planification prioritaire et algorithme génétique ,comme indiqué dans le suivant :

- **Algorithme d'ordonnancement FCFS(First come First Served)** Premier Arrivé Premier Servi Les tâches sont ordonnancées dans l'ordre où elles sont reçues. Le processus qui sollicite le CPU le premier sera servi le premier. On utilise une structure de file.[24]
- **Algorithme d'ordonnancement PSO(Particle Swarm Optimization)** : L'algorithme d'Optimisation par (PSO), L'optimisation de l'essaim de particules (PSO) est une technique d'optimisation évolutive et stochastique inspirée par la nature pour résoudre des problèmes d'optimisation informatiquement difficiles.[24]
- **Algorithme d'ordonnancement Max -min** : C'est l'inverse de SJF en sélectionnant les tâches les plus importantes à exécuter en premier.[16]
- **Algorithme d'ordonnancement Min-min** : L'algorithme Min-Min est un algorithme simple et efficace qui produit un meilleur calendrier qui minimise le temps total d'achèvement des tâches que les autres algorithmes de la littérature.[16]
- **Algorithme d'ordonnancement RR(Round Robin)** : Cet algorithme suit une stratégie simple qui consiste à distribuer de manière équitable les taches sur les machines virtuelles disponibles c-a-d que le nombre de taches sur les machines virtuelle est le même .cet algorithme est implémenté dans le simulateur ClouSim.[25]

- **Algorithme d'ordonnement SJF(Shortest-Job First)** : Le Plus Court Job d'abord, Cet algorithme nécessite la connaissance du temps d'exécution estimé de chaque processus. Le CPU est attribué au processus dont le temps d'exécution estimé est minimal .[24]

2.2.3 Critères de planification dans le Cloud Computing

Les critères de planification influencent la conception des techniques de planification. Contrairement à la planification des applications en grilles de calcul où la réduction de la durée d'exécution est dominante [16], la plupart des techniques de planification dans le Cloud sont multi objectifs ;le temps et le coût étant pris en compte conjointement lors de la planification .Néanmoins, d'autres objectifs sont également pris en compte.

2.2.3.1 Temps

Makespan ou le temps total d'exécution d'une application est un objectif dominant dans la plupart des techniques de planification depuis l'ère du calcul en grille. L'objectif temporel peut être spécifié en tant que contrainte dure, telle que la date limite(ou Deadline), et en tant que contrainte souple à minimiser au mieux.[16]

2.2.3.2 Coût

Économiquement motivés, les fournisseurs de cloud commercial s'efforcent d'offrir de meilleures garanties de services par rapport aux fournisseurs de grille. Les fournisseurs de cloud profitent également des économies d'échelle, en fournissant des ressources de calcul, de stockage et de bande passante, à un coût très faible grâce, à la virtualisation. Ainsi l'utilisation des services de cloud public pourrait être économique et une alternative moins coûteuse, par rapport à l'utilisation de ressources dédiées, qui sont plus chères. Un des avantages de l'utilisation des ressources virtuelles pour l'exécution de workflow, plutôt que d'un accès direct à la machine physique, est le besoin réduit pour sécuriser les ressources physiques des codes malveillants. Cependant, l'effet à long terme de l'utilisation de ressources virtuelles dans les clouds qui partagent efficacement une "tranche" de la machine physique, plutôt que d'utiliser des ressources dédiées pour les workflows de calculs intensifs, est une question de recherche intéressante.[1]

2.2.3.3 Fiabilité

Autre les critères de temps et de coût les plus courants, la fiabilité d'exécution d'une application est également abordée. Cet objectif garantit que les ressources sélectionnées dans une planification vont probablement terminer l'exécution des tâches qui leur sont programmées.[28]

2.2.3.4 Sécurité

La sécurité des données, la confidentialité et la gouvernance sont devenues un problème important lorsqu'une entreprise décide d'adopter une solution d'informatique en Cloud. Bien que des mécanismes de sécurité puissent être mise en œuvre sur des instances et des réseaux des machines virtuelles, la gouvernance des données peut spécifier des réglementations ou peut être imposée par des lois pour empêcher les données de quitter des infrastructures sur site.pour répondre à ces préoccupations, il convient de définir les exigences de sécurité et de confidentialité des données devant être traitées par une application répartie.cette exigence impose généralement une contrainte forte à la sélection des ressources dans le Cloud.[27]

2.3 Conclusion

Dans ce chapitre, nous avons présenté d'abord l'ordonnancement des taches puis on a vu les algorithmes d'ordonnancement dans l'environnement cloud computing ,le Critères de planification dans le Cloud Computing. le chapitre suivant présente nos contributions pour l'optimisation de workflow dans l'environnement cloud computing.

Chapitre 3

Approches Proposés

3.1 Introduction

Les méta-heuristiques travaillent sur un ensemble de points de l'espace de recherche, en commençant avec une population de solutions initiales puis elles s'efforcent de l'améliorer au fur et à mesure des itérations. L'intérêt de ces méthodes est d'explorer un très vaste espace de recherche et d'utiliser la population comme facteur de diversité. Parmi ces méta heuristiques, on distingue les algorithmes évolutionnaires, basés sur la théorie de l'évolution de Darwin , et l'intelligence en essaim basée sur le comportement des espèces vivant en colonies, comme les fourmis ou les abeilles . . .etc.

Dans ce chapitre, nous considérons principalement les algorithmes évolutionnaires, plus précisément les algorithmes génétiques et l'optimisation par essaim particulière (PSO) pour résoudre le problème d'ordonnancement de workflows dans le cloud. un workflow est représenté sous forme d'un graphe où les sommet sont les taches et les données. Les arcs de ce même graphe représentent l'allocation de donnée la tache ou la nécessité de cette donnée à la tache reliée. L'objectif est de faire une comparaison entre les deux méta-heuristiques (GA, PSO).

3.2 Représentation de workflows scientifique

Formellement, une spécification de workflow comme un multi graphe orienté étiqueté, dont les sommets représentent les tâches du workflow et les arcs représentent le flot des données entre les tâches. Considérons un multi-graphe acyclique $G = (V,E)$, où :

$V = \{v_1, v_2, \dots, v_n\}$ est l'ensemble de sommets qui représentant les taches qui le compose et qui sont exécutées de façon indépendantes. E un ensemble d'arcs représentant les contraintes de données entre les taches avec $E \subseteq V \times V$ La Figure 3.1 représente un exemple d'une couche de workflow scientifique constituée de cinq tâches et cinq données notées respectivement t_1, t_2, \dots, t_5 et d_1, d_2, \dots, d_5 .

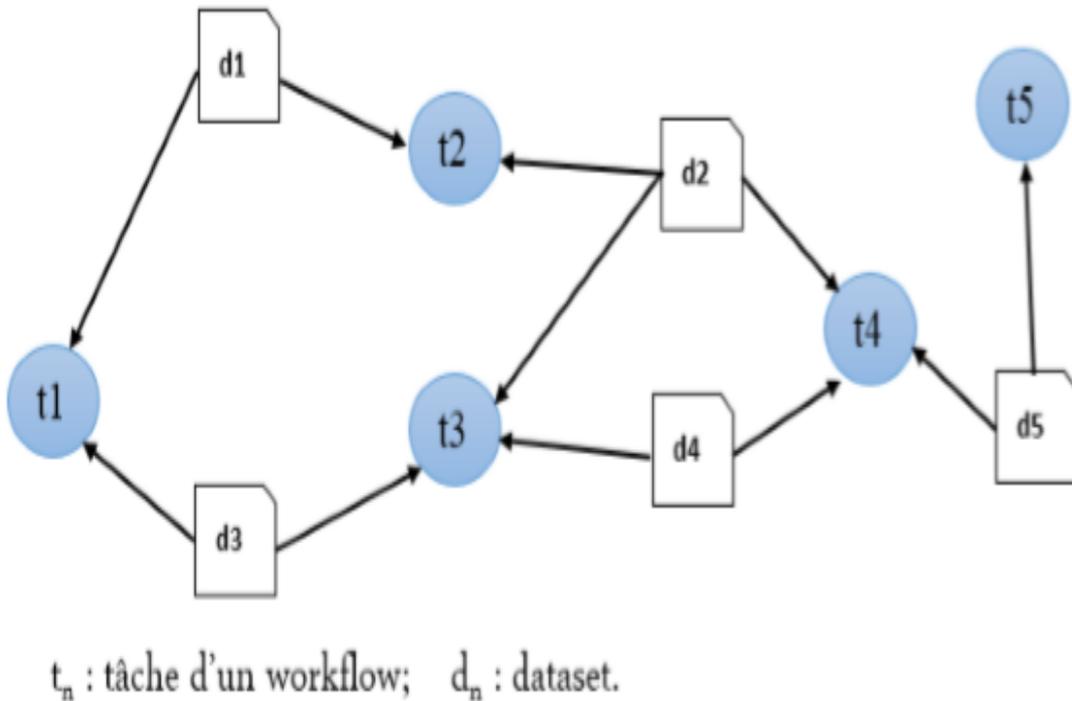


FIGURE 3.1 – Exemple d'une instance simple de workflow scientifique [30]

3.3 Description de l'adaptation Algorithme Génétique

Le mécanisme des algorithmes génétiques consiste à faire évoluer une population de solutions de manière itérative vers l'optimum du problème.

L'algorithme va procéder à l'évaluation des individus de la population, puis à la création de nouveaux individus. Pour ce faire, un certain nombre d'individus de la population sont sélectionnés selon une stratégie de sélection donnée, on les appelle les "parents". l'opérateur de croisement va permettre, à partir de ces parents, d'obtenir des individus dont l'information génétique consistera en un mélange de l'information présente dans les deux parents : les "enfants". Un opérateur de mutation sera éventuellement appliqué à ces enfants pour les diversifier.

Finalement, aux termes de l'application des opérateurs de sélection, croisement et mutation, une stratégie de remplacement décide ensuite des solutions, parmi la population courante et les enfants, qui constitueront la nouvelle population. Cet enchaînement d'étapes représente une génération. L'algorithme s'arrête lorsqu'il atteint un critère d'arrêt donné. Différents opérateurs de sélection, croisement, mutation, remplacement, peuvent être utilisés sont illustré dans Le diagramme dessus qui montre

les différentes étapes de l’algorithme génétique :

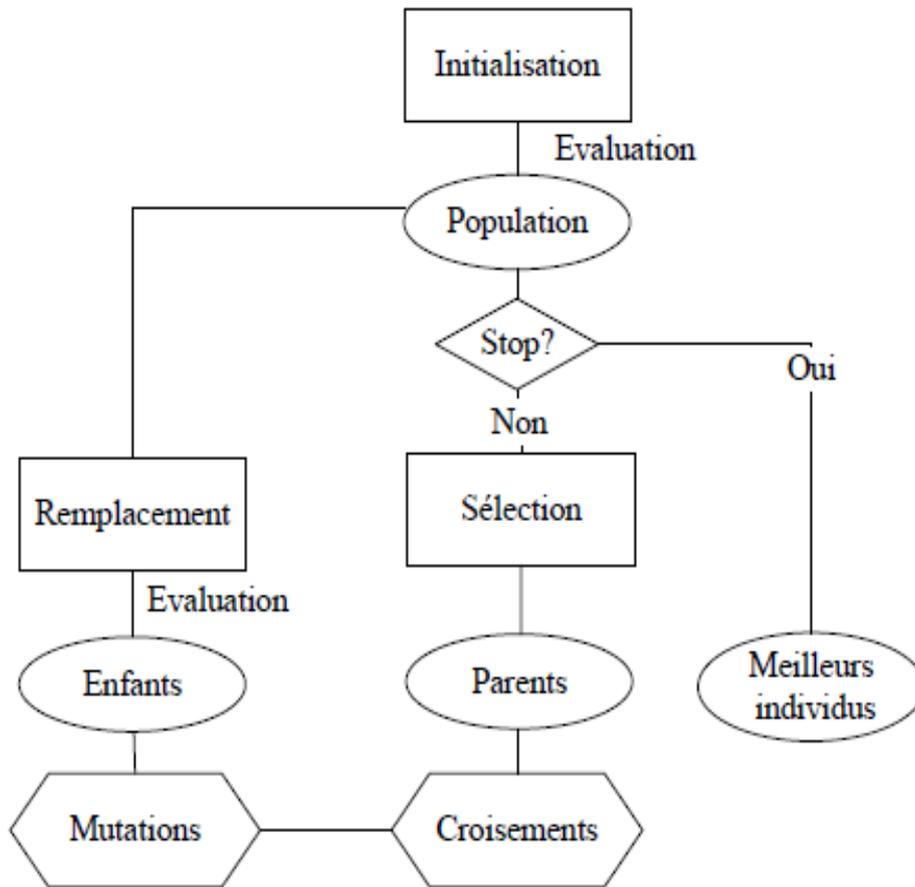


FIGURE 3.2 – organigramme d’un algorithme génétique [1]

