



République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et systèmes distribués(R.S.D)

Thème

**Ordonnancement des tâches et
allocation des ressources
Dans le Cloud –Computing**

Réalisé par :

- Kebir Fatima Zohra

Présenté le 13 Septembre 2018 devant le jury composé de MM.

- Mme ILES Nawel (Président)
- Mr Belhocine Amine (Encadreur)
- Mr Bambrik Ilyas (Examineur)

Année universitaire : 2017-2018

TABLE DES MATIERES

Introduction générale	4
- Contexte	4
- Problématique et motivation	4
- Contributions	5
- Organisation de l'étude.....	5
1 Le Cloud Computing	6
1.1 Introduction	7
1.2 Définition.....	7
1.3 Historique du Cloud Computing	8
1.4 Les principes de Cloud Computing	9
1.5 Les avantages et les limites de Cloud Computing.....	10
1.5.1 les avantages.....	10
1.5.1 les limites.....	10
1.6 Les modèles de services de Cloud Computing.....	12
1.6.1 SaaS « Software as a service »	12
1.6.2 PaaS « Platform as a service »	12
1.6.3 IaaS « Infrastructure as a service »	12
1.7 Les modèles de déploiement de Cloud Computing	13
1.7.1 Le Cloud public « Public Cloud Computing »	14
1.7.2 Le Cloud privé « Private Cloud Computing ».....	14
1.7.3 Le Cloud hybrid « Hybrid Cloud Computing ».....	14
1.8 Conclusion.....	15
2 L'ordonnancement des tâches et l'allocation des ressources	16
2.1 Introduction	16
2.2 Ordonnancement : Concepts et définitions	17
2.3 Les problèmes d'ordonnancement en ligne et hors ligne	19
2.4 Les critères d'optimisation	20
2.5 L'ordonnancement dans le cloud computing	21
2.6 Les principaux algorithmes d'ordonnancement	22
2.7 Conclusion	25

3 Implémentation et performance.....	..27
3.1 Introduction	27
3.2 Analyse comparative des algorithmes d'ordonnancement de cloud- sim dans le cloud computing	28
3.3 Comparaison entre les algorithmes d'ordonnancement des taches (FCFS, Round-Robin , ShortFirst).....	29
3.4 Comparaison entre les algorithmes d'ordonnancement des tâches et d'allocation des ressources (Max-Min et Min-Min).....	31
3.5 Comparaison entre les algorithmes d'ordonnancement des taches et d'allocation des ressources (FCFS, Round-Robin , ShortFirst Max-Min et Min-Min).....	34
3.6 Proposition d'un algorithme qui donne un résultat optimal pour le critère makespan	36
3.7 Conclusion	40
Conclusion Générale	41

Sommaire

- **Contexte**
 - **Problématique et motivation**
 - **Contributions**
 - **Organisation de la thèse**
-

Contexte

Le Cloud computing est une infrastructure dans laquelle la puissance de calcul et le stockage des données sont gérés par des serveurs distants et les utilisateurs se connectent avec l'internet , les ordinateurs , les téléphones mobiles , les tablettes et même d'autres objets connectés sont considérés comme des points d'accès pour accéder a des données stockées et des applications , le cloud computing permet au usagers l'utilisation de capacité de stockage et la puissances de calcul selon leurs besoin .

La technologie de Cloud computing représente un nouveau paradigme pour la fourniture des ressources informatiques .cette technologie a plusieurs avantages telles que l'élasticité des ressources , le modèle de payment de « pay-per-use » et d'autres avantages .

Problématique et motivation

Le Cloud computing permet de faire des allocations des ressources informatiques , mais les ressources ne sont pas toujours suffisantes . Elles ne peuvent pas souvent répondre aux besoin des utilisateurs . alors les mécanismes d'ordonnancement des taches et d'allocation des ressources sont nécessaires pour améliorer des critères d'optimisation .Les nouvelles stratégies peuvent employer quelques concepts d'ordonnancement et d'allocation pour fournir une meilleur ordonnancement des taches et allocation des ressources .

Contributions

Dans les travaux de cette étude , nous étudions 3 algorithmes (FCFS , ShortFirst , Round robin) pour le critère d'optimisation Dead-Line , Après avoir fait l'implémentation de Max-min et Min-max je compare entre ces deux algorithmes d'ordonnement pour le critère Make-Span , et à la fin je propose et j'implémente un autre algorithme d'ordonnement des tâches et d'allocations des ressources que je vois plus efficace et optimale pour le critères Make-Span que Min-min et Max-min .

Organisation de l'étude :

Le reste de l'étude est organisé comme suit : Dans le premier chapitre nous présentons les notions de base des concepts que nous jugeons nécessaires à la compréhension du contenu de cette étude . Nous présentons d'abord les concepts du Cloud computing et les notions fondamentales , les avantages et les limites , ensuite nous présentons les modèles de service de Cloud computing et nous terminons par les modèles de déploiement de Cloud computing . Dans le deuxième chapitre , nous parlons des concepts liés à l'ordonnement des tâches et l'allocation des ressources , ainsi que les travaux réalisés dans ce domaine . Le troisième chapitre est consacré à l'affichage de quelques résultats d'expérimentation et leurs interprétations.

Sommaire

1.1 Introduction

1.2 Définition

1.3 Historique du Cloud Computing

1.4 Les principes de Cloud Computing

1.5 Les avantages et les limites de Cloud Computing

1.5.1 les avantages

1.5.2 les limites

1.6 Les modèles de services de Cloud Computing

1.6.1 SaaS « Software as a service »

1.6.2 PaaS « Platform as a service »

1.6.3 IaaS « Infrastructure as a service »

1.7 Les modèles de déploiement de Cloud Computing

1.7.1 Le Cloud public « Public Cloud Computing »

1.7.2 Le Cloud privé « Private Cloud Computing »

1.7.3 Le Cloud hybrid « Hybrid Cloud Computing »

1.8 Conclusion

1.1 Introduction

Le monde de la technologie est un monde plein de développement et de changement, parmi les développements dans le domaine de la technologie en ce moment la « **virtualization** » et « **le Cloud Computing** ».

Que veut-on dire par la virtualization et quelle est sa relation avec le Cloud Computing ?

C'est un terme très ancien qui existe depuis les années 60 du dernier siècle, à la base c'était l'idée du professeur **John Maccarthy**. Il rêvait de convertir les services d'un micro-ordinateur normal ou l'infrastructure normale en une infrastructure virtuelle mais cette idée ne s'appliquait pas, et personne n'a travaillé sur cette idée jusqu'au début des années **2000**. L'idée s'est développée d'une manière incroyable et génial avec ce qu'on appelle « le Cloud Computing ». Ceci va dans le sens de ce qu'a dit **John Maccarthy** « le Cloud Computing peut être un jour organisé comme une entreprise d'utilité publique (comme les services publics , tel que l'eau et l'électricité).

La virtualization est exactement la transformation d'une infrastructure du monde réel en une infrastructure dans un monde virtuel.

Exemple : Si on a une société qui a besoin de serveurs de grande capacité donc très coûteux, elle n'a pas besoin d'acheter ses serveurs et peut directement les louer en utilisant le cloud. Ce qu'il lui évitera le casse tête de l'achat, de l'installation et de la maintenance du matériel (serveurs). Le cloud fournit donc des services élevés à l'utilisateur et en même temps lui permet d'accéder aux données (« Data ») à tout moment et de n'importe où et donc travailler à distance.

Le Cloud Computing est basé sur la virtualization , mais c'est quoi le Cloud Computing ?

1.2 Définition

Le Cloud Computing est une technologie très sophistiquée basée sur le transfert des traitements et d'espace de stockage de l'ordinateur à ce qu'on appelle « le Cloud » qui est

un serveur (ou ensemble de serveurs) qu'on peut accéder via Internet, et donc les programmes technologiques passent des produits aux services.[1]

Le Cloud Computing représente une richesse dans le monde Internet et son impact est évident dans le monde des affaires, en particulier, d'après la définition de l'Institut National Américain des Normes et de la Technologie, le Cloud Computing est un modèle de réseau approprié nécessaire pour accéder à un ensemble commun des ressources informatiques « Configurable Computing » (ex : Réseaux, Serveurs, Stockage, Application, Services) qui peuvent être lancés avec un minimum d'efforts administratifs ou d'interférence du fournisseur de service [2].



Figure 1-1 : Cloud Computing [2]

1.3 Historique du Cloud Computing

Comme tous les concepts relevant autant de l'économie que de la Technologie, il est difficile de dire avec précision quand a été inventé le Cloud Computing.

L'idée de Cloud Computing existait il y a plus de soixante années avec ce qu'on appelle des ordinateurs centraux qui sont utilisés pour connecter un écran et un clavier avec un énorme ordinateur central jouant le rôle de l'esprit et qui se trouve ailleurs, mais avec l'émergence des ordinateurs personnels qui ont des unités de stockage et des micro-processeurs indépendants, l'idée commença à disparaître, mais avec l'avènement de l'internet l'idée est revenue de nouveau.

On ne peut parler de Cloud Computing sans parler de l'Internet. Au début, en 1962, l'Internet est lancé par l'ARPA (« *Defense Advanced Research Projects Agency (DARPA)*, Agence pour les projets de recherche avancée de défense ») qui est une agence du département de la Défense des États-Unis chargée de la recherche et développement des nouvelles technologies destinées à un usage militaire. Jusqu'à

aujourd'hui, la DARPA a été à l'origine du développement de nombreuses technologies qui ont eu des conséquences considérables dans le monde entier dont les réseaux informatiques. Mais le nouveau dans le Cloud Computing c'est pas l'Internet, l'Internet reste un outil.

Le nouveau, et le brillant dans le Cloud Computing c'est la virtualization, et c'est le principe du Cloud. Il faut remonter en 1960, avec les travaux de l'Américain **John McCarthy** (1927-2011), un des pionniers de l'intelligence artificielle qui considérait d'emblée l'informatique comme un service, mais cette idée n'a pas vu la lumière jusqu'aux années 2000 avec **Amazon**, site de commerce électronique de dimension mondiale, qui a trouvé dans le Cloud Computing une solution élégante à la sous-utilisation de son parc de serveurs informatiques dans les périodes en dehors des fêtes qui représentent en termes de commandes un pic temporel ponctuel d'utilisation [3],

Any jassy à été le premier qui a proposé ce type de service en 2006 « Amazon web service ». ce qui confirme ce qu'a dit **John McCarthy** « le Cloud Computing peut être un jour organisé comme une entreprise d'utilité publique (comme les services publics, tel que l'eau et l'électricité).

