

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Saad Dahleb Blida 1

Faculté des Sciences



Département d'Informatique

Mémoire de fin d'étude pour l'obtention du diplôme de Master en  
Informatique

Option: Systèmes Informatiques et Réseaux

Présenté par:

M<sup>elle</sup> TETBIRT Achouak

---

**Thème:**

---

**Tolérance aux pannes réactive dans  
les systèmes distribués de type  
Cloud**

Encadré par :

- Mme BOUTOUMI Bachira (USDB)
- M. KHIAT Abdelhamid (CERIST)

Devant le jury composé de:

- M. BENYAHIA Mohamed.
- Mlle MANCER Yasmine.

Année universitaire: 2019-2020

# Dédicace

C'est avec profonde gratitude et sincères mots, Que je dédie ce modeste travail de fin d'étude:

À mes chers parents, qui ont sacrifié leur vie pour ma réussite et m'ont éclairé le chemin par leurs conseils judicieux, c'est grâce à vous et pour vous que j'ai fait mon mémoire. Aucun mot sur cette page ne saurait exprimer ce que je vous dois, ni combien je vous aime. J'espère qu'un jour, je pourrais leurs rendre un peu de ce qu'ils ont fait pour moi, que dieu leur prête bonheur et longue vie.

À mes frères, à mes sœurs, pour avoir contribué à la réussite de ce travail d'une manière indirecte, et pour leurs encouragements.

Merci à tous.



# Remerciement

**J**e tiens à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

Je commence d'abord à remercier mon encadreur Monsieur Khiat Abdelhamid pour son soutien, son aide précieux, et disponibilité durant toute cette période d'encadrement, pour la qualité de ses conseils et pour les nombreuses discussions que j'avais eues.

J'exprime ma profonde gratitude à Madame Boutoumi Bachira pour avoir accepté de m'encadrer, pour ses conseils précieux tout au long de la réalisation de ce mémoire.

Mes vifs remerciements aux membres du jury pour l'intérêt accordé à mon travail en l'examinant minutieusement et avec attention.

Enfin, je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail, principalement à tous l'effectif du Centre de Recherche sur l'Information Scientifique et Technique (CERIST).

Dans un système distribué de type Cloud, un nombre important de ressources de stockage sont utilisées, sur laquelle un important volume de données sous forme de fichiers sont hébergées, ce qui augmente la probabilité d'avoir une panne d'une ou de plusieurs ressources de stockage, d'où la nécessité de mettre en service une politique de tolérance aux pannes réactive, qui permet de récupérer les données perdues après la panne.

L'utilisation de réplication de données est une des politiques de gestion de pannes réactives les plus répandues dans un environnement Cloud, mais l'inconvénient majeur de cette politique réside au niveau de l'espace de stockage dédié aux sauvegardes qui devient rapidement trop volumineux, ce qui oblige la communauté des chercheurs et des développeurs à prendre en compte l'optimisation des espaces de stockage dans une telle solution de tolérance aux pannes réactive.

Ce travail consiste à concevoir une solution de gestion des pannes réactive dans un réseau distribué de type Cloud, tout en optimisant l'espace de stockage dédié aux sauvegardes de données.

**Mots-clés:** Systèmes Distribués, Cloud Computing, Virtualisation, Tolérance aux pannes, Réactive, Réseau redondant des disques indépendants (RAID).

In a distributed system like Cloud, when a large number of storage resources are used, a large volume of data in the form of files are hosted. Then, increases the probability of having a failure of one or more storage resources, hence the need to use a reactive fault tolerance policy, which allows recovering data after such a failure.

The use of data replication is one of the most used fault management policies in a Cloud environment, but the major drawback of this policy lies in the storage space dedicated to backups that quickly becomes too large, which leads the researcher community and developers to take into account the optimization of storage space in such a reactive fault tolerance solution.

This work consists of designing a reactive fault management solution in a distributed Cloud-based architecture, while optimizing the storage space dedicated to data backups.

**Keywords:** Distributed Systems, Cloud Computing, Virtualisation, Fault Tolerance, Reactive, Redundant Array of Independent Disks (RAID).

في الأنظمة الموزعة مثل السحابة، يتم استخدام عدد كبير من موارد التخزين، حيث يتم استضافة حجم كبير من البيانات في شكل ملفات، مما يزيد من احتمال حدوث خلل في واحد أو أكثر من موارد التخزين، وبالتالي الحاجة إلى وضع سياسة التسامح مع الأخطاء التفاعلي (أو ما يعرف كذلك بالتسامح مع الأخطاء برد الفعل)، والتي تجعل من الممكن استعادة البيانات المفقودة بعد الخلل.

يعد استخدام النسخ المتماثل للبيانات أحد أكثر سياسات إدارة الأخطاء استجابة في بيئة الحوسبة السحابية، ولكن العيب الرئيسي لهذه السياسة يكمن في مساحة التخزين المخصصة للنسخ الاحتياطية التي تصبح كبيرة جدًا بسرعة، مما أدى الباحثين والمطورين على مراعاة تحسين مساحة التخزين في مثل هذا الحل التفاعلي.

يتكون هذا العمل من تصميم حل إدارة خطأ تفاعلي في شبكة سحابية موزعة، مع تحسين مساحة التخزين المخصصة لنسخ البيانات الاحتياطية.

**الكلمات الرئيسية:** الأنظمة الموزعة، الحوسبة السحابية، المحاكاة الافتراضية، التسامح مع الأخطاء، رد الفعل، شبكة زائدة من الأقراص المستقلة (RAID).