3.4 Opérateurs de l’algorithme génétique

3.4.1 Population Initiale

Plusieurs mécanismes de génération de la population initiale sont utilisés dans la littérature. Le choix de l’initialisation se fera en fonction des connaissances que l’utilisateur a sur le problème. S’il n’a pas d’informations particulières, alors une initialisation aléatoire, la plus uniforme possible afin de favoriser une exploration de l’espace de recherche maximum, sera la plus adaptée. Par ailleurs, cette étape présente un problème principal qui est celui de choix de la taille de la population. En effet une taille de population trop grande augmente le temps de calcul et nécessite un espace mémoire considérable, alors qu’une taille de population trop petite conduit à l’obtention d’un optimum local .[33]

-La population initiale représente un ensemble de chromosome qu'on va le décrire dans la section suivante .[16]

3.4.2 Représentation d'un chromosome

Le chromosome représente un ordonnancement complet du workflow où chaque gène représente soit une tâche et la machine virtuelle requise pour l'exécuter ou l'emplacement d'une donnée sur cette dernière(VM).Chaque chromosome est une chaîne de gènes codant une solution spécifique .La figure 3.4 décrit la modélisation d'un chromosome pour une application scientifique composée de cinq tâches et six données qui vont être allouées sur trois machines virtuelles.

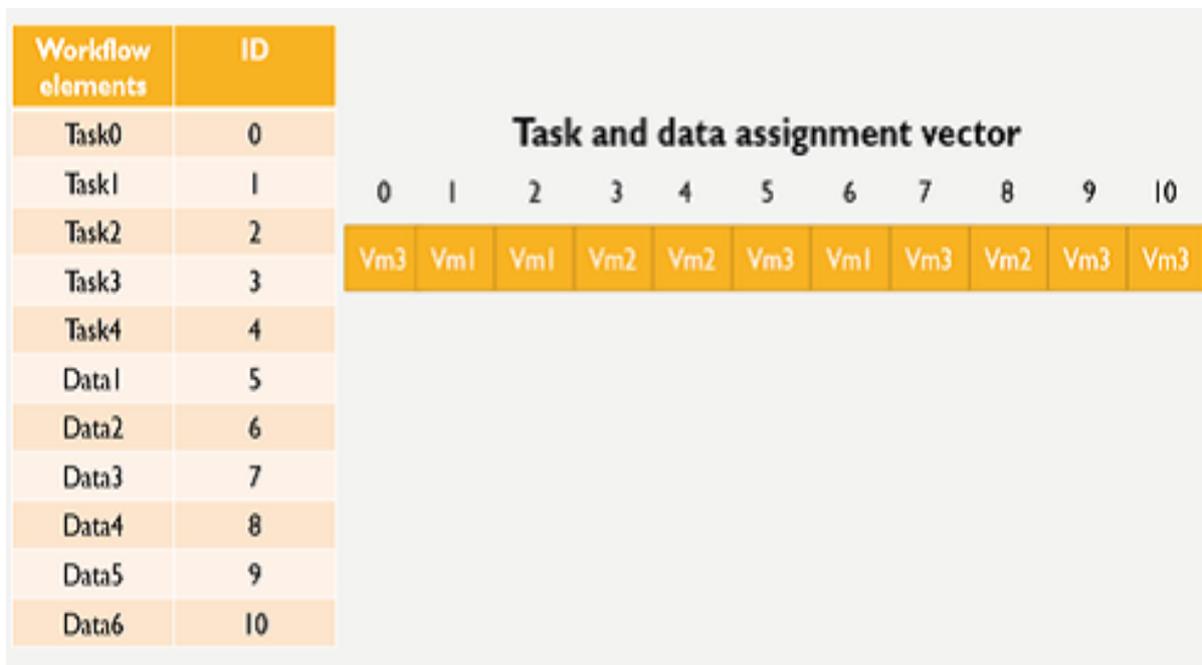


FIGURE 3.3 – Description du chromosome [16]

3.4.3 Opérateurs de sélection et de croisement

L'opérateur de croisement est utilisé pour générer une ou deux nouvelles solutions à partir de deux solutions parents.

Toutefois, avant d'effectuer le croisement, nous utilisons un processus de sélection afin de choisir les solutions parents à faire reproduire plusieurs algorithmes génétiques pour l'ordonnancement de workflow utilisent la méthode de sélection par roulette « **roulette wheel** », où chaque solution est attribuée plus ou moins de

chances pour participer à la génération de la population suivante, selon la valeur de sa fitness. L'objectif d'un opérateur de croisement est de combiner les caractéristiques de deux bonnes solutions afin de générer des solutions encore meilleures. Il existe plusieurs types de croisements tels que : le croisement aléatoire (Random Crossover), le croisement en un point (One-point Crossover), le croisement à deux points (Two-point Crossover), et autres. Dans cette thèse, nous avons implémenté seulement croisements en un point (Onepoint Crossover). Cet opérateur consiste à choisir aléatoirement un point de coupure pour partager chaque parent en deux parties. Le premier enfant est construit en utilisant la première partie du premier parent et la deuxième partie du deuxième parent. A l'inverse, le deuxième enfant est une combinaison de la seconde partie du premier parent et de la première partie du second parent.

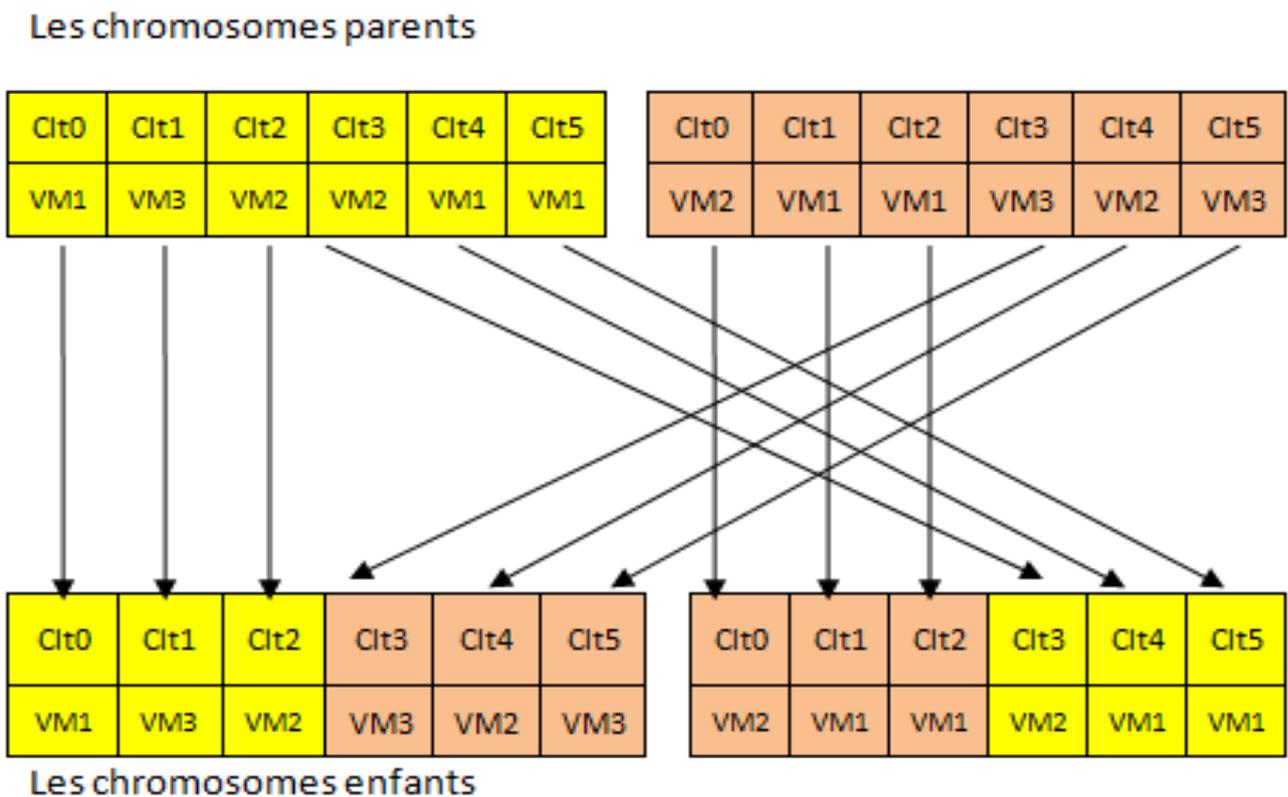


FIGURE 3.4 – Exemple du « Cut Crossover » pour l'ordonnancement de workflow.

3.4.4 Opérateur de mutation

L'objectif de l'opérateur de mutation est d'éviter une convergence vers une solution optimale locale, en réintroduisant des caractéristiques aléatoires qui n'appartiennent à aucune des solutions parents. Il existe plusieurs types de mutation tels que : Déplacer (Move), glisser (Swap), Déplacer et glisser (Move and Swap) et Ré-équilibrer (Rebalancing). Dans notre travail, nous avons implémenté seulement Swap simple. L'opérateur de swap produit une progéniture à partir d'un seul chromosome, il sélectionne d'abord une paire de gènes, puis il échange leurs valeurs. Une paire de valeurs de gènes est échangée à chaque opération de swap.

