UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Nouvelles Technologies de l'information et de la communication

Département Informatique

Mémoire Master

Domaine : Informatique

Filière : informatique

Spécialité : Administration et sécurité et Réseau.

Présenté par :

MECHERI RACHID BEN ABDELBASSET MOURD

Encadré par :

Dr KORICHI Wassila

Réplication des données dans le Cloud Computing

Soutenu le : 23/06/2021

devant le jury :

Dr BENMIR Abdelkader	président	UKM Ouargla
Dr MAHDJOUB BACHIR	Examinateur	UKM Ouargla
Dr KORICHI Wassila	Encadreur	UKM Ouargla

Année universitaire 2020/2021

Résumé :

Ces dernières années, les données sont devenues de plus en plus complexes et volumineuses, en raison de l'évolution rapide des technologies et des nouvelles applications à grande échelle. La gestion de ces données représente un défi important. Les fournisseurs de Cloud visent à maximiser leurs profits tout en satisfaisant les exigences des clients, par exemple, les performances.

Dans ce contexte, la réplication de données est une technique bien connue qui vise à augmenter la disponibilité de données, réduire les coûts d'accès aux données ce qui permet d'améliorer les performances et assurer une meilleure tolérance aux pannes.

Dans le Cloud Computing, les stratégies de réplication de données existantes sont orientées vers une plus large sélection de scénarios. De nombreuses stratégies existantes traitent de la disponibilité avec une minorité d'entre elles considérant également la performance. Seule une partie de stratégies de réplication de données proposées pour le Cloud prennent en compte la disponibilité et les performances simultanément.

Mots clés : Cloud Computing, réplication, Stratégie, Disponibilité, Performance.

ملخص

في السنوات الأخيرة، أصبحت البيانات معقدة وكبيرة بشكل متزايد، بسبب التغيرات السريعة في التكنولوجيا والتطبيقات الجديدة واسعة النطاق. تعد إدارة هذه البيانات تحديًا كبيرًا. يهدف مقدمو الخدمات السحابية إلى زيادة أرباحهم مع تلبية متطلبات العملاء، على سبيل المثال، الأداء.

في هذا السياق، يعد تكرار البيانات أسلوبًا معروفًا يهدف إلى زيادة توافر البيانات وتقليل تكاليف الوصول إلى البيانات، وبالتالي تحسين الأداء وضمان تحمل أفضل للأخطاء.

في الحوسبة السحابية، يتم توجيه استراتيجيات تكرار البيانات الحالية نحو مجموعة أوسع من السيناريوهات. تتعامل العديد من الاستراتيجيات الحالية مع التوافر مع أقلية منهم تفكر أيضًا في الأداء. جزء فقط من استراتيجيات نسخ البيانات المقدمة للسحابة يأخذ في الاعتبار التوافر والأداء في وقت واحد.

الكلمات المفتاحية: الحوسبة السحابية، التكرار، الاستراتيجية، التوفر، الأداء.

Abstract:

In recent years, data has become increasingly complex, large, due to rapid changes in technology, and new large-scale applications. Managing this data is a major challenge. Cloud providers aim to maximize their profits while meeting customer requirements, for example, performance.

In this context, data replication is a well-known technique that aims to increase data availability, reduce data access costs, thereby improving performance and ensuring better fault tolerance.

In Cloud Computing, existing data replication strategies are geared towards a wider selection of scenarios. Many existing strategies deal with availability with a minority of them also considering performance. Only a part of the data replication strategies offered for the cloud consider availability and performance simultaneously.

Keywords: Cloud Computing, replication, Strategy, Availability, Performance.

Dédicaces

Je dédie ce modeste travail :

A l'âme de ma mère ;

A mon père ;

A toute ma famille ;

A mes professeurs,

A mon épouse, mes enfants et a tous mes amis ;

Merci.

Rachid

Dédicaces

Je dédie ce modeste travail :

A ma mère ;

A mon père ;

A toute ma famille ;

A mes professeurs,

A mon épouse, mon enfant et a tous mes amis ;

Merci.

Mourad

Remerciements

Nous tenons premièrement à prosterner en remerciant Allah le tout-puissant de nous avoir donné le courage et la patience afin de terminer ce travail. À *l'issue de la rédaction de cette thèse, nous sommes convaincues que la thèse est loin d'être un travail solitaire. En effet, nous n'aurons jamais pu réaliser ce travail doctoral sans le soutien de quelques personnes.*

En premier lieu, nous tenons à remercier notre encadreur de thèse, Madame ***Korichi Wassila*** Enseignante *à l'Université d'Ouargla, pour la confiance qu'il nous a accordée en acceptant d'encadrer ce travail, pour sa grande disponibilité et ses encouragements tout au long de la préparation de cette thèse. On veut, à cet effet, lui adresser tous nous sincères remerciements pour la confiance accordée ainsi que la grande liberté d'idées et de travail qu'elle nous a donné.*

Table des matières

Introduction générale 1
I. Généralités sur le Cloud Computing
I.1 Introduction 4
I.2 L'émergence du Cloud Computing 4
I.3 Définition du Cloud Computing 5
I.4 Modèles de services 6
I.4.1. SaaS (Software as a Service) 7
I.4.2. PaaS (Plateforme as a Service) 8
I.4.3. IaaS (Infrastructure as a Service) 9
I.5 Modèles de déploiement 9
I.5.1 Le Cloud Privée 9
I.5.2 le Cloud public 10
I.5.3 Cloud Communautaire 10
I.5.4 Le Cloud hybride 10
I.6 Les cinq grandes caractéristiques et bénéfices du Cloud 11
I.7 Domaines d'application du Cloud Computing 13
I.8 Challenges 14
I.9 Conclusion 15
II. Gestion des données dans le Cloud Computing
II.1 Introduction 16
II.2 Gestion des données dans le Cloud. 18
II.2.1 Caractéristiques de Cloud 18
II.2.2 Applications de gestion de données dans le Cloud. 20
II.2.2.1 Gestion des données transactionnelles 20
II.2.2.2 Gestion des données analytiques 22
II.3 Disponibilité dans le Cloud Computing 24
II.3.1 Importance de la haute disponibilité dans le Cloud Computing. 24
II.3.2 Les solutions de disponibilité dans le Cloud. 24
II.4 Cohérence des données dans le Cloud. 26
II.4.1 Modèles de cohérence 26
II.4.1.1 Modèles sans synchronisation 26
II.4.1.2 Modèles avec synchronisation 28
II.5 Challenges dans l'environnement de réplication dans le Cloud 28
II.6 Conclusion.. 29
III. La réplication de données dans le Cloud Computing
III.1 Introduction. 30
III.2 Quelque définition 30
III.2.1 Réplication des données 30
III.2.2 Objets, réplique et site 31
III.2.3 Les mises à jour, propagation et exécution. 31
III.3 La stratégie de réplication. 31
III.4 Principe de réplication. 32

III.5 Protocoles de réplication. 33
III.5.1 Protocole de réplication passive 33
III.5.2 Protocole de réplication active. 34
III.5.3 Protocole de réplication semi- active. 35
III.6 Les types de réplication. 36
III.6.1 Réplication de capture instantanée. 36
III.6.2 La réplication transactionnelle 36
III.6.3 La réplication de fusion.. 37
III.7 Réplication basée sur le placement des répliques. 38
III.7.1 Aucune réplication. 38
III.7.2 Réplication totale. 38
III.7.3 Réplication partielle. 38
III.8 Avantages et problèmes de la réplication 39
III.9 Conclusion 40
IV. Etude comparative entre les stratégies de réplication
IV.1 Introduction. 41
IV.2 Travaux liés sur la gestion de répliques dans les Clouds. 41
IV.3 Stratégies de réplication satisfaisant la disponibilité 43
IV.4 Stratégies de réplication satisfaisant le temps de réponse. 47
IV.5 Classifications existantes. 51
IV.5.1 Réplication statique vs. Réplication dynamique. 51
IV.5.2 Réplication centralisée vs. Réplication décentralisée. 51
IV.5.3 Réplication déclenchée par le serveur vs. Par Le client. 52
IV.5.4 Classification basée sur les fonctions objectives. 52
IV.6 Etude comparative. 52
IV.7 Conclusion. 55
CHAPITRE V : Evaluation et expérimentation.
V.1 Introduction……56
V.2. Environnements de développement ……..56
V.3 Paramètres d'évaluation ……..…57
V.3.1 Impact du budget……58
V.3.2 Impact du nombre de centres de données……59
V.3.3 Impact du nombre de requêtes……59
V.4 Conclusion……..59
Conclusion générale……60
Bibliographie……..61

Liste des figures

Figure I.1 Cloud Computing…… ……6

Figure I.2 : les niveaux du Cloud…… ……..7

Figure I.3 : la vision du Cloud Computing…… ..……11

Figure III.1 : Technique de Réplication …… …….33

Figure III.2 : Représente la réplication passive ……34

Figure III.3 : Représente la réplication active……...……....35

Figure III.4 : Représente la réplication semi-active……...……...36

Fig. V.1. Architecture du CloudSim…… ……58

Liste des tableaux

IV.1 Tableau comparatif entre les stratégies de réplication …….53

Introduction Générale :

La révolution technologique qui a fusionné de multiples sources d'information, ainsi que la diffusion des technologies de l'information dans divers secteurs et domaines, a conduit à une explosion du volume de données, qui reflète le changement d'échelle en termes de tailles, de nombres et de types. Ces augmentations massives ont conduit à des changements dans la manière dont les données sont gérées, stockées et accessibles. Les dernières étapes du développement informatique ont produit de nouvelles technologies, notamment :

Cloud Computing, et réplication en particulier dans les systèmes distribués.

L'informatique dans les nuages (Cloud Computing en anglais) s'est imposée ces dernières années comme un paradigme majeur d'utilisation des infrastructures informatiques, Il existe généralement trois types de services dans le Cloud Computing, Infrastructure (IaaS), plateforme (PaaS) et logiciel (SaaS). De plus, les clients Cloud peuvent louer les ressources de calcul et de stockage sur demande et ne payer que pour leurs consommations. Le Cloud Computing offre aux organisations qui gèrent d'immenses volumes de données et qui disposent de ressources internes limitées, la possibilité d'acquérir les ressources selon leurs besoins sur demande, tout en minimisant les coûts. Ils n'auront pas besoin d'investir sur l'installation des nouvelles infrastructures informatiques, ni d'investir sur des ressources humaines pour maintenir et gérer de telles infrastructures.

La réplication de données sur plusieurs nœuds de Cloud est considérée comme une solution efficace pour obtenir de bonnes performances en termes de temps de réponse, d'équilibrage de charge et, surtout, de disponibilité et de fiabilité élevées des données.

Dans contexte, la réplication de données est une technique largement utilisée dans environnements distribués qui stocke des données sur plusieurs sites, elle fournit la disponibilité et la performance. Si un site de stockage tombe en panne, le système peut continuer à fonctionner en utilisant des données répliquées, ce qui augmente la disponibilité et la tolérance aux pannes. En même temps, comme les données sont stockées sur plusieurs sites de stockage. La réplication des données permet de rapprocher les données des utilisateurs pour diminuer la latence, ou de créer de nombreuses copies pour des données extrêmement populaires afin de répartir la charge sur différents serveurs.

Les principaux objectifs de la réplication sont d'améliorer les performances des demandes de données locales et en particulier d'augmenter la disponibilité des données. Les lectures sont

effectuées sur place avec la copie la plus proche du client, ce qui réduit le nombre de transferts de données. En matière de disponibilité des données, la réplication permet de ne pas s'appuyer sur un site central, et d'en utiliser une copie sur un site lorsqu'elle n'est pas disponible sur un autre site.

Bien que très utile, la réplication souffre encore d'un inconvénient majeur dans l'interopérabilité des données (toutes les copies des mêmes données doivent être identiques) Il peut arriver que les copies des données ne soient pas identiques. Par conséquent, des mécanismes doivent être mis en place pour garantir que les copies qui divergent à un moment donné se retrouvent toutes dans le même état. En outre, la propagation des mises à jour d'une copie à toutes les autres versions peut parfois avoir un impact négatif sur les performances du système, ce qui ne devrait pas affecter inutilement le temps de réponse global.

Les stratégies proposées dans les systèmes Cloud doivent alors non seulement répondre aux problématiques classiques citées connus, à savoir : (i) les conditions de la réplication, (ii) le choix des données à répliquer, (iii) le placement des nouvelles répliques et (iv) le nombre de répliques nécessaires à la satisfaction de la Qualité de service, mais elles doivent également prendre en compte les coûts économiques liés à la réplication tels que le coût engendré par la réplication et la rentabilité du fournisseur. En effet, un fournisseur de Cloud a pour but de générer un profit comme toute entreprise économique en plus d'assurer une certaine Qualité de service pour les clients.

Logiquement, les clients s'attendent aux meilleures performances possibles au moyen de faibles coûts. Cependant, cela est parfois difficile à fournir et très coûteux pour le fournisseur, surtout lorsque ces performances sont attendues dans un environnement caractérisé par une charge de travail dynamique. En effet, le faible coût et les bonnes performances sont deux objectifs souvent contradictoires. C'est pour cela que la plupart des Cloud commerciaux n'incluent pas les performances, notamment en termes de temps de réponse, dans le SLA (Service Level Agreement), un contrat de niveau service entre le fournisseur et son client (Wu and Buyya, 2010). De ce fait, les stratégies de réplication de données dans les systèmes Cloud doivent alors répondre aux objectifs suivants :

(i) Fournir une Qualité de service fiable aux clients en satisfaisant le SLA sans sacrifier le profit économique du fournisseur (Buyya et al., 2009),

(ii) Permettre une gestion élastique des ressources en ajustant de manière dynamique l'allocation de ces ressources par le fournisseur à ses clients suivant le modèle de tarification 'pay as you go' (Armbrust et al., 2010), c'est-à-dire que le client ne paye au fournisseur que ce qu'il consomme comme ressources.

Alors que certaines stratégies s'intéressent à la satisfaction des objectifs de performances pour les clients tout en prenant en compte les coûts économiques engendrés par la réplication et le profit économique (monétaire) du fournisseur (Wu et al., 2013; Zeng et al., 2016; Casas et al., 2017; Liu and Shen, 2017; Mansouri and Buyya, 2019). Il convient alors de proposer des stratégies de réplication de données qui répondent au compromis entre les performances pour les clients et le profit économique pour le fournisseur.

Le présent mémoire est structuré autour de quatre principaux chapitres qui se résument comme suit :

Le premier chapitre : Nous présentons un aperçu historique de l'émergence du Cloud Computing, et de ses définitions, y compris l'architecture du Cloud, les fonctionnalités et les caractéristiques, puis la découverte des services de Cloud Computing.

Le deuxième chapitre : nous allons introduire la notion de disponibilité et présenter les concepts de base des données distribués et la cohérence et de réplication qui sont fondamentaux dans la gestion des données dans le Cloud Computing.

Dans le troisième chapitre : nous avons présenté concepts liés à la réplication des données dans le Cloud Computing, les stratégies de réplication, le principe et les types de réplication, De plus, nous avons couvert les concepts de cohérence et de réplication.

Le quatrième chapitre : Nous passons en revue un large état de l'art à travers les classifications existantes des stratégies de réplication proposées dans la littérature. Afin de permettre aux chercheurs d'améliorer les stratégies de réplication à travers les différents critères de classification.

Le Cinquième chapitre : nous décrivons l'environnement de simulation Cloudsim, ainsi les différentes critères d'évaluation des stratégies de réplication dans le Cloud Computing.

1.1 Introduction :

Dans ce chapitre, un aperçu historique de l'émergence du cloud computing et de ses définitions, y compris la structure du cloud et les fonctionnalités et caractéristiques du cloud computing, sont présentés, puis les services de cloud computing sont identifiés, qui sont des logiciels en tant que service SaaS, les plates-formes en tant que service PaaS et l'infrastructure en tant que service IaaS.

Les types standards de nuages sont également couverts, car chacun d'eux a ses propres caractéristiques et caractéristiques.

I.2 L'émergence du Cloud Computing :

La bulle Internet atteint son apogée le 10 mars 2000, puis éclate progressivement au cours des semaines suivantes, avec la vente massive d'actions par des grands noms de la technologie de pointe, tels que Dell et Cisco.

Pour continuer à survivre, les entreprises doivent repenser ou ajuster leur modèle commercial et leurs offres destinés aux clients. Parmi les plus récentes, nombreuses sont celles qui décident de proposer des services basés sur Internet, plutôt que de l'utiliser comme moyen de passer commande ou de communiquer avec les clients.

La fin des années 1990 et le début des années 2000 représentent une période propice pour créer une entreprise en ligne ou investir dans une telle activité. Avec le développement des architectures multi-tenant, l'omniprésence du haut débit et la mise en place de normes d'interopérabilité universelles entre les logiciels, c'est le cadre idéal pour permettre au Cloud Computing de décoller. Salesforce.com est lancé en 1999. Il est le premier site à proposer des applications d'entreprise à partir d'un simple site Web standard, accessible via un navigateur Web : c'est ce qu'on appelle aujourd'hui le Cloud Computing.

Amazon.com lance Amazon Web Services en 2002. Ce nouveau service permet aux utilisateurs de stocker des données et tire profit des compétences d'un très grand nombre de personnes pour de très petites tâches (par exemple, sur Amazon Mechanical Turk). Facebook est fondé en 2004 et révolutionne la façon dont les utilisateurs communiquent et stockent leurs propres données (photos et vidéos), en faisant involontairement du Cloud un service personnel.

En 2006, Amazon développe ses services Cloud. Le premier à voir le jour est Elastic Compute Cloud (EC2), qui permet aux utilisateurs d'accéder à des ordinateurs et d'y exécuter leurs propres applications, le tout sur le Cloud. Le deuxième service lancé est Simple Storage Service (S3). Il permet d'introduire le modèle de paiement à l'utilisation auprès des clients et du secteur en général, modèle qui représente désormais une pratique courante.

Salesforce.com lance ensuite Force.com en 2007. Cette plate-forme en tant que service (PaaS) permet aux développeurs de concevoir, de stocker et d'exécuter toutes les applications et tous les sites Web nécessaires à leurs activités sur le Cloud.

Google Apps arrive en 2009 et permet à ses utilisateurs de créer et de stocker des documents entièrement sur le Cloud.

Plus récemment, les entreprises de Cloud Computing ont cherché à accroître davantage l'intégration de leurs produits. En 2010, salesforce.com lance sa base de données Cloud avec Database.com pour les développeurs, marquant ainsi le développement des services de Cloud Computing utilisables sur n'importe quel terminal, exécutables sur n'importe quelle plate-forme et écrits dans n'importe quel langage de programmation.