Introduction générale .....	1
1. Introduction.....	1
2. Problématiques .....	1
3. Contribution .....	2
4. Organisation du mémoire .....	2
Partie 1: Etat de l’art.....	3
Chapitre 1 Cloud Computing.....	4
1.1 Introduction .....	5
1.2 Partie 1: Généralités sur les systèmes distribués .....	5
1.3 Exemples des systèmes distribués .....	6
1.3.1 Calcul Pair à Pair (Peer to Peer Computing).....	6
1.3.2 Les grappes de calcul (Clusters Computing).....	6
1.3.3 Les grilles de calcul (Grid Computing).....	6
1.3.4 Les calculs à haute performance (High-performance Computing).....	7
1.3.5 Internet des objets (IoT Computing).....	7
1.3.6 Le calcul bénévole (Volunteer Computing) .....	8
1.3.7 Cloud, Edge et Fog Computing .....	8
1.3.8 L’informatique de la jungle (Jungle Computing).....	9
1.4 Partie2 : Le Cloud Computing.....	10
1.4.1 Historique .....	10
1.4.2 Définition .....	10
1.4.3 Les cinq caractéristiques fondamentales du Cloud Computing .....	11
1.4.4 Eléments constitutifs du Cloud Computing .....	11
1.4.5 Techniques de la virtualisation.....	13
1.4.6 Types de la virtualisation .....	14
1.4.7 Les modèles de services .....	16
1.4.8 Les modèles de déploiement .....	18

1.4.9	Avantages et inconvénients du Cloud Computing.....	19
1.5	Conclusion.....	20
<b>Chapitre 2</b>	<b>Tolérance aux pannes.....</b>	<b>21</b>
2.1	Introduction .....	22
2.2	Notion de sûreté de fonctionnement .....	22
2.2.1	Attributs de la sûreté de fonctionnement .....	23
2.2.2	Entraves à la sûreté de fonctionnement.....	23
2.2.3	Moyens de la sûreté de fonctionnement.....	24
2.3	Classification des pannes .....	24
2.3.1	Panne selon la gravité des défaillances .....	25
2.3.2	Panne selon le degré de la permanence.....	26
2.4	Tolérance aux pannes.....	26
2.4.1	Les politiques de tolérances aux pannes .....	26
2.5	Techniques utilisées pour assurer la tolérance aux pannes .....	27
2.5.1	Réplication.....	28
2.5.2	Resoumission des taches (Task Resubmission) .....	28
2.5.3	Migration.....	28
2.5.4	Check-Point / Redémarrage.....	28
2.5.5	RAID.....	28
2.6	Travaux connexes sur la tolérance aux pannes dans le Cloud.....	29
2.7	Conclusion.....	30
<b>Chapitre 3</b>	<b>RAID « Redundant Array of Independent Disks» .....</b>	<b>31</b>
3.1	Introduction .....	32
3.2	Définition et historique .....	32
3.3	Types de RAID .....	32
3.3.1	Le RAID logiciel .....	32
3.3.2	Le RAID matériel .....	33
3.4	Les concepts du RAID .....	34

- 3.4.1 Stripage (Striping) ..... 34
- 3.4.2 Miroir (Mirroring)..... 34
- 3.4.3 Parité (Parity)..... 34
- 3.5 Les niveaux de RAID..... 34
  - 3.5.1 RAID0 (Striping)..... 34
  - 3.5.2 RAID1 (Mirroring) ..... 35
  - 3.5.3 RAID5 (Striping with distributed parity) ..... 36
  - 3.5.4 RAID6 (Striping with dual parity)..... 37
  - 3.5.5 Le RAID 1+0 (RAID10) ..... 38
  - 3.5.6 Le RAID 5+0 (RAID50) ..... 38
  - 3.5.7 Le RAID 6+0 (RAID60) ..... 39
- 3.6 Avantages de RAID ..... 40
  - 3.6.1 Tolérance aux pannes..... 40
  - 3.6.2 Performance..... 40
  - 3.6.3 Capacité..... 40
- 3.7 Conclusion..... 40

**Partie 2: Conception, implémentation et simulation de la solution proposée**  
 .....41

- Chapitre 4 Conception de l’approche proposée ..... 42**
  - 4.1 Introduction ..... 43
  - 4.2 Description et architecture de l’approche proposée..... 43
    - 4.2.1 Description ..... 43
    - 4.2.2 Architecture de système ..... 43
    - 4.2.3 L’utilisation de RAID5 et la reconstruction des données ..... 44
  - 4.3 Processus de l’approche proposée et les algorithmes utilisés ..... 45
    - 4.3.1 Processus de fonctionnement ..... 45
    - 4.3.2 Description des algorithmes utilisés ..... 47

4.4	Conception des diagrammes .....	49
4.4.1	Diagramme cas d'utilisation globale .....	50
4.4.2	Diagramme de séquence .....	51
4.4.3	Diagramme de classes .....	57
4.5	Conclusion.....	58
<b>Chapitre 5 Implémentation et Simulation .....</b>		<b>59</b>
5.1	Introduction .....	60
5.2	Contexte de l'implémentation .....	60
5.3	L'environnement de travail .....	60
5.3.1	Le système d'exploitation. ....	60
5.3.2	Caractéristiques de l'ordinateur.....	60
5.4	Architecture, langages et environnements de développement .....	61
5.4.1	Le design pattern MVC (Model, Vue, Contrôle) .....	61
5.4.2	Java entreprise édition (Java EE).....	61
5.4.3	Le serveur Apache Tomcat.....	62
5.4.4	Eclipse .....	62
5.4.5	Xampp.....	62
5.4.6	MySQL.....	63
5.4.7	HTML5 .....	63
5.4.8	CSS3.....	63
5.4.9	JavaScript.....	63
5.4.10	Bootstrap.....	63
5.4.11	jQuery.....	64
5.5	Présentation de l'application .....	64
5.6	Tests et Evaluations .....	77
5.6.1	Evaluation par implémentation.....	78
5.6.2	Evaluation par simulation.....	80
5.6.3	Discussion .....	81