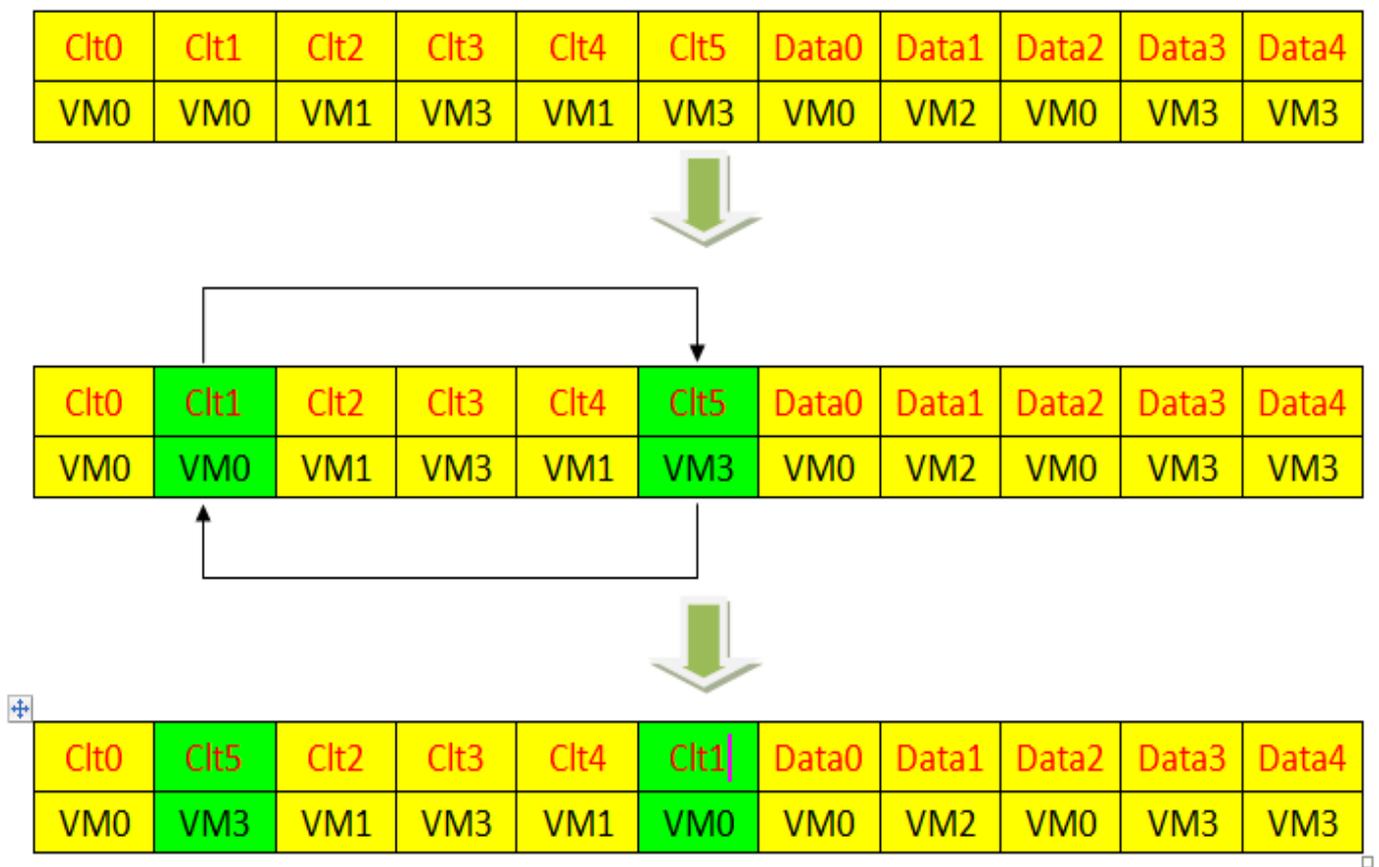


FIGURE 3.5 – Exemple du « Application de l'opérateur mutation de swap simple

3.5 Description de la fonction de fitness

L'objectif de notre approche est d'estimer le temps de réponse de chaque tâche avec le Coût de la machine virtuelle. Cette section expose des détails sur des paramètres d'optimisations des tâches, comme indiqué à la Figure 3.6

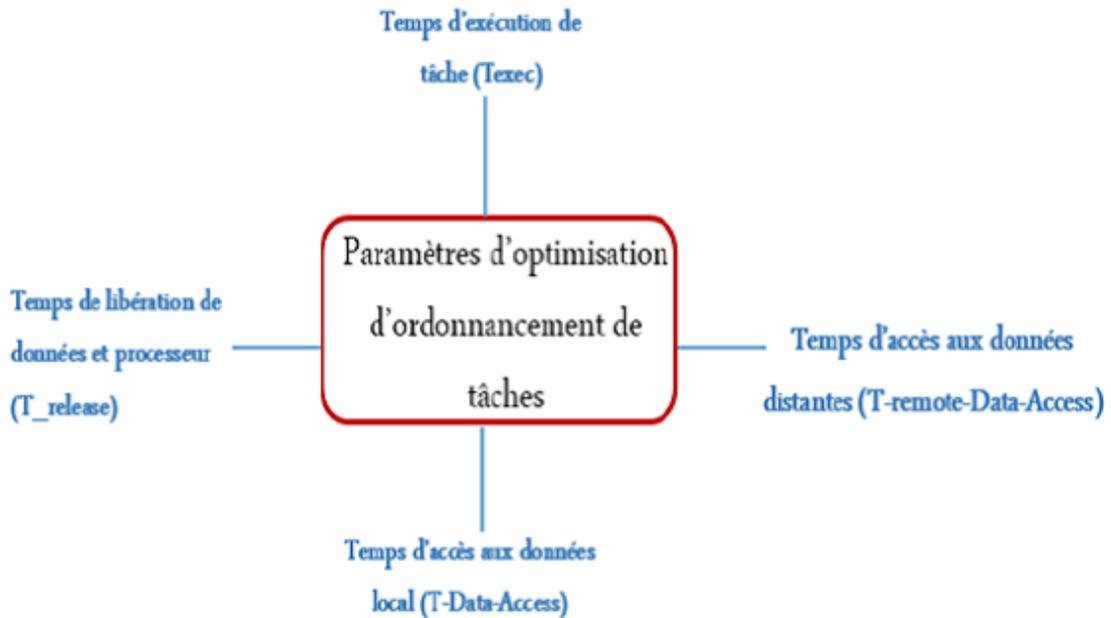


FIGURE 3.6 – Paramètres d’optimisation [30]

3.5.1 Temps d’execution estime T_execi

Le temps d’exécution estimé est mesuré en se basant sur la capacité de traitement de processeur de la machine virtuelle cible et la taille d’une tâche .Il représente le temps de traitement d’une tâche sur la ressource VMj.[31]

$$T_{exec_i} = \frac{T_{length}}{(capacityVM_j * PE_{Number}VM_j)}$$

Où :Tilenght : représente la taille de la tâche en terme du nombre d’instruction ;
 Capacity VMj : est la vitesse d’un processeur en MIPS (Million Instruction Per Second) ;
 PENumber : représente le nombre de processeur qui se trouve dans la machine virtuelle VMj.

Nous définissons le temps de libération des ressources comme étant le temps d’attente pour libérer les ressources CPU et l’ensemble de données nécessaire pour l’exécution de la tâche comme montre la formule suivante :

3.5.2 Estimation du temps d'accès aux données $T_{Data_access_i}$

Estimation du temps d'accès aux données représente le temps de traitement des données locales et distantes en se basant sur les deux formules suivantes : [30]

$$T_{DataAccess_i} = \sum_{k=1}^n \frac{LocalDatasetSize_k}{DiskTransfertCapacityVM_j} + T_{RemoteDataAccess_j}$$

$$T_{remoteDataAccess_i} = \left(\sum_{p=1}^L\right) \frac{RemoteDatasetSize_p}{BP_{vm_j}} + \frac{RemoteDatasetSize_p}{DiskTransfertCapacityVM_j}$$

Le principe de cette estimation est de mesurer le temps de traitement des données stockées localement, et le temps de déplacement de données manquantes pour cette tâche ainsi que leur traitement sur la ressource VM_j .

3.5.3 Temps de libération des ressources $T_{release}$

Nous définissons le temps de libération des ressources comme étant d'attente pour libérer les ressources CPU et l'ensemble de données nécessaire pour l'exécution de la tâche comme montre la formule suivante : [30]

$$T_{release} = T_{release_processorVM_j} + T_{release_dataset}$$

3.6 description du Coût(\$)

le coût d'utilisation de toutes les ressources nécessaire pour exécuter la tâche tels que :

le coût d'utilisation du CPU est $=C_{CPU}$

le coût d'utilisation du réseau(CBP) pour le déplacement de données est $=C_{BP}$

le coût de stockage est $=C_S$

Coût(\$) $=C_{CPU} + C_{BP} + C_S$

3.7 Description de la fonction Globale

$$\text{Min}_{(FCT)} = \text{Min}(\text{Temps}_{\text{Reponse}}, \text{Coût})$$

3.8 Description de l'adaptation de l'algorithme PSO

L'Optimisation par Essaim Particulaire (OEP), ou Particle Swarm Optimization (PSO) en anglais, est une méthode d'optimisation stochastique basée sur une population de solutions. Elle a été développée par Kennedy et Eberhart en 1995. Elle est inspirée du comportement social des colonies d'insectes, des nuées d'oiseaux, des bancs de poissons et bien d'autres sociétés animales évoluant en essaim. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim. Des règles simples, telles que « aller à la même vitesse que les autres », « se déplacer dans la même direction », ou encore « rester proche de ses voisins », suffisent pour maintenir la cohésion de l'essaim, permettant ainsi de mettre en oeuvre des comportements collectifs complexes et adaptés. L'intelligence globale de l'essaim est la conséquence directe des interactions locales entre les différentes particules de l'essaim. Comme l'algorithme génétique, l'algorithme d'optimisation par essaim particulaire fonctionne à base de population d'individus. Chaque individu est appelé "particule" et l'ensemble est appelé essaim ou "swarm". Mais elle se distingue par le fait qu'elle n'utilise pas les opérateurs d'évolution (sélection, mutation, croisement).

- le diagramme suivant illustre le mécanisme de la méta heuristique PSO pour l'optimisation d'ordonnancement de workflow scientifique dans le cloud computing.