1.4 Les principes de Cloud Computing

Un Cloud est un ensemble de matériels, de raccordements réseau et de logiciels qui permettent de fournir des services que les individus et les collectivités peuvent exploiter depuis n'importe où dans le monde et à n'importe quel moment. Le Cloud computing est un basculement de tendance : au lieu d'obtenir de la puissance de calcul par acquisition de matériel et de logiciel, le consommateur se sert de puissance mise à sa disposition par un fournisseur via Internet.

Le Cloud utilise des technologies telles que la « virtualisation du matériel informatique » et les service web.

Exemple :

Un exemple de service qui a un grand-public, basé sur le Cloud Computing, est le jeu à la demande (« gaming on demand (GoD) ou Cloud gaming »). L'utilisateur peut jouer à des jeux vidéo sur son ordinateur, alors que le(s) logiciel(s) du jeu tournent sur des

serveurs à distance, qui renvoient la vidéo de ce qui a été joué en lecture en continu (« streaming»). Le jeu est hébergé et stocké sur des serveurs, dont l'utilisateur ne connaît pas la localisation ni les caractéristiques. Il n'a pas besoin de supports, comme les CD, ou de matériel. Les joueurs doivent seulement posséder un ordinateur relié à Internet.

1.5 Les avantages et les limites du Cloud Computing

Le Cloud doit avoir cinq propriétés, d'après le NIST « National Institute of Standards Technology » et qui peuvent être considérés comme des avantages.

1.5.1 Les avantages :

1-« On demand self service » : Ce qui signifie en français ' Self service à la demande c'est-à-dire Avec un simple clic on peut accéder à des ressources et les utiliser, avec le Cloud Computing on peut minimiser le temps pour accéder à la dernière technologie .

2-« Broad Network Access » Ce qui signifie en français 'Accès au réseau élargi ' c'est à dire je peux accéder aux sources par n'importe quel moyen ; ordinateur, tablette , téléphone ...

3-« Ressources pooling » : Ce qui signifie en français ' mutualisation des ressources ' c'est à dire Les ressources sont toutes agrégées puis divisées en fonction de la demande et de l'utilisation .

4-« Rapid Elasticity » : Ce qui signifie en français 'élasticité rapide ', c'est à dire Je peux augmenter et réduire la quantité de ressources nécessaires facilement et sans effort.

5- « Measured services » : Ce qui signifie en français 'Services mesurés ' c'est à dire La chose la plus importante est que je paye en fonction de ce que j'ai utilisé et donc les services doivent être mesurés.

1.5.2 Les limites : Chaque nouveauté dans la technologie à des avantages et des inconvénients, mais ce que les gens craignent dans le Cloud c'est :

1- Privacy : Ce qui signifie ' le secret ', Toutes vos données et vos activités seront transportées dans un endroit étranger possible sur lequel vous n'avez aucun contrôle. Il est arrivé qu'un groupe de pirates pénètre le Cloud et publié des photos de personnalités célèbres .

2- Il n'y a pas de consolidation dans le travail : Chaque entreprise a un moyen d'adopter et de faire fonctionner ses services et donc il est possible d'arriver à ce qu'on appelle « Vendor Locking » : Cela signifie que vous resterez dans une certaine entreprise parce que vous ne pourrez pas transférer vos données à une deuxième entreprise parce qu'elle ne sera pas en mesure de comprendre comment la première fonctionne.

3-la sécurité : Dans le monde informatique il n'y a pas de sécurité total, la grande vitesse des service de Cloud peut risquer de perde des données . L'absence d'un service pendant quelques minutes peut signifier la perte de plusieurs millions de dollars pour une organisation. L'accès aux bases de données des informations qui transitent dans les nuages lorsqu'elles se retrouvent dans les ordinateurs des fournisseurs de services est encore plus inquiétant car s'il est difficile de s'introduire dans les systèmes virtuels de l'informatique dématérialisée, il peut être beaucoup plus facile de le faire une fois que l'information n'est plus dans les nuages.

4- La disponibilité : Les données sont stockées en dehors du réseau de l'entreprise, peut-être même à l'étranger, ce qui peut enfreindre la réglementation et les lois de votre pays en matière de protection des données, il n'y a pas de loi qui protège vos données dans le cas où il y a des problème politique par exemple entre votre pays et le pays où vous stockez vos données .

5- Intégrité des données : Le recours à un prestataire de Cloud Computing crée un risque d'atteinte à l'intégrité globale du système d'information en raison de la perte d'expertise technique, voire de dépendance au fournisseur.[4]

1.6 Les modèles de services de Cloud Computing



Figure1-2 Schéma donnant un aperçu sur les facteurs principaux du cloud computing .

1.6.1 SaaS « *Software as a service* » : Il s'agit d'utiliser une application spécifique stockée sur le Cloud, par exemple un programme Word existant dans un centre de données et de s'y connecter via Internet pour écrire, modifier, ajouter des données et en obtenir la sortie, et tout cela sur le Cloud et votre appareil n'est qu'un outil de communication. L'utilisateur ne peut pas contrôler le système d'exploitation dans le Cloud et ne contrôle pas la connectivité matérielle ou réseau. YouTube peut être considéré dans cette catégorie, car le navigateur vidéo sur site est l'application chargée sur le nuage et à travers laquelle vous pouvez accéder aux vidéos existantes, mais vous ne pouvez rien changer sur le site.

1.6.2 PaaS « *Platform as a service* » : Utiliser le Cloud comme plate-forme pour y placer plusieurs applications sur les quelles vous pouvez travailler. Vous pouvez également configurer un système d'exploitation complet et il existe une intégration entre les applications.

Les applications Google sont basées sur une plate-forme qui vous permet d'ajouter des applications comme vous le souhaitez.

1.6.3 IaaS « *Infrastructure as a service* » : C'est un modèle où l'utilisateur dispose d'une infrastructure informatique (des serveurs, des réseaux, des espaces de stockage ou des espaces au sein de Data Centers) qui se trouve physiquement chez le fournisseur. Le stockage peut être sur un serveur virtuel, mais les utilisateurs peuvent également transférer leurs charges de travail vers des machines virtuelles (VM). Les IaaS sont

flexible et scalables pour s'adapter à la charge de travail. Les utilisateurs disposent d'un espace de stockage allouée. Ils peuvent démarrer, arrêter ou configurer la machine virtuelle et le stockage. Les infrastructures fournies peuvent être petites, moyennes, grandes ou très grandes pour répondre aux différents besoins. Avec cette infrastructure, les entreprises n'ont pas besoin d'investir dans leur propre matériel.

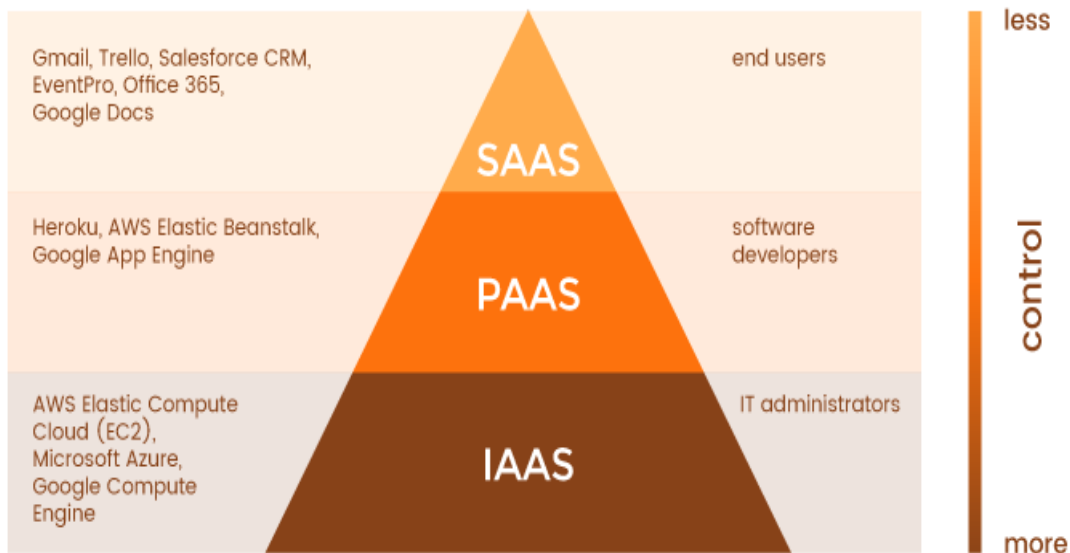


Figure1-3 SAAS –PAAS-IAAS [5].

1.7 Les modèles de déploiement de Cloud Computing

Le Cloud Computing s'impose peu à peu dans les entreprises avec une promesse de baisse des coûts, de gains en agilité, mais aussi ses contraintes. À chaque entreprise de choisir le type de Cloud qui correspond le mieux à ses besoins.

Selon le déploiement de Cloud Computing on distingue 3 types.



Figure1-4 Cloud Computing Public, Privé ou Hybride [6]

1.7.1 Le Cloud public « *Public Cloud Computing* » : Est une infrastructure qui fournit dynamiquement des ressources informatiques sur Internet à plusieurs clients, et différentes applications client sont généralement mélangées sur des serveurs Cloud. La vitesse dans ce modèle est très grande, l'accès est meilleur, mais il y a moins de confidentialité donc les données sensibles ne peuvent pas être stockées sur ce genre de Cloud, les entreprises qui sont basées sur la confidentialité ne peuvent pas utiliser le Cloud public, la défense par exemple.

1.7.2 Le Cloud privé « *Private Cloud Computing* » : C'est une infrastructure louée par un seul client et il travaille pour son propre compte avec un contrôle total des données, la sécurité et la qualité du service. C'est la solution qui, sur le papier, répond le mieux aux attentes des entreprises qui refusent de voir leurs données hébergées chez des autres peut être dans un autre pays, cela implique que l'entreprise soit propriétaire de ses serveurs et équipements réseaux. L'entreprise a un contrôle total de son infrastructure serveur, réseau ainsi que sur la sécurité des données. Ce modèle est nécessaire pour certains domaines comme la défense.

1.7.3 Le Cloud hybride « *Hybrid Cloud Computing* » : Combine plusieurs modèles du Cloud public et privé. Le Cloud hybride montre le processus complexe pour déterminer comment les applications sont distribuées à travers les nuages privés et publics. Dans ce modèle le Cloud est divisé en deux privé et public, par exemple la partie de développement peut se mettre sur le « Private Cloud » et la partie application sur « Public Cloud » parce qu'il contient plus de ressources, infrastructure plus grande, très grande vitesse et un meilleur accès. C'est une solution qui combine entre la rapidité de « Public Cloud » et la confidentialité de « Private Cloud ».