I.3. Définition du cloud computing :

Plusieurs approches de définitions ont été proposées par différents auteurs. Mais la définition la plus couramment adoptée est celle du National Institute of Standard and Technology (NIST) formulée en ces termes : “Cloud Computing is a model for enabling ubiquitous, convinient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or provider interaction”. C’est donc un modèle permettant un accès réseau convenable, à la demande et omniprésent à un pool de ressources configurables et partagées pouvant être rapidement approvisionnées et libérées avec un minimum d’effort ou interaction du fournisseur.

En outre, le Cloud Computing peut être défini comme l'une des méthodes de calcul, dans laquelle les ressources informatiques sont fournies en tant que services, et les utilisateurs sont autorisés à y accéder via Internet sans avoir besoin de posséder des connaissances, de l'expérience ou même de contrôler l'infrastructure qui prend en charge ces services. . Le Cloud Computing peut être considéré comme un concept général qui inclut le logiciel en tant que logiciel en tant que service Software As A Service, le Web et d'autres tendances récentes dans le monde de la technologie qui partagent l'idée de s'appuyer sur Internet pour répondre aux besoins informatiques des utilisateurs.

Dans son document intitulé « Livre blanc pour un leadership éclairé » réalisé en 2010 dans le cadre de clarifier le flou entourant le cloud computing, la société de construction de matériels informatiques IBM précise que le cloud computing est à la fois un mode de travail pour les utilisateurs et un modèle de gestion. Par mode de travail, il expose qu'il s'agit d'une informatique d'un nouveau genre où les applications, les données et les ressources sont offertes aux utilisateurs sous la forme de services distribués sur le réseau. Par gestion, il énonce qu'il s'agit d'une façon de gérer un grand nombre de ressources virtualisées qui peuvent être installées sur plusieurs sites de manière à les présenter comme une ressource unique. L'accès à ces ressources se faisant par la suite d'une façon simple avec beaucoup plus de flexibilité à l'aide d'une plateforme Web (Figure I.1).

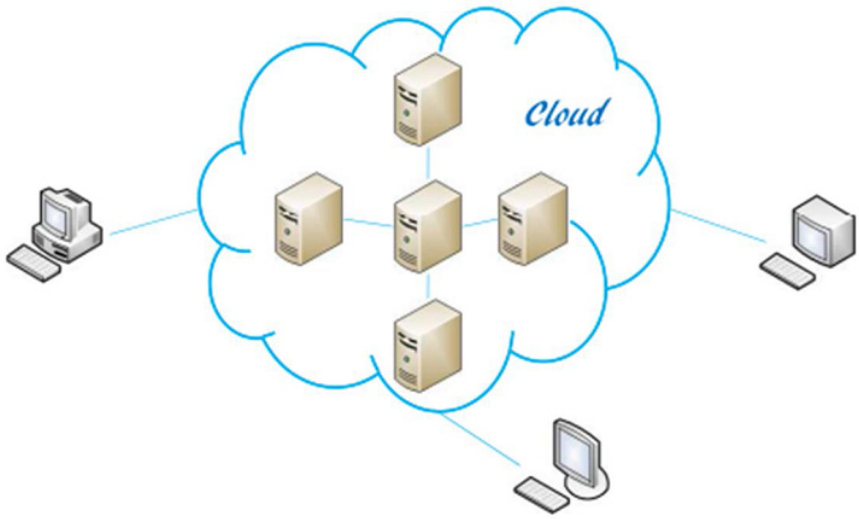

Figure I.1: Cloud Computing.

I.4. Modèles de services :

Trois grands modèles d'usage du Cloud se dégagent actuellement, tous présentent des caractéristiques différentes et n'ont pas le même niveau de maturité :

SaaS (Software as a Service)

PaaS (Plateforme as a Service)

IaaS (Infrastructure as a service)

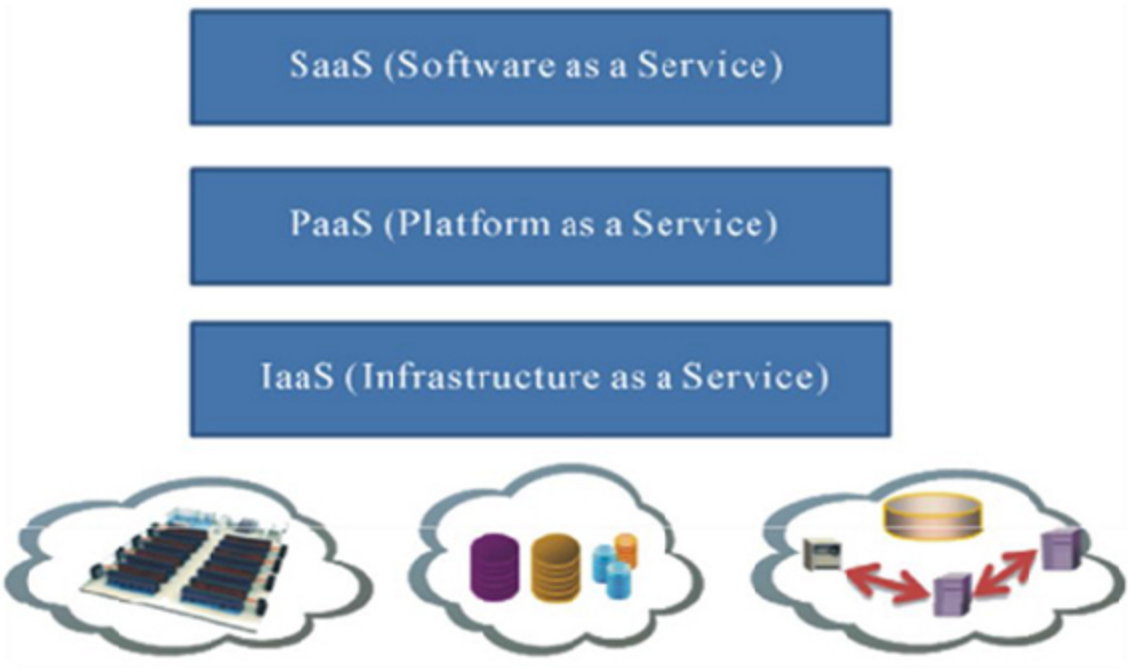

Figure I.2 : les niveaux du Cloud

I.4.1. SaaS (Software as a Service, logiciel en tant que services) :

Le Software as a Service (SaaS), ou Logiciel en tant que Service, est un modèle de distribution de logiciel au sein duquel un fournisseur tiers héberge les applications et les rend disponibles pour ses clients par l'intermédiaire d'internet.

Dans ce type de service, des applications sont mises à la disposition des consommateurs. Les applications peuvent être manipulées à l'aide d'un navigateur web, et le consommateur n'a pas à se soucier d'effectuer des mises à jour, d'ajouter des patches de sécurité et d'assurer la disponibilité du service. Gmail est un exemple de tel service. Il offre aux consommateurs un service de courrier électronique et le consommateur n'a pas à se soucier de la manière dont le service est fourni. D'autres exemples de logiciels mis à disposition en SaaS sont Google Apps, Oracle, Microsoft, Office Web Apps, Adobe Creaitve Cloud ou LotusLive (IBM).

En effet un fournisseur de software as a service peut exploiter des services de type plateforme as a service, qui peut lui-même se servir de infrastructure as a service. D'autres services également disponibles sont :

- Le Data as a Service correspond à la mise à disposition de données délocalisées quelque part sur le réseau. Ces données sont principalement consommées par ce que l'on appelle des mashups.

- BPaaS : il s'agit du concept de Business Process as a service (BPaaS) qui consiste à externaliser une procédure d'entreprise suffisamment industrialisée pour s'adresser directement aux managers d'une organisation, sans nécessiter l'aide de professionnels de l'informatique

- Desktop as a Service : le Desktop as a Service (DaaS ; aussi appelé en français « bureau en tant que service », « bureau virtuel » ou « bureau virtuel hébergé ») est l'externalisation d'une Virtual Desktop Infrastructure auprès d'un fournisseur de services. Généralement, le Desktop as a Service est proposé avec un abonnement payant.

- Network as a Service (NaaS) : le Network as a Service correspond à la fourniture de services réseaux, suivant le concept de Software Defined Networking (SDN).

- STaaS : STorage as a Service correspond au stockage de fichiers chez des prestataires externes, qui les hébergent pour le compte de leurs clients. Des services grand public, tels que SugarSync et Box.net, proposent ce type de stockage, le plus souvent à des fins de sauvegarde ou de partage de fichiers.

Exemples : AmazonS3, Dropbox, GoogleDrive, iCloud, SkyDrive, Ubuntu One, Windows Live Mesh, Wuala

- Workplace as a Service (WaaS)

Le Workplace as service est un environnement de travail virtuelle disponible partout quelle que soit l'appareil utilisé, il permet aux utilisateurs finaux professionnels un accès sécurisé, permanent et en tout lieu à leurs applications et données d'entreprise

Exemples : Econocom workplace as a service, workspace as a service de colt

I.4.2. PaaS (Plateforme as a Service) :

En français plate-forme en tant que service. Dans ce type de service, situé juste au-dessus du précédent, le système d'exploitation et les outils d'infrastructure sont sous la responsabilité du fournisseur. Le consommateur a le contrôle des applications et peut ajouter ses propres

outils. La situation est analogue à celle de l'hébergement web où le consommateur loue l'exploitation de serveurs sur lesquels les outils nécessaires sont préalablement placés et contrôlés par le fournisseur. La différence étant que les systèmes sont mutualisés et offrent une grande élasticité - capacité de s'adapter automatiquement à la demande, alors que dans une offre classique d'hébergement web l'adaptation fait suite à une demande formelle du consommateur

Parmi les principales fonctionnalités proposées par les fournisseurs de PaaS, on dénombre le système d'exploitation, l'environnement de programmation, le système de gestion de base de données, le logiciel de serveur, le support, le stockage, l'accès réseau, les outils de design et de développement, et l'hébergement. Bien entendu, les fournisseurs se distinguent en proposant des fonctionnalités supplémentaires plus spécifiques.

I.4.3. IaaS (Infrastructure as a Service)

L'Infrastructure en tant que Service (IaaS) est une forme de Cloud Computing offrant des ressources informatiques au sein d'un environnement virtualisé (le Cloud) par le biais d'internet ou d'une autre connexion.

L'infrastructure en tant que service : c'est le service de plus bas niveau. Il consiste à offrir un accès à un parc informatique virtualisé. Des machines virtuelles sur lesquelles le consommateur peut installer un système d'exploitation et des applications. Le consommateur est ainsi dispensé de l'achat de matériel informatique. Ce service s'apparente aux services d'hébergement classiques des centres de traitement de données, et la tendance est en faveur de services de plus haut niveau, qui fait d'avantage abstraction de détails techniques

I.5. Modèles de déploiement :

I.5.1. Le Cloud Privée :

Le Cloud Privé est un mode de consommation de l'informatique (IaaS, PaaS, SaaS, …) s'appuyant sur des ressources (serveur, stockage, réseau, licences logicielles…) mises à disposition exclusive d'une entreprise. Les ressources peuvent être géographiquement situées dans le périmètre de l'entreprise (on parlera d'un Cloud privé interne) ou chez un intégrateur/service provider (on parlera d'un Cloud privé managé ou hosté). L'exploitation du Cloud privé peut être réalisée uniquement par les équipes informatiques du client (Cloud

privé interne), ou par un prestataire externe (Cloud privé interne, Cloud privé hosté). Les services disponibles le sont via un catalogue de services exposés dans un portail, leur mise en service est automatisés, et peut faire l'objet d'une facturation liée à la consommation.

I.5.2. le Cloud public :

Le Cloud public est une structure souple et ouverte, géré par un fournisseur tiers. Plusieurs utilisateurs (individuels ou entreprises) peuvent y accéder via Internet. Avec le Cloud public, de multiples entités se partagent les mêmes ressources informatiques (mises à disposition par le fournisseur).

I.5.3. Cloud Communautaire :

Le Cloud Communautaire est une extension du Cloud privé. Dans ce modèle, les ressources, services et la propriété sont partagés à l'échelle d'une communauté (ex. : à l'échelle d'un Etat, d'une ville, d'une académie, etc.). Les services sont accessibles via l'interconnexion de réseaux appartenant aux organisations de cette communauté.

I.5.4. Le Cloud hybride :

Malgré les avantages offerts par le cloud computing, les organisations ne sont généralement pas disposées à partager leurs données sensibles ou risquent de les placer sur des serveurs cloud, et ces organisations n'ont pas la capacité de les contrôler entièrement.

D'où l'importance d'utiliser le modèle hybride du cloud computing, où le modèle hybride est le plus flexible parmi les modèles de cloud computing.

Le Cloud hybride est une structure mixte qui permet de combiner les ressources internes du Cloud privé à celles externes du Cloud public. Une entreprise qui utilise un Cloud hybride peut par exemple avoir recours au Cloud public ponctuellement, lors de pics d'activité et le reste du temps se contenter des ressources à disposition en interne.

I.6. Les cinq grandes caractéristiques et bénéfices du Cloud :

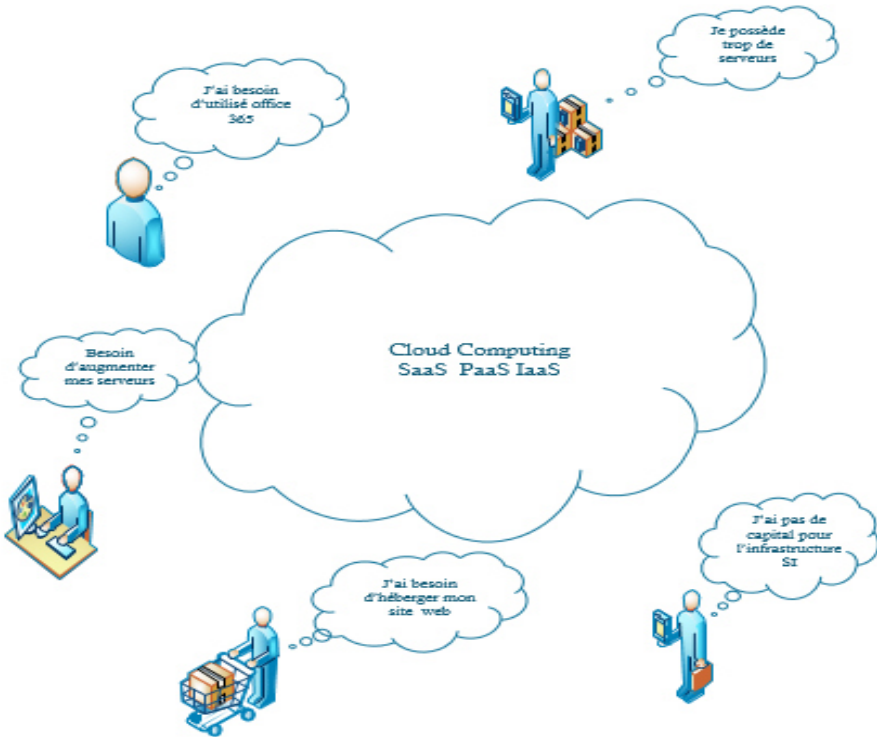

Figure I.3 : la vision du Cloud Computing

Selon le NIST le Cloud propose plusieurs services de caractéristiques essentielles notamment l'élasticité des ressources offertes par les services en Cloud, leur simplicité d'accès via le réseau, la mutualisation des ressources, l'agilité accrue des SI en Cloud ainsi que la facturation à l'usage des services Cloud :

- **Élasticité des ressources :**

Dans le Cloud, de nouvelles capacités peuvent être automatiquement mises à disposition des utilisateurs en cas d'accroissement de la demande. A l'inverse elles peuvent être rapidement mises en sommeil lorsqu'elles ne sont plus nécessaires.

Cette élasticité des services Cloud crée pour l'utilisateur final l'illusion d'une capacité infinie qui peut être mise en service à tout moment.

Pour les entreprises clientes, cette caractéristique d'élasticité permet par exemple de faire face aux pics d'activités que l'infrastructure interne n'aurait pu absorber, mais aussi d'envisager de nouvelles applications, par exemple des applications de calcul intensif nécessitant plusieurs centaines de machines pendant seulement quelques heures, applications que le cout d'une infrastructure interne aurait rendues impossibles sans le Cloud. Selon

Bernard Ourghanlian le directeur technique et sécurité de Microsoft, à propos du Cloud « Virtuellement, la puissance est infinie »

- **Un accès simple via le réseau :**

Les services de types Cloud sont accessibles au travers du réseau qu'il s'agisse du réseau LAN de l'entreprise pour un Cloud interne, d'un accès interne, d'un accès Internet ou d'une liaison louée IP (ou d'un accès VPN) pour un Cloud externe. Cet accès s'effectue au moyen de mécanismes et protocoles standards qui permettent notamment l'utilisation des services Cloud depuis de multiples types de terminaux.

- **Des couts contrôlés grâce à la mutualisation des ressources et aux effets d'échelles :**

Les ressources Cloud sont mises en commun et mutualisées afin de servir de multiples utilisateurs (plusieurs départements ou divisions dans le cadre d'un Cloud interne à l'entreprise ou plusieurs entreprises dans le cas d'un service en Cloud public).Cette mutualisation peut intervenir à de multiples niveaux qu'il s'agisse des ressources physiques (serveurs, stockage, réseau, serveur d'application …)

Grace à cette mise en commun des ressources, ces dernières sont réallouées de façon dynamique en fonction de la demande et des SLA (service level agrement) sans que l'utilisateur n'ait à effectuer quelque opération que ce soit. Chaque utilisateur est ainsi assuré de l'atteinte des objectifs de performances définis dans le cadre de son contrat.

- **Un SI plus agile :**

Dans le Cloud, l'utilisateur final du service peut provisionner rapidement les ressources dont il a besoin (serveurs, réseaux, stockage, applications…..) et disposer sans avoir à passer par de longues et complexes étapes de configuration manuelle. Ces capacités de « provisioning rapide » et de « self-service » permettent au SI de répondre plus vite aux besoins des métiers, aux demandes de changements, ainsi qu'aux exigences croissantes de time-to-market. Dans certains modèles de consommation, cette agilité est poussée à l'extrême notamment dans les modèles de types SaaS ou PaaS ou l'entreprise s'affranchit d'un grand nombre de contraintes de mise en place de ses applications.

- **Une facturation à l'usage :**

Avec le Cloud se généralise un nouveau mode de facturation à l'usage qui peut être résumé simplement : on ne paie que ce que l'on consomme réellement. Dans le Cloud d'infrastructure on paie ainsi au nombre de coeurs processeurs consommé, à la quantité de

mémoire utilisée, au nombre d'opération d'entrées/sorties effectuées ou à la quantité de données stockée. Dans un mode SaaS, on paie au nombre d'utilisateurs et en fonction de leur usage des ressources.

