République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ A.MIRA-BEJAIA

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Faculté des Sciences Exactes

Département de Recherche Operationnelle

Unité de Recherhe LaMOS

THÈSE

EN VUE DE L'OBTENTION DU DIPLÔME DE DOCTORAT

Domaine : Mathématique et Informatique Filière : Mathématiques Appliquées

Spécialité : Recherche Opérationnelle et Aide à la Décision

Présentée par

Melle Assia OUTAMAZIRT

Thème

Application des Modèles d'Attente pour l'Évaluation des Performances dans le Cloud Computing

Soutenue : le 25/04/2019 **Devant le Jury composé de :**

Mr	**ADJABI Smail**	**Professeur**	**Univ. de Bejaia**	**Président**
Mr	**AÏSSANI Djamil**	**Professeur**	**Univ. de Bejaia**	**Rapporteur**
Mr	**BARKAOUI Kamel**	**Professeur**	**CNAM Paris**	**Co-Rapporteur**
Mme	**DJELLAB Natalia**	**Professeur**	**Univ. de Annaba**	**Examinatrice**
Mr	**YAZID Mohand**	**M. C. A**	**Univ. de Bejaia**	**Examinateur**
Mme	**LEKADIR Ouiza**	**M. C. A**	**Univ. de Bejaia**	**Examinatrice**

Année Universitaire 2018/2019

A la mémoire de ma mère.

A mon très cher père.

A toute ma famille.

A mes amis.

Remerciements

Je tiens à exprimer mes sincères gratitude et remerciements à mon directeur de thèse, Monsieur Djamil AÏSSANI, Professeur à l'Université de Bejaia, pour son encadrement exceptionnel. Je lui suis reconnaissante de m'avoir fait partager ses brillantes intuitions. Je lui suis également reconnaissante pour le temps qu'il m'a accordé et ses qualités pédagogiques et scientifiques.

Je tiens à exprimer toute ma reconnaissance et mes plus vifs remerciements à mon co-directeur de thèse, Monsieur Kamel BARKAOUI, professeur des Universités au Conservatoire National des Arts et Métiers (Cnam-Paris), pour ses qualités tant scientifiques qu'humaines et sa disponibilité malgré ses nombreuses charges. Je tiens également à le remercier pour son aide qu'il m'a apportée durant ces années de recherche. Qu'il trouve ici ma profonde gratitude.

J'adresse également tous mes remerciements à Monsieur Smail ADJABI, Professeur à l'Université de Bejaia, de l'honneur qu'il m'a fait en acceptant d'être président du jury de cette thèse.

Je tiens à exprimer ma gratitude à Madame Natalia DJELLAB, Professeur à l'Université de Annaba, Monsieur Mohand YAZID et Madame Ouiza LEKADIR, Maîtres de Conférences classe A à l'Université de Bejaia, qui ont bien voulu être examinateurs.

Je tiens à exprimer toute ma reconnaissance à Monsieur Mohamed ESCHEIKH, Maître de Conférences classe A à l'École Nationale d'Ingénieurs de Tunis (INIT), de m'avoir accueilli au sein de son équipe. Il a répondu à toutes mes questions et, ses réponses m'ont toujours éclairci les idées. Je lui en suis très reconnaissante.

Je tiens à remercier vivement Madame Samia BOUZEFRANE, Maître de Conférences/HDR au Conservatoire National des Arts et Métiers (Cnam-Paris), pour son aide constante.

Enfin, je tiens particulièrement à exprimer ma profonde reconnaissance à ma famille et mes amis pour leur soutien et leurs encouragements.

Productions scientifiques

Journaux internationaux

1. A. Outamazirt, M. Escheikh, D. Aïssani, K. Barkaoui and O. Lekadir *"Performance analysis of the M/G/c/c + r queuing system for cloud computing data centres"*, Int. J. Critical Computer-Based Systems, Vol. 8, Nos. 3/4, pp.234–257, 2018. (DOI : 10.1504/IJCCBS.2018.096441).

2. A. Outamazirt, K. Barkaoui and D. Aïssani *"A novel queuing model for maximizing profit of data centers with heterogeneous services"*. Article accepté pour publication dans IEEE/CAA Journal of Automatica Sinica.

3. A. Outamazirt, D. Aïssani and K. Barkaoui *"Modeling and Calculation of Elasticity in Cloud Computing"*. Article soumis.

Conférences internationales à comité de lecture

1. A. Outamazirt, M. Escheikh, D. Aïssani, K. Barkaoui and O. Lekadir, *"On the modeling and Performance Evaluation of Cloud Computing Centers Using M/G/c/c+r Queuing System"*, Proceedings of the 10th Workshop on Verification and Evaluation of Computer and Communication System (VECoS'2016), Tunis, Tunisia, Vol 1689, pp 77-84, 2016.

2. A. Outamazirt, D. Aïssani and K. Barkaoui *"A New analytical formula for computing the delay probability in cloud center modeling as M/G/c/k queue"*, The First International Conference on the Evolution of Contemporary Mathematics and their Impact in Sciences and Technology (ECMISciTech'2017), Constantine, Algeria, 2017.

3. A. Outamazirt, K. Barkaoui and D. Aïssani *"Maximizing profit in cloud computing using M/G/c/k queuing model"*, International Symposium on Programming and Systems (ISPS'2018), Algiers, Algeria, pp. 1-6, 2018. (DOI : 10.1109/ISPS.2018.8379008). https ://ieeexplore.ieee.org/document/8379008.

Conférences nationales

1. A. Outamazirt, K. Barkaoui and D. Aïssani *"Evaluation des performances de centre de données du cloud computing via les files d'attente"*, 3ème journée doctorale en Informatique Décisionnelle et Informatique Distribuée, Bordj Bou Arreridj, Algérie, 2015.

2. A. Outamazirt, D. Aïssani and K. Barkaoui *"Modélisation et calacul de l'élasticité dans le cloud computing"*, Actes des Premières Doctoriales Nationales de Recherche Opérationnelle, Béjaia, Algérie, pp. 29-30, 2018.

TABLE DES MATIÈRES

TABLE DES FIGURES

INTRODUCTION GÉNÉRALE

L'apparition de la virtualisation, de l'infogérance, de l'externalisation et de la démocratisation de l'informatique à la fin de XX^e siècle, a permis d'assister ces dernières décennies à l'explosion du Cloud Computing (l'informatique en nuage). Ce concept constitue une tendance mondiale en matière d'acquisition de services technologiques et une véritable révolution dans l'utilisation de l'informatique en amenant de nouvelles possibilités de mutualisation de services et d'économies pour les entreprises, notamment par le fait de diminuer les coûts d'exploitation des infrastructures technologiques et des applications. Il s'agit d'un nouveau mode d'acquisition qui permet aux utilisateurs d'accéder, via Internet, à un bassin de ressources informatiques configurables, externalisées et qui sont proposées sous forme de services [1]. Ce nouveau mode de livraison de services permet aux consommateurs de s'approvisionner en services de technologies de l'information auprès d'un prestataire du Cloud Computing de façon automatisée et sur demande. La consommation des services est mesurée et facturée selon l'utilisation.

L'avancement accéléré de l'utilisation du Cloud Computing a développé un large éventail de services Cloud. La réussite du grand déploiement de ces services par les fournisseurs de service Cloud (Cloud service providers), tels que Amazon EC2 (Elastic Compute Cloud), Google Cloud, IBM (International Business Machines) Cloud, Microsoft Cloud, ..., dépend étroitement de la garantie des caractéristiques de qualité annoncées exprimées en fonction de divers attributs de performance et de qualité de service et obtenues par l'adoption de stratégies appropriées d'approvisionnement en ressources informatiques. En effet, afin d'avoir le succès commercial de ces services, il est important que les fournisseurs de service en nuage procurent des services qui répondent aux besoins variés en ressources informatiques de leurs clients selon les exigences de qualité de service spécifiées dans les SLAs (Service Level Agreements). Par conséquent, la

performance du service Cloud a un impact significatif sur la performance globale de la future infrastructure d'information. Par ailleurs, les fournisseurs de service Cloud doivent avoir une connaissance approfondie de la relation entre la performance du service et les ressources informatiques disponibles pour pouvoir utiliser pleinement leurs infrastructures tout en répondant aux demandes de leurs clients et en respectant les contrats SLAs conclus avec eux. Les utilisateurs de services Cloud veulent également des méthodes efficaces pour évaluer les performances afin de prendre de bonnes décisions concernant la sélection et la composition optimales des services. Ainsi, l'évaluation des performances des services Cloud est importante et avantageuse pour les fournisseurs de service et les consommateurs. Par conséquent, développer des méthodes efficaces et précises pour évaluer les performances de ces services est devenu un problème de recherche très important qui a suscité une attention considérable de la part des milieux universitaires et industriels. Parmi les approches fondamentales permettant d'évaluer les performances dans le Cloud Computing, on trouve celles basées sur la modélisation stochastique.

La modélisation stochastique fondée sur la théorie des files d'attente (Queuing Theory, ou Théorie des services des masses) est largement utilisée dans la représentation et l'étude du comportement dynamique de systèmes sophistiqués. Cette théorie est née d'exigences pratiques liées à la nécessité de fournir aux décideurs des méthodes mathématiques d'aide à l'organisation rationnelle du service massif de clients, ou de phénomènes apparentés, et qui conduisent à la formation de files d'attente [2] ; et elle a pour objectif l'analyse des indicateurs de mesure des performances du service.

L'objectif de cette thèse est de proposer des modèles analytiques pour la modélisation et l'évaluation des performances dans le Cloud Computing. Particulièrement, nous nous intéressons à l'analyse mathématique de modèles de files d'attente pour l'évaluation des performances des Cloud Data Centers qui permettent d'offrir des services de qualité plus accessibles pour les fournisseurs de service Cloud et qui répondent aux caractéristiques dynamiques de l'environnement du Cloud Computing.

Les principales contributions de cette thèse peuvent être résumées comme suit :

- Nous avons proposé le modèle de files d'attente $M/G/c/k$ pour la modélisation et l'évaluation des performances du Cloud Data Center [**?**, 4]. En effet, ce modèle répond à certaines caractéristiques des Cloud Data Centers : les temps de service de loi générale, un grand nombre de serveurs et une capacité finie du système. Nous avons effectué une analyse critique des modèles élaborés dans la littérature sur l'évaluation des performances dans le Cloud Computing en présentant les lacunes de chaque modèle, puis nous avons proposé des

améliorations [3]. Par la suite, nous avons effectué une analyse mathématique du modèle de files d'attente $M/G/c/k$ en introduisant le processus stochastique qui convient mieux pour le nombre de demandes de services présentes dans le système aux instants d'arrivées [4].

- ⊵ Nous avons étendu le modèle de files d'attente $M/G/c/k$ en tenant compte du comportement des clients impatients et nous nous sommes intéressés au problème de la configuration optimale pour maximiser le profit des fournisseurs de service Cloud [5].
- ⊵ Nous avons adopté le modèle de files d'attente $MMPP/G/c/k$ pour la modélisation analytique du Cloud Data Center afin de tenir compte de la variation des taux d'arrivées des demandes des utilisateurs de service Cloud dans le temps [6]. Cette modélisation reflète la nature réelle des Cloud Data Centers. Nous nous sommes intéressés au problème de la configuration optimale pour maximiser le profit des fournisseurs de service Cloud dans les Cloud Data Centers qui fournissent des services hétérogènes.
- ⊵ Enfin, nous nous sommes intéressés en particulier à l'élasticité qui est l'une des cinq caractèritiques du Cloud Computing. Ici, notre contribution concerne le développement d'un modèle analytique qui nous a permis d'analyser et de calculer la valeur de l'élasticité dans le Cloud Computing d'une manière précise.

Plan de la thèse

Après une introduction générale, le reste de cette thèse est organisé en deux parties, une conclusion générale et s'achève par une bibliographie.

La première partie concerne les contextes et l'état de l'art. Cette partie est constituée de deux chapitres :

- ⊵ Dans le premier chapitre, nous présentons les principaux concepts liés au Cloud Computing.
- ⊵ Dans le deuxième chapitre, nous présentons les notions et techniques de base sur les systèmes de files d'attente classiques et nous réalisons une étude bibliographique liée aux modèles non-markoviens à plusieurs serveurs.

La deuxième partie concerne les principales contributions réalisées durant cette thèse. Cette partie est organisée en quatre chapitres :

- ⊵ Le premier chapitre est consacré à l'évaluation des performances du Cloud Data Ceter modélisé comme un modèle de files d'attente $M/G/c/k$.

- ⊵ Dans le deuxième chapitre, nous formulons et résolvons le problème de la configuration optimale pour maximiser le profit des fournisseurs de service Cloud.

- ⊵ Le troisième chapitre est consacré à l'évaluation des performances du Cloud Data Center qui fournit des services hétérogènes modélisé comme un modèle de files d'attente $MMPP/G/c/k$.

- ⊵ Enfin, le quatrième chapitre est consacré à la modélisation analytique et l'évaluation des performances du Cloud élastique.

Première partie

Contexte et état de l'art

CHAPITRE 1

CONCEPTS DU CLOUD COMPUTING

Introduction

Le Cloud Computing est le hype informatique de cette dernière décennie et le fruit des évolutions récentes des technologies de l'information. Il constitue une véritable révolution dans l'utilisation de l'informatique en amenant des nouvelles possibilités de mutualisation de services et d'économies pour les entreprises.

L'objectif de ce chapitre est de présenter les principaux concepts du Cloud Computing.

1.1 Qu'est-ce que le Cloud Computing ?

1.1.1 Définitions

Dans la littérature, il existe plusieurs définitions concernant le concept du Cloud Computing (voir par exemple [1, 7–10]). Dans [7], les auteurs définissent le Cloud Computing comme une solution d'approvisionnement de ressources, de déploiement, d'équilibreur de charge, de modèle économique et d'architecture Web 2.0. D'autres auteurs tels que R. Buyya et al. [8] mettent l'accent sur l'accord de prestation de service SLA (Service-Level Agreement) et la qualité de service, ainsi ils considèreent le Cloud Computing en tant que une solution de virtualisation de machines interconnectées et approvisionnées à la demande comme un ensemble de ressources unifiées en se basant sur le SLA établi entre le fournisseur et le consommateur. Dans [9], les auteurs considèrent la mise à l'échelle automatique, le modèle à la demande sur la base du principe "payez uniquement ce que vous consommez", l'utilisation ubiquitaire et la virtualisation comme élé-

ments essentiels qui définissent le Cloud Computing. Dans le même contexte, l'organisme NIST (National Institute of Standards and Technology) [1] le définit comme un modèle qui permet d'accéder au réseau, de façon ubiquitaire facile et à la demande, à un ensemble de ressources informatiques configurables (réseaux, serveurs, stockage, applications et services) pouvant être rapidement provisionnées et libérées par un minimum d'efforts de gestion ou d'interaction avec le fournisseur de service en nuage.

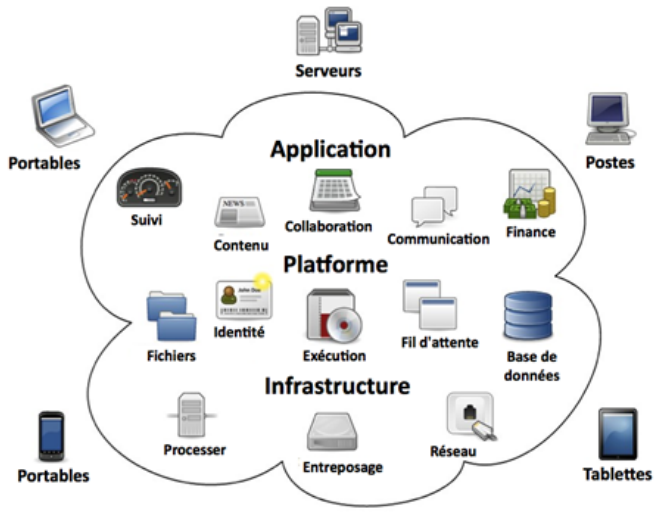

FIGURE 1.1 – Facteurs principaux du Cloud Computing.

Le Cloud Computing, tel que défini par le NIST, se compose de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement.

1.1.2 Caractéristiques essentielles du Cloud Computing

Cinq caractéristiques clés et essentielles du Cloud le différencient de l'informatique traditionnelle :

1. **Un accès en libre service à la demande :** Un client pourra commander des ressources informatiques en fonction de ses besoins. Les ressources informatiques sont fournies d'une manière entièrement automatisée et c'est le client, au moyen d'une interface, qui met en place et gère la configuration à distance.
2. **Un accès ubiquitaire au réseau :** L'ensemble des ressources sont disponibles et accessibles sur le réseau par tout type de terminal client (téléphones mobiles, tablettes, ordinateurs portables, stations de travail, etc).

3. **Une mise en commun des ressources :** Les ressources informatiques du fournisseur telles que la bande passante réseau, machines virtuelles, mémoire, puissance de traitement, capacité de stockage,..., sont mises à la disposition des clients sur un modèle multi-locataires, avec une attribution dynamique des ressources physiques et virtuelles en fonction de la demande. Le client n'a généralement aucun contrôle ni connaissance sur la localisation exacte des ressources fournies, mais peut éventuellement spécifier une localisation à un niveau d'abstraction supérieur (par exemple, pays, état ou centre de données).
4. **Une élasticité rapide :** Les ressources informatiques mises à disposition du client peuvent être ajustées (augmenter ou diminuer) rapidement (quelques minutes voire quelques secondes) de manière automatique, de telle façon à ce que les ressources fournies soient conformes à la demande du système.
   Dans le dernier chapitre de ce document, nous allons présenter en détail cette caractéristique.
5. **Un service mesuré en permanence :** Les ressources consommées sont contrôlées et communiquées au client et au fournisseur de service de façon transparente. Cela garantit un niveau de disponibilité adapté aux besoins spécifiques des clients.

1.1.3 Modèles de services

Les principaux services proposés en Cloud Computing sont `SaaS (Software-as-a-Service)`, le `PaaS (Platform-as-a-Service)` et le `IaaS (Infrastructure-as-a-Service)`. Ces trois modèles de service doivent être déployés sur des infrastructures qui possèdent les cinq caractéristiques essentielles citées précédemment pour être considérés comme des services type Cloud. Les facteurs de différenciation entre ces trois modèles de service sont la nature du service, le niveau de contrôle client-fournisseur et d'engagement.

1. **SaaS :** Ce modèle de service est caractérisé par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud. Les applications sont accessibles à partir de divers périphériques clients via une interface de client léger, telle qu'un navigateur Web (par exemple, une messagerie Web), ou une interface de programme. Autrement dit, il n'est plus nécessaire pour le consommateur d'effectuer les installations des logiciels, les mises à jour ou encore les migrations de données avec le SaaS. Le consommateur ne gère pas et ne contrôle pas l'infrastructure sous-jacente (réseaux, serveurs, applications, stockage).
   De bons exemples de SaaS sont les logiciels de messagerie au travers d'un navigateur

comme Gmail, Yahoo mail, ...

2. **PaaS :** La capacité fournie au consommateur consiste à déployer sur l'infrastructure de Cloud des applications créées ou acquises par le consommateur et créées à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur. Le consommateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous jacente (le réseau, les serveurs, les systèmes d'exploitation ou le stockage) mais contrôle l'application déployée et sa configuration.

   Comme exemple de PaaS, on peut citer un des plus anciens Intuit Quickbase qui permet de déployer ses applications bases de données en ligne ou Google Apps Engine (GAE) pour déployer des services Web.

3. **IaaS :** La capacité fournie au consommateur consiste à fournir le traitement, le stockage, les réseaux et autres ressources informatiques fondamentales permettant au client de déployer et d'exécuter n'importe quel type de logiciel, pouvant inclure des systèmes d'exploitation et des applications. Le consommateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous jacente mais il a le contrôle sur les systèmes d'exploitation, le stockage et les applications. Il peut aussi choisir les caractéristiques principales des équipements réseau comme le partage de charge, les pare-feu, etc.

   L'exemple emblématique de ce type de service est Amazon Web Services qui fournit du calcul (EC2), du stockage (S3, EBS), des bases de données en ligne (Simple DB) et quantité d'autres services de base.

Chacun de ces types de services représente une offre différente dans le Cloud.
La figure 1.2 résume comment les tâches d'administration des trois modèles de services du Cloud Computing sont réparties.

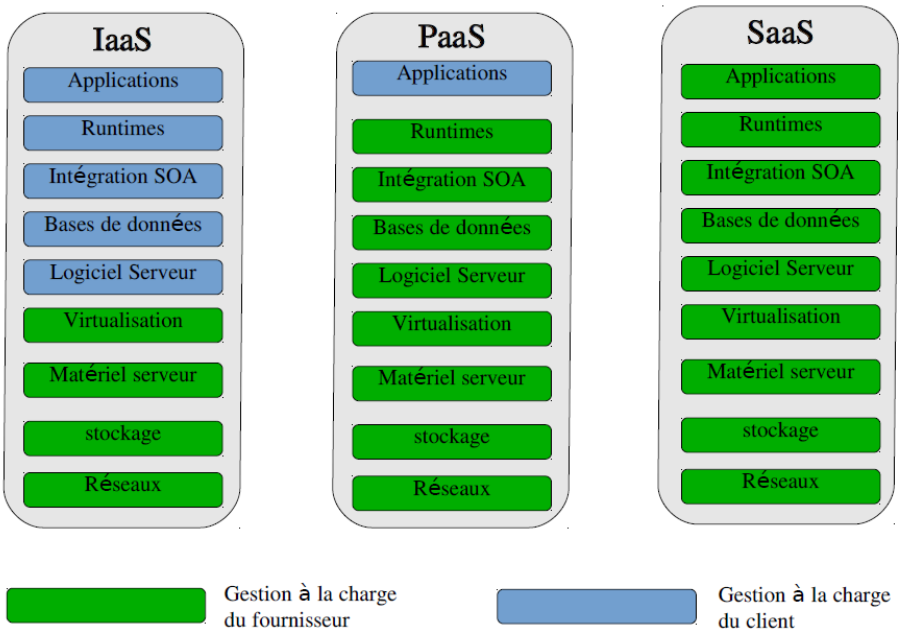

FIGURE 1.2 – Répartition des tâches d'administration des modèles de services.

1.1.4 Modèles de déploiement

L'externalisation des données peut être réalisée de différentes manières. On distingue ainsi plusieurs types de Cloud selon son caractère public, privé, communautaire ou hybride.

1. **Le Cloud public :** Nuage externe à l'entreprise, accessible via Internet ou un réseau privé, géré par un opérateur externe propriétaire des infrastructures, avec des ressources totalement partagées entre tous ses clients.

   Un Cloud public est un service IaaS, PaaS ou SaaS proposé et hébergé par un tiers. Amazon, Google et Microsoft proposent un Cloud public dans lequel n'importe quel particulier ou n'importe quelle entreprise peut y héberger ses applications, ses services ou ses données. Pour les consommateurs, il n'y a donc aucun investissement initial fixe et aucune limite de capacité.

   Les fournisseurs de Cloud public facturent à l'utilisation et garantissent une disponibilité de services au travers des contrats SLA.

   Un exemple de service grand-public fourni en Cloud Computing, est le jeu à la demande (aussi appelé jeu sur demande, et, en anglais, gaming on demand (GoD) ou Cloud gaming). Il permet de jouer normalement à des jeux vidéo sur son écran d'ordinateur, alors que le ou les logiciels de jeu tournent sur des serveurs à distance, qui renvoient la vidéo de ce qui a été joué en lecture en continu (ce qui est communément appelé streaming). Le jeu est

hébergé et stocké sur des serveurs, dont l'utilisateur ne connait pas la localisation ni les caractéristiques. Il ne nécessite plus de supports comme les CD, ou de matériel comme les consoles de jeux. Les joueurs doivent seulement posséder un ordinateur relié à l'Internet, et le cas échéant une manette de jeu.

2. **Le Cloud privé :** Nuage réservé à l'usage exclusif d'une seule organisation. Il peut être possédé, géré et opéré par cette organisation, un intervenant extérieur ou une combinaison des deux. Généralement il est situé dans les locaux de l'organisation.
3. **Le Cloud communautaire :** Nuage réservé à l'usage d'une communauté spécifique de consommateurs partageant des intérêts communs. Il peut être possédé, géré et opéré par un ou plusieurs organismes participant à la communauté, un intervenant extérieur ou une combinaison d'entre eux. Il est situé dans les locaux des organismes participant ou dans ceux d'un hébergeur externe.
4. **Le Cloud hybride :** Nuage résultat d'une conjonction de deux ou plusieurs Clouds différents (public, privé ou communautaire) amenés à coopérer, à partager entre eux les applications et les données.

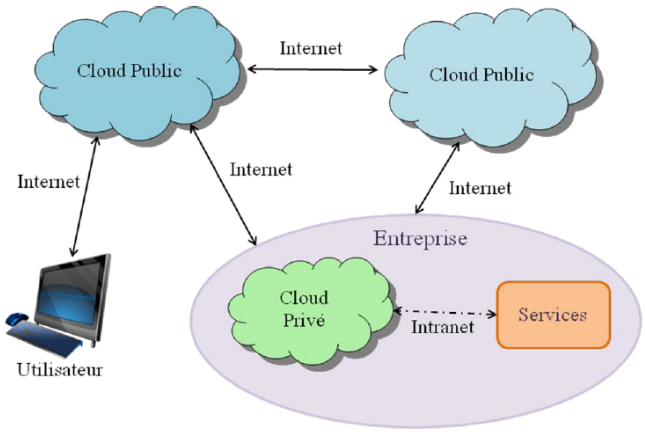

FIGURE 1.3 – Un Cloud hybride.

1.2 Avantages et inconvénients du Cloud Computing

1.2.1 Avantages

Le Cloud Computing permet aux utilisateurs professionnels et aux utilisateurs finaux de disposer de trois principaux avantages suivants : **l'approvisionnement en libre-service** ; **le**

paiement à l'utilisation et **l'élasticité**. L'approvisionnement en libre-service permet aux utilisateurs finaux de commander des ressources informatiques selon leurs besoins et d'y accéder à la demande. Le paiement à l'utilisation autorise les firmes à ne payer que pour les ressources consommées. Enfin, l'élasticité offre l'opportunité d'approvisionner et de désapprovisionner des ressources informatiques de manière automatique, de telle façon à ce que les ressources fournies soient conformes à la demande de l'entreprise.

Une multitude d'autres avantages peuvent être cités :

- Economie : Vous pouvez bénéficier d'économies d'échelle qui ont une répercussion économique. Il est inutile d'investir dans une infrastructure qui serait très onéreuse à l'achat et vous avez une possibilté d'avoir accès à des services parfois coûteux à moindre prix et de manière évolutive.
- Gain de temps sur la maintenance : Vous n'avez plus à vous soucier des mises à jour à effectuer, des problématiques de stockages et de performances. Grâce au Cloud tout ceci est géré par votre prestataire.
- Accessibilité et mobilité : Les applications et services que vous utilisez dans le Cloud sont accessibles à tout moment et à partir de n'importe quel type d'appareil à condition que celui-ci soit doté d'une connexion internet.
- Mutualisation des ressources : La mutualisation des ressources permet de disposer de capacités illimitées en matière de stockage et de bande passante.
- Coût : Du fait que le même service est proposé à de nombreux utilisateurs, son coût en est nettement amoindri.

Cependant, cette technologie présente quelques inconvénients qu'on citera dans la sous-section suivante.

1.2.2 Inconvénients

- Connexion internet : Le Cloud utilise de manière intensive le transfert de données, il faut avoir une connexion très performante.

  L'utilisateur est extrêmement dépendant du réseau ; si il n'y a pas de connexion internet, ou si le réseau est en panne, il ne pourra pas accéder à sa plateforme de travail.
- Sécurité et confidentialité : L'utilisation des réseaux publics entraîne des risques liés à la sécurité du Cloud. En effet, la connexion entre les postes et les serveurs applicatifs passe par le réseau Internet et expose à des risques supplémentaires de cyberattaques et

de violation de confidentialité.

1.3 Les acteurs du Cloud Computing

Il serait bien trop conséquent d'analyser tous les acteurs du Cloud Computing présents sur le marché actuel. Dans cette section, nous survolerons les différentes solutions apportées par les principaux acteurs : Salesforce.com, Amazon, Google et Microsoft.

Salesforce.com est une société créée en 1999 par Marc Benioff. Elle est devenue l'une des pionnières du modèle SaaS notamment grâce à son outil historique de CRM (Customer Relationship Management) intitulé Salesforce.

Amazon Web Services (AWS) est une division du groupe américain de commerce électronique Amazon.com, lancé officiellement en 2006 par Andy Jassy. AWS est dédiée aux services de Cloud Computing à la demande pour les entreprises.

En 2017, AWS propose plus de 90 services, comprenant le calcul, le stockage, le réseau, la base de données, l'analyse de données, des services applicatifs, du déploiement, de la gestion de système, de la gestion d'applications mobiles, des outils pour les développeurs et pour l'Internet des objets. Les services les plus populaires sont Amazon Elastic Compute Cloud (EC2) fournissant des serveurs virtuels et Amazon Simple Storage Service (S3) fournissant un stockage basé sur les services web.

En 2008, Google a lancé son Cloud public orienté pour les services Web offrant une plate-forme PaaS nommée `Google App Engine` et permettant l'hébergement d'applications Python ou Java, ainsi que des applications SaaS regroupées dans la gamme `Google App`.

Lancée en 2010. Comme les autres fournisseurs de services Cloud, Microsoft Azure permet de profiter de ressources de Cloud Computing à la demande. Cette plateforme Cloud permet de simplifier l'utilisation et l'administration de technologies Microsoft comme Windows Server, Active Directory et SharePoint.

1.4 Virtualisation

Le concept de virtualisation a été la première pierre vers l'ère du Cloud Computing. Ce concept permet de faire abstraction des détails des ressources physiques et de les considérer comme des pools de ressources.

1.4.1 Principe de la virtualisation

La virtualisation est le processus qui consiste à créer une version virtuelle d'une entité physique. La virtualisation peut s'appliquer aux serveurs, au stockage, aux applications et aux réseaux [11–13]. Il s'agit de la manière la plus efficace de réduire les dépenses informatiques tout en stimulant l'efficacité et la flexibilité des entreprises de toute taille. Cette technologie apporte de très nombreux avantages ; en effet, elle permet d'accroître l'agilité et l'évolutivité de l'infrastructure informatique tout en assurant des économies considérables. Mobilité accrue des charges de travail, optimisation des performances et de la disponibilité, automatisation des opérations. Tous ces avantages simplifient la gestion de l'informatique et réduisent les coûts de possession et d'exploitation. D'autres avantages de la virtualisation peuvent être cités, tels que :

- Réduction ou élimination des interruptions de service.
- Provisionnement plus rapide des applications et des ressources.
- Continuité et reprise d'activité optimales.
- Gestion simplifiée du Data Center.
- ...

1.4.2 Machine virtuelle

La virtualisation fait appel au logiciel pour émuler l'existence du matériel et créer un système informatique virtuel. Ce modèle permet aux entreprises d'exécuter plusieurs machines virtuelles sur un seul et même serveur.

Chaque système informatique virtuel correspond à une machine virtuelle ou VM (Virtual Machine), c'est-à-dire, un conteneur de logiciels totalement isolé, capable d'exécuter ses propres systèmes d'exploitation et applications, à l'instar d'un ordinateur physique. Chaque machine virtuelle est une entité autonome et complètement indépendante et, plusieurs de ces machines virtuelles peuvent être hébergées en parallèle sur une seule machine physique [14, 15]. Aussi, chaque machine virtuelle se comporte exactement comme un ordinateur physique. Elle contient un processeur, une mémoire RAM, un disque dur et une carte d'interface réseau virtuels qui lui sont propres.

Les caractéristiques des machines virtuelles offrent plusieurs avantages :

Partitionnement :

- Exécutez plusieurs systèmes d'exploitation sur une machine physique.
- Répartissez les ressources entre les machines virtuelles.

Interopérabilité du matériel :

- Provisionner ou migrer n'importe quelle machine virtuelle vers n'importe quel serveur physique.