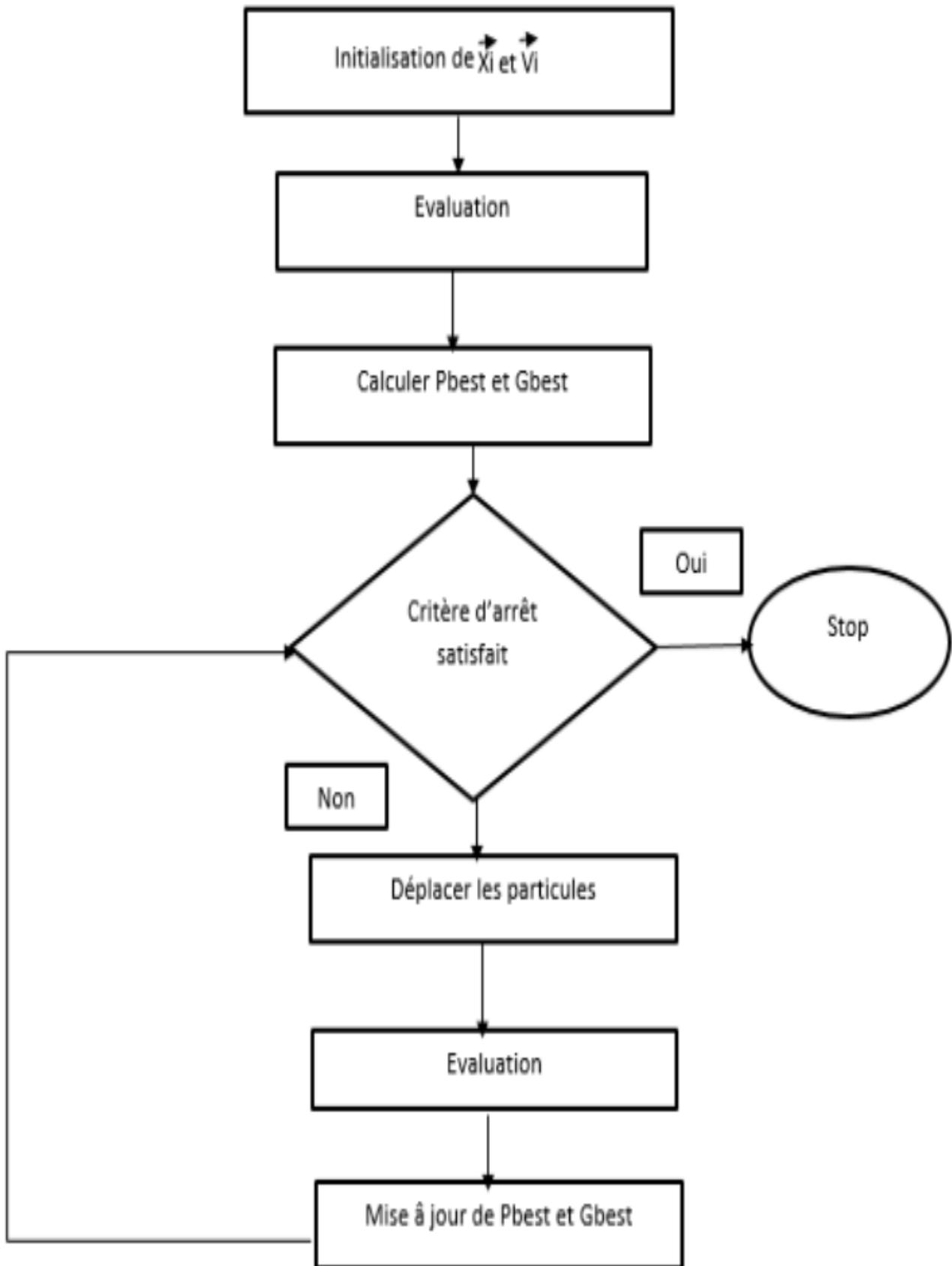


FIGURE 3.7 – Organigramme algorithme PSO

3.8.1 particule

Les particules représentent des solutions potentielles au problème d'optimisation. Elles survolent l'espace de recherche, afin de trouver l'optimum global. Chaque particule possède une position (le vecteur solution) CP et une vitesse(V). de plus, elle possède une mémoire lui permettant de se souvenir de sa meilleure performance (en position et en valeur) et de la meilleure performance atteinte par les particules « voisines » (informatrices).

Dans notre thèse, le particule a pris la même forme que le chromosome, car il se présente sous la forme d'un vecteur qui contient Machine Virtuel, Cloudlet et Donnée, et nous avons calculé temps de réponse et l'optimum global(Globale best) pour trouver la mieur particule

3.8.2 Essaim

L'essaim de particules correspond à une population d'agents simples, appelés particules. Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, « survole » l'espace de recherche, à la recherche de l'optimum global(Globale best). Le déplacement d'une particule est influencé par les trois composantes suivantes :

1. Une composante d'inertie : la particule tend à suivre sa direction courante de déplacement ;
2. Une composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;
3. Une composante sociale : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins. La stratégie de déplacement d'une particule est illustrée dans la figure suivante :

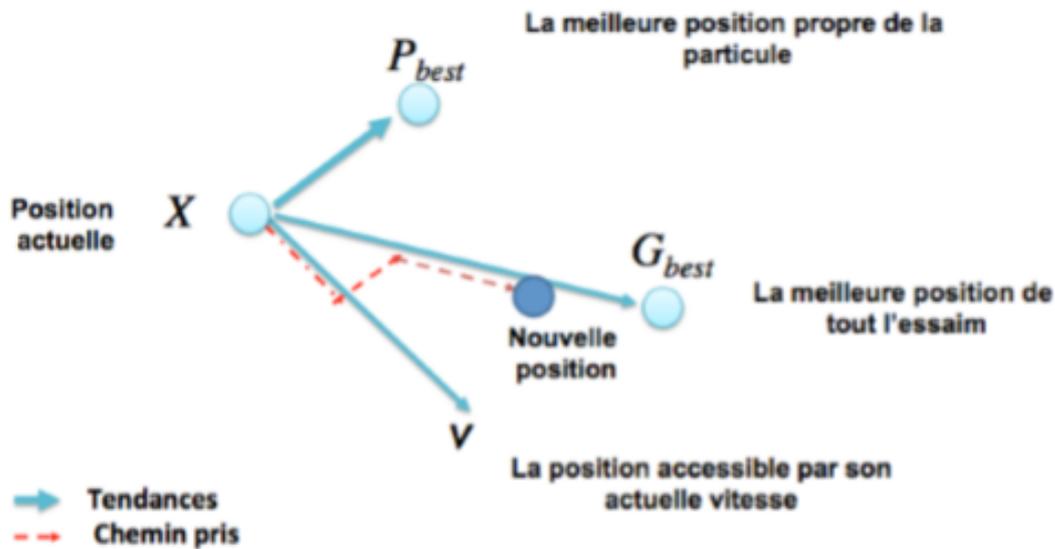


FIGURE 3.8 – Déplacement d’une particule

3.9 Conclusion

Dans ce chapitre nous avons proposé des nouvelles approches d’optimisation d’ordonnancement de Workflow scientifique basé sur méta heuristique l’algorithme génétique et PSO, nous avons expliqué toutes les étapes GA, qui sont représentées par la Population Initiale, Sélection, Croisement, et la mutation, et la description de méta-heuristique PSO comment sa marche par son organigramme.

Dans le prochain chapitre nous ferons une comparaison appliquée entre l’approche l’algorithme génétique et l’approche PSO en termes de temps de réponse et le coût.

Chapitre **4**

Expérimentation et évaluation

4.1 Introduction

Pour tester les nouvelles stratégies et techniques développées par les chercheurs, ces derniers ont besoin d'outils leur permettant d'évaluer ces stratégies avant un déploiement réel dans un environnement où l'on peut reproduire les tests. Avec les Clouds, l'accès aux infrastructures nécessite de payer, même s'il s'agit juste de tester les besoins d'une application. Les outils de simulation offrent des avantages significatifs car ils permettent aux développeurs Cloud de tester gratuitement la performance, l'efficacité et la fiabilité de leurs stratégies dans un environnement reproductible et contrôlable. Ils offrent également la possibilité de changer les paramètres afin d'ajuster et optimiser ces stratégies avant un déploiement réel dans le Cloud.

Ce chapitre est consacré à la réalisation et la concrétisation de notre approche proposée, en étendant le simulateur CloudSim afin de gérer l'ordonnancement de Workflows scientifique dans les environnements de Cloud Computing. Dans un premier temps, nous présentons l'environnement de notre travail, puis nous définissons les différents services du simulateur CloudSim, et finalement nous présentons une série de simulations et leurs interprétations pour mettre en évidence nos propositions.

4.2 Langage et environnement de travail

4.2.1 Langage de programmation JAVA

Le langage Java est un langage généraliste de programmation synthétisant les principaux langages existants lors de sa création en 1995 par Sun Microsystems. Il permet une programmation orientée-objet (à l'instar de SmallTalk et, dans une moindre mesure, C++), modulaire (langage ADA) et reprend une syntaxe très proche de celle du langage C. Outre son orientation objet, le langage Java a l'avantage d'être modulaire (on peut écrire des portions de code génériques, c-à-d utilisables par plusieurs applications [36],

Il s'agit d'un langage de conception très performant qui a été adopté par la majorité des fournisseurs. Ses caractéristiques intégrées de sécurité offrent un sentiment de confiance aux programmeurs comme aux utilisateurs des applications. De plus, Java incorpore des fonctionnalités qui facilitent grandement certaines tâches de programmation avancées comme la gestion des réseaux, la connectivité des bases de données

ou le développement d'applications multitâches, etc. [37][38]

4.2.2 Environnement de travail Netbeans

NetBeans est un environnement de développement intégré EDI(Environnement de développement intégré) , placé en open source par Sun en juin 2000 sous licence CDDL(Common development and distribution licence) . En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C,C++, JavaScript....etc. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme.

L'IDE Netbeans s'enrichit à l'aide de plugins, Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web..etc) .[39]

4.2.3 Simulateur CloudSim

4.2.3.1 Définition

CloudSim est une nouvelle structure de simulation généralisée et extensible qui permet la modélisation des environnements hétérogènes, la simulation et l'expérimentation de Cloud émergent des infrastructures de calcul et des services d'application. CloudSim couvre la plupart des fonctionnalités ayant lieu dans un centre de traitement des données en détail qui contient la simulation de la définition de matériel de centre de traitement des données (Datacenter) en termes de machines physiques composées de processeurs, de dispositifs de stockage, de mémoire et de largeur de bande interne, ainsi que la création et la destruction des machines virtuelles et le plus intéressant la simulation de l'exécution des programmes de l'utilisateur sous

forme de Cloudlet est cela en définissons leur charge et leur nombre.[39]

4.2.3.2 Caractéristiques de CloudSim

CloudSim appuie la recherche et le développement dans le domaine émergent du Cloud Computing et les fonctionnalités suivantes :[39]

- Support pour la modélisation et la simulation à grande échelle d'infrastructure de Cloud Computing, y compris des centres de données sur un seul noeud physique.
- Une plateforme indépendante pour la modélisation des Datacenters, des Brokers, de l'ordonnancement et des politiques d'allocation des ressources.

Parmi les principales caractéristiques de CloudSim, nous pouvons citer :

- La disponibilité de moteur de virtualisation, ce qui facilite la création et la gestion de services virtualisés multiples, indépendants et hébergés sur un noeud du Data-center.
- La flexibilité pour commuter entre l'allocation en espace partagé et en temps partagé des coeurs de traitement aux services virtualisés.

4.2.3.3 Architecture CloudSim

La structure logicielle de CloudSim et ses composants est représentée par une architecture en couches (Figure 4.1), cette architecture est composée de 3 couches :

- **SimJava** [39] La couche la plus basse représente l'outil SimJava, le moteur de simulation d'évènement discret qui met en oeuvre les principales fonctionnalités requises pour des structures de simulation de haut niveau comme la formation d'une file d'attente le traitement d'évènements, la création de composants système (les services, les machines (Host), le centre de données (Data center), le courtier (Broker), les machines virtuelles), la communication entre les composants et la gestion de l'horloge de simulation.
- **Cloud Sim** [39] La prochaine couche identifie l'outil CloudSim qui fournit le nouveau support pour la modélisation et la simulation des environnements virtualisés des centres de données de Cloud comme des interfaces dédiées de gestion de VMs, de mémoire, de stockage et de bande passante.
- **Code User** [39] La plus haute couche dans la pile de simulation est l'User Code qui expose la configuration des fonctionnalités liées aux Hosts (le nombre de