2 Deployment Models



Figure 1-5 Modèles de déploiement de cloud (*private –public*) [7]

1.8 Conclusion

Où allons nous ? À quoi s'attendre avec cette technologie ?

Le cloud se développe d'une manière croissante et rapide. Dans ce cadre il est possible que dans les prochaines années on entendra de moins en moins parler d'ordinateurs avec de grande capacité de traitement puisque le tout se fera sur le cloud.

L'ordonnancement des tâches et l'allocation des ressources

Sommaire

2.1 Introduction

2.2 Ordonnancement : Concepts et définitions

2.3 Les problèmes d'ordonnancement en ligne et hors ligne

2.4 Les critères d'optimisation

2.5 L'ordonnancement et la virtualisation dans le cloud computing

2.6 Les principaux algorithmes d'ordonnancement

2.7 Conclusion

2.1 Introduction

Récemment, il y a eu une augmentation spectaculaire de la popularité des systèmes de cloud computing qui louent des ressources informatiques à la demande, un ensemble virtuel de ressources qui est fourni aux utilisateurs sur Internet. Il fournit aux utilisateurs des ressources informatiques illimitées sans avoir à gérer l'infrastructure de base. L'un des principaux objectifs consiste à utiliser efficacement les ressources et à en tirer le meilleur parti. Parce que le fournisseur de services cloud doit servir de nombreux utilisateurs dans le système informatique en nuage. Ainsi, la planification est le principal problème dans la création de systèmes de cloud computing. Les algorithmes de planification doivent fonctionner de manière à équilibrer les performances avec la qualité de service tout en préservant l'efficacité et l'équité entre les fonctions. Dans ce chapitre nous visons à étudier différentes méthodes d'ordonnancement « *Scheduling* ». Une bonne technologie d'ordonnancement aide également à utiliser correctement et efficacement les ressources. Beaucoup de techniques d'ordonnancement ont été mis au

point par des chercheurs tels que GA (algorithme génétique), PSO (optimisation de la rotation des particules), Min- Min, Max-Min, l'algorithme Planificateur de tâches sur une base prioritaire [8].

Ce chapitre passe en revue certaines des méthodes de gestion des ressources et de planification des tâches dans le cloud computing.

La planification fait référence à la stratégie de contrôle de l'ordre de travail à exécuter par un système informatique. L'objectif des algorithmes d'ordonnement dans le cloud computing minimise la charge sur les processeurs et maximise leur utilisation tout en minimisant le temps d'exécution total de la tâche. La planification des tâches joue un rôle important pour améliorer la puissance potentielle des systèmes. L'objectif principal est d'assigner des tâches aux ressources allouées en fonction du temps, ce qui implique de découvrir un ordre approprié dans lequel les tâches peuvent être exécutées sous des contraintes de logique de transaction. Il existe différents types d'algorithmes d'ordonnement dans le système informatique en nuage, et l'ordonnement des tâches en fait partie. Le principal avantage de l'ordonnement des tâches est d'obtenir une haute performance du système informatique qui conduit au meilleur débit du système. L'ordonnement régule la disponibilité des ressources et une bonne politique d'ordonnement permet une utilisation maximale des ressources informatiques.

Il existe deux modes de planification des tâches

a-Space Shared (Espace partagé) : C'est l'exécution des tâches l'une après l'autre (détaillé plus tard).

b- Time Shared (Temps partagé) : C'est l'exécution des tâches en même temps (détaillé plus tard).

2.2 Ordonnement : Concepts et définitions

L'ordonnement est l'organisation de l'exécution des tâches dans le temps, en tenant compte des contraintes de temps (contraintes de temps, contraintes d'enchaînement, ...), et des contraintes sur l'utilisation et la disponibilité des ressources requises [18] .

Ordonnancement : Un problème d'ordonnancement consiste à ordonner dans le temps un ensemble des tâches contribuant à la réalisation d'un même projet, l'objectif est de déterminer la durée de réalisation du projet.

Les algorithmes d'ordonnancement jouent un rôle important lorsque l'objectif est de planifier efficacement les tâches afin de réduire le temps de réponse et améliorer l'utilisation des ressources.

Allocation des ressources : L'allocation des ressources est la répartition idéale des ressources disponibles pour répondre au mieux aux besoins (beaucoup de demandes et moins de ressources), ce qui implique que des choix doivent être faits. Ces choix et ces décisions sont les processus d'allocations des ressources.

Dans le cloud computing, si la répartition et l'attribution de ces ressources n'est pas contrôlée dans le bon sens, cela perturbera la bonne conduite des opérations.

L'allocation optimale des ressources est essentielle pour accélérer la mise en œuvre des opérations et répondre au maximum des demandes.

Les ressources : une ressource est un moyen technique ou humaine utilisé pour effectuer une tâche, mais disponible en quantité et en capacité limitées, et il existe plusieurs types de ressources. La ressource est renouvelable si elle est réaffectée avec la même quantité après son utilisation (hommes, machines et matériel en général) mais la quantité de ressources utilisables à tout moment est limitée à chaque instant, la ressource peut être consommable (si elle n'est pas renouvelable exp : budget).

La disponibilité des ressources peut changer au fil du temps, et il est généralement connu que la courbe de disponibilité est connue à l'avance, sauf si elle dépend de l'état de certaines tâches de génération, il y a des ressources qui ne peuvent servir qu'une seule tâche.

Les contraintes : Les contraintes expriment des restrictions sur les valeurs que peuvent prendre simultanément les variables de décision. On distingue :

- Des contraintes temporelles concernant les contraintes de temps alloué, issues généralement d'impératifs de gestion et relatives aux dates limites des tâches (délai de livraisons, disponibilité des approvisionnements) ou à la durée total d'un projet et les contraintes de gammes qui décrivent des relations d'ordre relatif entre les différents tâches.

- Des contraintes de ressources décrivent des contraintes d'utilisation de ressources qui expriment la nature et la quantité des moyens utilisés par les tâches, ainsi que les caractéristiques d'utilisation de ces moyens et les contraintes de disponibilité des ressources qui précisent la nature et la quantité des moyens disponibles au cours du temps. Toutes ces contraintes peuvent être formalisées sur la base des distances entre débuts des tâches.

L'équilibrage de la charge (load balancing) : L'équilibrage de charge est une technique relativement nouvelle qui facilite l'exécution des tâches entre les ressources en fournissant une productivité maximale avec un temps de réponse minimal. Le trafic entre serveurs peut envoyer des champs de données et recevoir sans trop de retard. Différents types d'algorithmes sont disponibles. Sans équilibrage de charge, les utilisateurs peuvent être sujets à des retards, des temps de latence et des réponses système potentielles, et des solutions d'équilibrage de charge sont appliquées aux serveurs redondants qui permettent une meilleure distribution du trafic de communication.

2.3 Les problèmes d'ordonnancement en ligne et hors ligne

L'objectif de l'ordonnancement est d'exécuter d'une façon optimale les tâches en prenant en considération leurs contraintes. un problème est constitué de : tâches, ressources, contraintes et objectif.

On distingue deux catégories de problème :

- 1- Les problèmes d'ordonnancement en ligne (*online*) : dans ce cas la nous ne pouvons pas connaître la date d'arrivée des taches à l'avance .
- 2- Les problèmes d'ordonnancement hors ligne (*offline*) : contrairement à l'ordonnancement *online*, dans l'ordonnancement *offline* la date d'arrivée des taches (jobs) est connue à l'avance (les caractéristiques des taches sont connues avant l'ordonnancement , en général toutes les taches sont prêts à $t=0$) .

L'ordonnancement *online* est plus compliqué si on le compare avec l'ordonnancement *offline* car on a pas toutes les informations sur les taches (manque des donnée) donc il est difficile de savoir l'emplacement des tâches dans l'exécution de l'ordonnancement online (on ne peut prévoir l'avenir).

Chaque processus d'ordonnancement doit passer par les trois étapes suivantes :

- **La phase *task prioritizing*** : Dans cette phase les tâches sont ordonnées selon leurs contraintes et leurs propriétés pour avoir à la fin une liste bien ordonnée.
- **La phase *resource provisioning/allocation*** : Réserve et Calcule le nombre de ressources (machines virtuelles) qui seront alloués pour l'ordonnancement.
- **La phase *scheduling/mapping*** : Sélectionne les ressources parmi celles précédemment allouer qui permettent d'exécuter les taches selon l'ordre prédéfini d'une manière optimal .

2.4 Les critères d'optimisation :

Dans l'environnement de cloud la sélection des ressources se fait de plusieurs manières comme Round Robin qui est aléatoire, ou d'une manière gourmande selon le temps d'exécution et les capacités de traitement des ressources. Il existe d'autres manières selon des critères différents.

La sélection des tâches se fait par priorité selon le temps d'arrivée « *arrival time* » FCFS (First Come First Served) ou la longueur de tache « *length* » SJF (Short Job First).

Avec les stratégies d'ordonnancement on choisit la tâche à exécuter et la machine adéquate pour cette tâche, en améliorant avec ces stratégies certains objectifs (minimiser le temps d'exécution , minimiser le temps d'attente , respecter le délai). chaque stratégie a ses avantages.

L'objectif de suivre un algorithme d'ordonnancement pour un problème précis c'est d'améliorer et optimiser certains critères, et chaque algorithme a pour but d'optimiser certains critères. On ne peut pas dire qu'il y a un critère d'optimisation universel, mais le choix de critère d'optimisation est très important dans la qualité de travail, parmi les critères d'optimisations les plus utilisé on distingue :

***Makespan* (C_{max})** : c'est le temps de terminaisons de la dernière tâche, correspond au temps passé par le système pour terminer tout son travail (le temps global pour exécuter toutes les tâches).

Flow time (F_i) : Dans le cloud les ressources sont limitées et quand les virtuelles machines sont toutes occupées les tâches attendent dans une files d'attente. Le problème dans ce cas c'est le temps d'attente, c'est pour ça qu'on peut considérer le temps d'attente moyen *Flow time* (F_i) comme un critère d'optimisation et des algorithmes sont utilisés pour le minimiser ($F_i = C_i - a_i / a_i$: c'est le temps d'arrivé du job_i, C_i : la fin de l'exécution de job_i), mais l'inconvénient c'est que la minimisation de temps d'attente moyen implique l'augmentation du temps d'attente des petits jobs cette optimisation conduit à des résultats indésirables. [19]

Le débit : c'est mieux d'utiliser le débit comme critère d'optimisation (fraction de tâche par unité de temps i.e second).

2.5 L'ordonnancement dans le cloud computing

L'ordonnancement dans le cloud computing est basé sur deux niveaux, niveau utilisateurs et niveau système [15].