## TABLE DES MATIERES

5.7 Conclusion.....	82
Conclusion générale et perspectives .....	83
1. Conclusion générale.....	83
2. Perspectives .....	84
Bibliographie	
Webographie	

**Table des illustrations :**

Figure 1 Peer to Peer Computing.....	6
Figure 2 Internet of Things.....	7
Figure 3 Cloud, Fog et Edge Computing .....	8
Figure 4 Jungle Computing .....	9
Figure 5 Historique du Cloud Computing .....	10
Figure 6 Le Cloud Computing .....	10
Figure 7 Data Center .....	12
Figure 8 Architecture de la virtualisation .....	12
Figure 9 Les modèles de services .....	16
Figure 10 Les modèles de déploiement. ....	18
Figure 11 Cloud hybride .....	19
Figure 12 L'arbre de la sûreté de fonctionnement .....	22
Figure 13 La chaîne fondamentale des entraves à la sûreté de fonctionnement.....	23
Figure 14 Classification générale des pannes .....	25
Figure 15 Panne byzantine .....	26
Figure 16 La ligne de temps de détection des pannes proactives .....	27
Figure 17 La ligne de temps de détection des pannes réactives .....	27
Figure 18 RAID logiciel .....	32
Figure 19 RAID matériel.....	33
Figure 20 RAID0 .....	35
Figure 21 RAID1 .....	35
Figure 22 RAID 5 .....	36
Figure 23 RAID6 .....	37
Figure 25 RAID50 .....	38
Figure 24 RAID10 .....	38
Figure 26 RAID 60 .....	39
Figure 27 Architecture du système .....	43
Figure 28 RAID5 et le principe de parité.....	44
Figure 29 Pseudo algorithme "Ajout de ressources" .....	48
Figure 30 Pseudo algorithme "Récupération de ressources".....	49
Figure 31 Diagramme cas d'utilisation global. ....	50
Figure 32 Diagramme de séquence "Ajouter MP" .....	52
Figure 33 Diagramme de séquence "Supprimer MP" .....	52

Figure 34 Diagramme de séquence " Récupérer MP" .....	53
Figure 35 Diagramme de séquence " Ajouter MV" .....	54
Figure 36 Diagramme de séquence " Supprimer MV" .....	55
Figure 37 Diagramme de séquence " Récupérer MV" .....	56
Figure 38 Diagramme de classes .....	57
Figure 39 Modèle MVC .....	61
Figure 40 Interface "Accueil" .....	64
Figure 41 Interface "A propos" .....	65
Figure 42 Interface "Inscription" .....	65
Figure 43 Message reçu après l'inscription .....	66
Figure 44 Informations des utilisateurs enregistrées sur la BD .....	66
Figure 45 Interface "Authentification" .....	66
Figure 47 Menu "Admin" .....	67
Figure 46 Menu "Client" .....	67
Figure 48 Interface "Status de Cloud" pour l'admin .....	68
Figure 49 Interface "Status de Cloud" pour le client .....	68
Figure 50 Ajout Client par l'admin.....	69
Figure 51 Interface "Gestion des Comptes Clients" .....	69
Figure 52 Fenêtre de confirmation d'initialisation de Cloud.....	70
Figure 53 Fenêtre Ajouter MPs .....	71
Figure 54 Interface "Gestion de Cloud" après l'ajout de MPs.....	71
Figure 55 Interface "Gestion de Cloud" après la suppression de MPs .....	72
Figure 56 Récupération réussite de MPs.....	73
Figure 57 Ajout de MV pour le client "sir2020" par l'admin .....	74
Figure 58 MV de client a été ajoutée .....	74
Figure 59 Interface « Gestion MVs » avec un ajout de MV par le client "achouak" .....	75
Figure 60 MV de client "achouak" a été ajoutée .....	75
Figure 61 Interface « Gestion MVs » après une suppression de MVs par l'admin .....	76
Figure 62 Interface « Gestion MVs » après une suppression de MVs par le client "achouak".....	76
Figure 63 Récupération réussite de MVpar le client "achouak".....	77
Figure 64 Comparaison CRAID5 vs Réplication "Cas1" .....	79
Figure 65 Comparaison CRAID5 vs Réplication "Cas2" .....	79
Figure 66 Comparaison CRAID5 vs Réplication "Cas3" .....	80
Figure 67 Comparaison CRAID5 vs Réplication par simulation .....	81

**Table des tableaux:**

Tableau 1 Les leaders mondiaux du Cloud Computing .....	17
Tableau 2 Avantages et Inconvénients des services Cloud.....	17
Tableau 3 Caractéristiques de certains niveaux de RAID.....	39
Tableau 4 Tableau de vérité de l'opération XOR.....	45
Tableau 5 Description des cas d'utilisation .....	51
Tableau 6 Tableau descriptif des classes.....	58
Tableau 7 Caractéristiques de l'ordinateur .....	60
Tableau 8 Comparaison CRAID5 vs Réplication "Cas1" .....	78
Tableau 9 Comparaison CRAID5 vs Réplication "Cas2" .....	79
Tableau 10 Comparaison CRAID5 vs Réplication "Cas3" .....	80
Tableau 11 Paramètres de comparaison CRAID5 vs Réplication par simulation .....	81
Tableau 12 Résultat obtenus de comparaison CRAID5 vs Réplication par simulation .....	81

## LISTE DES ABREVIATIONS

API: **A**pplication **P**rogramming **I**nterface.

BD: **B**ase de **D**onnées.

BTRFS: **B**-tree **F**ile **S**ystem.

CRM: **C**ustomer **R**elationship **M**anagement.

