République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

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

MEMOIRE DE MASTER

OPTION :
Réseaux Informatique et Systèmes Répartis

Thème :

Ordonnancement des workflows scientifiques dans un environnement Cloud Computing

Présenté par :
ZERROUKI Ahlem
ZERROUKI Imene

Encadré par :
Dr. KOUIDRI Siham

Promotion : Septembre 2020

REMERCIEMENTS

Avant tout nous remercions, le dieu qui nous a donner la puissance et la volonté, la patiance pour accomplire ce travail.

Nous tenons à remercier notre encadreur **Dr. KOUIDRI Siham** , pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nos plus vifs remerciements vont aussi aux membres du jury qui ont bien voulu consacrer du temps à expertiser ce modeste travail.

Je remercie également **Mm.Kouidri Shaima** pour m'avoir donner l'aide qu'il m'a accordé .

Notre profonde reconnaissance va également à tous nous enseignants. C'est aussi grâce à eux et à tout ce qu'ils nous ont transmis que ce mémoire a pu voir le jour.

Dédicaces

Avant tout je remercie, le dieu qui ma donner la puissance et la volonté, la patiance pour accomplire ce travail.

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Que ce modeste travail, soit l'éxaucement de vos voeux tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler

A tous mes oncles et tantes et leurs familles et enfants.

A mon camarade **ZERROUKI Imene** pour son soutien et sa disponibilité son aide et ses conseils durant la réalisation de ce travail, sans oublié sa famille.

A tous mes aimables amis et camarades.

A toute la promotion d'informatique 2020

Zerrouki Ahlem

Dédicaces

Avant tout je remercie, le dieu qui ma donner la puissance et la volonté, la patiance pour accomplire ce travail.

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Que ce modeste travail, soit l'éxaucement de vos voeux tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler

A tous mes oncles et tantes et leurs familles et enfants.

A mon camarade **ZERROUKI Ahlem** pour son soutien et sa disponibilité son aide et ses conseils durant la réalisation de ce travail, sans oublié sa famille.

A tous mes aimables amis et camarades.

A toute la promotion d'informatique 2020

Zerrouki Imene

Table des matières

Liste des figures

Liste des figures

Liste des tableaux

Acronyms

NIST : The National Institute of Standards and Technology.

IaaS : Infrastructure as a service.

SaaS : Software as a service.

PaaS : Platform as a service.

VM : Virtual Machine.

GA : Genetic Algorithm.

FCFS : First Come First Served.

RR : Round Robin algorithme.

SJF : Shortest Job First la tâche la Plus courte d'abord.

PSO : Particle Swarm Optimization.

QoS : quality of service.

Résumé :

Le Cloud Computing est de plus en plus reconnu comme une nouvelle façon d'utiliser les services à la demande, de calcul, de stockage et de réseau de manière transparente et efficace. L'environnement Cloud Computing se compose de grands clients qui demandent des ressources cloud. De nos jours, le problème de planification des tâches est le sujet de recherche actuel dans le cloud computing. La planification d'une tâche consiste à l'assigner à une ressource (machine virtuelle), afin d'atteindre un objectif final tel que la minimisation du temps total d'exécution du workflow. La planification des workflows est considéré comme un problème NP-complet, c'est-à-dire un problème non résoluble dans un temps polynomial avec les ressources actuelles. Dans cet article, nous fournissons une revue de la littérature des travaux effectués dans ce domaine et une description de l'approche proposée.

Mots clés :

Cloud Computing, Workflow Scientific, Ordonnancement , Algorithme génétique

Abstract :

Cloud Computing is increasingly recognized as a new way to use on-demand, computing, storage and network services in a transparent and efficient way. Cloud Computing environment consists of large customers requesting for cloud resources. Nowadays, task scheduling problem is the current research topic in cloud computing .The scheduling of a task consists of assigning it to a resource (virtual machine), in order to fulfill a final goal such as minimizing total workflow execution time .the scheduling of workflows is considered to be an NP-complete problem, i.e. a problem not solvable within polynomial time with current resources. In this paper, we provide literature review of work done in this area and description of the proposed approach.

Keywords:

Cloud Computing, Workflow Scientific, Scheduling, Genetic Algorithm.

ملخص :

يتم التعرف على الحوسبة السحابية بشكل متزايد كطريقة جديدة لاستخدام خدمات الحوسبة والتخزين والشبكات عند الطلب بطريقة شفافة وفعالة. تتكون بيئة الحوسبة السحابية من عدد كبير من العملاء يطلبون موارد السحابة. في الوقت الحاضر ، تعد مشكلة جدولة المهام هي موضوع البحث الحالي في الحوسبة السحابية. تتكون جدولة المهمة من تعيينها إلى مورد(جهاز افتراضي) ، من أجل تحقيق هدف نهائي مثل تقليل إجمالي وقت تنفيذ سير العمل. تعتبر هده المشكلة غير قابلة للحل خلال وقت متعدد الحدود مع الموارد الحالية. في هذه الورقة ، نقدم مراجعة الأدبيات للعمل المنجز في هذا المجال ووصف النهج المقترح

الكلمات المفتاحية :

الحوسبة السحابية ، سير العمل العلمي ، الجدولة ، الخوارزمية الجينية

Introduction générale

Les applications de workflow sont devenues un paradigme attrayant pour la programmation des infrastructures informatiques distribuées, qui sont largement appliqués dans divers domaines de calcul scientifique comme l'astronomie, la bio-informatique et la physique. Avec la complexité croissante des systèmes informatiques scientifiques, les applications de workflow sont devenues des applications de Big Data, qui exige des infrastructures à grande échelle pour pouvoir être exécutées un délai raisonnable.

Parmi ces infrastructures, l'environnement de cloud computing est l'un des intérêts particuliers. Le cloud computing est une grande échelle informatique distribuée portée par les besoins émergents de et des opérations système efficaces, dans lesquelles les infrastructures sont disponible dans un système de paiement à l'utilisation et peut fournir une mise à l'échelle dynamique en réponse aux besoins des applications de workflow.

Récemment, le des systèmes de stockage cloud et non cloud sont respectivement déployés pour les scientifiques biomédicaux de mener des expériences contrôlées sur comparaison des performances. Les résultats montrent que le système cloud surpasse le système non cloud en termes de temps d'exécution, cohérence et amélioration de l'efficacité. En outre, le type et le nombre de ressources VM pour l'exécution du workflow peuvent être provisionné à la demande dans les nuages.

Bien qu'une quantité infinie de accessibles dans le contexte des nuages, les utilisateurs doivent porter une attention particulière au coût économique encouru par la ressource bail. Les fournisseurs de cloud commerciaux facturent généralement les utilisateurs modèle de tarification horaire. Par conséquent, le coût est calculé en fonction du modèle d'unité de temps au lieu de l'utilisation réelle des ressources, qui signifie que les utilisateurs doivent payer pour toute l'heure même s'ils ne louer la VM pour une seconde.

Problématique :

Workflows scientifiques inclure des centaines ou des milliers de tâches de calcul qui sont 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. Ces facteurs poussent les workflows scientifiques à produire un nombre élevé de combinaisons pour répartir leurs tâches sur les ressources informatiques. 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 . Il est bien connu comme un problème NP-complet.
Pour faire face à ce problème, les systèmes informatiques ont une étape de planification. la production d'une configuration de planification optimale devient un problème sérieux à mesure que le nombre de tâches augmente.

Objectif :

Le but de ce mémoire est d'étudier les problèmes d'ordonnancement et d'optimisation dans le contexte de plateformes de type cloud. Proposer de solution pour minimiser le temps d'exécution de l'application. Cette solution peut se présenter sous la forme d'algorithme d'optimisation qu'il soit métaheuristiques (Algorithme génétique).

Le travail que nous avons mené dans le cadre de la problématique est résume dans le présent document qui structuré en quatre chapitres :
Chapitre 1 : présente les notions et concepts de base du Cloud computing, ses services, ses types.
Chapitre 2 : présente un état de l'art sur l'ordonnancement des workflows dans le cloud computing.
Chapitre 3 : Est dédié à la conception de notre approche proposée. Dans un premier temps, nous modélisons notre problème. En deuxième lieu nous présentons notre algorithme proposé avec leurs différentes étapes.
Chapitre 4 : sera consacré à l'implémentation de notre approche, les outils de développement, ainsi que les résultats obtenus seront tous présentés dans ce chapitre.

Chapitre 1

CLOUD COMPUTING

1.1 introduction :

L'avancement rapide des technologies de l'information et de la communication a permis le développement de nouveaux paradigmes informatiques, où les techniques de traitement, de stockage, de communication, de partage et de diffusion de l'information ont radicalement changés. Les individus et les organisations sont de plus en plus recours à des serveurs externes pour le stockage et la diffusion efficace et fiable d'informations.
Le **Cloud Computing**, ou " informatique dans les nuages ", est une nouvelle technologie informatique qui permet le déplacement des traitements et fichiers informatiques de l'ordinateur local vers des serveurs distants. Elle consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui grâce à une connexion internet.
Basé sur le principe de fournir les ressources informatiques sous forme de services et de facturer leur utilisation en fonction de leur usage (**pay as you go** en anglais), le **Cloud Computing** permet d'effectuer des économies d'échelle grâce à l'externalisation de ces ressources vers des fournisseurs spécialisés.
C'est le cas par exemple de Google App Engine, Amazon EC2 ou Microsoft Azure, les offres permettant d'utiliser les infrastructures de calcul et de communication de Google, Amazon ou Microsoft comme des services utilitaires.
Ces offres proposent différents types de service généralement spécifiques à l'usage que les utilisateurs peuvent en faire. Une architecture en couches a été proposée pour catégoriser les différents types de services et les usages correspondants. Cette architecture est appelée **modèle SPI** (**Software** / **Platform** / **Infrastructure**).

1.2 Historique :

Les fondations de cloud computing peuvent être retracées jusqu'aux années soixante ou John McCarthy, pionnier de l'intelligence artificielle ,a pour la première fois formule l'idée d'une " informatique utilitaire "en anglais utility computing. L'idée consiste à pouvoir fournir à l'utilisateur de la puissance de calcul ,des capacités de stockage et des capacités de communication, de la même façon que l'on lui fournit l'électricité ou l'eau dans les réseaux publics.

Bien avant la naissance du terme de Cloud computing, les informaticiens utilisaient déjà des services de Cloud computing comme le webmail2,le stockage de données en ligne(photos, vidéos...)ou encore le partage d'informations sur les réseaux sociaux. Dans les années 90,un autre concept avait déjà prépare le terrain au Cloud computing.

Il s'agit de L'ASP(Application Service Provider) qui permettait au client de louer l'accès a un logiciel installé sur les serveurs distants d'un prestataire ,sans installer le logiciel sur ses propres machines. Le Cloud computing ajoute à cette offre la notion d'élasticité avec la possibilité d'ajouter de nouveaux utilisateurs et de nouveaux services d'un simple clic de souris.

Il est communément admis que le concept de Cloud Computing a été initie par le géant Amazon en 2002. Le cybermarchand avait alors investi dans un parc informatique an de pallier les surcharges des serveurs dédies au commerce en ligne constatées durant les fêtes de fin d'année.

A ce moment-là, Internet comptait moins de 600 millions d'utilisateurs mais la fréquentation de la toile et les achats en ligne étaient en pleine augmentation. En dépit de cette augmentation, les ressources informatiques d'Amazon restaient peu utilisées une fois que les fêtes de fin d'année étaient passées.

Ce dernier a alors eu l'idée de louer ses capacités informatiques le reste de l'année à des clients pour qu'ils stockent les données et qu'ils utilisent les serveurs.

Ces services étaient accessibles via Internet et avec une adaptation en temps réel de la capacite de traitement, le tout facturé a la consommation. Cependant, ce n'est qu'en 2006 qu'Amazon comprit qu'un nouveau mode de consommation de l'informatique et d'internet faisait son apparition.

Réalisant ce qu'ils pourraient faire de toute cette puissance, de nombreuses compagnies ont ensuite commencé a montrer un certain intérêt a échanger leurs

anciennes infrastructures et applications internes contre ce que l'on appelle les "pay per-use service" (services payés a l'utilisation).

1.3 Cloud Computing :

1.3.1 Définition :

De nombreuses définitions ont été proposées pour le **Cloud Computing**, en mettant l'accent sur les différents aspects qui caractérisent le paradigme. Nous considérons la définition de Cloud Computing proposée par l'Institut national de la norme et de la technologie (NIST), [1], car elle illustre les aspects essentiels du Cloud .

Le cloud computing est un modèle pour permettre un accès réseau omniprésent, pratique et à la demande à un partage pool de ressources informatiques configurables qui peut être rapidement approvisionné et publié avec un effort de gestion minimal ou une interaction avec le fournisseur de services.

L'expression " **Cloud Computing** ", ou " informatique dans les nuages " peut se définir comme une approche visant à disposer d'applications, de puissance de calcul, de moyens de stockage, etc. comme autant de " services ". Ceux-ci seront mutualisés, dématérialisés (donc indépendants de toutes contingences matérielles, logicielles et de communication), contractualisés (en termes de performances, coûts...), évolutifs (en volume, fonction, caractéristiques ...etc) et en libre-service.

La figure suivante présente une description générale du Cloud Computing.

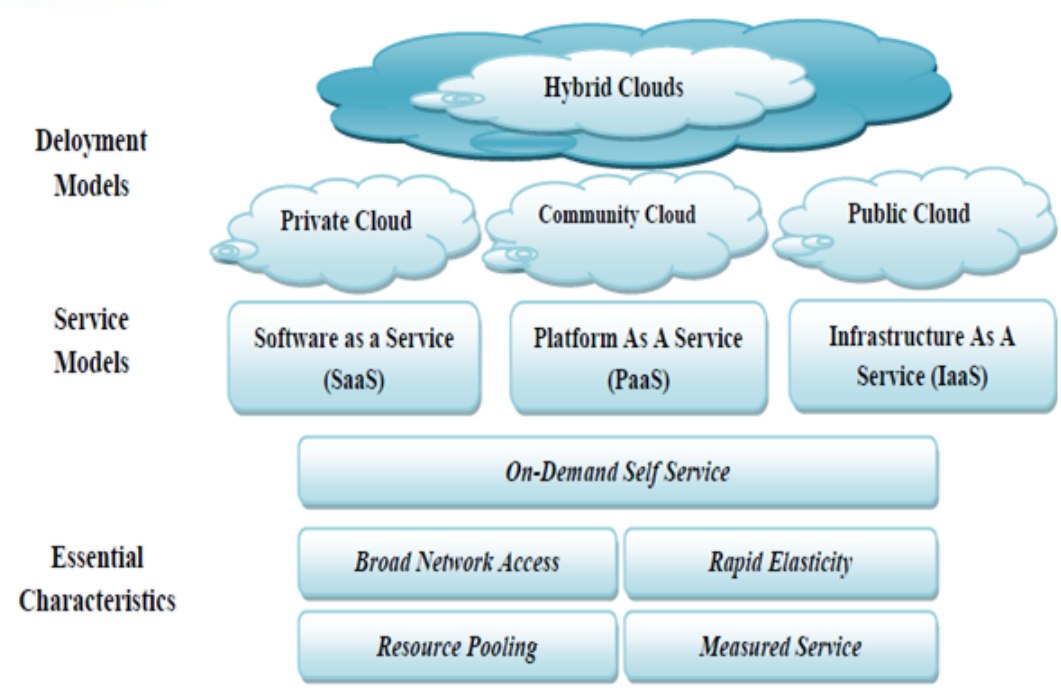

FIGURE 1.1 – Description du Cloud Computing

1.3.2 Caractéristiques essentielles : :

Le Cloud computing se distingue des solutions traditionnelles par les caractéristiques suivantes[1] :

- **Libre-service à la demande :** Un consommateur peut fournir unilatéralement des capacités informatiques, telles que l'heure du serveur et le stockage réseau, selon les besoins, automatiquement sans nécessiter d'interaction humaine avec chaque fournisseur de services.