Isolation :

- Assurer l'isolation des pannes et la protection de la sécurité au niveau matériel.
- Maintenir les performances en déployant des contrôles avancés des ressources.

Encapsulation :

- Enregistrer dans des fichiers l'état complet de chaque machine virtuelle.
- Déplacer et copier des machines virtuelles aussi facilement que des fichiers.

1.4.3 Domaines de la virtualisation

Les domaines de la virtualisation (Stockage, serveurs, applications ...) différant très significativement l'un de l'autre. Dans cette sous-section, nous présentons brièvement ces différents domaines de la virtualisation, à savoir :

La virtualisation de stockages

La virtualisation de stockage consiste a présenter une source de stockage uniforme, c-à-d, elle consiste à regrouper des ressources de stockage physique issues de plusieurs dispositifs de stockage en réseau comme un dispositif de stockage unique, administré depuis une console centrale. Son principe de base est de gérer une interface qui permet de dissocier la gestion physique des disques

(et des baies de stockage) vis-à-vis des serveurs qui l'utilisent.

La virtualisation de serveurs

D'une manière générale, la virtualisation de serveur est un principe permettant de faire fonctionner simultanément, sur un seul serveur physique, plusieurs serveurs virtuels. Cette technique permet aux entreprises d'utiliser des serveurs virtuels au lieu de serveurs physiques. Si cette virtualisation est faite au sein de la même entreprise, le but est de mieux utiliser la capacité de chaque serveur par une mise en commun de leur capacité.

FIGURE 1.4 – Virtualisation des serveurs.

La virtualisation d'applications

La virtualisation d'application est une technologie logicielle qui permet d'améliorer la portabilité et la compatibilité des applications en les isolant du système d'exploitation sur le quel elles sont exécutées. Elle consiste à encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné.

La virtualisation d'application nécessite l'ajout d'une couche logicielle supplémentaire entre un programme donné et le système d'exploitation ; son but est d'intercepter toutes les opérations d'accès ou de modification de fichiers ou de la base de registre afin de les rediriger de manière totalement transparente vers une localisation virtuelle.

La virtualisation de réseaux

De manière générale, la virtualisation des réseaux consiste à partager une même infrastructure physique (débit des liens, ressources CPU des routeurs,...) au profit de plusieurs réseaux virtuels isolés. Elle permet aux applications de s'exécuter sur un réseau virtuel comme si c'était

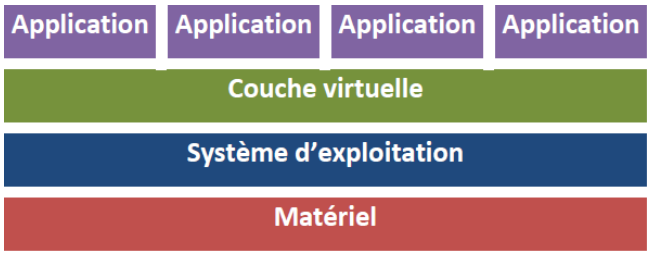

FIGURE 1.5 – Virtualisation d'applications.

un réseau physique, mais avec des avantages opérationnels supérieurs et sans dépendance vis-à-vis du matériel. La virtualisation de réseau présente les périphériques et les services réseau logiques (ports, commutateurs, routeurs, pare-feu, répartiteurs de charge, VPN et autres) aux charges de travail connectées.

La virtualisation des postes de travail

La virtualisation de poste permet de dé-corréler les éléments logiques (système d'exploitation et applications) des éléments physiques (processeurs, mémoires) du PC traditionnel. Les éléments logiques peuvent cohabiter sur un même serveur centralisé, tout en étant indépendants au niveau du fonctionnement. L'objectif de cette technologie réside dans l'implémentation de la machine virtuelle dans un serveur distant du système ce qui permet à l'utilisateur d'accéder à l'intégralité de ses programmes, applications, processus et données et ce quel que soit le client matériel qu'il utilise [16]

1.5 Data Center

Un Data Center ou centre de données est un lieu regroupant des équipements constituants de système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc). Il peut être interne ou externe à l'entreprise, exploité ou non avec le soutien de prestataires. Les Data Centers ne sont pas déterminés par leur taille physique. Les petites entreprises peuvent utiliser une petite salle où sont juxtaposés plusieurs serveurs et espaces de stockage interconnectés. Les entreprises informatiques de grande envergure, comme Facebook, Amazon ou Google, peuvent quant à elles remplir un immense entrepôt.

Un Data Center comprend en général un contrôle sur l'environnement (climatisation, système de prévention contre l'incendie et les dégâts extérieurs), une alimentation d'urgence et redondante pour les risques de coupure électrique, ainsi qu'une sécurité physique élevée pour les risques

d'intrusion ou encore contre l'accès de personnes malveillantes sur les serveurs (voir la Figure 1.6).

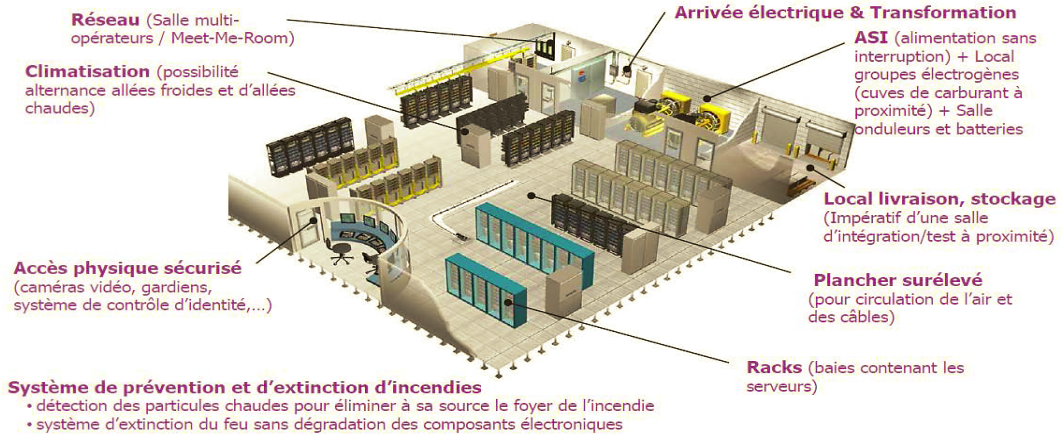

FIGURE 1.6 – Modélisation d'un Data Center.

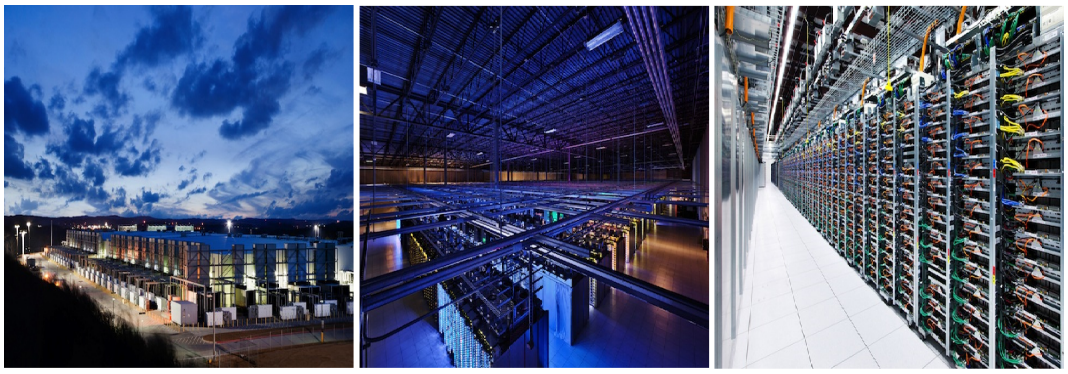

FIGURE 1.7 – Data Center de Google : extérieur, intérieur et racks de serveurs.

1.6 Gestion des niveaux de service

1.6.1 Définition

La gestion des niveaux de service ou SLM (Service-Level Management) est l'un des processus référencé dans la conception des services d'ITIL (Information Technology Infrastructure Library). Fondamentalement, SLM comprend les activités spécifiques d'une organisation dans le but d'assurer un niveau de qualité convenu pour chacun de ses services informatiques fournis aux consommateurs de services [17].

La mise en place d'un SLM a pour but de déterminer les niveaux de services à fournir, afin

de supporter les métiers de l'entreprise. Elle permet de s'assurer que les niveaux demandés ont été atteints. Si les niveaux demandés n'ont pas été atteints, la gestion des niveaux de services initialise des actions pour éradiquer la mauvaise qualité des services fournis aux organisations métiers et elle fournit les éléments indispensables permettant d'identifier les raisons d'un éventuel accord non atteint.

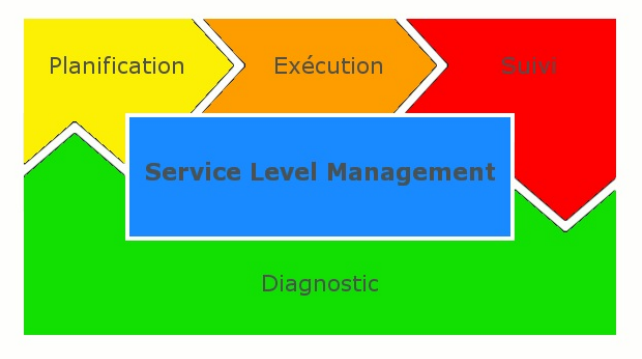

FIGURE 1.8 – Diagramme de gestion des niveaux de service.

Objectifs du SLM

Les objectifs du SLM sont de :

- Définir, surveiller, mesurer et examiner le niveau de service informatique fourni.
- Améliorer les relations et la communication entre les entreprises et les clients.
- Contrôler et améliorer la satisfaction des clients.
- Définir de façon claire et non équivoque le niveau de service à fournir.
- Veiller à l'amélioration continue des niveaux de service, même si tous les objectifs convenus sont déjà atteints.

Rôles du SLM

Concrétement, le responsable de la gestion des niveaux de services prend en charge :

- La préparation, la coordination, la rédaction, la signature, le suivi des SLAs.
- La revue permanente de l'atteinte des objectifs pour s'assurer que la qualité de Service requise et budgétairement justifiable est maintenue et améliorée progressivement.

1.6.2 Accords de niveaux de services

Les accords de niveaux de services ou SLAs sont au cœur de SLM afin de garantir un certain niveau de qualité pour un service informatique donné. Il existe fondamentalement différentes définitions pour le terme SLA. Selon Marilly et al. [18], par exemple, un SLA peut être défini comme suit :
"Un contrat de niveau de service (SLA) est un contrat entre fournisseurs de services ou entre fournisseurs de services et clients qui spécifie, généralement en termes mesurables, les services que le fournisseur de services fournira et les pénalités que le fournisseur de services paiera s'il ne peut pas atteindre les objectifs fixés."

Une définition plus détaillée d'un SLA peut être trouvée dans ITIL [17, 19] :
"Un accord entre un fournisseur de services informatiques et un client. Le SLA décrit le service informatique, documente les objectifs de niveau de service et spécifie les responsabilités du fournisseur de services informatiques et du client. Un seul contrat de niveau de service peut couvrir plusieurs services informatiques ou plusieurs clients."

Objectifs du SLA

Le SLA vise à s'assurer que les attentes et les besoins du client comme du fournisseur soient clairement définis. Il a pour but de prévoir des critères d'évaluation ainsi que des moyens de mesure. Il permet à chaque partie de vérifier mutuellement le respect de leurs engagements. À terme et à condition qu'il soit respecté, le SLA contribue à établir une relation de confiance et/ou de partenariat entre le client et le fournisseur.

Contenu du SLA

Les composants d'une bonne entente de niveau de service doivent comprendre les éléments suivants [20] :

- Le type de service à fournir : il doit spécifier les types de services ainsi que tous les détails de ces derniers. Dans le cas d'une connectivité réseau IP, les types de services doivent décrire les fonctions comme l'utilisation et la maintenance des équipements réseau, la largeur de bande de connexion à fournir, etc.
- Le niveau de performance souhaité des services, en particulier sa fiabilité et sa réactivité : un service fiable est celui qui souffre de perturbations minimes durant un espace de temps spécifique, mais également celui qui est disponible presque tout le temps. Un service avec

une bonne réactivité réalisera les actions rapidement auprès des clients.

- Les étapes à suivre pour signaler les problèmes du service : cette étape a pour but de spécifier les coordonnées à signaler et l'ordre dans lequel les détails sur le problème doivent être communiqués. Le contrat doit également informer sur l'intervalle de temps au cours duquel le problème sera examiné.
- Le temps de réponse et les solutions aux problèmes examinés : le temps de réponse est la période de temps au cours duquel le fournisseur de service va lancer son enquête sur le problème. Le temps de résolution du problème est la période durant laquelle le problème actuel du service sera résolu et corrigé.
- Le suivi des processus et les rapports de niveau de service : ce composant décrit comment les niveaux de performance sont supervisés et surveillés. Ce processus implique la collecte de différents types de statistiques, la fréquence à laquelle ces statistiques seront collectées et la façon dont ces statistiques seront accessibles par les clients.
- Les répercussions pour le fournisseur de services qui ne respecte pas son engagement : si le fournisseur n'est pas en mesure de satisfaire aux exigences énoncées dans le SLA, ce dernier devra faire face aux conséquences pour cet échec. Ces conséquences peuvent inclure le droit du client de résilier le contrat ou de demander un remboursement pour les pertes subies par le client en raison de la défaillance du service.

1.6.3 Qualité de service

Les contrats de niveau de service garantissent qu'un service présente une certaine qualité globale, souvent appelée qualité de service ou QoS (Quality of Service). En outre, la qualité de service n'est pas nécessairement liée aux services logiciels et les définitions existantes diffèrent principalement en ce qui concerne le domaine d'application considéré. Une définition générique de la qualité de service dans le domaine des réseaux de communication est donnée par Schmitt et al. dans [21] : "La qualité de service est le comportement bien défini et contrôlable d'un système en ce qui concerne les paramètres quantitatifs."

Conclusion

Nous venons de voir dans ce chapitre les concepts liés au cloud computing, ses caractéristiques essentielles, ses services, ses modèles de déploiement et ses principaux acteurs. Nous avons également identifié ses avantages et ses inconvénients.

Comme tout est fourni aux utilisateurs du Cloud en tant que services, la qualité de service a un impact important sur la croissance et l'acceptabilité du paradigme du Cloud Computing. Dans les chapitres qui suivent, nous allons voir comment les fournisseurs de services Cloud peuvent garantir aux utilisateurs de service Cloud la qualité de service appropriée.

CHAPITRE 2

MODÈLES DE FILES D'ATTENTE

Introduction

Dans ce chapitre, nous présentons les éléments essentiels de quelques systèmes classiques de files d'attente dont l'étude nous sera nécessaire pour la compréhension des prochains chapitres.

2.1 Chaînes de Markov homogènes

Dans cette section, nous rappelons brièvement les concepts et les propriétés principales des chaînes de Markov.

2.1.1 Processus stochastique

Soit t un paramètre prenant des valeurs dans un ensembre T, soit $X(t)$ une variable aléatoire ou une variable stochastique pour tout t dans T. La famille de variables aléatoires $\{X(t), t \in T\}$ est appelée processus stochastique. Le paramètre t est généralement interprété comme le temps et la variable aléatoire $X(t)$ comme l'état du processus à l'instant t. L'ensembre T peut être dénombrable comme il peut être non dénombrable. Si T est dénombrable on parle de processus discret, s'il est non dénombrable on parle de processus continu. Par exemple, $\{X_n, n = 0, 1, 2, ...\}$ est un prcessus discret et $\{X(t), t \geq 0\}$ est un processus continu. L'ensemble de toutes les valeurs possibles que peut prendre la variable aléatoire $X(t)$ est appelé l'espace des états du processus et sera noté S. Si cet ensemble est fini ou dénombrable, le processus est une chaîne.
Les processus stochastiques peuvent être classés en quatre types :

1. processus à temps discret avec espace d'états discret,
2. processus à temps discret avec espace d'états continu,
3. processus à temps contin avec espace d'états discret,
4. processus à temps continu avec espace d'états continu.

Définition 2.1. Un processus stochastique $\{X(t), t \geq 0\}$ défini sur un ensemble d'états S satisfait la propriété de Markov si, pour tout instant $t \geq 0$ et tout sous-ensemble d'états $I \subseteq S$, on a

$$P\{X_{t+\Delta} \in I | X_u, 0 \leq u \leq t\} = P\{X_{t+\Delta} \in I | X_t\}, \forall \Delta \geq 0. \tag{2.1}$$

2.1.2 Chaîne de Markov à temps discret

Définition 2.2. Une chaîne de Markov à temps discret est un processus stochastique $\{X_n\}$ satisfaisant les trois restrictions suivantes :

1. le processus est à temps discret,
2. l'espace des états S est un ensemble fini ou dénombrable,
3. le processus satisfait la propriété de Markov (2.1).

La probabilité conditionnelle $P\{X_{n+1} = j | X_n = i\} = p_{ij}(n)$ est appelée la probabilité de transition de l'état i vers l'état j à l'instant n.

Définition 2.3. Une chaîne de Markov est dite homogène dans le temps si les probabilités de transition ne sont pas affectées par une translation dans le temps :

$$\Pr\{X_n = j \mid X_{n-1} = i\} = \Pr\{X_{n+m} = j \mid X_{n+m-1} = i\}, \text{quel que soit } m \geq 0.$$

La matrice $P = (p_{ij}), i, j \in S$ est appelée la matrice des probabilités de transition associée à la chaîne de Markov. P est une matrice carrée non-négative avec $\sum_j p_{ij} = 1$ pour tout $i \in S$ et $0 \leq p_{ij} \leq 1$.

Connaissant l'état initial $\pi_i^{(0)} = \Pr\{X_0 = i\}, i \in S$, du processus, nous pouvons trouver la probabilité que la chaîne de Markov sera dans un certain état j à un instant donné n. Nous définissons les probabilités de transition en $n-$étapes $p_{ij}^{(n)}$ comme suit :

$$p_{ij}^{(n)} = P\{X_{r+n} = j | X_r = i\}..$$

La dernière partie de l'équation découle par l'homogénéité. Alors nous avons

$$\Pr\{X_n = j\} = \sum_{i \in S} \pi_i^{(0)} p_{ij}^{(n)}.$$

Théorème 2.1. *[22]*
La probabilité $p_{ij}^{(n)}$ qu'une chaîne de Markov se retrouve dans l'état j après n étapes, si elle se trouve actuellement dans l'état i, est donnée par l'élément (i, j) de la matrice P^n.

Si, partant d'une distribution initiale $\pi^{(0)}$, on peut trouver la distribution $\pi^{(n)}$ des états de la chaîne après n étapes :

$$\pi^{(n)} = \pi^{(0)} P^n.$$

Définition 2.4. Deux états i et j d'une chaîne de Markov **communiquent** (on écrit $i \leftrightarrow j$), s'ils existent $m \geq 0$ `et` $n \geq 0$ tels que $p_{ij}^{(m)} > 0$ `et` $p_{ji}^{(n)} > 0$.

Définition 2.5. Une chaîne de Markov est **irréductible**, si et seulement si, pour tout état i et j, il existe $m \geq 0$ (pouvant dépondre de i et j) tel que

$$p_{ij}^{(m)} > 0.$$

Définition 2.6. Soit P une matrice de transition d'une chaîne de Markov irréductible. (Si la chaîne de Markov est réductible, alors nous pouvons prendre P pour chacune des classes récurrentes).
La période $d = d(i)$ d'un état i est définie comme étant le plus grand diviseur commun de l'ensemble :

$$J_i = \{n \geq 0 : \quad p_{ij}^{(n)} > 0\}.$$

Lorsque $d = 1$ (resp. $d > 1$) pour un état i, alors cet état est **apériodique** (resp. **périodique**). Une chaîne irréductible est apériodique ou a la même période d pour tous ses états. À ce niveau, on peut énoncer l'un des principaux théorèmes :

Théorème 2.2. *[22]*
Si P est une matrice de transition d'une chaîne de Markov irréductible et apériodique. Il existe alors un unique vecteur (ligne) invariant $\pi = (\pi_1, \pi_2, \pi_3, ...)$ de probabilité tels que :

$$\pi P = \pi.$$

De plus , si $\pi^{(0)}$ est le vecteur iniatial des probabilités, alors

$$\lim_{n \to \infty} \pi^{(0)} P^n = \pi,$$

où $\pi_i > 0, \ \forall i \in S$.

2.1.3 Chaîne de Markov à temps continu

Une chaînes de Markov à temps continu est la combinaison d'une chaîne de Markov à temps discret et d'un temps de séjour aléatoire.

Définition 2.7. Une chaîne de Markov à temps continu $\{X(t), t \geq 0\}$ est un processus stochastique à temps continu satisfait la propriété (2.1), l'espace des états est un ensemble fini ou dénombrable et le temps passé dans chaque état est une variable aléatoire réelle positive, suivant une loi exponentielle.

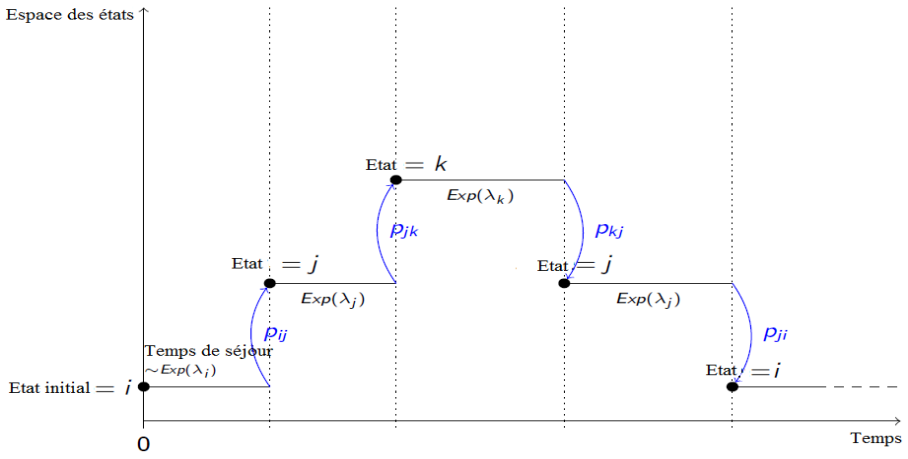

FIGURE 2.1 – La trajectoire d'une chaîne de Markov à temps continu.

Matrice de taux de transition

Des probabilités aux taux :

Une chaîne de Markov à temps continu est caractérisée par deux ensembles de paramètres :

▷ λ_i qui sont l'inverse du temps moyen de séjour dans l'état i.

▷ p_{ij} qui sont les probabilités qu'à un instant donné, le processus passe de l'état i à l'état j.

Malheureusement, ces deux paramètres qui ont une signification intuitive ne sont pas les meilleurs pour l'analyse mathématique des chaînes de Markov à temps continu. Ainsi, nous devons construire des quantités particulières, que nous appellerons les taux de transition :

▷ $q_{ij} = \lambda_i \times p_{ij}$ qui sont des réels positifs et qui peuvent être interprétés comme le nombre moyen de transitions de i à j par unité de temps (pour $i \neq j$).

▷ $q_{ii} = -\lambda_i$ qui sont choisis pour que les rangées somment à zéro, i.e. $\sum_j q_{ij} = 0$.

2.2 Description d'une file d'attente classique

Une file d'attente peut être décrite comme un système stochastique composé d'un certain nombre (fini ou non) de places d'attente d'un ou plusieurs serveurs et de clients arrivant à des instants aléatoires. Quand les serveurs sont tous occupés, les clients doivent alors patienter dans un espace d'attente (s'il existe) jusqu'à ce qu'un serveur soit disponible.

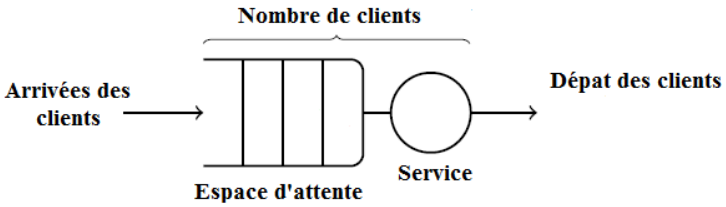

FIGURE 2.2 – Représentation schématique d'une file d'attente simple.

La description précédente d'une file d'attente se fait avec les éléments principaux suivants :

- ▷ **Processus d'arrivée** : Le processus d'arrivée spécifie les instants auxquels les clients arrivent dans le système.
- ▷ **Processus de service** : Les temps de service nécessaires au traitement des clients sont supposés être des réalisations de variables aléatoires indépendantes et identiquement distribuées. La description du processus de service revient alors à préciser la loi de probabilité de ces variables aléatoires.
- ▷ **Nombre de serveurs** : Le nombre de serveurs correspond au nombre maximal de clients pouvant être traités simultanéments. Tous les serveurs sont supposés identiques, en particulier les temps de service sont indépendants d'un serveur à l'autre et distribués selon une même loi de probabilité.
- ▷ **Capacité du système** : La capacité d'accueil d'un système de files d'attente correspond au nombre maximal de clients pouvant être présents dans le système à un instant quelconque. Il est égal à la somme du nombre de serveurs et du nombre de places d'attente disponibles. Si un client arrive dans un système ayant atteint sa capacité maximale d'accueil, il est refoulé et doit quitter le système sans avoir été servi.
- ▷ **Discipline de service** : Ordre de traitement des clients en attente. Les disciplines de service classiques, ainsi que leurs acronymes, sont
  - FIFO : first in first out "premier arrivé, premier servi", c'est la discipline de service

employée le plus souvent et c'est celle qui sera admise par défaut.

- LIFO : last in, first out "dernier arrivé, premier servi".
- SIRO : service dans un ordre aléatoire.
- RR : round robin "les clients sont servis à tour de rôle pendant un intervalle de temps fixe, appelé quantum".
- PS : processor sharing "cas limite de la discipline RR lorsque le quantum tend vers zéro".

La liste suivante résume les lois de probabilité les plus utilisés pour décrire les processus d'arrivées et de services ainsi que les symboles associés :

- ▷ M : loi exponentielle (Markovienne).
- ▷ D : loi constante (cas déterministe).
- ▷ E_k : loi Erlang d'ordre k.
- ▷ Hk : loi hyper-exponentielle ordre k.
- ▷ GI : loi générale indépendante.
- ▷ G : loi générale.

Notation de Kendall

La notation de Kendall permet de ramener la description textuelle des différents éléments constituant un modèle de files d'attente simple à une formule symbolique [23]. Cette dernière est définie par la notation suivante :

$$A/B/c/k/P - D,$$

où

- ▷ A nature du processus d'arrivée,
- ▷ B nature du processus de service,
- ▷ c nombre de serveurs,
- ▷ K capacité maximale du modèle de files d'attente
- ▷ P taille de la population,
- ▷ D discipline de service.

Dans sa version courte, seuls les trois premiers symboles A/B/c sont utilisés. Dans un tel cas, on suppose que la discipline est FIFO et que le nombre de places d'attente est illimité.

2.3 Analyse mathématique d'un modèle de files d'attente

L'étude mathématique d'un système de files d'attente se fait généralement par l'introduction d'un processus stochastique, défini de façon appropriée. Le plus souvent, on s'intresse au nombre $X(t)$ de clients se trouvant dans le système à l'instant t $(t \geq 0)$.

En fonction des quantités qui définissent le système, on cherche à déterminer :

▷ Le régime transitoire du processus stochastique $\{X(t), t \geq 0\}$, défini par les probabilités d'état :

$$P_n(t) = P(X(t) = n).$$

▷ Le régime stationnaire du processus stochastique $\{X(t), t \geq 0\}$, défini par les distributions stationnaires de ce processus :

$$\pi_n = \lim_{t \to \infty} P_n(t) = P(X(+\infty) = n) = P(X = n), (n = 0, 1, 2, ...).$$

Le calcul explicite du régime transitoire s'avère généralement pénible, voire impossible, pour la plupart des modèles donnés. On se contente donc de déterminer le régime stationnaire.

2.4 Modèles d'attente markoviens

Dans cette section, nous présentons brièvement quelques modèles d'attente markoviens. Ces modèles sont caréctérisés par les deux quantités stochastiques principales, le temps des inter-arrivées et la durée de service, qui sont des variables aléatoires indépendantes exponentiellement distribuées. La propriété d'absence de mémoire de la loi exponentielle facilite l'étude de ces modèles.

2.4.1 File d'attente $M/M/1$

Le système d'attente $M/M/1$ est décrit par les spécificités suivantes. Les clients se présentent au système aléatoirement selon un processus de Poisson de taux λ. Le temps de service suit une loi exponentielle de taux μ, indépendamment d'un client à l'autre. Un serveur et la capacité du système est infinie. La discipline de service est FIFO. Représentant l'état de ce système à un instant quelconque par le nombre de clients présents dans le système, le graphe des transitions possibles entre ses différents états correspont à la Figure 2.3.

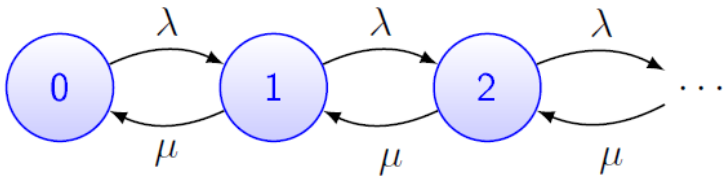

FIGURE 2.3 – Graphe représentatif d'une file $M/M/1$.

Les équations de balance s'écrivent :

$$\begin{aligned}
\lambda\pi_0 &= \mu\pi_1 \\
(\lambda+\mu)\pi_1 &= \lambda\pi_0 + \mu\pi_2 \\
(\lambda+\mu)\pi_2 &= \lambda\pi_1 + \mu\pi_3 \\
\vdots \quad &= \quad \vdots \qquad \vdots \\
(\lambda+\mu)\pi_n &= \lambda\pi_{n-1} + \mu\pi_{n+1}
\end{aligned}$$

De ce système d'équations linéairement dépendantes, on tire

$$\begin{aligned}
\pi_1 &= \frac{\lambda}{\mu}\pi_0 \\
\pi_2 &= \left(\frac{\lambda}{\mu}\right)^2\pi_0 \\
\vdots \quad &= \quad \vdots \\
\pi_n &= \left(\frac{\lambda}{\mu}\right)^n\pi_0, \qquad n \geq 0
\end{aligned}$$

En utilisant la condition de normalisation $\sum_{n=0}^{\infty}\pi_n = 1$, on obtient pour $\rho = \frac{\lambda}{\mu} < 1$ (l'intensité du trafic),

$$\pi_0 = (1-\rho) \quad \text{et} \quad \pi_n = (1-\rho)\rho^n, \qquad n = 0, 1, ...$$

Partant de cette distribution stationnaire, la plupart des performances moyennes de ce modèle d'attente peuvent être calculées. Le taux d'utilisation du serveur est égal à la probabilité que le système ne soit pas vide :

$$U = \sum_{n=1}^{\infty}\pi_n = 1 - \pi_0 = \rho.$$

Le nombre moyen de clients présents dans le système est :

$$\overline{N} = \sum_{n=0}^{\infty} n\pi_n = \frac{\rho}{1-\rho},$$

et le nombre moyen de clients en attente est :

$$\overline{Q} = \sum_{n=1}^{\infty}(n-1)\pi_n = \frac{\rho^2}{1-\rho}.$$

Le calcul des temps moyens de séjour (ou de réponse) $\overline{T}$ et d'attente $\overline{W}$ d'un client dans le système se fait à l'aide de la formule de Little :

$$\overline{T} = \frac{\overline{N}}{\lambda} = \frac{1}{\mu - \lambda},$$

$$\overline{W} = \frac{\overline{Q}}{\lambda} = \frac{\rho}{\mu - \lambda}.$$

2.4.2 Propriété PASTA

Lorsqu'ils arrivent selon un processus de Poisson, les clients voient le système à l'état stationnaire à leur arrivée. Ainsi la probabilité qu'un client voit la file dans l'état x à son arrivée est égale à $\pi(x)$. D'après le théorème ergodique, $\pi(x)$ est également la fraction du temps que le système passe dans l'état x. On appelle donc cette propriété PASTA, pour Arrival Poisson See Time Averages. Cette propriété clé des processus de Poisson, liée au fait que les arrivées sont totalement indépendantes les unes des autres, de sorte que la connaissance de temps d'arrivée d'un client ne donne aucune information sur les temps d'arrivées des autres clients, et en particulier des clients arrivés précédemment.