CSS: **C**ascading **S**tyle **S**heets.

CPU: **C**entral **P**rocessing **U**nit.

FTM: **F**ault **T**olerance **M**anager.

FTMC: **F**ault **T**olerance **M**odel for **C**loud.

FTP: **F**ile **T**ransfer **P**rotocol.

HA: **H**igh **A**vailability.

HPC: **H**igh **P**erformance **C**omputing.

HTML: **H**yper**T**ext **M**arkup **L**anguage.

IaaS: **I**nfrastucture as a **S**ervice.

IBM: **I**nternational **B**usiness **M**achines Corporation.

IET: **i**SCSI **E**ntreprise **T**arget

IoT: **I**nternet of **T**hings.

iSCSI: **I**nternet **S**mall **C**omputer **S**ystem **I**nterface.

IT: **I**nformation **T**echnology.

JDBC: **J**ava **D**ata**B**ase **C**onnectivity.

JSP: **J**ava **S**erver **P**ages.

LAN: **L**ocal **A**rea **N**etwork.

MVC: **M**odèle **V**ue **C**ontrôle**u**r.

NAS: **N**etwork **A**ttached **S**torage.

## LISTE DES ABREVIATIONS

NIST: **N**ational **I**nstitute of **S**tandards and **T**echnology.

NFS: **N**etwork **F**ile **S**ystem.

QoS: **Q**uality of **S**ervice.

RAID: **R**edundant **A**rray of **I**ndependent / **R**edundant **A**rray of **I**nexpensive **D**isks.

SaaS: **S**oftware **a**s **a** **S**ervice.

SAN: **S**torage **A**rea **N**etwork.

SDF: **S**ûreté **D**e **F**onctionnement.

SDK: **S**oftware **D**evelopment **K**it.

SE: **S**ous **E**nsemble.

SGBD: **S**ystème de **G**estion de **B**ase de **D**onnées.

P2P: **P**eer to **P**eer.

PaaS: **P**latform **a**s **a** **S**ervice.

PC: **P**ersonal **C**omputer.

PM/MP: **M**achine **P**hysique.

UML: **U**nified **M**odeling **L**anguage.

VM/MV: **M**achine **V**irtuelle.

VPN: **V**irtual **P**rivate **N**etwork.

WAN: **W**ide **A**rea **N**etwork.

XML: **e**Xtended **M**arkup **L**anguage.

ZFS: **Z**ettabyte **F**ile **S**ystem.

## 1. Introduction

Le Cloud Computing (L'informatique dans les nuages, en Français) c'est un paradigme majeur d'utilisation des infrastructures informatiques, il correspond à l'accès à des services informatiques via Internet à partir d'un fournisseur. Il se caractérise par sa souplesse qui permet aux fournisseurs d'adapter automatiquement la capacité de stockage, la haute disponibilité (HA), la qualité de service (QoS), et la puissance de calcul ainsi que plusieurs autres avantages, qui répondent aux besoins et aux demandes croissantes des utilisateurs.

Compte tenu des avantages offerts par cette technologie, un nombre important de services ont été migrés vers l'environnement Cloud, ce qui implique un besoin énorme en termes de ressources utilisées, y compris l'espace de stockage. Avec cette augmentation de la quantité de ressources utilisées dans l'environnement Cloud, la probabilité d'obtenir une panne de ressources telles que les machines physiques (MPs) et les machines virtuelles (MVs) augmente également, provoquant éventuellement une interruption de service. La récupération de ces ressources en panne peut être obtenue en utilisant certaines solutions de gestion des tolérances aux pannes.

L'un des paramètres les plus importants à prendre en compte dans toute solution de tolérance aux pannes est le stockage d'espace utilisé, qui devrait être minimisé autant que possible. Une telle minimisation peut également aider à optimiser d'autres paramètres importants, tel que le coût et l'énergie consommée.

## 2. Problématiques

La gestion de tolérance aux pannes dans l'environnement Cloud Computing est indispensable, surtout pour l'assurance de continuité de services demandés par les clients. Dans un environnement Cloud, d'une part, on trouve le client qui cherche toujours à ne pas perdre ses données, d'une autre part, le propriétaire de Cloud qui est le fournisseur de services, vise à optimiser l'espace de stockage dans le Cloud. Malheureusement, la panne de ressources (MPs et MVs) dans le Cloud est toujours probable et inévitable, ce qui motive la communauté de chercheurs et de développeurs à déployer plus d'effort pour proposer de bonnes solutions afin de répondre à ce problème.

Pour cela, nous nous proposons dans ce mémoire d'aborder la problématique relative à la gestion de tolérance aux pannes réactive dans le Cloud qui permet la récupération des MPs et MVs perdues en optimisant l'espace de stockage utilisé.

### 3. Contribution

Dans ce travail, nous proposons CRAID5, une politique réactive de gestion de tolérance aux pannes orientée vers les architectures distribués de type Cloud. CRAID5 est basée sur un concept puissant connu sous le nom de Redundant Array of Independent Disks (RAID). Ce dernier est largement utilisé par la plupart des systèmes d'exploitation libres, et qui peut être fournis comme des solutions logicielles ou matérielles. CRAID5 permet de récupérer les ressources après une telle panne. La contribution majeure de la solution proposée réside dans l'optimisation de l'espace de stockage à l'aide d'un niveau spécifique de RAID, (RAID5).