- **Large accès au réseau :** Les capacités sont disponibles sur le réseau et accessibles via des mécanismes standard qui favorisent l'utilisation par des plates-formes clientes hétérogènes minces ou épaisses (par exemple, téléphones mobiles, tablettes, ordinateurs portables et postes de travail).

- **Mise en commun des ressources :** Les ressources informatiques du fournisseur sont regroupées pour servir plusieurs consommateurs à l'aide d'un modèle à locataires multiples, avec différentes ressources physiques et

virtuelles attribuées et réaffectées dynamiquement selon la demande des consommateurs. Il y a un sentiment de localisation indépendance en ce que le client n'a généralement aucun contrôle ni aucune connaissance sur l'emplacement exact des ressources fournies mais peut être en mesure de spécifier l'emplacement à un niveau d'abstraction plus élevé (par exemple, pays, état ou centre de données). Des exemples de ressources incluent le stockage, traitement, mémoire et bande passante réseau.

– **Élasticité rapide :** Les capacités peuvent être provisionnées élastiquement et libérées, dans certains cas automatiquement, pour évoluer rapidement vers l'extérieur et vers l'intérieur en fonction de la demande. Pour le consommateur, les capacités disponibles pour l'approvisionnement semblent souvent être illimitées et peuvent être appropriées en toute quantité et à tout moment.

– **Service mesuré :** Les systèmes cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de mesure1 à un certain niveau d'abstraction approprié au type de service (par exemple, stockage, traitement, bande passante et comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillé, contrôlé et signalé, assurant la transparence pour le fournisseur et le consommateur du service utilisé.

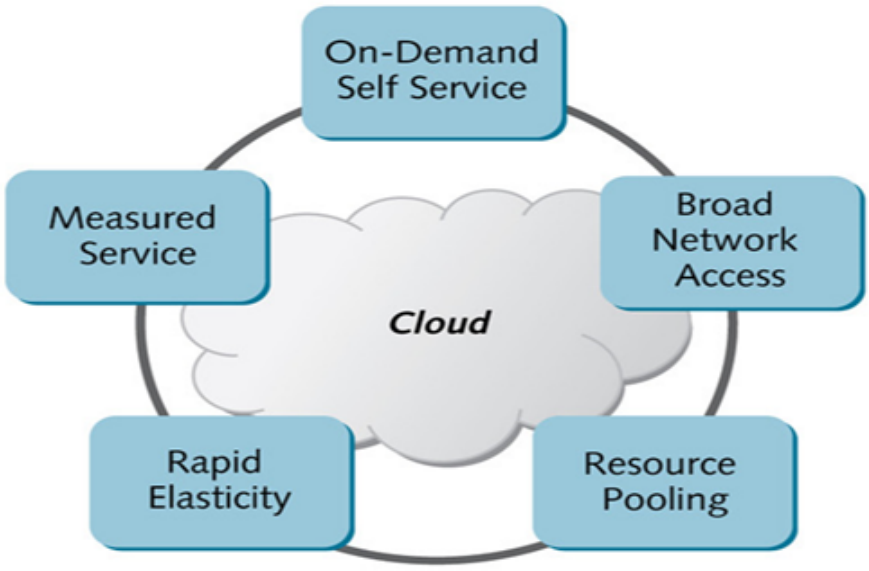

FIGURE 1.2 – les caractéristiques de cloud computing

1.3.3 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.Virtualisation :

La virtualisation est une technique qui permet le partage d'une instance physique d'une application ou d'une ressource entre plusieurs clients ou organisation.

L'utilisation principale de cette technologie est de fournir aux applications des versions standard à leurs utilisateurs du cloud[7].

L'objectif principal de la virtualisation est de cacher les caractéristiques physiques des ressources informatiques afin que les autres systèmes, les applications ou les utilisateurs finaux interagissent avec ces ressources[8].

La virtualisation constitue la base du cloud computing .les fournisseurs peuvent personnaliser la plate-forme pour répondre aux besoins des client,soit par des applications exposant en cours d'exécution au sein de machines virtuelles permettant ainsi aux client de créer des services avec leur propres applications.

En outre, le Cloud computing est non seulement basé sur la virtualisation de ressources, mais aussi sur la répartition intelligente des ressources pour la gestion des demandes concurrentes de ressources des client. la figure illustre une exploitation de la technologie de virtualisation dans les environnements de cloud computing.

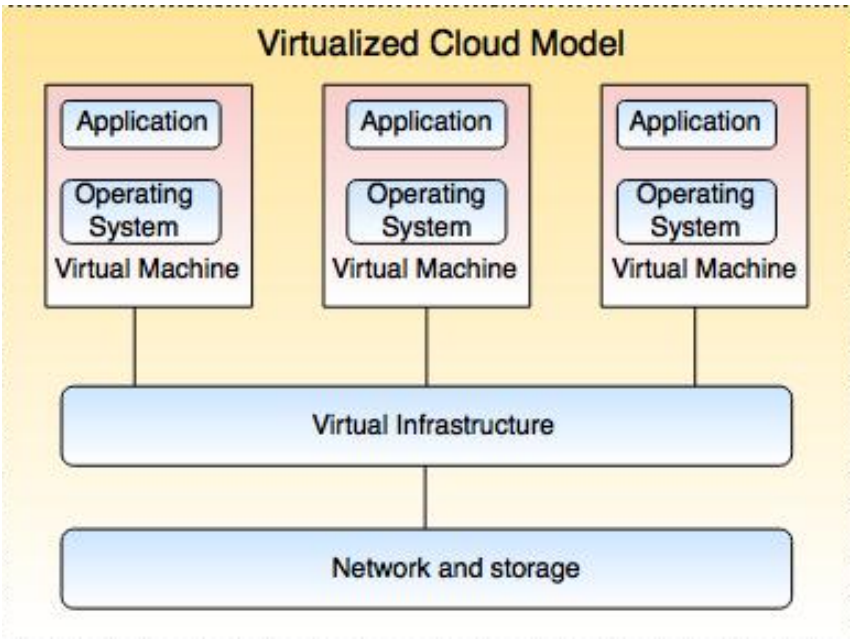

FIGURE 1.3 – virtulisation de cloud computing

2.Grid Computing

L'informatique en grille est la structure de l'informatique distribuée, dans laquelle un groupe de ressources informatiques provenant de divers emplacements sont connectés les uns aux autres pour obtenir un objectif commun.
Les ressources informatiques sont différentes et géographiquement dispersées. Les systèmes de grille sont conçus pour le partage des ressources grâce à une informatique en cluster distribuée et à grande échelle. Le calcul en grille divise les tâches composites en plus petits morceaux, qui sont distribués aux CPU et constitués dans la grille[7].

3.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 avec compteur. Tous les services informatiques gérés, Grid computing, cloud computing sont basés sur le concept de grid computing[7].
Avec l'approvisionnement des ressources à la demande et le paiement à l'usage, les fournisseurs de services peuvent maximiser l'utilisation des ressources et minimiser leurs coûts d'exploitation[9].

4.L'autonomie computing :

vise à construire des systèmes informatiques capables de s'auto-administrer en s'adaptant à des changements internes et externes sans intervention humaine. Le but de l'informatique autonome est de surmonter la complexité de la gestion des systèmes informatiques d'aujourd'hui. Bien que le cloud computing présente certaines caractéristiques autonomes, telles que l'approvisionnement automatique des ressources, son objectif est de diminuer le coût des ressources, plutôt que de réduire la complexité du système[9].

1.3.4 Modèles de déploiement :

Un modèle de déploiement cloud est une "configuration" de certains paramètres d'environnement cloud tels que la taille du stockage, l'accessibilité et la propriété.Ils existe 4 modèle de déploiement de cloud computing :

1. Le Cloud Public :

Dans ce modèle de déploiement, le fournisseur de la solution de Cloud est externe, il est propriétaire de son infrastructure et ses services sont accessibles à tout le monde (sous réserve de payer bien entendu) [3].

Cette forme d'utilisation des systèmes informatiques permet aux entreprises de se concentrer sur les processus métiers représentant le fondement de leurs activités en confiant la gestion de leurs systèmes informatiques aux fournisseurs distants. Plusieurs autres avantages sont également liés aux services publics : il s'agit de la mobilité à travers un accès omniprésent et rapide aux ressources, le partage de ressources notamment des machines de grandes performances accessibles à partir de simples clients légers, etc. Les grands acteurs de ces services sont Google, Amazon et Salesforce[5].

2. Le Cloud Privé :

Ce modèle de déploiement est interne aux entreprises ou organisations qui en sont les propriétaires. Ce modèle correspond aujourd'hui à une évolution des centres de données virtualisés et à l'émergence de l'IT as a Service (le système d'information et les équipes informatiques qui se transforment en centre de services pour le reste de l'entreprise). Dans cette optique d'IT as a Service, on voit parfaitement la pertinence de certaines caractéristiques du Cloud Computing (service mesurable et facturable aux différentes divisions de l'entreprise notamment) [3].

3. Le Cloud Communautaire :

Dans ce modèle de Cloud, les ressources, services et la propriété sont partagées à l'échelle d'une communauté (ex : à l'échelle d'un état, d'une ville, d'une académie, d'un GIE, etc.) [3].
Dans un cloud communautaire, l'infrastructure est déployée pour un usage exclusif par un groupe d'entreprises ou d'organisations partageant les mêmes intérêts. Dans une telle architecture, l'administration du système peut être effectuée par l'une ou plusieurs des organisations partageant les ressources du cloud[5].

4. Le Cloud Hybride :

Ce modèle est une combinaison de 2 ou 3 des modèles décrits ci-dessus. Le futur devrait confirmer l'émergence de ce type de Cloud avec une combinaison de Cloud privé et public[3].

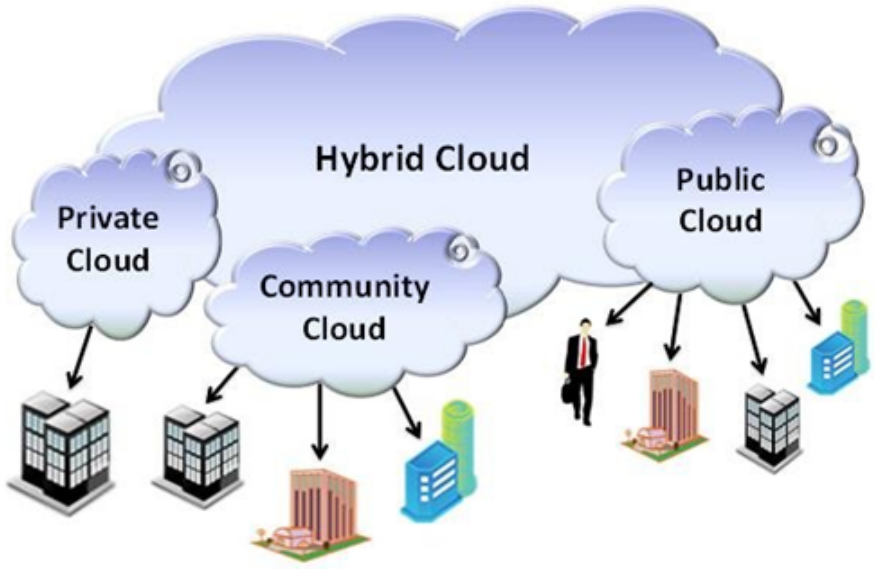

FIGURE 1.4 – Les modèles de déploiement dans le cloud computing

1.3.5 Les Services de Cloud Computing :

Les offres de Cloud computing se décomposent en trois familles de services : IaaS, PaaS et SaaS [2] :

1. Infrastructure as a service (IaaS) :

La catégorie la plus basique des services de cloud computing. Avec l'IaaS, vous louez une infrastructure informatique (serveurs, machines virtuelles, stockage, réseaux, systèmes d'exploitation) auprès d'un fournisseur de services cloud, avec un paiement à l'utilisation.

2. Platform as a service (PaaS) :

L'expression plateforme en tant que service (PaaS, Platform-as-a-Service) qualifie les services de cloud computing qui offrent un environnement à la demande pour développer, tester, fournir et gérer des applications logicielles. PaaS est conçu pour permettre aux développeurs de créer rapidement des applications web ou mobiles sans avoir à se préoccuper de la configuration ou de la gestion de l'infrastructure de serveurs, de stockage, de réseau et de bases de données nécessaire au développement.

3. Software as a service (SaaS) :

Le logiciel en tant que service (SaaS, Software-as-a-Service) est une méthode de diffusion d'applications logicielles via Internet, à la demande et en général sur abonnement. Avec le SaaS, les fournisseurs de services cloud hébergent et gèrent les applications logicielles et l'infrastructure sous-jacente, et gèrent la maintenance, par exemple la mise à niveau des logiciels et l'application des correctifs de sécurité. Les utilisateurs se connectent à l'application via Internet, en général par l'intermédiaire d'un navigateur web sur leur téléphone, leur tablette ou leur PC.

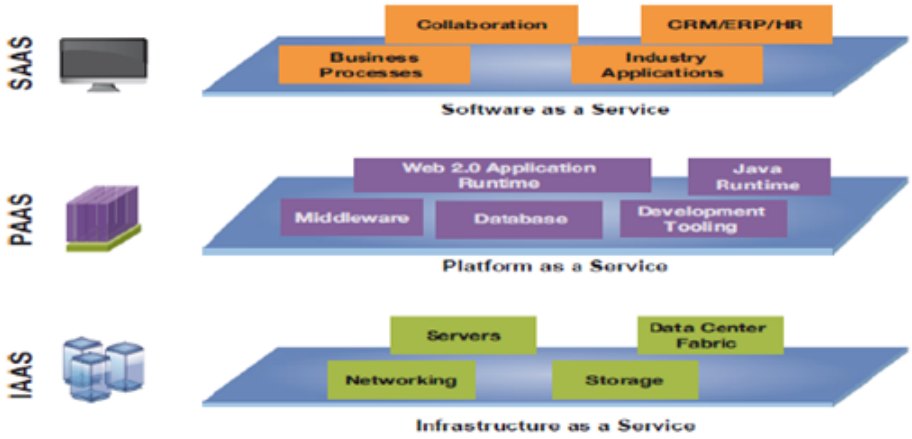

FIGURE 1.5 – Les service de cloud computing

1.3.6 Acteurs du cloud computing :

Selon l'architecture de référence du cloud computing décrite par le NIST (Fang, 2011) [4], on distingue cinq acteurs principaux comme le montre la figure :

- **Consommateur (cloud Consumer) :** Une personne ou une organisation qui utilise un service fourni par un fournisseur. Dans cette thèse, nous utilisons aussi le terme utilisateur pour désigner un consommateur.