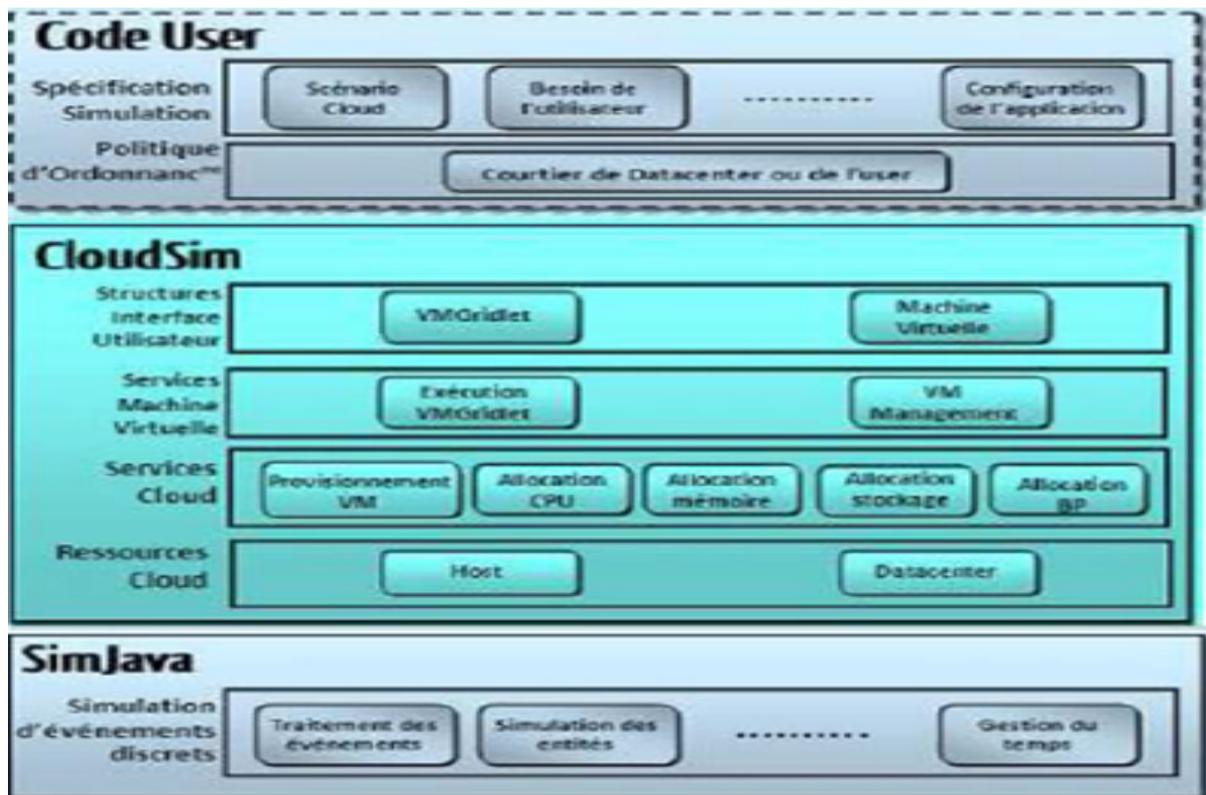


FIGURE 4.1 – Architecture de CloudSim

[39]

machines, leur spécification, etc.), les applications (le nombre de tâches et leurs conditions), les VMs, le nombre d'utilisateurs et leurs types d'application et les politiques d'ordonnancement du Broker.

4.2.3.4 Différente classe de CloudSim

Le simulateur CloudSim est composé de plusieurs classes (figure4.1) que nous pouvons citer les fondamentaux :

— Data Center

Cette classe modélise l'infrastructure du noyau du service (matériel, logiciel) par des fournisseurs de ressources dans un environnement de Cloud Computing. Il encapsule un ensemble de machines de calcul qui peuvent être homogènes ou hétérogènes en ce qui concerne leurs configurations de ressources (mémoire, noyau, capacité, et stockage). En outre, chaque composant de DataCenter instancie un composant généralisé d'approvisionnement de ressource qui implémente un ensemble de politiques d'allocation de bande passante, de mémoire, et des dispositifs de stockage. Voici une image d'un exemple de Data

Center :



FIGURE 4.2 – Data Center de Microsoft
[40]

— **Datacenter Broker**

Cette classe modélise le courtier (Broker), qui est responsable de la médiation entre les utilisateurs et les prestataires de service selon les conditions de QoS des utilisateurs et il déploie les tâches de service aux travers les Clouds. Le Broker agissant au nom des utilisateurs identifie les prestataires de service appropriés du Cloud par le service d'information du Cloud CIS (Cloud Information Services) et négocie avec eux pour une allocation des ressources qui répond aux besoins de QoS des utilisateurs.

— **Virtual Machine**

Cette classe modélise une instance de machine virtuelle (VM), dont la gestion pendant son cycle de vie est une responsabilité de la machine (Host). Un Host peut simultanément instancier de multiples VMs et assigner des coeurs à base de politiques prédéfinies de partage de processeur (espace partagé, temps partagé).

— Cloudlet

Cette classe modélise les services d’application de base du Cloud (la livraison, la gestion de réseau sociale, le déroulement des opérations), qui sont généralement déployés dans des Data Centers.

4.3 Interface principale

La version de Cloudsim n’a pas d’interface graphique, son exécution se fait sur console, donc nous avons créé une interface qui facilite l’accès à la configuration de simulateur.

4.3.1 Configuration de simulation

Pour lancer la simulation d’un Cloud avec notre version étendue du simulateur CloudSim, nous devons configurer les paramètres de simulation dans un premier temps. (Figure 4.3) au-dessous montrent les différentes fenêtres de configuration de simulation qui contient 05 parties principales :

4.3.1.1 le Cloud

la première chose à effectuer est le déploiement d’un nouveau workflow sur le système. l’utilisateur introduit certaines informations concernant le workflow. Il s’agit du nombre de Datacenter requis, nombre de machines virtuelles, nombre de tâches (cloudlet) et nombre de données nécessaire pour chaque application.

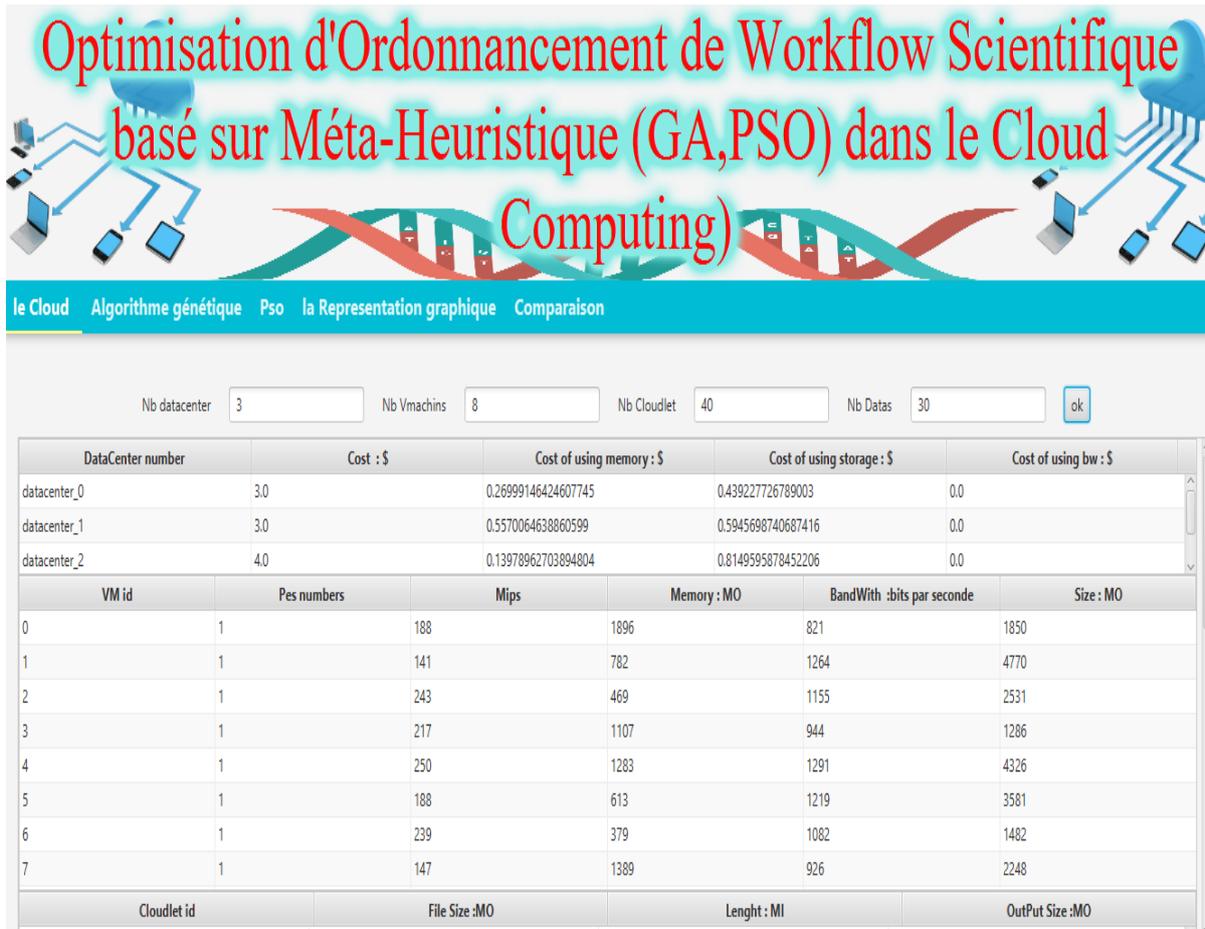


FIGURE 4.3 – interface principale

Après avoir personnalisé les paramètres de simulation, l'utilisateur peut lancer la simulation en cliquant sur le bouton ok, Après la création du Nb data center (nombre centre de donnée), Nb Vm machines (nombre machines virtuel), Nb Cloulet(nombre des taches), Nb datas(nombre des données),le lancement de la simulation, l'utilisateur peut visualiser les résultats et les différents caractéristiques du DataCenter ,virtuel machines,cloudlet et les données (voir Figure 4.3) .

— **Configuration du data centers**

Cette étape consiste à l'utilisateur peut visualiser les résultats et les différents caractéristiques du Data Center (voir Figure 4.4) comme : le nombre du Data Center, le coût de traitement, le coût de la mémoire, le coût de stockage, le coût de la bande passante...etc.