2.4.3 File d'attente $M/M/c/k$

Le système d'attente $M/M/c/k$ possède c serveurs identiques, la durée d'un service est une variable aléatoire distribuée selon une loi exponentielle de paramètre μ et les clients arrivent au système suivant un processus de Poisson de taux λ. La capacité du système est limitée. Si un client arrive alors que la capacité k du système est déjà atteinte, il est refoulé et repart immédiatement sans avoir été servi.

Le graphe représentatif de la file $M/M/c/k$ est donné en Figure 2.4. Il correspont à un processus de naissance et de mort défini par les taux de naissance

$$\lambda_k = \lambda, \qquad n = 0, 1, ..., k-1.$$

et les taux de mort

$$\mu_n = \begin{cases} n\mu, & n = 0, 1, ..., c-1 \\ c\mu, & n = c, c+1, ..., k. \end{cases}$$

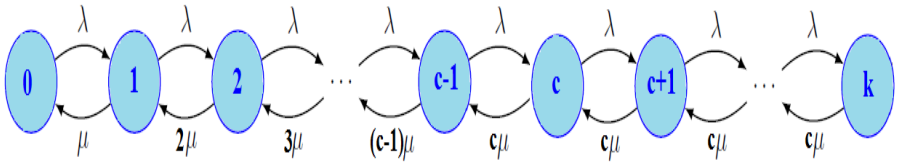

FIGURE 2.4 – Graphe représentatif d'une file $M/M/c/k$.

Distribution stationnaire

Comme la file d'attente $M/M/c/k$ possède un nombre fini d'états, alors le processus markovien décrivant l'évolution du nombre de clients dans le système est toujours ergodique, donc le système est stable quels que soient les taux d'arrivées λ et de service μ. Ainsi, la distribution stationnaire existe et unique. La probabilité d'avoir n clients dans le système est donnée comme suit :

$$\pi_n = \begin{cases} \frac{(c\rho)^n}{n!}\pi_0, & n = 0, 1, ..., c-1 \\ \\ \frac{\rho^n c^c}{c!}\pi_0, & n = c, c+1, ..., k \end{cases}$$

avec l'intensité du traffic $\rho = \frac{\lambda}{c\mu}$. L'utilisation de la condition de normalisation $\sum_{n=0}^{k} \pi_n = 1$ nous permet de calculer la probabilité d'observer le système vide, π_0, comme suit :

$$\pi_0 = \begin{cases} \left[1 + \frac{(1-\rho^{k-c+1})(c\rho)^c}{c!(1-\rho)} + \sum_{n=1}^{c-1} \frac{(c\rho)^n}{n!}\right]^{-1} & \text{if } \rho \neq 1; \\ \\ \left[1 + \frac{c^c}{c!}(k-c+1) + \sum_{n=1}^{c-1} \frac{(c)^n}{n!}\right]^{-1} & \text{if } \rho = 1. \end{cases}$$

Partant de cette distribution stationnaire, le nombre moyen de clients présents dans le système est :

$$\overline{N} = \sum_{n=1}^{k} n\pi_n = \frac{\rho}{1-\rho},$$

et le nombre moyen de clients en attente est :

$$\overline{Q} = \sum_{n=c+1}^{k} (n-c)\pi_n = \frac{\rho^2}{1-\rho}.$$

Le calcul des temps moyens de séjour (ou de réponse) $\overline{T}$ et d'attente $\overline{W}$ d'un client dans le système se fait à l'aide de la formule de Little :

$$\overline{T} = \frac{\overline{N}}{\lambda'},$$

$$\overline{W} = \frac{\overline{Q}}{\lambda'}.$$

où $\lambda' = \sum_{n=0}^{k-1} \lambda(1 - \pi_k)$ est le taux effectif d'arrivée des clients au système.

Le taux d'utilisation de chaque serveur est donné par :

$$U = \frac{\lambda'}{c\mu}.$$

2.5 Modèles d'attente non-markoviens

En l'absence de l'exponentialité ou plutôt lorsque l'on s'écarte de l'hypothèse d'exponentialité de l'une des deux quantités stochastiques : le temps des inter-arrivées et la durée de service, ou en prenant en compte certaines spécificités des problèmes par introduction de paramètres supplémentaires, on aboutit à un modèle non-markovien. La combinaison de tous ces facteurs rend l'étude mathématique du modèle très délicate. On essaye alors de se ramener à un processus de Markov judicieusement choisi à l'aide de quelques méthodes d'analyse à savoir :

- ▷ **Méthode des étapes d'Erlang** : Son principe est d'approximer toute loi de probabilité ayant une transformation de Laplace rationnelle par une loi de Cox (mélange de lois exponentielles), cette dernière possède la propriété d'absence de mémoire par étape.
- ▷ **Méthode de la chaîne de Markov induite** : Cette méthode, élaborée par Kendall [24], est souvent utilisée. Elle consiste à choisir une séquence d'instants $1, 2, 3, \ldots n$ (déterministes ou aléatoires) telle que la chaîne induite $\{X_n, \geq 0\}$, où $X_n = X(n)$, soit markovienne et homogène.
- ▷ **Méthode des variables supplémentaires** : Elle consiste à compléter l'information sur le processus $\{X(t), t \geq 0\}$ de telle manière à lui donner le caractère markovien. Ainsi, on se ramène à l'étude du processus $\{X(t), A(t_1), A(t_2), A(t_3), ..., A(t_n), t \geq 0\}$. Les variables $A(t_k)$, $k \in \{1, 2, 3, ..., n\}$ sont dites supplémentaires.
- ▷ **Méthode des événements fictifs** : Le principe est d'introduire des événements fictifs qui permettent de donner une interprétation probabiliste aux transformées de Laplace et aux variables aléatoires décrivant le système étudié.

▷ **Simulation** : C'est un procédé d'imitation artificielle d'un processus réel effectué sur ordinateur. Elle nous permet d'étudier les systèmes les plus complexes, de prévoir leurs comportements et de calculer leurs caractéristiques. Les résultats obtenus ne sont qu'approximatifs, mais peuvent être utilisés avec une bonne précision. Cette technique se base sur la génération de variables aléatoires suivant les lois gouvernant le système.

Dans ce travail, nous allons opté pour la méthode de la chaîne de Markov induite pour l'étude mathématique de nos modèles non-markoviens proposés pour l'évaluation des performances des Cloud Data Centers.

2.5.1 File d'attente $M/G/1$

Description du modèle

Les clients arrivent au système selon un processus de Poisson de taux λ. De ce fait, le temps entre deux arrivées successives suit une loi exponentielle de moyenne $\frac{1}{\lambda}$. Le service est assuré par un seul serveur. A l'arrivée d'un client, si le serveur est libre, le client sera pris en charge immédiatement. Dans le cas contraire, il rejoint la file d'attente (de capacité illimitée), les durées de service Y sont des variables aléatoires indépendantes et identiquement distribuées de loi générale dont la fonction de répartition $H(y)$ et la transformée de Laplace-Stieltjes $H^*(s)$. La discipline de service est FIFO.

Chaîne de Markov induite

Considérons le processus $\{X(t)\}$ aux instants t_1, t_2, t_3,... où les clients terminent leurs services et quittent le système. On définit ainsi un processus stochastique à temps discret $\{X_n = X(t_n), n = 1, 2, ...\}$ où t_n est l'instant de départ du n-ième client.

Soit A_n le nombre de clients entrant dans le système pendant le service du n-ième client. Les variables aléatoires A_n sont indépendantes entre elles, leur distribution commune est

$$P(A_n = k) = a_k = \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} dH(t), \tag{2.2}$$

où $a_k \geq 0$ et $k = 0, 1, 2, ...$

Il est clair que :

$$X_{n+1} = \begin{cases} X_n + 1 - A_{n+1}, & \text{si} \quad X_n \geq 1; \\ A_{n+1}, & \text{si} \quad X_n = 0, \end{cases} \tag{2.3}$$

avec $n \geq 1$. On peut écrire $X_{n+1} = X_n - \delta_n + A_{n+1}$, avec :

$$\delta_n = \begin{cases} 1, & \text{si} \quad X_n \geq 1; \\ 0, & \text{si} \quad X_n = 0. \end{cases} \tag{2.4}$$

X_{n+1} ne dépend que de X_n et de A_{n+1} et non pas des valeurs de X_{n-1} X_{n-2} X_{n-3},..., donc la suite $\{X_n, n \geq 1\}$ est une chaîne de Markov induite du processus $\{X(t), t \geq 0\}$. Ses probabilités de transition $p_{ij} = P(X_{n+1} = j | X_n = i)$ se calculent par [25] :

$$p_{ij} = \begin{cases} a_j, & \text{si} \quad j \geq 0, i = 0; \\ a_{j-i+1}, & \text{si} \quad 1 \leq i \leq j+1, \\ 0, & \text{ailleurs.} \end{cases} \tag{2.5}$$

La matrice des probabilités de transition prend la forme [26] :

$$\boldsymbol{P} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & \dots \\ a_0 & a_1 & a_2 & a_3 & a_4 & \dots \\ 0 & a_0 & a_1 & a_2 & a_3 & \dots \\ 0 & 0 & 0 & a_0 & a_1 & \dots \\ . & . & . & . & . & \dots \end{pmatrix}.$$

Régime stationnaire

On définit l'intensité du trafic par $\rho = \frac{\lambda}{\mu}$ et soit $\boldsymbol{\pi} = (\pi_0, \pi_1, \pi_2, ...)$ la distribution stationnaire associée au processus $\{X_n, n \geq 1\}$. On peut montrer que pour qu'un régime stationnaire s'établit, la condition $\rho < 1$ doit être férifiée.

Mesures de performance

Le nombre moyen de clients dans le système :

$$L = \rho + \frac{\rho^2}{2(1-\rho}(1 + \mu^2\sigma^2), \tag{2.6}$$

où σ^2 est la variation de la loi de service $H(t)$

Le nombre moyen de clients en attente :

$$L_q = \frac{\rho^2}{2(1-\rho}(1 + \mu^2\sigma^2). \tag{2.7}$$

Le temps moyen d'attente d'un client dans la file :

$$W_q = \frac{\rho^2}{2\mu(1-\rho}(1 + \mu^2\sigma^2). \tag{2.8}$$

Le temps moyen de séjour d'un client dans le système :

$$T = \frac{1}{\mu} + \frac{\rho^2}{2\mu(1-\rho}(1+\mu^2\sigma^2). \tag{2.9}$$

2.5.2 File d'attente $M/G/c$

Description du modèle

Des clients arrivent au système suivant un processus de Poisson. Cela signifie que la durée séparant deux arrivées successives $A(x) \triangleq P[X \leq x]$ suit une loi exponentielle de paramètre $\lambda(> 0)$, de densité de probabilité $a(x) = \lambda e^{-\lambda x}$ et de transformée de Laplace :

$$A^*(s) = \int_0^\infty e^{-sx} a(x)dx = \frac{\lambda}{\lambda + s}. \tag{2.10}$$

Les durées de service sont des variables indépendantes identiquement distribuées de loi générale. On note $H(y) \triangleq P[Y \leq y]$ la fonction de répartition de cette loi avec $E(H) = \frac{1}{\mu}$ le temps moyen de service et $h(y)$ sa densité de probabilité. Sa transformée de Laplace est donnée par :

$$H^*(s) = \int_0^\infty e^{-sy} h(y)dy. \tag{2.11}$$

L'intensité du trafic s'exprime généralement de la manière suivante :

$$\rho \triangleq \frac{\lambda E(H)}{c}.$$

Quand la capacité du système est finie, on parle d'un modèle de files d'attente avec perte.

2.5.3 Problèmes et notes bibliographiques

Des résultats et formulations théoriques sont bien établis pour les modèles de files d'attente markoviens. Mais pas pour les modèles non-markoviens dont l'étude analytique est très complexe. En effet, pour la majorité des modèles non-markoviens, il n'existe aucune formule analytique exacte des indicateurs de performances.

Le calcul de certains indicateurs de performance de quelques modèles de files d'attente, tels que les $M/G/1$, $M/G/c$ et $G/G/c$, devient simple si on peut estimer la probabilité d'attente, i.e. la probabilité qu'un client qui arrive attende avant d'être servi, et évaluer les temps de service résiduels des clients en cours de service. Les temps de service résiduels sont définis comme les temps restants pour terminer les services des clients en cours de service au moment où un nouveau client entre dans le système. En général, dans le cas des modèles de files d'attente mono-serveur, on s'intéresse à la valeur moyenne des temps de service résiduels, appelée le résidu de service,

et dans le cas des modèles d'attente multi-serveurs, on s'intéresse à la valeur moyenne du plus petit des résidus de service pour l'ensemble des clients en cours de service.

Pour les modèles d'attente $M/G/1$, le temps de service résiduel d'un client est en moyenne égal à $\frac{(1+cv^2)}{2}E(H)$ [27], où cv est le coefficient de variation de temps de service. Cette expression permet d'obtenir des formules simples pour les indicateurs de performance moyens de la file $M/G/1$. Cependant, pour les modèles de files d'attente à multi-serveurs avec durées de service générales, il n'existe aucune expression exacte du minimum des résidus. Ainsi, le calcul des mesures de performance de tels modèles a été mené dans plusieurs travaux de recherche par différentes méthodes. Dans [28–31], les auteurs ont proposé des solutions approchées pour le calcul du minimum des résidus en adaptant la formule exacte pour la file $M/G/1$ au cas multi-serveur. Dans [32], l'auteur a appliqué la méthode de l'approximation de diffusion pour fournir des formules approximatives des distributions du nombre de clients, du temps d'attente et de la période d'occupation dans la file $M/G/c$. En outre, dans [33], l'auteur a décrit approximativement la distribution de la longueur de la file d'attente à l'état stationnaire dans la file $M/G/c$ avec un espace d'attente fini. Une approche similaire dans le contexte du modèle $M/G/c$ a été décrite par [34], mais a été étendue de manière à approcher la probabilité de blocage et, ainsi, à déterminer la plus petite capacité de buffer, de sorte que le taux des clients perdues reste sous un niveau prédéfini. Dans [35], en se basant sur la solution exacte du modèle $M/M/c/k$, l'auteur a proposé une approximation différente pour la probabilité de blocage. Dans [36], en utilisant une excellente approximation de temps moyen d'attente dans la file $M/G/c$, l'auteur a fourni une approximation plus précise de la probabilité d'attente pour des petites valeurs de nombre de serveurs.

Cependant, la plupart de ces approximations présentent des limitations et elles ne conviennent pas à l'évaluation des performances des Cloud Data Centers :

(i) par exemple, les approximations proposées dans [33,35,36] sont raisonnablement précises lorsque le nombre de serveurs est petit (< 10), alors que les Cloud Data Centers contiennent des centaines voire des milliers de serveurs ;

(ii) Les approximations proposées par [28,37] sont imprécises lorsque le coefficient de variation de temps de service est supérieur à 1 ;

(iii) Les erreurs d'approximation sont particulièrement prononcées lorsque l'intensité du trafic ρ est faible et/ou quand le nombre de serveurs c est grand, et le coefficient de variation du temps de service est supérieur à 1 ([32,38,39].

Conclusion

Dans ce chapitre, nous avons rappelé et présenté les notions et techniques de base sur les systèmes de files d'attente classique. Nous avons réalisé une étude bibliographique liée aux modèles non-markoviens à plusieurs serveurs et nous avons conclu que la plupart des approximations proposées dans la littérature sont dans certains cas imprécises et ne conviennent pas pour l'évaluation des performances des Cloud Data Centers. Dans les chapitres qui suivent, nous allons faire des propositions pour atténuer ces limitations.

Deuxième partie

Contributions

CHAPITRE 3

ÉVALUATION DES PERFORMANCES DU CLOUD DATA CENTER VIA LE MODÈLE DE FILES D'ATTENTE M/G/C/K

Introduction

En raison de la nature de l'environnement du Cloud Computing, il est souvent difficile d'obtenir un modèle approprié pour l'évaluation des performances du Cloud Data Center. Khazaei et al. ont proposé l'application du système de files d'attente $M/G/c/k$ pour l'évaluation des performances du Cloud Data Center [40]. La recherche théorique sur les files d'attente a montré que la résolution analytique de ce système reste, à ce jour, un problème ouvert et difficile car une solution analytique exacte sous forme fermée de sa distribution strationnaire est difficile à atteindre, ce qui nécessite des approximations appropriées pour obtenir des résultats satisfaisants. Cependant, les approximations proposées dans la littérature présentent des limitations, soit parce qu'lles ne conviennent pas à l'évaluation des performances des Cloud Data Centers (voir par exemple [28,33,35], [38]), soit parce qu'lles n'introduisent pas d'une manière appropriée le processus stochastique décrivant l'évolution du système dans le temps (voir [41] et [42]). Dans ce chapitre, nous présentons une nouvelle approximation améliorée permettant de mieux décrire le processus stochastique du système de files d'attente $M/G/c/k$. En effet, nous proposons de nouvelles formules explicites pour calculer les différents éléments de la matrice des probabilités de transition de la chaîne de Markov induite associée à ce système. Afin d'examiner la précision de nos formules approximatives, nous les testons numériquement sur quelques exemples. Puis, nous calculons la distribution stationnaire et certains indicateurs de performance tels que

la probabilité de blocage, le temps moyen de réponse, la probabilité de service immédiat et la probabilité d'attente.

3.1 Synthèse bibliographique

Le Cloud Computing a fait l'objet de nombreuses recherches tant dans le monde universitaire que dans l'industrie. Étant donné que ce nouveau paradigme est une informatique orientée service, l'analyse des performances du service Cloud devient un enjeu important. En raison de la nature dynamique et virtualisée des environnements du Cloud Computing, de la diversité des demandes des utilisateurs et de la dépendance temporelle de la charge, fournir la QoS attendue tout en évitant le sur-approvisionnement n'est pas une tâche simple [43]. Pour s'assurer que la QoS perçue par les clients est acceptable, les fournisseurs de services Cloud doivent exploiter des techniques et des mécanismes garantissant un niveau minimal de cette QoS, tout en maximisant leur profit. Dans cette section, nous allons examiner et analyser les propositions développées dans ce domaine.

Dans [43], **Xiong et Perros** ont proposé un modèle de réseau de files d'attente pour étudier les performances des services informatiques Cloud. Ils ont développé une méthode approximative permettant de calculer la transformation de Laplace de la distribution de temps de réponse. En utilisant cette distribution, les auteurs ont trouvé la relation entre le nombre maximum de clients, le nombre minimum de ressources et le plus haut niveau de service afin de fournir des services avec une QoS garantie.

Yang et al. [44] ont modélisé le Cloud Data Center comme un modèle de files d'attente $M/M/c/k$ à partir duquel la distribution de temps de réponse a été déterminée. Dans ce travail, les auteurs ont divisé le temps de réponse en trois périodes : la période d'attente, de service et d'exécution. Ils ont supposé que ces trois périodes sont indépendantes ; ce qui est irréaliste, selon l'argument des auteurs.

Vilaplana et al. [45] ont analysé la conception d'une architecture du Cloud avec les exigences de QoS. Dans ce travail, les Cloud plates-formes ont été modélisées par un réseau de Jackson ouvert afin de déterminer et de mesurer la QoS garantie par le Cloud en termes de temps de réponse. Les auteurs ont montré que leur modèle peut être très utile pour optimiser les performances du service.

Dans [46], afin d'analyser les performances des services Cloud, les auterurs **Guo et al.** ont modélisé le Cloud Data Ceter comme un modèle de files d'attente $M/M/c$ et ils ont développé

une méthode d'optimisation synthétique pour optimiser les performances. Afin de valider leur méthode d'optimisation, ils ont fait une étude de simulation. Les résultats de simulation ont montré que la méthode proposée peut permettre moins de temps d'attente, une longueur de file d'attente plus petite et plus de clients peuvent obtenir le service.

Eisa et al. [47] ont proposé un modèle pour la planification du Cloud Computing basé sur plusieurs files d'attente ($M/M/1$ et $M/M/c$), ce qui a permis d'améliorer la QoS en minimisant le temps d'exécution par tâche, le temps d'attente et le coût des ressources nécessaires pour répondre aux besoins des utilisateurs. Dans ce travail, les auteurs ont indiqué que leur modèle augmente l'utilisation du planificateur global et réduit le temps d'attente ; ce qui a été montré par les résultats expérimentaux.

Dans [48], **Ani Brown Mary and Saravanan** ont modélisé le Cloud Data Center par un modèle de files d'attente [(M/G/1) :(∞/GDMODEL)]. Ils ont évalué les performances du système à l'aide de méthodes analytiques et ils ont obtenu quelques indicateurs de performance, tels que le nombre moyen de tâches, la probabilité de blocage et la probabilité de service immédiat.

Kamble and Channe [49] ont proposé une approche pour décomposer le temps de réponse dans les Cloud Data Ceters modélisé par un simple modèle de files d'attente ainsi que un algorithme de simulation correspondant.

Ben el Aattar et al. [50] ont proposé un modèle analytique pour l'évaluation des performances d'un Cloud Data Center en le modélisant par un modèle de files d'attente $GE/G/c/k$. En raison de la nature de l'environnement du Cloud, les auteurs ont considéré que les arrivées suivent une distribution exponentielle généralisée (GE). Ils ont fourni des formules analytiques pour les indicateurs de performance tels que le nombre moyen de tâches dans le système, la probabilité de blocage, la probabilité de service immédiat et le temps moyen de réponse.

Ellens et al. [51] ont examiné le problème général de l'approvisionnement de ressources dans le Cloud Computing. Ils ont analysé le problème d'allocation des ressources à différents clients de manière à respecter les SLAs pour tous ces clients. Ils ont proposé un modèle de files d'attente $M/M/c/c$ avec des classes de priorités différentes pour évaluer les performances d'un Cloud Data Center lorsque plusieurs SLAs sont négociés entre le fournisseur de services et les clients. Pour chaque classe, le contrat SLA est spécifié par les probabilités de rejet des demandes des clients de cette classe.

Anupama and Keerthi [52] ont proposé un modèle de files d'attente avec un nombre infini de serveurs pour l'analyse de performances de service Cloud afin de réduire le temps d'attente et la longueur de la file d'attente.

Dans [53], les auteurs **Sowjanya et al.** ont montré que l'utilisation des systèmes avec plusieurs serveurs permet d'augmenter la performance d'un service Cloud par rapport à l'utilisation des systèmes à unique serveur. En effet, ils ont comparé le temps d'attente dans un système modélisé par la file d'attente $M/M/2$ par rapport à un système modélisé par la file d'attente $M/M/1$; et leurs résultats numériques montrent clairement que la file $M/M/2$ réduise la longueur de la file d'attente et le temps d'attente par rapport à la file $M/M/1$.

Lakshmi and Bindhu [54] ont proposé un modèle de files d'attente $M/M/c$ avec plusieurs serveurs afin de réduire les temps d'attente et la longueur de la file d'attente, et d'améliorer les performances du réseau et la qualité de service dans l'environnement du Cloud Computing.

Dans [55], **Firdhous et al.** ont modélisé un système du Cloud Computing à l'aide de la théorie des files d'attente, plus précisément à l'aide des formules d'Erlang C. Quatre modèles de différentes complexités ont été présentés par la combinaison de plusieurs files d'attente $M/M/c$ en fonction de la configuration choisie. Les modèles présentés ont été simulés afin de caractériser la performance des modèles. Les performances des systèmes ont été analysées du point de vue des clients plutôt que du point de vue des fournisseurs. Ainsi, seuls les temps de réponse de différentes configurations ont été étudiés dans ce travail.

Nan et al. [56] ont étudié les problèmes d'allocation de ressources pour des services multimédias différenciés. Ils ont proposé tout d'abord un modèle de files d'attente pour caractériser les services différenciés dans le Cloud Data Center. Sur la base de ce modèle, ils ont optimisé les ressources informatiques dans le scénario du premier arrivé premier servi (FCFS) et le scénario de priorité. Dans chaque scénario, ils ont formulé et résolu le problème optimale d'allocation de ressources afin de minimiser le coût des ressources dans les limites de temps de réponse. Les résultats de la simulation montrent que les schémas d'allocation de ressources proposés peuvent utiliser les ressources du Cloud de manière optimale pour fournir des services satisfaisants pour différentes classes de demandes avec un coût minimal.

Dans [57], **Murugesan, et al.** ont proposé un modèle approximatif pour évaluer les performances d'un Cloud Data Center en utilisant le modèle de files d'attente $M/G/c$. Ils ont décrit une nouvelle approximation permettant d'évaluer les mesures de performance du Cloud Data Center (par exemple la distribution de probabilité de temps de réponse des demandes de sevices).

Circonstances d'application : Cloud Data Center		
Réf.	Modélisation du système	Mesures de performance évaluées
[43]	$M/M/1$	Temps de réponse
[44]	$M/M/c/k$	Temps de réponse
[45]	Réseau de Jackson	Temps de réponse
[46]	$M/M/c$	Temps d'attente
[47]	$M/M/1$ et $M/M/c$	Temps d'exécution, Temps d'attente
[48]	(M/G/1) :(∞/GDMODEL)	Probabilités de blocage et de service immédiat
[49]	$M/M/1$, Algorithme de simulation	Temps de réponse
[50]	$GE/G/m/k$	Probabilités de blocage et de service immédiat, Temps de réponse
[51]	$M/M/c/c$ avec différentes classes de priorité	Probabilités de rejet des demandes des clients de chaque classe
[52]	$M/M/1$ et $M/M/\infty$	Temps d'attente, La longueur de la file d'attente
[53]	$M/M/1$ et $M/M/2$	Temps d'attente, La longueur de la file d'attente
[54]	$M/M/c$	Temps d'attente, La longueur de la file d'attente
[55]	$M/M/c$	Temps de réponse
[56]	$M/M/1$ et $M/H/1$	Minimiser le coût des ressources dans les limites de temps de réponse
[57]	$M/G/c$	Distribution de probabilité de temps de réponse

TABLE 3.1 – Comparaison des méthodes de modélisation analytique pour l'évaluation de performances du service Cloud

3.1.1 Comparaison des méthodes de modélisation analytique pour l'évaluation de performances du service Cloud

Le tableau 3.1 présente une comparaison des méthodes d'évaluation basées sur la modélisation analytique examinées ci-dessus, y compris les mesures de performance évaluées, les modèles analytiques utilsés et les circonstances d'application des méthodes.

La plupart des travaux cités ci-dessus supposent une durée de service distribuée selon une loi exponentielle lors de la modélisation des systèmes de service Cloud. Une telle hypothèse, bien que simplifiant la modélisation et l'analyse, ne représente pas avec précision le temps de service réaliste des infrastructures Cloud. Considérant l'hétérogénéité des technologies de mise en œuvre pour la fourniture des services Cloud, la distribution générale serait plus appropriée pour la modélisation de temps de service d'un serveur Cloud. Mais, l'hypothèse d'une distribution de temps de service générale peut conduire à une plus grande complexité d'analyse.

Comme nous l'avons montré dans le chapitre précédent, fournir une solution analytique exacte aux modèles de files d'attente $M/G/c$ n'est pas évident, ce qui a conduit à l'utilisation des approximations. Cependant, les approximations proposées dans la littérature présentent des limitations et elles ne permettent pas de représenter certaines caractéristiques spéciales du Cloud Computing, par exemple un grand nombre de serveurs. Par conséquent, les auteurs

de [41] et [58] ont proposé une nouvelle approximation qui est suffisamment précise dans les cas de grand nombre de serveurs et lorsque la distribution de temps de service a un coefficient de variation supérieur à un. Cette approche est basée sur la transformation de Laplace et la technique de la chaîne de Markov induite pour calculer la matrice des probabilités de transition de la file d'attente $M/G/c/k$. La matrice obtenue a été divisée en quatre régions et les auteurs ont proposé des formules approximatives pour calculer les probabilités de transition de chaque région. Cependant, dans ce travail, les transitions entre les états du système exprimées en probabilités conditionnelles ne sont pas décrites avec précision.

Le problème de calcul de la matrice des probabilités de transition de la file d'attente $M/G/c/k$ a également été abordé dans [42]. Dans ce travail, les auteurs ont proposé d'autres formules approximatives pour le calcul des probabilités de transition dans les régions 3 et 4. Cela a abouti à une matrice stochastique pour la chaîne de markov induite du système $M/G/c/k$ uniquement lorsque le temps de service suit une distribution Gamma. Une telle hypothèse est inappropriée, car la matrice doit être stochastique quelle que soit la distribution des temps de service considérées (puisque les temps de service suivent une loi quelconque).

Dans ce qui suit, nous présentons les lacunes des formules approximatives proposées dans [41] et [42]. En outre, nous proposons des améliorations visant à raffiner les approximations susmentionnées.

3.2 Description du modèle d'attente M/G/c/k

Nous modélisons le Coud Data Center par un système de files d'attente $M/G/c/k$. Dans ce système, les demandes de service Cloud (tâches) arrivent au Data Center suivant un processus de Poisson. Cela signifie que la durée séparant deux arrivées successives de tâches $A(x) \triangleq P[X \leq x]$ suit une loi exponentielle de paramètre $\lambda(> 0)$, de densité de probabilité $a(x) = \lambda e^{-\lambda x}$ et de transformée de Laplace :

$$A^*(s) = \int_0^\infty e^{-sx} a(x) dx = \frac{\lambda}{\lambda + s}. \tag{3.1}$$

Les durées de service sont des variables indépendantes identiquement distribuées selon une loi de probabilités quelconque, d'une fonction de répartition $H(y) \triangleq P[Y \leq y]$, d'un temps moyen de service $\overline{h} = E(H)$ et d'une transformée de Laplace :

$$H^*(s) = \int_0^\infty e^{-sy} h(y) dy. \tag{3.2}$$

Ce système contient c (≥ 1) serveurs qui rendent service dans l'ordre d'arrivée. La capacité du système k ($= c + r$) est finie, ce qui signifie que la capacité du buffer est égale à r.

Le temps de service résiduel est noté par H_+ et sa transformée de Laplace est donnée par [59] comme suit :

$$H_+^*(s) = \frac{1 - H^*(s)}{s\ \overline{h}}. \tag{3.3}$$

L'intensité du trafic s'exprime généralement de la manière suivante :

$$\rho \triangleq \frac{\lambda \overline{h}}{c}.$$

3.3 Analyse mathématique du modèle d'attente M/G/c/k

3.3.1 Chaîne de Markov induite

Le système de files d'attente $M/G/c/k$ est un système non-markovien [60] qui peut être analysé en utilisant la technique de la chaîne de Markov induite similaire à celle proposée dans [41]. Cette technique consiste à choisir les instants t_n de l'arrivée de la n-ième tâche au système. En effet, si on énumère ces instances par $0, 1, 2, \ldots$, on aura à étudier une chaîne de Markov à temps discret $\{X_n = X(t_n), n \geq 0\}$ avec un espace d'état $S = \{0, 1, 2, ..., k\}$, où X_n représente le nombre de tâches trouvées dans le système immédiatement avant t_n et

$$X_{n+1} = \begin{cases} X_n + 1 - B_{n+1}, & \text{si} \quad X_n < k; \\ X_n - B_{n+1}, & \text{si} \quad X_n = k, \end{cases} \tag{3.4}$$

où B_{n+1} est le nombre de tâches quittant le système durant le temps d'inter-arrivée $T = t_{n+1} - t_n$ (voir la Figure 3.1).

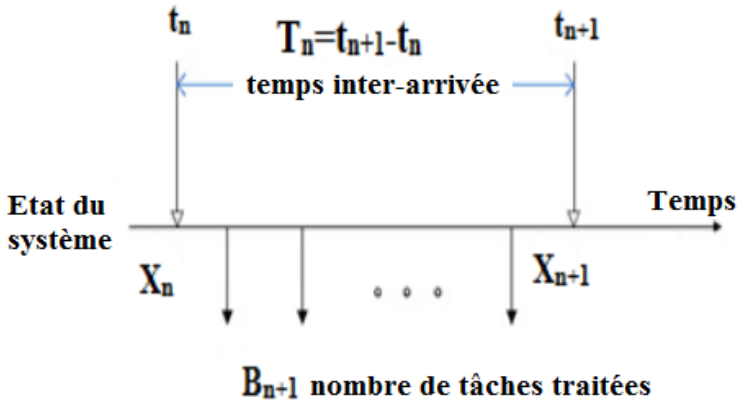

FIGURE 3.1 – Chaîne de Markov induite.