I.7. Domaines d'application du Cloud Computing :

Actuellement et sans que les utilisateurs se rendent compte, de nombreuses applications sont passées dans le Cloud :

– Domaine militaire : La technologie employée dans un environnement militaire doit être opérationnelle en toute circonstance. Qu'il s'agisse de garantir l'exécution correcte des opérations quotidiennes, le fonctionnement constant des communications internationales ou la gestion réussie de situation de crise, la solution militaire Camera manager (vidéo protection dans le Cloud) de Panasonic permet de garder un œil constant sur les organisations militaires. Il ne se limite pas à l'enregistrement et l'affichage en temps réel, mais il embarque également de nombreuses fonctions courantes de gestion vidéo-programmation, alarme, utilisation d'une API, gestion des utilisateurs, et est capable de supporter tout ce qu'on fait, quelles que soient les difficultés à surmonter.

– Domaine médical : Le Cloud a la capacité d'améliorer la collaboration dans l'industrie des soins de santé. En permettant aux professionnels de stocker et accéder aux données à distance. Les professionnels de santé aux quatre coins du monde peuvent accéder aux données des patients et appliquer les soins nécessaires au plus vite. De plus, la téléconférence, les mises à jour des dernières innovations en soins de santé et les conditions du patient en temps réel, permettent d'économiser du temps pour sauver des vies.

Followmed est une solution Cloud proposant de multiples fonctionnalités ayant pour objectif de simplifier le quotidien des professionnels de santé, gérez leur activité médicale, innovez leur communication sur le réseau social dédié aux professionnels de la santé, utilisez ce logiciel médical en Cloud pour stocker et accéder à l'ensemble de leurs données où qu'ils soient.

– Domaine de l'éducation : Le Cloud est un excellent levier technologique pour fournir des services collaboratifs extrêmement riches permettant de créer des sessions extrêmement interactives et de réaliser des opérations d'apprentissage à distance qui vont être complémentaires à l'enseignement traditionnel.

Par exemple des usages de type blending-learning (ou formation mixte) avec de la vidéo en apprentissage seul, du cours en présentiel et du cours virtuel connecté qui va permettre aux étudiants d'affranchir des limites de l'espace et offrir un panel de possibilité plus important.

Virtual Computing Lab (VCL) est une infrastructure Cloud qui fédère des ressources informatiques de plusieurs sites (serveurs, systèmes de stockage, instances de logiciels). Elle permet aux élèves des différents établissements primaires et secondaires de l'Etat ainsi qu'aux étudiants des différents campus de l'université d'accéder, où qu'ils se trouvent (en ce compris chez eux), à un pool de ressources techniques et pédagogiques évoluées et à jour.

– Domaine industriel : Pour pérenniser la croissance de l'industrie manufacturière, les fabricants doivent sans cesse améliorer la précision et la rapidité de leurs procédés, et optimiser chaque interaction avec les fournisseurs, distributeurs et prestataires de services. On assiste ainsi à la démocratisation des architectures technologiques "just in time" (ou dynamiques), basées sur des Clouds publics et privés, qui aident les fabricants à tenir leurs objectifs et à maintenir leur compétitivité [11]. Salesforce est une plateforme Cloud complète permettant de gérer les interactions entre les fournisseurs et leurs clients, et les aider à prospérer et réussir.

I.8. Challenges :

Le Cloud présente plusieurs défis :

- Sécurité : La sécurité est l'une des principales préoccupations entravant la propagation du cloud computing. Étant donné que la présence d'informations privées sur le cloud public dans des endroits accessibles à tous est une préoccupation pour les entreprises et les particuliers, qui peuvent préférer garder les données hors de l'accès public ; Surtout si ces informations sont sensibles ou de grande importance. De nombreuses organisations ont été créées dans le but de promulguer des lois de sécurité associées à l'utilisation du cloud computing. Cependant, les brèches qui surviennent parfois entravent encore l'avancée de l'idée du cloud computing (pénétration du cloud d'Apple, par exemple).

- Disponibilité : Le client d'un service de Cloud Computing devient très dépendant de la qualité du réseau pour accéder à ce service. Aucun fournisseur de service Cloud ne peut garantir une disponibilité de 100%.

- Confidentialité : Le problème de confidentialité des modèles de cloud computing est une faille que de nombreux utilisateurs de cloud critiquent, en raison de la capacité des fournisseurs de services cloud à accéder et à voir les informations sur leurs clients dans leurs clouds, que ce soit légalement ou illégalement. Par exemple, la société de télécommunications américaine AT&T a enregistré près de 10 millions d'appels vers ses clients, ce qui a créé un climat de mécontentement public et de rejet de la politique de l'entreprise.

- Engagement : Les fournisseurs de services cloud doivent se conformer aux lois et normes de leur pays d'accueil, ce qui peut créer un problème avec les clients car ils ne connaissent pas les lois du pays d'hébergement (qui ne sont généralement pas connues des clients. Par exemple, une entreprise peut placer ses serveurs cloud dans un pays autre que le pays auquel cette entreprise appartient, et devient ainsi obligé de suivre les lois et réglementations en matière d'information de ce pays, il peut donc ne pas convenir à certains clients.

- Légitimité et légalité : Cette négativité relève également des règles et lois du pays d'accueil, comme les droits de propriété intellectuelle existant dans certains pays qui peuvent être le pays d'hébergement du service cloud peut ne pas être compatible avec certains clients qui ne relèvent pas des lois de ces pays. Ce point peut permettre à certains fournisseurs de services de cloud computing d'accéder aux données et fichiers de certains clients et de prendre des mesures contre les contrevenants.

Mais malgré ces inconvénients, le cloud computing et ses services sont répandus dans le monde entier et les fournisseurs de services cloud continuent de développer leurs capacités et leurs services tout en essayant de surmonter ces inconvénients en échange des grands avantages offerts par le cloud computing. Aussi, la question de la sécurité reste ouverte à la recherche et au développement en utilisant diverses méthodes qui offrent une plus grande sécurité de l'information, que ce soit en cryptant ces données ou en utilisant d'autres méthodes.

I.9. Conclusion :

Dans ce chapitre, les définitions du Cloud Computing et des services qu'il fournit sont présentées, à savoir les applications en tant que service SaaS, les plates-formes en tant que service PaaS et l'infrastructure en tant que service IaaS.

Ensuite, les types de cloud standard, cloud public, cloud privé, cloud communautaire et cloud hybride ont été présentés. Les caractéristiques du Cloud Computing, ses avantages, ses

domaines d'application dans la réalité et les défis auxquels il est confronté sont abordés.

Dans le chapitre suivant, nous détaillerons la gestion des données dans le Cloud Computing.

II. Gestion des données dans le Cloud Computing

II.1 Introduction :

Bien que tout le monde ne soit pas d'accord sur la définition exacte du Cloud Computing [1], la plupart conviennent que la vision englobe un déplacement général du traitement informatique, du stockage et de la livraison de logiciels loin des serveurs de bureau et locaux, à travers le réseau et dans les centres de données de nouvelle génération hébergés par de grandes sociétés d'infrastructure telles que Amazon, Google, Yahoo, Microsoft ou Sun. Tout comme le réseau électrique a révolutionné l'accès à l'électricité, un il y a cent ans, libérant les entreprises de la nécessité de générer leur propre électricité et leur permettant de se concentrer sur leurs différenciateurs commerciaux, le Cloud Computing est estimé comme une révolution informatique, libérant les entreprises des grandes entreprises informatiques investissements en capital et leur permettant de se connecter à des ressources informatiques extrêmement puissantes sur le réseau.

Les applications de gestion des données sont des candidats potentiels au déploiement dans le Cloud. En effet, un système de base de données d'entreprise sur site s'accompagne généralement d'un coût initial élevé, parfois prohibitif, à la fois en matériel et en logiciel. Pour de nombreuses entreprises (en particulier pour les start-ups et les entreprises de taille moyenne), le modèle de Cloud Computing pay as-you-go, ainsi que le fait que quelqu'un d'autre se soucie de la maintenance du matériel, est très attrayant. Ainsi, le Cloud Computing rappelle le fournisseur de services applicatifs (ASP) et paradigmes de base de données en tant que service (DaaS). En pratique, les plateformes de Cloud Computing, comme celles proposées par Amazon Web Services, AT&T's Synaptic Hosting, AppNexus, GoGrid, Rackspace Cloud Hosting et, dans une certaine mesure, le HP/Yahoo/Intel Cloud Computing Testbed et l'initiative IBM/Google Cloud fonctionnent différemment des ASP et DaaS. Au lieu de posséder, d'installer et de maintenir le logiciel de base de données pour vous (souvent dans une architecture), les fournisseurs de Cloud Computing ne maintiennent généralement guère plus que le matériel et offrent aux clients une ensemble de machines virtuelles dans lesquelles installer leur propre logiciel. La disponibilité des ressources est généralement élastique, avec une puissance de calcul et un stockage apparemment infinis disponibles à la demande, dans un mode de paiement uniquement pour ce que vous utilisez des applications de gestion à déployer dans le Cloud. En raison du besoin toujours croissant d'analyses plus poussées sur plus de données dans le monde de l'entreprise d'aujourd'hui, ainsi qu'une

correspondance architecturale dans les options de déploiement actuellement disponibles, nous concluent que les applications de gestion de données principalement analytiques sont mieux adaptées au déploiement dans le Cloud que les applications de gestion de données transactionnelles. Nous décrivons ainsi un programme de recherche pour l'analyse de données à grande échelle dans le Cloud, en montrant pourquoi les systèmes actuellement disponibles ne sont pas parfaitement adaptés au déploiement dans le Cloud, et en faisant valoir qu'il existe un besoin pour un SGBD nouvellement conçu, spécialement conçu pour les plateformes de Cloud Computing.

II.2 Gestion des données dans le cloud

II.2.1 Caractéristiques de Cloud

La puissance de calcul est élastique, mais uniquement si la charge de travail est équilibré. L'un des avantages souvent cités du Cloud .l'informatique est son élasticité face aux conditions changeantes. Par exemple, lors de pics saisonniers ou inattendus en demande pour un produit commercialisé par une entreprise de e-commerce, ou en phase de croissance exponentielle pour un réseau social site Web de mise en réseau, des ressources de calcul supplémentaires peuvent être allouées à la volée pour gérer la demande en quelques minutes (au lieu des nombreux jours qu'il faut pour se procurer l'espace et les biens d'équipement nécessaire pour étendre les ressources de calcul en interne). De même, dans cet environnement, on ne paie que ce que on a besoin, donc des ressources accrues peuvent être obtenues pour gérer les pics de charge, puis libérées une fois que le pic a apaisé. Cependant, obtenir des ressources de calcul supplémentaires n'est pas aussi simple qu'une mise à niveau magique vers un plus grand, une machine plus puissante à la volée (avec des augmentations proportionnelles des processeurs, de la mémoire et du stockage local) ; plutôt, les ressources supplémentaires sont généralement obtenues en allouant des instances de serveur supplémentaires à une tâche. Par exemple, L'Elastic Compute Cloud (EC2) d'Amazon répartit les ressources de calcul en petites, grandes et très grandes instances de serveur privé, dont la plus grande ne contient pas plus de quatre cœurs. Si une demande ne peut pas prendre tirer parti des instances de serveur supplémentaires en déchargeant une partie de son travail requis sur les nouvelles instances qui fonctionner en parallèle avec les anciennes instances, alors avoir les instances de serveur supplémentaires disponibles ne sera pas d'une grande aide.

En général, les applications conçues pour s'exécuter sur une architecture sans partage (où un ensemble des machines accomplissent une tâche avec un chevauchement minimal des

ressources) sont bien adaptées à un tel environnement. Quelque les produits de Cloud Computing, tels que Google App Engine, fournissent non seulement une infrastructure de Cloud Computing, mais également une pile logicielle complète avec une API restreinte afin que les développeurs de logiciels soient obligés d'écrire des programmes qui peut s'exécuter dans un environnement sans partage et ainsi faciliter la mise à l'échelle élastique.

Les données sont stockées sur un hôte non fiable. Bien que cela puisse sembler insensé pour une entreprise de Cloud Computing société d'accueil à violer la vie privée de ses clients et à accéder aux données sans autorisation, une telle possibilité rend néanmoins certains clients potentiels nerveux. En général, le déplacement de données hors des locaux augmente le nombre des risques potentiels pour la sécurité et des précautions appropriées doivent être prises. De plus, bien que le nom « Cloud informatique » donne l'impression que les ressources de calcul et de stockage sont délivrées d'un emplacement, le fait est, bien sûr, que les données sont physiquement situées dans un pays particulier et sont soumises à des règles et règlements. Par exemple, aux États-Unis, le US Patriot Act permet au gouvernement d'exiger accès aux données stockées sur n'importe quel ordinateur ; si les données sont hébergées par un tiers, les données doivent être transmises à l'insu ou sans l'autorisation de la société ou de la personne utilisant le service d'hébergement. Puisque la plupart des fournisseurs de Cloud Computing donnent au client peu de contrôle sur l'endroit où les données sont stockées (par exemple, Amazon S3 n'autorise qu'un client de choisir entre les options de stockage de données des États-Unis et de l'UE), le client n'a pas d'autre choix que d'assumer la pire et qu'à moins que les données ne soient cryptées à l'aide d'une clé non située sur l'hôte, les données peuvent être consultées par un tiers à l'insu du client.

Les données sont répliquées, souvent sur de grandes distances géographiques La disponibilité et la durabilité des données sont primordiales pour fournisseurs de stockage en nuage, car la perte ou l'indisponibilité des données peut nuire à la fois aux résultats (en omettant de toucher objectifs fixés dans les accords de niveau de service [2]) et à la réputation de l'entreprise (les pannes font souvent l'actualité [3]).

La disponibilité et la durabilité des données sont généralement obtenues grâce à une réplication secrète (c'est-à-dire que les données sont automatiquement répliqué sans ingérence ni demande du client). Grands fournisseurs de Cloud Computing avec des centres de données répartis à travers le monde ont la capacité de fournir des niveaux élevés de tolérance aux pannes en répliquant les données sur de grandes distances géographiques. Le service de stockage dans le Cloud S3 d'Amazon réplique les données dans toutes les « régions » et « la

disponibilité zones" afin que les données et les applications puissent persister même en cas de défaillance d'un emplacement entier. Le consommateur doit être prudent pour comprendre les détails du schéma de réplication cependant ; pour exemple, l'EBS d'Amazon (Elastic Block Store) ne répliquera les données que dans la même zone de disponibilité et est donc plus sujet aux échecs.

II.2.2 Applications de gestion de données dans le Cloud

Les caractéristiques du Cloud décrites ci-dessus ont des conséquences claires sur le choix des applications de gestion de données à migrer vers le Cloud. Dans cette section, nous écrivons la pertinence de déplacer les deux plus grands composants du marché de la gestion des données dans le Cloud : gestion des données transactionnelles et gestion des données analytiques.

II.2.2.1 Gestion des données transactionnelles

Par « gestion des données transactionnelles », nous désignons le pain et le beurre de l'industrie des bases de données, les bases de données qui back banking, réservation de compagnies aériennes, e-commerce en ligne et applications de gestion de la chaîne d'approvisionnement. Ces applications reposent généralement sur les garanties ACID fournies par les bases de données et ont tendance à être assez gourmandes en écriture. Nous spéculer que les applications de gestion des données transactionnelles ne seront probablement pas déployées dans le Cloud, du moins dans un avenir proche, pour les raisons suivantes :

Les systèmes de gestion de données transactionnelles n'utilisent généralement pas une architecture sans partage. Le marché des bases de données transactionnelles est dominé par Oracle, IBM DB2, Microsoft SQL Server et Sybase . De ces quatre produits, ni Microsoft SQL Server ni Sybase ne peuvent être déployés à l'aide d'une architecture sans partage.

IBM a publié une implémentation sans partage de DB2 au milieu des années 90, qui est maintenant disponible en tant que « base de données Partitioning Feature” (DPF) complémentaire à leur produit phare [4], mais est conçu pour aider à faire évoluer les applications analytiques exécutées sur des entrepôts de données, et non sur la gestion des données transactionnelles [5]. Oracle n'avait rien partagé mise en œuvre jusqu'à très récemment (septembre 2008 avec la sortie d'Oracle Database Machine qui utilise une architecture sans partage au niveau de la couche de stockage), mais encore une fois, cette implémentation est conçue uniquement pour être utilisée pour entrepôts de données [6].

La mise en œuvre d'un système de base de données transactionnelle utilisant une architecture sans partage n'est pas triviale, car les données sont partitionnées sur plusieurs sites et, en général, les transactions ne peuvent pas être limitées à l'accès aux données d'un seul site.

Cela se traduit par des protocoles complexes de verrouillage et de validation distribués, et par l'envoi de données sur le réseau entraînant une latence accrue et des goulots d'étranglement potentiels de la bande passante du réseau. De plus, le principal avantage d'une architecture sans partage est son évolutivité [7]; cependant cet avantage est moins pertinent pour les données transactionnelles traitement pour lequel l'écrasante majorité des déploiements ont une taille inférieure à 1.

Il est difficile de maintenir les garanties ACID face à la réplication des données sur de grandes distances géographiques.

Le théorème CAP [8] montre qu'un système de données partagées ne peut choisir au plus que deux propriétés sur trois : cohérence, disponibilité et tolérance aux partitions. Lorsque les données sont répliquées sur une vaste zone, cela laisse juste la cohérence et la disponibilité pour un système à choisir. Ainsi, la partie « C » (cohérence) d'ACID est généralement compromis pour assurer une disponibilité raisonnable du système.

Afin d'avoir une idée des problèmes inhérents à la création d'une base de données répliquée sur un réseau étendu, il est intéressant de noter les approches de conception de certains systèmes récents. SimpleDB d'Amazon [9] et Yahoo PNUTS [10] implémentent tous deux des bases de données sans partage sur un réseau étendu, mais surmontent les difficultés de réplication distribuée en assouplissant les garanties ACID du système. En particulier, ils affaiblissent la cohérence en mettant en œuvre diverses formes de cohérence éventuelle/temporelle afin que toutes les répliques n'aient pas à s'accorder sur la valeur actuelle d'une valeur stockée (en évitant les protocoles de validation distribués). De même, les recherches menées par Brantner et. Al. ont constaté qu'ils devaient assouplir les garanties de cohérence et d'isolement dans la base de données sur laquelle ils s'appuyaient haut de la couche de stockage S3 d'Amazon [11]. Bigtable de Google [12] implémente une base de données répliquée sans partage, mais n'offre pas une API relationnelle complète et affaiblit la garantie « A » (atomicité) d'ACID. En particulier, il s'agit d'un simple magasin de lecture/écriture ; les transactions à usage général ne sont pas mises en œuvre (les seules actions atomiques sont séquences lecture-modification-écriture sur les données stockées sous une seule clé de ligne). Données SimpleDB et Microsoft SQL Server Les services

fonctionnent de la même manière. Le projet H-Store[13] vise à construire une base de données transactionnelle sans partage à grande échelle qui adhère aux garanties ACID strictes en utilisant une conception de base de données minutieuse pour minimiser le nombre de transactions qui accèdent aux données de plusieurs partitions ; cependant, le projet reste au stade de la vision, et la faisabilité de l'approche sur un jeu de données réel et une charge de travail de requête n'a pas encore été démontrée.