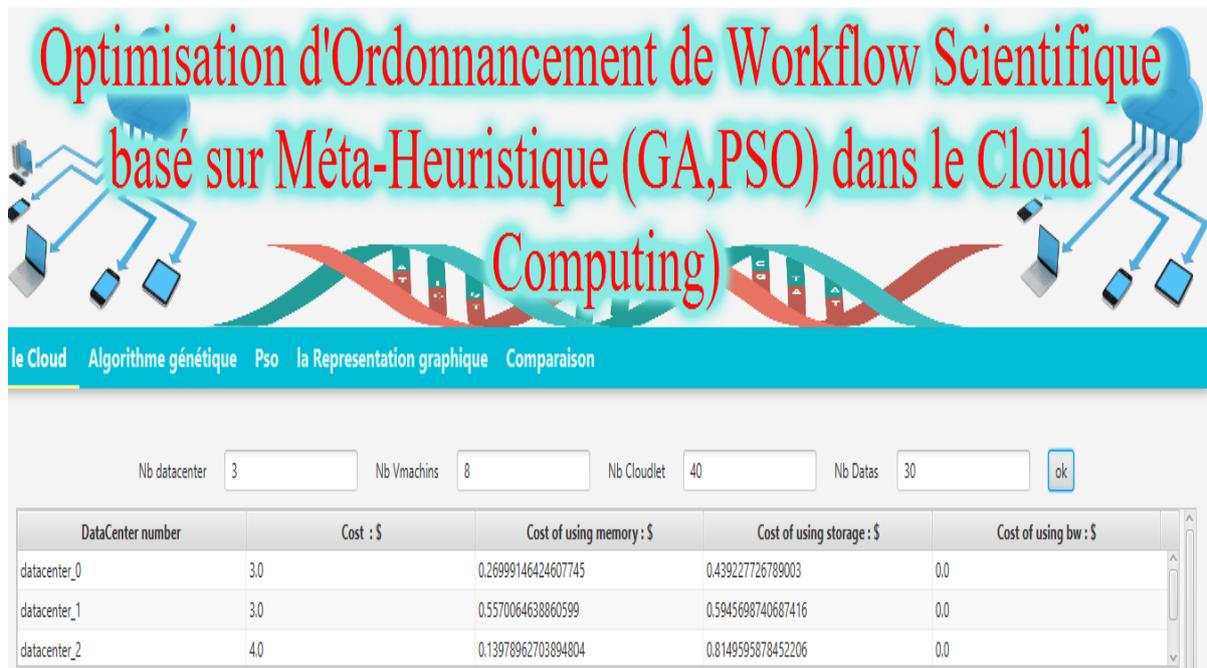


FIGURE 4.4 – Configuration du data centers

— Configuration des machines virtuelles

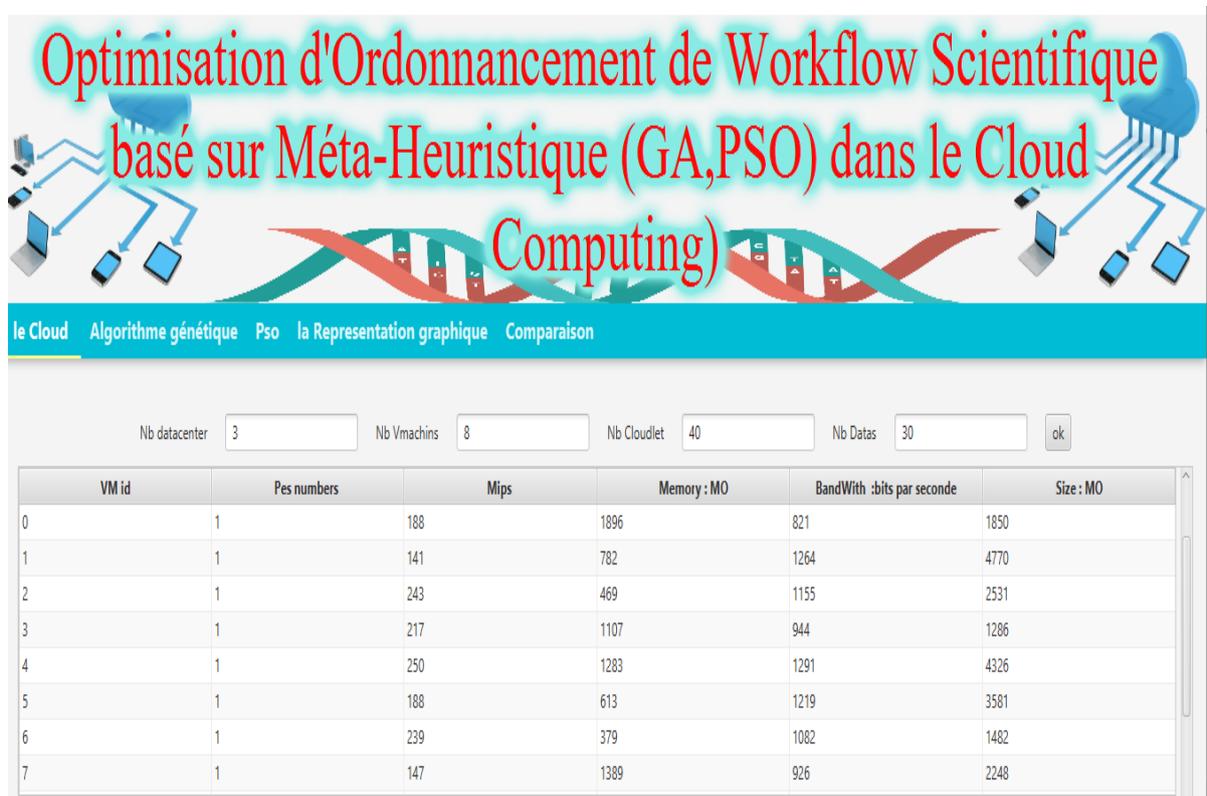


FIGURE 4.5 – Configuration des machines virtuels

La création des machines virtuelles(VM) hétérogènes se fait en cliquant sur le bouton ok (voir Figure 4.5). Pour cela, il faut préciser le nombre des VM, le nombre de processeur, MIPS, RAM, la bande passante, la taille...etc

- **Configuration des cloudlets** La création des Cloudlets se fait en cliquant sur le bouton ok (voir (Figure 4.6)).Pour celà, l'utilisateur voir les différentes caractéristiques des Cloud lets, identificateur de cloud.la taille de fichier .la longueur du cloudlet ...etc

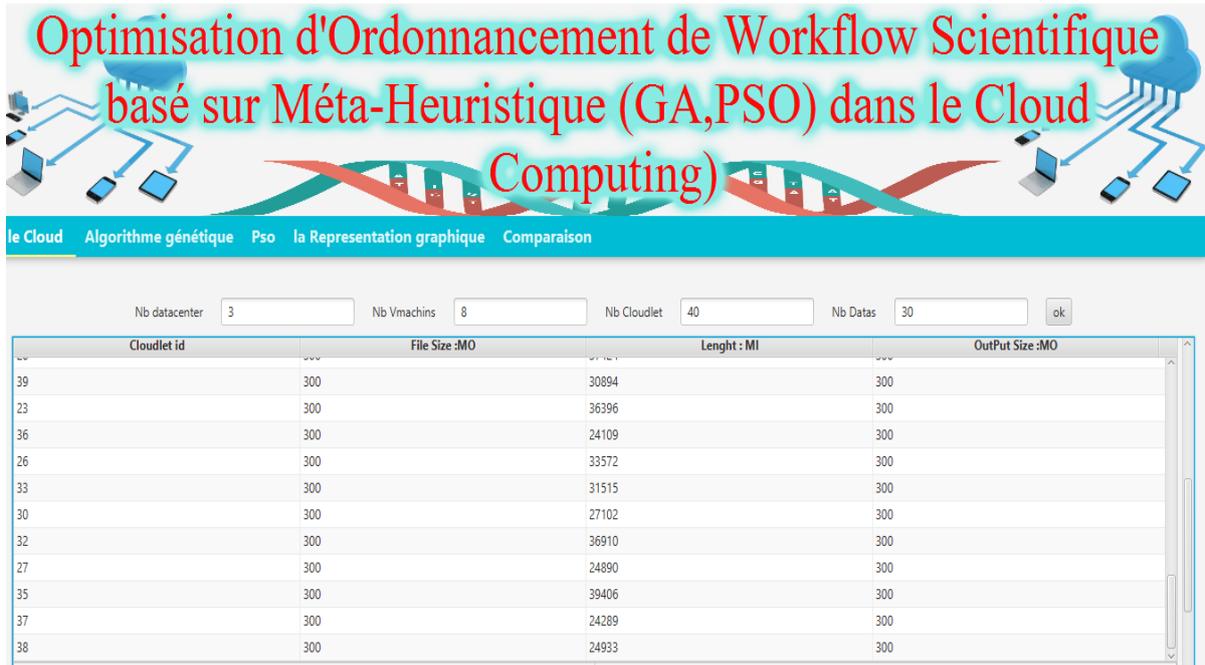


FIGURE 4.6 – Configuration des cloudlets

- **Configuration des données**
La création des données se fait en cliquant sur le bouton ok (voir Figure 4.7).Pour celà, l'utilisateur voir les différentes caractéristiques des données en précisant la donnée, la taille et nombre de répliques.



FIGURE 4.7 – Configuration des données

4.3.1.2 algorithme génétique

On détermine ici le nombre de population représentés par les chromosomes, ainsi que le nombre de générations pour obtenir un nouvel individu (voir Figure 4.8).



FIGURE 4.8 – algorithme génétique

4.3.1.3 PSO

La méthode PSO est une méta-heuristique qui permet de trouver l'optimum d'une fonction en un temps de traitement raisonnable, sauf pour les instances de grande taille où le calcul scientifique est intensif ce qui nécessite un temps de calcul considérable. Nous avons utilisé cet algorithme dans cette application pour comparer avec l'algorithme génétique en termes de makespan et le coût.

4.3.1.4 la représentation graphique

la présentation graphique des deux solutions proposées algorithme génétique et PSO en ce qui concerne ce dont chaque algorithme a besoin, ce dont chaque tâche(cloudlet) a besoin de données et ce dont une machine virtuel a besoin de tâches(cloudlet) et de données.

4.3.1.5 comparaison entre deux Aproches

Une comparaison graphique pour montrer le temps de réponse et le coût..

4.3.2 Résultats expérimentaux

Pour mettre une valeur de nos approches,et pour le but d'étudier le comportement de nos proposition et d'analyser ses résultats obtenus par la simulation, nous allons les comparer les deux approches. Plusieurs séries de simulation ont été lancées selon plusieurs paramètres. Pour montrer les deux approches, nous allons présenter le résultat numérique obtenus ainsi que les résultat graphique réalisé.

Dans cette simulation, nous avons créé deux DataCenter contenant :

- 04 VM, chaque Vm avec une vitesse variante en MIPS entre (100, 1000).
- bande passante (100,2000).
- Nous supposons que nous avons reçu 06 cloudlet de longueur entre (2000, 4000).
- 5 données de taille (300 mo, 1000 mo)
- les résultats obtenus par les simulations sont résumés :