3.3.2 Régime stationnaire

La chaîne de Markov $\{X_n, n \geq 0\}$ est homogène et ergodique (voir [41]), donc la distibution stationnaire du nombre de tâches trouvées dans le système aux instants d'arrivée existe et unique. Soit $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_k)$ la distribution stationnaire associée au processus $\{X_n, n \geq 0\}$. Le calcul direct de cette distribution stationnaire revient à la résolution du système d'équations linéaires suivant :

$$\begin{cases} \boldsymbol{\pi} P = \boldsymbol{\pi} \\ \boldsymbol{\pi} \mathbf{1} = 1, \end{cases} \tag{3.5}$$

où P est une matrice dont les éléments sont des probabilités de transition p_{ij} en une étape et $\mathbf{1}$ est un vecteur colonne d'éléments égaux à 1.

Avant de calculer la distribition stationnaire $\boldsymbol{\pi}$ de la chaîne de Markov induite, nous devont premièrement calculer la matrice des probabilités de transition associée à cette chaîne.

3.3.3 Matrice des probabilités de transition

Les probabilités de transition de la matrice P sont définies par :

$$p_{ij} \triangleq P(X_{n+1} = j | X_n = i). \tag{3.6}$$

En prenant en cosidération les points suivants :

(i) p_{ij} représente la probabilité d'avoir exactement $(i + 1 - j)$ tâches traitées pendant T, lorsque nous avons $i < k$;

(ii) étant donné la définition de X_n, évidemment

$$p_{ij} = 0 \quad \text{pour tout} \quad j > i + 1; \tag{3.7}$$

(iii) si la n-ième arrivée trouve le système dans l'état k (le système est complet), alors p_{kj} représente la probabilité que $(i - j)$ tâches ont été traitées pendant T. De même, si la n-ième arrivée trouve le système dans l'état $k-1$, alors p_{k-1j} représente la probabilité que $(i - j)$ tâches ont été traitées pendant T. Ainsi, on aura

$$p_{kj} = p_{k-1j} \quad \text{pour} \quad \text{tout} \quad j;$$

(iv) si on définit $b_\omega = P(B_{n+1} = \omega)$ comme la probabilité d'avoir ω tâches traitées pendant T, alors $b_\omega \triangleq p_{ij}$;

on aura la matrice des probabilités de transition P de la chaîne de Markov induite définie comme suit :

$$
P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ c-1 \\ c \\ \vdots \\ c+r-2 \\ c+r-1 \\ c+r \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} 0 & 1 & 2 & \dots & c-1 & c & \dots & c+r-2 & c+r-1 & c+r \end{array} \\
\left(\begin{array}{ccccccccccc}
b_1 & b_0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\
b_2 & b_1 & b_0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\
b_3 & b_2 & b_1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\
b_c & b_{c-1} & b_{c-2} & \dots & b_1 & b_0 & \dots & 0 & 0 & 0 \\
b_{c+1} & b_c & b_{c-1} & \dots & b_2 & b_1 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\
b_{c+r-1} & b_{c+r-2} & b_{c+r-3} & \dots & b_r & b_{r-1} & \dots & b_1 & b_0 & 0 \\
b_{c+r} & b_{c+r-1} & b_{c+r-2} & \dots & b_{r+1} & b_r & \dots & b_2 & b_1 & b_0 \\
b_{c+r} & b_{c+r-1} & b_{c+r-2} & \dots & b_{r+1} & b_r & \dots & b_2 & b_1 & b_0
\end{array}\right)
\end{array}
$$

Region 2 — Region 1 — Region 4 — Region 3

Comme on peut le voir, cette matrice a quatre régions. Avant de calculer b_ω dans chaque région, nous définissons d'abord les probabilités de départ.

Probabilités de départ

On définit la probabilité de terminer le service d'une tâche, qui a déjà été en service au cours de l'intervalle d'observation précédent, dans l'intervalle en cours comme :

$$P_x \triangleq P(A > H_+) = H_+^*(\lambda), \tag{3.8}$$

et la probabilité d'achever le service d'une tâche durant le même intervalle d'observation comme suit :

$$P_y \triangleq P(A > H) = H^*(\lambda). \tag{3.9}$$

Si un serveur termine le service d'une tâche qui a déjà commencé au cours de l'intervalle d'observation précédent, dans l'intervalle actuel, ce serveur sera inactif. Si la file d'attente n'est pas vide, ce serveur peut également effectuer un deuxième service dans l'intervalle en cours, et si la file d'attente est toujours non vide, un nouveau service peut être achevé, et ainsi de suite jusqu'à ce que la file d'attente soit vide. Ainsi, la probabilité que k services soient complétés par un seul serveur est donnée par la formule suivante :

$$\begin{aligned} P_{z,k} &= \Big[\prod_{i=2}^{k} P\Big(A > H | A > (k-i)H + H_+\Big)\Big] \\ &\quad \times P(A > H_+). \end{aligned} \tag{3.10}$$

Avec ces probabilités de départ, nous pouvons décrire les quatre différentes régions de la matrice des probabilités de transition P.

Région 1 : De la formule (4.1), nous avons $p_{ij} = 0$ pour $i + 1 < j$.

Région 2 : Pour $i < c$, $j \leq c$, et $i + 1 \geq j$, toutes les tâches sont en service (pas d'attente). La probabilité que ω tâches soient traitées pendant T est donnée par :

$$\begin{aligned} p_{ij} &= \binom{i}{i-j} P_x^{i-j}(1-P_x)^j P_y + \\ &\quad \binom{i}{i-j+1} P_x^{i-j+1}(1-P_x)^{j-1}(1-P_y). \end{aligned} \tag{3.11}$$

Région 3 : Pour $c \leq i \leq k$, $c \leq j \leq k$, et $i + 1 \geq j$, tous les serveurs sont occupés durant T. Afin de minimiser l'erreur, nous supposons que chaque serveur ne traite pas plus de trois services de tâches entre deux arrivées successives. Par conséquent, la probabilité que ω tâches soient traitées pendant T dans cette région est donnée par :

$$\begin{aligned} p_{ij} &= \sum_{s_1=min(\omega,1)}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,1)}^{min(\omega-s_1,s_1)} \binom{s_1}{s_2} P_{z,2}^{s_2}(1-P_{z,2})^{s_1-s_2} \\ &\quad \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2} \\ &\quad (1-P_{z,3})^{s_2-(w-s_1-s_2)}\Phi, \end{aligned} \tag{3.12}$$

où Φ est la fonction indicatrice :

$$\Phi = \begin{cases} 1, & si \;\; \omega - s_1 - s_2 \leq s_2; \\ \\ 0, & si \;\; \omega - s_1 - s_2 > s_2. \end{cases} \tag{3.13}$$

Comme il a été souligné dans (iii), $p_{kj} = p_{k-1j}$ pour tout j lorsque $i = k$.

Région 4 : Pour $c \leq i \leq k$, $j < c$, et $i + 1 \geq j$, tous les serveurs sont occupés au début de l'intervalle T et $(c - j)$ serveurs sont inactifs à la fin de T. Ainsi, la probabilité que ω tâches soient traitées pendant T est donnée par :

$$\begin{aligned} p_{ij} &= \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,min(s_1,i-c+1))} \binom{s_1}{s_2} P_{z,2}^{s_2} \\ &\quad (1-P_{z,2})^{min(s_1,i-c+1)-s_2} \\ &\quad \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2} \\ &\quad (1-P_{z,3})^{max(i-c+1-s_1,0)-(\omega-s_1-s_2)}\Phi. \end{aligned} \tag{3.14}$$

Compte tenu également du point (iii) dans cette région, nous aurons $p_{kj} = p_{k-1j}$ pour tout j lorsque $i = k$.

3.3.4 Discussion

Les formules qu'on a proposé pour les deux régions 1 et 2 sont identiques à celles proposées dans [41]. Cependant, en raison de la particularité du comportement du modèle dans les régions 3 et 4, nous avons adopté une analyse plus précise qui donne de nouvelles formules approximatives pour ces régions.

Examinons maintenant en détail ces nouvelles formules et comparons-les à celles proposées par [41].

Région 3 ($\mathbf{c \leq i \leq k, c \leq j \leq k, i+1 \geq j}$) :
Dans cette région, la n-ième arrivée trouve tous les c serveurs sont occupés et $(i-c)$ tâches dans le buffer. Si le nombre de tâches dans le système est strictement inférieur à k (i.e. $i < k$), alors la n-ième arrivée pourra entrer dans le système. Par conséquent, il devrait y avoir $(i-c+1)$ tâches dans le buffer au début de l'intervalle T. Parmi ces c serveurs, s_1 d'entre eux accomplissent au moins un service pendant T. Parmi ces s_1 serveurs, s_2 d'entre eux termineront un deuxième service pendant T. Comme chaque serveur n'accomplit pas plus de trois services entre deux arrivées successives de tâches, alors les $(\omega - s_1 - s_2)$ tâches restantes doivent être traitées avant la fin de l'intervalle T ; Ces tâches seront traitées par un sous-ensemble de s_2 serveurs. Le nombre de serveurs dans ce sous-ensemble est égal à $(\omega - s_1 - s_2)$, ça veut dire, il y a $(\omega - s_1 - s_2)$ serveurs dont chacun achève exactement trois services durant T et, le nombre de serveurs qui sont toujours occupés à traiter le troisième service devrait être égal à $s_2 - (\omega - s_1 - s_2)$. Notez que ce nombre de serveurs est égal à s_2 dans la formule approximative proposée par [41], avec une probabilité $(1 - P_{z,3})^{s_2}$. En effet, selon ces auteurs, cette formule approximative est donnée comme suit :

$$\begin{aligned} p_{ij} \quad = \quad & \sum_{s_1=min(\omega,1)}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{m-s_1} \\ & \sum_{s_2=min(\omega-s_1,1)}^{min(\omega-s_1,s_1)} \binom{s_1}{s_2} P_{z,2}^{s_2}(1-P_{z,2})^{s_1-s_2} \\ & \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2}(1-P_{z,3})^{s_2}. \end{aligned} \tag{3.15}$$

Cependant, il est impossible de trouver exactement, dans cette formule, c serveurs occupés à la fin de l'intervalle T, car le nombre de serveurs qui sont encore en train de traiter le troisième

service est égal à s_2. Par conséquent, une nouvelle formule a été proposée dans [42] et elle est définie comme suit :

$$\begin{aligned} p_{ij} &= \sum_{s_1=min(\omega,1)}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,1)}^{min(\omega-s_1,s_1)} \binom{s_1}{s_2} P_{z,2}^{s_2}(1-P_{z,2})^{s_1-s_2} \\ &\quad \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2} \\ &\quad (1-P_{z,3})^{s_2-(w-s_1-s_2)}. \end{aligned} \tag{3.16}$$

En outre, dans les formules (3.15) et (3.16), toutes les $(\omega - s_1 - s_2)$ tâches restantes qui doivent quitter le système devraient être traitées par le sous-ensemble de s_2 serveurs. Lorsque le nombre de ces s_2 serveurs est petit, il peut arriver que le nombre de tâches restantes $(\omega - s_1 - s_2)$ dépasse s_2, cela est dû à l'hypothèse que "`chaque serveur n'accomplit pas plus de trois services de tâches pendant` T". Donc, pour prendre en compte l'effet de cette hypothèse dans notre formule, nous avons défini la fonction d'indicatrice Φ qui est donnée dans la formule (3.13).

Aussi, dans cette région, quand la n-ième arrivée trouve le système complet (i.e. $i = k$), elle sera perdue. Par conséquent, il y aura $(i - j)$ tâches qui vont quitter le système entre deux arrivées successives de tâches au lieu de $(i + 1 - j)$ tâches. Cela n'a pas été pris en compte dans les formules (3.15) et (3.16). Pour y remédier, nous avons défini notre chaîne de Markov induite par la formule (3.4).

Les considérations prises en compte lors de notre analyse de la région 3 seront maintenues dans notre analyse de la région 4.

Région 4 ($\mathbf{c \leq i \leq k, 0 \leq j < c, i + 1 \geq j}$) :

Dans cette région, au début de l'intervalle T, il y a $(i - c + 1)$ tâches dans le buffer et tous les c serveurs sont occupés, tandis qu'à la fin de T, le buffer est vide et il y a $(c - j)$ serveurs inactifs.

La formule approximative proposée dans [41] pour cette région est donnée par :

$$\begin{aligned} p_{ij} &= \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,c-j)}^{min(w-s_1,s_1)} \binom{s_1}{s_2} P_{z,2}^{s_2}(1-P_{z,2})^{s_1-s_2} \\ &\quad \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{w-s_1-s_2}(1-P_{z,3})^{s_2}. \end{aligned} \tag{3.17}$$

Dans cette formule, les auteurs de [41] n'ont pas pris en compte le nombre de tâches dans le buffer au début de T. Par conséquent, à la fin de T, le nombre de serveurs inactifs diffère de $(c-j)$, c'est-à-dire que certains serveurs sont occupés alors qu'il n'y a pas de tâches à traiter, ce qui est contradictoire.

Compte tenu du nombre de tâches dans le buffer au début de T, on distinguera deux cas à étudier, à savoir :

- **Case 1** : Si $\mathbf{s_1 \geq i - c + 1}$, alors toutes les $(i-c+1)$ tâches dans le buffer vont être traitées par les s_1 serveurs qui ont terminé leurs premiers services durant T et il y aura $\big(s_1 - (i-c+1)\big)$ serveurs inactifs car le buffer est vide.
- **Case 2** : Si $\mathbf{s_1 < i - c + 1}$, il y aura s_1 tâches qui vont être traitées par les s_1 serveurs qui ont terminé leurs premiers services durant T et $\big((i-c+1) - s_1\big)$ tâches en attente de service.

Les auteurs de [42] ont proposé une nouvelle formule approximative pour cette région :

$$\begin{aligned} p_{ij} \quad = \quad & \frac{1}{(1-P_{z,3})^{c-j}} \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1} (1-P_x)^{c-s_1} \\ & \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,s_1)} \binom{min(i-c+1,s_1)}{s_2} P_{z,2}^{s_2} \\ & (1-P_{z,2})^{s_1-s_2} \binom{min(max(i-c+1-s_1,0),s_2)}{\omega-s_1-s_2} \\ & P_{z,3}^{\omega-s_1-s_2} (1-P_{z,3})^{s_2-max(i-c+1-s_1-s_2,0)}. \end{aligned} \tag{3.18}$$

Dans cette formule, les auteurs ont considéré que le nombre de serveurs qui sont toujours en train d'achever le troisième service est égal à $(s_2 - max(i-c+1-s_1-s_2, 0))$, car ils ont supposé que tous les serveurs qui entrent dans l'état inactif durant T sont toujours en état d'activité. Afin d'avoir exactement $(c-j)$ serveurs inactifs à la fin de T, ils ont multiplié leur formule par $\frac{1}{(1-P_{z,3})^{c-j}}$.

Dans le cas où $s_1 \geq i - c + 1$, la formule (3.18) sera égale à :

$$\begin{aligned} p_{ij} \quad = \quad & \frac{1}{(1-P_{z,3})^{c-j}} \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1} (1-P_x)^{c-s_1} \\ & \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,s_1)} \binom{i-c+1}{s_2} P_{z,2}^{s_2} (1-P_{z,2})^{s_1-s_2} \\ & \binom{min(max(i-c+1-s_1,0),s_2)}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2} \\ & (1-P_{z,3})^{s_2-max(i-c+1-s_1-s_2,0)}. \end{aligned} \tag{3.19}$$

Donc $max(i - c + 1 - s_1, 0) = 0$, $min(max(i - c + 1 - s_1, 0), s_2) = 0$ et $max(i - c + 1 - s_1 - s_2, 0) = 0$.

Comme le buffer devient vide dans ce cas (car toutes les $(i - c + 1)$ tâches dans la file d'attente entrent en service comme $s_1 \geq i - c + 1$) alors le nombre de serveurs qui vont achever trois services durant T est égal à 0 (i.e. $\omega - s_1 - s_2$=0). Par conséquent, le nombre de serveurs qui seront inactifs à la fin de T est égal à s_2, c-à-d, $c - j = s_2$. Ainsi, la formule (3.19) sera égale à :

$$\begin{aligned} p_{ij} &= \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,s_1)} \binom{i-c+1}{s_2} P_{z,2}^{s_2} \\ &\quad (1-P_{z,2})^{s_1-s_2}. \end{aligned} \tag{3.20}$$

Dans cette formule, si le nombre de serveurs qui achèvent le second service est égal à $(i - c + 1)$, alors à la fin de T, le nombre de serveurs qui n'ont pas encore achevé leur second service est égal à $(i - c + 1 - s_2)$, tandis que dans la formule (3.20), ce nombre est égal à $(s_1 - s_2)$. Par conséquent, le nombre de tâches dans le système à la fin de T dépasse j. Ainsi, le coefficient $\frac{1}{(1-P_{z,3})^{c-j}}$ n'est pas valide lorsque $s_1 \geq i - c + 1$, mais il est valide juste quand $s_1 < i - c + 1$.

Compte tenu de l'analyse ci-dessus pour cette région, nous étudions séparément les deux cas sus-cités :

- Dans le cas 1, s_2 tâches parmi $(i - c + 1)$ tâches quittent le système avec une probabilité $P_{z,2}^{s_2}$ et $(i - c + 1 - s_2)$ tâches restent en service avec une probabilité $(1 - P_{z,2})^{i-c+1-s_2}$. En d'autres termes, les s_2 serveurs qui vont accomplir un second service doivent être sélectionnés parmi les s_1 serveurs qui ont achevé leurs premiers services durant T $\left(\text{i.e. } \binom{s_1}{s_2}\right)$, et le nombre maximum de ces serveurs est égal à $min(\omega - s_1, i - c + 1)$. La formule qu'on a proposé pour le calcul des probabilités de transition dans ce cas est donnée par :

$$\begin{aligned} p_{ij} &= \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ &\quad \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,i-c+1)} \binom{s_1}{s_2} P_{z,2}^{s_2} \\ &\quad (1-P_{z,2})^{i-c+1-s_2}. \end{aligned} \tag{3.21}$$

- Dans le cas 2, comme $s_1 < i - c + 1$, alors s_2 tâches parmi s_1 tâches quittent le système avec une probabilité $P_{z,2}^{s_2}$, $(s_1 - s_2)$ tâches restent en service avec une probabilité

$(1-P_{z,2})^{s_1-s_2}$, et $\big((i-c+1)-s_1\big)$ tâches en attente de service. Ces dernières seront traitées par un sous-ensemble de s_2 serveurs qui ont effectué deux services durant T. Parmi ces $\big((i-c+1)-s_1\big)$ tâches, $(\omega-s_1-s_2)$ tâches quittent le système avec une probabilité $P_{z,3}^{\omega-s_1-s_2}$ et $\big((i-c+1-s_1)-(\omega-s_1-s_2)\big)$ tâches restent en service avec une probabilité $(1-P_{z,3})^{(i-c+1-s_1)-(\omega-s_1-s_2)}$. Ainsi, nous avons proposé une formule approximative pour calculer les probabilités de transition dans ce cas comme suit :

$$\begin{aligned} p_{ij} \;=\; & \sum_{s_1=c-j}^{min(\omega,c)} \binom{c}{s_1} P_x^{s_1}(1-P_x)^{c-s_1} \\ & \sum_{s_2=min(\omega-s_1,c-j)}^{min(\omega-s_1,s_1)} \binom{s_1}{s_2} P_{z,2}^{s_2}(1-P_{z,2})^{s_1-s_2} \\ & \binom{s_2}{\omega-s_1-s_2} P_{z,3}^{\omega-s_1-s_2} \\ & (1-P_{z,3})^{(i-c+1-s_1)-(\omega-s_1-s_2)}\Phi. \end{aligned} \tag{3.22}$$

En combinant les deux formules (3.21) et (3.22), nous avons obtenu la formule (3.14) pour calculer les probabilités de transition dans la région 4.

Cette analyse nous a permis de proposer une nouvelle matrice des probabilités de tansition plus appropriée pour la file d'attente $M/G/c/k$.

3.4 Exemple d'application

Afin de confirmer nos résultats théoriques, nous considérons dans cette section l'exemple de système d'attente $M/G/2/4$ pour lequel nous allons calculer la matrice P en utilisant les formules approximatives qu'on a proposé et celle proposées dans [41].

Soit $\widetilde{P}$ la matrice des probabilités de transition de la file d'attente $M/G/2/4$ trouvée en utilisant nos formules approximatives :

$$\widetilde{P} = \begin{array}{c} \\ 0 \\ 1 \\ \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \left(\begin{array}{ccccc} & & & 0 & 0 \\ & \widetilde{A} & & 0 & 0 \\ \hline & & & & \\ \widetilde{B} & & & \widetilde{C} & \\ & & & & \end{array}\right) \end{array},$$

où :

$$\widetilde{\boldsymbol{A}} = \begin{pmatrix} P_y & 1-P_y & 0 \\ P_xP_y & \Big[(1-P_x)P_y + P_x(1-P_x)\Big] & (1-P_x)(1-P_y) \end{pmatrix},$$

$$\widetilde{\boldsymbol{B}} = \begin{pmatrix} 2P_x^2P_{z,2} & \Big[2P_x(1-P_x)P_{z,2} + P_x^2 \times(1-P_{z,2})\Big] \\ P_x^2P_{z,2}^2 & \Big[2P_x(1-P_x)P_{z,2}P_{z,3} + 2P_x^2 \times P_{z,2}(1-P_{z,2})\Big] \\ P_x^2P_{z,2}^2 & \Big[2P_x(1-P_x)P_{z,2}P_{z,3} + 2P_x^2 \times P_{z,2}(1-P_{z,2})\Big] \end{pmatrix},$$

$$\widetilde{\boldsymbol{C}} = \begin{pmatrix} 2P_x(1-P_x)(1-P_{z,2}) & (1-P_x)^2 & 0 \\ \Big[2P_x(1-P_x)P_{z,2} \times(1-P_{z,3})\Big] + \Big[P_x^2(1-P_{z,2})^2\Big] & \Big[2P_x(1-P_x)\Big] \times\Big[(1-P_{z,2})\Big] & (1-P_x)^2 \\ \Big[2P_x(1-P_x)P_{z,2} \times(1-P_{z,3})\Big] + \Big[P_x^2(1-P_{z,2})^2\Big] & \Big[2P_x(1-P_x)\Big] \times\Big[(1-P_{z,2})\Big] & (1-P_x)^2 \end{pmatrix}.$$

Soit $\widetilde{\widetilde{P}}$ la matrice des probabilités de transition de la file d'attente $M/G/2/4$ trouvée en

utilisant les formules approximatives proposées dans [41] :

$$\widetilde{\widetilde{P}} = \begin{array}{c} \\ 0 \\ 1 \\ \\ 2 \\ \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \quad 4 \end{array} \\ \left(\begin{array}{cc|cc} & & 0 & 0 \\ & \widetilde{\widetilde{A}} & 0 & 0 \\ \hline & & & \\ \widetilde{\widetilde{B}} & & \widetilde{\widetilde{C}} & \end{array} \right) \end{array},$$

où :

$$\widetilde{\widetilde{\boldsymbol{A}}} = \begin{pmatrix} P_y & 1-P_y & 0 \\ P_x P_y & (1-P_x)P_y + P_x(1-P_x) & (1-P_x)(1-P_y) \end{pmatrix},$$

$$\widetilde{\widetilde{\boldsymbol{B}}} = \begin{pmatrix} \Big[2P_x^2 P_{z,2}(1-P_{z,2})\Big] \times \Big[(1-P_{z,3})\Big] & \Big[2P_x(1-P_x)P_{z,2}(1-P_{z,3})\Big] + \Big[P_x^2(1-P_{z,2})^2\Big] \\ P_x^2 P_{z,2}^2(1-P_{z,3})^2 & \Big[2P_x(1-P_x)P_{z,2}P_{z,3} \times (1-P_{z,3})\Big] + \Big[2P_x^2 P_{z,2} (1-P_{z,2})(1-P_{z,3})\Big] \\ 2P_x^2 P_{z,2}^2 P_{z,3}(1-P_{z,3})^2 & \Big[2P_x^2 P_{z,2}(1-P_{z,2})P_{z,3} \times (1-P_{z,3})\Big] + \Big[P_x^2 P_{z,2}^2(1-P_{z,3})^2\Big] \end{pmatrix},$$

$$\widetilde{\widetilde{C}} = \begin{pmatrix} \Big[2P_x(1-P_x) & (1-P_x)^2 & 0 \\ \times(1-P_{z,2})\Big] & & \\ \\ \Big[2P_x(1-P_x) & \Big[2P_x(1-P_x) & (1-P_x)^2 \\ \times P_{z,2}(1-P_{z,3})\Big] & \times(1-P_{z,2})\Big] & \\ +\Big[P_x^2(1-P_{z,2})^2\Big] & & \\ \\ \Big[2P_x(1-P_x)P_{z,2} & & \\ \times P_{z,3}(1-P_{z,3})\Big] & \Big[2P_x(1-P_x) & \Big[2P_x(1-P_x) \\ +\Big[2P_x^2P_{z,2} & \times P_{z,2}(1-P_{z,3})\Big] & \times(1-P_{z,2})\Big] \\ \times(1-P_{z,2}) & +\Big[P_x^2(1-P_{z,2})^2\Big] & \\ \times(1-P_{z,3})\Big] & & \end{pmatrix}.$$

Comme nous le savons, une matrice des probabilités de transition est une matrice stochastique par définition. Ainsi, notre principale contribution réside dans l'obtention d'une matrice stochastique, qui ne peut être obtenue en utilisant les formules approximatives proposées dans [41], que nous confirmerons dans ce qui suit.

Dans une matrice stochastique, la somme des probabilités de transition de l'état i vers tous les autres états doit être égale à 1. Donc, s'il y a au moins un état i tel que cette somme soit différente de 1, alors cette matrice n'est plus stochastique.

Calculons par exemple la somme de la ligne 4 pour $\widetilde{P}$ (resp. $\widetilde{\widetilde{P}}$) :

$$\begin{aligned} \sum_{j=0}^{4} \widetilde{p}_{4j} &= 1 - 2P_xP_{z,2}P_{z,3} + 2P_x^2P_{z,2}P_{z,3} + 2P_xP_{z,2}P_{z,3} - \\ &\quad 2P_x^2P_{z,2}P_{z,3} = 1. \\ \sum_{j=0}^{4} \widetilde{\widetilde{p}}_{4j} &= 2P_x - P_x^2 - 2P_{z,2}P_{z,3} + 2P_x^2P_{z,2}P_{z,3} - 2P_x^2P_{z,2}P_{z,3}^2 - \\ &\quad P_x^2P_{z,2}^2P_{z,3}^2 + 2P_x^2P_{z,2}^2P_{z,3}^3 + 2P_{z,2}P_{z,3} - 2P_{z,2}P_{z,3}^2 - \\ &\quad 2P_x^2P_{z,2}P_{z,3} + 2P_x^2P_{z,2}P_{z,3}^2. \\ &= 2P_x - P_x^2 - P_x^2P_{z,2}^2P_{z,3}^2 + 2P_x^2P_{z,2}^2P_{z,3}^3 - 2P_{z,2}P_{z,3}^2. \end{aligned}$$

Maglré que nos formules sont approximatives, nous avons obtenu, dans ce cas, $\sum_{j=0}^{4} \widetilde{p}_{4j}$ est exactement égale à 1, ce qui explique que nos formules approximatives sont plus précises.

$\sum_{j=0}^{4} \widetilde{\widetilde{p}}_{4j}$ ne peut être vérifiée que par une analyse numérique, pour cela, nous donnons quelques exemples numériques où nous supposons différentes distributions de temps de service avec la même intensité du trafic ρ supposée dans [41].

Distribution exponentielle

Exemple 3.1. *Les durées de service suivent la loi exponentielle de paramètre* $\mu = 0.6$. *On considère* $\rho = 0.85$ *:*

$$\widetilde{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.37 & 0.62 & 0 & 0 & 0 \\ 0.13 & 0.46 & 0.39 & 0 & 0 \\ 0.03 & 0.18 & 0.40 & 0.39 & 0 \\ 0.00 & 0.03 & 0.16 & 0.40 & 0.39 \\ 0.00 & 0.03 & 0.16 & 0.40 & 0.39 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 1.00 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \end{array}.$$

$$\widetilde{\widetilde{P}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.37 & 0.62 & 0 & 0 & 0 \\ 0.13 & 0.46 & 0.39 & 0 & 0 \\ 0.03 & 0.16 & 0.40 & 0.39 & 0 \\ 0.00 & 0.03 & 0.16 & 0.40 & 0.39 \\ 0.00 & 0.00 & 0.03 & 0.16 & 0.40 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \\ \sum_j = 0.59 \end{array}.$$

Distribution d'Erlang

Exemple 3.2. *Les durées de service suivent la loi d'Erlang avec paramètre de forme* $a = 4$, *paramètre d'échelle* $\mu = 0.6$, *et* $\rho = 0.85$ *:*

$$\widetilde{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.24 & 0.75 & 0 & 0 & 0 \\ 0.10 & 0.47 & 0.41 & 0 & 0 \\ 0.04 & 0.23 & 0.44 & 0.30 & 0 \\ 0.00 & 0.04 & 0.20 & 0.44 & 0.30 \\ 0.00 & 0.04 & 0.20 & 0.44 & 0.30 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 1.01 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \end{array}.$$

$$\widetilde{\widetilde{P}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.24 & 0.75 & 0 & 0 & 0 \\ 0.10 & 0.47 & 0.41 & 0 & 0 \\ 0.03 & 0.20 & 0.44 & 0.30 & 0 \\ 0.00 & 0.03 & 0.20 & 0.44 & 0.30 \\ 0.00 & 0.00 & 0.03 & 0.20 & 0.44 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.97 \\ \sum_j = 0.97 \\ \sum_j = 0.67 \end{array}.$$

Distribution Weibull

Exemple 3.3. *Les durées de service suivent la loi de Weibull avec paramètre de forme* $a = 0.8$*, paramètre d'échelle* $\mu = 0.6$*, et* $\rho = 0.85$ *:*

$$\widetilde{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.35 & 0.64 & 0 & 0 & 0 \\ 0.09 & 0.43 & 0.47 & 0 & 0 \\ 0.01 & 0.10 & 0.35 & 0.53 & 0 \\ 0.00 & 0.01 & 0.09 & 0.35 & 0.53 \\ 0.00 & 0.01 & 0.09 & 0.35 & 0.53 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.99 \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \end{array}.$$

$$\widetilde{\widetilde{P}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.35 & 0.64 & 0 & 0 & 0 \\ 0.09 & 0.43 & 0.47 & 0 & 0 \\ 0.01 & 0.09 & 0.35 & 0.53 & 0 \\ 0.00 & 0.01 & 0.09 & 0.35 & 0.53 \\ 0.00 & 0.00 & 0.01 & 0.09 & 0.35 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \\ \sum_j = 0.45 \end{array}.$$

Distribution Gamma

Exemple 3.4. *Les durées de service suivent la loi Gamma avec paramètre de forme* $a = 0.2$*,*

paramètre d'échelle $\mu = 0.6$, *et* $\rho = 0.85$ *:*

$$\widetilde{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.63 & 0.36 & 0 & 0 & 0 \\ 0.13 & 0.57 & 0.28 & 0 & 0 \\ 0.01 & 0.08 & 0.28 & 0.61 & 0 \\ 0.00 & 0.01 & 0.07 & 0.28 & 0.61 \\ 0.00 & 0.01 & 0.07 & 0.28 & 0.61 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.98 \\ \sum_j = 0.97 \\ \sum_j = 0.97 \end{array}$$

$$\widetilde{\widetilde{P}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0.63 & 0.36 & 0 & 0 & 0 \\ 0.13 & 0.57 & 0.28 & 0 & 0 \\ 0.01 & 0.07 & 0.28 & 0.61 & 0 \\ 0.00 & 0.01 & 0.07 & 0.28 & 0.61 \\ 0.00 & 0.00 & 0.01 & 0.07 & 0.28 \end{pmatrix} \end{array} \begin{array}{c} \\ \sum_j = 0.99 \\ \sum_j = 0.98 \\ \sum_j = 0.97 \\ \sum_j = 0.97 \\ \sum_j = 0.36 \end{array}$$

A partir de ces exemples, on remarque que la somme des éléments de chaque ligne pour les matrices obtenues par nos formules approximatives, est égale ou très proche de 1, en opposition à celle obtenue en utilisant les formules approximatives de [41], qui est très loin de 1, voir en particulier la dernière ligne.