Au niveau utilisateur : l'ordonnancement gère les problèmes des services entre le fournisseur et le client .

Au niveau système : le *Datacenters* est constitué de plusieurs *Host* (machines physiques) mais les demandes sont illimitées , des millions de tâches sont prêtes à être exécutées par ces machines dans des délais prédéfinis.

L'attribution de ces tâches aux ressources à une grande importance et un très grand effet sur l'amélioration de performance du Datacenter.

Les ordonnancements jouent un rôle important pour plusieurs exigences qui doivent être pris en compte tel que la satisfaction du temps, la qualité de service , et SLA (*service level argument*)... .

L'ordonnancement dans le cloud computing est pour réguler l'offre à la demande, les ressources dans l'environnement de cloud sont virtuelles et livrées en tant que service à l'utilisateur, plusieurs algorithmes d'ordonnancement sont proposés [17] .

Il a une nécessité de planification prenant en considération des paramètres tel que SLA qui représente un contrat entre le client et l'utilisateur du cloud pour l'allocation optimale des ressources. A l'échelle mondiale et dans le but d'équilibrer la charge dans

le cloud, une approche d'ordonnement existe sur chaque serveur dans chaque centre de données [13].

Les ressources dans le cloud sont représentées sous forme de machines virtuelles VMs et l'utilisateur peut augmenter ou minimiser les capacités de ces VMs [16].

2.6 Les principaux algorithmes d'ordonnement :

L'objectif principal des algorithmes d'ordonnement est de planifier des tâches sur des machines virtuelles (VM) en fonction du temps d'adaptation, ce qui implique de trouver une séquence appropriée dans laquelle les tâches peuvent être effectuées sous les contraintes de la logique de transaction [9]. La planification des tâches dans le cloud computing est un défi. Pour relever ce défi, nous passons en revue un nombre d'algorithmes de planification des tâches.

L'ordonnement des tâches de l'infonuagique consiste à répartir les tâches de calcul sur le regroupement des ressources entre différents utilisateurs de ressources en fonction de certaines règles d'utilisation des ressources dans des conditions de cloud données. À l'heure actuelle, il n'existe pas de norme uniforme pour la planification des tâches dans le cloud computing. La gestion des ressources et l'ordonnement des tâches sont les techniques clés de l'informatique en nuage qui jouent un rôle essentiel dans une gestion efficace des ressources du cloud.

Les algorithmes d'ordonnement de tâches et d'allocation des ressources suivants sont actuellement évoqués dans le contexte du cloud.

Algorithme Min-Min : Min-Min commence par un ensemble de tâches qui sont toutes non assignées. Tout d'abord, il calcule le temps d'achèvement minimum pour toutes les tâches sur toutes les ressources. Puis, parmi ces temps minimum, la valeur minimale est sélectionnée, qui est le temps minimum entre toutes les tâches sur toutes les ressources. Ensuite, cette tâche est planifiée sur la ressource sur laquelle elle prend le moins de temps et le temps disponible de cette ressource est mise à jour pour toutes les autres tâches. Il est mis à jour de cette manière. Supposons qu'une tâche est affectée à une machine et que la tâche affectée dure 10 secondes, alors les temps d'exécution de toutes les autres tâches sur cette machine assignée seront augmentés de 10 secondes.

Après cela, la tâche assignée n'est pas considérée et le même processus est répété jusqu'à ce que toutes les tâches aient des ressources affectées.

Un exemple d'application de l'algorithme pour 6 tâches et 4 machines virtuelles, le tableau 2.1 présente les temps d'exécution (en milliseconde) des tâches sur les machines virtuelles [11] .

	M ₀	M ₁	M ₂	M ₃
T ₀	200	250	220	300
T ₁	150	170	190	160
T ₂	300	320	180	360
T ₃	400	380	350	310
T ₄	100	120	140	160
T ₅	220	250	280	200

Table 2-1 – Le temps d'exécution des tâches.

En utilisant différentes techniques d'ordonnancement, les tâches sont assignées dans une séquence différente à des machines différentes pour l'exécution. Lorsque nous appliquons la technique d'ordonnancement Min-Min, les tâches seront assignées aux machines comme indiqué dans la figure suivante:

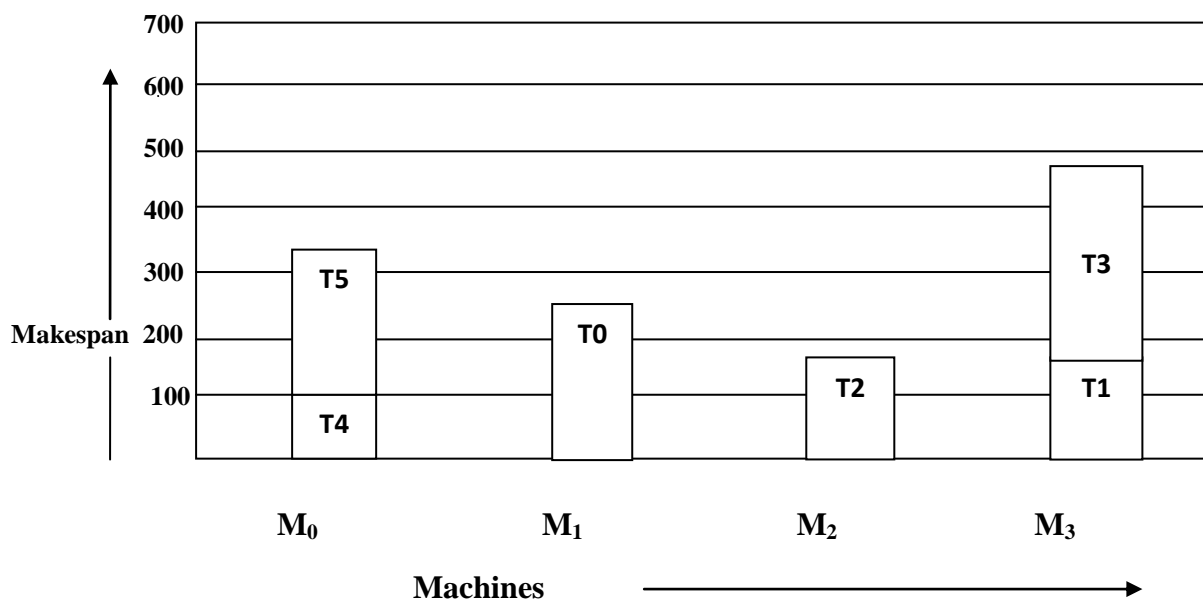


Figure 2.1 – Affectation des tâches par l'algorithme Min-Min

Algorithme Max-Min : Max-Min suit le même principe que l'algorithme Min-Min excepté ce qui suit : dans ce cas, après avoir trouvé le temps de fin, les temps d'exécution minimum sont trouvés pour chaque tâche. Puis, parmi ces temps minimum, la valeur maximale est sélectionnée, qui est le temps maximum entre toutes les tâches

sur toutes les ressources. Ensuite, cette tâche est planifiée sur la ressource sur laquelle elle prend le moins de temps, puis le temps d'exécution de toutes les autres tâches sont mise à jour sur cette machine. La mise à jour se fait de la même manière que pour le Min-Min. Toutes les tâches sont affectées aux ressources par cette procédure.

Si nous appliquons la technique d'ordonnancement Max-Min, les tâches seront assignées aux machines comme indiqué dans la figure suivante:

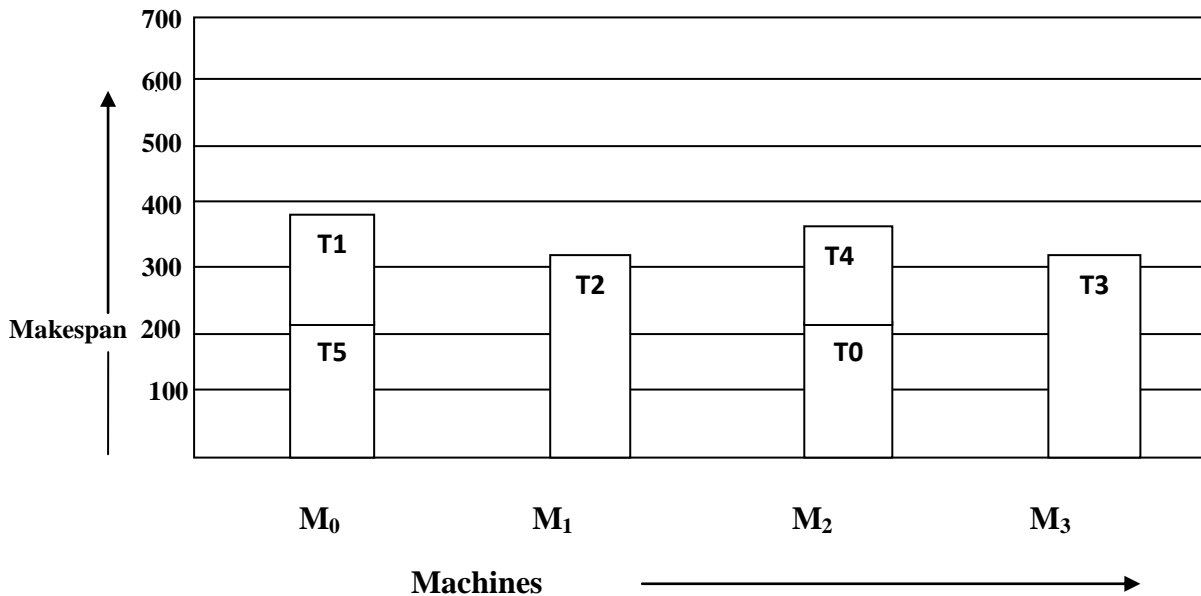


Figure 2-2 – Affectation des tâches par l'algorithme Max-Min

Le terme "*makespan*" dans les techniques d'ordonnancement Min-Min et Max-Min, comme expliquer auparavant est le temps d'exécution maximum sur n'importe quelle machine parmi les machines sur lesquelles les tâches sont planifiées. Par exemple, dans la Figure 2-1 "470" est le rendu parce que c'est le temps d'exécution maximal entre les quatre machines.

Sur la figure 2-1 et la figure 2-2, l'axe des abscisses représente les différentes machines et l'axe des y représente les temps d'exécution. Nous avons obtenu les différentes valeurs de *makespan* suivantes pour les deux techniques : Min-Min 470
Max-Min 370.

En fonction des différents temps d'exécution des tâches sur les ressources, une technique peut surpasser l'autre et l'affectation des ressources aux tâches peut changer, c'est-à-dire si une tâche est affectée à une machine si nous utilisons une technique, la

même tâche peut être assignée à une autre machine si nous utilisons une autre technique.