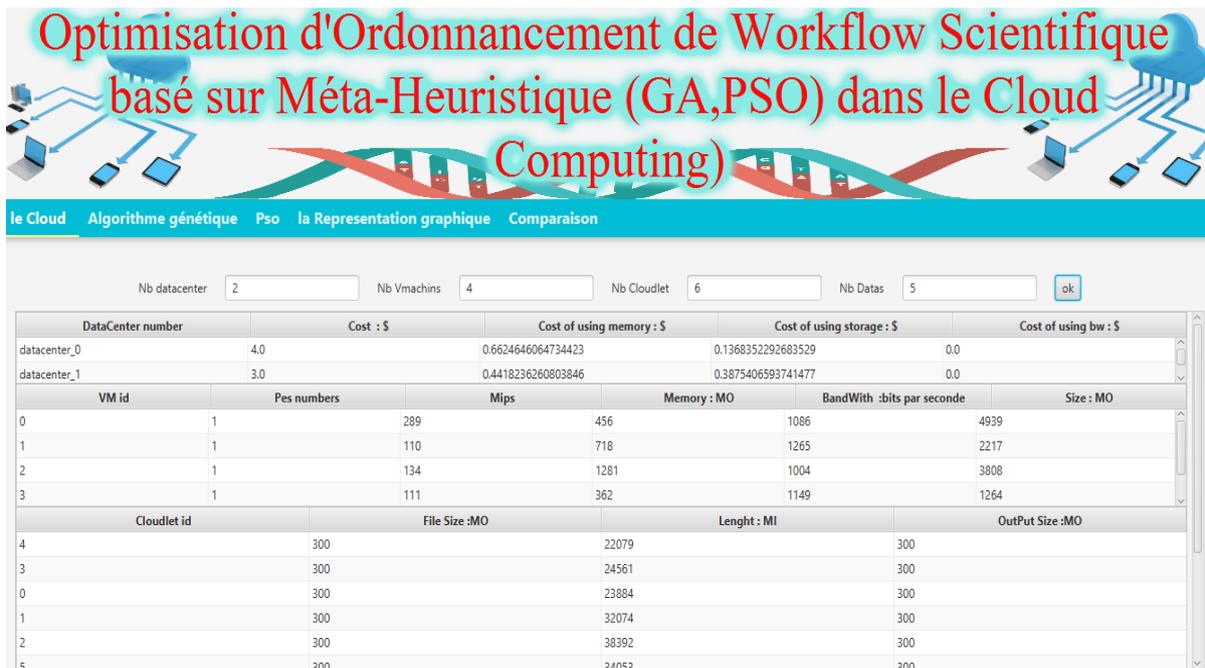


FIGURE 4.9 – interface principale avec les caractéristiques décrites ci-dessus

— présentation graphique de la simulation de l’algorithme génétique

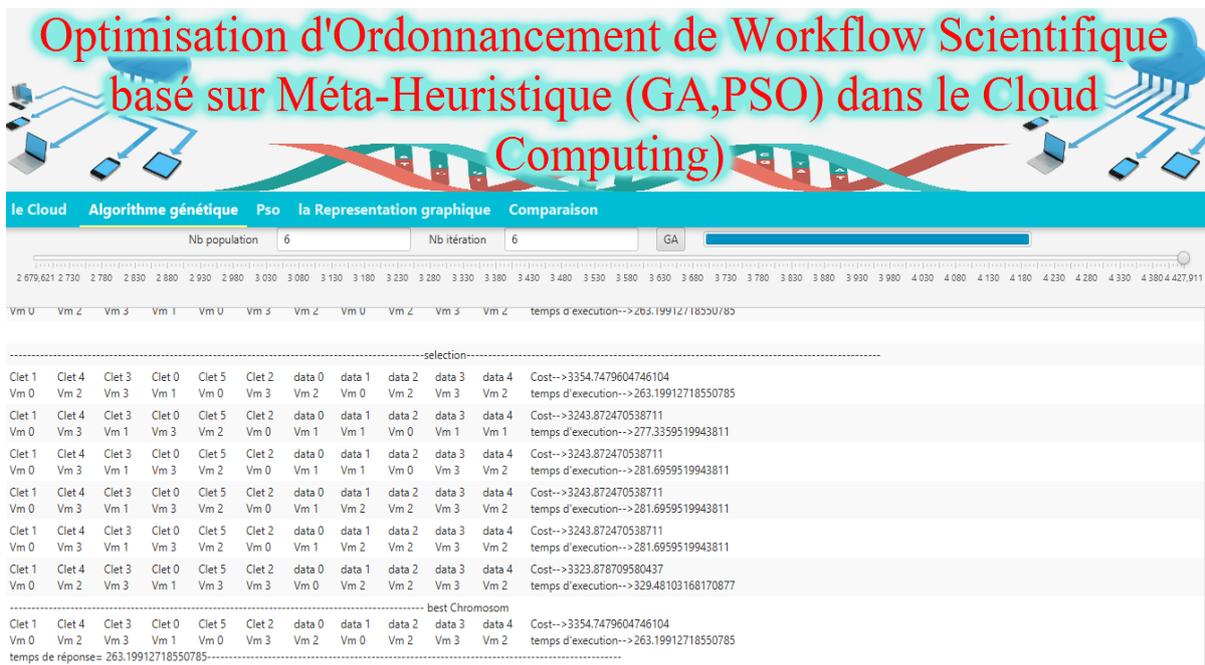


FIGURE 4.10 – présentation graphique de la simulation de l’algorithme génétique

— présentation graphique de la simulation de PSO

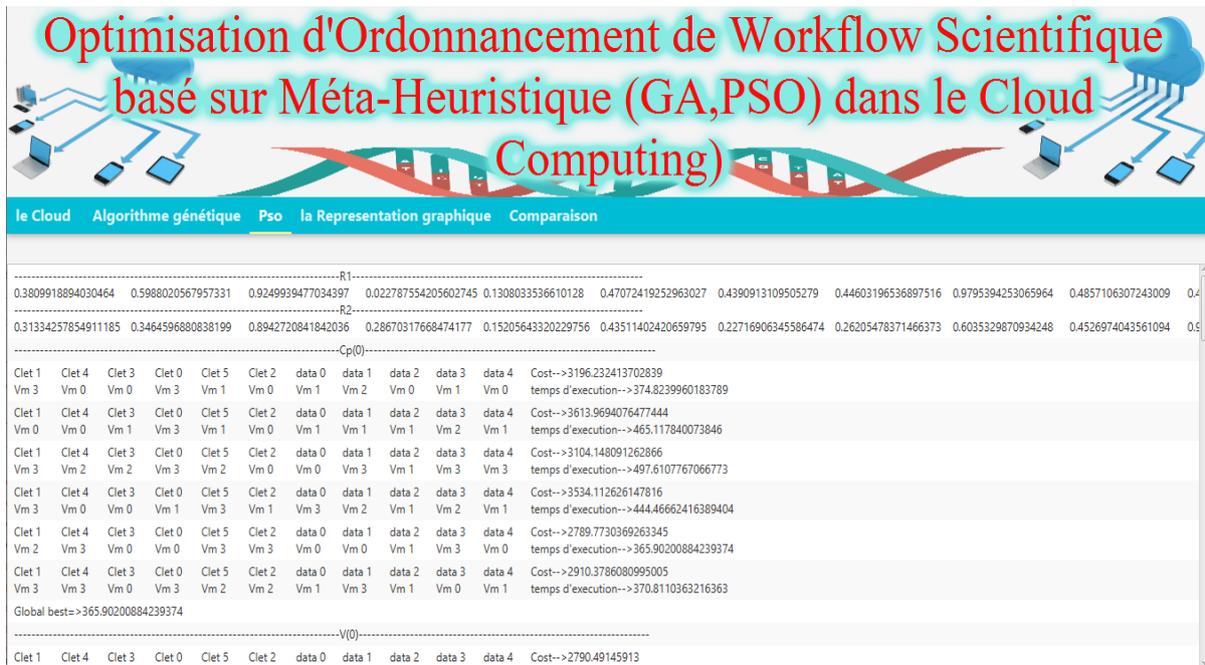


FIGURE 4.11 – présentation graphique de la simulation de PSO

— la Représentation graphique

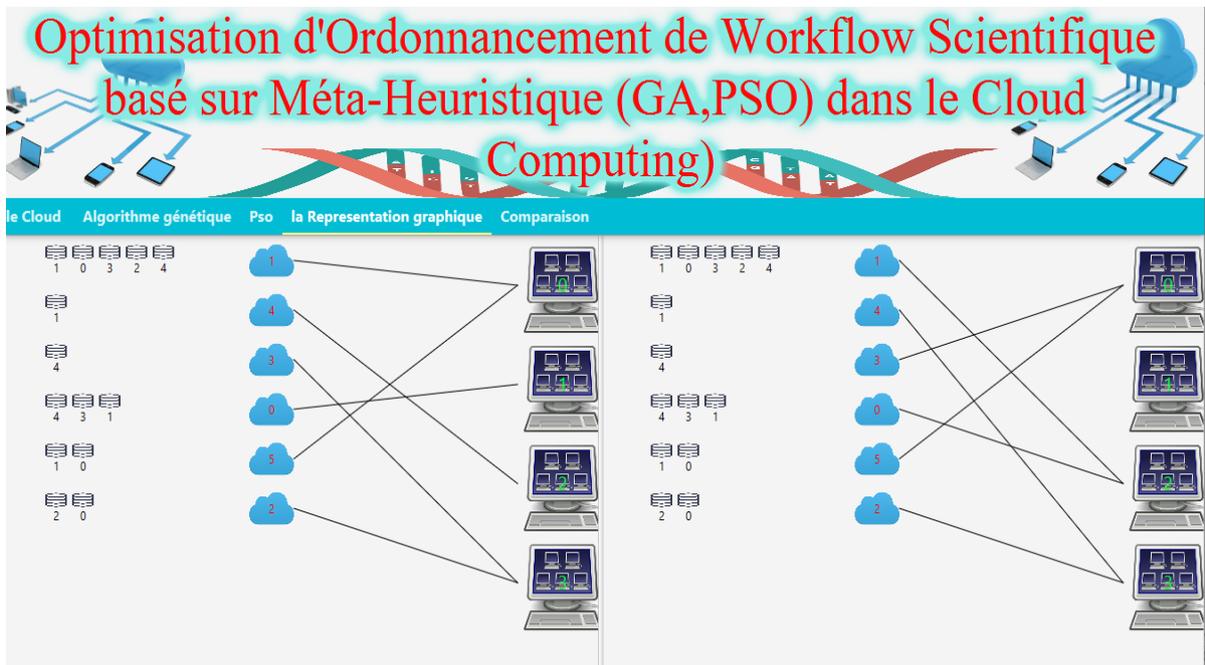


FIGURE 4.12 – la Représentation graphique

- présentation graphique de la simulation de la comparaison (Response time) temps de reponse et le coût entre deux Approches

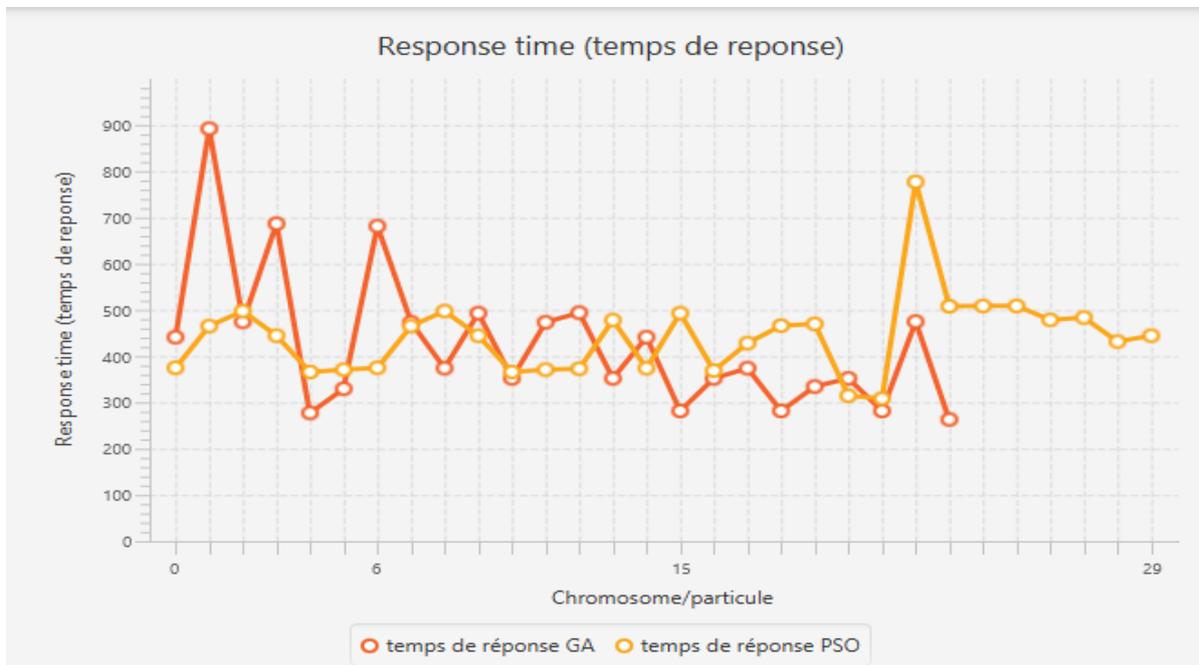


FIGURE 4.13 – présentation graphique de la simulation de la comparaison (Response time) temps de reponse et le coût entre deux Approches