Cet exemple d'application prouve que nos matrices sont stochastiques, ce qui confirme que nos formules approximatives sont plus précises.

3.5 Evaluation de performances

Dans cette section, nous allons résoudre les équations de balance du modèle proposé en utilisant MATLAB pour obtenir la distribution stationnaire. Ensuite, nous calculerons quelques indicateurs de performance tels que la probabilité de blocage, le temps moyen de réponse, la probabilité de service immédiat et la probabilité d'attente. Enfin, nous validerons les résultats obtenus par simulation.

3.5.1 Equations de balance

Après avoir calcué la matrice des probabilités de transition P, nous pouvons établir les équations de balance suivantes :

$$\pi_i = \sum_{j=0}^{j=k} \pi_j p_{ij}, \qquad 0 \leq i \leq k, \tag{3.23}$$

En raison de l'ergodicité du système, la distribution stationnaire du nombre de tâches présentes aux instants d'arrivée $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_{c+r})$, où $\pi_i = \lim_{n\to\infty} P(X_n = i)$, $0 \leq i \leq k$ existe et unique. Dû à la propriété PASTA, la distribution du nombre de tâches présentes dans le système à l'instants d'arrivée est identique à la distribution du nombre de tâches présentes dans le système à un instant arbitraire. $\boldsymbol{\pi}$ peut être obtenue en résolvant le système d'équations linéaires 3.13. Ce système ne peut pas être résolu sous forme fermée, nous devons donc recourir à une solution numérique.

3.5.2 Résultats numériques

Nous effectuons nos résultats numériques sous les mêmes hypothèses utilisées dans [41], c'est-à-dire nous considérons que les temps de service suivent une distribution Gamma dont le coefficient de variation de service prend deux valeurs $cv = 0,5$ et $1,4$ sous une itensité du traffic très elevée $\rho = 0,85$.

Dans toutes les figures les résultats analytiques et les résultats de simulation sont étiquetés avec Ana et Sim respectivement.

- Nous présentons d'abord la probabilité de blocage que nous illustrons dans la Figure 3.2.

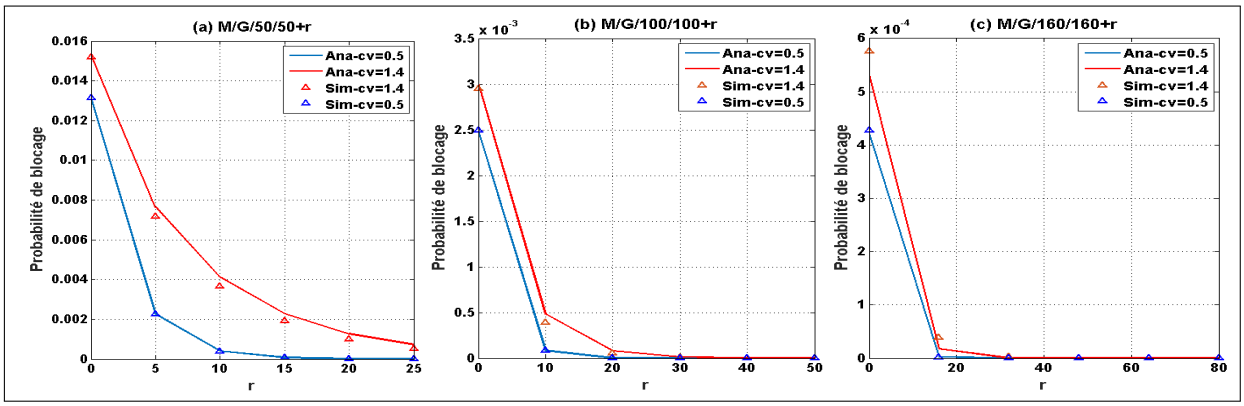

FIGURE 3.2 – Probabilité de blocage vs. capacité du buffer r

Les résultats présentés dans cette Figure confirment que si la capacité du buffer augmente

linéairement, la probabilité de blocage diminuerait de manière exponentielle. Dans le système avec 50 serveurs, la probabilité de blocage varie de $0,002$ à 4×10^{-4} lorsque la capacité du buffer varie de 5 à 10 respectivement. Elle est égal à 0.8×10^{-4} dans le système avec 100 serveurs lorsque la capacité du buffer est égale à 10. Alors que pour le système de 160 serveurs, la probabilité de blocage est beaucoup plus faible.

A partir de ces résultats, nous pouvons estimer la plus petite capacité du buffer, de sorte que la probabilité de blocage reste inférieure à une valeur prédéfinie ϵ. Pour $\epsilon = 0.8 \times 10^{-4}$, la capacité du buffer doit être d'au moins 10.

- La probabilité de service immédiat est illustrée dans la Figure 3.3.

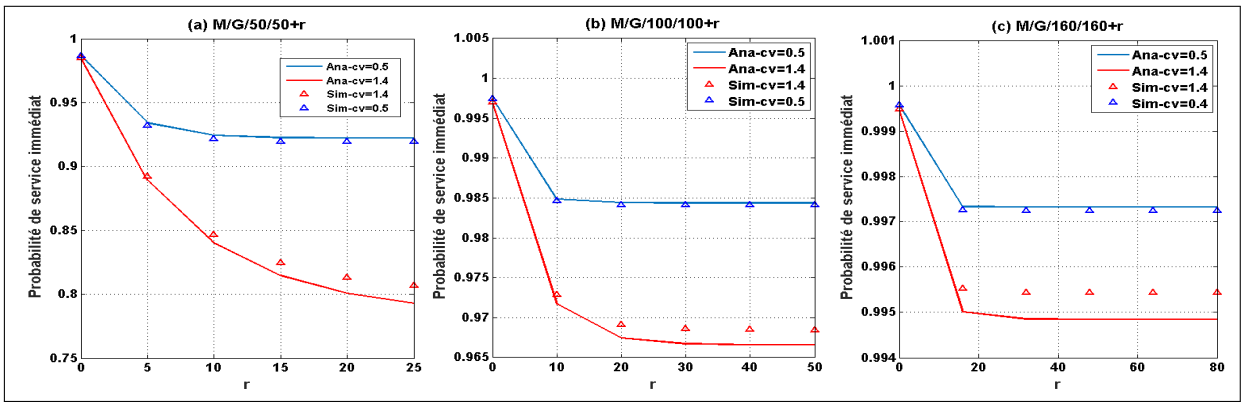

FIGURE 3.3 – Probabilité de service immédiat vs. capacité du buffer r.

Comme on peut le voir, la probabilité de service immédiat diminue avec l'augmentation de la capacité du buffer. On remarque également dans la Figure 3.3 que cette probabilité est proche de la valeur 1 quand la valeur de la capacité du buffer est faible, ce qui explique qu'une nouvelle arrivée puisse être directement servie sans rejoindre le buffer.

- Il est également important de connaître la probabilité d'attente par le fournisseur de service Cloud et ses clients, car il peut arriver qu'un client arrivée quitte le système sans obtenir de service en raison de la longueur de la file d'attente. Nous calculons donc cette probabilité et les résultats sont présentés dans la Figure 3.4 :

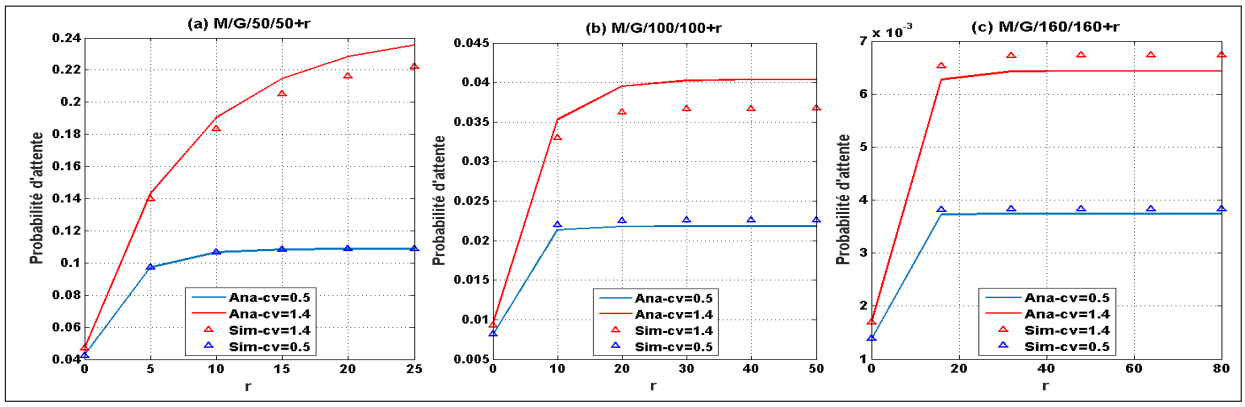

FIGURE 3.4 – Probabilité d'attente vs. capacité du buffer r.

Comme on peut le voir, la probabilité d'attente augmente rapidement lorsque la capacité du buffer augmente. Cette probabilité est faible lorsque la capacité du buffer est très faible, ce qui explique qu'une nouvelle arrivée ne doive attendre que la durée de service des tâches en service. Nous remarquons également dans cette Figure que la probabilité d'attente dans le système avec un grand nombre de serveurs est différente de celle dans le système avec un petit nombre de serveurs.

- Enfin, la Figure 3.5 montre le temps moyen de réponse :

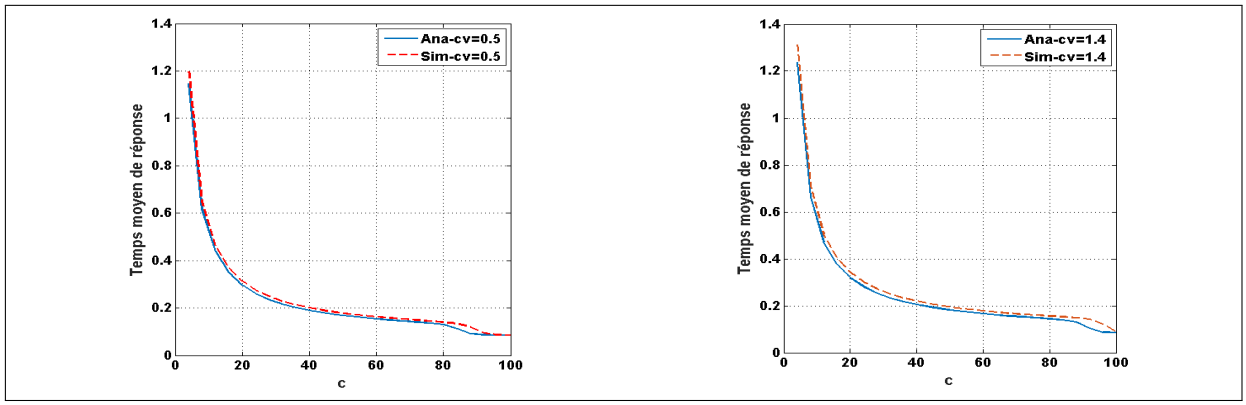

FIGURE 3.5 – Temps moyen de réponse vs. nombre de serveurs c.

Comme on peut le voir sur cette Figure, le temps moyen de réponse diminue lorsque le nombre de serveurs augmente.

Enfin, nous notons que les résultats analytiques obtenus sont proches de ceux obtenus par simulation, ce qui confirme la validité de notre modèle analytique.

Conclusion

Dans ce chapitre, nous avons effectué une analyse mathématique du modèle de files d'attente $M/G/c/k$ en introduisant le processus stochastique qui convient mieux pour le nombre de demandes de tâches présentes dans le système aux instants d'arrivées. Ensuite, nous avons proposé de nouvelles formules approximatives pour calculer la matrice des probabilités de transition de ce modèle. Afin d'examiner la précision de nos formules, nous les avons testé numériquement. À partir de cette matrice des probabilités de transition, nous avons calculé la distribution stationnaire et quelques indicateurs de performance tels que la probabilité de blocage, le temps moyen de réponse, la probabilité de service immédiat et la probabilité d'attente. Nous avons aussi estimé la plus petite capacité du buffer de telle sorte que la probabilité de blocage des demandes des utilisateurs en nuage reste inférieure à une valeur prédéfinie ϵ. Enfin, nous avons validé nos résultats par simulation.

Dans le chapitre suivant, nous étendons ce modèle en tenant compte de l'effet du comportement des clients impatients sur le profit total des fournisseurs de service Cloud.

CHAPITRE 4

PROBLÈME LA CONFIGURATION OPTIMALE POUR MAXIMISER LE PROFIT DES FOURNISSEURS DE SERVICE CLOUD

Introduction

Le problème de maximisation de profit des fournisseurs de service Cloud est de plus en plus un problème crucial. Pour ces fournisseurs, le choix de la configuration optimale des systèmes multi-serveur pour maximiser le profit est très important.

L'utilisation des modèles de files d'attente permet aux fournisseurs de service de prendre la bonne décision en terme de nombre de serveurs et de leur vitesse d'exécution afin de maximiser le profit tout en respectant le contrat SLA. Dans ce chapitre, le problème de la configuration optimale des systèmes multi-serveur pour maximiser les profits dans un environnement du Cloud Computing est étudié.

4.1 Synthèse bibliographique

Le problème de la configuration optimale des systèmes multi-serveur dans les environnements du Cloud Computing a été analysé dans quelques travaux de recherche (voir par exemple [61–64]). Dans [61], un système multi-serveur est traité comme un modèle de files d'attente $M/M/c$, de sorte que le problème d'optimisation peut être formulé et résolu analytiquement. Dans [62], le Cloud Data Center a été considéré comme un modèle de files d'attente avec une capacité finie, les temps d'inter-arrivée et de service sont distribués selon une loi exponentielle ; l'objectif principal des auteurs était de maximiser le profit des fournisseurs de service sous une probabilité

de perte prédéfinie. Dans [63], un système multi-serveur est considéré comme un modèle de files d'attente $M/M/c + D$ où les temps d'inter-arrivée et de service sont supposées exponentiellement distribuées et le système a une capacité variable.

Généralement, les systèmes multi-serveur sont considérés comme des modèles de files d'attente markoviens, car ces derniers ont des formules analytiques explicites de la densité de probabilité de temps d'attente [61]. Cependant, l'hypothèse de la distribution exponentielle de temps de service est inappropriée pour justifier le temps réel de service des demandes des utilisateurs du Cloud. Par conséquent, nous supposons un temps de service général pour les demandes de service et nous traitons le problème de la configuration optimale comme un modèle de files d'attente non-markovien $M/G/c/k$.

Pour les files d'attente $M/G/c$, il est bien connu que la probabilité d'attente dans la file $M/M/c$, c'est-à-dire la formule d'attente d'Erlang, est généralement une bonne approximation pour les autres distributions de temps de service. Dans [36], l'auteur a donné une excellente approximation pour le temps moyen d'attente afin de fournir une approximation plus précise de la probabilité d'attente dans la file $M/G/c$ pour un petit nombre de serveurs. Mais la prolifération du Cloud Computing a abouti à la création des systèmes avec un grand nombre de serveurs, ce qui rend cette approximation inappropriée pour évaluer la probabilité d'attente dans les systèmes du Cloud Computing.

Dans ce chapitre, nous fournissons une nouvelle formule analytique pour calculer la probabilité d'attente dans la file $M/G/c/k$ pour n'importe quel nombre de serveurs considérés. Lorsque la capacité de la file $M/G/c/k$ tend vers l'infini, la probabilité d'attente dans la file $M/G/c$ peut être calculée. Aussi, en utilisant cette formule, nous calculons le temps moyen d'attente de chaque nouvelle arrivée au système. En outre, nous nous intéressons au calcul de cette mesure de performance car le temps d'attente est la source de la satisfaction/insatisfaction du client ; le principal facteur influant sur les décisions du client, en effet, il peut arriver qu'un client quitte le système sans obtenir de service en raison de la longueur de la file d'attente, ce qui entraînera directement des pertes de revenus pour un fournisseur de Cloud ; aussi c'est l'un des facteurs qui peut déterminer les frais d'un service Cloud.

4.2 Modèle multi-serveur

4.2.1 Description du modèle

Nous modélisons un système multi-serveur (Cloud plate-forme) par un modèle de files d'attente $M/G/c/k$ où le comportement des clients impatients a été pris en considération. Dans ce modèle, nous considérons que tous les c serveurs fonctionnent avec une même vitesse s_p (mesurée par le nombre de demandes de service qui peuvent être exécutées en une unité de temps). Les demandes de service des clients arrivent selon un processus de Poisson, ce qui signifie que le temps d'inter-arrivée A est exponentiellement distribué avec un taux λ (> 0). Sa fonction de distribution cumulative (CDF) est $A(x) = P[X \leq x]$, sa fonction de densité de probabilité est $a(x) = \lambda e^{-\lambda x}$ et sa Transformée de Laplace Stieltjes (LST) est donnée par :

$$A^*(s) = \int_0^\infty e^{-sx} a(x)dx = \frac{\lambda}{\lambda + s}.$$

La discipline de service est FIFO. Lorsqu'une nouvelle arrivée de demande de service trouve le système complet, elle est perdue. Le nombre de demandes à exécuter sont des variables aléatoires indépendantes et identiquement distribuées selon une distribution générale R d'une moyenne $E(R)$. Les temps de service des demandes sont également des variables aléatoires indépendantes et identiquement distribuées selon une loin générale $H(y) = P[Y \leq y]$ avec $y = \frac{r}{s_p}$ et de moyenne $E(H) = \frac{E(R)}{s_p}$. La LST de temps de service est :

$$H^*(s) = \int_0^\infty e^{-sy} h(y)dy.$$

Soit cv le coefficient de variation de la distribution de service H et $\rho = \frac{\lambda E(H)}{c}$ l'intensité du trafic.

4.2.2 Probabilité d'abandon

Les clients décident d'abandonner le service avec une probabilité P_a ou de l'accepter avec une probabilité $1 - P_a$. Ainsi, le taux d'abandon λ_a peut être obtenu comme suit :

$$\lambda_a = \lambda' \times P_a,$$

où λ' est le taux effectif d'arrivée, P_a est égal au produit de temps d'attente moyen, $E(W)$, et de l'indice potentiel d'abandon d, avec $d \in [0, 1]$.

Le taux final d'arrivée, λ_f, peut être défini comme suit :

$$\lambda_f = \lambda' - \lambda_a.$$

Nombre de demandes de service présentes dans le système	Probabilité d'abandon P_a
21	0.004
25	0.10
32	0.33
35	0.44
44	0.74
49	0.91
52	1.00

TABLE 4.1 – La probabilité d'abandon de client en fonction du nombre de demandes de service présentes dans le système.

Exemple 4.1. *$M/G/20/52$ queue :*

Dans cet exemple, nous calculons la probabilité d'abandon de client en fonction du nombre de demandes de service présentes dans le système.

Comme on le voit dans le Tableau 4.1, la probabilité d'abandon augmente avec l'augmentation du nombre de clients dans le système. En d'autres termes, lorsque le nombre de clients augmente dans le système, le temps d'attente augmente également ; Par conséquent, certains clients voient que le temps d'attente est trop long et ils abandonnent le système sans obtenir de service.

4.3 Temps moyen d'attente

Dans cette section, nous proposons une formule analytique pour calculer le temps moyen d'attente d'une nouvelle demande de service arrivée au système. Pour ce faire, nous supposons que tous les c serveurs sont occupés lorsque la n-ième demande arrive au système. Soit L_n le nombre de demandes de service trouvées dans le système à l'instant d'arrivée de la n-ième demande de service. Pour calculer le temps moyen d'attente de cette nouvelle arrivée, nous considérons deux variables aléatoires W et V qui représentent les temps d'attente actual et virtuel respectivement.

Etant donné que $W > 0$, le temps d'attente virtuel V peut être décomposé en [36] :

$$V = V_R + V_q, \tag{4.1}$$

où :

▷ V_R est le plus petit des temps de service résiduels des c demandes en service.

▷ V_q est le temps nécessaire pour que toutes les demandes en attente entrent en service.

Et

$$V_R = V_q = 0, \quad \text{lorsque} \quad W = 0. \tag{4.2}$$

Dû à la propriété PASTA, il est évident que

$$E(V) = E(W) = E(V_R) + E(V_q). \tag{4.3}$$

Nous supposons que le temps nécessaire pour vider un système avec c serveurs est c fois plus petit qu'avec un seul serveur lorsque tous les c serveurs sont occupés. Ainsi, nous pouvons utiliser cette approximation :

$$E(V_q) \simeq \frac{E(H)}{c}(\lambda' E(W)). \tag{4.4}$$

En utilisanr les formules (4.1), (4.3) et (4.4), on obtient

$$E(W) \simeq P(W > 0)E(V_R|W > 0) + \frac{E(H)}{c}(\lambda' E(W)). \tag{4.5}$$

Et par conséquent,

$$E(W) \simeq \frac{P(W > 0)E(V_R|W > 0)}{1 - \frac{E(H)}{c}\lambda'}. \tag{4.6}$$

Pour calculer $E(V_R/W > 0)$, nous donnons quelques propriétés asymptotiques. Cependant, pour calculer $P(W > 0)$, nous fournissons une formule analytique dans la section suivante.

Soit H_e le CDF stationnaire associé au CDF de temps de service H, où

$$H_e(y) = \frac{1}{E(H)} \int_0^y \big(1 - H(u)\big)du, \quad y \geq 0,$$

et soit

$$I_H(c) = \int_0^\infty \big(1 - H_e(y)\big)^c dy, \quad c \geq 1.$$

Nous considérons le lemme suivant :

Lemme 4.1. *[36] :*

Pour $c \geq 1$,

$$\lim_{\rho \to 0} E(V_R|W > 0) \simeq I_H(c). \tag{4.7}$$

Et

$$\lim_{\rho \to 1} E(V_R|W > 0) \simeq \frac{(1 + cv^2)}{2c} E(H). \tag{4.8}$$

La propriété de trafic faible (4.7) et la propriété de trafic élevé (4.8) sont des conséquences directes des théorèmes limites de [65] et de [66], respectivement.

Nous supposons qu'un fournisseur de service Cloud essaie de maintenir l'intensité du trafic le plus possible afin d'assurer une meilleure exploitation des ressources informatiques, ainsi nous considérons que l'intensité de trafic ρ est élevé ($\rho \to 1$). Par conséquent, nous utilisons la propriété (4.8) pour calculer le temps moyen d'attente de la n-ième arrivée au système et nous obtenons la formule suivante :

$$E(W) \simeq \frac{P(W>0)}{2(\frac{c}{E(H)} - \lambda')}(1 + cv^2). \tag{4.9}$$

4.4 Probabilité d'attente

Dans cette section, nous fournissons une nouvelle formule analytique pour calculer la probabilité d'attente afin d'estimer le temps moyen d'attente d'une nouvelle arrivée au système dans la file d'attente $M/G/c/K$.

Considérons la fonction indicatrice, $U(n)$, suivante :

$$U(n) = \begin{cases} 1, & si \;\; L_n \geq c; \\ \\ 0, & si \;\; L_n < c. \end{cases} \tag{4.10}$$

Nous avons :

$$\begin{aligned} E(U(n)) &= 1 \times P(L_n \geq c) + 0 \times P(L_n < c) \\ &= P(L_n \geq c). \end{aligned} \tag{4.11}$$

En appliquant la proporiété de PASTA, on aura :

$$P(W>0) = P(L_n \geq c). \tag{4.12}$$

Aussi, dû à la propriété PASTA la probabilité d'attente, π_W, c'est-à-dire la probabilité qu'une nouvelle demande arrivée au système doive attendre parce que tous les serveurs sont occupés, est égale à

$$\pi_W = P(W>0). \tag{4.13}$$

Donc il suffit de calculer π_W afin de calculer $P(W>0)$. Pour ce faire, nous utilisons d'abord la propriété (4.8) et nous obtenons :

$$\begin{aligned} \frac{(1+cv^2)}{2c}E(H) &= \frac{1}{c} \times \frac{(1+cv^2)}{2}E(H) \\ &= \frac{1}{c} \times E(H_+). \end{aligned} \tag{4.14}$$

Comme nous le savons, lorsque la n-ième demande de service arrive au système trouve $(L_n \geq c)$ demandes de service, elle doit attendre l'achèvement de service du $(L_n - c + 1)$ demandes pour qu'elle entre en service. Comme nous avons supposé que le temps nécessaire pour vider un système avec c serveurs est c fois plus petit qu'avec un seul serveur lorsque tous les c serveurs sont occupés, nous démontrons que le nombre de demandes de service qui doivent être traitées afin que la n-ième demande commence son service est égal à :

$$c \times \lfloor \frac{L_n}{c} \rfloor. \tag{4.15}$$

Par la suite, en utilisant les formules (3.3) , (3.8), (3.9) et (3.10) qu'on a défini dans le chapitre précédent et la formule (4.15), on obtient la formule de la probabilité d'attente π_W comme suit :

$$\pi_W = \sum_{L_n=c}^{k-1} \left[P_x P_{z,2} \dots P_{z,\lfloor \frac{L_n}{c} \rfloor} \right]. \tag{4.16}$$

4.5 Analyse des revenus et des coûts

4.5.1 Frais de service

Dans ce travail, le niveau de service est reflété par le temps d'attente des demandes de service. Par conséquent, nous définissons la fonction des frais de service pour une demande comme suit :

$$C = \begin{cases} aE(R), & si \quad 0 \leq E(W) \leq D; \\ \\ 0, & si \quad E(W) > D. \end{cases} \tag{4.17}$$

- ▷ D : deadline. Si le temps moyen d'attente de la nouvelle demande de service $E(W)$ dépasse D, un événement de violation du contrat SLA se produit.
- ▷ Si $E(W)$ ne dépasse pas D, le fournisseur de service considère que la demande de service est traitée avec un niveau de qualité de service élevé et, facture au client $aE(R)$, où a est les frais de service par unité de quantité de service.
- ▷ Si $E(W)$ dépasse D, le fournisseur de service considère que la demande de service a attendu trop longtemps. Donc il n'y aura pas de frais et le service sera gratuit.

4.5.2 Revenu d'un fournisseur de service Cloud

Le nombre de demandes de service qui peuvent être traitées par unité de temps est égal à λ_f, ainsi le revenu d'un fournisseur de service Cloud par unité de temps sera égal à :

$$R = \lambda_f C. \tag{4.18}$$

4.5.3 Coûts d'un fournissuer de service Cloud

Un fournisseur de service paie les serveurs loués à un fournisseur d'infrastructure. Ce loyer est déterminé par le nombre de serveurs loués et le prix de location par serveur par unité de temps. Si nous supposons que le prix de location d'un serveur par unité de temps est C_{rc} et si nous avons c serveurs loués, alors le coût de location par unité de temps sera calculé comme suit :

$$C_{\text{rental cost}} = c \times C_{\mathrm{rc}}.$$

Le coût de la consommation d'énergie est un autre élément majeur du coût payé par le fournisseur de services. Généralement, la consommation d'énergie dans les circuits CMOS numériques peut être donnée par :

$$\gamma s_p^{\alpha} + P^*,$$

où γs_p^{α} est la consommation d'énergie dynamique et P^* la consommation d'énergie lorsqu'un serveur est inactif. Dans ce travail, la valeur de γs_p^{α} est proche de celle du processeur Intel Pentium M [67] ; nous définissons donc $\gamma = 9.4192$ et $\alpha = 2.0$.

Comme on prend en compte du comportement des clients impatients, nous donnons la quantité moyenne d'énergie consommée par un serveur en une unité de temps par :

$$\lambda_f \frac{E(r)}{c} \gamma s_p^{\alpha-1}.$$

Et la quantité moyenne d'énergie consommée par tous les c serveurs en une unité de temps par :

$$\lambda_f E(r) \gamma s_p^{\alpha-1}.$$

Si nous définissons ε comme le coût de l'énergie par Watt, le coût total de consommation d'énergie de notre système en une unité de temps sous c serveurs est égal à :

$$C_{\text{cost of energy consumption}} = \big(\lambda_f E(r) \gamma s_p^{\alpha-1} + cP^*\big)\varepsilon. \tag{4.19}$$

4.5.4 Fonction de profit

Après avoir défini les revenus et les coûts du fournisseur de service Cloud, nous donnons la fonction de profit dans l'équation suivante :

$$F = R - \big(C_{\text{rental cost}} + C_{\text{cost of energy consumption}}\big).$$

Le résultat présenté dans la Figure 4.1 montre le revenu R et le profit F pour différentes valeurs de λ lorsque le temps d'exécution d'une demande de service Cloud suit la distribution $\Gamma(0.5, 0.5)$, $c = 20$, $k = 40$, $E(R) = 1$ milliard d'instructions, $s_p = 1$ milliards d'instructions par seconde, $a = 10$ unité par milliard d'instructions, $D = 1$ seconde, $d = 0.5$, $\alpha = 2.0$, $\gamma = 9.4192$, $\varepsilon = 0.1$ unité par watt et $C_{\text{rc}} = 1.5$ par seconde.

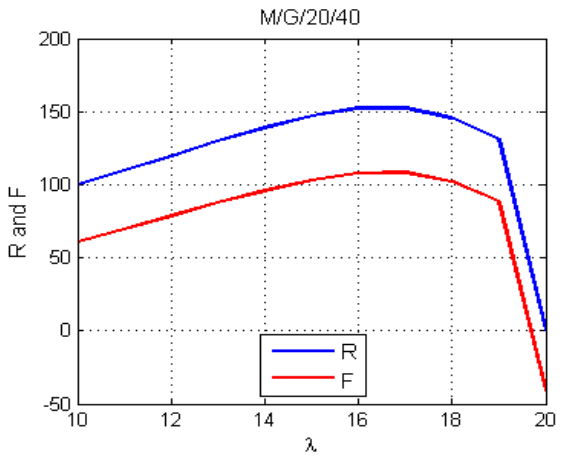

FIGURE 4.1 – Revenue R et Profit F vs. λ.

Sur cette figure, nous observons que R et F augmentent tous les deux avec λ presque linéairement, puis diminuent progressivement et enfin chutent brusquement après un certain point. En d'autres termes, plus de demandes de service génèrent plus de revenus et profit ; à mesure que la longueur de la file d'attente augmente avec l'augmentation du nombre de demandes de service, certains clients quittent le système sans obtenir de service, ce qui entraîne directement des pertes de revenus pour un fournisseur de service Cloud. Une fois que le nombre de demandes de service a atteint un certain point, le temps d'attente moyen dépasse la deadline D et l'événement de violation du contrat SLA s'est produit. Par conséquent, le service sra gratuit, donc il n'y a pas de revenu et le profit sera négatif.

4.6 Maximisation de profit

4.6.1 Nombre de serveurs optimal

Donnons λ, s_p, k, $E(R)$, a, D, d, α, γ, ε et C_{rc}, notre objectif est de trouver le nombre de serveurs c qui peut garantir un maximum de profit .

En utilisant les mêmes paramètres utilisés dans la Figure 4.1, nous montrons sur la Figure 4.2 le profit F en une unité de temps en fonction de c et λ.
Pour $\lambda = 14.9, 13.9, 12.9$, la valeur optimale de c est égale à $20, 19, 17$ respectivement.