Algorithme Round Robin : Cet algorithme suit une stratégie simple qui consiste à distribuer de manière équitable les tâches sur les machines virtuelles disponibles, c'est – à-dire que le nombre de tâches pour chaque machine virtuelle est le même. Cet algorithme est implémenté dans le simulateur CloudSim [14].

L'algorithme Round Robin se concentre principalement sur la distribution de la charge de manière égale à toutes les ressources.

Le résultat de l'algorithme Round Robin montre un meilleur temps de réponse et un meilleur équilibrage de charge [12].

Algorithme FCFS : L'algorithme FCFS "*First Come First Served* " est tout simplement la politique FIFO "*First in First out*" c'est l'un des algorithmes les plus simples, la stratégie c'est d'ajouter chaque tâche et ressource dans une file et d'exécuter chaque tâche et ressource par ordre d'arrivée , cet algorithme est implémenté dans le simulateur CloudSim.

Algorithme SJF : l'algorithme SJF « *Short Job First* » exécute les *jobs* selon l'ordre du temps d'exécution le plus court c'est-à-dire il choisit d'exécuter en premier la tâche qui sera la plus courte.

Earliest deadline First scheduling : ("échéance proche = préparation en premier") est un algorithme d'ordonnancement préemptif, à priorité dynamique, utilisé dans les systèmes temps réel. Il attribue une priorité à chaque requête en fonction de l'échéance de cette dernière selon la règle : Plus l'échéance d'une tâche est proche, plus sa priorité est grande. De cette manière, au plus vite le travail doit être réalisé, plus il a de chances d'être exécuté.

Cet algorithme est optimal pour tous types de système de tâches, cependant, il est assez difficile à mettre en œuvre et est donc peu utilisé. De plus, il ne prévoit aucun compromis "satisfaisant" en cas de surcharge du système. [R]

2.7 Conclusion

La planification des tâches et la technique de l'allocation des ressources dans les systèmes de cloud computing suscitent un intérêt croissant à mesure que la popularité du cloud augmente. En général, l'ordonnancement des tâches consiste à affecter des

tâches aux ressources disponibles en fonction des propriétés et des conditions du travail. C'est un aspect important du fonctionnement efficace du cloud, car il existe différents paramètres de tâche qui doivent être pris en compte pour l'ordonnancement approprié.

implémentation et performance

Sommaire

3.1 Introduction

3.2 Analyse comparative des algorithmes d'ordonnancement de cloud- sim dans le cloud computing .

3.3 Comparaison entre les algorithmes d'ordonnancement des tâches (FCFS, Round-Robin , ShortFirst)

3.4 Comparaison entre les algorithmes d'ordonnancement des tâches et d'allocation des ressources (Max-Min et Min-Min).

3.5 Comparaison entre les algorithmes d'ordonnancement des tâches et d'allocation des ressources (FCFS, Round-Robin , ShortFirst Max-Min et Min-Min).

3.6 Proposition d'un algorithme qui donne un résultat optimal pour le critère makespan .

3.7 Conclusion

3-1 Introduction

Ce chapitre est consacré à la phase d'implémentation des stratégies d'ordonnancement et d'allocation des ressources proposées. Il permettra d'évaluer et valider nos stratégies proposées dans le Cloud computing. Dans ce but nous avons réalisé plusieurs simulation en utilisons le moyen de simulation « Cloud-sim » . Pour effectuer des expérimentations dont les résultats et les interprétations font l'objet de ce chapitre .

- La plate forme

Le **micro-ordinateur** que j'ai utilisé c'est un micro-ordinateur avec :

Un processeur : Intel (R) Pentium (R) CPU N3540 @ 2.16 GHZ .

Mémoire (RAM) : 4.00 Go (3.89 Go utilisable)

Type de système : Système d'exploitation 64 bits .

Le moyen de simulation c'est le cloud-sim cloudsim-3.0.3

3.2 Analyse comparative des algorithmes d'ordonnancement de cloud-sim dans le cloud computing .

Il existe deux politiques qui sont définies (implémentés) dans le Cloud-Sim

- La politique d'ordonnancement Space-Shared (Espace partagé)
- La politique d'ordonnancement time-Shared (Temps partagé)

La première des choses, nous examinons les capacités pratiques de Space-Sahred et Time-Shared. En compare les performances des algorithmes d'ordonnancement Space-Sahred et Time-Shared de cloud-sim (*scheduling algorithm of cloud-sim*) .

A. Scenario

Le but de cette étude est de montrer la capacité de Space-Shared à respecter les délais où Time-Shared ne peut faire de même (*dead-line*). Pour ce faire, nous recommandons ce qui suit. Nous utilisons deux Data-Center identiques composées chacun d'un Host avec une seule Vm. Chaque machine virtuelle a une vitesse de 1000 MIPS. bande with (BW) disponible de 1 Gb/ s (1000 Mb/ s). Une capacité de stockage de 10000 MB. Une mémoire vive (RAM) de 512 Mb. Nous traitons 1000 cloudlets sur les deux data-centers utilisant deux stratégies : Space-Shared et Time-Shared . Pour chaque cloudlet on donne une date limite générée au hasard où la valeur minimale est la moitié du nombre de cloudlets (500) et La valeur maximale est égale au nombre de cloudlets (1000).

B. Task Profit (Critère « Dead-line »)

Le nombre des tâches terminées avec succès avant le délai. Space-Shared est plus efficace pour terminer les tâches avant l'arrivée de l'échéance que l'algorithme

d'ordonnement à Time-Shared. Nous comparons le nombre de tâches accomplies par chaque algorithme. Dans la figure 1, Nous pouvons voir les résultats de l'expérience. Sur l'axe des abscisses, est indiqué le nombre des tâches et l'axe des y indique le nombre de tâches terminées avec succès avant que l'échéance ne soit atteinte. En examinant la figure, nous pouvons dire que l'algorithme Time-Shared est inefficace pour respecter le délai « **Deadline** ». Nous pouvons remarquer que, dans ce cas, les deux approches semblent être indiscernables, mais Space-Shared complète presque toutes les tâches avant le « **Deadline** ». En augmentant le nombre des tâches et des machines virtuelles, nous pouvons observer une plus grande différence et conclure que Space-Shared est meilleur que Time-Shared .

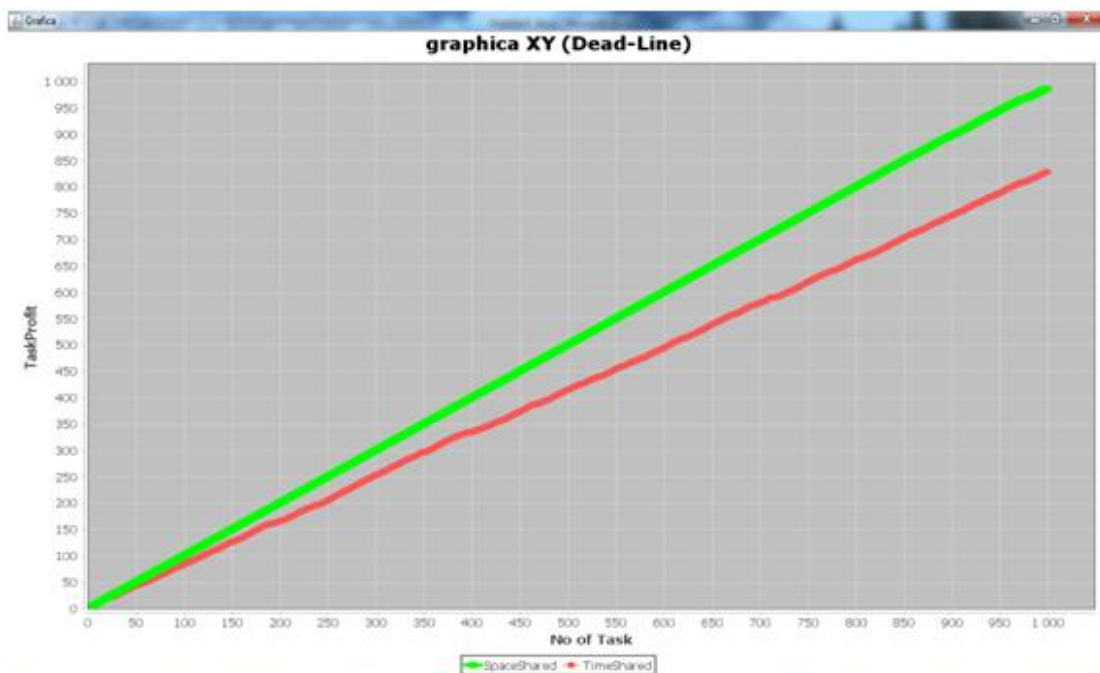


Figure 1 : Comparaison sur la base du profit de la tâche avec les deux politique (Space-Shared, Time-Shared)

3.3 Comparaison entre les algorithmes d'ordonnement des tâches (FCFS, Round-Robin, ShortFirst)

A. Scenario

Le but de cette étude est de montrer quel est l'algorithme d'ordonnement

Qui respecte le délai (*dead-line*) d'une manière efficace. Nous utilisons trois (3) Data-Centers identiques composé chacun d'un Host avec une seule Vm. Chaque machine virtuelle a une vitesse de 1000 MIPS. bande with (BW) disponible de 1Gb/s (1000 Mb/s). Une capacité de stockage de 10000 MB. Une mémoire vive (RAM) de 512 Mb. Nous traitons 200 cloudlets sur les trois data-centers utilisant trois Algorithmes d'ordonnancement : *FirstCom-FirstServed (FCFS)*, *Roud-Robin* et *Short-First* (*Short-First n'est pas implémenté dans Cloud-Sim , j'ai fait son implémentations*). Pour chaque cloudlet on donne une date limite générée au hasard où la valeur minimale est 0 et La valeur maximale est 100.

B. Task Profit (Critère « Dead-line »)

Si on prend en considération le nombre des tâches terminées avec succès avant le délai (critère : « **Deadline** »), l'algorithme d'ordonnancement *ShortFirst* est plus efficace pour terminer les tâches avant l'arrivée de l'échéance que l'algorithme d'ordonnancement *FCFS*, et *FCFS* est meilleur que l'algorithme d'ordonnancement *Round-Robin*. Nous comparons le nombre de tâches accomplies par chaque algorithme. Dans la figure 2. Nous pouvons voir les résultats de l'expérience. Sur l'axe des abscisses, il indique le nombre des taches et l'axe des y indique le nombre de tâches terminées avec succès avant que l'échéance ne soit atteinte. En examinant la figure, nous pouvons dire que l'algorithme *ShortFirst* est le plus efficace parmi ces algorithmes pour respecter le « **Deadline** ». En augmentant le nombre des tâches nous pouvons remarquer une plus grande différence et conclure que *ShortFirst* est meilleur que *FCFS* et *FCFS* est meilleur que **RounRobin**.