### 4. Organisation du mémoire

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

- **Le chapitre 1:** présente un état de l'art sur les concepts du Cloud Computing ainsi que certaines notions générales sur quelques systèmes distribués;
- **Le chapitre 2:** sera consacré à la notion de tolérance aux pannes, ses techniques ainsi aux types des pannes et quelques travaux connexes sur la tolérance aux pannes dans l'environnement Cloud;
- **Le chapitre 3:** décrit les principes de la technologie RAID ainsi que ses différents niveaux;
- **Le chapitre 4:** sera réservé à la description détaillée de la conception de l'approche gestion de tolérance aux pannes réactive proposée. Cette conception va être faite à l'aide d'algorithmes et de diagrammes UML;
- **Le chapitre 5:** ce dernier chapitre présentera les étapes de l'implémentation de l'approche proposée. Nous y détaillerons la réalisation de certaines fonctionnalités ainsi que l'étude d'évaluation de cette solution avec une comparaison avec une autre technique très connue et très utilisée dans le domaine de tolérance aux pannes dans l'environnement Cloud.

Enfin pour terminer ce manuscrit, nous présentons une conclusion pour synthétiser notre travail ainsi qu'un ensemble de perspectives pour des travaux futurs.

**PARTIE 1 :**

**ETAT DE L'ART**

# Chapitre 1 :



## 1.1 Introduction

Cloud, Clusters et Grilles de calculs sont considérées comme de bons exemples des systèmes distribués, qui sont de plus en plus utilisés dans plusieurs domaines. Le Cloud est un des types de ces systèmes qu'a connu un grand succès de nos jours non seulement dans les environnements académiques et industriels mais aussi par la concurrence entre les grands propriétaires du Cloud dans le monde. L'utilisation potentielle des ressources augmente de plus en plus. C'est un obstacle pour l'évolution des entreprises, organismes...etc. Pour remédier à ce problème, les entreprises se sont orientées vers des fournisseurs possédant des infrastructures énormes reposant sur des technologies de virtualisation et d'automatisation. Ce qui permet l'apparition de la notion du Cloud Computing.

Vu que notre travail sera réalisé dans un environnement distribué de type Cloud, dans ce chapitre nous présenterons d'abord une vue générale sur les systèmes distribués après nous allons focaliser sur le Cloud Computing, ses caractéristiques, ses modèles ainsi que ses avantages et inconvénients.

## 1.2 Partie 1: Généralités sur les systèmes distribués

La puissance de calcul et l'espace mémoire fournie par un seul ordinateur n'ont jamais été suffisants par rapport à la quantité de calculs que nous avons à effectuer au quotidien. Il a été proposé de relier les ordinateurs entre eux, afin de créer de «grands» ordinateurs «super ordinateurs», beaucoup plus puissant et permettant de fournir des ressources à un plus grand nombre d'utilisateurs [1], d'où vient l'idée des systèmes distribués, qui sont un ensemble d'ordinateurs indépendants qui apparaît comme un système unique et cohérent dont les utilisateurs ont l'impression d'utiliser un seul système. Une raison pour que les systèmes distribués sont devenus la tendance, est de connecter des machines et de les faire se coordonner afin d'augmenter les performances en temps d'exécution et le calcul réparti. Quand on effectue de lourdes opérations, un moyen d'accélérer l'exécution est d'investir dans une machine plus puissante, Il nous faut alors une disposition de plusieurs unités de calcul, que ce soit les cœurs d'un processeur, plusieurs processeurs sur une seule machine ou plusieurs machines en réseau. On parle de système distribué ou système réparti [2]. Nous distinguons plusieurs autre raisons ou intérêts des systèmes distribués qui sont les suivants [3] :

- **Partage des ressources (données, programme, services):** qui permet un travail collaboratif ;
- **Accès distant:** c'est-à-dire qu'un même service peut être utilisé par plusieurs acteurs situés à des endroits différents ;
- **Amélioration des performances:** la mise en commun de plusieurs unités de calcul permet d'effectuer des calculs parallélisables en des temps plus courts ;
- **Confidentialité:** les données brutes ne sont pas disponibles partout au même moment, seules certaines vues sont exportées ;

- **Disponibilité:** des données en raison de l'existence de plusieurs copies ;
- **BD unique:** maintien d'une vision unique de la base de données malgré la distribution ;

Un tel système est soumis à des problématiques de synchronisation et cohérence des données, de disponibilité, de sécurité...etc. [2]

### 1.3 Exemples des systèmes distribués

Vu que notre travail sera réalisé sur l'environnement Cloud, qu'est considéré comme un cas particulier des systèmes distribués, nous allons décrire en bref quelques exemples des systèmes distribués, tout en donnant plus de détails sur le Cloud Computing par la suite de ce chapitre.

#### 1.3.1 Calcul Pair à Pair (Peer to Peer Computing)

Le pair-à-pair est un modèle de réseau informatique proche du modèle client-serveur où chaque ordinateur connecté au réseau est susceptible de jouer tour à tour le rôle de client et celui de serveur.

P2P est une architecture pouvant être centralisée (les connexions passant par un serveur central intermédiaire) ou décentralisée (les connexions se faisant directement). Le pair-à-pair peut servir au partage de fichiers, au calcul distribué ou à la communication entre nœuds ayant la même responsabilité dans le système. On appelle souvent « nœuds », les postes connectés par un réseau pair-à-pair. [3]

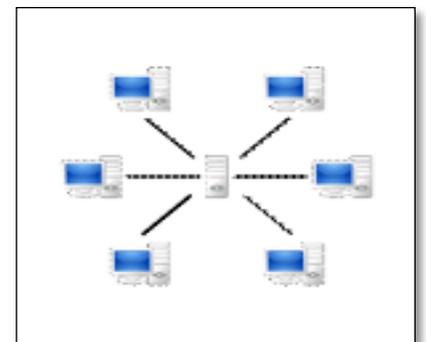


Figure 1 Peer to Peer Computing [3]

#### 1.3.2 Les grappes de calcul (Clusters Computing)

Au sein d'un système informatique, un cluster de serveurs est un groupe de serveurs et d'autres ressources indépendantes fonctionnant comme un seul système. Les serveurs sont généralement situés à proximité les uns des autres, et sont interconnectés par un réseau dédié. Un client dialogue avec le groupe de serveurs comme s'il s'agissait d'une seule machine. [4]

#### 1.3.3 Les grilles de calcul (Grid Computing)

Le Grid Computing, ou Grille informatique en français, est une infrastructure virtuelle constituée d'un ensemble de ressources informatiques. Ces ressources sont partagées, distribuées, hétérogènes, délocalisées et autonomes.