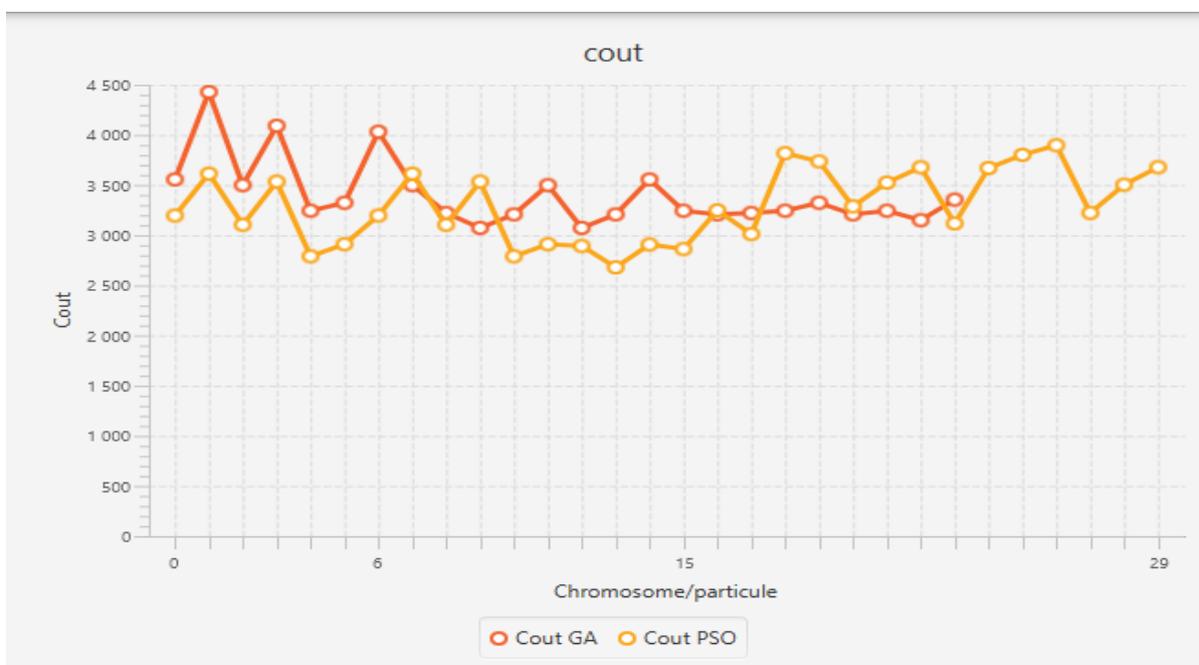


FIGURE 4.14 – comparaison selon le coût entre deux Approches

4.4 Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre application ainsi que les résultats obtenus. Aussi, nous avons réalisé plusieurs séries des simulations dans le but de comparer approche GA avec l'approche PSO tout en variant différents paramètres comme : le nombre Data, le nombre de VM, nombre de Cloudlet . Les résultats montrent que l'utilisation de l'approche PSO réduit le temps de réponse et le coût par rapport GA.

Conclusion générale

Le Cloud Computing agrandit le domaine des systèmes informatiques distribués en offrant des services Internet de pointe qui complètent des fonctionnalités de l'informatique distribuée fournie par le Web, Grille et Peer to Peer. Le Cloud est une technologie de calcul naissante qui se consolide rapidement comme Prochaine étape normale dans le développement et le déploiement d'un nombre Croissant des applications réparties. Les applications workflows scientifiques de traitement de données intensives Traitent des volumes extrêmement importants, de plus, elles nécessitent des Temps de traitement très rapides .La plupart de ces applications sont déployées sur le Cloud pour tirer parti de cette infrastructure, notamment l'évolutivité, la Fiabilité et la haute disponibilité, avec un coût réduit.

La théorie de l'ordonnancement est une branche de la recherche opérationnelle qui s'intéresse au calcul de dates d'exécution optimales de tâches. Pour cela, il est très souvent nécessaire d'affecter même temps les ressources nécessaires à l'exécution de ces tâches. Un problème d'ordonnancement peut être considéré comme un sous-problème de planification dans lequel il s'agit de décider de l'exécution opérationnelle des tâches planifiées. Pour planifier et ordonner de ressources dans le Cloud Computing, nous avons proposé une stratégie d'optimisation d'ordonnancement de workflow Scientifique basé sur Méta-Heuristique (GA, PSO) dans le Cloud. Dans ce travail, nous avons simulé nos approches, Nous avons étendu le simulateur Cloud-Sim Pour implémenter les approches proposés, tester l'algorithme génétique par une méthode de sélection par roulette parce qu'est la plus connue et la plus utilisée.

Comme perspective nous envisageant dans le futur de faire :

- Etendre notre solution par l'intégration d'un service d'optimisation d'allocation des ressources machines virtuelles.
- Tester notre algorithme avec d'autre méthode du sélection d'ordonnancement.

Bibliographie

- [1] Sonia Yassa. Allocation optimale multicontraintes des workflows aux ressources d'un environnement Cloud Computing. Université de Cergy Pontoise, Français, 2014.
- [2] Hamza Ouarnoughi. Placement autonome de machines virtuelles sur un système de stockage hybride dans un cloud IaaS. Université de Bretagne occidentale - Brest, Français, 2017.
- [3] Daniel. J, Abadi Yale. Data Management in the Cloud : Limitations and Opportunities University New Haven, CT, USA.
- [4] D. Yuan, Y. Yang, X. Liu, J. Chen. A data placement strategy in scientific cloud workflows, 2010.
- [5] Mustafizur Rahman, Rafiul Hassan, Rajiv Ranjan , and Rajkumar Buyya. Adaptive workflow scheduling for dynamic grid and cloud computing environment, 2013.
- [6] Qing Zhao , Congcong Xiong , Kunyu Zhang , Yang Yue and Jucheng Yang. A Data Placement Algorithm for Data Intensive Applications in Cloud, 2016.
- [7] M. Bedjaoui Mohammed Arslane, M. Khiat Menouar. Etude de la consommation d'énergie dans le Cloud Computing. Mémoire de fin d'étude, Université Abou Bakr Belkaid– Tlemcen, Algérie, 2018.
- [8] EREMY Geelan. Twenty one experts define cloud computing. Virtualization , Electronic Magazine, , 2008 [http ://virtualization.scon.com/](http://virtualization.scon.com/),(consulter 2008)
- [9] Peter Mell Timothy Grance. NIST Special Publication 800-145 The NIST Definition of Cloud Computing.
- [10] R. Buyya, C. S. Yeo and S. Venugopal. Market-Oriented Cloud Computing : Vision, Hype, and Reality for Delivering IT Services as Computing Utilities,

- IEEE International Conference on High Performance Computing and Communications, pages 5-13, 2008.
- [11] Khadidjatou Iman BAMBÀ. Fondamentaux du Cloud Computing, 2014.
- [12] FAREH. Mohamed El-Kabîr. Une approche basée agents pour l'allocation des ressources dans le Cloud Computing. Mémoire de fin d'étude, Université Mohamed Khider- Biskra, Algérie, 2019.
- [13] S. François. La Virtualisation. Projet de recherche et communication scientifique, Université libre de Bruxelles. 2010. [cf. p. 7].
- [14] BERRAMDAN Marouan. Équilibrage de charge dans le Cloud Computing, Mémoire de fin d'étude, Université Mohamed Khider – BISKRA, Algérie, 2019.
- [15] T. HICHAM, E.AHMED, M. TALEA, E.B. LAHMAR. Vers une Architecture de Sécurité de Cloud Computing. Conférence, 2014.
- [16] ZERROUKI Ahlem, ZERROUKI Imane. Ordonnancement des workflows scientifiques dans un environnement Cloud Computing, Mémoire de fin d'étude, unv saida-Dr Moulay Tahar, Algérie, 2017
- [17] I. Foster, Y. Zhao, I.Raicu, S. Lu. Cloud Computing and Grid Computing, Washington, DC, USA. IEEE Computer Society. Pages 1–10. 2008.
- [18] Li. M, Qiu. J, Niu. W, Gao. Z, Zong and X. Qin, Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems, EEE/WIC/ACM International Conférence on Web Intelligence and Intelligent Agent Technology, pages 561-564, 2010.
- [19] PK.Gupta, N.Rakesh, Different job scheduling Methodologies for web application and web server in a cloud computing environment. International Conférence on Emerging Trends in Engineering and Technology, IEEE XplorePress, pages 569-572, 2010
- [20] B. Yang, X. Xu, F .Tan, D.H.Park. An utility based job scheduling algorithm for cloud computing considering reliability factor. International Conférence on Cloud and Service Computing, IEEE Xplore Press, Hong Kong, pages 95-10, 2011.
- [21] Jiandun Li a , Junjie Peng .a,b, Zhou Lei a , Wu Zhang. An Energy-efficient Scheduling Approach Based on Private Clouds, Journal of Information Computational Science, pages 716–724, 2011 Available at <http://www.joics.com>.

- [22] Bessai. K. Gestion optimale de l'allocation des ressources pour l'exécution des Processus dans le cadre du Cloud .thèse de Doctorat, Université Paris1 Panthéon- Sorbonne, 2014.
- [23] Park. G, L. Shirazi, and Marquis. J. A new approach for duplication based scheduling for distributed memory multi processor system, pages 157-166, 1997.
- [24] BISWAJIT Mahanty. Particule Swarm Optimization, département engenering indian institute of technology kharagpur, 2020.
- [25] Djebbar Asma insaf.Optimisation d'Ordonnancement d'Allocation de ressource dans le Cloud Computing, thèse doctorat, Université d'Oran, Algérie, 2016.
- [26] Luiz. F, BITTENCOURT, Edmundo. R, M. MADEIRA et Nelson. L, Da FONSECA. Scheduling in Hybrid Clouds. In : IEE, pages. 42-47,2012.
- [27]] Ling Fang ZENG, Bharadwaj VEERAVALLI et Xiaorong LI. SABA : A Security- Aware and Budget Aware Workflow Scheduling Strategy in Clouds,pages. 141-151, 2015.
- [28] L. ZHAO, Y. REN et K. SAKURAI. Reliable Workflow Scheduling with Less Resource Redundancy. In : Parallel Computing 39.10, pages. 567- 585, 2013,.
- [29] Sarah Cohen-Boulakia, Christine Froidevaux, Jiuqiang Chen. Réécriture de workflows scientifiques et provenance, 2012.
- [30] KOUIDRI Siham. Gestion de données sur le Cloud Computing, Thèse de Doctorat, Université Mentouri d'Oran, Algérie, 2020.
- [31] Zoulikha TITRAOUI. L'Optimisation multi Objectifs pour l'Ordonnancement de tâches dans le Cloud Computing, Mémoire de fin d'étude, Université de M'sila, Algérie, 2019.
- [32] Levet. A, Bordes, Clément. Y, et al. Acute. Aquatic toxicity of organic solvents modeled by QSARs. J Mol Model 22, 288 2016. (Article available At) [HTTPS ://doi.org/10.1007/s00894-016-3156-0](https://doi.org/10.1007/s00894-016-3156-0)
- [33] Ali LEMOUARI. Systèmes Adaptatifs et Apprentissage Incrémental de l'inspiration biologique à la résolution collective des problèmes, Thèse de Doctorat, Université Mentouri de Constantine, Algérie, 2009..

-
- [34] Zemzami, M.Elhami, N.Makhloufi, A. Itmi, M, Hmina. N. Application d'un modèle parallèle de la méthode PSO au problème de transport d'électricité, 2016.
- [35] COOREN .Y. Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique. Thèse de Doctorat, Université de Paris 12 Val de Marne, France. 2008.
- [36] Djelil, F. Montesinos, M. T. S, Gilliot, J. M. Une approche didactique pour l'introduction de la Programmation Orientée-Objet en classe, Février 2020.
- [37] CLAUDE Delannoy. Programmer en Java .Eyrolles,2007.
- [38] Bruce Eckel. Thinking in Java. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [39] CHEBLI Asma. Réalisation d'un Service de gestion de cohérence de donnée hybride dans un environnement Cloud Computing, Mémoire de fin d'étude, unv saida-Dr Moulay Tahar, Algérie, 2017.
- [40] M. Bedjaoui Mohammed Arslane, M. Khiat Menouar. la Consommation d'Energie dans le Cloud Computing , Mémoire de fin d'étude , Université Abou Bakr Belkaid– Tlemcen, Algérie , 2018