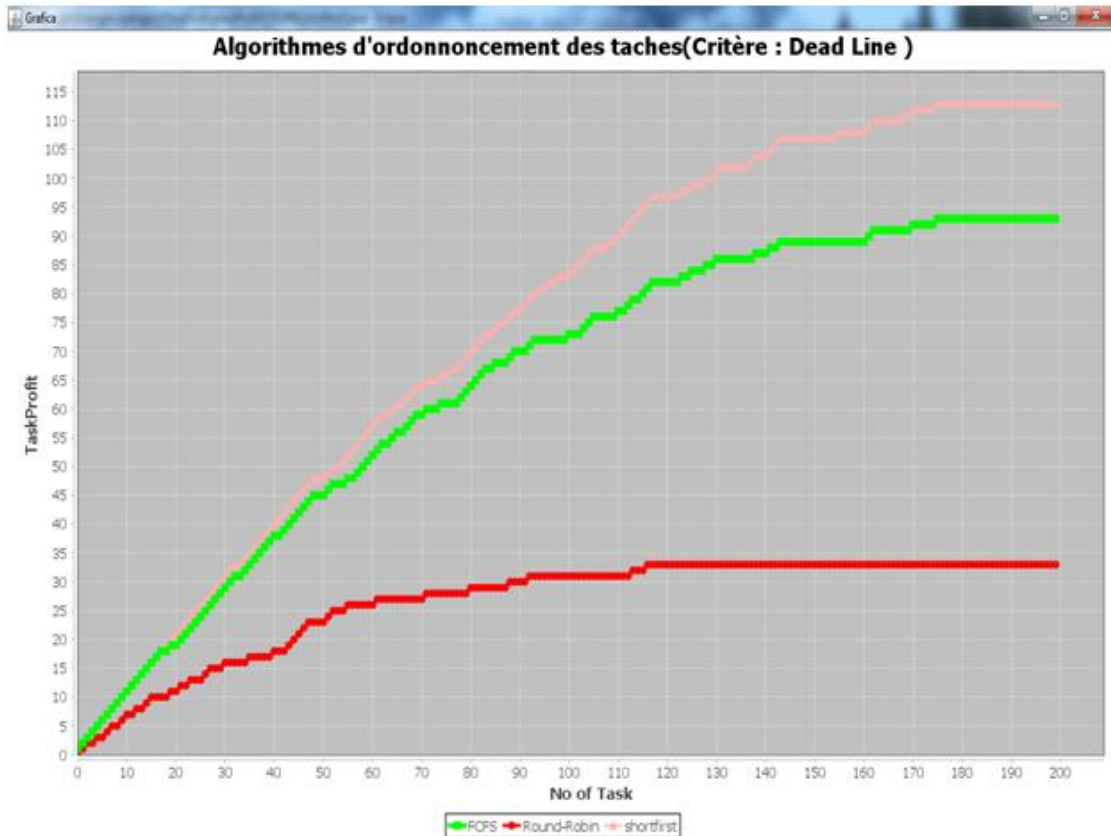


Figure2 : Comparaison sur la base du profit de la tâche avec les 3 algorithmes (ShortFirst , FCFS ,Round-Robin)

3.4 Comparaison entre les algorithmes d'ordonnancement des tâches et d'allocation des ressources (Max-Min et Min-Min)

A. Scenario

Le but de cette étude est de montrer quel est l'algorithme d'ordonnancement

Qui réalise le « **MakeSpan** » minimal .

MakeSpan : C'est un critère d'optimisation très utilisé, il représente le temps de terminaison de la dernière tâches de graphe. Il s'agit du temps de complétion maximale noté C_{max} qui correspond au temps passé par le système pour réaliser son travail.

Nous utilisons deux (2) Data-Centers identiques composée chacun d'un Host avec 3 Vms avec les même caractéristiques mais avec des vitesses différentes.

Vm	La vitesse (MIPS)	Size	RAM	BW
Vm0	250	10000 MB	512 MB	1000 Mb/S
Vm1	200	10000 MB	512 MB	1000 Mb/S

Vm2	500	10000 MB	512 MB	1000 Mb/S
-----	-----	----------	--------	-----------

Nous traitons 7 cloudlets avec des longueurs différents (*length*).

Cloudlets	Cloudlet0	Cloudlet1	Cloudlet2	Cloudlet3	Cloudlets4	Cloudlets5	Cloudlets6
Length /million instruction MI	1500	4000	2500	5000	3000	1000	3500

Nous traitons ces *cloudlets* sur les deux data-centers utilisant deux Algorithmes d'ordonnancement : *Max-Min* et *Min-Min* (*Remarque : Max-Min et Min-Min ne sont pas implémenté dans Cloud-Sim , j'ai fait leur implémentations*).

Après l'exécution de la simulation avec ces deux algorithmes *Max-Min* et *Min-Min* les resultats suivants sont affichés.

- 1- Dans l'algorithmes *Min-Min* chaque Virtuelle machine exécute en premier la cloudlets la plus petite. ($Time = Legth/Mips$ le Mips est fixe pour chaque Vm cela veux dire : $Time \text{ est petit} \Rightarrow length \text{ est petit}$).

Vm id	Cloudlet Id	Time	StarTime	FinshTime
0	2	10	0.1	10.1
1	6	17.5	0.1	17.6
2	5	2	0.1	2.1
	0	3	2.1	5.1
	4	6	5.1	11.1
	1	8	11.1	19.1
	3	10	19.1	29.1

Table1 .L'attribution des cloudlets aux ressources

On remarque que dans l'algorithmes *Min-Min* Vm2 exécute la tache 5 ensuite tache0, tache4, tache1 et à la fin tache3 selon l'ordre croisent de leurs longueurs (1000 ,1500,3000,4000,5000 en MI million instruction).

- 2- Dans l'algorithmes *Max-Min* chaque Virtuelle machine exécute en premier la cloudlets la plus grande dans la longueur . ($Time = Legth/Mips$, le Mips est

fixe pour chaque Vm cela veut dire que : $Time \text{ est grand} \Rightarrow length \text{ est grand}$)

<i>Vm id</i>	<i>Cloudlet Id</i>	<i>Time</i>	<i>StarTime</i>	<i>FinshTime</i>
0	1	16	0.2	16.2
	0	6	16.2	22.2
1	4	15	0.2	15.2
	5	5	15.2	20.2
2	3	10	0.2	10.2
	6	7	10.2	17.2
	2	5	17.2	22.2

Table2 .L'attribution des cloudlets aux ressources

On remarque que dans l'algorithme Max-Min Vm2 exécute la tache 3 ensuite tache 6, et à la fin tache 2 selon l'ordre décroissent de leurs longueurs (5000 ,3500,2500 en MI).

B. MakeSpan

Si on prend en considération le « **MakeSpan** » le temps passé par le système pour réaliser son travail, l'algorithme d'ordonnancement **Max-Min** est plus efficace pour terminer le travail (toutes les tâches) dans une durée minimal que l'algorithme d'ordonnancement **Min-Min**. Dans la figure 4, nous pouvons voir les résultats de l'expérience. Sur l'axe des abscisses, il indique le **FinshTime** et l'axe des y indique le « No of Task » En examinant la figure, nous pouvons dire que l'algorithme **Max-Min** est le plus efficace que **Min-Min** pour réaliser un « **MakeSpan minimal** ». On peut conclure que **Max-Min** est meilleur que **Min-Min** si on prend en compte le critère « **MakeSpan** ».

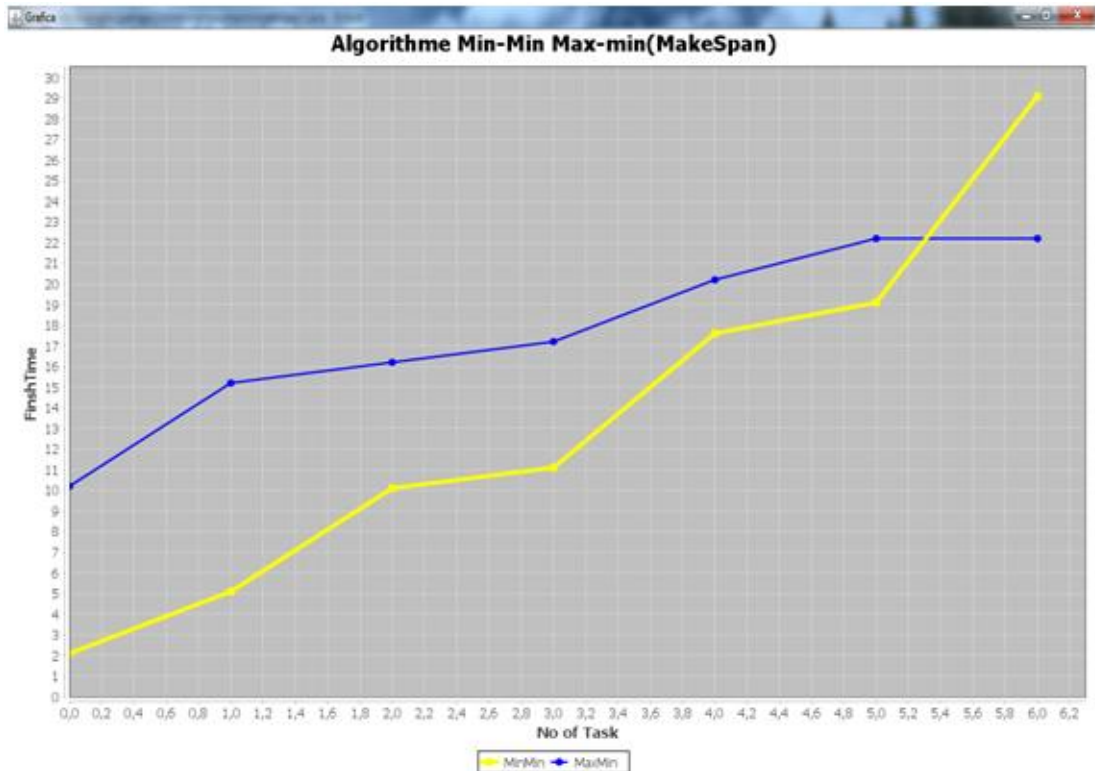


Figure3 : Comparaison selon le critère MakeSpan des deux algorithmes

Max-Min et Min-Min

3.5 Comparaison entre tous les algorithmes d'ordonnancement des taches et d'allocation des ressources (FCFS, Round-Robin , ShortFirst Max-Min et Min-Min)

A. Scenario

Le but de cette étude est de montrer quel est l'algorithme d'ordonnancement parmi tous les algorithmes précédents Qui réalise le « **MakeSpan** » minimal.