- **Fournisseur (cloud Provider) :** Une personne, une organisation ou une entité chargée de rendre un service à la disposition des parties intéressées.

- **Courtier (cloud broker) :** Une entité qui gère l'usage, la performance et l'approvisionnement de services, et qui négocie les relations entre les fournisseurs et les consommateurs.

- **Auditor (cloud auditor) :** Une entité qui peut procéder à une évaluation indépendante des services de cloud telle que la performance et la sécurité du cloud.

- **Carrier (cloud carrier) :**Un intermédiaire qui offre la connectivité et le transport des services des fournisseurs aux consommateurs.

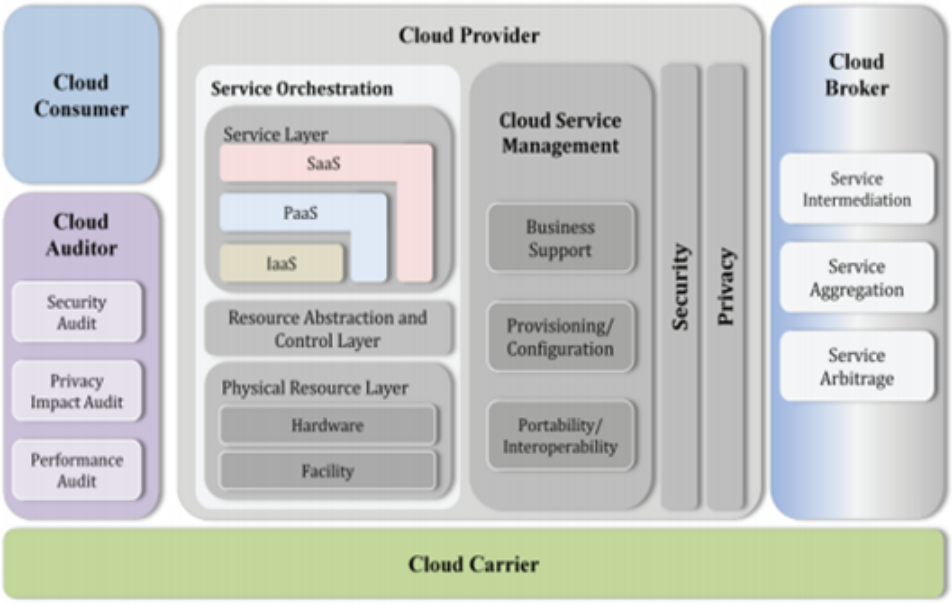

FIGURE 1.6 – les acteurs de cloud computing

1.3.7 Challenges de recherche en environnement de cloud computing :

Même si certaines des caractéristiques essentielles du cloud computing ont été réalisées par des efforts commerciaux et universitaires, de nombreux problèmes existants n'ont pas été pleinement pris en compte, et d'autres nouveaux défis continuent d'émerger [6], Dans cette section, nous résumons quelques enjeux de recherche dans le cloud computing.

Migration des données entre les environnements non standards :

La plupart des fournisseurs de services de cloud computing utilisent des applications cloud propriétaires. Ces applications ne sont pas interopérables, ce qui rend très difficile pour les utilisateurs de faire passer leurs données à un autre fournisseur de cloud ou de revenir à leur machines.

Sécurité de données et confidentialité :

Dans le cloud computing, les données doivent être transférées entre les dispositifs de l'utilisateur et les Datacenter des fournisseurs de services de cloud computing, ce qui les rendra cible facile pour les pirates. La sécurité des données et la confidentialité doivent être garanties, que ce soit sur le réseau ou encore dans les Datacenter de cloud où elles seront stockées.

Migration de machines virtuelles :

La virtualisation peut offrir des avantages importants dans le cloud computing en permettant la migration de machine virtuelle pour équilibrer la charge de travail entre les Datacenter. Elle permet un approvisionnement robuste et très réactif dans les Datacenter. Les principaux avantages de la migration de MV est d'éviter les points chauds (hot spots) ; Toutefois, cela n'est pas simple à réaliser. Actuellement, la détection de points chauds et l'initiation d'une migration manque de souplesse pour répondre aux changements brusques de charge de travail.

Consolidation de serveurs :

La consolidation de serveurs est une approche efficace pour maximiser l'utilisation des ressources, tout en minimisant la consommation d'énergie dans un environnement de cloud computing. La technologie de migration de MV est souvent utilisée pour consolider les machines virtuelles se trouvant sur plusieurs serveurs sous-utilisés sur un seul serveur, de sorte à mettre ces derniers en mode d'économie d'énergie. Le problème de la consolidation optimale des serveurs est un problème d'optimisation NP-complet. Cependant, les activités de consolidation de serveurs ne devraient pas affecter les performances de l'application. Il est connu que l'utilisation des ressources de machines virtuelles individuelles peut varier au cours du temps [22].

Ordonnancement :
L'ordonnancement est un enjeu important qui influence considérablement les performances de l'environnement de cloud computing.
Le processus d'ordonnancement est appliqué sur les différents niveaux selon la nature de service de cloud : (1) orienté-marché (l'ordonnancement au niveau de service et au niveau de tâche) ; (2) non orienté-marché (l'ordonnancement au niveau de MVs) [23].
L'ordonnancement au niveau de service est statique et concerne une partie de la couche de gestion des ressources. Le principale critère à optimiser au ce niveau c'est le profil. L'ordonnancement au niveau de tâche d'ordonnancement est dyna-

mique, adapté au changement de cloud.
Son objectif est d'optimiser l'assignement selon les contraintes de QoS de chaque tâche et de chaque client, tout en minimisant le temp d'exécution. L'ordonnanceur au niveau de tâche est dédié à un seul data center (cloud), et il est incapable de gérer les ressource d'un autre cloud d'un autre fournisseur. Le processus de correspondance entre la tâche et la MV se fait par un courtier. L'ordonnancement au niveau machine virtuelle c'est la couche d'ordonnancement la plus basse, utilisée pour fournir l'ensemble des MVs demandés par la tâche dans l'ordonnancement au niveau de tâches. Le but de ce niveau est de trouver un meilleur ordonnancement des machines virtuelles sur les hôtes qui composent les data center (cloud) [23].

1.3.8 Avantages du Cloud Computing :

Avantages du Cloud Computing L'intérêt des Cloud Computing est évident dans le sens ou au lieu d'acheter cher des serveurs et des logiciels, qui ne sont pas utilisés à 100%, les entreprises les louent et ne paient que pour l'usage qu'elles en font, En plus de cela,Cloud offre plusieurs avantages aux utilisateurs[6] :

- Réduction du cout du a l'efficacité opérationnelle et déploiement plus rapide de nouveaux services aux entreprises.
- Flexibilité de la répartition des couts pour les clients.
- Un démarrage rapide :Le cloud computing permet de tester le business plan rapidement, à coûts réduits et avec facilité.
- L'agilité pour l'entreprise :Résolution des problèmes de gestion informatique simplement sans avoir à vous engager à long terme.
- Un développement plus rapide des produits :Réduisons le temps de recherche pour les développeurs sur le paramétrage des applications.
- Pas de dépenses de capital :Plus besoin des locaux pour élargir vos infrastructures informatiques.

1.3.9 Inconvénients du Cloud Computing :

En dépit de ses nombreux avantages, le cloud computing présente aussi quelques inconvénients[6] :
La bande passante peut faire exploser votre budget : La bande passante qui serait nécessaire pour mettre cela dans le Cloud est gigantesque, et les couts seraient tellement importants qu'il est plus avantageux d'acheter le stockage soi-même plutôt que de payer quelqu'un d'autre pour s'en charger.
Les performances des applications peuvent être amoindries : Un Cloud public n'améliorera définitivement pas les performances des applications.

La fiabilité du Cloud : Un grand risque lorsqu'on met une application qui donne des avantages compétitifs ou qui contient des informations clients dans le Cloud.
Taille de l'entreprise : Si votre entreprise est grande alors vos ressources sont grandes, ce qui inclut une grande consommation du cloud. Vous trouverez peut-être plus d'intérêt à mettre au point votre propre Cloud plutôt que d'en utiliser un externalisé. Les gains sont bien plus importants quand on passe d'une petite consommation de ressources à une consommation plus importante.

1.4 Conclusion :

A niveau de ce chapitre nous avons fourni une base théorique sur le Cloud Computing, en présentant ses types, ses services (IaaS, PaaS, SaaS), ainsi nous avons présenté quelques enjeux de recherche dans le cloud computing. Enfin,ses avantages et inconvénients.

Dans le chapitre suivant nous allons présenter L'ordonnancement de workflow dans le cloud.

Chapitre 2

L'ordonnancement des workflows scientifique dans le cloud computing

2.1 Introduction

Pour rationaliser l'utilisation des ressources informatiques, de plus en plus de sociétés ont recours à l'hébergement et l'utilisation distante de leurs moyens de calcul ou de stockage. Cette tendance, liée à la fiabilité des réseaux et à la généralisation des techniques de virtualisation, conduit à disposer de ressources d'exécution ou de stockage louées chez des fournisseurs. Dans ce contexte, on parle d'informatique dans les nuages, " cloud computing ".

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 clients, via un modèle de paiement à l'usage. Généralement, il peut fournir trois types de services : IaaS (Infrastructure as a Service), PaaS (Platform as a Service), et le SaaS (Software as a Service). Ces services sont offerts avec différents niveaux de qualités de services (QoS) afin de répondre aux besoins de différents groupes de clients. Bien que de nombreux services de cloud computing ont une fonctionnalité similaire (par exemple des services de calculs, services de stockages, services de réseaux, etc), ils diffèrent les uns des autres par des qualités 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 différent contrats : SLA (Service Level Agreements).
L'exécution d'une application (workflow) dans le cloud peut ainsi être associée à un coût puisque la facturation des fournisseurs de ressources se fait à l'heure de calcul. Les tarifs sont alors dépendants des caractéristiques des ressources utilisées et des contraintes de QoS exigées dans le SLA. Pour respecter ces contraintes, il est alors nécessaire de prévoir les temps d'exécution et de réaliser les demandes

de ressources au plus juste.

2.2 L'ordonnancement de tâches

La performance des applications est reliée à la bonne gestion de l'utilisation des ressources, donc il est nécessaire de proposés des algorithmes et des outils efficaces pour gérer une utilisation efficace des ressources. Plusieurs algorithmes d'ordonnancements ont été proposés dans la plate- forme de type cloud. Le travail de Bessai.K dans [15] a été proposé une classification, des algorithmes existants, en trois catégories selon les types de tâches constituant l'application : tâches indépendantes, tâches dépendantes (DAG) et les DAGs s'exécutant en parallèle.

2.2.1 l'ordonnancement de tâches indépendantes

La plupart des études ont été proposé dans leurs études des tâches indépendantes par rapport aux deux autres types d'applications, cependant très rare des travaux utilisent des plates-formes hétérogènes. Les heuristiques d'ordonnancement dynamique des tâches indépendantes et séquentielles proposés dans [16] sont classées en deux classes :

2.2.1.1 Des heuristiques d'ordonnancement en ligne

Les tâches qui arrivent dans le système sont ordonnancées et affectées à des ressources dès leurs arrivées.

2.2.1.2 Des heuristiques d'ordonnancement par lot (batch sheduling)

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 [15].

2.2.2 Ordonnancement de 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 détaches séquentielles a été largement étudié et plusieurs algorithmes heuristiques ont été proposés. Ces algorithmes peuvent être classés en trois catégories [15] :

2.2.2.1 Les algorithmes d'ordonnancement de listes

Les algorithmes de cette classe commencent par le calcul de la priorité des tâches constituant l'application, ensuite ces algorithmes consiste 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 heuristiques ont l'avantage d'être peu complexes par rapport aux autres types d'heuristiques et produisent des résultats similaires.

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 de 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 [17] classe les algorithmes d'ordonnancement de duplication de tâches en deux catégories : l'ordonnancement avec duplication partielle et l'ordonnancement avec double-duplication.

2.2.2.4 Les algorithmes d'ordonnancement méta-heuristiques

Les méta-heuristiques pour l'ordonnancement des taches sont des méthodes plus efficaces et largement utilisées et pour l'ordonnancement des graphes de tâches séquentielles. Les heuristiques basés sur les algorithmes génétiques produit des bons résultats par rapport aux autres heuristiques (Recherche Tabou [18], Recuit Simulé [19]), mais restent plus couteux en termes de complexité et temps de calcul. On trouve dans cette classification les travaux de recherche de Yassa.S [20] [21] [22] qui utilisent l'algorithme NSGAII et l'algorithme génétique

pour la résolution du problème d'ordonnancement des workflows dans le cloud computing.

2.3 Algorithmes d'ordonnancement de tâches

Les principaux exemples d'algorithmes de planification sont l'algorithme de planification du FCFS, l'algorithme PSO, l'algorithme de Max-min, l'algorithme de Min-min, l'algorithme de Round Robin, l'algorithme de SJF, l'algorithme génétique comme le montre le suivant :

1- Algorithme d'ordonnancement FCFS : est considéré comme une méthode simple dans la planification des algorithmes, où les processus sont ordonnés en fonction de temps d'arrivée et de la soumission à la CPU [23].

2- Algorithme d'ordonnancement PSO : est un algorithme d'optimisation stochastique basé sur la population, motivé par le comportement collectif intelligent de certains animaux tels que les troupeaux d'oiseaux ou les bancs de poissons.[48]

3- Algorithme d'ordonnancement Max-min : C'est l'inverse de SJF en sélectionnant les tâches les plus importantes à exécuter en premier [23].

4- 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 [24].

5- Algorithme d'ordonnancement Round Robin : un autre type de planification prioritaire. Ce type est un algorithme de planification le plus simple, le plus équitable et le plus utilisé. Tous les processus s'exécutent dans une file d'attente circulaire avec la plus petite unité de temps appelée tranches de temps ou quantique [23].

6- Algorithme d'ordonnancement SJF : Shortest job first ou shortest job next (SJN), est une politique de planification qui sélectionne le processus en attente avec le plus petit temps d'exécution à exécuter ensuite. SJN est un algorithme non préemptif. [25].

7-Algorithme génétique : Pour résoudre ces problèmes, nous utilisons des heuristiques afin de trouver la solution optimale, ou à défaut, la moins mauvaise,

pour le problème. L'idée principale des heuristiques est d'explorer l'espace des solutions en essayant de converger vers la meilleure solution. Cependant, il est important d'éviter une convergence prématurée de l'algorithme vers un extremum, ou optimum, local. Un extremum local est la meilleure solution dans une zone restreinte, en opposition à l'extremum global, qui est la meilleure solution dans l'ensemble [26].

2.3.1 Algorithmes d'ordonnancement des tâches dans le cloud computing

La planification des tâches dans le cloud computing peut définir un ordre des tâches, où se trouve l'équilibre entre l'amélioration de la qualité de service et en même temps maintenir l'efficacité et l'équité entre les tâches en choisissant la meilleure ressource appropriée disponible pour l'exécution des tâches ou pour allouer l'ordinateur les machines aux tâches de manière à minimiser le plus possible le temps d'exécution [27][28].

Les principaux objectifs de l'algorithme de planification des tâches dans le cloud computing sont la réduction du temps de réponse et l'amélioration de l'utilisation des ressources [29].