Le stockage de données transactionnelles sur un hôte non fiable comporte des risques énormes. Bases de données transactionnelles contiennent généralement l'ensemble complet des données opérationnelles nécessaires pour alimenter les processus métier critiques. Les données incluent des détails à la granularité la plus faible et incluent souvent des informations sensibles telles que des données client ou numéros de carte de crédit. Toute augmentation des failles de sécurité potentielles ou des violations de la vie privée est généralement inacceptable.

Nous concluons ainsi que les applications de gestion de données transactionnelles ne sont pas bien adaptées au déploiement dans le Cloud.

Malgré cela, il existe quelques entreprises qui vous vendront une base de données transactionnelle pouvant s'exécuter sur Amazon.

cloud : Postgres Plus Advanced Server et Oracle d'EnterpriseDB. Cependant, il n'y a pas encore de publication ou études de cas de clients mettant en œuvre avec succès une base de données transactionnelle critique à l'aide de ces nuages produits et, au moins dans le cas d'Oracle, la version cloud semble être principalement destinée à la sauvegarde de bases de données [14].

II.2.2.2 Gestion des données analytiques

Par « gestion des données analytiques », nous faisons référence aux applications qui interrogent un magasin de données pour une utilisation dans la planification d'entreprise, résolution de problèmes et aide à la décision. Données historiques ainsi que les données de plusieurs bases de données opérationnelles sont généralement tous impliqués dans l'analyse. Par conséquent, l'échelle des systèmes de gestion des données analytiques est généralement plus volumineuse que les systèmes transactionnels (les systèmes analytiques franchissent de plus en plus la barrière du pétaoctet). De plus, les systèmes analytiques ont tendance à être lus principalement (ou en lecture seule), avec des insertions occasionnelles par lots. La gestion des données analytiques représente 3,98 milliards de dollars [15] des Marché des bases de données de 14,6 milliards de dollars [16] (27 %) et croît à un taux de 10,3 % par an. Nous

spéculons que les systèmes de gestion de données analytiques sont bien adaptés pour fonctionner dans un environnement Cloud et seront parmi les premiers des applications de gestion de données à déployer dans le Cloud, pour les raisons suivantes :

L'architecture sans partage convient parfaitement à la gestion des données analytiques. Teradata, Netezza, Greenplum, DATAllegro (récemment acquis par Microsoft), Vertica et Aster Data utilisent tous une architecture sans partage (au moins dans la couche de stockage) dans leurs produits de SGBD analytiques, avec IBM DB2 et récemment Oracle ajoutant également produits analytiques sans partage. La quantité toujours croissante de données impliquées dans les charges de travail d'analyse de données est le principal moteur du choix d'une architecture sans partage, car l'architecture est largement considérée comme évolutive le meilleur [17]. De plus, les charges de travail d'analyse de données ont tendance à consister en de nombreuses analyses les agrégations et les jointures de schéma en étoile, qui sont toutes assez faciles à paralléliser entre les nœuds dans un rien partagé réseau. Enfin, les écritures peu fréquentes dans la charge de travail éliminent le besoin d'un verrouillage distribué complexe et engager des protocoles.

Les garanties ACID ne sont généralement pas nécessaires. Les écritures peu fréquentes dans les charges de travail de la base de données analytique, ainsi que avec le fait qu'il suffit généralement d'effectuer l'analyse sur un instantané récent des données (plutôt que sur jusqu'à la deuxième des données les plus récentes) fait le « A », « C » et « I » (atomicité, cohérence et isolement) d'ACID facile à obtenir. D'où les compromis de cohérence qui doivent être faits à la suite de la réplication distribuée des données dans les bases de données transactionnelles ne sont pas problématiques pour les bases de données analytiques.

Des données particulièrement sensibles peuvent souvent être exclues de l'analyse. Dans de nombreux cas, il est possible d'identifier les données qui seraient les plus dommageables si elles étaient consultées par un tiers, et soit les laisser hors du magasin de données analytiques, l'inclure uniquement après avoir appliqué une fonction d'anonymisation, ou l'inclure uniquement après le cryptage. De plus, des versions moins granulaires des données peuvent être analysées au lieu des données les plus détaillées et les plus détaillées.

Nous concluons que les caractéristiques des données et des charges de travail des applications de gestion de données analytiques typiques sont bien adaptées au déploiement dans le Cloud. La disponibilité élastique des ressources de calcul et de stockage du Cloud est facilement exploitée par une architecture sans partage, tandis que les risques de sécurité peuvent être

quelque peu atténués. en particulier, nous nous attendons à ce que le Cloud soit une option de déploiement privilégiée pour les entrepôts de données pour les entreprises de taille moyenne (en particulier celles qui ne disposent pas actuellement d'entrepôts de données en raison des dépenses d'investissement initiales importantes nécessaire au démarrage d'un projet d'entrepôt de données), pour des projets d'intelligence d'affaires soudains ou à court terme qui surviennent en raison de l'évolution rapide des conditions commerciales (par exemple, un magasin de détail analysant les habitudes d'achat dans le suite d'un ouragan) et pour les entrepôts de données orientés client qui contiennent une fenêtre de données d'entrepôt de données destiné à être consulté par le public (pour lequel la sécurité des données n'est pas un problème).

II.3 Disponibilité dans le Cloud Computing

Définition : Un service est dit hautement disponible (Highly avalable en anglais) si toute requête reçue par un nœud n'étant victime d'aucune défaillance retourne un résultat.

La haute disponibilité (HA pour « High Availability ») d'un service caractérise sa capacité à assurer son effective accessibilité durant une période donnée. Le service devra donc pouvoir détecter les points de défaillances et réduire l'impact de leur potentielle défaillance grâce à la mise en place de techniques de redondance et/ou réplication. [18]

II.3.1 Importance de la haute disponibilité dans le Cloud Computing

La disponibilité d'un service est donnée par la quantité de temps pendant laquelle le service est utilisé par les clients dans des conditions normales et anormales.

La haute disponibilité découle du fait que ses clients ont besoin d'un accès permanent au service[19].

L'indisponibilité de certains services à un impact négatif pour leurs clients, c'est le cas des institutions bancaires, des entreprises de télécommunications, des applications militaires ou des hôpitaux. Les infrastructures Cloud doivent offrir une disponibilité supérieure à 99,9% ; Donc la dégradation des performances est une préoccupation plus grave que les défaillances de ressources dans de tels environnements.

L'augmentation de la demande pour la disponibilité continue des systèmes de calcul haute performance (HPC : High-performance Computing systems) est évidente. C'est une étape majeure vers l'informatique de capacité, où les applications scientifiques demandent des

quantités considérables de temps (semaines et mois) sans interruption sur les machines HPC disponibles les plus rapides [20].

Ces systèmes de calcul haut de gamme (HEC : High-end Computing systems) doivent pouvoir fonctionner en cas de défaillance de manière à ce que leur potentiel ne soit pas rigoureusement dégradé. La haute disponibilité (HA) a longtemps joué un rôle important dans les applications critiques, comme dans les secteurs militaire, bancaire et des télécommunications.

II.3.2 Les solutions de disponibilité dans le Cloud :

L'indisponibilité est principalement due à des pannes matérielles et logicielles, au réseau de communication, dénis de service ou catastrophes naturelles. Les solutions de disponibilité existantes se basent soit sur la réplication et la redondance matérielle, soit sur le rajeunissement.

L'une des solutions basées sur le rajeunissement utilise le mécanisme de migration en live, elle consiste à répliquer au bon moment le contenu des machines virtuelles dégradées sur un nœud de veille qui joue le rôle du nœud principal en attendant la fin des mises à jour correctives. Cette technique a pour objectif de réduire la consommation de ressources et de produire également des améliorations significatives, car elle permet de rétablir le fonctionnement initial des machines virtuelles et donc la disponibilité prévue au départ est maintenue à condition que la migration se fasse au bon intervalle de temps avant la procédure de rajeunissement [21].

L'une des solutions basées sur la réplication décrit et surveille la disponibilité des ressources du Cloud, il est également utilisé par les clients pour décrire leurs besoins en terme de qualité de service (QoS) orienté disponibilité. Ce modèle QoS combine les caractéristiques du Cloud avec les attentes des clients pour surveiller la disponibilité du service.

L'idée est de faire correspondre une requête d'un client au nœud adéquat qui offre la meilleure QoS. Dans cette approche un algorithme de sélection du nœud de service offrant la meilleure QoS en termes de disponibilité parmi plusieurs nœuds est proposé et modélisé par un système d'équation.

La fonction objective représente le plus grand taux de QoS, tandis que les contraintes sont issues du modèle de QoS des demandes des clients. A la résolution de cette fonction, le nœud de service ayant ce score maximum sera attribué au service demandé par le client [22].

II.4 Cohérence des données dans le Cloud

Définition : Un stockage distribué cohérent se réfère souvent au fait que chaque mise à jour soit appliquée à tous les nœuds concernés en même temps logique [23].

Une conséquence de cette définition est que, si une donnée est répliquée sur des nœuds différents, elle serait toujours la même sur chaque nœud à chaque instant.

II.4.1 Modèles de cohérence

Un modèle de cohérence pour des données réparties spécifie un contrat entre un client et le système de gestion de données (avec engagement mutuel).

Les données réparties sont un ensemble d'éléments (un élément peut avoir une granularité quelconque : octet, enregistrement, fichier, etc.). Les opérations sont lire et écrire (éventuellement des opérations plus complexes, mais toujours divisées entre consultation et modification). Les modèles de cohérence les plus courants sont [24]:

II.4.1.1 Modèles sans synchronisation

- Cohérence forte : Dans les systèmes traditionnels de stockage et de bases de données distribuées, la manière instinctive et correcte de traiter la cohérence des répliques était d'assurer un état de cohérence forte de toutes les répliques dans le système à tout moment. La cohérence forte garantie que toutes les répliques sont dans un état cohérent immédiatement après une mise à jour, avant de retourner un succès. Ce modèle exige des mécanismes très couteux en termes de performance et de disponibilité et limitent l'évolutivité du système. Ce ne fut pas un problème dans les premières années des systèmes de stockage distribués, les performances nécessaires à l'époque n'étaient pas aussi importantes.

Cependant, à l'ère du Big Data et Cloud Computing, ce modèle de cohérence peut être pénalisant en particulier si une telle cohérence forte n'est en réalité pas requise par les applications [25].

- Cohérence séquentielle : Un système est cohérent séquentiellement si le résultat de n'importe quelle exécution est le même si les opérations de tous les opérations de tous les processus sont exécutées dans un ordre séquentiel, et les opérations de chaque processus individuel apparaissent dans cette séquence dans l'ordre spécifié dans son programme. Dans

un système essentiellement cohérent, tous les processus doivent s'entendre sur l'ordre des effets observés (on impose un ordre total sur les écritures). Les lectures locales à chaque processus lisent la dernière valeur écrite [26].

- Cohérence causale : Modèle plus faible que la cohérence séquentielle, car on ne considère que des événements reliés par une relation de causalité.

Exemple : Soit deux événements E1 et E2 tels que E1 E2 (précédence causale).

Alors tout processus doit "voir" E1 avant E2.[26].

- Cohérence FIFO : Des écritures réalisées par un même processus doivent être vues par tous les processus dans leur ordre de réalisation. Des écritures réalisées des processus différents peuvent être vues dans un ordre différent [26].

- Cohérence stricte : On parle de cohérence atomique, indivisible, stricte ou

non-interruptible quand une lecture renvoie toujours le résultat de la dernière écriture effectuée, ce qui implique l'existence d'une base de temps globale. Ce modèle de cohérence est le plus fort qui soit [27].

- Cohérence cache : Pour chaque emplacement d'une donnée, il existe un ordre total des écritures sue cette donnée. Les écritures sur différents emplacements peuvent être vues dans un ordre différent par différents processeurs[26].

- Cohérence éventuelle : Ce modèle est devenu très populaire dans les dernières années, car il assure une meilleure disponibilité, et de plus grandes performances.

On peut tolérer des données incohérentes pendant un certain temps

T, si la condition suivante est respectée : si aucune mise à jour n'a eu lieu pendant T, toutes les répliques deviendront progressivement la dernière version de la valeur. Ce type de cohérence peut poser un problème lorsqu'un client accède successivement à plusieurs répliques différentes d'une donnée. D'où une classe de modèles de cohérence définis à partir de la vue du client.

– Lectures monotones : Si un processus a lu la valeur d'une donnée, toute lecture ultérieure par ce même processus doit rendre la même valeur ou une valeur plus récente. Ce modèle garantie qu'un client ne reviendra pas en arrière.

– Écritures monotones : Si un processus exécute deux écritures successives sur une même donnée, la deuxième écriture ne peut être réalisée que quand la première a été terminée.

– Cohérence écriture-lecture : Si un processus écrit la valeur ’x’ sur une donnée ’d’, toute lecture ultérieure par ce processus doit retourner ’x’.

– Cohérence lecture- écriture : Si un processus lit une donnée ’D’, récupère sa version v, puis écrit sur cette même donnée la valeur ’y’, l’écriture modifiera partout la donnée ’D’ ayant une version au moins aussi récente que v.

II.4.1.2 Modèles avec synchronisation

– Cohérence faible : La mise en œuvre du modèle de cohérence forte impose, dans de nombreux cas, des limitations dans les choix de conception de systèmes et des performances de l’application. Pour surmonter ces limitations, le modèle de cohérence faible a été introduit. L’accès aux données en lecture et en écriture est considéré comme faible s’il remplit les trois conditions suivantes :

- Tous les accès à une variable de synchronisation partagée sont fortement (séquentiellement) commandés. Tous les processeurs perçoivent le même ordre des opérations.

- Aucun accès à une variable de synchronisation n’est fait par un processeur avant que tous les accès aux données précédentes aient été réalisés.

- Aucune lecture ou écriture ne peut être effectuée par un processeur tant qu’un accès à une variable de synchronisation n’est pas terminé.

II.5. Challenges dans l’environnement de réplication dans le Cloud

Les défis posés pour la réplication dans le Cloud se résument en :

a. Cohérence des données : Le maintien de l’intégrité et de la cohérence des données dans un environnement répliqué est une importance majeure. Les applications de haute précision peuvent nécessiter une cohérence stricte des mises à jour effectuées par les transactions.

b. Temps d’inactivité lors de la création d’une nouvelle réplique : Si une cohérence stricte des données doit être maintenue, Les performances sont gravement affectées si une nouvelle réplique doit être créée. Les sites ne seront pas en mesure de répondre à la demande en raison des exigences de cohérence.

c. Coût : Si les fichiers sont répliqués sur plusieurs sites, ils occupent plus d’espace de stockage. Cela nécessite une maintenance supplémentaire et devient une surcharge en stockant plusieurs fichiers.

d. Performances : La performance des opérations d'écriture peut être considérablement inférieure dans les applications nécessitant des mises à jour élevées dans un environnement répliqué, car la transaction peut avoir besoin de mettre à jour plusieurs copies.

II.6. Conclusion

La réalisation de ce chapitre nous a permis de comprendre l'importance de la disponibilité, la cohérence des données dans les bases de données distribuées, les avantages de la réplication, ainsi que les modèles de cohérence avec leur importance dans la synchronisation des échanges entre les processus. Dans le chapitre qui suit, nous allons étudier la réplication des données dans le Cloud Computing avec de plus des détails.

III.1. Introduction :

Le Cloud Computing est basé sur le concept de la visualisation qui consiste à faire fonctionner plusieurs systèmes sur un même serveur physique. Les principaux avantages étant d'offrir une large accessibilité, une grande fiabilité et une meilleure qualité de service, donnent au même temps naissance à certaines contraintes parmi elle devoir offrir une haute disponibilité aux clients. La réplication peut être une solution à ce problème, toutefois celle-ci soulève la question de la cohérence des données entre les répliques.

La réplication des données est un outil très efficace utilisé dans les systèmes distribués, pour donner plus de données disponibles et donc plus de performances d'accès aux données.

La réplication est une stratégie qui crée des copies des données et les stocke à divers endroits du réseau. Les coûts de création, de maintenance et de mise à jour des répliques posent des problèmes importants et difficiles.

Dans ce qui suit, dans ce chapitre, nous présenterons une étude théorique des stratégies de réplication dans les systèmes à grande échelle et certaines de leurs caractéristiques, telles que la mise à jour de réplication, les modes de réplication et la convergence, et discuterons de leurs avantages et inconvénients.

III.2. Quelque définition :

III.2.1. Réplication des données :

Une application de base de données repose sur un modèle client-serveur. Dans ce modèle, le client se connecte au SGBD pour passer des ordres. Ces ordres sont de deux natures : lecture (on parle alors de requêtes) ou mise à jour (on parle alors de transactions). Pour les transactions il a une modification des données sur le serveur, mais cela reste des ordres de courte durée. A l'inverse, dans le cas d'une lecture, il n'y a pas de modification des données mais les traitements peuvent être longs et porter sur une grande masse de données.

On comprend donc aisément que, dans le cadre d'un site web par exemple, un nombre important de requêtes (ou complètement) le serveur. Il existe plusieurs solutions pour palier à

ce genre de problème et, ça tombe bien, la réplication en est une l'objectif principal de la réplication est de faciliter l'accès aux données en en augmentant la disponibilité.
Soit parce que les données sont copiées sur différents sites permettant de répartir les requêtes, soit parce qu'un site peut prendre la relève lorsque le serveur principal s'écroule. Une autre application tout aussi importante est la synchronisation des systèmes embarqués non connectés en permanence. Ce qui peut se résumer à l'aide des trois types de scénarii suivants :

- deux serveurs distants sur lesquels les données doivent être consistantes ;
- deux serveurs, un comme serveur principal, l'autre comme serveur de backup à chaud ;
- plusieurs serveurs en cluster utilisés pour de l'équilibrage de charge et de la tolérance à la panne.

III.2.2. Objets, réplique et site :

Un objet est l'unité minimale dans un système de réplication. Une réplique est une copie de l'objet qui peut et remodifiée.
Elle est stockée sur un site. Les sites tels que les ordinateurs, les tablettes et les téléphones portables peuvent stocker des répliques de plusieurs objets.

III.2.3. Les mises à jour, propagation et exécution :

Lorsque l'utilisateur change l'état d'un objet, des modifications sont générées.
Ces des dernières ont un effet immédiat sur la copie locale de l'objet.
Ensuite, elles sont propagées aux autres répliques. Selon le type de propagation utilisé, ces mise à jour sont propagées sous forme d'état soude séquence.