Nous utilisons cinq (5) Data-Centers identiques composée chacun d'un Host avec 3 Vms avec les même caractéristiques mais avec des vitesses différentes.

Vm	La vitesse (MIPS)	Size	RAM	BW
Vm0	250	10000 MB	512 MB	1000 Mb/S
Vm1	200	10000 MB	512 MB	1000 Mb/S
Vm2	500	10000 MB	512 MB	1000 Mb/S

Nous traitons **6** cloudlets avec des longueurs différentes (*length*).

Cloudlets	Cloudlet0	Cloudlet1	Cloudlet2	Cloudlet3	Cloudlets4	Cloudlets5
Length	1500	4000	2500	5000	3000	3500
/MI						

Nous traitons ces *cloudlets* sur les deux data-centers utilisant cinq Algorithmes d'ordonnancement : **Short-First** , **FCFS** , **Round-Robin** *Max-Min* et *Min-Min* (*Remarque : Max-Min et Min-Min et Short-First ne sont pas implémenté dans Cloud-Sim , j'ai fait leur implémentations*) .

Dans l'algorithmes Min-Min et Max-Min fait l'allocation des ressources chacun avec leurs principes mais l'allocation des ressources c'est Roud-Robin (Par défaut) dans les trois autres algorithmes d'ordonnancement (Short-First , FCFS , Roud-Robin) et qui correspond au partage des taches sur les ressources d'une manière équitable .

B. MakeSpan

Si on prend en considération le « *MakeSpan* » *Max-Min* est plus efficace pour terminer le travail (toutes les tâches) dans une durée minimal que les autres algorithmes d'ordonnancement *Min-Min, Short-first, FCFS, Round-Robin* Dans la figure 6, nous pouvons voir les résultats de l'expérience. Sur l'axe des abscisses, est indiqué le **FinshTime** et l'axe des y indique le « No of Task » En examinant la figure, nous pouvons dire que l'algorithme *Max-Min* est plus efficace que les autres algorithmes présentés pour réaliser un « **MakeSpan minimal** » .

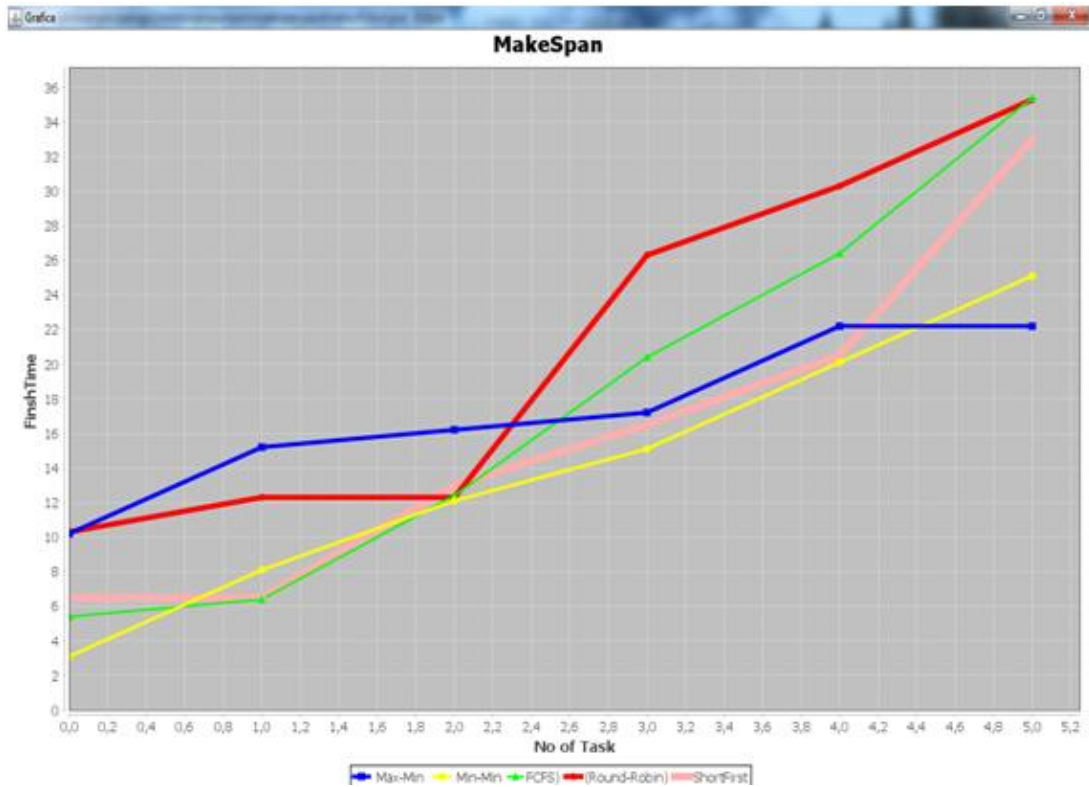


Figure 4 : Comparaison selon le critère MakeSpan tout les algorithmes (ShortFirst , FCFS , RoundRobin ,Max-Min et Min-Min)

3.6 Proposition d'un algorithme qui donne un résultat optimal pour le critère make-span

Je propose un autre algorithme qui donne un résultat optimal pour le makespan, c'est-à-dire un algorithme qui donne un makespan plus petit, par rapport à MaxMin et MinMin , cette algorithme que j'appelle « optimal », améliore le makespan de tel sorte qu'on ne peut avoir un makespan meilleur que ça .

L'astuce de cet algorithme c'est quoi ?

Nous allons procéder par l'exemple :

Cloudlets /Vms	Vm1	Vm2	Vm3
Cloudlet1	1		
Cloudlet2		1	
Cloudlet3	1		
Cloudlet4	1		
Cloudlet5			1

Cloudlet6		1	
Cloudlet7			1

Tableau d'**Allocation des ressources**

Remarque : les positions des 1 sont arbitraires

Si nous supposons que le 1 dans ce tableau signifie que **le Vm j va exécuter la cloudlet i**, et tant que chaque cloudlet peut être exécuté par une et une seule Vm (Vm1 ou Vm2 ou Vm3) cela veut dire qu'on peut avoir un seul **1** pour chaque ligne dans le tableaux qui peut prendre 3 position .

Cloudlets /Vms	Vm1	Vm2	Vm3
Cloudleti	1		

Ou

Cloudlets /Vms	Vm1	Vm2	Vm3
Cloudleti		1	

Ou

Cloudlets /Vms	Vm1	Vm2	Vm3
Cloudleti			1

Donc il ya **3** permutation pour le **1** dans chaque ligne, et tant que nous avons **7** lignes (7 cloudlets) alors le nombre de permutation possibles pour le **1** dans le tableau en total est **3^7** c'est-à-dire le nombre de **Vms**^{le nombre des cloudlets} .

Le but de notre algorithme est de calculer à chaque permutation (attribution des cloudlets au Vms) le make-span. Et nous prenons à la fin la permutation qui donne le meilleur make-spans (make-span le plus petit) et bien sur sa sera **le make-span optimal** donc à ce niveau en peut relier chaque cloudlets à la Vm Appropriée (Allocation des ressources optimal).

A. Scenario

Le but de cette étude est de montrer que l'algorithme que j'ai proposé donne un résultat optimal et meilleur que celui donné par Max-min et Min-min pour le critère **MakeSpan**.

Nous utilisons deux (3) Data-Centers identiques composée chacun d'un Host avec 3 Vms avec les mêmes caractéristiques mais avec des vitesses différentes.

Vm	La vitesse (MIPS)	Size	RAM	BW
Vm0	200	10000 MB	512 MB	1000 Mb/S
Vm1	300	10000 MB	512 MB	1000 Mb/S
Vm2	400	10000 MB	512 MB	1000 Mb/S

Nous traitons **7** cloudlets avec des longueurs différentes (*length*).

Cloudlets	Cloudlet0	Cloudlet1	Cloudlet2	Cloudlet3	Cloudlets4	Cloudlets5	Cloudlets6
Length / MI	2500	2000	3000	1500	1200	1300	2800

Nous traitons ces *cloudlets* sur les trois data-centers en utilisant trois Algorithmes d'ordonnancement et d'allocation des ressources : *Max-Min* et *Min-Min* et *l'algorithme optimal*.

Après l'exécution de la simulation avec ces trois algorithmes *Max-Min* et *Min-Min* et *l'algorithme proposé* les résultats suivants sont affichés.

1-L'algorithme Min-Min

Vm id	Cloudlet Id	Time	StarTime	FinshTime
0	1	10	0,1	10.1
1	5	4.33	0.1	4.43
	0	8.33	4.43	12.76
2	4	3	0.1	3.1
	3	3.75	3.1	6.85
	6	7	6.85	13,85
	2	7.5	13.85	21.35

Table1 .L'attribution des cloudlets sur les ressources par Min-min

(1) On remarque que le *Make-span* = **21.35**

2- Dans l'algorithme Max-Min

<i>Vm id</i>	<i>Cloudlet Id</i>	<i>Time</i>	<i>StarTime</i>	<i>FinshTime</i>
0	0	12.5	0.2	12.7
1	6	9.33	0.2	9.53
	3	5	9.53	14.53
	4	4	14.53	18.53
2	2	7.5	0.2	7.7
	1	5	7.7	12.7
	5	3.25	12.7	15.95

Table2 .L'attribution des cloudlets sur les ressources par Max-min

(2) On remarque que le **Make-span = 18.53**

3-Dans l'algorithmes *Optimal*

<i>Vm id</i>	<i>Cloudlet Id</i>	<i>Time</i>	<i>StarTime</i>	<i>FinshTime</i>
0	2	15	0.3	15.3
1	1	6.67	0.3	6.97
	3	5	6.97	11.97
	5	4.33	11.97	16.3
2	6	7	0.3	7.3
	0	6.25	7.3	13.55
	4	3	13.55	16.55

Table 3 .L'attribution des cloudlets sur les ressources par l'algorithme proposé

(3) On remarque que le **Make-span = 16.55**

On remarque que les résultats de l'algorithme que je propose donne un make-span plus petit (optimal) par rapport aux autres algorithmes Max-Min et Min-Min .