Un rôle très important a joué dans la façon de répondre aux exigences des utilisateurs cloud de QoS et a utilisé les ressources du cloud de manière efficace, ce qui est rentable.

L'optimisation des ressources d'utilisation est représentée comme la principale raison de la planification en améliorant la tâche terminée au moindre coût et en même temps, où l'utilisateur est propriétaire de la même qualité de service [30].

Les processus de planification du cloud sont divisés en trois étapes, à savoir[31] :

- **La ressource est en train de découvrir et de filtrer :** les ressources présentées dans le système réseau et les informations d'état collectées à ce sujet par le Data Center Broker.
- **Sélection des ressources :** la ressource sélectionnée cible en fonction de certaines exigences de tâche et de ressource.
- **Affectation des tâches :** la tâche est allouée pour sélectionner la ressource.

2.3.2 Problèmes avec la planification de tâches :

L'ordonnancement classé comme [28] :

- Niveau utilisateur : le problème est soulevé par les fournisseurs et les clients en raison de la fourniture de services.
- Niveau système : le problème est soulevé par la gestion des ressources et le centre de données.

2.3.2.1 Taxonomie des algorithmes d'ordonnancement dans le Cloud :

- Les tâches peuvent être classées comme [32] :
  - **Indépendant :** ne nécessite aucune communication entre les tâches.
  - **Dépendant :** les tâches ont un type d'ordre à suivre pendant le processus de planification.

2.3.2.2 Technique de planification :

Une technique de planification est le mécanisme que le planificateur choisit pour planifier des tâches d'une application en prenant en considération le coût des ressources utilisées. Dans la littérature, deux types de techniques ont été adoptés dans les approches d'optimisation des coûts, à savoir : (i) technique statique et (ii) technique dynamique [32].

  - **Ordonnancement statique :** le calendrier des tâches connaît l'environnement et les estimations de la tâche temps d'exécution / d'exécution avec a les informations sur la structure complète des tâches et cartographie des ressources avant exécution.
  - **Ordonnancement dynamique :**dépend de l'état actuel du système, des machines informatiques et les tâches soumises à l'environnement cloud pour prendre des décisions de planification.

- Stratégie statique appliquée de deux manières [32] :
  - **Les algorithmes basés sur l'heuristique :** en faisant des hypothèses réalistes sur les connaissances a priori concernant les caractéristiques de chargement du processus et du système, mais ne peut pas donner une réponse optimale. Il ne nécessite que le montant le plus raisonnable de coût et autre système ressources pour remplir leur fonction.
  - **Les algorithmes basés sur la recherche aléatoire guidée :** GA est un exemple de ce type, qui est recherché pour une solution quasi optimale dans les espaces.

L'aléatoire guidé ressemble étroitement au phénomène existant dans la nature et également appelé " heuristique de la nature ". Il les choisit au hasard et les guide à travers espace de problème.

- Stratégie dynamique appliquée de deux manières :
  - **Mode immédiat / en ligne :** arrivée du travail dès son arrivée sans attente du prochain intervalle de temps sur les ressources disponibles à ce moment.
  - **Mode batch / hors ligne :** les magasins du planificateur arrivent des tâches (dans une file d'attente par exemple) et résolvent le processus d'exécution sur des intervalles de temps successifs, de sorte qu'il est préférable de mapper une tâche pour des ressources appropriées en fonction de ses caractéristiques.

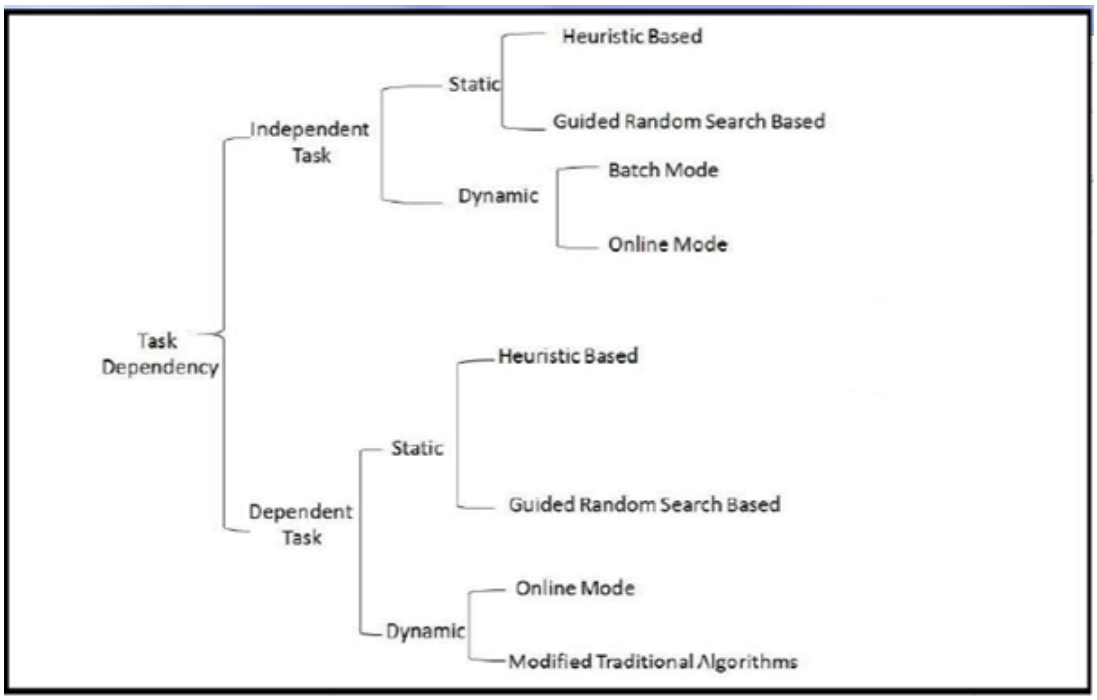

FIGURE 2.1 – Taxonomie des algorithmes de planification dans le cloud en fonction de la dépendance des tâches [32]

2.4 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 [33], 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 [34]. Néanmoins, d'autres objectifs sont également pris en compte.

2.4.1 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.

2.4.2 Coût :

Le coût est devenu un objectif important dans la recherche sur la planification des applications dans le Cloud. Le coût total engendré par l'exécution d'une application distribuée peut comprendre de nombreux éléments de coût, tels que le coût de calcul et le coût de transfert de données. Un budget peut être défini comme une contrainte difficile [35]. Cependant, il est plus courant que le coût soit spécifié en tant que contrainte à minimiser tout en respectant un délai déterminé. Dans certains cas, le temps et le coût ne sont pas spécifiés [36] ; il est donc possible d'obtenir de nombreuses solutions avec des compromis entre les deux contraintes. Des techniques de planification tenant compte des coûts ont également été introduites pour les grilles de calcul avant l'avènement du Cloud Computing [37]. Cependant, le modèle de coût dans le Cloud diffère par rapport à celui des grilles, mais pas entièrement.

2.4.3 Fiabilité :

Outre 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. L'échec de l'exécution de la tâche est généralement traité en redémarrant et en répliquant la tâche. Cependant, les deux

mécanismes peuvent respectivement causer la perte de temps [38].

2.4.4 Consommation d'énergie :

Avec la préoccupation environnementale croissante, la réduction de la consommation d'énergie ou de l'empreinte carbone a commencé à attirer l'attention [39]. Bien que ce problème ne soit pas propre au Cloud Computing, il a également attiré des chercheurs dans ce domaine. Par exemple, cet objectif est optimisé comme un des critères de planification multi objectifs [36]. Luo et Zhou [40] ont récemment abordé cette question en introduisant un modèle de consommation d'énergie. Le modèle est utilisé pour estimer l'énergie consommée par les services en Cloud afin que l'algorithme puisse sélectionner une stratégie de planifications satisfaisant les contraintes de temps et de coût mais avec une consommation minimale d'énergie.

En raison de la virtualisation et de l'abstraction des ressources physiques et du fait que cet objectif est davantage lié aux fournisseurs de Cloud, les planificateurs pourraient ne pas être en mesure de traiter directement ce problème, sauf si les informations relatives à la consommation d'énergie au moment de l'exécution de chaque instance de machine virtuelle sont mises à la disposition des planificateurs par les fournisseurs de services d'informatique en Cloud [40].

2.4.5 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 mis en IJuvre sur des instances et des réseaux de 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 [35]. Cette exigence impose généralement une contrainte forte à la sélection des ressources dans le Cloud.

La prise en compte de la sécurité et de la confidentialité peut être au niveau de l'application distribuée ou au niveau de la tâche constituant l'application. En supposant au niveau de l'application, l'intégralité ne peut être exécutée que sur les ressources privées. D'autre part, en supposant au niveau de la tâche, chaque tâche peut être planifiée pour une ressource privée ou une ressource de Cloud

public en fonction des exigences de sécurité et de confidentialité [41].

Avec la prise en compte de la sécurité, la définition de tâche introduite précédemment devrait être élargie. Toutefois, cela dépend de la manière dont la sécurité est considérée dans chaque application. Par exemple, seule une liste d'images de machine virtuelle approuvées et sécurisées peut être spécifiée pour chaque tâche. En outre, un indicateur de sécurité peut être inclus dans la définition de la tâche pour limiter l'exécution sur des clouds privés et sécurisés.

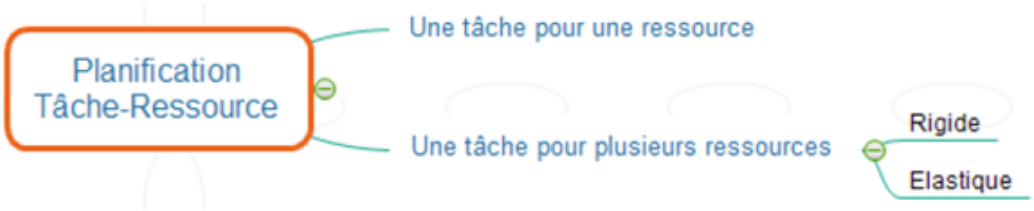

FIGURE 2.2 – Taxonomie des tâches

2.5 Taxonomie des tâches :

Dans cette section, nous discutons de l'impact du Cloud sur la classification du mappage tâche-ressource comme illustré dans la Figure 2.2. Cet aspect de classification fait référence au nombre de ressources, ou d'instances de machines virtuelles, requises par une tâche constituant l'application répartie.

2.5.1 Une tâche pour une machine :

Lors de la planification d'une application distribuée dans le domaine des grilles de calcul, une tâche est généralement supposée occuper un nIJud de calcul pour son exécution [42]. Mais dans le Cloud, de nombreuses techniques de planification supposent implicitement que chaque tâche est exécutée par une seule instance de machines virtuelles [43].

2.5.2 Une tâche pour plusieurs machines :

Avec les modèles de programmation MPI et MapReduce, il est possible qu'une tâche d'une application nécessite plusieurs noeuds ou instances de machine virtuelle pour son exécution en parallèle [44]. Cette catégorie peut être divisée en

deux catégories : " rigide ", où le nombre d'instances requises est fixe pendant l'exécution, et " élastique ", où le nombre peut être modifié pendant l'exécution. Cette considération n'est toujours pas courante dans la documentation sur la planification dans le Cloud. En raison de la complexité de la planification provoquée par cette hypothèse, seuls quelques algorithmes tels que l'algorithme " PBTS " [45] et son prédécesseur, l'algorithme " BTS " [46], abordent ce problème. Dans ce cas, la définition de tâche peut être affinée en tant que t = (id ;MI ;mr), où mr spécifie le nombre d'instances de machine virtuelle requises par t [47].

Avec cette considération, une technique de planification doit estimer le nombre de machines virtuelles nécessaires dans chaque intervalle de temps [45] [46] afin de déterminer le coût. La planification peut être plus compliquée. Par exemple, si des tâches avec un nombre différent d'instances de machine virtuelle requises doivent être planifiées dans le même intervalle, un planificateur doit d'abord déterminer le nombre d'instances à instancier. Si toutes les instances requises sont instanciées à la fois, le coût serait maximal. Afin de minimiser les coûts, le planificateur peut choisir d'instancier moins d'instances.

Dans ce cas, il doit hiérarchiser correctement les tâches afin d'éviter, par exemple, qu'une tâche nécessitant plus d'instances ne soit trop retardée par des tâches nécessitant moins d'instances.

2.6 Ordonnancement d'applications concurrentes :

Les applications qui exécutent en parallèle (applications concurrentes) peuvent tomber dans un problème de famine. Ce problème produit par l'attente d'un certain temps d'application et n'a jamais accès aux ressources disponibles (indéfiniment retardées). Le makespan et le débit sont les critères les plus considérés par les applications qui partagent les mêmes ressources.

2.7 Conclusion :

Dans ce chapitre, nous avons présenté d'abord l'ordonnancement des taches puis on a vu les algorithmes d'ordonnancement, les algorithmes d'ordonnancement dans l'environment cloud computing, le problème avec la planification de taches, Taxonomie des algorithmes d'ordonnancement dans le Cloud, techniques de planification, Critères de planification dans le Cloud Computing, Taxonomie des tâches et Ordonnancement d'applications concurrentes.

Le chapitre suivant présente nos contributions pour l'optimisation de workflow dans l'environnement cloud computing.

Chapitre 3

Approche proposée

3.1 introduction :

La plate-forme cloud computing est caractérisée par l'utilisation d'un grand pool de ressources informatiques accessibles à la demande à travers Internet. Il offre une variété de ressources, tels que le réseau, le stockage, aux utilisateurs adoptées par IaaS, PaaS, SaaS ... etc

Dans les Workflows scientifiques intensifs en données ou calcul, les tâches nécessitent l'exécution de plusieurs jeux de données. Cependant, lorsque ces tâches sont exécutées dans différents centres de données, le transfert de données deviendrait inévitable. Pour résoudre ce problème, nous proposons une adaptation d'une technique évolutionnaire (les algorithmes génétiques) dans l'objectif de trouver une meilleure affectation des tâches et des données à l'ensemble des instances des machines virtuelles (MVs) pour une meilleure exécution des applications distribuées.

3.2 Représentation de workflows scientifique :

Un workflow scientifique est représenté par un graphe orienté et sans cycle (acyclique) noté G = (T ;E), avec :

- T = t1, .., tn est l'ensemble fini des tâches qui le compose et qui sont exécutées de façon indépendantes.
- E est l'ensemble de ses arcs représentant les contraintes de données entre les tâches T.

La Figure 3.1 représente un exemple d'une couche de workflow scientifique constituée de sept tâches et sept données notées respectivement t1,t2,Ĕ.,t7 et d1,d2,..,d7.