III.3. La stratégie de réplication :

La réplique de données sur plusieurs sites est une solution efficace, elle permet d'améliorer le temps d'accès, la disponibilité de données, la consommation de largeur de bande et la scalability du système global.
Pour bénéficier au maximum du gain que peuvent apporter les répliques de données, leur placement stratégique dans le système est critique.
Les trois questions fondamentales auxquelles n'importe quelle stratégie de placement de répliques doit répondre sont :

— Quand les répliques devraient-elles être créées ?
— Quels dossiers devraient être répliqués ?
— Où devraient être replacées les répliques ?
Les réponses à ces questions nous mènent à différentes stratégies de réplication. Dans les auteurs ont décrits quelques stratégies de réplication dynamiques :
— Meilleur client (Best Client) : sélectionné le meilleur client pour une réplication basée sur le choix d'un chier demandé par le maximum de tâches.
— Cache simple (plain caching) : le site sauvegarde localement des copies des chiers nécessaires.
— Diffusion rapide (Fast Spread) : dans le cas des requêtes aléatoires, une donnée est copiée (répliquée) à chaque niveau sur le chemin vers le client. Quand l'environnement de la grille est hiérarchique : Après comparaison de certaines stratégies de réplication par rapport au temps de réponses moyen et à la bande passante totale utilisée, l'auteur de présente les stratégies qui ont les meilleurs résultats :

1. La stratégie de Cascade : la réplication est faite en inondant l'arbre qui modélise la grille (comme une fontaine) en utilisant les mesures de besoins à chaque niveau.
2. La stratégie de la Diffusion rapide : déjà. Dans l'auteur parle aussi de l'algorithme ADR (Adaptive Data Réplication) qui adapte la réplication des données dynamiquement en fonction des accès qui sont fait à celles-ci.

III.4. Principe de réplication :

La réplication de données (La réplication crée plusieurs copies d'une entité existante sur plusieurs nœuds), comme il est montré dans la FigureIII.1 est une solution efficace pour obtenir de bonnes performance set une meilleure disponibilité de données.
L'objectif principal de la réplication est, de faciliter l'accès aux données, tout en augmentant la disponibilité, soit parce que les données sont copiées sur déférents sites permettant de répartir les requêtes, soit parce qu'un site peut prendre la relève lorsque le serveur principal s'écroule.
Le traitement parallèle des demandes d'accès améliore les performances en laissant les utilisateurs accéder aux données dans un nœud voisin pour éviter l'accès aux données dans un site distant. La réplication fournit également un accès transparent et sans faille aux ressources en cas de défaillance du système et un meilleur temps de réponse.

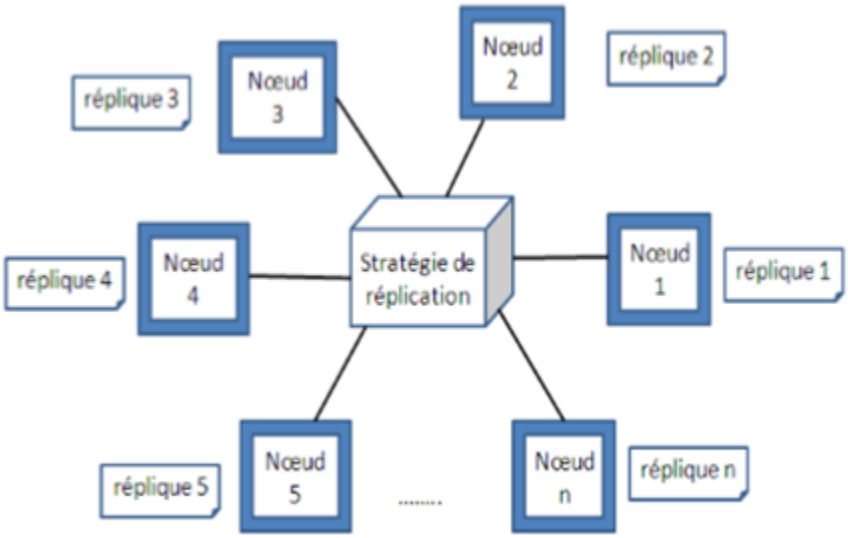

Figure III.1 : Technique de Réplication

III.5. Protocoles de réplication :

Dans la zone de recherche sur les systèmes distribués, la réplication est principalement utilisée pour fournir une tolérance aux pannes.

L'entité en cours de réplication est un processus. Deux stratégies de réplication ont été utilisées dans les systèmes distribués : réplication active et passive.

III.5.1. Protocole de réplication passive :

Dans un protocole de réplication passive (passive réplication ou primary/backup réplication) une seule copie, appelée copie primaire (primary copy), reçoit la requête d'un client et l'exécute. Les autres copies, nommée copies secondaires (secondary copies) ou copies de sauvegardes (backup), ne sont là que pour prendre le relais en cas de défaillance de copie primaire.

Afin d'assurer la cohérence des copies secondaires, la copie primaire leur diffuse régulièrement son nouvel état (un point de reprise).Cette méthode assure que toutes les copies

ont la même valeur même si les traitements sont non déterministes car il y a diffusion de l'état de la copie primaire vers les copies secondaires et non-exécution des requêtes.
Dans la réplication passive, il n'y a qu'un seul serveur (appelé primaire) qui traite les demandes des clients. Après le traitement d'une requête, le serveur principal met à jour l'état sur les autres serveurs (de sauvegarde) et renvoie la réponse au client.

Si le serveur principal échoue, l'un des serveurs de sauvegarde prend sa place. La réplication passive peut être utilisée même pour les processus non déterministes.
L'inconvénient de la réplication passive par rapport à l'actif est que, en cas d'échec, la réponse est retardée.

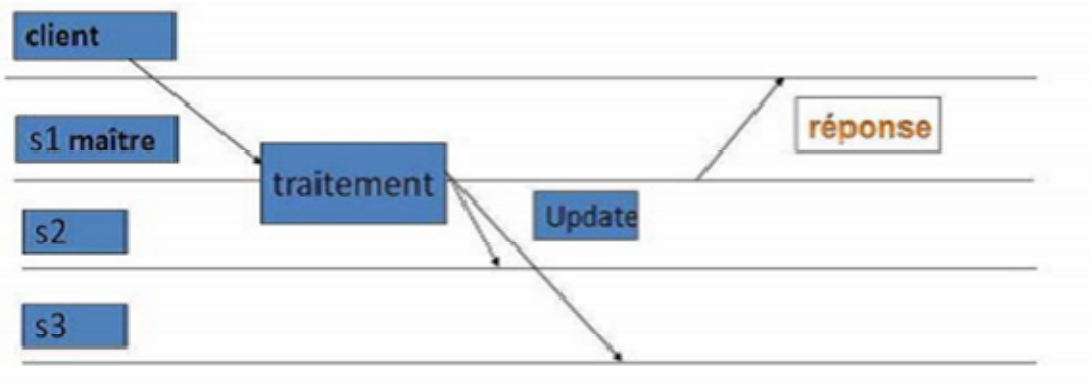

Figure III.2 : représente la réplication passive

III.5.2. Protocole de réplication active :

Dans un protocole de réplication active (active réplication ou state machine approche) toutes les copies jouent le même rôle. Elles reçoivent toutes la même séquence totalement ordonnée de requêtes de la part des clients, les exécutent de manière déterministe et renvoient la même séquence totalement ordonnée des réponses. La condition pour que toutes les copies reçoivent et exécutent toutes les requêtes dans le même ordre peut être décomposée en deux conditions distinctes :

— Toutes les copies non fautives reçoivent les mêmes requêtes.
— Toutes les copies non fautives exécutent les requêtes dans le même ordre relatif.

En réplication active, chaque demande client est traitée par tous les serveurs. Active Réplication a été introduite pour la première fois par Leslie Lamport sous le nom de réplication de machine d'état. Cela nécessite que le processus hébergé par les serveurs soit déterministe.

Déterministe signifie que, étant donné le même état initial et une même séquence de requêtes, tous les processus produiront la même séquence de réponse et finiront dans le même état final. Pour que tous les serveurs reçoivent la même séquence d'opérations, un protocole de diffusion atomique doit être utilisé.

Un protocole de diffusion atomique garantit que tous les serveurs reçoivent un message ou aucun, plus qu'ils reçoivent tous des messages dans le même ordre. Le grand inconvénient de la réplication active est que, dans la pratique, la plupart des serveurs du monde réel sont non déterministes.

La réplication encore active est le choix préférable lorsqu'il s'agit de systèmes temps réel qui nécessitent une réponse rapide même en présence de défauts ou avec des systèmes qui doivent gérer les défauts byzantins.

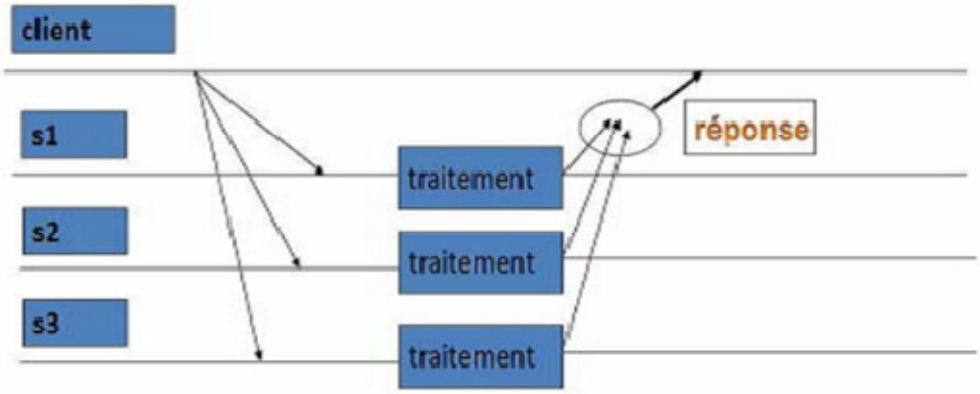

Figure III.3 : représente la réplication active

III.5.3. Protocole de réplication semi- active :

C'est un protocole hybride qui se situe entre les deux protocoles précédents, ou toutes les copies exécutent en même temps les requêtes des clients, mais une seule copie (leader) d'entre elles émet la réponse. Les autres copies (suiveurs) mettent à jour leur état interne et sont donc étroitement synchronisées avec le leader.

Figure III.4 : représente la réplication semi-active

III.6. Les types de réplication :

Microsoft SQL Server offre trois types de réplication. Chaque type de réplication correspond à des besoins différents des applications [28].
Selon les besoins de votre application, vous pouvez utiliser un ou plusieurs types de réplication dans une topologie.

III.6.1. Réplication de capture instantanée :

Les données sont copiées du site principal vers les sites secondaires. Les changements au niveau des sites secondaires doivent provenir du principal. Ainsi, seuls les secondaires sont interrogés, mais leurs données ne peuvent être modifiées par les clients.

La réplication de capture instantanée est plus appropriée quand les modifications de données sont substantielles mais peu fréquentes. Par exemple, si une entreprise commerciale gère des tarifs de produits, et que ces tarifs sont tous mis à jour en même temps, une à deux fois par an, il est recommandé de procéder à une réplication de l'intégralité de la capture instantanée des données modifiées.

III.6.2. La réplication transactionnelle :

La donnée est copiée complètement, suivie des mises à jour qui sont copiées périodiquement des sites principaux vers les sites secondaires.
Réplication transactionnelle est généralement utilisée dans des environnements de serveur à serveur.
L'application requiert l'accès aux états intermédiaires des données. Par exemple, si une ligne change cinq fois, la réplication transactionnelle permet à une application de répondre à chaque modification (par exemple activer un déclencheur), et pas simplement au résultat final des modifications de la ligne. Le serveur de publication a un volume très élevé d'activités d'insertion, de mise à jour et de suppression.
Le serveur de publication ou l'Abonné est une base de données non-SQL Server, par exemple Oracle Par défaut, les Abonnés à une publication transactionnelle doivent être traités en lecture seule, car les modifications ne sont pas propagées en sens inverse au serveur de publication. Cependant, la réplication transactionnelle n'offre pas d'options qui permettent des mises à jour sur l'Abonné.

III.6.3. La réplication de fusion :

Les données sont combinées de deux ou plusieurs sites vers un troisième site. Cette méthode est plus difficile à mettre en œuvre qu'une réplication instantanée.
Réplication par fusion est généralement utilisée dans des environnements de serveur à client.
L'application requiert le résultat des modifications des données au lieu de devoir accéder aux états intermédiaires des données. Par exemple, si une ligne change cinq fois sur un Abonné avant qu'il se synchronise avec un serveur de publication, la ligne ne change qu'une seule fois sur le serveur de publication pour refléter le résultat final des modifications (c'est-à-dire la cinquième valeur).

La réplication de fusion permet à plusieurs sites de fonctionner de manière autonome, pour fusionner ensuite leurs mises à jour en un résultat unique et uniforme. Étant donné que les mises à jour sont effectuées sur plusieurs nœuds, les mêmes données sont peut être mises à jour par le serveur de publication et par plusieurs Abonnés.
Par conséquent, des conflits peuvent se produire quand des mises à jour ont fusionnées ; la réplication de fusion fournit plusieurs moyens de gérer ces conflits.

III.7. Réplication basée sur le placement des répliques :

Toute stratégie de placement doit répondre à trois problématiques principales : quand faut-il créer des répliques ? Quel les données doivent être répliquées ? Et ou doit-on placer les répliques ? Ce qui conduit à différentes stratégies de placement des répliques, qui diffèrent d'un système à un autre.
Le placement des répliques est un critère de classification qui focalise sur le contenu de copies de la base. Si toutes les répliques possèdent une copie de chaque objet de la base, alors c'est une réplication totale. Sinon la réplication est partielle.

III.7.1. Aucune réplication :

Les données ne sont pas dupliquées entre les sources de système. L'accès aux données distribuées est fait par un serveur centralisant l'information sur le quelles différents nœuds se connectent. Donc si ce serveur tombe en panne. Alors toutes ces données sont perdues.

III.7.2. Réplication totale :

Cela implique de copier des données entières de la source vers le système cible, y compris les informations nouvelles, modifiées et présentes. Cependant, cette technique de réplication de données nécessite plus de puissance de traitement et augmente la charge sur le réseau. De plus, le coût augmente généralement car le maintien de la cohérence devient difficile lors de la copie de gros volumes de données.

III.7.3. Réplication partielle :

Dans cette technique de réplication de données, seule une partie des données est répliquée, comme les données mises à jour. Ainsi, il est plus rapide que la réplication de table complète car il traite un volume comparativement plus petit, ce qui réduit la charge du réseau et les problèmes de cohérence.

III.8. Avantages et problèmes de la réplication :

La réplication présente des avantages différents selon le type de réplication et les options choisies. Mais, l'intérêt général de la réplication est la disponibilité des données à tout moment et en tout lieu. Le recours à la technique de réplication procure certains avantages que nous pouvons énumérer comme suit :

— Disponibilité des données : Permettre à un serveur secondaire de prendre rapidement la place du serveur principal lorsque celui-ci échoue, ou alors les répliques répondent aux requêtes qui sollicitent une même donnée. S'il y a une perte de toute connectivité réseau, avoir une réplique complète signifie qu'il y a toujours accès à toutes les données.

— L'amélioration du temps de réponse des requêtes d'interrogation du fait d'une meilleure bande passante suite à la réplication de données distantes (Meilleur temps de réponse, diminution du temps de transfert des données, etc.).

— Amélioration de la fiabilité et de la sureté de fonctionnement,

— La redondance permet une meilleure protection contre la corruption de fichiers.

— Autonomie accrue : Les utilisateurs peuvent manipuler des copies de données hors connexion, puis propager leurs modifications aux autres bases de données lorsqu'ils sont connectés.

Néanmoins, la réplication peut présenter quelques problèmes tels que :

— Choix de l'entité 'à répliquer : quoi répliquer ?

— Degré de la réplication : combien répliquer ?

— Moment de réplication : quand répliquer ? Statique ou dynamique ?

— Placement des répliques : ou' placer les répliques ?

— Localisation des meilleures répliques ;

— Partitionnement du système : lorsque un lien entre deux nœuds tombe en panne, les répliques de la même donnée dans les deux systèmes doivent converger vers le même état ;

— Choix de la meilleure réplique d'un point de vue cohérence pour le traitement d'une requête.

— Le coût de mise à jour : la modification de la copie originale engendre la mise à jour de l'ensemble des copies. Le coût de mise à jour dépend donc du nombre de copies. De plus, la propagation des mises à jour nécessite généralement un certain temps ce qui entraine une

période de repos (pas de services). Une requête qui arrive pendant cette période sera donc retardée.

— Gestion de la cohérence des répliques : il est nécessaire de définir un protocole de cohérence qui puisse garantir que toutes les répliques d'une même donnée soient cohérentes.

III.9. Conclusion :

La réplication est un outil puissant pour fournir plus de données et plus de performances de latence dans les systèmes distribués.

Ce chapitre traite des concepts liés à la redondance des données dans le Cloud Computing. Nous avons tenté d'étudier brièvement chaque élément et de montrer la relation entre ces éléments, y compris la stratégie de réplication, le principe et les types de réplication. En outre, les avantages et des problèmes de réplication sont présentés, notamment en ce qui concerne la gestion de la cohérence pour différentes répliques.

IV.1. Introduction :

La réplication est une technique de gestion des données très connue qui a été couramment adoptée par de nombreux systèmes traditionnels, notamment les systèmes de gestion de bases de données (SGBD).les systèmes parallèles et distribués ,les systèmes mobiles et les autres systèmes à grande échelle incluant le P2P et systèmes de grille de données.

Dans ce chapitre, nous présentons les travaux liés sur la gestion de répliques dans les Clouds basés sur :

- Stratégies de réplication satisfaisant la disponibilité.
- Stratégies de réplication satisfaisant le temps de réponse.

Par la suite nous allons comparer les travaux liées à la réplication suivant les paramètres :

- Objectif de la stratégie (disponibilité /temps de réponse).
- La technique utilisée.
- Le respect du SLA « Service Level Agreement » « Accord de niveau de service » (positif /négatif).
- Gestion de répliques (centralisé/décentralisé).

IV.2. Travaux liés sur la gestion de répliques dans les Clouds :

Les performances d'une stratégie de placement de répliques peuvent être affectées par des décisions telles que :

- Le modèle du système utilisé
- Le modèle de coût adopté

-Les critères utilisés comme métriques pour améliorer les performances dans un environnement de Cloud, où les demandes d'accès fréquentes sont placées sur des données à grande échelle, un temps de réponse réduit et une haute disponibilité sont essentiels pour les clients du Cloud. Comme dans de nombreux systèmes, un certain nombre de stratégies de réplication de données ont été proposées [29] pour améliorer la qualité de service grâce à des contrats SLA (Service Level Agreement) dans les systèmes Cloud.