On dénombre plusieurs différences entre le Grid Computing et le cluster Computing. Parmi elles, Les grilles sont généralement distribuées dans un réseau LAN, métropolitain ou WAN. Les ordinateurs et serveurs d'une grappe sont quant à eux généralement rassemblés au sein d'un même endroit. Le cluster Computing est le fait d'utiliser une grappe de serveurs. Le Grid Computing

consiste également à connecter plusieurs ordinateurs pour résoudre de grands problèmes, d'où la fréquente confusion entre le Grid Computing et le cluster Computing. La grande différence est que la grappe est homogène, tandis que les grilles sont hétérogènes.

Les serveurs et ordinateurs qui constituent une grille peuvent exécuter différents systèmes d'exploitation et des systèmes embarqués différents composants, tandis que les grappes de serveurs embarquent tous les mêmes composants et le même système d'exploitation. [4]

### 1.3.4 Les calculs à haute performance (High-performance Computing)

Le HPC permet de traiter des données et d'effectuer des calculs complexes en parallèle à grande vitesse afin de diminuer les temps de ces calculs. [5] Les entreprises utilisent maintenant des clusters à grande échelle qui sont souvent partagés entre plusieurs départements et facilement accessibles au public. [6]

Il permet aussi d'obtenir aujourd'hui des résultats qui ne seront atteignables sur de simples PCs que dans plus de dix ans. Les enjeux du HPC sont stratégiques : simulation et compréhension du climat, optimisation de la production d'énergie, analyse financière temps réel, analyse de sol pour la recherche pétrolière... et plus récemment en support aux applications de Big Data et de Machine Learning. Les pays les plus industrialisés de la planète font une course sans fin pour disposer des plus gros moyens de calculs, allant jusqu'à des systèmes de milliers de cœurs. Mais des systèmes de calcul à haute performance plus modestes sont maintenant couramment utilisés dans de très nombreuses entreprises et laboratoires, et deviennent accessibles dans des « Clouds » ou dans des systèmes embarqués. [7]

### 1.3.5 Internet des objets (IoT Computing)

De plus en plus d'objets de la vie quotidienne et de capteurs sont connectés à l'Internet. Cela permet à la fois de fournir de nouvelles fonctionnalités aux utilisateurs mais aussi d'optimiser l'utilisation de ressources. Avec l'Internet des objets ou web des objets, les objets communicants commencent à faire partie de notre quotidien. De la santé connectée, à la maison connectée comme le montre la Figure 2, ces différents objets communicants auront besoin d'échanger d'énormes quantités de données. [8]

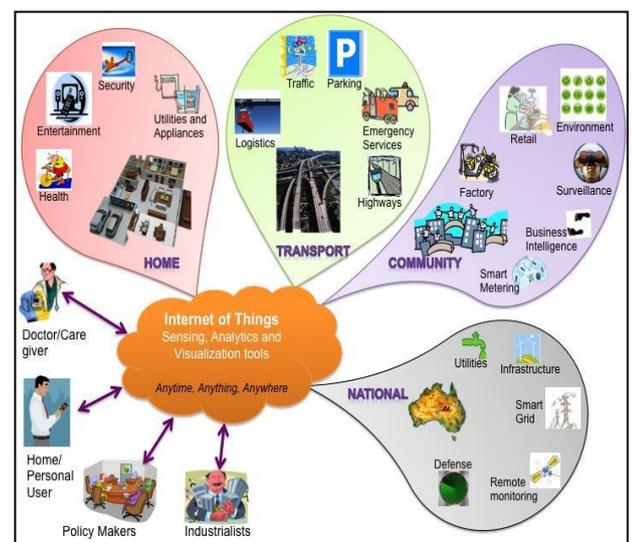


Figure 2 Internet of Things [8]

### 1.3.6 Le calcul bénévole (Volunteer Computing)

Le Volunteer Computing, est un type de calcul distribué dans lequel les propriétaires d'ordinateurs peuvent faire un don de leurs ressources informatiques de réserve (puissance de traitement, stockage et connexion Internet) à un ou plusieurs projets de recherche.

Les ordinateurs de bureau, portables et téléphones mobiles des volontaires peuvent être connectés pour former l'équivalent d'un seul, énorme et superpuissant superordinateur virtuel. Les ordinateurs n'utilisent souvent qu'entre 10 et 15 % de leur capacité de traitement totale, ce qui signifie qu'il y a beaucoup de puissance potentielle disponible. Un PC moderne peut effectuer des milliards d'opérations par seconde ; ainsi, même lorsque vous tapez 60 mots par minute, l'ordinateur est en grande partie inactif. Des calculs qui prendraient des dizaines de milliers d'années à exécuter sur une machine ordinaire peuvent être traités en quelques mois seulement en additionnant toutes ces ressources inutilisées. Dans le calcul bénévole, un problème est divisé en plusieurs tâches, chacune d'entre elles étant résolue par un ou plusieurs ordinateurs à la fois. [9]

### 1.3.7 Cloud, Edge et Fog Computing

Parlant en générale, dans cette section on va juste donner quelques petites définitions et quelques exemples simples pour montrer l'évolution de ces technologies, de Cloud (informatique dans les nuages), à l'Edge (informatique en périphérie) au Fog Computing (l'informatique en brouillard) (Figure 3).