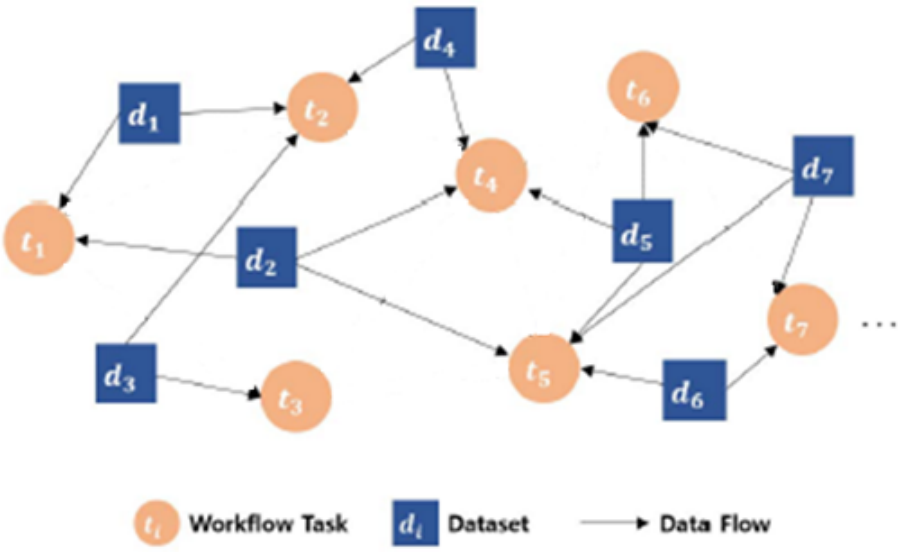

FIGURE 3.1 – Exemple d'une instance d'une seule couche de workflow scientifique

Pour exécuter une telle application, nous disposons d'un ensemble de machines virtuelles, Chaque VM possède ses propres caractéristiques telles que le débit, le temps d'exécution, etc...

3.3 Description de l'approche proposé :

Pour trouver un ordonnancement optimal, un processus évolutionnaire basé sur l'algorithme génétique a été appliqué [9]. L'algorithme génétique (GA) est une métaheuristique motivé par l'évolution génétique avec des caractéristiques importantes pour l'optimisation combinatoire. Il s'agit d'une technique robuste pour résoudre des problèmes complexes en ingénierie et en science en raison de sa capacité à détecter un optimum global dans l'espace de recherche complet. Contrairement aux solutions heuristiques actuelles, GA ne construit pas une seule solution [14].

L'Algorithme Génétique **(GA)** démarre avec un groupe de solutions possibles. Chaque chromosome est une chaîne de gènes codant une solution spécifique. La nature particulière d'un problème d'optimisation définit les caractéristiques des chromosomes et des gènes. Grâce au processus génétique, GA sélectionne les chromosomes les plus aptes, en les combinant pour produire une solution finale solide.

La première phase de production d'une nouvelle population est l'opérateur de sélection. Son objectif est de sélectionner des chromosomes pour produire la prochaine population. Une technique de sélection fréquemment utilisée est la roulette qui attribue à chaque chromosome une partie de la roulette en fonction de sa valeur de fitness ; par conséquent, les chromosomes avec des valeurs plus élevées se voient allouer plus d'emplacements avec plus de chances d'être sélectionnés pour la prochaine population. Ensuite, les opérateurs génétiques combinent les chromosomes pour, espérons-le, produire des chromosomes avec des valeurs de fitness plus élevées [14] :

L'opérateur **croisement (Crossover)** divise et combine les gènes entre deux chromosomes sélectionnés selon une probabilité prédéfinie.
L'opérateur **mutation** sélectionne au hasard les gènes d'un chromosome et modifie leurs valeurs en fonction d'une autre probabilité prédéfinie. De plus, les chromosomes les plus aptes sont directement copiés dans la population suivante.

Enfin, GA se termine lorsqu'il répond aux critères sélectionnés. Les critères les plus utilisés sont le temps d'exécution total, le nombre d'itérations, la valeur de fitness et l'amélioration minimale conditionnelle. La fonction fitness de GA évalue la qualité de chaque chromosome [14].

Les différentes étapes de l'algorithme GA sont illustrées dans l'organigramme de la Figure 3.2 L'algorithme itère le processus jusqu'à atteindre un nombre maximum d'itérations ou jusqu'à ce que la valeur globale de la fitness ne s'améliore plus[4].

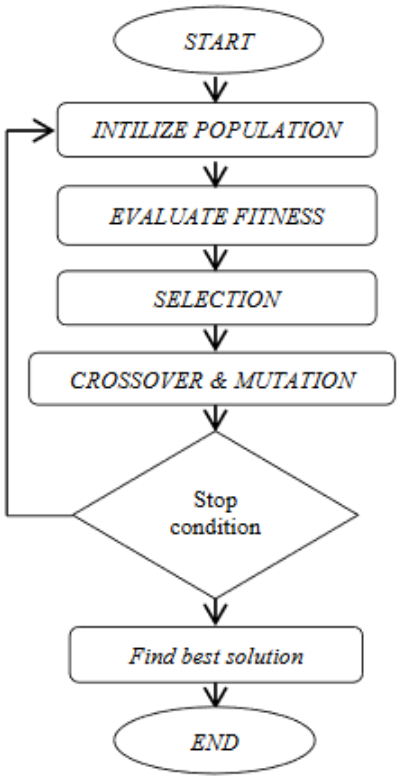

FIGURE 3.2 – organigramme générale du processus génétique

3.3.1 Population Initiale :

Le succès de chaque contribution dans la recherche repose sur l'approche de la modélisation des problèmes et sur la manière d'adapter la méta-heuristique au problème d'ordonnancement, qui codent des solutions de planification à un problème d'optimisation, évolue vers des meilleures solutions. La population initiale représente un ensemble de chromosome qu'on va le décrire dans la section suivante.

3.3.2 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 3.3 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

3.4 Opérateurs de l'algorithme génétique :

Les opérateurs génétiques représentent le cIJur d'un algorithme génétique. L'utilisation de tels opérateurs permet de générer de nouvelles solutions à partir de celles déjà existantes [4].

3.4.1 Opérateur de sélection :

Plusieurs algorithmes génétiques 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. Dans notre algorithme, nous procédons en deux étapes pour sélectionner les solutions parentes [4] :

Chromosome label	Chromosome string	Fit	Fit Radio %
X1	31231233111	20	2.60
X2	13223312121	267	34,72
X3	33212123321	112	14.56
X4	12213321231	205	26.65
X5	33212133212	87	11.31
X6	22311233123	78	10.14

TABLE 3.1 – calcul de la roulette

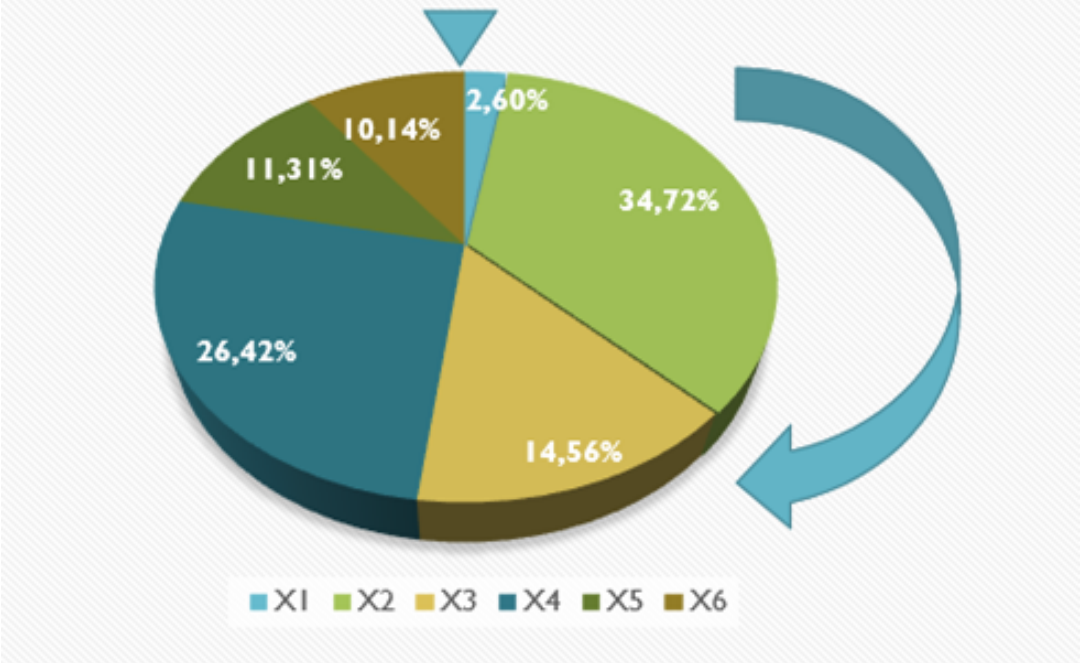

FIGURE 3.4 – Modélisation de la roulette selon le tableau Tab 3.1

3.4.2 Opérateur croisement(crossover) :

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 en un point (One-point Crossover), le croisement à deux points (Two-point Crossover), et autres [4]. Dans notre travail, nous avons implémenté seulement croisements en un point (One-point 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. La figure 3.5. illustre le principe du Cut Crossover pour l'ordonnancement d'un workflow composé de 5 tâches et 6 données sur cloud computing composée de 3 VM.

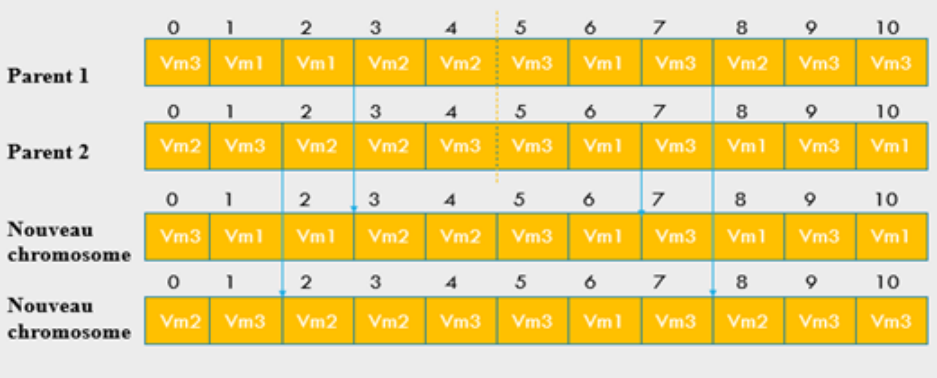

FIGURE 3.5 – Application de l'opérateur crossover one point sur la même application.

3.4.3 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 Rééquilibrer (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.

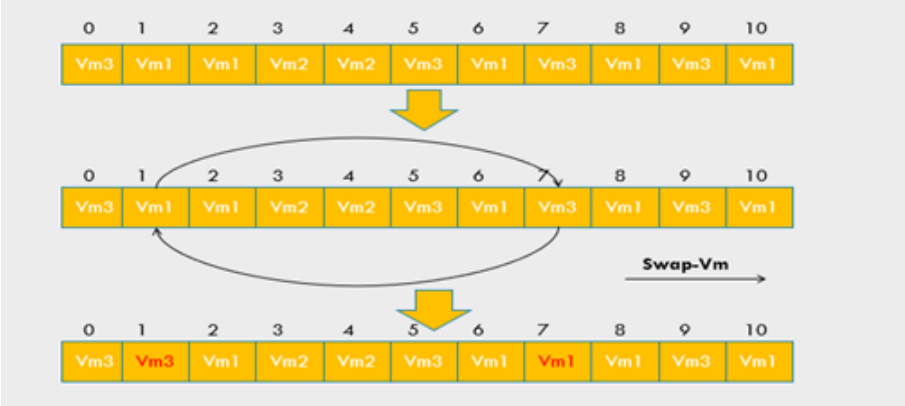

FIGURE 3.6 – Application de l'opérateur mutation de swap sur la même application

3.5 Fonction de fitness :

Les opérateurs génétiques représentent le cIJur d'un algorithme génétique. L'utilisation de tels opérateurs permet de générer de nouvelles solutions à partir de celles déjà existantes[4].

Nous avons introduit une fonction objective nous permettant de connaître le temps de réponse de l'application scientifique.

$$MinimiserF = \sum_{i=1}^{n} (Texeci + Temps_Transfert_file + T_release)$$

Cette section expose des détails sur les paramètres d'optimisation des tâches., comme indiqué à la Figure3.7.

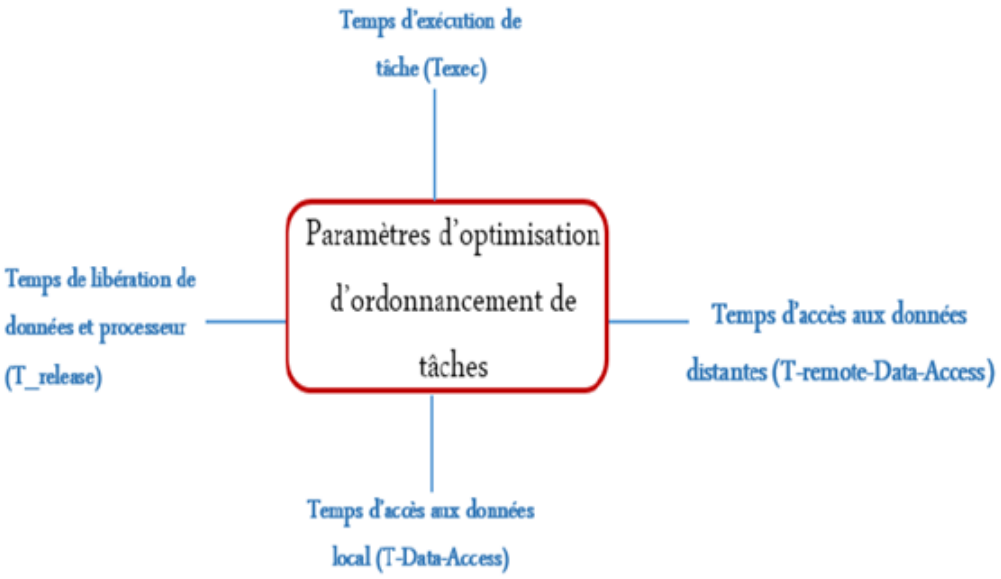

FIGURE 3.7 – Paramètres d'optimisation

3.5.1 Paramètres d'optimisation :

3.5.1.1 Estimation du Temps d'exécution d'une tâche Texeci :

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 VM_j.

$$T_exec_i = \frac{T_ilength}{(capacityVM_j) * PE_Number VM_j)}$$

Où :
$T_ilenght$ **:** représente la taille de la tâche en terme du nombre d'instruction ;
$capacityVM_j$ **:** est la vitesse d'un processeur en MIPS (Million Instruction Per Second) ;
PE_Number **:** représente le nombre de processeur qui se trouve dans la machine virtuelle VMj .

3.5.1.2 Estimation du Temps Transfert file :

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 :

$$TDataAccess_i = \sum_{k=1}^{n} \frac{LocalDatasetSize_k}{DiskTransfertCapacityVM_j} + TRemoteDataAccess_j$$

$$TremoteDataAccess_i = (\sum_{p=1}^{L} \frac{RemoteDatasetSize_p}{BPvm_j} + \frac{RemoteDatasetSize_p}{DiskTransertCapacityVM_j})$$

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.

3.5.1.3 Temps de libération des ressources $T - release$:

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 :

$$T - release = T - release - processorVM_j + T - release - Dataset$$

3.6 Description de notre algorithme proposé :

L'algorithme génétique est présenté dans les algorithmes 1 . Cet algorithme est composé de cinq procédures principales : (i) la génération initiale de la population ; (ii) sélection ; (iii) croisement ; (iv) mutation ; et (v) l'evaluation de la fonction du fitness. Chacune de ces étapes est expliquée ci-dessous.