Une partie des stratégies de réplication de données proposées à base de SLA se concentrent sur la minimisation de la consommation d'une certaine ressource de Cloud [30], par exemple : *la bande passante*, tandis que d'autres traitent de la réplication de données pour satisfaire le SLA sans tenir compte de l'impact monétaire des décisions de réplication [31].

Les garanties de performance, par exemple : temps de réponse, ne sont souvent pas offertes aux utilisateurs par les fournisseurs de Cloud dans le cadre du SLA. La plupart des stratégies de réplication de données dans le Cloud se concentrent uniquement sur la satisfaction de la disponibilité [32]. Il y a aussi un manque de garanties de performance dans les SLA pour les Clouds commerciaux proposés par Amazon10, Google11 et Microsoft12. Afin de combler cette lacune, plusieurs travaux ont été proposés [33] dans la littérature pour proposer des mécanismes permettant d'inclure des garanties de temps de réponse dans la SLA. En ce qui concerne la réplication des données, seules quelques études s'intéressent particulièrement à l'amélioration du temps de réponse [34]. En outre, encore moins de ces études [35] prennent en compte l'impact économique de la réplication des données.

La réplication de données est une partie intégrale du Cloud et de nombreux systèmes de gestion de données distribués sur le Cloud, tels que Google File System (GFS) et Hadoop File System (HDFS), implémentent déjà une forme de réplication. À titre d'exemple, HDFS utilise des modèles de réplication afin d'améliorer à la fois les performances et la disponibilité [36].

Cependant, à mesure que l'échelle des systèmes de Cloud’s s'élargit et que le nombre de centres de données augmente à l'échelle, il est nécessaire que les stratégies de réplication de données répondent à cette tendance.

Les sous-sections suivantes classent les stratégies de réplication de données existantes dans les systèmes de Cloud selon leurs objectifs. Étant donné que le temps de réponse et/ou la disponibilité sont des critères les plus considérés dans chaque processus de réplication de données et que la contribution de cette thèse porte sur la satisfaction de ces critères, la classification est délibérément basée sur la satisfaction de l'objectif de la disponibilité ou la satisfaction de l’objectif du temps de réponse.

Bien sûr, les limites de la classification des stratégies de réplication des données dans le Cloud peuvent parfois ne pas apparaître clairement en raison d'un certain chevauchement des critères de classification. Par exemple, de nombreuses stratégies qui mettent l'accent sur une disponibilité accrue observent également une amélioration des performances du temps de réponse, même si la performance peut ne pas être un objectif dans de telles stratégies. Cet avantage de la performance se produit en raison de plusieurs réplications d'ensembles de données.

IV.3. Stratégies de réplication satisfaisant la disponibilité :

L'objectif le plus largement considéré par la réplication de données dans le Cloud est sans doute la satisfaction d'un niveau de disponibilité souhaité. La disponibilité des ensembles de données dépend souvent de la disponibilité des nœuds qui les hébergent ou de la charge de leur système, car les noeuds surchargés ne pourront pas répondre aux demandes. La disponibilité croissante des données doit donc être satisfaite en augmentant le nombre de répliques en conséquence.

[Yuan et al. 2010] ont proposé une stratégie de clustering k-means basée sur une matrice pour le placement de données dans des systèmes cloud. Dans leur stratégie, ils tentent de conserver les ensembles de données dans un seul centre de données, dans le but que lorsque les tâches sont ordonnancées dans ce centre de données, la plupart, sinon la totalité, des données dont ils ont besoin sont stockées localement. Leur idée était d'étudier et de déterminer la matrice de dépendance pour toutes les données d'application, ce qui montre les dépendances entre tous les ensembles de données, y compris les ensembles de données qui peuvent avoir des emplacements fixes. Ensuite, ils ont appliqué une technique de clustering [McCormick et al. 1972] pour grouper la matrice et la partitionner de telle sorte que les ensembles de données dans chaque partition soient fortement dépendants les uns des autres. Les résultats de la simulation indiquent qu'avec cette stratégie, le déplacement de données entre les centres de données est considérablement réduit par rapport à un placement de données aléatoires.

La gestion de la réplication dynamique à base de coût (cost-effective dynamic replication management CDRM) [37] souligne que le fait d'avoir trop de répliques n'augmente pas la disponibilité mais entraîne des rendements réduits. Le CDRM calcule et maintient un nombre minimum de répliques pour satisfaire un niveau de disponibilité donné. Dans le stockage en nuage considéré, les ensembles de données sont fragmentés en plusieurs blocs. Les auteurs calculent la disponibilité des ensembles de données en fonction de la disponibilité de chacun de leurs fragments.

Avec ces informations, CDRM calcule un nombre requis de réplicas pour chaque fragment. Selon CDRM, le placement de répliques est effectué de manière à équilibrer la charge sur tous les sites et est basé sur plusieurs paramètres tels que la puissance du processeur, la capacité de la mémoire, la bande passante du réseau et la saturation du système de stockage des noeuds de données. En conséquence de la charge équilibrée, les auteurs observent également des performances d'accès accrues. L'étude de la relation entre le nombre de répliques et le niveau de disponibilité a également fait l'objet d'autres études [38]. Une autre approche très similaire

au CDRM [Sun et al., 2012a] traite également de la recherche du nombre minimum de répliques du point de vue de la popularité des données dans les clusters de stockage du Cloud. Dans ce travail, pas toutes les données, mais seulement un sous-ensemble de fichiers qui dépasse un seuil d'accès sont concernés pour la réplication.

[Li et al. 2011] ont présenté une nouvelle stratégie de coût de réplication dynamique des données, appelée (Cost effective Incremental Replication CIR). Cette stratégie a pour objectif de réduire les coûts de stockage et de répondre aux exigences de fiabilité des données en même temps. CIR est une stratégie de fiabilité des données pour les applications Cloud où l'accent est mis sur la gestion rentable de la question de la fiabilité des données dans un centre de données. L'approche proposée dans CIR consiste à calculer le point de temps de création de la réplique, qui indique la durée de stockage à laquelle l'exigence de fiabilité peut être satisfaite. Afin de garantir l'exigence de fiabilité, CIR augmente dynamiquement le nombre de réplicas en prédisant quand une réplique supplémentaire est nécessaire. Le modèle de stockage de données utilisé dans CIR exploite la relation entre le nombre de réplicas et la durée de stockage. Les résultats de la simulation démontrent que la stratégie CIR peut réduire considérablement le nombre de répliques dans un centre de données et, par conséquent, le coût de stockage de l'ensemble du système de stockage peut être considérablement réduit. Néanmoins, CIR n'a pas examiné la question des compromis entre le coût et la performance.

Il est également possible d'obtenir une qualité de service souhaitée du point de vue de la disponibilité en calculant le niveau minimum de redondance par rapport aux ressources Cloud hétérogènes [39]. Dans ces environnements hétérogènes, le calcul de la disponibilité est effectué par rapport à chaque site de Cloud ayant des niveaux de fiabilité différents. Cette hétérogénéité de la fiabilité détermine également le nombre de répliques qu'un site de Cloud peut également héberger. La question de la fiabilité peut également être traitée comme un problème de tolérance de panne en modélisant mathématiquement les taux de défaillance des sites [40]. De cette façon, l'étude mentionnée permet de satisfaire l'objectif de disponibilité à travers un cadre de tolérance aux pannes basé sur la réplication des données. [Hussein et Mousa, 2012] ont proposé une stratégie de réplication de données qui sélectionne de manière adaptative des fichiers de données pour les répliquer afin d'améliorer la fiabilité globale du système et de répondre à la qualité de service requise. La stratégie proposée étudie la disponibilité et l'accès efficace de chaque fichier dans le centre de données, et analyse comment améliorer la fiabilité des fichiers de données en fonction de la prédiction de l'accès de l'utilisateur aux blocs de chaque fichier. La stratégie proposée vise à améliorer la disponibilité et l'accès efficace des fichiers dans les centres de données. Cette stratégie

détermine les fichiers populaires identifiés en analysant l'historique des accès du fichier. La popularité d'un fichier est déterminée par le taux d'accès à ce fichier par les clients. Une fois que le nombre d'accès d'un fichier populaire dépasse un seuil prédéfini, le système de réplication est déclenché. Par conséquent, la stratégie adaptative trouve l'emplacement de réplication approprié en fonction d'une recherche heuristique du meilleur facteur de réplication de chaque fichier. Les résultats de la simulation montrent que la stratégie adaptative améliore la disponibilité du système de Cloud étudié, mais ne tient pas compte de l'équilibrage de la charge du système.

La modélisation d'une stratégie de placement de données pour améliorer la disponibilité peut également être réalisée à l'aide de modèles de coûts économiques personnalisés. Certains présentent un modèle d'enchères pour implémenter une politique de placement de répliques dans un environnement de stockage Cloud à grande échelle [41]. Si le niveau de disponibilité souhaité ne peut pas être maintenu, une enchère est lancée pour déterminer le placement d'une nouvelle réplique. Le prix de l'enchère dépend de plusieurs propriétés des noeuds, notamment la probabilité de défaillance, la bande passante du réseau et l'espace disponible. Une augmentation des performances est également observée comme un avantage consécutif de la réplication des données. D'autres traitent un problème similaire dans un contexte multi-Cloud[42], où les utilisateurs reçoivent des services de plusieurs fournisseurs de Cloud. Ces exemples utilisent un modèle économique personnalisé qui est adapté pour prendre en compte l'hétérogénéité causée par les variations de ressources et de prix entre plusieurs fournisseurs de Cloud. Comme c'est le cas des stratégies existantes, ce modèle économique est utilisé pour minimiser un modèle de coût [43].

Certaines autres stratégies traitent de la disponibilité comme un problème de fiabilité qui concerne les sites du Cloud [44]. Cependant, la disponibilité est un objectif considéré fréquemment ; ce qui distingue ces deux stratégies est leur prise en compte d'une fonction de coût dans la satisfaction de la disponibilité. En déterminant la durée de conservation des réplicas en fonction d'une mesure de fiabilité, ces stratégies peuvent optimiser le nombre de réplicas pour chaque fichier, économisant ainsi les coûts du fournisseur sur la ressource de stockage.

Dans un autre travail similaire, Skute est présenté par [Bonvin et al. 2010a] en tant que stratégie de réplication de données basée sur une économie virtuelle prenant en compte l'utilité marginale, l'utilisation du stockage et la charge de requêtes. Les nœuds virtuels agissent de manière autonome et annoncent périodiquement leur loyer aux autres nœuds. De plus, les nœuds accumulent également de la richesse en répondant à des requêtes et dépensent

cette richesse sur d'autres nœuds pour y stocker des répliques, en fonction de leur loyer. Skute auto-organise la configuration de la réplique parmi les nœuds virtuels afin de minimiser les coûts de communication tout en maximisant les bénéfices économiques. Même si les stratégies de ce type utilisent une forme d'économie virtuelle, ce n'est pas un modèle de coût monétaire réel de la relation fournisseur-locataire [45].

L'augmentation de la disponibilité en répliquant les données populaires plus près des emplacements qui génèrent le plus de demandes est également un problème fréquemment étudié. Certaines stratégies proposent même des mécanismes pour prédire les futurs accès à partir des enregistrements d'accès historiques afin de répliquer les données de manière préventive afin de répondre à l'augmentation de la demande [46]. De plus, d'autres stratégies traitent de la popularité des données qui atteint un sommet pendant une courte durée, puis diminue progressivement [47]. Afin de faire face à l'explosion rapide de la popularité, ces stratégies de réplication de données répondent rapidement et modifient la configuration de la réplique en conséquence. L'augmentation de la localisation des données en fonction de la popularité diminue également la consommation du réseau [48].

[Zeng et al. 2016] traitent de la création et du placement de répliques dans des systèmes de stockage de Cloud en gardant une charge équilibrée. Dans leur travail, la hiérarchie du cloud est décrite sous forme des fournisseurs de services qui achètent des services auprès de vendeurs Cloud et, à leur tour, des clients qui achètent des services auprès de ces fournisseurs de services. Les auteurs visent à minimiser les coûts des fournisseurs de services tout en maximisant l'utilisation du stockage pour les utilisateurs. Ils mesurent la qualité du service et calculent une mesure d'importance des ensembles de données. Ce qu'il faut répliquer est alors déterminé en fonction de la métrique d'importance. Le placement des répliques est effectué de manière à maximiser le volume de transfert de données par dépense unitaire.

Un objectif intéressant de la réplication des données est de cibler spécifiquement certaines dépenses particulièrement intéressantes dans le Cloud. À cet égard, certaines stratégies de réplication de données [Boru et al., 2015a, b] se concentrent sur la réduction de la consommation d'énergie et des retards de communication par la réplication des données. Ces stratégies tiennent compte des fréquences d'accès aux données pour une reconfiguration de réplique périodique. Au cours de cette évaluation périodique, la stratégie de réplication prédit une valeur future des ensembles de données incluant l'énergie et la demande de bande passante. Selon cette estimation, un emplacement approprié est trouvé pour les répliques afin de minimiser leur consommation d'énergie et de bande passante dans des intervalles de temps ultérieurs.

Une autre stratégie [Maheshwari et al., 2012] poursuivant un objectif similaire en intégrant une stratégie de placement de données qui soit reliée au contrôleur d'énergie d'un cluster de stockage. De cette façon, les réplicas peuvent être reconfigurés pour mettre à l'échelle le cluster vers le haut ou vers le bas et désactiver les sites inactifs en fonction des modifications de la charge de travail afin d'économiser de l'énergie.

IV.4. Stratégies de réplication satisfaisant le temps de réponse :

Alors que dans les stratégies, basées sur la disponibilité, de la sous-section précédente, la performance et le temps de réponse ne figure généralement pas parmi les garanties de service, mises en avant, dans le contrat SLA (Service Level Agrement) du client [Tos et al., 2016]. Les stratégies de cette sous-section prennent généralement en compte la qualité de service et le temps de réponse et ajustent de manière élastique le nombre de répliques d'une manière ou d'une autre pour garantir des garanties de performance à l'utilisateur.

Certaines stratégies telles que RepliC [Sousa et Machado, 2012] effectuent cette tâche dans un environnement de base de données élastique à de multiples utilisateurs.

RepliC surveille l'utilisation du système par rapport aux changements de la charge de travail afin de gérer la variation en dirigeant les transactions vers les réplicas avec les ressources disponibles. Par conséquent, la stratégie RepliC satisfait la qualité de service avec des violations de SLA minimales.

L'augmentation de la localisation des données, plus précisément la localisation spatiale grâce aux rendements de la réplication des données, permet d'améliorer les performances en plaçant stratégiquement les répliques plus près des sites demandeurs.

Même si certaines stratégies ne considèrent pas directement la performance comme un objectif, une augmentation de performance est observée en conséquence de la réplication des données [49]. Une autre forme de croissance de localisation des données pour améliorer les performances se base sur la localité temporelle. De cette manière, certaines stratégies font valoir que les ensembles de données fréquemment consultés resteront probablement populaires à l'avenir [Jayalakshmi et Rashmi, 2015] et décideront ce qu'il faut répliquer selon l'évaluation de la popularité mentionnée. C'est une stratégie simple mais efficace pour satisfaire les garanties de performance comme en témoigne la stratégie RTRM [Bai et al.,2013]. Lorsque l'accès à des données populaires entraîne un temps de réponse moyen supérieur à celui souhaité, ces ensembles de données sont répliqués pour résoudre le problème de performances. Ce qui est intéressant à propos de RTRM, c'est qu'il ne reproduit pas toutes les données, sauf les données populaires, afin de conserver l'utilisation des ressources.

Le modèle d'optimisation multi-objectif pour la réplication des données est une autre approche intéressante, telle que décrite par la stratégie MORM [Long et al.,2014]. Établir des modèles mathématiques pour plusieurs objectifs, y compris la disponibilité, le temps de réponse, la latence du réseau ; le placement de données peut être effectué en fonction d'une fonction d'optimisation qui prend en compte ces objectifs. De cette façon, il est possible de satisfaire plus d'un objectifs simultanément.

L'idée de provisionnement dynamique des données est également utilisée dans les systèmes de gestion de bases de données dans le Cloud. [Sakr et Liu, 2012] ont mis en place une stratégie orientée sur le client, prenant en compte la satisfaction de ses objectifs, dans laquelle les serveurs de base de données sont mis à l'échelle en fonction des exigences demandées par le client. Ils ont choisi le temps total d'exécution des transactions en tant qu'objectif principal du processus de décision.

Dans la stratégie proposée, le système de Cloud est étroitement surveillé et les fournisseurs de Cloud définissent de manière déclarative des règles spécifiques aux applications pour mettre à l'échelle les ressources de manière adaptative.

Tandis que l'avantage de la performance peut être obtenu en équilibrant la charge des sites, il est également possible d'augmenter les performances en réduisant les retards de réseau dans l'accès aux données. [Sharov et al, 2015] présentent une telle stratégie de réplication de données pour le stockage en Cloud basé sur les leaders. Une réplique de chaque ensemble de données est choisie en tant que leader qui coordonne les tâches de réplication. Alors que toutes les répliques peuvent être utilisées pour une opération de lecture, les leaders répondent souvent aux lectures ou à l'écriture de transactions. Les auteurs déclarent que le placement des leaders a un impact sur la performance. Ils proposent trois algorithmes pour gérer le placement des leaders, des électeurs et des répliques afin de réduire la latence des opérations d'accès aux données. Ils montrent que leur stratégie améliore la latence de l'accès aux données jusqu'à 50% dans un système de fichiers distribué.

Ce n'est pas toutes les stratégies de réplication de données qui traitent des performances en tant que mesure temporelle du traitement des données (temps de réponse). L'objectif de performance peut consister simplement à fournir les données au demandeur. En ce sens, AREN [Silvestre et al., 2012] est un bon exemple d'application de SLA en tant que garantie de livraison de données. AREN effectue une réplication de données prenant en charge la popularité afin d'optimiser l'utilisation de la bande passante afin de réduire les violations de SLA et de réduire la consommation de stockage pour améliorer l'expérience utilisateur globale.

Sur un objectif de performance similaire, [Zhao, 2013] traite du délai de réplication, c'est-à-dire de la rapidité avec laquelle un serveur de base de données actualise toutes les répliques. Si le délai de réplication est supérieur à un seuil prédéterminé, de nouvelles répliques sont créées pour résoudre la dégradation des performances. Cette stratégie vise à satisfaire le SLA d'une manière orientée client, d'où le bénéfice du fournisseur n'est pas considéré. Dans une autre étude [Sakr et al.,2011], on discute d'un système de gestion de base de données élastique et basé sur le cloud qui permet la gestion automatique des tâches de gestion des données, y compris une implémentation basée sur des règles de réplication.