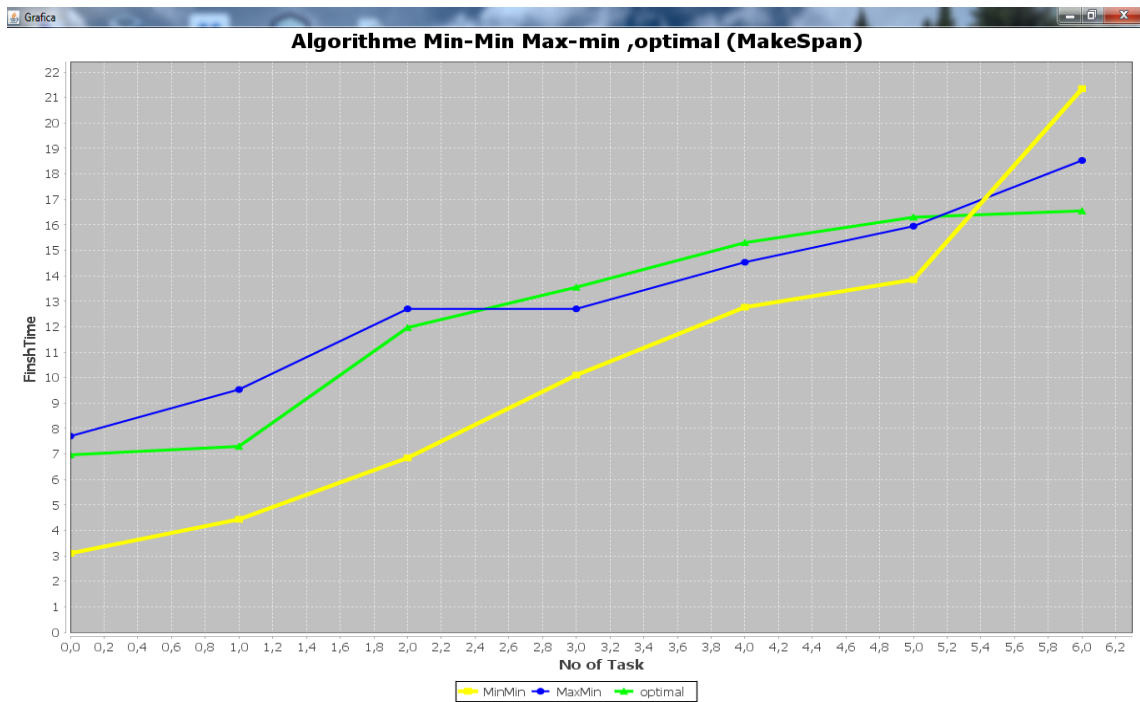


Figure 5 : Comparaison selon le critère MakeSpan les trois algorithmes

Max-Min , Min-Min et l’algorithme que je propose (optimal)

3.7 Conclusion

Au terme de notre étude, nous avons pu établir que l’algorithme Short-First comme algorithme d’ordonnancement des taches donne de meilleur résultat pour le critère dead-line , et pour le critère Make-spans l’algorithme Max-min et Min-min sont deux algorithme d’ordonnancement des taches et d’allocation des ressources pour améliorer ce critère. Après nos études en constate que Max-min est plus efficace que Min-min. Mais l’algorithme que j’ai proposé donne un résultat optimal et est donc meilleur que les deux algorithmes Max-min et Min-min.

Conclusion Générale

Le Cloud computing est la prochaine génération dans l'informatique, probablement les gens peuvent avoir tout ce dont ils ont besoin sur le Cloud .

Le Cloud est une étape dans l'évolution des services à la demande, c'est une technologie naissante qui permet le déploiement d'un grand nombre des applications réparties, des puissances de calcul et une capacité de stockage.

L'ordonnancement des tâches et l'allocation des ressources, est un aspect très important dans l'efficacité de fonctionnement du Cloud.

La théorie d'ordonnancement est une branche de la recherche opérationnelle qui s'intéresse au calcul du temps d'exécution optimal des tâches.

Dans le domaine de l'optimisation de l'ordonnancement des tâches et l'allocation des ressources nous avons comparé entre les deux politiques d'ordonnancement qui sont déjà implémenté dans le Cloud-sim pour le critère dead-line.

Nous avons aussi fait l'implémentation de l'algorithme d'ordonnancement Short-First et le comparer avec les autres algorithmes comme Round-Robin et FCFS pour le critère de dead-line.

Nous avons remarqué que l'algorithme d'allocation des ressources par défaut est RR.

Nous avons comparé les résultats des deux algorithmes d'ordonnancement des tâches et d'allocation des ressources Max-min et Min-min après que nous avoir fait leurs implémentation parce qu'ils ne sont pas implémentés dans le Cloud-sim, nous avons fait cette comparaison selon le critère de Make-span.

Et à la fin nous avons vu qu'il est important de proposer un algorithme optimal pour le critère Make-span. Nous avons fait l'implémentation de cet algorithme qui est plus efficace que Max-min et Min-min. Avec le principe de cet algorithme on peut obtenir des résultats efficaces même pour d'autres critères.

Résumé

Le Cloud computing est une technologie de calcul et de stockage naissante, Cette technologie nous permet de utiliser des ressources à distance , mais vu les demandes croissantes dans le Cloud computing , les ressources sont insuffisantes .

Des stratégies d'ordonnancements et d'allocation des ressources sont la solution pour améliorer l'utilisation des ressources et répondre à la demande .

Dans ce travail nous étudions des stratégies d'ordonnancements et d'allocation des ressources pour améliorer des critères tel que dead-line et make-span .

Nous proposons une nouvelle stratégie que nous voyons la meilleure pour améliorer certains critères .

Mots clé : *Cloud computing , ordonnancement des taches , allocation des ressources .*

Abstract

Cloud computing is a technology of computation and storage nascent, This technology allows us to use resources remotely, but given the increasing demands Cloud computing, resources are insufficient.

Scheduling and resource allocation strategies are the key to improving resource utilization and meeting demand.

In this work we study scheduling and resource allocation strategies to improve criteria such as dead-line and make-span.

We propose new strategy that we see the best to improve certain criteria.

Keywords: *Cloud computing, task scheduling, resource allocation.*

ملخص

الحوسبة السحابية هي تقنية للحوسبة والتخزين الناشئة ، تسمح لنا هذه التكنولوجيا باستخدام الموارد عن بُعد ، ولكن نظراً للطلبات المتزايدة في الحوسبة السحابية ، فإن الموارد غير كافية .

تعد استراتيجيات الجدولة وتخصيص الموارد هي المفتاح لتحسين استخدام الموارد وتلبية الطلب .

في هذا العمل ، ندرس جدولة واستراتيجيات لتخصيص الموارد لتحسين المعايير مثل الخط النهائي والامتداد .

نقترح استراتيجية جديدة نرى أنها الأفضل لتحسين معايير معينة .

كلمات البحث: الحوسبة السحابية ، جدولة المهام ، تخصيص الموارد .

Bibliographie

- [1] François Pêcheux professeur à l'université de Paris VI-Pierre –et –Marie-Curie , « CLOUD COMPUTING ou INFORMATIQUE EN NUAGE » , <https://www.universalis.fr/encyclopedie/cloud-computing-informatique-dans-les-nuages/2-quelques-jalons->, consulté le 12-02-2018
- [2] Peter Mell (NIST) , Tim Grance (NIST) “the NIST definition of cloud ” <https://csrc.nist.gov/publications/detail/sp/800-145/final> (12-02-2018).
- [3] ...”petite hystorique de cloud computing “<https://www.usinenouvelle.com/article/petite-histoire-du-cloud-computing.N173570> [consulté le 12-02-2018]
- [4], « Cloud computing et entreprise /Avantages limites du Cloud », https://fr.wikiversity.org/wiki/Cloud_computing_et_entreprise/Avantages_et_Limites_du_Cloud_computing , consulté le 12-02-2018 .
- [5] Wikipedia SAAS PAAS IAAS : A clear explanation of cloud service <https://rubygarage.org/blog/iaas-vs-paas-vs-saas> (consulté le 14-02-2018) .
- [6] Sam Johnston — modification of the Wikipedia file, Cloud computing.svg, created by Sam Johnston using OmniGroup's OmniGraffle and Inkscape (includes Computer.svg by Sasa Stefanovic) https://en.wikipedia.org/wiki/Cloud_computing (12-02-2018)
- [7] ... tels que définis par NIST “comprendre le cloud computing : Modèles de déploiement” <http://cloudcomputinginfrench.blogspot.com/2013/01/comprendre-le-cloud-computing-4-modeles.html> (29-05-2018).
- [8] Mohammad Hamdaqa and Ladan Tahvildari , “Cloud Computing Uncovered: A Research Landscape”. Elsevier Press. pp. 41–85. ISBN 0-12-396535-7.
- [9] Huang Q.Y., Huang T.L., “An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing”, IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS).
- [10] Cristian Mateos, Elina Pacini & Carlos Garcia Garino, (2013), An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments.
- [11] K. Etmnani, and M. Naghibzadeh, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling,"The Third IEEE/IFIP International Conference on Internet, Uzbekistan,
- [12] Pooja Samal and Pranati Mishra, (2013), "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", International Journal of Computer Science and Information Technologies, pp. 416-419, Vol. 4(3)

Bibliographie

[14] the cloud computing and distributed systems (cloud laboratory) , university of merbourne (cloud sim)[http:// www. Loud bus org /cloud sim /](http://www.Loud bus org /cloud sim /)

[15] Fei Teng resources allocation and scheduling models for cloud computing , ecole international paris

[16] Qi Zhang , QuanYarzhu and Raouf Boutaba, Dynamic resource allocation for spot markets in cloud computing environnement , in the fourth international conference .

[17] Han Zhao : market oriented task scheduling .in the international parallel and distributed processing .

[18] Rodrigues , Mendez and Lloret . Distributed Media –aware flow scheduling in cloud computing environment .Septembre 2012 .

[19] Michael Mender , Soumen chakrabarti and Muthukrishnan , Flow and stretch metrics for scheduling continuous job streams .In proceeding of the 9th annua ACM-SIAM Symposium On Discrete Algorithms .

Dédicaces

A mes parents

A mes frères , mes sœurs et mes neveux

A mon encadreur

A mes amis et mes collègues

A tous ceux qui m'ont encouragé et aidé

Remerciements

Je remercie Allah de m'avoir donné le courage et la volonté ainsi que la conscience et la patience d'avoir pu terminer mon travail .

Je tiens à exprimer mes vifs remerciements à mon encadreur Mr Belhocine Amine pour m'avoir donné l'opportunité de réaliser ce sujet sous sa direction . La confiance faite ainsi ses conseils fructueux , et son temps consacré tout au long du travail .

Je remercie les membres jury d'avoir accepté d'être membre dans la soutenance .

Enfin , un merci particulier à tous ce qui m'ont soutenu de près ou de loin par leurs soutiens et encouragements .