Algorithme 1. GA

1: Set parameters

2: Choose encode method

3: Generate the initial population

4: while i < Max_Itereation and Best fitness < Max fitness

do

5: Fitness evaluation

6: Selection

7: Crossover

8: Mutation

9: end while

12: decode individual with minimum fitness

13: return Best solution

FIGURE 3.8 – Application de l'algorithme génétique sur le problème d'ordonnancement de workflow scientifique.

3.7 Conclusion :

Dans ce chapitre nous allons proposer une nouvelle approche d'ordonnancement de workflows scientifiques basées sur la métaheuristique (algorithme génétique). Ensuite nous avons présenté notre codage d'une solution au problème d'ordonnancement de workflow dans l'environnement cloud computing, les différents opérateurs génétiques et une nouvelle fonction de fitness générique capable de traiter le temps de réponse d'une application scientifique. Son algorithme est très simple, il donne de bons résultats et peut facilement s'adapter à des problèmes d'optimisation mono-objectifs dans les Clouds.

Notre but dans ce travail est de garantir le minimum de temps réponse et en même temps le meilleur ordonnancement de workflow, Pour cela nous proposons dans le chapitre suivant une nouvelle stratégie d'ordonnancement basées sur la métaheuristique d'algorithme génétique.

Chapitre 4

Implémentation

4.1 introduction :

Ce chapitre est consacré à la réalisation et la concrétisation des approches proposées, 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 développement :

L'approches proposée dans ce travail ont été implémentées et testées dans un environnement possédant les caractéristiques suivantes :Un processeur In- tel(R) Core(TM)i3-4005 U CPU @ 1.7 GHz, doté d'une capacité de mémoire de 4 GB ,sous Windows 10 64bits. Nous avons utilisé l'environnement de développement Eclipse.

4.2.1 Langage de programmation Java :

Java est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution portable. Il est crée par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au Sun-World. Le langage Java a la particularité principale que les logiciels écrits avec

ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que : Unix, Microsoft Windows, Mac OS ou Linux avec ou sans modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java. Java est devenu aujourd'hui une direction incontournable dans le monde de la programmation parmi les différentes caractéristiques qui sont attribuées à son succès , nous avons :

- L'indépendance de toute plate-forme : le code reste indépendant de la machine sur la quelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possède en tune Java Virtual Machine.
- Java est également portable, permettant à la simulation d'être distribuée facilement sans avoir à recompiler le code pour les différents systèmes.
- Le code est structuré dans plusieurs classes dont chacune traite une partie différente de la simulation.
- Java est multitâches : il permet l'utilisation de Threads qui sont des unités d'exécution isolées.

4.2.2 Environnements de développement :

4.2.2.1 Eclipse :

Eclipse est un environnement de développement intégré, libre, extensible, universel et polyvalent, permettant de créer des projets de développement mettant en IJuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), grâce à des bibliothèques spécifiques, ce langage est également utilisé pour écrire des extensions. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de Plug-in (en conformité avec la norme OSGi), toutes les fonctionnalités de cet atelier logiciel sont développées en tant que Plug-in. Plusieurs logiciels commerciaux sont basés sur ce logiciel libre, comme par exemple IBM Lotus Notes 8, IBM Symphony ouWebSphere Studio Application Développer, etc...

4.2.2.2 CloudSim :

L'objectif principal de simulateur CloudSim est de fournir un cadre de simulation généralisé et extensible qui permet la modélisation, la simulation et l'expérimentation des nouvelles infrastructures du Cloud Computing et les services d'application, permettant aux utilisateurs de se concentrer sur des questions de conception du système qu'ils veulent étudier, sans être préoccupé aux détails relatifs aux services et infrastructures Cloud.

Nous avons utilisé pour la réalisation de notre travail la version du simulateur CloudSim 3.0.3.

4.2.2.3 Architecture générale de CloudSim :

La Figure 4.1 montre une architecture multicouche de la structure logicielle CloudSim et ses composantes.

La couche CloudSim fournit un support pour la modélisation et la simulation des Data Centers dans l'environnement des clouds computing y compris les interfaces de gestion dédiées aux machines virtuelles, la mémoire, le stockage et la bande passante. L'affectation des VMs à des hôtes, la gestion d'exécution des applications et le suivi de l'état du système dynamique sont traités par cette couche. Un fournisseur Cloud doit implémenter ses stratégies dans cette couche afin d'étudier l'efficacité des différentes politiques qui permettent l'attribution des VMs à ses hôtes.

Au niveau supérieur de la couche CloudSim, nous avons le code utilisateur (user code) fournissant les entités de base pour les hôtes (nombre de machines, leur spécification et ainsi de suite), les applications (nombre de tâches et leurs exigences), VMs, nombre d'utilisateurs, types d'application et les politiques d'ordonnancement du Broker. Un développeur d'applications Cloud peut générer plusieurs activités parmi lesquelles nous citons : Des Scénarios de disponibilité des modèles clouds, effectuer des tests robustes basés sur les configurations personnalisées supportées par le CloudSim et implémenter des techniques de provisionnement des applications personnalisées dans les clouds.

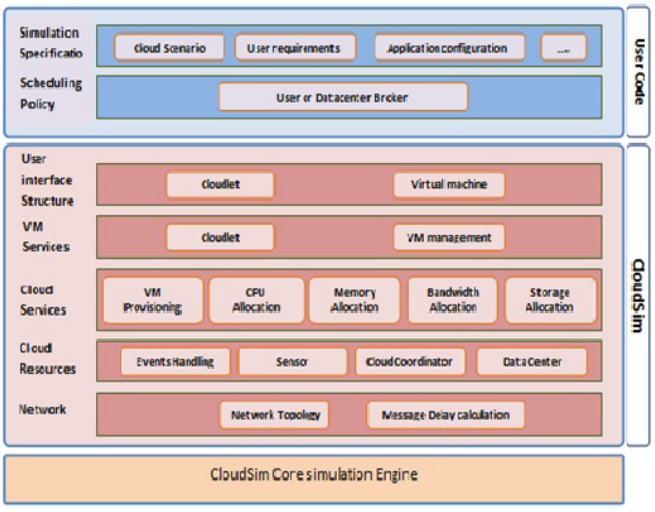

FIGURE 4.1 – Architecture générale de CloudSim

4.2.2.4 Les classes de CloudSim :

Le simulateur CloudSim est composé de plusieurs classes que nous pouvons classer en deux catégories : des classes qui modélisent les entités comme le Data Center, le Broker ,etc .et des classes modélisant les politiques d'allocation.

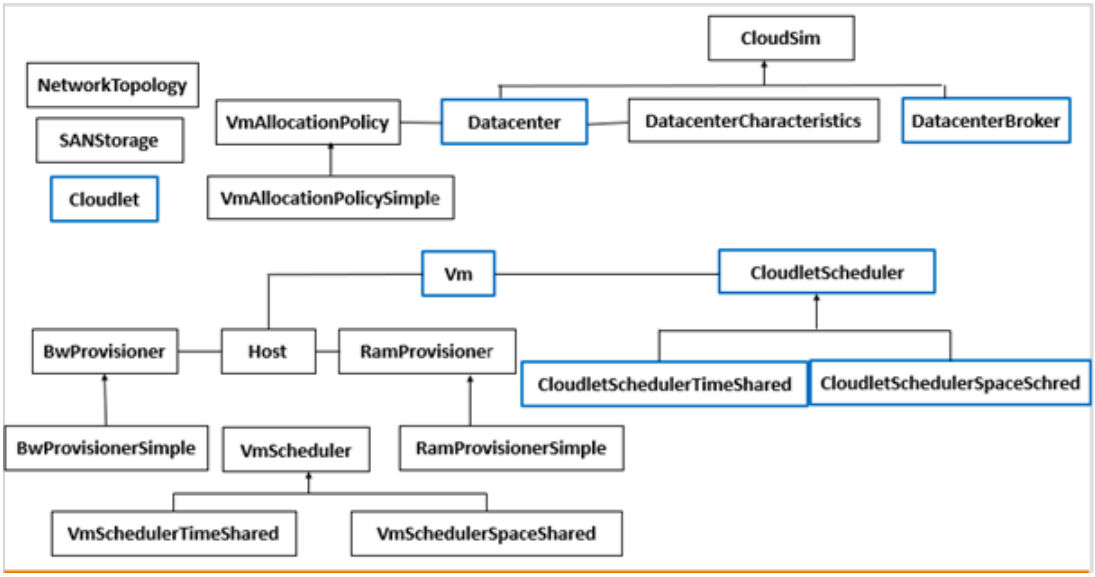

FIGURE 4.2 – Les classes de CloudSim

SimEntity : Il s'agit d'une classe abstraite, elle représente l'entité de simulation qui est capable d'envoyer des messages à d'autres entités et de gérer les messages reçus ainsi que les évènements. Toutes les entités doivent étendre cette classe et redéfinir ses trois principales méthodes : startEntity(), processEvent() et shutdownEntity(). Ces méthodes définissent les actions pour l'initialisation de l'entité, le traitement des évènements et la destruction de l'entité.

Datacenter : Cette classe modélise les services au niveau des infrastructures de base (matériel) qui sont offerts par les fournisseurs de Cloud (Amazon, Azure, App Engine). Elle encapsule un ensemble d'hôtes qui peuvent être homogènes ou hétérogènes par rapport à leurs configurations matérielles (mémoire, noyaux, capacité et stockage). La classe PowerDataCenter. Java utilisée dans le package «Power» hérite de la classe DataCenter. Java et elle permet la simulation des centres de données en calculant leurs énergie consommée.

DatacenterBroker : Cette classe modélise le Broker, qui est responsable de la médiation des négociations entre les utilisateurs et les prestataires de service

selon les conditions de QoS des utilisateurs.

Il déploie aussi les tâches de service à travers les Clouds. Le Broker, au nom des utilisateurs, agit sur les prestataires du service approprié 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.

VM : Cette classe représente une instance de machine virtuelle(VM) qui est gérée et hébergée par une machine physique (hôte). Chaque composant VM a accès à un composant qui stocke les caractéristiques suivantes liées à une VM telles que : mémoire accessible, le processeur, capacité de stockage et les poli- tiques de provisionnement interne de la machine virtuelle. Dans le but de réduire l'énergie consommée par les DataCenters, la classe VMPower. Java héritant de la classe VM.java enregistre l'historique de l'utilisation de CPU.
Cet historique est utilisé dans les politiques de sélection et d'allocations des VMs.

Cloudlet : Cette classe modélise les services d'application du Cloud (comme la livraison, réseaux sociaux et le workflow d'affaires). CloudSim représente la complexité d'une application en fonction de ses besoins informatiques.
Chaque service d'application à une taille d'instruction pré-assigne et la quantité de flux de transfert de données qu'il doit entreprendre au cours de son cycle de vie.
Cette classe peut également être étendu pour supporter la modélisation de la performance et d'autres paramètres de composition pour les applications telles que les transactions dans les applications orientées base de données.

4.3 Interface principale :

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

4.3.1 Configuration des paramètres de simulation :

Pour lancer la simulation d'un Cloud avec notre version étendu du simulateur CloudSim, nous devons configurer les paramètres de simulation dans un premier temps.
Les Figures au dessous montrent les différentes fenêtres de configuration de simulation qui contient 4 parties principales :

4.3.1.1 Datacenters :

Cette étape consiste à faire entrer les paramètres nécessaires propres à la topologie du réseau comme : le nombre de Datacenter, d'hôtes, de CPU, la vitesse de chaque CPU, le cout de traitement, la taille de la mémoire, le cout de la mémoire, la taille du disque dur, le cout de stockage et la bande passante ainsi que son cosut.

Cette étape consiste à faire entrer les paramètres nécessaires propres à la topologie du réseau comme : le nombre de Data Center, d'hôtes, de CPU, la vitesse de chaque CPU, le cout de traitement, la taille de la mémoire, le cout de la mémoire, la taille du disque dur, le cout de stockage et la bande passante ainsi que son cout.

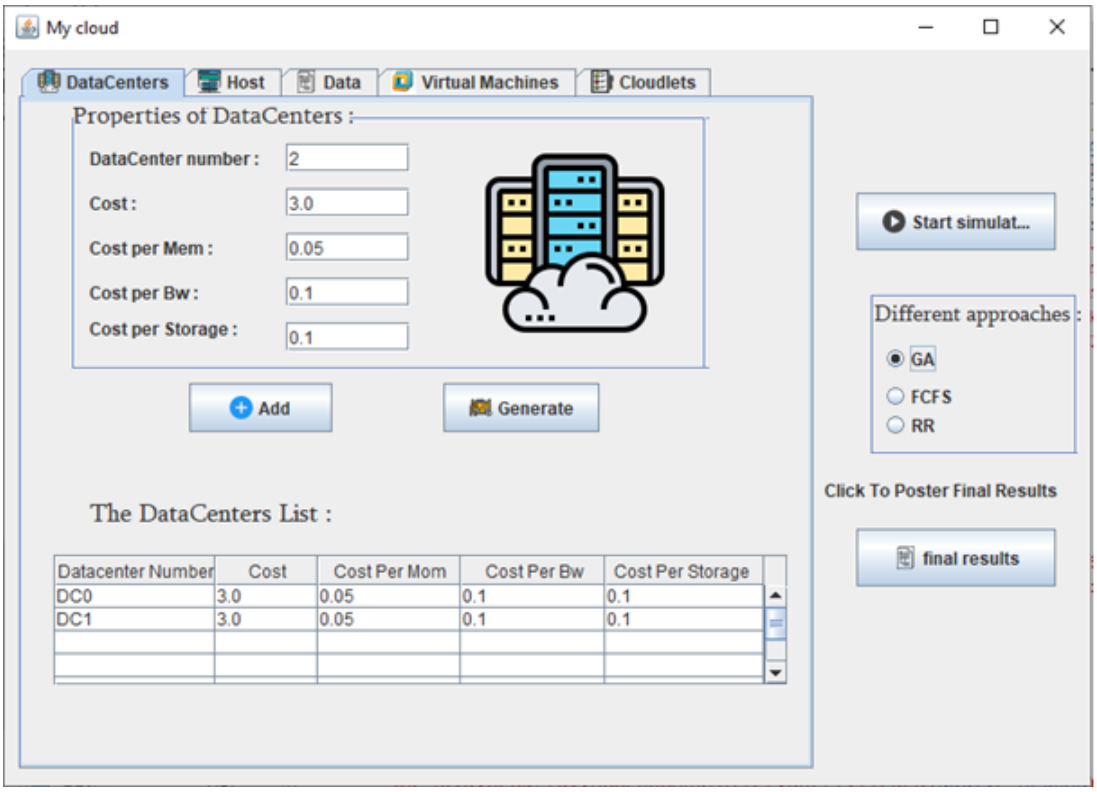

FIGURE 4.3 – Configuration du Datacenter

4.3.1.2 Host :

La Figure 4.4 représente l'étape de création des machines physiques. La création se fait par le click sur le boutton Add, et d'une façon hétérogène en précisant le nombre des Host, nombre de processeurs, MIPS, RAM, capacité de stockage et la bande passante.