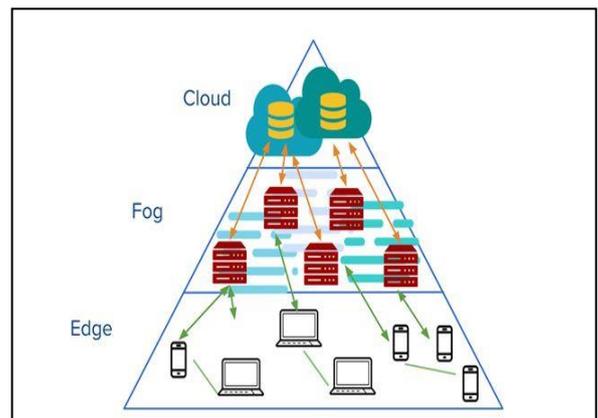


Figure 3 Cloud, Fog et Edge Computing [64]

Vu presque tout le monde entend le mot Cloud, donc il y a eu le Cloud Computing, un terme tellement vague et inexpressif qu'il a fallu rapidement le décliner en public, hybride ou privé ainsi qu'en une multitude de « as a Service » pour réellement décrire ce qu'il apportait.

Depuis quelque temps, le marketing des grands acteurs de l'IT nous vend de l'« Edge Computing », parce qu'il aurait probablement été trop minimaliste de la qualifier d'informatique embarquée et bien trop « has been » de l'appeler « infrastructure locale ».

Mais comme le terme « Edge » est plus généré, les services marketing des uns et des autres se sont déjà trouvés un nouveau « buzz word » : le Fog Computing !

Mais qu'est-ce donc ? Pour faire simple, il s'agit d'un mini-Cloud intercalé entre les objets et le Cloud (principalement le Cloud public). On parle parfois d'une informatique géo-distribuée : une

infrastructure décentralisée dont les ressources et les applications sont stratégiquement placées aux emplacements où elles seront les plus efficaces.

Le concept de «Fog Computing » a été inspiré par CISCO (avec sa vision réseau) alors que celui de l' « Edge Computing » proviendrait plutôt du marketing d'IBM (avec une vision plus applicative et serveur).

Bien qu'il n'existe aucune définition officielle de ces deux concepts similaires, on voit peu à peu se dessiner une subtile distinction entre les deux termes. Prenons un exemple simple : une ville décide de gérer de façon intelligente et dynamique ses feux rouges. Elle a des capteurs à chaque feu. Une intelligence pourrait être placée sur chaque feu ou plus intelligemment sur chaque boulevard pour adapter leur fonctionnement à la circulation. Là **c'est de l'Edge Computing**. Maintenant supposons que la ville veuille gérer de façon plus globale la situation en fonction de la météo, et de donner par exemple la priorité aux vélos puis aux bus en cas de pluie. Il faut une perception étendue du trafic de la ville avec notamment des capteurs de pluie permettant d'anticiper la situation tout en conservant la vision de tous les capteurs de trafic. Il faut donc une intelligence au niveau de la ville, ou plus exactement de chaque ville, et c'est du « **Fog Computing** ». Évidemment les flux de données de chaque ville finissent centralisés quelque part dans le « **Cloud Computing** ». [10]

### 1.3.8 L'informatique de la jungle (Jungle Computing)

L'informatique de la jungle est une combinaison simultanée de l'informatique hétérogène et distribuée de ressources. Dans de nombreux domaines de recherche scientifique réalistes, Les experts de domaine sont contraints d'utiliser simultanément plusieurs clusters, les grilles, Clouds, les ordinateurs indépendants, et plus encore. Ces nouveaux paradigmes de l'informatique distribuée ont conduit à une collection de ressources à la disposition des chercheurs, notamment des machines autonomes, les systèmes de clusters, les grilles et les Clouds, comme le montre la Figure 4, et cela a varié est appelée "informatique de la jungle". [6]

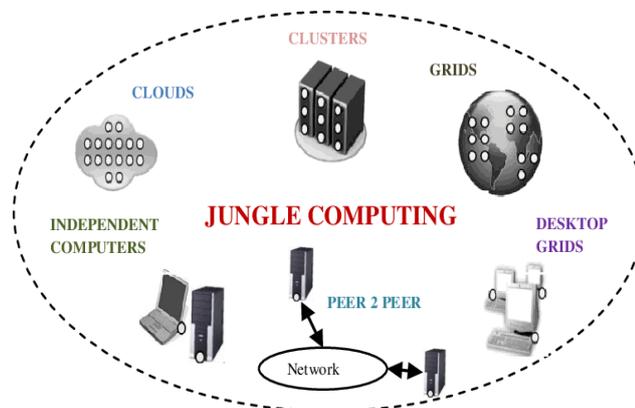


Figure 4 Jungle Computing [6]

## 1.4 Partie2 : Le Cloud Computing

### 1.4.1 Historique

Le concept de Cloud Computing est loin d'être nouveau, il est exploité depuis les années 2000, comme le montre la Figure 5, c'est la cinquième génération de l'informatique après les Mainframes, les PCs, les Clients/Serveurs et le Web [11], les changements qui ont permis l'apparition du Cloud Computing sont nombreux. Aussi il y a le concept de virtualisation qui permet une mutualisation des serveurs et offre donc une mise en production simplifiée et une meilleure utilisation des ressources. Le Cloud Computing est donc la juxtaposition de ces technologies pour passer à la vitesse supérieure sur l'exploitation de données à travers Internet. Le concept du Cloud Computing a été mis en œuvre en 2002 par Amazon, un leader du business, pour absorber la charge importante des commandes faites. [12]