[Rajalakshmi et al. 2014] ont présenté une architecture dynamique de sélection et de placement de réplicas (DRSP) pour gérer les réplicas dans l'environnement cloud. La stratégie proposée se concentre sur la conception d'un algorithme pour la sélection et le placement optimal des répliques afin d'augmenter la disponibilité des données dans le Cloud. Dans une première phase, la stratégie consiste à localiser et créer des répliques à l'aide du catalogue et de l'index. L'index est utilisé pour stocker le fichier de réplique dans un site candidat local ou distant. Dans la seconde phase, la politique détermine s'il y a suffisamment d'espace, ou non, dans le site candidat pour stocker le fichier demandé. L'évaluation expérimentale montre que DRSP stocke efficacement les données dans des sites sélectionnés et améliore les performances d'accès et l'utilisation de la bande passante. De plus, cette stratégie n'a pas traité le problème d'utilisation du stockage et la cohérence du fichier de données.

[Kumar et al. 2014] ont proposé SWORD, un système de placement de données et de sélection de répliques qui tient compte de la charge de travail. Les auteurs ont introduit une nouvelle mesure nommée durée de requête, qui est le nombre moyen de noeuds utilisés dans l'exécution d'une requête. Leur approche vise à minimiser la durée des requêtes afin de réduire les frais généraux de communication, la consommation de ressources, l'empreinte énergétique et les coûts de transaction. Ils affirment que SWORD gère la dégradation des performances avec un repartitionnement incrémental des données. Bien que le bénéfice des fournisseurs ne soit pas un objectif de cette étude, les auteurs montrent l'efficacité de leur travail en effectuant une analyse expérimentale pour mesurer la durée de la requête et les délais de transaction.

La minimisation du coût des ressources peut également être envisagée dans un contexte de Cloud hétérogène [Mansouri et al., 2017]. Les centres de données situés dans différents endroits peuvent avoir des coûts variables pour le stockage, le réseau, etc… pour placer des répliques. Une stratégie de réplication doit prendre en considération cette tarification hétérogène tout en répondant par la réplication aux futures variations des charges de travail. Dans de nombreux autres environnements distribués, la réduction des transferts réseau entre

les data-centers via la réplication garantit à la fois une meilleure performance due à un accès rapide aux données et minimisation des coûts des fournisseurs en utilisant moins souvent des liaisons réseau plus coûteuses [Vulimiri et al. ., 2015].

Une stratégie de réplication de données dynamique est proposée par [Gill et Singh, 2016] pour un environnement Cloud hétérogène. Leur stratégie maintient les performances et la disponibilité à un niveau souhaité tout en optimisant le coût de la réplication. Le coût de la réplication est calculé en fonction du coût de l'unité de réplication par centre de données et de la présence ou non d'une réplique. Une métrique appelée facteur de réplique est utilisée pour calculer les fréquences d'accès pondérées. Le facteur de réplique est comparé à un seuil pour prendre une décision de réplication. Le placement des répliques est effectué avec une technique d'optimisation qui obtient le coût de réplication le plus bas avec une valeur de réplique élevée.

[Janpet et Wen, 2013] ont conçu une stratégie de réplication de données pour minimiser le temps d'accès aux données en trouvant le chemin d'accès le plus court aux objets de données. Ils modélisent la fréquence d'accès, le délai et le budget de réplication pour trouver le nœud le plus proche et le plus approprié pour le placement des répliques. Le budget de réplication est prédéfini et il est seulement utilisé comme un facteur limitant pour les utilisateurs de manière à réguler le nombre de répliques. Une relation économique détaillée entre les utilisateurs et le fournisseur de Cloud n'est pas abordée. L'étude expérimentale montre qu'en plaçant les objets de données plus près des nœuds avec une fréquence d'accès élevée, le temps de réponse est amélioré.

Dans le travail de [Radi 2016], l'auteur a présenté une stratégie de placement de réplicas basée sur AHP (processus hiérarchie analytique) dans le stockage de données dans les Clouds (RPAHP). Le placement de répliques a été modélisé comme un problème classique de prise de décision multi-attributs (MADM). Le processus de hiérarchie analytique (AHP) est une technique structurée pour organiser et analyser des décisions complexes. RPAHP a pris en compte six attributs (critères), à savoir CPU, mémoire, bande passante, utilisation du processeur, utilisation de la mémoire et utilisation de la bande passante, qui caractérisaient le centre de données et affectaient les performances du DC. Nous avons remarqué que les critères d'optimisation qui ont été choisis pour le placement de répliques ne concernaient que les composants du DC. L'auteur a négligé les critères jugés beaucoup plus importants pour un meilleur choix de sites candidats pour le placement de réplicas tels que : le temps de réponse, la variance de charge et la fréquence de répliques. La méthode AHP a également été utilisée dans d'autres travaux de [Xiuguo et Hongjun 2016]. Un algorithme basé sur l'évaluation

complète floue, basé sur la méthode AHP, a été implémenté. Un modèle QoS (qualité de service) de stockage en Cloud a été proposé et qui a été introduit pour traiter les contraintes individuelles de sensibilité (IQS).

[Mansouri, 2016] a proposé une approche appelée «stratégie de réplication de données adaptative (ADRS) dans le cloud computing pour l'amélioration des performances». Elle a proposé une stratégie de placement de réplicas basée sur cinq paramètres : le temps de service moyen, la probabilité d'échec, la variance de la charge, la latence et l'utilisation du stockage. À la base de ces paramètres, elle a calculé la fonction de coût pour chaque site. Plus la valeur Cost-Function est petite, meilleure est la condition physique du site. Lorsqu'une nouvelle réplique a été générée, l'ADRS a choisi l'un des sites légèrement chargé pour stocker la nouvelle réplique. Les résultats de la simulation ont montré que l'ADRS donnait de meilleures performances par rapport à d'autres travaux tels que CIR, DRSP et CDRM, selon les cinq paramètres cités plus haut.

II.5 Classifications existantes :

Un nombre significatif des stratégies de réplication de données ont été proposées dans la littérature qui ont été classées selon les caractéristiques suivantes :

II.5.1. Réplication statique vs. Réplication dynamique :

Lorsque le modèle de trafic réseau est stable et prévisible, la stratégie de réplication statique est la meilleure option. Le nombre et l'emplacement des répliques sont prédéterminés pour atteindre le niveau de performance souhaité. Les coûts associés à la stratégie de réplication statique sont directement proportionnels au nombre des répliques actives. En revanche, avec le mécanisme de réplication dynamique, les répliques sont créées et supprimées dynamiquement par le composant système dédié en réponse aux modifications du mode d'accès utilisateur, de la capacité de stockage et de la consommation de la bande passante .

II.5.2. Réplication centralisée vs. Réplication décentralisée

Les mécanismes statiques et dynamiques peuvent être regroupés en algorithmes de réplication distribués ou centralisés. Les stratégies de réplication centralisée sont gérées par une autorité centrale d'une organisation pour planifier, surveiller et contrôler tous les aspects et processus

de la réplication de données. De plus, tous les autres noeuds lui font rapport. D'autre part, les stratégies de réplication décentralisées sont gérées et exécutées avec la collaboration et la coordination d'un certain nombre de noeuds

II.5.3. Réplication déclenchée par le serveur vs. Par Le client

Une autre façon de classification des stratégies de réplication est l'origine, client ou serveur, du déclenchement de l'événement de réplication. Il peut être déclenché par un client (pull based replication) ou un serveur (push based replication).

II.5.4. Classification basée sur les fonctions objectives

Les auteurs ont proposé une autre classification qui prend en compte la fonction objective de chaque stratégie de réplication. Ces objectifs comprennent :

- Les différentes formes de localité de données, telles que la localité temporelle, géographique et spatiale
- Localité de niveau de réseau
- Modèle de coût,
- Comportement économique.

Les objectifs des stratégies de réplication sont souvent en conflit les uns avec les autres (minimisant les coûts par rapport à la réduction du temps d'accès) et / ou sans rapport (consommation d'énergie, temps de réponse). En raison de la nature de la fonction objective, la plupart des stratégies existantes ne satisfont pas simultanément plusieurs fonctions objectives.

IV.6. Etude comparative :

Les stratégies de réplications présentées dans la section IV.2 peuvent être résumées dans le tableau IV.1. Dans ce dernier, nous présentons des stratégies de réplications dans les Clouds, leurs objectifs, la technique utilisée, si ces stratégies présentent des contrats de garanties de performances à travers des contrats SLA et le type de gestion de répliques utilisé par la stratégie.

N°	Stratégie	Objectif de la stratégie	Technique utilisée	Respect SLA	Gestion de répliques
1	[Yuan et al, 2010] "A data placement strategy in scientific cloud Workflows". *Future Generation Computer Systems*	Disponibilité	Limiter transfert de données entre Data Centres	-	Centralisé
2	CDRM [Wei et al, 2010] A cost-effective dynamic replication management scheme for cloud storage Cluster. In *IEEE International Conference on Cluster Computing*,	Disponibilité	Nœuds moins chargés et plus puissant	-	Centralisé
3	CIR [Li et al, 2011] A novel cost-effective dynamic data replication Strategy for reliability in cloud data centres. In *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*	Disponibilité et réduire le coût de stockage	Réduire le nombre de répliques	-	Centralisé
4	[Pamies-Juarez et al, 2011] Towards the design of optimal data redundancy schemes for heterogeneous cloud Storage infrastructures. *Computer Networks*	Disponibilité et tolérance aux pannes	Niveau minimum de redondance	-	Centralisé
5	[Hussein et Moussa, 2012] "A light-weight data replication for cloud data centers environment". *International Journal of Engineering and Innovative Technology*	Disponibilité	Fichiers populaires	-	Centralisé
6	[Zhang et al, 2014] Data replication placement strategy based on bidding mode for cloud storage cluster	Disponibilité	A base d'enchères	-	Centralisé

7	SKUTE [Bonvin et al,2010] self-organized ,faulttolerant and scalable replication scheme for cloud storage categories and subject descriptors.	Disponibilité et bénéfices économiques	Modèle économique virtuel	-	Auto-organisé
8	[Zeng et al, 2016] Toward cost-effective replica placements in cloud storage systems with QoS-awareness.	Disponibilité	Equilibrage de charge	-	Centralisé
9	[Boru et al, 2015] Models for efficient data replication in cloud computing datacenters. In *2015 IEEE International Conference on Communications (ICC)*,	Disponibilité et l'énergie	A base d'énergie et bande passante	-	Décentralisé
10	RepliC[Sousa et Machado, 2012] Towards elastic multi-tenant database replication with quality of service.	Temps de réponse	Equilibrage de charge	+	Centralisé
11	RTRM [Bai et al, 2013] A response time-based replica management strategy for cloud storage system.	Performances	Popularité des noeuds	+	Centralisé
12	MORM[Long et al, 2014] A multi-objective optimized replication management strategy for cloud storage cluster.	performances	Optimisation multiobjectif	+	Centralisé
13	[Sakr et Liu, 2012] SLA-based and consumer-centric dynamic provisioning for cloud databases.	Temps total d'exécution	Provisionnement dynamique	+	Décentralisé
14	[Sharov et al, 2015] "Take me to your leader!:Online optimization of distributed storage configurations,"	Performances	Réduire les retards du réseau	+	Centralisé
15	AREN[Silvestre et al,2012] A popularity aware replication scheme for cloud storage.	Temps de réponse	Fréquence d'accès	+	Centralisé
16	[Zhao, 2013] *Database Replication on Cloud Platforms*.	Performances	Délai de réplication	+	Centralisé
17	DRSP[Rajalakshmi et al,2014] "An improved dynamic data replica selection and placement in cloud"	Performances et minimisation de bande passante	Catalogue et index des répliques	+	Centralisé
18	SWORD[Kumar et al,2014] workload-aware data placement and replica selection for cloud data management systems.	Energie et temps de réponse	Améliorer la durée de requête	+	Centralisé
19	[Mansouri et al, 2017] Cost optimization for dynamic replication and migration of data in cloud data centers.	Performance et minimisation des coûts du fournisseur	Minimisation du coût des ressources	+	Centralisé
20	[Janpet et Wen, 2013] Reliable and available data replication planning	Performance	Délais d'accès et	+	Centralisé

	for cloud storage.		budget réduit		
21	[Radi, 2016] "Replica placement strategy based on analytic hierarchy process in heterogeneous cloud data storage".	Performances du DC	Optimisation multicritères	-	Centralisé
22	[Mansouri, 2016] Cost optimization for dynamic replication and migration of data in cloud data centers.	Performances	Fonction du coût et somme pondérée	-	Centralisé

IV.1 Tableau comparatif entre les stratégies de réplication

Dans les systèmes traditionnels, de nombreuses stratégies de réplication de données disponibles créent autant de répliques que possible pour atteindre une disponibilité ou des performances du système élevées. Par conséquent, une telle approche de réplication de données agressive, tout en satisfaisant éventuellement les performances, peut également entraîner des coûts d'exploitation plus élevés que souhaités pour le système. La création et la maintenance d'un nombre inutilement élevé de répliques sont donc inévitablement liées à l'impact de la complexité de la stratégie proposée.

Dans les systèmes de gestion de données à grande échelle où les données sont distribuées géographiquement, c'est-à-dire à l'échelle des réseaux étendus [Goel et Buyya, 2006], un accès fréquent aux données peut saturer les liaisons réseaux, surcharger les sources de stockages éloignés, et dégrader globalement les performances de calcul. D'autre part, placer des copies locales d'ensembles de données à chaque nœud est coûteux et tout simplement pas réaliste. Par conséquent, le placement des données joue un rôle important dans tous les systèmes de gestion de données à grande échelle.

Par conséquent, dans les systèmes de Cloud Computing, les principales questions de quoi répliquer, quand répliquer, combien de répliques à créer et où placer ces réplique, doivent être présentées pour satisfaire les exigences du client.

Parmi les travaux cités dans cette section, nous nous somme focalisés sur la stratégie (ADRS) proposée par [Mansouri, 2016] qui utilisait un maximum de critères (le temps de service moyen, la probabilité d'échec, la variance de la charge, la latence et l'utilisation du stockage). Entretemps, sa stratégie a été comparée à quelques travaux cités plus haut (CIR, CDRM, DRSP) et avait des performances améliorées par rapport au temps de réponse moyen, à la variation de charge, à l'utilisation effective du réseau, à la fréquence de réplication et au stockage utilisé. Elle a calculé une fonction de coût, pour chaque site, en fonction du temps de service, de la variance de la charge, de l'utilisation du stockage, de la probabilité de

défaillance et de la latence. La fonction de coût qu'elle a utilisée a été calculée en appliquant la méthode de la somme pondérée où l'utilisateur a proposé des poids pour chaque critère. Néanmoins, cette méthode de la somme pondérée, avait plusieurs limites :

- La logique d'agrégation sous-jacente est totalement compensatoire. Un poids très faible sur un critère peut être compensé par un ou plusieurs poids élevés sur d'autres critères.
- Certains nœuds de données efficaces (non dominés) peuvent ne jamais apparaître comme une solution optimale pour une somme pondérée, quel que soit l'ensemble des poids sélectionnés.
- Haute sensibilité aux petites variations de pondération. De très légères variations de poids peuvent conduire à des solutions radicalement différentes
- Pas de correspondance intuitive entre les valeurs des poids et la solution optimale obtenue par une somme pondérée.

Afin de remédier à ces limites, nous proposons dans le chapitre 4 deux approches adaptées qui modélisent le système comme un problème d'optimisation multicritère, en appliquant deux méthodes d'optimisation plus robustes que la somme pondérée, à savoir AHP (Analytic Hierarchy Process) et ELECTRE-I.

IV.7. Conclusion :

Dans ce chapitre nous avons fait une présentation des travaux de la réplication dans le Cloud.

Dans le Cloud Computing, les stratégies de réplication de données existantes sont orientées vers une plus large sélection de scénarios. De nombreuses stratégies existantes traitent de la disponibilité avec une minorité d'entre elles considérant également la performance. Seule une partie de stratégies de réplication de données proposées pour le Cloud prennent en compte la disponibilité et les performances simultanément.

V.1 Introduction

Pour tester les nouvelles stratégies et techniques développées par les chercheurs, ces derniers ont besoin d'outils leur permettant d'évaluer ces nouvelles stratégies avant un déploiement réel dans un environnement où l'on peut reproduire les tests.

Avec les Clouds, l'accès aux infrastructures nécessite de payer, même s'il s'agit juste de tester les besoins d'une application. Les outils de simulation offrent des avantages significatifs car ils permettent aux développeurs Cloud de tester gratuitement la performance, l'efficacité et la fiabilité de leurs stratégies dans un environnement reproductible et contrôlable. Ils offrent également la possibilité de changer les paramètres afin d'ajuster et optimiser ces stratégies avant un déploiement réel dans le Cloud. Dans ce chapitre nous présentons l'environnement de notre travail, ensuite nous découvrons le simulateur CloudSim, et finalement nous présentons une série de critères d'évaluation.

V.2. Environnements de développement :

1. **NetBeans :** NetBeans est un projet open source ayant un succès et une base d'utilisateur très Large, une communauté en croissance constante, et prés100 partenaires mondiaux et des centaines de milliers d'utilisateur à travers le monde. Sun Microsystems a fondé le projet open source NetBeans en Juin2000 et continue d'être le sponsor principal du projet. [Ins16]

Aujourd'hui, deux projets existent : L'EDI NetBeans et la Plateforme NetBeans. L'EDI NetBeans est un environnement de développement, un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans.

L'EDI NetBeans est un produit gratuit, sans aucune restriction quant a son usage.

Egalement disponible, la Platform netbeans; une fondation modulable et extensible utilisée comme brique logicielle pour la création d'applications bureautiques.

Les partenaires privilégies fournissent des modules a valeurs rajoutées qui s'intègrent facilement à la Plateforme et peuvent être utilisés pour développer ses Propres outils et solutions. Les deux produits sont open source et gratuits pour un usage commercial et non-commercial. Le code source est disponible pour réutilisation sous la Common développent and Distribution License(CDDL).

2. **CloudSim :** CloudSim est un Framework de simulation généralise et extensible qui permet la modélisation, la simulation et l'expérimentation des nouvelles infrastructures de Cloud Computing et des services d'application associes. Nous avons utilisé pour la réalisation de nos travaux de thèse la version du simulateur CloudSim3.0.3.

Le simulateur CloudSim est compose de plusieurs classes(Figure5.1).Par miles classes fondamentales qui forment les blocs constitutifs du simulateur CloudSim, nous pouvons citer.

3. **Datacenter :** La classe Datacenter permet de modéliser le cœur de l'infrastructure du Cloud. elle encapsule un ensemble de machines physiques appelées Hosts qui se caractérisent par leurs configurations (mémoire, CPU, stockage et nombre de curs).