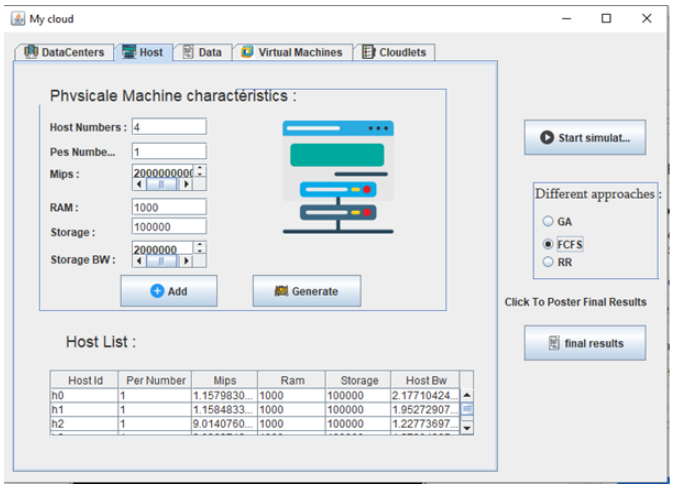

FIGURE 4.4 – Configuration du Host

4.3.1.3 Les données :

Cette étape consiste à crées des données. Pour ajouter une donnée, on doit configurer les paramètres suivants : le nom de la donnée, sa taille, le Datacenter qui héberge cette donnée et l'hôte où se trouve cette donnée.

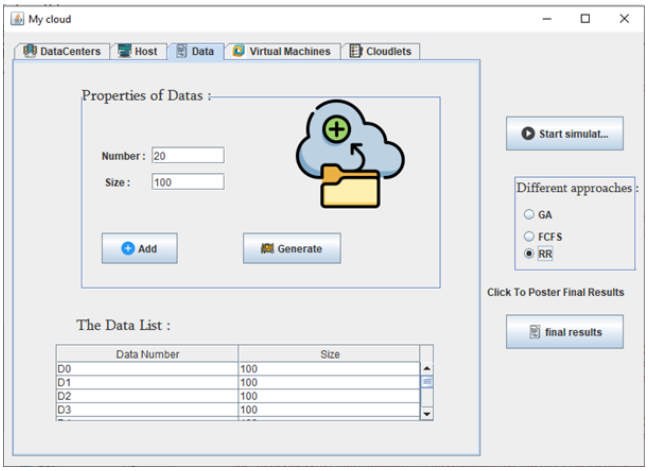

FIGURE 4.5 – Configuration des données

4.3.1.4 Virtual Machines :

Permet de configurer les machines virtuelles et leur caractéristique : le nombre de CPU dans une VM la taille de la mémoire, la taille du disque dur et la bande passante.

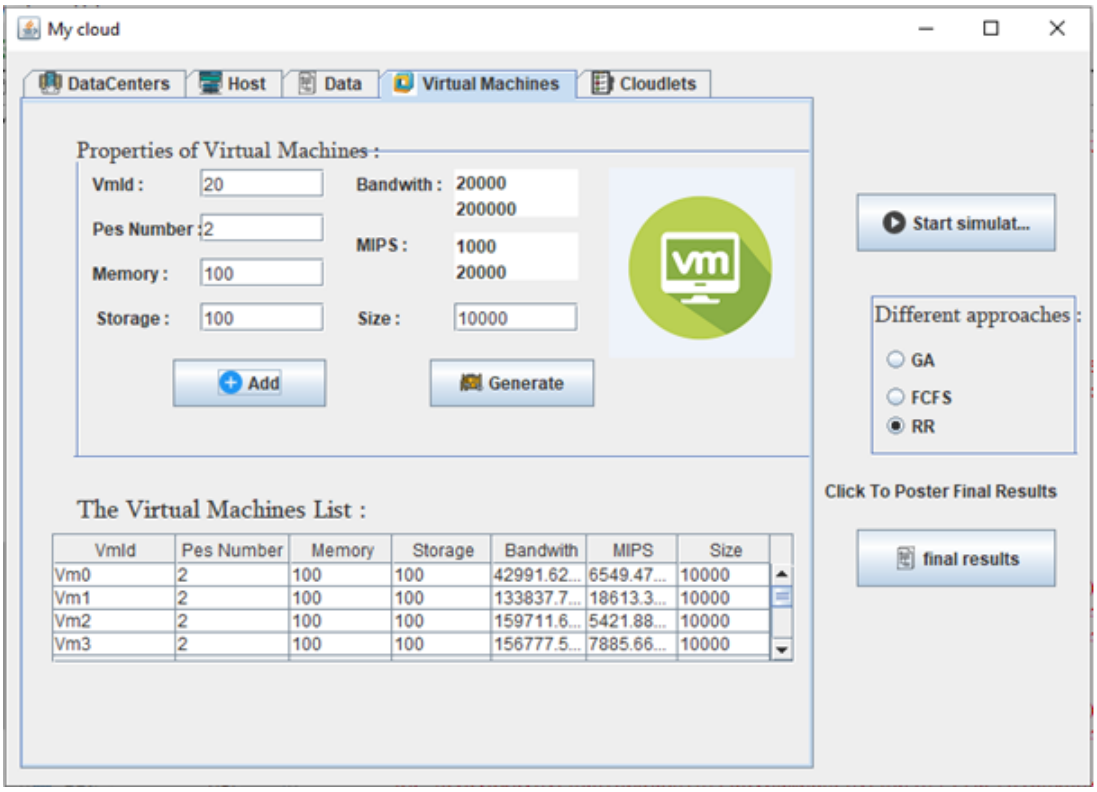

FIGURE 4.6 – Configuration des machines virtuelles

4.3.1.5 Cloudlet :

Permet de configurer les propriétés de la Cloudltet et leur caractéristiques : la taille de la Cloudlet, le fichier requis pour l'exécution, le budget de cloudlet et le temps d'utilisation de fichier.

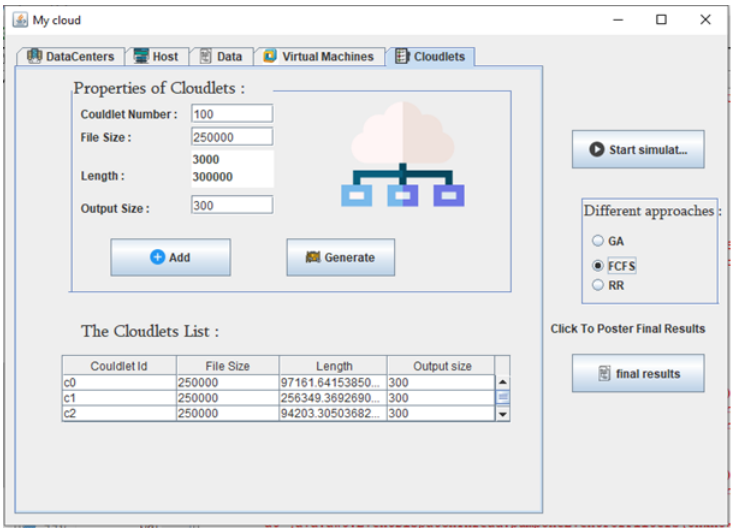

FIGURE 4.7 – Configuration des Cloudlets

4.3.2 Lancement de la simulation :

Après avoir personnaliser les paramètres de simulation, l'utilisateur peut lancer la simulation en cliquant sur le bouton Start Simulation ou bien à partir du menu Simulation en choisissant l'item qui correspond à son choix (voir Figure 4.12). Après la création du Cloud et le lancement de la simulation, l'utilisateur peut visualiser les résultats et l'état de traitement de Cloudlets. Les résultats obtenus après la simulation, ils sont accessibles en cliquant sur le bouton final results.

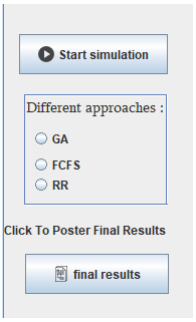

FIGURE 4.8 – Interface de lancement de la simulation et de la visualisation des résultats

4.3.3 Résultats expérimentaux :

Pour mettre en valeur les apports de notre approche, nous allons se focaliser sur la métrique de temps de réponse. Dans le but d'étudier le comportement de notre proposition et d'analyser ses résultats obtenus par la simulation, nous allons les comparer d'autre approche. Plusieurs séries de simulation ont été lancées.

4.3.3.1 Expérience 1 : Impact de nombre de cloudlets sur le temps de réponse

Dans cette expérience, nous avons créé deux Data Center contenant 4 Host hétérogène, chaque Host posséde un seul processeur avec une vitesse variante en MIPS entre(20000, 200000), bande passante entre (10000,20000), la taille de la donnée est 200 MB. Cette simulation consiste à varier le nombre de Cloudlet par pas de 200 et la taille variante en length entre(3000,30000) voir l'impact sur le temps de réponse.

Nombre de cloudlet	**GA**	**FCFS**	**RR**
200	941	3028	9040
400	1299	5985	10500
600	1800	8990	12060
800	2541	20901	46100
1000	2910	51890	98120

TABLE 4.1 – Impact du nombre de Cloudlets sur le temps de Réponse

Les résultats du tableau 4.1 sont schématisés dans la Figure 4.8

Dans le tableau 4.3 consiste à faire une simulation en utilisant la variation du nombre de Cloudlets, nous remarquons une très grande différence dans le temps de réponse avec l'approche proposeé par rapport aux autres stratégies.

4.3.3.2 Expérience 2 : Impact de la taille de donnée sur le temps de réponse

Dans cette expérience, nous avons créé deux Data Center contenant 4 hôsts hétérogène, Chaque hôte posséde un seul processeur avec une vitesse variante en MIPS entre (20000,200000), bande passante entre(10000,20000),on fixes le nombre de cloudlets a 200 Coudlets. Cette simulation consiste à varier la taille de la donnée (200, 400, 600, 800, 1000) ;

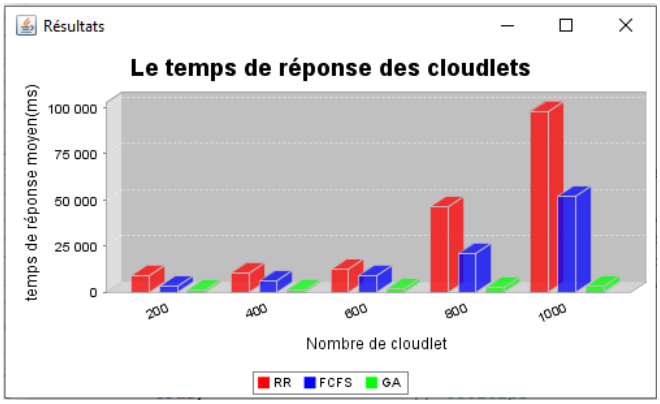

FIGURE 4.9 – Impact du nombre de Cloudlets sur le temps de Réponse

Taille de donnée	**GA**	**FCFS**	**RR**
200	9000	12305	30500
400	10000	18000	40035
600	15100	20027	56290
800	20000	31008	64500
1000	25015	40000	80085

TABLE 4.2 – Impact de la taille de donnée sur le temps de réponse

Les résultats du tableau 4.2 sont schématisés dans la Figure 4.10

Le tableau 4.2 est un tableau de comparaison qui se repose sur trois stratégies (RR , FCFS , GA) pour une seul variant (la taille de donnée), ou nous pouvons déduire que le temps de réponse de notre stratégie est beaucoup plus réduit par rapport aux deux autres stratégies.

4.3.3.3 Expérience 3 : Impact de nombre de de VM sur le temps de réponse

Dans cette expérience, nous avons créé deux Data Center contenant 4 Host hétérogène, chaque Host posséde un seul processeur avec une vitesse variante en MIPS entre(20000, 200000), bande passante entre (10000,20000), la taille de la donnée est 200 MB,le nombre de Cloudlets est fixé a 100 Cloudlets. Cette expérience consiste à varier la vitesse de VM entre(1000 , 20000) et la bande passante entre (2000 ,20000) et le nombre de Vms (10, 20, 30, 50, 100) et voir l'impact sur le temps de réponse.

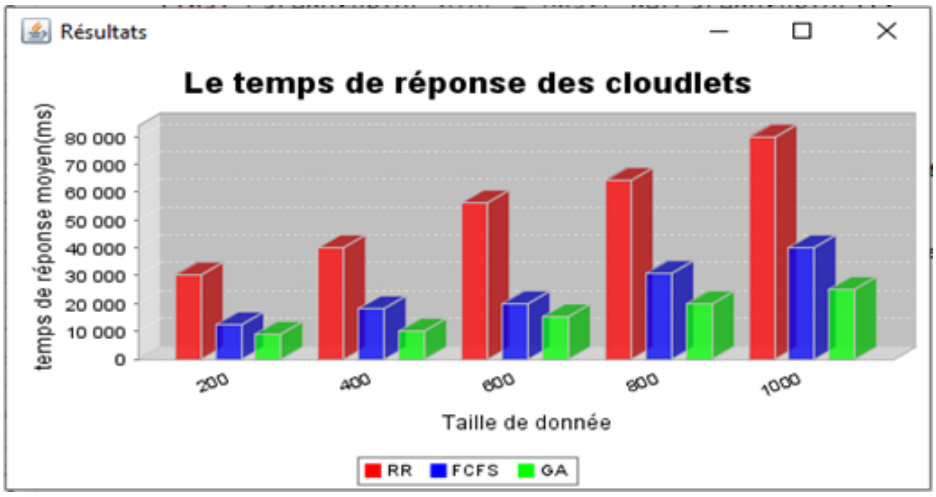

FIGURE 4.10 – Impact de la taille du donnée sur le temps de Réponse

Nombre de VM	**GA**	**FCFS**	**RR**
10	992	32931	45000
20	890	8126	25400
30	859	6900	19000
50	675	6153	7500
80	110	4102	4890

TABLE 4.3 – Impact de nombre de de VM sur le temps de réponse

Les résultats du tableau 4.3 sont schématisés dans la Figure 4.11

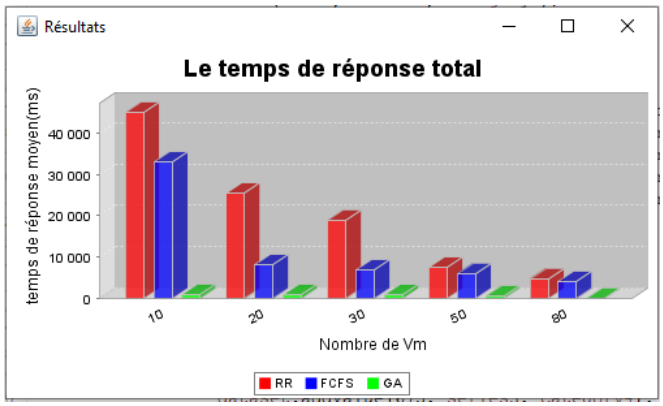

FIGURE 4.11 – Impact du nombre de Vm sur le temps de Réponse

Dans le tableau 4.3 consiste à faire une expérimentation sur la valeur du nombre de machine virtuelles utilise ou on remarque une diminution dans le temps d'exécutions par contre aux autres stratégie on des valeurs supérieures à notre approche proposeé.

4.3.3.4 Expérience 4 : Comparaison entre les trois approches sur le temps de réponse moyenne

Une simulation a été lancé selon plusieurs variantes pour faire comparaison entre les trois approches, et nous avons choisis les mêmes paramètres cités ci dessue pour montrer les résultats graphiques nous avons utilisés la bibliothéque JFreeChart. Dans cette simulation, nous avons créé deux Data Center contenant :

- 4 hosts hétérogène, Chaque Host possède un seul processeur.
- 20 machines virtuelles.
- 20 données, chaque donnée est de taille (100 KB).
- Nous supposons que nous avons reçu 100 Cloudlets.