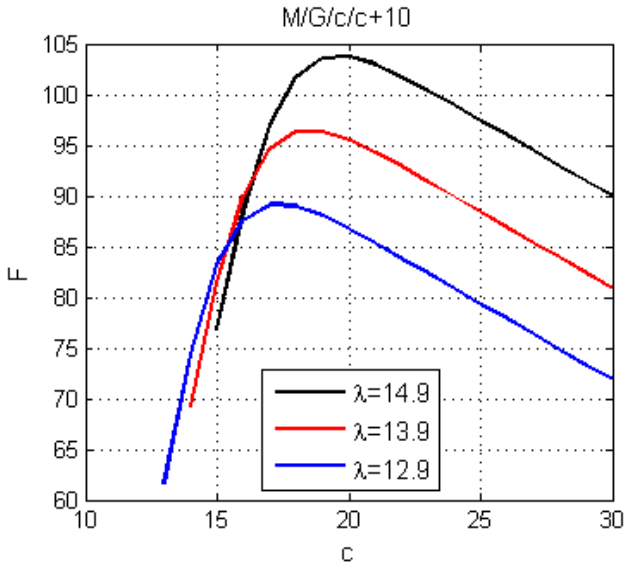

FIGURE 4.2 – Profit F vs. c et λ.

Comme on peut le voir sur la Figure 4.2, lorsque le nombre de serveurs c est petit, le profit F est faible. Ce qui explique que lorsque le nombre de serveurs est petit, la probabilité de blocage et la probabilité d'abandon augmentent, ce qui entraîne des pertes de revenus pour un fournisseur de service Cloud. Par conséquent, le revenu R est faible, ainsi que le profit F. Cependant, lorsque le nombre de serveurs augmente, la probabilité de blocage et la probabilité d'abandon diminuent considérablement, mais le coût du fournisseur de service Cloud (le coût de la location et de la consommation d'énérgie) augmente. Par conséquent, le profit sera réduit. Donc, il existe un choix optimal du nombre de serveurs c qui pourra maximiser le profit.

4.6.2 Vitesse d'exécution optimale

Donnons λ, c, k, $E(R)$, a, D, d, α, γ, ε et C_{rc}, notre objectif est de trouver la vitesse d'exécution d'un serveur s_p qui peut garantir un maximum du profit.

En utilisant les mêmes paramètres utilisés dans la Figure 4.1 et la Figure 4.2, nous montrons sur la Figure **??** le profit F en une unité de temps en fonction de s_p et λ.

Nous remarquons qu'il existe un choix optimal de s_p de telle sorte que F soit maximisé. Pour $\lambda = 14.9, 13.9, 12.9$, la valeur optimale de s_p est respectivement de $1.05, 1.00, 0.95$.

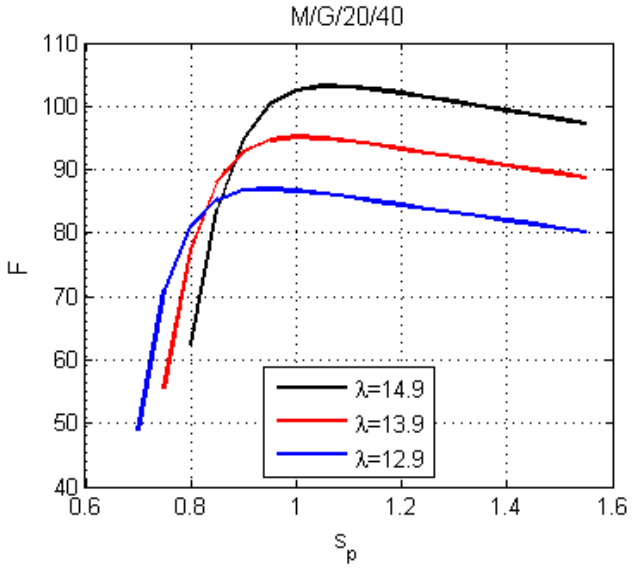

FIGURE 4.3 – Profit F vs. s_p et λ.

Comme le montre la Figure 4.3, lorsque la vitesse d'exécution des serveurs s_p est petite, le profit F est faible. Ce qui explique que lorsque s_p est faible, la probabilité de blocage et la probabilité d'abandon augmentent, ce qui entraîne des pertes de revenus pour un fournisseur de service Cloud. Par conséquent, le revenu R est faible, ainsi que le profit F. Cependant, à mesure que la vitesse d'exécution augmente, le coût de la consommation d'énergie augmente. Par conséquent, l'augmentation des revenus est bien inférieure à l'augmentation des coûts. Ainsi, le profit sera réduit. Donc il existe un choix optimal de s_p de telle sorte que le profit soit maximisé.

4.6.3 Configuration optimale

Donnant λ, k, $E(R)$, a, D, d, α, γ, ε et C_{rc}, notre objectif est de trouver le nombre de serveurs c ainsi que leur vitesse d'exécution s_p qui peuvent grantir un maximum de profit, c'est-à-dire on cherche la meilleure configuration de notre système multi-serveur.

En utilisant les mêmes paramètres utilisée dans les Figures 4.1, 4.2 et **??**, avec $\lambda = 14.9$, nous montrons sur la Figure 5.7 le profit F en une unité de temps en fonction de c et s_p.

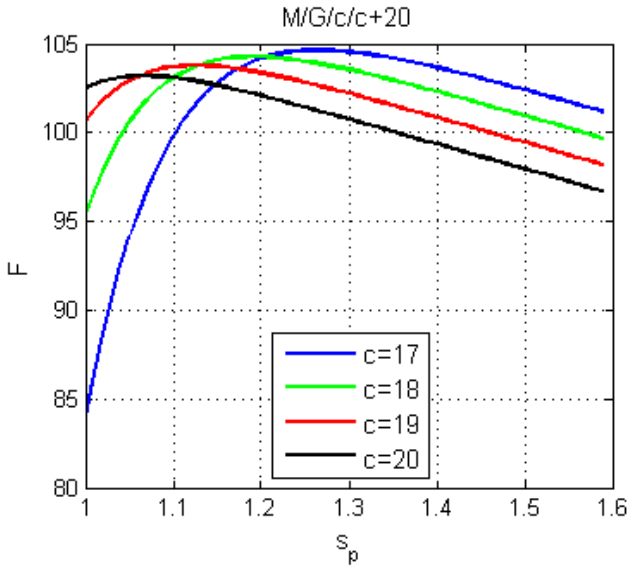

FIGURE 4.4 – Profit F vs. s_p et c.

Comme on peut le voir sur la Figure 4.4, la configration optimale est $(c = 17, s_p = 1,26)$ qui donne un profit maximal de valeur $F = 104.63$.

Conclusion

Dans ce chapitre, nous avons considéré un système multi-serveur comme un modèle de files d'attente $M/G/c/k$ où le comportement des clients impatients a été pris en considération. Afin de formuler et de résoudre le problème de la configuration optimale, nous avons premièrement proposé une formule analytique de temps moyen d'attente de la nouvelle demande de service arrivée au sytème ainsi que sa probabilité d'attente. Deuxièmement, nous avons calculé le revenu et le profit d'un fournisseur de service Cloud par unité de temps.

Dans le chapitre suivant, nous allons étendre ce modèle afin de traiter le problème de la configuration optimale pour maximiser le profit des fournisseurs de service dans les Cloud Data Centers qui fournissent des services hétérogènes.

CHAPITRE 5

MMPP/G/C/K POUR L'ÉVALUATION DES PERFORMANCES DU CLOUD DATA CENTER HÉTÉROGÈNE

Introduction

Le Cloud Computing a été conçu pour prendre en charge un large éventail d'utilisateurs [68]. L'une des caractéristiques de ce nouveau paradigme est l'éclatement (burstiness) exposé par les différents services, c'est-à-dire les augmentations et les diminutions intermittentes de demandes de service des utilisateurs Cloud.

En théorie de probabilité, l'éclatement est due à des changements dans la distribution de probabilité des temps d'inter-arrivées [**?**]. Le processus des arrivées pourrait être mieux décrit en utilisant des distributions à queue lourdes [69] ou des processus plus généraux que les processus de Poisson. Selon [70], il est recommandé d'utiliser les processus doublement stochastiques pour modéliser le processus d'arrivée des événements. Ces processus sont obtenus comme une extension naturelle du processus de Poisson en permettant au taux d'arrivée d'être un processus stochastique.

Dans ce chapitre, nous considérons le processus Markov-modulated Poisson process (MMPP), qui est une sous-classe des processus doublement stochastiques, pour modéliser le processus d'arrivée de demandes de service des utilisateurs Cloud. Par conséquent, nous proposons le modèle de files d'attente $MMPP/G/c/k$ comme modèle adéquat pour la modélisation analytique du Cloud Data Center.

5.1 Processus d'arrivée : processus doublement stochastiques

Dans cette section, nous décrivons brièvement quelques processus à temps continu qui sont couramment utilisés pour la modélisation des processus d'arrivée en cas de présence de phénomènes d'éclatement.

5.1.1 Markovian arrival process

Un MAP (Markovian Arrival Process) est une généralisation du processus de Poisson en permettant aux temps d'inter-arrivée qui ne sont pas exponentiels de maintenir la propriété de Markov [70]. Cette classe contient elle-même la famille des processus de Poisson modulés par une chaîne de Markov (MMPP) et celle des processus de renouvellement de type phase. Une généralisation du MAP avec arrivées par lots est dite BMAP (Batch Markovian Arrival Process).

Description du MAP

Considérons un processus de Poisson $\{N(t)\}$ de taux λ, où $\{N(t)\}$ est le nombre d'arrivées durant l'intervalle $(0, t]$. Soit $\{J(t)\}$ un processus de phase qui prend des valeurs dans $\{1, 2, ..., m\}$ tel que, lorsque le processus est dans l'état $j \in J(t)$, les clients arrivent selon un processus de Poisson de taux λ_j ; et le taux de transition de l'état j à l'état j' est $C_{jj'}$, où j, $j' \in J(t)$. Donc, $\{J(t)\}$ est une chaîne de Markov irréductible à temps continu.
Le processus bi-dimensionnel {N(t), J(t)} est un MAP qui représente un processus de Markov, où $N(t)$ compte le nombre d'arrivées durant $(0, t]$ et $J(t)$ représente la phase du processus d'arrivée.

Le diagramme de transition du MAP à deux états est représenté dans la figure 5.1.

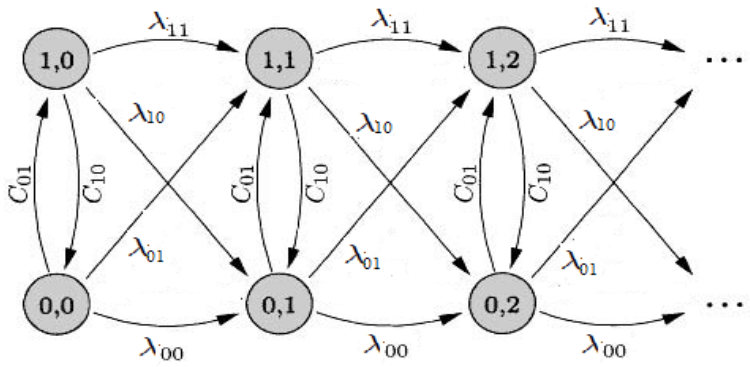

FIGURE 5.1 – Diagramme de transition du MAP(2).

5.1.2 Batch markovian arrival process

Le BMAP a été proposé par [71] comme une extension du MAP offrant une vision beaucoup plus précise du trafic IP, car il capture deux propriétés statistiques importantes du trafic IP, à savoir l'autosimilarité (self-similarty) et l'éclatement. Le diagramme de transition du BMAP à deux états est représenté dans la figure 5.2.

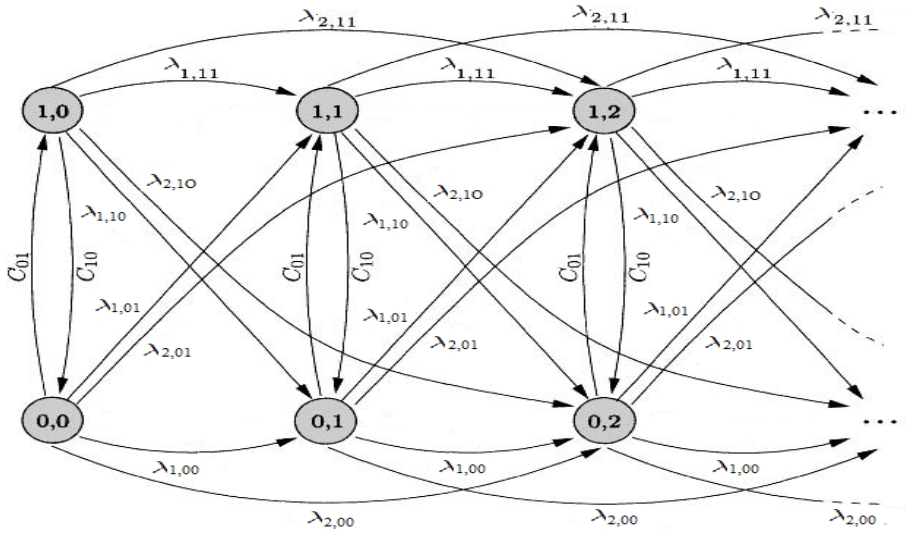

FIGURE 5.2 – Diagramme de transition de BMAP(2).

5.1.3 Markov-modulated Poisson process

Le processus MMPP (Markov-Modulated Poisson Process) est une sous-classe des processus de Poisson doublement stochastiques dont le taux varie en fonction d'un processus de Markov à temps continu. L'utilisation de ces processus permet de modéliser les taux d'arrivée variables dans le temps.

En général, le générateur infinitésimal Q du processus MMPP est donné par :

$$\boldsymbol{Q} = \begin{pmatrix} D_0 & D_1 & 0 & 0 & 0 & \dots \\ 0 & D_0 & D_1 & 0 & 0 & \dots \\ 0 & 0 & D_0 & D_1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix},$$

où

$$\boldsymbol{D_0} = \begin{pmatrix} -(\lambda_1+\beta_1) & \beta_{12} & \beta_{13} & \dots & \beta_{1m} \\ \beta_{21} & -(\lambda_2+\beta_2) & \beta_{23} & \dots & \beta_{2m} \\ \beta_{31} & \beta_{32} & -(\lambda_3+\beta_3) & \dots & \beta_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \beta_{m1} & \beta_{m2} & \beta_{m3} & \dots & -(\lambda_m+\beta_m) \end{pmatrix},$$

$$\boldsymbol{D_1} = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_m \end{pmatrix}.$$

Les cas spéciaux du MMPP sont le processus SPP (Switched Poisson Process), qui est un MMPP à deux états (voir [72, 73]) et le processus IPP (Interrupted Poisson Process) [73], qui est un SPP dont l'un des taux d'arrivée est égal à zéro.

Le diagramme de transition de MMPP à deux états est représenté dans la figure 5.3.

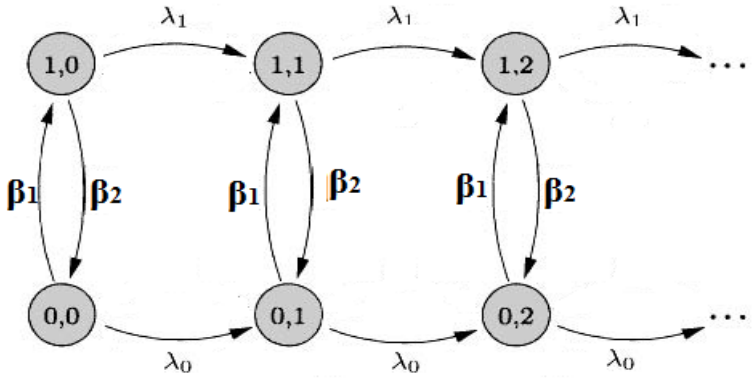

FIGURE 5.3 – Diagramme de transition de MMPP(2).

Dans ce chapitre, nous considérons le processus MMPP à deux états, avec $\lambda_1 > \lambda_2$. Soit p_m la probabilité que le processus soit en phase m, où $m = 1, 2$. Nous avons

$$C_{01}p_1 = C_{10}p_2.$$

$$1 = p_1 + p_2.$$

Et le taux moyen d'arrivée est donné par :

$$\begin{aligned} \lambda &= \lambda_1 p_1 + \lambda_2 p_2 \\ &= \frac{\lambda_1 C_{01}}{C_{01} + C_{10}} + \frac{\lambda_2 C_{10}}{C_{01} + C_{10}}. \end{aligned}$$

5.2 Le modèle $MMPP/G/c/k$

5.2.1 Description du modèle

Pour la modélisation du Cloud Data Center qui fournit des services hétérogènes, nous considérons le modèle de files d'attente $MMPP/G/c/k$ avec c (≥ 1) serveurs qui fonctionnent avec une même vitesse s_p, une capacité limitée du système k ($\geq c$) et FIFO comme discipline de service. Les demandes de service arrivent selon le processus MMPP et elles sont perdues si elles trouvent le système complet. Soit $A(x) = P[X \leq x]$ la fonction de distribution cumulative des temps d'inter-arrivées et $A^*(s) = \int_0^\infty e^{-sx} a(x)dx$ sa transformation de Laplace stieltjes. Le nombre de demandes à exécuter sont des variables aléatoires indépendantes et identiquement distribuées selon une distribution générale R d'une moyenne $E(R)$. Les temps de service des demandes sont également des variables aléatoires indépendantes et identiquement distribuées selon une loin générale $H(y) = P[Y \leq y]$ avec $y = \frac{r}{s_p}$ et de moyenne $E(H) = \frac{E(R)}{s_p}$. La fonction de densité de probabilité est $h(y)$ et la transformation de Laplace stieltjes de temps de service est :

$$H^*(s) = \int_0^\infty e^{-sy} h(y)dy.$$

Soit cv le coefficient de variation de la distribution de service H et $\rho = \frac{\lambda E(H)}{c}$ l'intensité du trafic.

Dans ce modèle, nous supposons que :

- ▷ Chaque serveur ne peut traiter qu'une seule demande de service à la fois.
- ▷ Chaque demande de service ne peut être traitée que par un serveur à la fois.
- ▷ Une fois qu'un serveur a commencé à traiter une demande de service, il continue de s'exécuter sur cette demande jusqu'à ce que la demande soit terminée.

5.2.2 Chaîne de Markov induite

Le modèle de files d'attente $MMPP(2)/G/c/k$ est un modèle non-markovien, il peut être analysé à l'aide de la technique de la chaîne de Markov induite similaire à celle adoptée dans [41].

Cette technique consiste à sélectionner les points de Markov aux instants d'arrivée d'une nouvelle demande de service au système.

Soient $t_n(n \geq 0)$ les instants d'arrivées, $X(t_n)$ (resp. $J(t_n)$) le nombre de demandes trouvées dans le système (resp. la phase du procesus MMPP) aux instants t_n. Le processus $\{(X(t_n), J(t_n)), n \geq 0\}$ forme une chaîne de Markov à temps discret avec un espace d'état fini $\{0, 1, 2, \ldots, k\} \times \{1, 2\}$.

5.3 Quelques indicateurs de performance

Une analyse de performances de la file d'attente $MMPP/G/c/k$ n'existe que dans des cas particuliers, tels qu'un service exponentiel et/ou un serveur unique (voir [74–76]).

Dans cette section, notre objectif est de calculer quelques indicateurs de performance du modèle $MMPP/G/c/k$ qui seront nécessaires dans la suite de notre travail. Nous proposons donc une formule analytique pour calculer la probabilité d'attente et une formule analytique pour calculer le temps moyen d'attente conditionnel en compte tenu du fait que l'attente est positive.

5.3.1 Probabilité d'attente

Pour calculer la probabilité d'attente pour le service, P_{ws}, nous définissons d'abord les probabilités de départ, puis nous calculons le nombre de demandes de service qui doivent être traitées pour que la nouvelle demande arrivée au système commence son service.

Probabilités de départ

▷ Nous définissons la probabilité de terminer le service résiduel d'une demande de service, avec le processus est en phase j, lorsque la n-ème demande arrive au système avec le processus $MMPP$ est en phase j', avec $j, j' = \overline{1,2}$ par :

$$\begin{aligned} A_{jj'} &= P(A > H_+ | J(t_n) = j, J(t_{n+1}) = j') \\ &\quad \times P(J(t_n) = j, J(t_{n+1}) = j') \\ &= H_+^*(\lambda_j) \times p_{j'}, \quad \text{with} \quad j, j' = \overline{1,2}. \end{aligned} \tag{5.1}$$

▷ Lorsqu'il y a un serveur inactif, la n-ième demande qui arrive au système avec $MMPP$ en phase m ($m = 1, 2$) pourra être servie immédiatement. Ainsi, nous définissons la probabilité

qu'elle soit traitée comme suit :

$$\begin{aligned} B_m &= P(A > H | J(t_n) = m) \times P(J(t_n) = m) \\ &= H^*(\lambda_m) \times p_m, \quad \text{where} \quad m = \overline{1,2}. \end{aligned} \tag{5.2}$$

▷ Lorsqu'un serveur termine le service d'une demande, ce serveur devient inactif. Si la file d'attente n'est pas vide, ce serveur achèvera un autre nouveau service, et ainsi de suite. Ainsi, nous définissons la probabilité d'achèvement k demandes de service par ce serveur par :

$$\begin{aligned} P_{k \text{ services}} &= \Big[\sum_{j=1}^{2}\sum_{j'=1}^{2} H_+^*(\lambda_j) \times p_{j'}\Big] \times \\ &\quad \Big[\sum_{m=1}^{2} H^*(\lambda_m) \times p_m\Big]^{k-1}. \end{aligned} \tag{5.3}$$

Nombre de départs

Soit t_n l'instant d'arrivée de la nouvelle demande de service au système et t'_n l'instant juste avant le début du service de cette nouvelle demande. Soit $N(t)$ le nombre de départs dans l'intervalle $t = (t_n, t'_n]$.

En supposant que tous les c serveurs occupés achèvent leurs services en moyenne ensemble, nous donnons aussi dans ce chapitre le nombre de demandes de service qui doivent être traitées pour que la nouvelle demande entre en service par :

$$N(t) = c \times \Big\lfloor \frac{X(t_n)}{c} \Big\rfloor, \tag{5.4}$$

où $\Big\lfloor \frac{X(t_n)}{c} \Big\rfloor$ est la partie entière de $\frac{X(t_n)}{c}$.

En utilisant les formules (5.1), (5.2), (5.3) et (5.4), nous fournissons une nouvelle formule analytique pour calculer la probabilité d'attente pour le service dans la file $MMPP(2)/G/c/k$ comme suit :

$$\begin{aligned} P_{ws} &= \sum_{X(t_n)=c}^{k-1} \Big\{ \Big(\sum_{j=1}^{2}\sum_{j'=1}^{2} H_+^*(\lambda_j) \times p_{j'} \Big)^{\left\lfloor \frac{X(t_n)}{c} \right\rfloor} \times \\ &\quad \prod_{i=0}^{\left\lfloor \frac{X(t_n)}{c} \right\rfloor - 1} \Big(\sum_{m=1}^{2} H^*(\lambda_m) \times p_m \Big)^i \Big\}. \end{aligned} \tag{5.5}$$

La Figure 5.4 représente la probabilité d'attente sous différents taux de service et capacités du système.

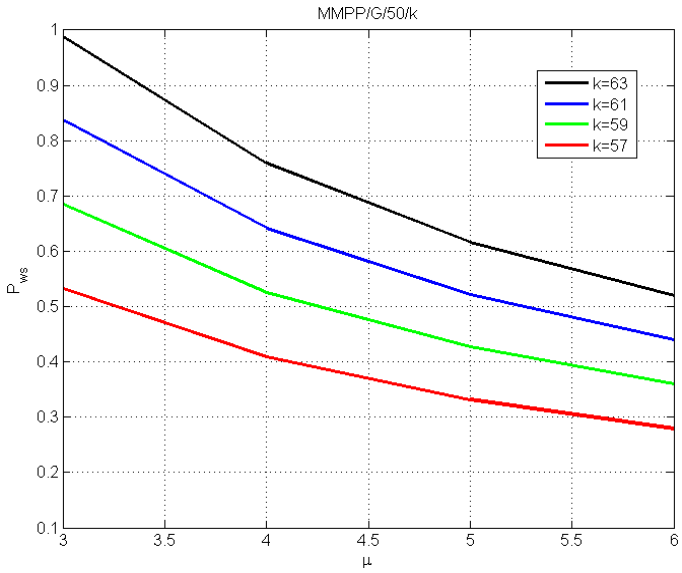

FIGURE 5.4 – La probabilité d'attente pour le service vs. taux de service et capacités du système.

Dans cette figure, on peut voir que la probabilité d'attente est une fonction décroissante de taux de service, c'est-à-dire la probabilité qu'une demande entrante doive attendre car tous les serveurs sont occupés diminue lorsque le taux de service augmente.

5.3.2 Temps moyen d'attente conditionnel

Pour calculer le temps moyen d'attente conditionnel, nous nous appuyons sur la formule de Pollaczek-Khinchin. En utilisant la formule (5.5), nous proposons la formule suivante suit :

$$\overline{w}_c = \frac{P_{ws}}{1-a}\frac{(1+cv^2)E(H)}{2c}, \tag{5.6}$$

où $a = \frac{\lambda(1-P_{loss})E(H)}{c}$ est le taux d'utilisation de chaque serveur, en supposant qu'il soit proche de un. P_{loss} est la probabilité de perte que nous avons calculée en utilisant l'approche proposée dans [4] et les formules (5.1), (5.2) et (5.3).

Le temps moyen d'attente conditionnel sous différentes capacités du système et taux d'arrivée est représenté dans la Figure 5.5 lorsque le temps de service suit la distribution $\Gamma(0.5, 2.45)$.

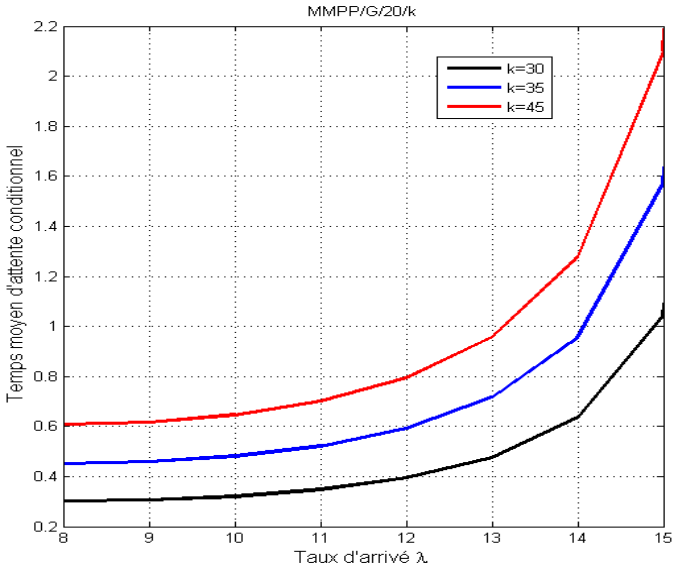

FIGURE 5.5 – Temps moyen d'attente conditionnel vs. taux d'arrivée et capacité du système.

Comment on le voit sur cette figure, le temps d'attente augmente lorsque le taux d'arrivée augmente, ce qui est envident.

Dans la Figure 5.6, nous comparons le temps moyen d'attente conditionnel d'un service hétérogène par rapport au temps moyen d'attente conditionnel d'un service homogène sous divers taux d'arrivée.

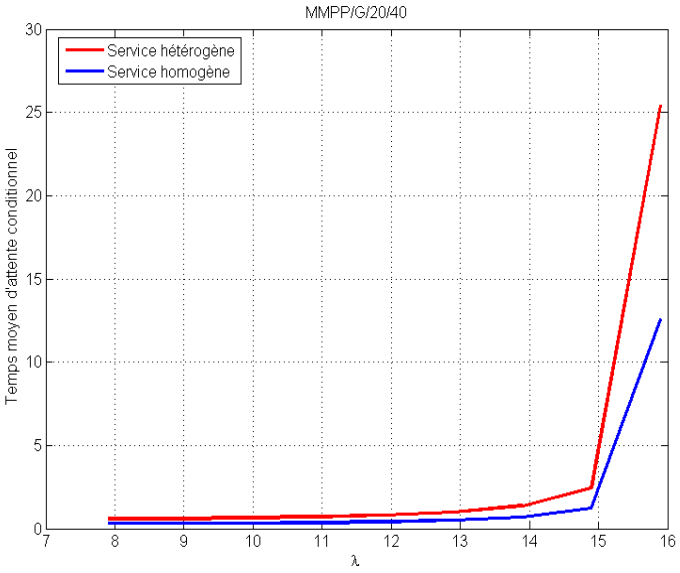

FIGURE 5.6 – Temps moyen d'attente conditionnel vs. λ.

Dans cette Figure, nous pouvons conclure qu'un Cloud Data Center qui fournit des services hétérogènes peut imposer à ses clients un temps d'attente plus long par rapport à un Cloud Data Center qui fournit des services homogènes.

5.4 Fonction de Profit

En utilisant la même fonction de frais de service qu'on a défini dans le chapitre précédent, nous donnons dans l'équation suivante le revenu total de fournisseur de services par unité de temps :

$$R = \lambda(1 - P_{loss}) \times C.$$

où $\lambda(1 - P_{perte})$ est le nombre de demandes de service traitées dans une unité de temps.

De plus, si on prend en compte le comportement des clients impatients dans notre système, le revenu total attendu par unité de temps sera égal à :

$$R = (\lambda(1 - P_{loss}))(1 - d\overline{w}_c) \times C.$$

Les coûts de fourisseur de service qu'on a pris en considération dans notre système sont :

▷ le coût total de consommation d'énergie :

$$C_{\text{cost of energy consumption}} = \Big[(\lambda(1 - P_{loss}))(1 - d\overline{w}_c)E(r)\gamma s_p^{\alpha-1} + cP^*\Big]\varepsilon,$$

où γs_p^{α} est la consommation d'énergie dynamique et P^* la consommation d'énergie lorsqu'un serveur est inactif, avec $\gamma = 9.4192$ et $\alpha = 2.0$.

▷ le coût de la location de serveur :

$$C_{\text{rental cost}} \quad = \quad c \times C_{\text{rc}}.$$

où C_{rc} est le prix de location d'un serveur par unité de temps.

La fonction de profit est :

$$F \ = \ \text{Revnu-Coûts}$$

5.5 Maximisation des profits

Dans cette section, nous présentons le profit attendu par unité de temps pour un fournisseur de service dans un Cloud Data Center qui fournit des services hétérogènes.

Le problème d'optimisation auquel on est abouti consiste à maximiser une fonction de profit à deux variables (nombre de serveurs et vitesse d'exécution d'un serveur). Cependant, le calcul

analytique du maximum de cette fonction, i. e. la configuration optimale, n'est pas possible. Par conséquent, nous optons pour une étude numérique. Cette étude de sensibilité nous permettra d'obtenir une configuration optimale qui maximise le profit. Pour l'obtenir, nous résolvons numériquement notre problème d'optimisation à l'aide de MATLAB.

5.5.1 Nombre de serveurs

Le résultat présenté dans la Figure 5.7 montre le profit F par unité de temps en fonction de c et λ où les temps de service de loi Gamma $\Gamma(0.5, 2.5)$, la capacité du buffer $= 10$, $s_p = 1,245$ milliards d'instructions par seconde, $a = 10$ unité par milliard d'instructions, $D = 1$ seconde, $d = 0.5$, $\alpha = 2.0$, $\gamma = 9.4192$, $\varepsilon = 0.1$ unité par Watt et $C_{\text{rc}} = 1.5$ unité par seconde.
Nous remarquons qu'il existe un choix optimal de c tel que F soit maximisé.

Pour $(\lambda_1, \lambda_2) = (20, 18), (18, 16), (16, 14)$ avec $p_1 = p_2 = 0.5$, la valeur optimale de c est $26, 23, 21$ respectivement.