Chaque Datacenter implémente un ensemble d'algorithmes pour l'allocation de la bande passante, la mémoire et le stockage aux différents hosts et machines virtuelles du Cloud.

4. **Cloudlet :** La classe Cloudlet modélisé les applications. Elle a un nombre d'instructions et de données connu à exécuter et à transférer. A noter que cette classe est étendue par Work flow Simen Task puis en Job pour modéliser les relations de dépendances entre les taches.

5. **Datacenter Broker :** Cette classe modélisé le courtier(Broker), qui est responsable de la médiation entre les utilisateurs et les prestataires de service selon les conditions de QoS des utilisateurs et elle permet de déployer les tâches de service à travers les Clouds.

Le Broker agit à un om des utilisateurs, il identité les prestataires de service approprie du Cloud par le service d'information du CloudCIS (Cloud Information Services) et négocie avec eux pour une allocation des ressources qui répond aux besoins de QoS des utilisateurs.

6. **Host :** Cette classe modélise une ressource physique comme le serveur de stockage ou de calcul. Elle encapsule des informations importantes telles que la quantité de mémoire et de stockage, le type de cœurs de traitement (pour représenter une machine (multi-core), une politique d'allocation pour le partage de la puissance du traitement entre les machines virtuelles et les politiques d'attribution de mémoire et de bande passante aux machines virtuelles.

7. **Machine virtuelle :** Cette classe modélisé une instance de la machine virtuelle(VM),

Dont la gestion pendant son cycle de vie, est une responsabilité de la machine(Host).

Un Host peut simultanément instancier de multiples VMS et assigner des cœurs a base des politiques prudences de partage de processeur espace partage ou temps partage.

V.3 Paramètres d'évaluation :

Afin d'évaluer une stratégie de réplication, nous avons utilisé le simulateur CloudSim, nous l'avons étendu afin de gérer la réplication des données et la mesure des coûts des ressources.

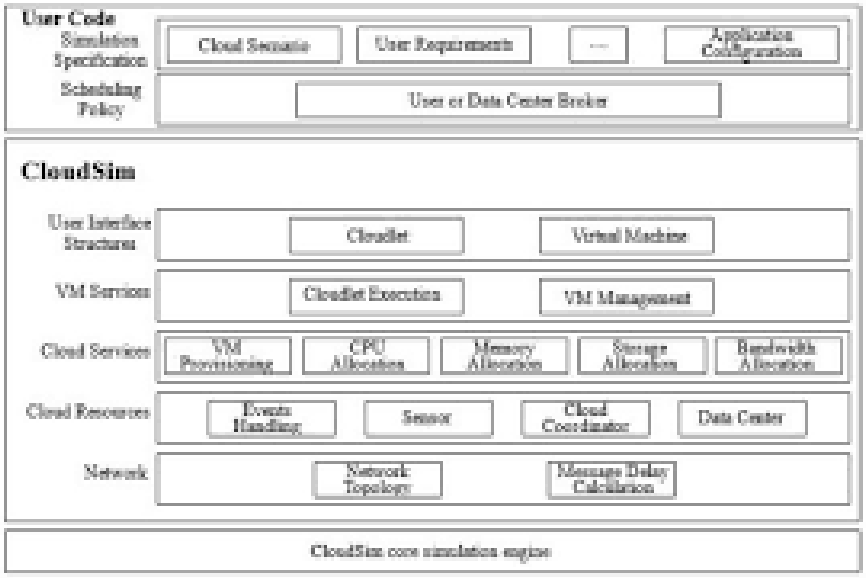

Fig. V.1. Architecture du CloudSim

Également l'impact des requêtes parallèles sur le nombre de répliques. Ensuite, nous évaluons l'impact des requêtes parallèles sur le temps de réponse et la violation du SLA. Pour ces expériences, Enfin, nous mesurons l'impact du nombre de centres de données sur l'équilibrage de charge et les performances du système.

V.3.1 Impact du budget

A. Effets sur le nombre de répliques

Nous mesurons l'impact des requêtes parallèles sur le facteur de répliques. L'impact du budget est également évalué. Nous mesurons le nombre de répliques en variant le nombre de requêtes parallèles.

B. Effets sur la disponibilité

Nous essayons de voir l'effet du budget sur la disponibilité des répliques.

C. Effets sur le temps de réponse

Le but de la simulation suivante est d'étudier l'influence de la valeur du budget sur le temps de réponse moyen ;

V.3.2 Impact du nombre de centres de données

Nous mesurons l'impact du nombre de DCs sur l'équilibrage de charge du système.

A. Effet sur l'équilibrage de la charge du système

L'emplacement des répliques et la planification des requêtes ont un impact significatif sur l'équilibrage de charge. Dans cette expérience, nous comparons le pourcentage de charge sur différents centres de données.

B. Effet sur le temps de réponse

Notre objectif est de voir comment l'augmentation du nombre de DCs peut affecter le temps de réponse moyen des requêtes.

C. Effets sur la consommation de bande passante

Nous mesurons la consommation de bande passante. Moins de transfert de données est nécessaire, ce qui a pour effet de réduire la consommation de bande passante.

V.3.3 Impact du nombre de requêtes

nous évaluons l'impact du nombre de requêtes sur les performances et la violation SLA.

A. Effets sur le temps de réponse

Pour étudier l'impact du nombre de Cloudlets sur le temps de réponse moyen,

B. Effets sur la violation du SLA

Une caractéristique importante qui différencie le Cloud aux modèles commerciaux traditionnels est le mécanisme de pénalité. En cas de violation d'un contrat SLA, le fournisseur est tenu de verser au locataire une somme d'argent convenue.

V.4 Conclusion

Dans ce chapitre nous avons largement étendu un simulateur de Cloud existant, CloudSim, Nous avons choisi plusieurs paramètres comme : le nombre de requêtes, le nombre de centres de données et le budget. L'impact de ces paramètres sur plusieurs métriques comme le temps de réponse, la bande passante, la disponibilité de la donnée, etc.

En raison de contraintes de temps, nous n'avons pas pu atteindre les résultats expérimentaux, basés sur la comparaison de deux stratégies de réplication.

Conclusion générale

A travers ce travail, nous avons étudié le problème de réplication de données dans le Cloud Computing. Plusieurs stratégies de réplication ont été proposées dans la littérature : elles visent principalement à créer et à placer de manière efficace les répliques afin qu'un site puisse trouver les données nécessaires à l'exécution de sa requête. Les approches sur lesquelles se base ces stratégies diffèrent pour pouvoir offrir un compromis entre les objectifs de la réplication, à savoir, la réduction du temps d'accès aux données, l'augmentation du niveau de disponibilité et la tolérance aux pannes.

Le problème de réplication de données dans le Cloud Computing est assez complexe, il présente plusieurs défis, tels que la dynamique des sites, la capacité de stockage et la surcharge de la bande passante. Une stratégie efficace de réplication doit s'adapter aux changements du Cloud et au comportement dynamique des utilisateurs en évitant de nuire aux performances du système en le surchargeant de transfert de données superflu.

D'autre part, le choix de l'architecture du Cloud est un facteur crucial dont dépendent les performances attendues du système. Dans ce contexte nous avons fait une présentation des travaux de la réplication dans le Cloud, et après une étude comparative est menée entre les différentes stratégies de réplication sur plusieurs critères.

Dans le Cloud Computing, les stratégies de réplication de données existantes sont orientées vers une plus large sélection de scénarios. De nombreuses stratégies existantes traitent de la disponibilité avec une minorité d'entre elles considérant également la performance. Seule une partie de stratégies de réplication de données proposées pour le Cloud prennent en compte la disponibilité et les performances simultanément.

Perspectives

Afin d'étendre notre travail, nous planifions de tester d'autres stratégies de réplication multicritères et même étudier la complexité des méthodes d'optimisation multicritères appliquées pour le placement de répliques.

Enfin, la performance de tout protocole est liée à des paramètres relatifs à l'environnement externe comme la mobilité, la connectivité, le nombre des clients participants, le trafic engendré, l'énergie…etc, que seul une simulation avancée peut évaluer. Ceci fera l'objet des prochains travaux dans lequel on va essayer d'analyser le comportement des stratèges de réplication étudies dans un environnement Cloud.

Bibliographie

[1] Slashdot. Multiple Experts Try Defining Cloud Computing. http://tech.slashdot.org/article.pl? consulté le 08/03/21/2117221.

[2] http://aws.amazon.com/s3-sla/ consulté le 24/03/2021.

[3]http://wiki.cloudcommunity.org/wiki/CloudComputing:Incidents_Database. consulté le 22/03/2021

[4] http://en.wikipedia.org/wiki/IBM_DB2. consulté le 24/03/2021

[5] http://www.ibm.com/developerworks/db2/library/techarticle/dm-0608mcinerney/ index.html. consulté le 12/03/2021

[6] http://www.oracle.com/solutions/business_intelligence/exadata.html consulté le 07/03/2021

[7] S. Madden, D. DeWitt, and M. Stonebraker. Database parallelism choices greatly impact scalability. DatabaseColumn Blog.

[8] S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News, 33(2):51–59, 2002.

[9] Amazon Web Services. SimpleDB. Web Page. http://aws.amazon.com/simpledb/. consulté le 02/04/2021

[10] B. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. Pnuts: Yahoo!s hosted data serving platform. In Proceedings of VLDB, 2008.

[11] Amazon Web Services. SimpleDB. Web Page. http://aws.amazon.com/simpledb/.

[12] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: a distributed storage system for structured data. In Proceedings of OSDI, 2006.

[13] M. Stonebraker, S. R. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland. The end of an architectural era (it's time for a complete rewrite). In VLDB, Vienna, Austria, 2007.

[14] C. Monash. Oracle Announces an Amazon Cloud Offering. DBMS2 Blog. http://www.dbms2.com/2008/ 09/22/oracle-announces-an-amazon-cloud-offering/. consulté le 06/04/2021

[15] D. Vesset. Worldwide data warehousing tools 2005 vendor shares. Technical Report 203229, IDC, August 2006.

[16] C. Monash. Oracle Announces an Amazon Cloud Offering. DBMS2 Blog. http://www.dbms2.com/2008/ 09/22/oracle-announces-an-amazon-cloud-offering/. consulté le 06/04/2021

[17] S. Madden, D. DeWitt, and M. Stonebraker. Database parallelism choices greatly impact scalability. DatabaseColumn Blog. http://www.databasecolumn.com/2007/10/database-parallelism-choices.html.

[18] Nicolas Degroodt, L'élasticité des bases de données sur le Cloud Computing. Mémoire de Master en Sciences Informatiques. Bruxelles : Académie UniversitaireWallonie Bruxelles, 2011.

[19] http//www.zdnet.fr/actualites/sante-quel-cloud-computing-pour-ce-secteursensible-39822904.htm. consulté le 22/03/2021

[20]Ait Ali Sylia, Lahdir Meriem, Gestion de la cohérence des données répliquées dans le Cloud. Mémoire de Master en Informatique. Option : Réseaux et systèmes distribués : Université A/Mira de Bejaia, 2016.

[21] M. Melo, P. Maciel, J. Araujo, R. Matos, C. Araujo : Availability Study onCloud computing Environments : Live Migration as a Rejuvenationn Mechanism,Dependable Systems and Networks (DSN), 43 e Annual IEEE, 2013.

[22] W. En Dong, W. Nan, L. Xu, QoS-oriented Monitoring Model of Cloud computing Ressources Availability, Computational and Information Sciences (ICCIS) Fifth International Conference, IEEE 2013, 1537-1540, 21-23 juin 2013

[23] Alouache Lilia, Ait Mesbah Sofia, Solution de disponibilité dans les environnements Cloud Computing. Mémoire de Master en Informatique. Option : Réseaux et systèmes distribués : Université A/Mira de Bejaia, 2015.

[24] Nagamani H Shahapure, P Jayarekha, Replication, A Technique for Scalability in Cloud Computing, International Journal of Computer Applications Volume 122 ? No.5, (0975 ? 8887),(Juillet 2015).

[25] Alouache Lilia, Ait Mesbah Sofia, Solution de disponibilité dans les environnements
Cloud Computing. Mémoire de Master en Informatique. Option : Réseaux et systèmes distribués : Université A/Mira de Bejaia, 2015.

[26] Sacha Krakowiak, Gestion répartie de données ? 1 Duplication et cohérence.Grenoble : INRIA et IMAG-LSR. École Doctorale de Grenoble Master 2 Recherche ?Systèmes et Logiciel ?, 2004.
[27] https//fr.wikipedia.org/wiki/Coh%C3%A9rence(donn%C3%A9es)#Coh.C3.A9rence_stricte
[28] G. Belalem et F. Bouharaous. «Stratégie de placement dynamique des réplique dans les grilles de données. In Setif 2007,4th International conférence : science et électronique, technologie et information and T-communications ; TUNISIA, MARS ».In:(2007)
[29] Alami Milani B., and Jafari Navimipour .N. A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions.Journal of Network and Computer Applications, 64:229–238, 2016. ISSN 10958592.doi: 10.1016/j.jnca.2016.02.005.
[30] Vulimiri. A, Carlo Curino, Brighten Godfrey, Thomas Jungblut, Jitu Padhye,and George Varghese. Global analytics in the face of bandwidth and regulatory constraints. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 323–336, 2015. ISBN 9781931971218.
[31] Sousa. F. R C and Javam C. Machado. Towards elastic multi-tenant database replication with quality of service. In *IEEE/ACM 5th International Conference on Utility and Cloud Computing (UCC 2012)*, pages 168–175, 2012. ISBN 9780769548623. doi: 10.1109/UCC.2012.36.
[32]Silvestre. G, Sebastien Monnet, Ruby Krishnaswamy, and Pierre Sens.AREN: A popularity aware replication scheme for cloud storage. In *IEEE 18thInternational Conference on Parallel and Distributed Systems*, pages 189–196. IEEE,dec 2012. ISBN 978-1-4673-4565-1. doi: 10.1109/ICPADS.2012.35.
[33] Kouki. Y, Thomas Ledoux, and Remi Sharrock. Cross-layer SLA selection for cloud services. In *First International Symposium on Network Cloud Computing and Applications*, pages 143–147. IEEE, nov 2011. ISBN 978-1-4577-1667-6. doi: 10.1109/NCCA.2011.30.
[34] Wei. Q, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng. CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster. In *IEEE International Conference on Cluster Computing*, pages 188–196.IEEE, sep 2010. ISBN 978-1-4244-8373-0. doi: 10.1109/CLUSTER.2010.24.
[35] Bonvin. N, Thanasis G Papaioannou, and Karl Aberer. A self-organized, faulttolerant and scalable replication scheme for cloud storage categories and subject descriptors. In *Proceedings of the 1st ACM symposium on Cloud computing - SoCC'10*, pages 205–216, 2010a. ISBN 9781450300360.
[36] Lee. J, Jaehwa Chung, and Daewon Lee. Efficient data replication scheme based on hadoop distributed file system. *International Journal of Software Engineering and Its Applications*, 9(12):177–186, 2015.
[37] Wei. Q, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng. CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster. In *IEEE International Conference on Cluster Computing*, pages 188–196. IEEE, sep 2010. ISBN 978-1-4244-8373-0. doi: 10.1109/CLUSTER.2010.24.
[38] Myint. J and Thinn Thu Naing. Management of data replication for pc cluster based cloud storage system. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, 1(3):31–41, 2011. doi: 10.5121/ijccsa.2011.1303.
[39] Pamies-Juarez. L, Pedro García-López, Marc Sánchez-Artigas, and Blas Herrera. Towards the design of optimal data redundancy schemes for heterogeneous cloud storage infrastructures. *Computer Networks*, 55(5):1100–1113, apr 2011. ISSN13891286. doi: 10.1016/j.comnet.2010.11.004.
[40] Sun. D. W, Gui-Ran Chang, Shang Gao, Li-Zhong Jin, and Xing-Wei Wang. Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. *Journal of Computer Science and Technology*, 27(2): 256–272, mar 2012a. ISSN 1000-9000. doi: 10.1007/s11390-012-1221-4.
[41] Zhang. H, Bing Lin, Zhanghui Liu, and Wenzhong Guo. Data replication placement strategy based on bidding mode for cloud storage cluster. In *11th Web Information System and Application Conference*, pages 207–212, 2014. ISBN 978- 1-4799-5726-2. doi: 10.1109/WISA.2014.45.
[42] Abouzamazem. A and Paul Ezhilchelvan. Efficient inter-cloud replication for high-availability services. In *013 IEEE International Conference on Cloud Engineering (IC2E)*, pages 132–139. IEEE, mar 2013. ISBN 978-0-7695-4945-3. doi:10.1109/IC2E.2013.27.
[43] Miglierina. M, Giovanni P. Gibilisco, Danilo Ardagna, and Elisabetta Di Nitto.Model based control for multi-cloud applications. In *ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, pages 37–43, 2013. ISBN9781467364478. doi: 10.1109/MiSE.2013.6595294.

[44] Li. W, Yun Yang, and Dong Yuan. A novel cost-effective dynamic data replication strategy for reliability in cloud data centres. In *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, pages 496–502.
IEEE, dec 2011. ISBN 978-1-4673-0006-3. doi: 10.1109/DASC.2011.95.
[45] Bonvin. N, Thanasis G Papaioannou, and Karl Aberer. A self-organized, faulttolerant and scalable replication scheme for cloud storage categories and subject descriptors. In *Proceedings of the 1st ACM symposium on Cloud computing – SoCC'10*, pages 205–216, 2010a. ISBN 9781450300360.
[46] Al Ridhawi. I, Nour Mostafa, and Wassim Masri. Location-aware data replication in cloud computing systems. *2015 IEEE 11th International Conference on Wireless andMobile Computing, Networking and Communications (WiMob)*, pages 20–27, 2015.doi: 10.1109/WiMOB.2015.7347936.
[47] Qu. Y and Naixue Xiong. RFH: A resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage. In *2012 41st International Conference on Parallel Processing*, pages 520–529. IEEE, sep 2012. ISBN 978-1-4673-2508-0. doi: 10.1109/ICPP.2012.3.
[48] Mengxing. H, Ye Xianglong, Wei Sanpeng, and Zhu Donghai. A strategy of dynamic replica creation in cloud storage. In *Proceedings of the 1st International Workshop on Cloud Computing and Information Security*, number Ccis, pages 389–392, Paris, France, 2013. Atlantis Press. ISBN 978-90-78677-88-8. doi: 10.
2991/ccis-13.2013.89.
[49] Lee. J, Jaehwa Chung, and Daewon Lee. Efficient data replication scheme based on hadoop distributed file system. *International Journal of Software Engineering and Its Applications*, 9(12):177–186, 2015.