La stratégie proposée est largement meilleure que l'autres

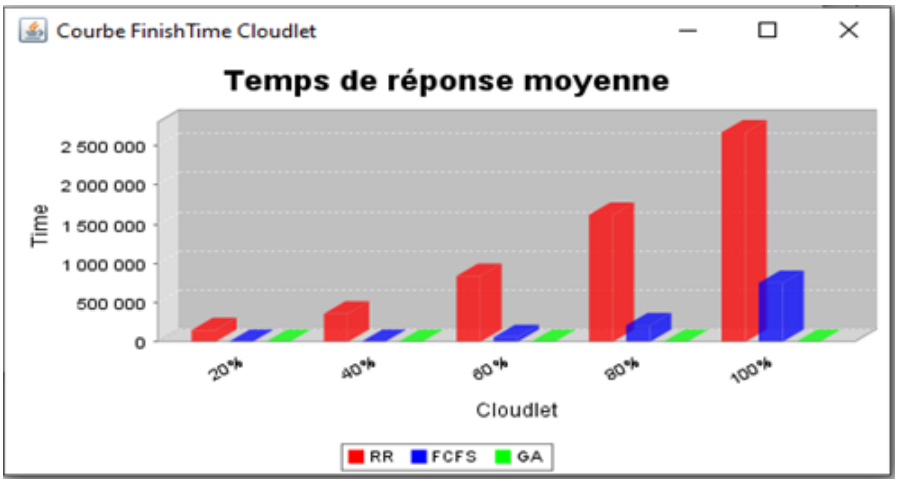

FIGURE 4.12 – Comparaison entre les trois approches sur le temps de réponse moyenne

Nous avons refait la même procédure pour mesurer le temps de réponse moyen par série de Cloudlets. Nous avons dévisser notre liste de Cloudlet en cinq sous liste chaque sous liste content 20Nous remarquons que la différence de temps de réponse entre les trois approche (GA , FCFS,RR) est égale dans la première 80% de la simulation. En outre chaque fois en augmente le nombre de Cloudlets ,le temps d'exécution augmente a des valeurs bas dans notre approche mais contrairement aux autres approches.

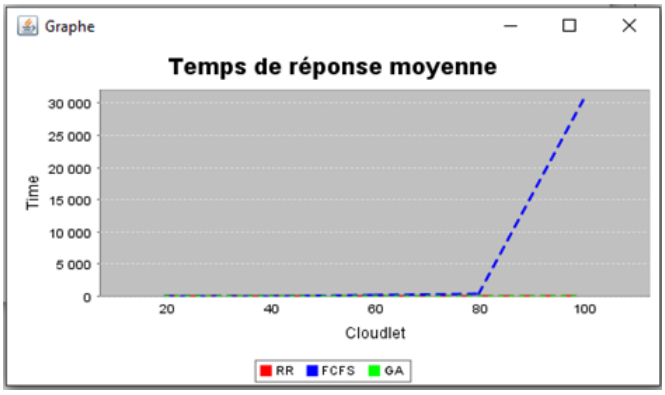

FIGURE 4.13 – Comparaison de moyen temps de réponse par série entre les trois Approches

L'histogramme de la figure 4.14 donne plus de détails sur le temps de réponse pour chaque requête. Il montre les résultats de temps de réponse obtenu par chaque requête et ceci en utilisant les trois startégies GA ,FCFS et RR.

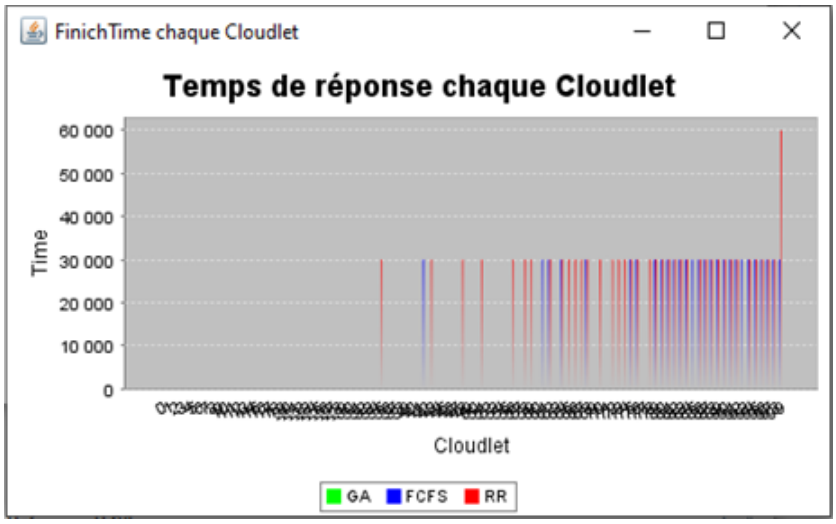

FIGURE 4.14 – Histogramme de Comparaison de temps de réponse entre les Trois Approches

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 de simulations dans le but de comparer notre approches avec l'approche FCFS et l'approche RR tout en variant différents paramètres comme : le nombre Data , le nombre de VM, nombre de Cloudlet. Les résultats montrent que l'utilisation de notre approche réduit le temps de réponse de Cloudlets.

Conclusion générale et perspectives

Le cloud computing est en pleine expansion et tend à s'imposer comme un des paradigmes Dominants dans l'univers informatique. Les infrastructures proposant des services de cloud computing deviennent donc de plus en plus nombreuses, et de plus en plus complexes pour répondre à cette demande croissante de services décentralisés. Aujourd'hui on n'a pas un problème de stockage mais on a un problème de gestion et de récupération de l'information avec le minimum temps.Il faut donc concevoir des techniques et outils afin de répondre à ces nouveaux besoins de gestion.

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 ils'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'ordonnancement basée sur l'algorithme génétique. Dans ce travail, nous avons simulé notre approche avec FCFS et RR, Nous avons étendu le simulateur CloudSim Pour implémenter l'approche proposé, tester notre algorithme 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 :

- Prendre en considération l'aspect consommation d'énergie.
- Le cloud computing se base sur le principe pays as you go c-à-d payer ce que vous allez utiliser. On aimerais bien d'intégrer l'aspect économique par le contrat SLA(Service Level Agreement).
- 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 comme tournoir , éliste et universelle stochastique pour prouver son utilité.

Bibliographie

[1] Mell, P. and Grance, T. The NIST Definition of Cloud Computing 08/03/2020

[2] https ://azure.microsoft.com/fr-fr/overview/what-is-cloud-computing/cloud-computing-models consulter le 05/03/2020

[3] Documentation technique Definition Cloud Computing https ://www.itnation.lu/content/uploads/2017/10/cloudcomputingdefinitions.pdf consulter 10/03/2020

[4] Sonia YASSA.Allocation optimale multi-contraintes des workflows aux ressources d'un environnement cloud computing,PhD thesis, Université de Cergy-Pontoise

[5] Chapitre 1 : Les concepts du cloud computing " https ://www.institut-numerique.org/chapitre-1-les-concepts-du-cloud-computing-51c0279ca534a "consulter le 11/03/2020

[6] Licence Professionnelle : Administration de systèmes, réseaux et applications à base de logiciels libres IUT Nancy CharlemagneCloud Computing" https ://members.loria.fr/LNussbaum/files/ptasrall2010-cloud-computing-rapport.pdf "consulter le 11/03/2020

[7] " https ://www.tutorialride.com/cloud-computing/cloud-computing-technologies.htm "consulter le 11/03/2020

[8] Une stratégie de réplication de données pour gérer la tolérance aux pannes et l'équilibrage de charge dans le cloud computing.thèse.

[9] FEMMAM Manel. Une approche formelle pour la planification des tâches pour la QoS dans le cloud-computing.thèse

[10] CLOUD COMPUTING ET SECURITE : Une architecture organique pour la sûreté de fonctionnement des processus métiers. MEMOIRE POUR L'OBTENTION DU DIPLOME DE MAGISTER EN INFORMATIQUE

[11] https ://www.researchgate.net/figure/Cloud-Computing-Deployment-Models-Mell-and-Grance-2011fig2275036700 23/04/2020

[12] https ://www.researchgate.net/figure/Virtualization-in-Cloud-Computing-fig1-317244855 23/04/2020

[13] https ://www.slideshare.net/anshib/basics-of-cloud-omputing 23/02/2020

[14] Scientific Workflow Scheduling for Cloud Computing Environments. DOCTORAL THESIS

[15] 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.

[16] Maheswaran, M., Ali, S., Siegal, H. J., Hensgen, D., and Freund, R. F. (1999). Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In Heterogeneous Computing Workshop, (HCW'99) Proceedings. Eighth (pp. 30-44).

[17] Park, G. L., Shirazi, B., and Marquis, J. (1997, April). DFRN : A new approach for duplication based scheduling for distributed memory multiprocessor systems. In Parallel Processing Symposium, Proceedings., 11th International (pp. 157-166).

[18]] Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm. Journal of optimization theory and applications, 45(1), (pp. 41-51).

[19] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers operations research, 13(5), (pp.533-549).

[20] Yassa, S. Allocation optimale multicontraintes des workflows aux ressources d'un environnement Cloud Computing, thése de Doctorat, Cergy-Pontoise.2014.

[21] Yassa, S., Chelouah, R., Kadima, H., and Granado, B. (2013, July). A Genetic Algorithm Approach to a Cloud Workflow Scheduling Problem with Multi-QoS Requirements. In 26th European Conference on Operational Research Workshop.

[22] Yassa, S., Sublime, J., Chelouah, R., Kadima, H., Jo, G. S., and Granado, B. (2013). A genetic algorithm for multi-objective optimisation in workflow scheduling with hard constraints. International Journal of Metaheuristics, 2(4), (pp.415-433).

[23] M.E, Computer Engineering, Alpha College of Engineering, Gujarat, India,"A SURVEY OF VARIOUS SCHEDULING ALGORITHM IN CLOUD COMPUTING ENVIRONMENT"

[24] Huankai Chen Future Computing Group, Professor Frank Wang Future C omputing Group, Dr Na Helian University of Hertfordshire, Gbola Akanmu

Future Computing Group. "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing" Journal of Parallel and Distributed computing,

[25] https ://www.geeksforgeeks.org/program-for-shortest-job-first-or-sjf-cpu-scheduling-set-1-non-preemptive/ 25/08/2020

[26] Site web http ://igm.univ mlv.fr/ dr/XPOSE2013/tleroux-genetic-algorithm/fonctionnement.html : 20/04/2020

[27] Mohialdeen, I. A. (2013).Comparative Study of Scheduling Algorithms in Cloud Computing Environment. Journal of Computer Science, 9(2), 252.

[28] Singh, L., Ahmed, J.(2014). A Greedy Algorithm For Task Scheduling and Resource Allocation Problems In Cloud Computing,International Journal Of Research and Development In Technology And Management Science,21(1)

[29] Mustafa, L. M., Elmahy, M. K., and Haggag, M. H. (2014). Improve Scheduling Task based Task Grouping in Cloud Computing System. International Journal of Computer Applications, 93(8) ,0975 - 8887

[30] Lovesum, S. J., Krishnamoorthy, K., and Prince, P. (2014). An Optimized Qos Based Cost Effective Resource Scheduling In Cloud. Journal of Theoretical and Applied Information Technology, 66(1).

[31] Kaur, R., and Kinger, S. (2014). Analysis of Job Scheduling Algorithms in Cloud Computing. International Journal of Computer Trends and Technology (IJCTT),9(7), 379- 386.

[32]]] Annette, J, R., Banu,W, A., and Shriram, S. (2013). A Taxonomy And Survey Of Scheduling Algorithms In Cloud : Based On Task Dependency. International Journal of Computer Applications, 82(15), 20-26.

[33] Sucha SMANCHAT et Suchon SRITAWATHON. "A Scheduling Algorithm for Grid Workflow Using Bottleneck Detection and Load Balancing". In : IJWIS 10.3 (2014), p. 263-274.

[34] Luiz F. BITTENCOURT, Edmundo R. M. MADEIRA et Nelson L. S. Da FONSECA. "Scheduling in Hybrid Clouds". In : IEEE Communications Magazine 50.9 (2012), p. 42-47.

[35]] Lingfang ZENG, Bharadwaj VEERAVALLI et Xiaorong LI. "SABA : A Security- Aware and Budget-Aware Workflow Scheduling Strategy in Clouds". In : J. Parallel Distrib. Comput 75 (2015), p. 141-151.

[36]] Hamid Mohammadi FARD et al. "A Multi-Objective Approach for Workflow Scheduling in Heterogeneous Environments". In : CCGRID. IEEE Computer Society, 2012, p. 300-309.

[37] Saeid ABRISHAMI, Mahmoud NAGHIBZADEH et Dick H. J. EPEMA. "Cost- Driven Scheduling of Grid Workflows Using Partial Critical Paths". In : IEEE Trans. Parallel Distrib. Syst 23.8 (2012), p. 1400-1414.

[38] L. ZHAO, Y. REN et K. SAKURAI. "Reliable Workflow Scheduling with Less Resource Redundancy". In : Parallel Computing 39.10 (2013), p. 567-585.

[39] F. Z. FILALI et B. YAGOUBI. "Classifying and Filtering Users by Similarity Measures for Trust Management in Cloud Environment". In : Scalable Computing : Practice and Experience 16.3 (2015), p. 289-302.

[40] Yonghong LUO et Shuren ZHOU. "Power Consumption Optimization Strategy of Cloud Workflow Scheduling Based on SLA". In : WSEAS Transactions on Systems 13 (2014), p. 368-377.

[41] Paul WATSON 0001. "A Multi-Level Security Model for Partitioning Workflows over Federated Clouds". In : J. Cloud Computing 1 (2012), p. 15.

[42] Sucha SMANCHAT et al. "Scheduling Multiple Parameter Sweep Workflow Instances on the Grid". In : eScience. IEEE Computer Society, 2009, p. 300-306.

[43] Saeid ABRISHAMI, Mahmoud NAGHIBZADEH et Dick H. J. EPEMA. "Deadline- Constrained Workflow Scheduling Algorithms for Infrastructure as a Service Clouds". In : Future Generation Comp. Syst 29.1 (2013), p. 158-16

[44] Gabriel MATEESCU, Wolfgang GENTZSCH et Calvin J. RIBBENS. "Hybrid Computing - Where HPC Meets Grid and Cloud Computing". In : Future Generation Comp. Syst 27.5 (2011), p. 440-453.

[45] Eun-Kyu BYUN et al. "Cost Optimized Provisioning of Elastic Resources for ApplicationWorkflows". In : Future Generation Comp. Syst 27.8 (2011), p. 1011- 1026.

[46]] Eun-Kyu BYUN et al. "BTS : Resource Capacity Estimate for Time-Targeted Science Workflows". In : J. Parallel Distrib. Comput 71.6 (2011), p. 848-862.

[47] Faten KARIM et Giselle RAMPERSAD. "Cloud Computing in Education in Developing Countries". In : Computer and Information Science 10.2 (2017), p. 87-96.

[48] A.I.Awada, N.A.El-Hefnawyb, H.M.Abdel-kaderc. ŞEnhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environmentsăż : International Conference on Communication, Management and Information Technology (ICCMIT 2015).