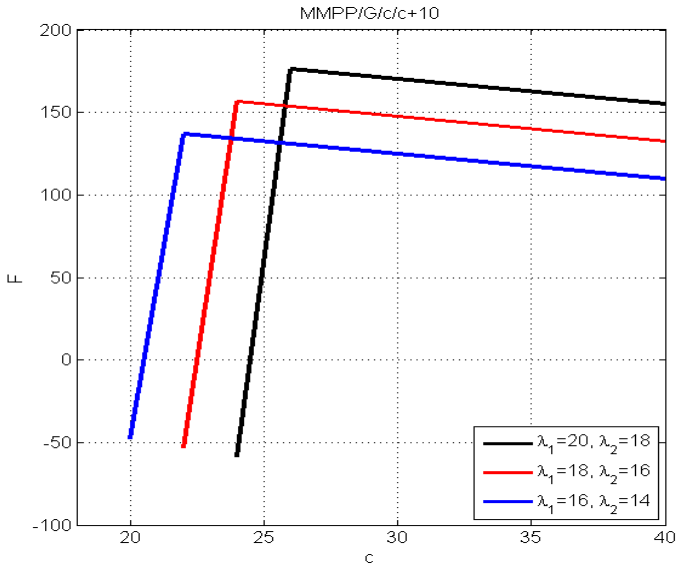

FIGURE 5.7 – Profit F vs. c etλ.

Comme le montre la Figure 5.7, lorsque le nombre de serveurs c est petit, le profit F est faible ; ce qui explique que lorsque c est petit, la probabilité de blocage et la probabilité d'abandon augmentent, ce qui entraîne des pertes de revenus pour un fournisseur de service Cloud. Par conséquent, le revenu R est faible, ainsi que le profit F. En outre, lorsque le nombre de serveurs c est petit, le temps moyen d'attente augmente. De plus, lorsque le temps moyen d'attente dépasse

la date limite D, l'événement de violation du contrat SLA se serait produit. Par conséquent, le service sera gratuit, il n'y a pas de revenu et le profit sera négatif. Cependant, à mesure que c augmente, le temps d'attente moyen, la probabilité de blocage et la probabilité d'abandon diminuent de manière significative, mais les coûts de location et de consommation d'énergie augmentent, de sorte que le profit réduit. Donc, il existe un choix optimal de c qui maximise le profit.

5.5.2 Vitesse d'exécution

Le résultat présenté dans la Figure 5.8 montre le profit F par unité de temps en fonction de s_p et λ où le temps de service de loi $\Gamma(0.5, 2.5)$, capacité du système $k = 30$, $c = 20$, $a = 10$ unité par milliard d'instructions, $D = 1$ seconde, $d = 0,5$, $\alpha = 2.0$, $\gamma = 9.4192$, $\varepsilon = 0.1$ unité par watt et $C_{\text{rc}} = 1.5$ unité par seconde.
Nous remarquons qu'il existe un choix optimal de s_p de sorte que F soit maximisé.

Pour $(\lambda_1, \lambda_2) = (15.9, 13.9), (14.9, 12.9), (13.9, 11.9)$ avec $p_1 = p_2 = 0.5$, la valeur optimale de s_p est $0.63, 0.68, 0.7$, respectivement.

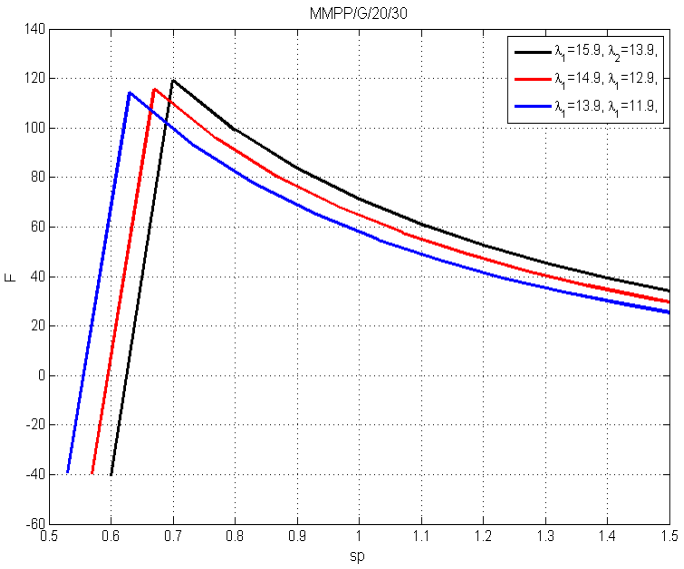

FIGURE 5.8 – Profit F vs. s_p et λ.

Comme le montre la figure 5.8, lorsque la vitesse d'exécution d'un serveur s_p est faible, le profit F est faible ; ce qui explique que lorsque s_p est faible, la probabilité de blocage et la probabilité

d'abandon augmentent, ce qui entraîne des pertes de revenus pour un fournisseur de service. Par conséquent, le revenu R est faible, ainsi que le profit F. De plus, lorsque s_p est faible, le temps d'attente moyen augmente. Donc, si il dépasse la date limite D, l'événement de violation du contrat SLA se serait produit. Par conséquent, il n'y a pas de revenu et le forfit sera négatif. Cependant, à mesure que s_p augmente, le coût de consommation d'énergie augmente. Ainsi, l'augmentation des revenus est bien inférieure à l'augmentation des coûts. Ainsi, le profit sera réduit. Donc, il existe un choix optimal de s_p de sorte que le profit soit maximisé.

5.5.3 Configuration optimale

Dans la Figure 5.9, nous montrons le profit F par unité de temps en fonction de c et s_p lorsque le temps d'exécution d'une demande de service suit la distribution $\Gamma(0.5, 2.5)$, $\lambda_1 = 16$, $\lambda_2 = 14$, $p_1 = p_2 = 0.5$, capacité de la file d'attente $= 15$, $a = 10$ unité par milliard d'instructions, $D = 1$ seconde, $d = 0.5$, $\alpha = 2.0$, $\gamma = 9.4192$, $\varepsilon = 0.1$ unité par Watt et $C_{\text{rc}} = 1,5$ unité par seconde.

La configuration optimale est $(c = 34, s_p = 0.8)$ qui donne un profit maximal de valeur $F = 165.71$.

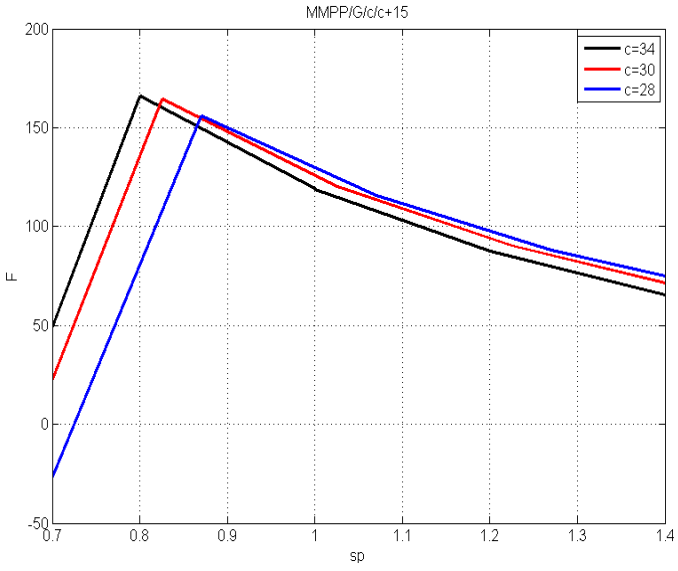

FIGURE 5.9 – Profit F vs. s_p et c.

Conclusion

Dans ce chapitre, nous avons décrit brièvement quelques processus d'arrivée markoviens, tels que markovian arrival process, batch markovian arrival process et Markov-modulated Poisson process. En raison de l'éclatement présentée par les différentes demande de service dans l'environnement du Cloud Computing, nous avons considéré le Markov-modulated Poisson Process (MMPP) pour modéliser le processus des arrivées des demandes de service Cloud. Nous avons modélisé le Cloud Data Center qui fournit des services hétérogènes par un modèle de files d'attente $MMPP/G/c/k$ et nous nous sommes intéressés au problème de la configuration optimale pour maximiser le profit des fournisseur de services. À notre connaissance, il n'existe pas d'étude d'analyse de performance de ce système de files d'attente. Par conséquent, nous avons proposé une formule analytique pour calculer la probabilité d'attente et le temps moyen d'attente conditionnel. Nous avons ensuite calculé le revenu attendu par unité de temps en tenant compte des pertes de revenus résultant de l'abandon de clients en raison d'un très long temps d'attente et des pertes de revenus résultant de la perte des clients due à une capacité finie du système. Nous avons développé une fonction de profit en prenant en compte également le coût de la location d'infrastructure et le coût de la consommation d'énergie. Enfin, nous avons résolu notre problème d'optimisation et nous avons obtenu numériquement la configuration optimale.

Dans ce chapitre et dans les chapitres précédents, nous avons considéré la modélisation analytique et l'évaluation de performanaces des Clouds non-élastiques. Mais l'émergence du Cloud Computing a donné naissance à un nouveau type de systèmes autonomiques dits systèmes Cloud élastiques. Dans le chapitre suivant, nous nous focaliserons sur la modéisation et l'évaluation de performances de ces derniers.

CHAPITRE 6

MODÉLISATION ANALYTIQUE DE L'ÉLASTICITÉ DANS LE CLOUD COMPUTING

Introduction

L'émergence du Cloud Computing a donné naissance à un nouveau type de systèmes autonomiques dits systèmes Cloud élastiques, où l'élasticité est un principe de conception clé.

L'élasticité est l'une des cinq caractéristiques du Cloud Computing [1]. Elle peut être considérée comme un atout majeur du paradigme Cloud qui permet de le distinguer des autres paradigmes [77]. L'origine du concept d'élasticité se retrouve dans le domaine de la physique, où un objet ou un matériau solide est dit élastique, s'il est capable de retrouver sa forme d'origine après avoir été déformé. Dans le Cloud Computing, l'élasticité est une caractéristique principale qui ajuste de manière dynamique la quantité de ressources allouées pour répondre aux changements des exigences de la charge de travail [78]. En effet, on peut comparer l'élasticité dans le Cloud Computing à l'élasticité dans le domaine de la physique : un système Cloud élastique ou une Cloud plate-forme est un objet solide ; l'utilisation des ressource (e.g. les machines virtuelles) et la qualité de service (e.g. le temps moyen de réponse d'une tâche) sont les propriétés de la plate-forme. La charge de travail dynamique (e.g. le nombre de demandes de service) est une force externe. Lorsque la charge de travail augmente (diminue, respectivement), l'utilisation des ressources augmente (diminue, respectivement) et la qualité de service diminue (augmente, respectivement). C'est-à-dire la plate-forme est déformée. Pour revenir à son statut d'origine, la plate-forme doit pouvoir s'ajuster, par exemple en augmentant (en diminuant, respectivement)

le nombre de machines virtuels, de sorte que l'utilisation des ressources et la qualité de service puissent revenir à leur statut d'origine [79].

L'élasticité dans le Cloud Computing joue un rôle important pour garantir le respect des contrats SLA établis entre un fournisseur de services Cloud et ses clients. Du point de vue fournisseur, l'élasticité maximise le profit financier en assurant une meilleure exploitation des ressources informatiques et en permettant à plusieurs clients d'être servis simultanément tout en les gardant satisfaits. Du point de vue client, l'élasticité assure un approvisionnement efficace de ressources qui garantit un maintien de la qualité de service sans dépasser un budget donné [80]. Ainsi, developper des approches systématiques pour modéliser, quantifier et analyser l'élasticité devient un défi principal du Cloud Computing.

L'objectif de ce chapitre est de proposer et d'étudier un modèle de files d'attente pour la modélisation, l'analyse et le calcul de l'élasticité dans le Cloud Computing.

6.1 Elasticité dans le Cloud Computing

6.1.1 Définitions et termes associés

Dans la littérature, il existe plusieurs définitions pour l'élasticité dans le Cloud Computing (voir par exemple [78, 81, 82], [83]). Certains auteurs considérent les concepts de scalabilité et d'élasticité comme identiques (e.g. [84, 85]), d'autres comme étant distincts (e.g. [83, 86]). Selon [83], l'élasticité est définie comme le degré auquel un système est capable de s'adapter aux demandes en approvisionnant et désapprovisionnant des ressources de manière automatique, de telle façon à ce que les ressources fournies soient conformes à la demande du système. Dans [86], l'élasticité est aussi définie comme la capacité d'un système d'ajouter et de supprimer des ressources pour s'adapter à la variation de charge en temps réel. Par contre la scalabilité est définie comme la capacité du système à supporter les charges croissantes en utilisant des ressources supplémentaires [83]. La scalabilité est indépendante du temps et elle est similaire à l'état d'approvisionnement en élasticité, mais le temps n'a aucun effet sur le système [86]. Dans [87], la scalabilité est définie comme la capacité du système à subvenir aux besoins en ressources, sans prendre en compte la rapidité, le temps, la fréquence, ni la granularité de ses actions.

L'équation suivante résume le concept d'élasticité dans le Cloud Computing [86] :

$$\text{Élasticité} = \underbrace{\text{Scalabilité+Automatisation}}_{\text{auto-scaling}} + \text{Optimisation},$$

cela signifie que l'élasticité s'appuie sur la scalabilité et elle peut être considérée comme une automatisation de ce concept, mais elle vise à optimiser au mieux et aussi rapidement que possible les ressources à un moment donné.

Un autre concept lié à l'élasticité est l'efficacité, qui caractérise la manière dont les ressources peuvent être utilisées de manière efficace. Elle s'agit d'une mesure reliant la capacité de la demande aux services consommés au fil du temps [88]. On trouve aussi le concept de résilience qui est la capacité d'un service à continuer de fonctionner malgré la défaillance d'un ou plusieurs éléments d'infrastructure [89].

6.1.2 Types d'élasticité

Il existe deux types d'élasticité : horizontale et verticale. Ces deux types d'élasticité s'appuient sur la technique de virtualisation consistant à reproduire le comportement d'une machine physique dans une machine virtuelle. L'élasticité horizontale offre la possibilité d'ajuster le nombre de machines virtuelles (ajouter/supprimer) en fonction de la demande [77,80]. Par contre, l'élasticité verticale consiste à augmenter/diminuer les ressources d'une machine virtuelle telles que la CPU ou la RAM [90,91].

6.1.3 Nouvelle définition

Du point de vue des auteurs [79] et [92], les définitions présentées dans la littérature sont classées en deux catégories. La première catégorie inclut les définitions qui sont uniquement qualitatives, mais non quantitatives (e.g. [78], [83]) et la deuxième catégorie inclut les définitions quantitatives, mais non analytiquement traitables (e.g. [93]). Par conséquent, les auteurs ont présenté une nouvelle définition quantitative et formelle de l'élasticité dans le Cloud Computing. En d'autre terme, les auteurs ont considéré qu'un système du Cloud Computing est dans (1) un état normal si les ressources informatiques fournies correspondent à la charge de travail actuelle ; (2) un état de sur-approvisionnement si les ressources informatiques fournies dépassent la charge de travail actuelle ; (3) un état de sous-approvisionnement si les ressources informatiques fournies ne peuvent pas gérer la charge de travail actuelle. Ainsi, ils ont défini l'élasticité d'une plateforme du Cloud Computing avec une charge de travail variable de manière dynamique comme le pourcentage de temps (ou de probabilité) pendant lequel le système est à l'état normal et ils ont donné l'équation suivante pour son calcul numérique :

$$\text{Élasticité} \;=\; \frac{T_{\text{normal}}}{T} \;=\; 1 - \frac{T_{\text{over}} + T_{\text{under}}}{T},$$

où $T = T_{\text{normal}} + T_{\text{over}} + T_{\text{under}}$ est une période de temps pendant laquelle le système fonctionne, T_{normal} (T_{over}, T_{under}, respectivement) est la durée totale pendant laquelle le système est à l'état normal (sur-approvisionnement, sous-approvisionnement, respectivement).
Si le système a fonctionné pendant une longue durée et il est dans un état stable, alors :

- ▷ $p_{\text{normal}} = \frac{T_{\text{normal}}}{T}$ est la probabilité que le système soit dans l'état normal.
- ▷ $p_{\text{over}} = \frac{T_{\text{over}}}{T}$ est la probabilité que le système soit dans l'état de sur-approvisionnement.
- ▷ $p_{\text{under}} = \frac{T_{\text{under}}}{T}$ est la probabilité que le système soit dans l'état de sous-approvisionnement.
- ▷ Élasticité $= 1 - (p_{\text{over}} + p_{\text{under}})$.

En se basant sur cette nouvelle définition, nous développons dans la section suivante un modèle analytique pour étudier et évaluer l'élasticité dans le Cloud Computing.

6.2 Modélisation analytique de l'élasticité

Il n'existe pas d'approches analytiques qui peuvent analyser et prévoir l'élasticité dans le Cloud Computing d'une manière précise. Dans [79], l'auteur a considéré le mdèle de files d'attente $M/M/c$ avec une infinité de serveurs qui s'activent/désactivent selon l'état du système (états normal, sur-approvisionnement, sous-approvisionnement). L'inconvénient principal de la chaîne de Markov développée dans ce travail est le manque d'expressions de forme fermée pour les métriques d'élasticité, tels que p_{normal}, p_{over}, p_{under}. Selon Keqin Li, cela rend très difficile l'étude analytique d'une plate-forme de Cloud Computing élastique [79].

6.2.1 Description du modèle

Une plate-forme du Cloud Computing est un système multi-serveur qui contient c $(= s + r)$ serveurs identiques (VMs). Dans ce chapitre, nous considérons un système multi-serveur comme un modèle de files d'attente $M/M/s + r/k$ où les demandes de service arrivent au système suivant un processus de Poisson de taux λ, la durée d'un service est une variable aléatoire distribuée selon une loi exponentielle de paramètre μ. La capacité du système est limitée. Si une demande de service arrive alors que la capacité k du système est déjà atteinte, elle est refoulée et repart immédiatement sans avoir été servie.

Comme dans une plate-forme du Cloud élastique, le nombre de serveurs s'adapte à la charge de travail actuelle, alors nous considérons le modèle d'attente $M/M/s + r/k$ avec un nombre variable de serveurs qui s'activent selon une règle spécifiée et fournissent le service suivant une distribution exponentielle de taux μ_j $(= \mu)$ pour le j-ème serveur, avec $j = 1, 2, ..., s + r$.

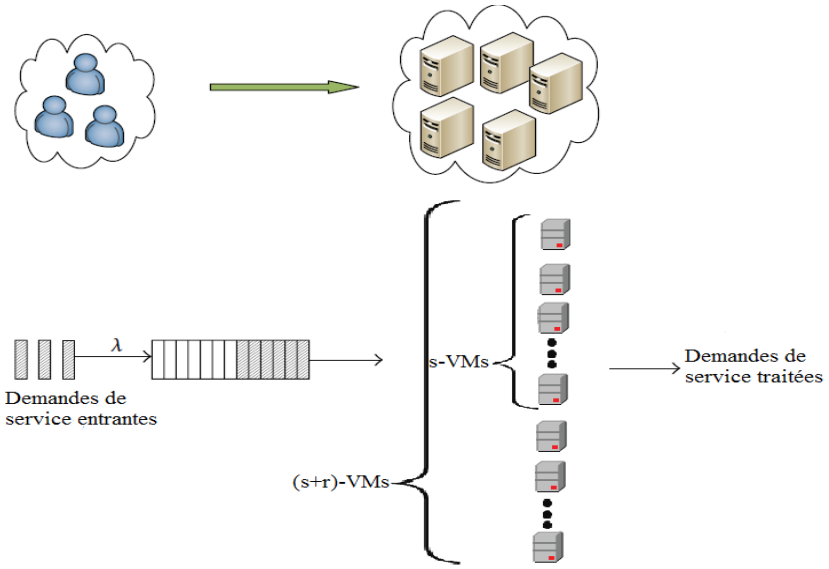

FIGURE 6.1 – Modélisation d'une plate-forme du Cloud élastique sous la forme d'un modèle de files d'attente $M/M/s+r/k$.

6.2.2 Notations et hypothèses

On note (a, b) une paire d'entiers naturels qui serve à déterminer le statut d'un état du système avec $a < b$ et $(i \geq 0)$ est le nombre de demandes de service présentes dans le système à l'instant t :

- ▷ Un état est un état de sur-approvisionnement si : $0 \leq i \leq a$.
- ▷ Un état est un état normal si : $a < i \leq b$.
- ▷ Un état est un état de sous-approvisionnement si : $i > b$.

Le nombre de serveurs actifs (VMs utilisées) dépend du nombre de demandes de service présentes dans le système selon une stratégie de seuil régie par les règles suivantes :

- ▷ Les premiers s serveurs sont en état de marche en permanence avec le système.
- ▷ Le temps pour initialiser un nouveau serveur est une variable aléatoire qui suit une loi exponentielle de moyenne $\frac{1}{\alpha}$ (i.e. le taux de démarrage d'une VM est α).
- ▷ Dès qu'il y a L_1 demandes de service en attente, le $(s+1)$-ème serveur commence à fournir le service, mais il sera supprimé du système si le nombre de demandes de service devient inférieure à L_1.
- ▷ En général, lorsque le nombre de demandes de service en attente atteint un niveau spécifique L_j, le $(s+j)$-ème serveur $(j = 1, 2, 3, ..., r)$ sera disponible pour le service. Dès que le nombre de demandes de service soit inférieur à L_j, le $(s+j)$-ème-ème serveur sera

supprimé du système.

Le graphe représentatif de notre file d'attente $M/M/s+r/k$ est donné en Figure 6.2.

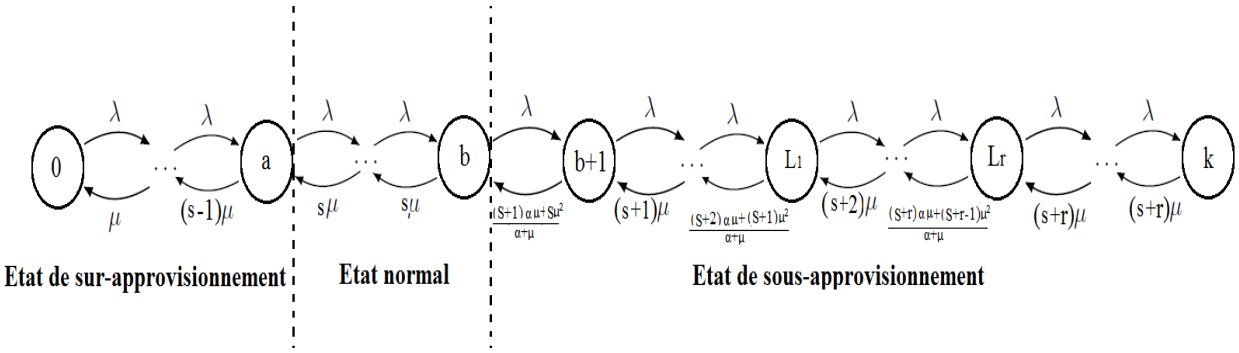

FIGURE 6.2 – Modélisation d'une plate-forme du Cloud élastique sous la forme d'un modèle de files d'attente $M/M/s+r/k$.

6.3 Régime stationnaire

Comme le modèle de files d'attente $M/M/s+r/k$ possède un nombre fini d'états, alors le processus markovien décrivant l'évolution du nombre de demandes de service dans le système est toujours ergodique, donc le système est stable quels que soient les taux d'arrivées λ et de service μ.

6.3.1 Equations de balance

Nous définissons π_i comme la probabilité qu'il y ait i demandes de service dans le système à l'état stationnaire. En utilisant le processus de naissance et de mort, les équations de balance

sont données comme suit :

$$\lambda\pi_0 = \mu\pi_1 \tag{6.1}$$

$$(\lambda + min(s-1,i)\mu)\,\pi_i = \lambda\pi_{i-1} + min(s-1,i+1)\mu\pi_{i+1} \text{ pour } 0 \leq i \leq a \tag{6.2}$$

$$(\lambda + s\mu)\pi_i = \lambda\pi_{i-1} + s\mu\pi_{i+1} \text{ pour } a < i \leq b \tag{6.3}$$

$$(\lambda + (s+j-1)\mu)\pi_{L_j-1} = \lambda\pi_{L_j-2} + \frac{(s+1)\alpha\mu + c(s+j-1)\mu^2}{\alpha+\mu}\pi_{L_j} \text{ pour } j = \overline{1,r-1} \tag{6.4}$$

$$\left(\lambda + \frac{(s+j)\alpha\mu + (s+j-1)\mu^2}{\alpha+\mu}\right)\pi_{L_j} = \lambda\pi_{L_j-1} + (s+j)\mu\pi_{L_j+1} \text{ pour } j = \overline{1,r} \tag{6.5}$$

$$(\lambda + (s+j-1)\mu)\pi_i = \lambda\pi_{i-1} + (s+j-1)\mu\pi_{i+1} \text{ pour } j = \overline{2,r},\ L_{j-1}+1 \leq i \leq L_j - 2 \tag{6.6}$$

$$(\lambda + (s+r)\mu)\pi_i = \lambda\pi_{i-1} + (s+r)\mu\pi_{i+1} \text{ pour } L_r + 1 \leq i < k \tag{6.7}$$

$$(s+r)\mu\pi_k = \lambda\pi_{k-1} \tag{6.8}$$

À partir des équations (6.1) à (6.8), nous pouvons obtenir les expressions du π_i, avec $i = \overline{0,k}$:

$$\pi_i = \begin{cases} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i \pi_0 \quad \text{pour} \ \ 0 \leq i \leq a+1 \leq s; \\ \\ \frac{1}{s^{i-s}s!}\left(\frac{\lambda}{\mu}\right)^i \pi_0 \quad \text{pour} \ \ s+1 \leq i \leq b; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+j)^{i-L_j}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-j}\pi_0 \\ \text{pour } j=1,\ i=L_1, \text{où } L_1 = b+1 \text{ et pour } j=\overline{1,r-1},\ L_j+1 \leq i \leq L_{j+1}-1; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+j-1)^{(i-L_{j-1})-1}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-j}\pi_0 \\ \text{pour } j=\overline{2,r},\ \ i=L_j; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{r}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{r-2}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+r-1)^{(L_r-1)-L_{r-1}}}\right)\left(\frac{1}{(s+r)^{i-L_r}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-r}\pi_0 \\ \text{pour } L_r+1 \leq i \leq k. \end{cases}$$

On note $\rho = \frac{\lambda}{\mu}$, $\varphi = \frac{1}{s^{b-s}s!}$, $\phi = \prod_{i'=1}^{r}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}$, $\sigma = \prod_{\kappa=0}^{r-2}\frac{1}{(s+\kappa)^\kappa}$ et $\varrho = \frac{1}{(s+r-1)^{(L_r-1)-L_{r-1}}}$, π_0 peut être obtenu en utilisant la condition de normalisation :

$$\sum_{i=0}^{s}\pi_i + \sum_{i=s+1}^{b}\pi_i + \pi_{b+1} + \sum_{j=1}^{r-1}\left[\sum_{i=L_j+1}^{L_{j+1}-1}\pi_i\right] + \sum_{j=2}^{r}\pi_{i=L_j} + \sum_{i=L_r+1}^{k}\pi_i = 1 \tag{6.10}$$

$$\sum_{i=0}^{s}\pi_i \;=\; \gamma_1\pi_0,\quad \gamma_1=\begin{cases}\sum\limits_{i=0}^{s}\frac{1}{i!}\rho^i, & si\ \ \rho\neq 1;\\[2ex] \sum\limits_{i=0}^{s}\frac{1}{i!}, & si\ \ \rho=1.\end{cases}$$

$$\sum_{i=s+1}^{b}\pi_i \;=\; \gamma_2\pi_0,\quad \gamma_2=\begin{cases}\sum\limits_{i=s+1}^{b}\frac{1}{s^{i-s}s!}\rho^i, & si\ \ \rho\neq 1;\\[2ex] \sum\limits_{i=s+1}^{b}\frac{1}{s^{i-s}s!}, & si\ \ \rho=1.\end{cases}$$

$$\pi_{b+1} \;=\; \gamma_3\pi_0,\quad \gamma_3=\begin{cases}\varphi\rho^b\frac{\lambda(\alpha+\mu)}{(s+1)\alpha\mu+s\mu^2}, & si\ \ \rho\neq 1;\\[2ex] \varphi\frac{\lambda(\alpha+\mu)}{(s+1)\alpha\mu+s\mu^2}, & si\ \ \rho=1.\end{cases}$$

$$\sum_{j=1}^{r-1}\left[\sum_{i=L_j+1}^{L_{j+1}-1}\pi_i\right] \;=\; \gamma_4\pi_0,\gamma_4=\begin{cases}\varphi\rho^b\sum\limits_{j=1}^{r-1}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\Big)\Big]\times\\ \Big(\frac{1}{s+j}\frac{1-(\frac{1}{s+j})^{L_{j+1}-1-L_j}}{1-\frac{1}{s+j}}\Big)\Big(\rho\frac{1-\rho^{L_{j+1}-1-L_j}}{1-\rho}\Big)\Big\}, si\ \ \rho\neq 1;\\[2ex] \varphi\sum\limits_{j=1}^{r-1}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\Big)\Big]\times\\ \Big(\frac{1}{s+j}\frac{1-(\frac{1}{s+j})^{L_{j+1}-1-L_j}}{1-\frac{1}{s+j}}\Big)\Big(L_{j+1}-1-L_j\Big)\Big\}, si\ \ \rho=1.\end{cases}$$

$$\sum_{j=2}^{r}\pi_{i=L_j} \;=\; \gamma_5\pi_0,\gamma_5=\begin{cases}\varphi\rho^b\sum\limits_{j=2}^{r}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\Big)\Big(\frac{1}{(s+j-1)^{(L_j-L_{j-1})-1}}\Big)\Big]\times\\ \rho^{L_j-b-j}\Big\}, si\ \ \rho\neq 1;\\[2ex] \varphi\sum\limits_{j=2}^{r}\Big\{\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\Big)\Big(\frac{1}{(s+j-1)^{(L_j-L_{j-1})-1}}\Big)\Big\},\\ si\ \ \rho=1.\end{cases}$$

$$\sum_{i=L_r+1}^{k}\pi_i \;=\; \gamma_6\pi_0,\quad \gamma_6=\begin{cases}\varphi\rho^b\phi\sigma\varrho\Big(\frac{1}{s+r}\frac{1-(\frac{1}{s+r})^{K-L_r+1}}{1-\frac{1}{s+r}}\Big)\Big(\rho\frac{1-\rho^{K-L_r}}{1-\rho}\Big), & si\ \ \rho\neq 1;\\[2ex] \varphi\phi\sigma\varrho\Big(\frac{1}{s+r}\frac{1-(\frac{1}{s+r})^{K-L_r+1}}{1-\frac{1}{s+r}}\Big), & si\ \ \rho=1.\end{cases}$$

Ainsi, l'équation (6.10) peut être écrite comme suit :

$$\pi_0=\frac{1}{\gamma_1+\gamma_2+\gamma_3+\gamma_4+\gamma_5+\gamma_6} \tag{6.11}$$

6.3.2 Autres mesures de performance

Nous trouvons maintenant quelques mesures de performance en utilisant la distribution stationnaire.

▷ Probabilité que le (s+j)-ème serveur (j = 1, 3, ..., r-1) fonctionne dans le système :

$$P(s+j) = P(i \geq b+1) = (\gamma_3 + \gamma_4 + \gamma_5 + \gamma_6)\pi_0.$$

▷ Probabilité que tous les serveurs fonctionnent dans le système :

$$P(s+r) = P(L_r + 1 \leq i \leq k) = \gamma_6 \pi_0.$$

▷ Probabilité que le 1er serveur fonctionne dans le système :

$$P(1) = P(i \geq 1) = 1 - \pi_0.$$

6.4 Calcul de l'élasticité dans le Cloud Computing

Á partir aussi de la distribution stationnaire, nous pouvons calculer la valeur de l'élasticité dans le Cloud Computing. La probabilité que le système se trouve dans un état de sur-approvisionnement est :

$$p_{\text{over}} = \sum_{i=0}^{a} \pi_i. \tag{6.12}$$

La probabilité que le système soit à l'état normal est :

$$p_{\text{normal}} = \sum_{i=a+1}^{b} \pi_i. \tag{6.13}$$

La probabilité que le système se trouve dans un état de sous-approvisionnement est :

$$p_{\text{under}} = \sum_{i=b+1}^{k} \pi_i. \tag{6.14}$$

En utilisant les probabilités 6.12, 6.15 et 6.14, la valeur de l'élasticité peut être obtenue :

$$\text{Élasticité} \ = \ \sum_{i=a+1}^{b} \pi_i = 1 - (\sum_{i=0}^{a} \pi_i + \sum_{i=b+1}^{k} \pi_i). \tag{6.15}$$

6.4.1 Illustration graphique

Nous présentons quelques données numériques juste pour montrer l'impact des paramètres du système sur l'élasticité.

Dans la Figure 6.3, nous illustrons p_{over}, p_{normal} et p_{under} en fonction de taux d'arrivée de demandes de service Cloud. On observe que lorsque le taux d'arrivée λ augmente, p_{over}

diminue, c'est-à-dire qu'un grand nombre de demandes de service réduit la probabilité de sur-approvisionnement ; p_{under} augmente, c'est-à-dire que plus de demandes de service accroît la probabilité de sous-approvisionnement et p_{normal} augmente, c.-à-d. que l'élasticité augmente.

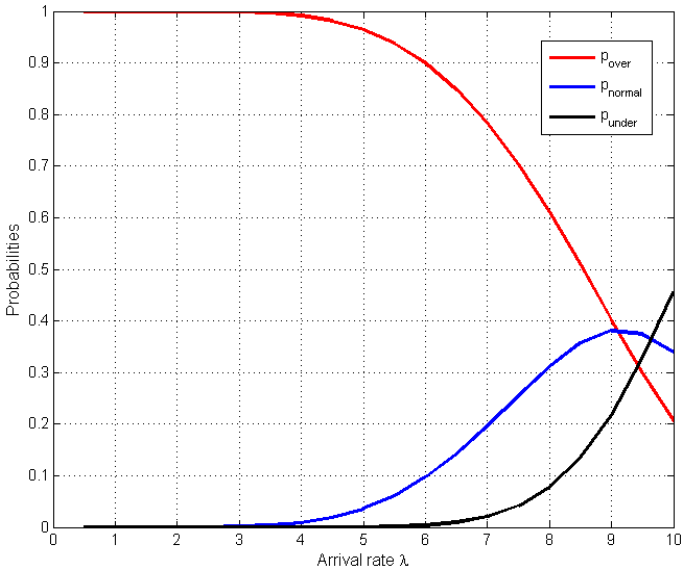

FIGURE 6.3 – p_{over}, p_{normal} et p_{under} vs. λ.

Dans la Figure 6.4, nous illustrons p_{over}, p_{normal} et p_{under} en fonction de taux de service μ. On observe que lorsque μ augmente, p_{over} augmente de manière significative, c'est-à-dire qu'un taux de service plus élvé accroît la probabilité de sur-approvisionnement ; p_{under} diminue, c'est-à-dire qu'un taux de service plus élvé réduit la probabilité de sous-approvisionnement et p_{normal} diminue de manière significative c'est-à-dire que l'élasticité diminue de manière significative.

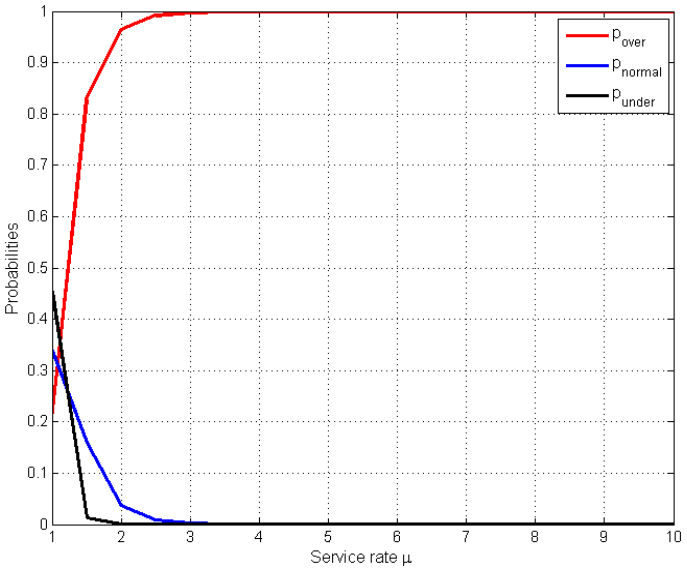

FIGURE 6.4 – p_{over}, p_{normal} et p_{under} vs. μ.

Conclusion

Dans ce chapitre, nous nous sommes intéressés à la modélisation analytique des systèmes Cloud élastiques. En considérant une définition quantitative et formelle de l'élasticité dans le Cloud Computing, nous avons développé un modèle analytique pour étudier l'élasticité en traitant une Cloud plate-forme comme un système de files d'attente $M/M/s + r/k$ avec le nombre de serveurs actifs qui dépend du nombre de demandes de service présentes dans le système. Pour analyser et calculer la valeur de l'élasticité d'une manière précise, nous avons effectué une étude quantitative d'analyse de l'état stationnaire de notre modèle.

Le modèle proposé dans ce chapitre peut permettre aux fournisseurs de service Cloud ainsi que ses utilisateurs d'obtenir une valeur précise de l'élasticité dans le Cloud Computing en utilisant quelques paramètres essentiels d'une Cloud plate-forme, tels que le taux de démarrage des machines virtuelles, le taux d'arrivée de demandes de service Cloud et le taux de service. Il est possible d'aller plus en avant dans l'analyse en prenant en considération le taux d'arrêt des mahines virtuelles.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

La qualité de service a un impact important sur l'adoption plus large du Cloud Computing. En effet, le maintien d'une qualité de service à un certain niveau acceptable pour les utilisateurs du Cloud nécessite une approche précise et bien adaptée d'analyse de performance. La contribution principale de cette thèse est le développement de modèles analytiques pour l'évaluation des performance dans le Cloud Computing en utilisant la théorie des files d'attente. La méthodologie utilisée est basée sur la proposition d'un modèle analytique de base, puis d'élargir progressivement la portée du modèle en rajoutant des paramètres spécifiques pour prendre en compte la nature de l'environnement du Cloud Computing.

Dans la première partie de cette thèse, nous avons présenté les principaux concepts liés au Cloud Computing, ses caractéristiques essentielles, ses services, ses modèles de déploiement, ses acteurs et ses avantages et inconvénients. Nous avons rappelé et présenté les notions et techniques de base sur les systèmes de files d'attente classique et nous avons réalisé une étude bibliographique liée aux modèles non-markoviens à plusieurs serveurs.

Dans la deuxième partie de cette thèse, nous avons présenté les principales contributions réalisées sur l'évaluation de performances des Cloud Data Centers :

- Dans la première contribution, nous avons proposé le modèle de files d'attente $M/G/c/k$ pour la modélisation analytique du Cloud Data Center. Nous avons effectué une analyse mathématique de ce modèle en introduisant le processus stochastique qui convient mieux pour le nombre de demandes de tâches présentes dans le système aux instants d'arrivées et nous avons proposé de nouvelles formules approximatives pour calculer la matrice des probabilités de transition associée à sa chaîne de Markov induite. Nous avons aussi estimé

la plus petite capacité du buffer de telle sorte que la probabilité de blocage des demandes des utilisateurs Cloud reste inférieure à une valeur prédéfinie.

- ⊵ La deuxième contribution présentée concerne le problème de maximisation des profits des fournisseurs de service Cloud. En effet, nous avons étendu le modèle $M/G/c/k$ pour tenir compte de l'effet du comportement des clients impatients sur le revenu total de fournisseurs de service Cloud.

- ⊵ Dans la troisième contribution, nous avons adopté le modèle de files d'attente $MMPP/G/c/k$ pour la modélisation analytique et l'évaluation des performances du Cloud Data Center afin de tenir compte de la variation des taux d'arrivées des demandes de service des clients. Ici, nous nous sommes intéressés au problème de la configuration optimale pour maximiser les profits des fournisseurs de service dans les Cloud Data Centers qui fournissent des services hétérogènes.

- ⊵ Enfin, la quatrième contribution concerne la modélisation analytique du Cloud élastique. En effet, nous avons développé un nouveau modèle de files d'attente et nous avons effectué une étude quantitative d'analyse de l'état stationnaire de notre modèle afin d'analyser et de calculer la valeur de l'élasticité dans le Cloud Computing d'une manière précise.

Dans la continuité de nos travaux de thèse de doctorat, nous envisageons les perspectives suivantes :

- ⊵ Extension du modèle $MMPP/G/c/k$ en considérant le cas de serveurs variables pour la modélisation analytique de l'élasticité dans le Cloud Computing.

- ⊵ Extention du modèle $MMPP/G/c/k$ en considérant le cas de taux de service variables pour tenir compte des demandes des utilisateurs de service Cloud aux différents nombres de ressources pour différentes durées.

- ⊵ Analyse mathématique du modèle d'attente $MMPP/G/c/k$ avec priorité et dépendance entre les arrivées des demandes de service Cloud.

BIBLIOGRAPHIE

[1] P. Mell and T. Grance *"The NIST Definition of Cloud Computing"*, Special Publication 800-145, 2011. [online] https ://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published.

[2] D. Aïssani and A. Aïssani *"La Théorie des Files d'attente. Fondements Historiques et Applications à l'Evaluation des Performances"*, vol. 24, no. 6, pp. 1087-1096, 2004.

[3] A. Outamazirt, M. Escheikh, D. Aïssani, K. Barkaoui and O. Lekadir *"Stochastic Model for Cloud Data Center with M/G/c/c+r Queue'*, Proceedings of the 10th International Workshop on VECoS'2016, Tunis, Tunisia, vol. 1689, pp. 77-84, 2016.

[4] A. Outamazirt, M. Escheikh, D. Aïssani, K. Barkaoui and O. Lekadir, *"Performance analysis of the M/G/c/c+r queuing system for cloud computing data centres"*, Int. J. Critical Computer-Based Systems, vol. 8, no. 3/4, pp.234–257, 2018.

[5] A. Outamazirt, K. Barkaoui and D. Aïssani, *"Maximizing profit in cloud computing using M/G/c/k queuing model"*, International Symposium on Programming and Systems (ISPS 2018), Algiers, Algeria, pp. 1-6, 2018.

[6] A. Outamazirt, K. Barkaoui and D. Aïssani, *"A novel queuing model for maximizing profit of data centers with heterogeneous services"*, Article accepté pour publication dans IEEE/CAA Journal of Automatica Sinica.

[7] J. Geelan, *"Twenty-one experts define cloud computing"*, Cloud Computing Journal, vol. 2, pp. 1-5, 2009.

[8] R. Buyya, C. S. Yeo and S. Venugopal *"Market-Oriented Cloud Computing : Vision, Hype, and Reality for Delivering IT Services as Computing Utilities"*, In the 10th IEEE Interna-

tional Conference on High Performance Computing and Communications, HPCC'08, pages 5–13, 2008.

[9] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner *"A break in the clouds : Towards a cloud definition"*. ACM SIGCOMM Computer Communication Review, vol. 39, 2008.

[10] L. Wang, G. V. Laszewski, A. Younge, X. He, M. KunzeJie and T. Fu *"Cloud Computing : a Perspective Study"*, New Generation Computing, vol. 28, issue 2, pp 137–146, 2010.

[11] T. Erl, Z. Mahmood and R. Puttini *"Cloud Computing. Concepts, Technology & Architecture"*, chapter Cloud-Enabling Technology, Prentice Hall, pp. 79–116. , 2013.

[12] R. Buyya, C. Vecchiola and S. T. Selvi *"Mastering Cloud Computing. Foundations and Applications Programming"*, chapter Virtualization, Elsevier, pp. 71–109, 2013.

[13] C. Baun, M. Kunze, J. Nimis and S. Tai *"Cloud Computing. Web-basierte dynamische IT-Services"*, Springer, 2010.

[14] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic *"Cloud Computing and Emerging IT Platforms : Vision, Hype, and Reality for Delivering Computing as the 5th Utility"*, Future Generation Computer Systems, vol. 25, issue 6, pp. 599–616, 2009.

[15] Q. Zhang, L. Cheng and R. Boutaba *"Cloud Computing : State-of-the-Art and Research Challenges"*, Internet Services and Applications, vol. 1, issue 1, pp. 7–18, 2010.

[16] https ://www.wisper.io/fr/la-virtualisation/quest-ce-que-la-virtualisation-de-poste-de-travail/

[17] B. S. Thejendra *"Practial IT Service Management : A Concise Guide for Busy Executives"*, chapter Service Level Management, IT Governance Publishing, second edition, pp. 135–148, 2014.

[18] E. Marilly, O. Martinot, S. Betgé-Brezetz and G. Delègue *"Requirements for Service Level Agreement Management"*, IEEE Workshop on IP Operations and Management, Dallas, TX, USA, pp. 57–62, 2002.

[19] E. Brewster, R. Griffiths, A. Lawes and J. Sansbury *"IT Service Management : A Guide for ITILR V3 Foundation Exam Candidates"*, chapter Service Level Management, British Informatics Society Limited, pp. 73–82, 2010.

[20] D. C. Verma *"Service level agreements on IP networks"*, Proceedings of the IEEE, vol. 92, issue 9, pp. 1382-1388, 2004.

[21] J. B. Schmitt *"Heterogeneous Network Quality of Service Systems"*, The Springer International Series in Engineering and Computer Science, pp. 3–14, 2001.

[22] P. Brémaud *"Markov Chains : Gibbs Fields, Monte Carlo Simulation, and Queues"*, Springer texts in Applied Mathematics. Springer, 1998.

[23] D. G. Kendall *"Stochastic inequalities for M/G/1 retrial queues"*, Operations Research Letters, vol. 16, pp. 285–290, 1994.

[24] D. G. Kendall *"Stochastic processes occuring in theory of queues and their analysis by the method of the imbedded markov chain"*. Annals of Mathematical Statistics vol. 24, pp. 338–354, 1953.

[25] A. Ruegg *"Processus stochastiques : avec applications aux phénomènes d'attente et de fiabilité"*, PPUR presses polytechniques. 1989.

[26] D. Gross, J. F. Shortle, J. M. Thompson, C. M. Harris *"Fundamentals of queueing theory"*, 4th Edition, John Wiley & Sons, Inc., Publication, Printed in the United States of America.

[27] L. Kleinrock, *"Queueing Systems"*, vol. 1, Theory. Wiley-Interscience, 1975.

[28] S. A. Nozaki and S. M. Ross *"Approximations in Finite-Capacity Multi-Server Queues with Poisson Arrivals"*, Journal of Applied Probability, vol. 15 no. 4, pp. 826-834, 1978.

[29] P. Hokstad *"Approximations for the M/G/m Queue"* Operations Research, vol. 26, no. 3, pp. 510-523, 1978.

[30] M. Miyazawa *"Approximation of the Queue-Length Distribution of an M/GI/s Queue by the Basic Equations"*, Journal of Applied Probability, vol. 23, no. 2, pp. 443-458, 1986.

[31] M. MORI *"Relations between Queue-Size and Waiting-Time Distributions"*, Journal of Applied Probability, vol. 17, no. 3, pp. 822-830, 1980.

[32] T. Kimura *"Diffusion Approximation for an M/G/m Queue"*, Journal of Operations Research, vol. 31 issue 2, pp. 304-321, 1983.

[33] T. Kimura *"A Transform-Free Approximation for the Finite Capacity M/G/s Queue"*, Journal of Operations Research, vol. 44 no. 6, pp. 984-988, 1996.

[34] T. Kimura *"Optimal Buffer Design of an M/G/s Queue with Finite Capacity"*, Journal of Communications in Statistics Stochastic Models, vol. 12, issue 1, pp. 165-180, 1996.

[35] J. M. Smith *"M/G/c/K Blocking Probability Models and System Performance"*, Journal of Performance Evaluation, vol. 52, pp. 237-267, 2003.

[36] T. Kimura, *"Approximations for the delay drobability in the M/G/s Queue"*, Mathematical and Computer Modelling, vol. 22, no. 10-12, pp. 157-165, 1995.

[37] D. D. Yao *"Refining the diffusion approximation for the M/G/m queue"*, Journal of Operations Research, vol. 33, issue 6, pp. 1266-1277, 1985.

[38] O. J Boxma, J. W. Cohen and N. Huffel *"Approximations of the Mean Waiting Time in an M/G/s Queueing System"*, Journal of Operations Research, vol. 27, issue 6, pp. 1115-1127, 1979.

[39] H. C. Tijms, M. H. V. Hoorn and A. Federgru *"Approximations for the Steady-State Probabilities in the M/G/c Queue"*, Journal of Advances in Applied Probability, vol. 13, no 1, pp. 186-206, 1981.

[40] H. Khazaei, J. Misic, and B. M. Vojislav *"Performance Modeling of Cloud Computing Centers"*, doctoral thesis, University of Manitoba Canada, 2013.

[41] H. Khazaei, J. Misic and B.M. Vojislav *"Performance analysis of cloud computing centers using M/G/m/m+r queueing systems"*, Journal of IEEE Transactions on Parallel and Distributed Systems, vol. 23, issue 5, pp. 936-943, 2012.

[42] X. Chang, B. Wang, J.K. Muppala and J. Liu *"Modeling active virtual machines on IaaS clouds using an M/G/m/m+K queue"*, Journal of IEEE Transactions on Services Computing, vol. 9, issue 3, pp. 408-420, 2016.

[43] K. Xiong and H. Perros *"Service Performance and Analysis in Cloud Computing"*, Proceedings of the 2009 Congress on Services, Los Angeles, California, USA, pp. 693-700, 2009.

[44] B. Yang, F. Tan, Y. Dai, and S. Guo *"Performance Evaluation of Cloud Service Considering Fault Recovery"*, Proceedings of the First International Conference, CloudCom 2009, Beijing, China, pp. 571-576, 2009.

[45] J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella and J. Rius *"A queuing theory model for cloud computing"*, Journal of Supercomputing, vol. 69, issue 1, pp. 492–507, 2014.

[46] L. Guo, T. Yan, S. Zhao and C. Jiang *"Dynamic Performance Optimization for Cloud Computing Using M/M/m Queueing System"*, Journal of Applied Mathematics, vol. 2014, pp. 1-8, 2014.

[47] M. Eisa, E.I. Esedimy and M.Z. Rashad *"Enhancing Cloud Computing Scheduling based on Queuing Models"*, International Journal of Computer Applications, vol.85, no. 2, pp. 17-23, 2014.

[48] N. A. Brown Mary and K. Saravanan *"Performance Factors of Cloud Computing Data Centers Using [(M/G/1) :(∞/GDMODEL)] Queuing Systems"*, International Journal of Grid Computing and Applications, vol.4, no.1, pp. 1-9, 2013.

[49] P. kamble and H. Channe *"Performance analysis of cloud computing centers by Breaking-down response time"*, International Journal of Advanced Computational Engineering and Networking, ISSN (p) : 2320-2106, vol. 1, issue 8, pp. 10-14, 2013.

[50] M. Ben el aattar and A. Haqiq *"Performance Modeling for a Cloud Computing Center Using GE/G/m/k Queuing System"*, International Journal of Science and Research, vol. 3, issue 5, pp. 783-789, 2014.

[51] W. Ellens, M. Zivkovi, J. Akkerboom and R. Litjens *"Performance of Cloud Computing Centers with Multiple Priority Classes"*, 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, pp. 245-252, 2012 ;

[52] A. Anupama and G.S. Keerthi *"Using queuing theory the performance measures of cloud with infinite servers"*, International Journal of Computer Science and Engineering Technology, ISSN : 2229-3345, vol. 5 no. 01, pp. 17-21, 2014.

[53] T. S. Sowjanya, D. Praveen, K. Satish and A. Rahiman *"The queueing theory in cloud computing to reduce the waiting time"*, International Journal of Computer Science and Engineering Technology, vol. 1, issue 3, pp. 110-112, 2011.

[54] G. V. Lakshmi and C.S. Bindhu *"A queuing model to improve quality of service by reducing waiting time in cloud computing"*, International Journal of Soft Computing and Engineering, ISSN : 2231-2307, vol. 4, issue 5, pp. 1-3 , 2014.

[55] M. Firdhous, O. Ghazali and S. Hassan *"Modeling of cloud system using erlang formulas"*, 17th Asia-Pacific Conference on Communications, Sutera Harbour Resort, Kota Kinabalu, Sabah, Malaysia, pp. 411-416, 2011.

[56] X. Nan, Y. He and L. Guan *"Towards optimal resource allocation for differentiated multimedia services in cloud computing environment"*, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Florence, Italy, pp. 684-688, 2014.

[57] R. Murugesan, C. Elango and S.Kannan *"Resource allocation in cloud computing with M/G/s queueing model"*,International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, issue 9, pp. 443-447, 2014.

[58] H. Khazaei, J. Misic and B.M. Vojislav *"Modelling of Cloud Computing Centers Using M/G/m Queues"*, 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, USA, pp. 87, 2011.

[59] H. Takagi *"Queueing analysis : Vacation and Priority Systems"*, , vol. 1, North-Holland, 1991.

[60] D. F. Heyman and M. J. Sobel *"Stochastic Models in Operations Research"*, vol. 1, Dover, 2004.

[61] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, *"Optimal Multiserver Configuration for Profit Maximization in Cloud Computing"*, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1087-1096, 2013.

[62] Y.-J. Chiang and Y. C. Ouyang, *"Profit Optimization in SLA-Aware Cloud Services with a Finite Capacity Queuing Model"*, Mathematical Problems in Engineering, vol. 2014, pp. 01-11, 2014.

[63] J. Mei, K. Li, A. Ouyang, and K. Li, *"A profit maximization scheme with guaranteed quality of service in cloud computing"*, IEEE Transactions on Computers, vol. 64, no. 11, pp. 3064-3078, 2015.

[64] M. Mazzucco, D. Dyachukand and R. Deters *"Maximizing cloud providers revenues via energy aware allocation policies"*, IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, pp. 131–138, 2010.

[65] D. Y. Burman and D. R. Smith *"A light-traffic theorem for multi-server queues"*, Mathematics of Operations Research, vol. 8, no. 1, pp. 15-25, 1983.

[66] J. Köllerström *"Heavy traflic theory for queues with several servers I"*, Journal of Applied Probability, vol. 11, no. 3, pp. 544-552, 1974.

[67] *"Enhanced Intel speedstep technology for the Intel Pentium M processor"*, White Paper, 2004.

[68] F. Oumellal, M. Hanini and A. Haqiq *"MMPP/G/m/m+r Queuing System Model to Analytically Evaluate Cloud Computing Center Performances"*, British Journal of Mathematics & Computer, vol. 4, issue 10, pp. 1301-1317, 2014. K.-I Goh and A.-L. Barabasi *"Burstiness and memory in complex systems"*, Physics Data, 2006.

[69] R. Lambiotte *"Burstiness and Spreading on Temporal Networks"*, University of Namur, 2013.

[70] O. Ibe *"Markov Processes for Stochastic Modeling"*, 2nd Edition, Elsevier, p. 514.

[71] D. M. Lucantoni *"New Results on the Single Server Queue with a Batch Markovian Arrival Process"*, Communications in Statistics. Stochastic Models, vol. 7, issue 1, pp. 1-46, 1991.

[72] M. H. Van Hoorn and L. P. Seelen *"The SPP/G/l queue : Single Server Queue with a Switched Poisson Process as Input Process"*, Operations Research-Spektrum, vol. 5, issue 4, pp. 207–218, 1983.

[73] D. M. Lueantoni, K. S. Meier-Hellstern and M. F. Neuts *"A Single-Server Queue with Server Vacations and a Cla.ss of Non-Renewal Arrival Processes"*, Advances in Applied Probability, vol. 22, no. 3, pp. 676-705, 1990.

[74] D. Il Choi, T.S. Kim, and S. Lee *"Analysis of an MMPP/G/1/K queue with queue length dependent arrival rates, and its application to preventive congestion control in telecommunication networks"*, European Journal of Operational Research, 187, pp. 652–659, 2008.

[75] A. Baiocchi and N. BlCfari-Melazzi *"Steady-State Analysis of the MMPP/G/1/k Queue"*, IEEE Transactions on Communications, vol. 41, no. 4, pp. 531-534, 1993.

[76] V. Ramaswami *"The N/G/l queue and its detailed analysis"*, advances in applied probability, vol. 12, pp. 222-261, 1980.

[77] S. Dustdar, Y. Guo, B. Satzger and H.-L. Truong *"Principles of Elastic Processes"*, IEEE Internet Computing, vol. 15, no. 5, pp. 66-71, 2011.

[78] L. Badger, T. Grance, R. Patt-Corner, and J. Voas *"Draft cloud computing synopsis and recommendations"*, NIST special publication, vol. 800, p. 146, 2011.

[79] K. Li *"Quantitative modeling and analytical calculation of elasticity in cloud computing"*, IEEE Transactions on Cloud Computing, vol. XX, no. YY, pp. 01-14, 2017.

[80] L. Rodero-Merino, L. M. Vaquero and R. Buyya *"Dynamically Scaling Applications in the Cloud"*, ACM SIGCOMM Computer Communication Review, vol. 41, issue 1, pp. 45-52, 2011.

[81] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes and J. N. de Souza *"Elasticity in Cloud Computing : a Survey"*, Annals of Telecommunications, pp. 1–21, 2015.

[82] G. Galante and L. C. E. d. Bona *"A Survey on Cloud Computing Elasticity"*, Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing, UCC '12. Washington, DC, USA : IEEE Computer Society, pp. 263–270, 2012.

[83] N. R. Herbst, S. Kounev, and R. Reussner *"Elasticity in Cloud Computing : What It Is, and What It Is Not"*, Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13), San Jose, CA : USENIX, pp. 23–27, 2013.

[84] D. M. Shawky and A. F. Ali *"Defining a measure of cloud computing elasticity"*, 2012 1st International Conference on Systems and Computer Science (ICSCS), Lille, France, pp. 1-5, 2012.

[85] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zahariacomputing *"A view of cloud"*, Communications of the ACM, vol. 53 issue 4, pp. 50-58, 2010.

[86] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah and Ph. Merle *"Elasticity in Cloud Computing : State of the Art and Research Challenges"*, IEEE Transactions on services computing, vol. 11, issue 2, pp. 430 - 447, 2017.

[87] E. F. Coutinho, D. G. Gomes and J. Neuman de Souza *"An analysis of elasticity in cloud computing environments based on allocation time and resources"*, 2nd IEEE Latin American Conference on Cloud Computing and Communications, Maceio, Brazil, pp. 7-12, 2013.

[88] S. Lehrig, H. Eikerling, and S. Becker *"Scalability, Elasticity, and Efficiency in Cloud Computing : a Systematic Literature Review of Definitions and Metrics"*, Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, QoSA'15. New York, NY, USA : ACM, pp. 83–92, 2015.

[89] R. Ghosh, D. Kim and K. S. Trivedi *"System resiliency quantification using non-state-space and state-space analytic models"* Reliability Engineering and System Safety, vol. 116, pp. 109-125, 2013.

[90] G. Copil, D. Moldovan, H.-L. Truong and S. Dustdar *"Multi-level Elasticity Control of Cloud Services"*, 11th International Conference, ICSOC 2013, Berlin, Germany, pp. 429-436, 2013.

[91] P. Kranas, A. Menychtas, V. Anagnostopoulos and T. Varvarigou *"ElaaS : An innovative Elasticity as a Service framework for dynamic management across the cloud stack layers"*, Proceedings of Sixth International Conference on Complex, Intelligent and Software Intensive Systems, Palermo, Italy, pp. 1042-1049, 2012.

[92] W. Ai, K. Li, S. Lan, F. Zhang, J. Mei, K. Li and R.Buyya *"On Elasticity Measurement in Cloud Computing"*, Hindawi Publishing Corporation, vol. 2016, pp. 01-14, 2016.

[93] S. Islam, K. Lee, A. Fekete and A. Liu *"How a consumer can measure elasticity for cloud platforms"*. Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, Boston, Massachusetts, USA, pp. 85-96 , 2012.

Résumé : Dans cette thèse, nous nous sommes intéressés à l'analyse mathématique des modèles de files d'attente pour l'évaluation des performances dans le Cloud Computing. Dans un premier temps, nous avons exploité le modèle de files d'attente M/G/c/k pour la modélisation analytique stochastique du Cloud Data Center. La résolution analytique de ce modèle reste, à ce jour, un problème ouvert et complexe, car une solution analytique exacte est difficile à obtenir. Ainsi, nous avons fourni de nouvelles formules approximatives pour calculer la matrice des probabilités de transition associée à sa chaîne de Markov induite. Dans un deuxième temps, nous avons étendu le modèle d'attente M/G/c/k en ajoutant le phénomène d'impatience afin de tenir compte de l'effet du comportement des clients impatients sur le revenu total des fournisseurs de service Cloud et nous nous sommes intéressés au problème de la configuration optimale pour maximiser leur profit. Dans un troisième temps, nous avons proposé le modèle d'attente MMPP/G/c/k pour la modélisation analytique stochastique du Cloud Data Center afin de tenir compte de la variation des taux d'arrivées des demandes de service Cloud dans le temps et nous nous sommes intéressés au problème de la configuration optimale pour maximiser le profit dans les Cloud Data Centers hétérogènes. Enfin, dans un quatrième temps, nous avons abordé la modélisation analytique du Cloud élastique.

Mots-clés: Cloud Computing, Modélisation, Modèles de files d'attente, Approximation, Évaluation de performance, Élasticité.

Abstract: In this thesis, we were interested in the mathematical analysis of queuing models for performance evaluation in Cloud Computing. In a first time, we exploited the M/G/c/k queuing model for stochastic analytical modeling of the Cloud Data Center. The analytical resolution of this model remains, to this day, an open and challenging issue because an exact analytical solution is difficult to reach. Thus, we have provided new approximate formulas to compute the transition-probability matrix associated with its embedded Markov chain. In a second time, we extended the M/G/c/k queuing model by adding the impatience phenomenon in order to take into account the effect of impatient customers behavior's on the total revenue of Cloud service providers and we were interested in the problem of optimal configuration to maximize their profit. In a third time, we proposed the MMPP/G/c/k queuing model for the stochastic analytical modeling of the Cloud Data Center in order to take into account the arrival rates fluctuations of customers Cloud service requests in the time and we focused on the problem of optimal configuration to maximize the profit in heterogeneous Cloud Data Centers. Finally, in a fourth time, we approached the analytical modeling of Cloud elastic.

Keywords: Cloud Computing, Modeling, Queuing models, Approximation, Performance evaluation, Elasticity.

الخلاصة: في هذه الرسالة ، كنا مهتمين بالتحليل الرياضي لنماذج الطابور لتقييم الأداء في الحوسبة السحابية. في المرة الأولى ، استغلنا نموذج قائمة الانتظار M /G/c/k من أجل النمذجة التحليلية العشوائية لمركز بيانات السحاب. يظل الحل التحليلي لهذا النموذج ، حتى يومنا هذا ، قضية مفتوحة وصعبة لأنه من الصعب الوصول إلى حل تحليلي دقيق. وبالتالي ، قدمنا صيغًا تقريبية جديدة لحساب مصفوفة احتمال الانتقال المرتبطة بسلسلة ماركوف المدمجة. في المرة الثانية ، قمنا بتوسيع نموذج قائمة انتظار M/G/c/k من خلال إضافة ظاهرة نفاد الصبر من أجل مراعاة تأثير سلوك العملاء الصبور على إجمالي الإيرادات لمقدمي الخدمات السحابية وكنا مهتمين بمشكلة التكوين الأمثل لتحقيق أقصى قدر من الأرباح. في المرة الثالثة ، اقترحنا نموذج قائمة الانتظار MMPP/G/c/k من أجل النمذجة التحليلية العشوائية لمركز البيانات السحابية من أجل مراعاة تقلبات معدلات الوصول لطلبات الخدمة السحابية للعملاء في الوقت المناسب وركزنا على مشكلة التكوين الأمثل لتعظيم الربح في مراكز البيانات السحابية غير المتجانسة. أخيرًا ، في المرة الرابعة ، تناولنا النمذجة التحليلية لمرونة السحابة.

الكلمات المفتاحية: الحوسبة السحابية ، النمذجة ، طوابير الانتظار ، التقريب ، تقييم الأداء ، المرونة.