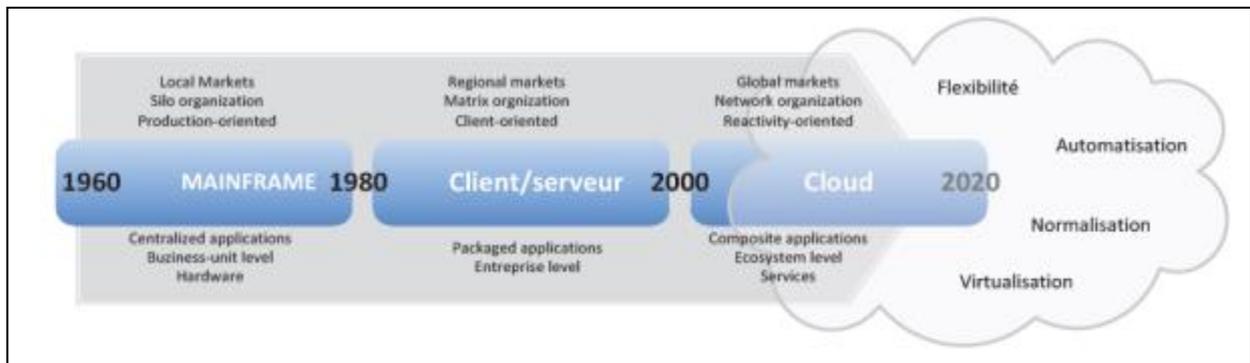


Figure 5 Historique du Cloud Computing [11]

### 1.4.2 Définition

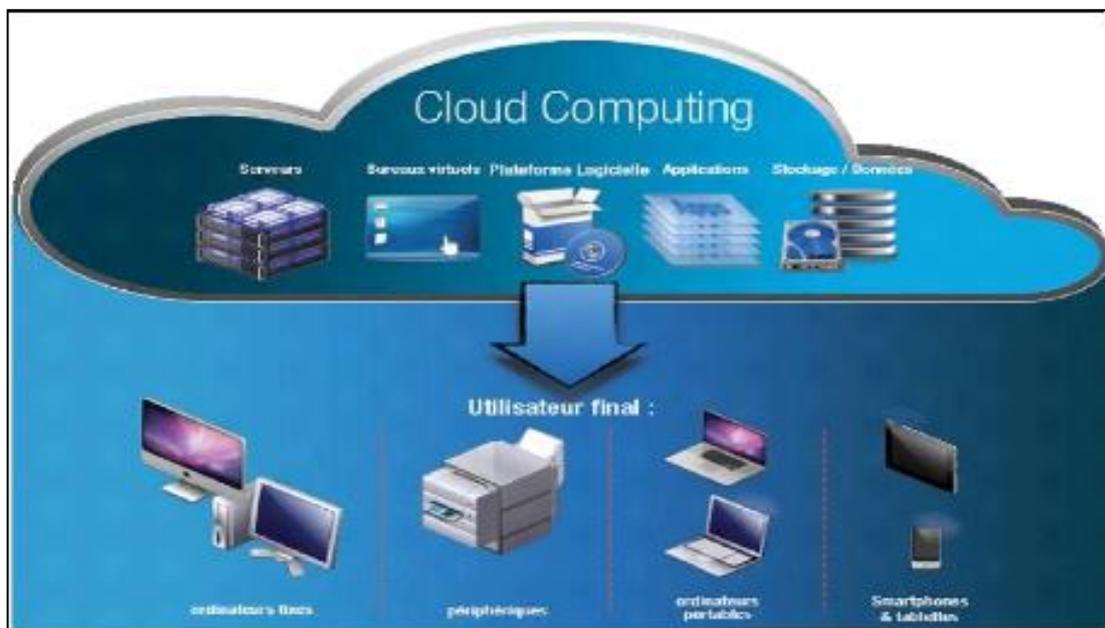


Figure 6 Le Cloud Computing [25]

Il existe dans la littérature plusieurs définitions du Cloud Computing. Cloud est un mot anglais qui veut dire « nuage ». Il est généralement matérialisé par un nuage qui dit « vous ne voyez pas ce qu'il y a à l'intérieur mais ne vous inquiétez pas ça fonctionne ». [13]

Pour NIST (National Institute of Standards and Technology), le Cloud Computing est un modèle permettant un accès réseau pratique et à la demande à un pool partagé de ressources informatiques (des réseaux, serveurs, stockage, applications, services) (Figure 6) pouvant être rapidement provisionnées avec un minimum d'effort de gestion du consommateur ou d'interaction avec le fournisseur de services. [14]

Autrement dit, le Cloud Computing, c'est de pouvoir utiliser des ressources informatiques sans les posséder ou les avoir physiquement. [13]

Ce modèle est composé de cinq caractéristiques essentielles, de trois modèles de services et de quatre modèles de déploiement.

### 1.4.3 Les cinq caractéristiques fondamentales du Cloud Computing

Le Cloud Computing vise à offrir des services qui vont au-delà de ces offres classiques et que l'on peut essayer d'offrir avec les caractéristiques suivantes [15]:

- ✓ Il s'agit d'un calcul distribué où les échanges sont gérés et centralisés par des serveurs distants, les applications étant stockées non plus sur le poste de travail, mais sur un "Cloud" de serveurs, accessibles par une connexion Internet et un navigateur Web;
- ✓ Les applications, les plateformes et les infrastructures nécessaires sont louées en fonction de l'usage qui en est fait, que ce soit lors du développement de ces applications ou lors de leur utilisation en production;
- ✓ Les applications, plates-formes et infrastructures sont facilement extensibles;
- ✓ Les ressources peuvent être allouées de manière dynamique en fonction des besoins;
- ✓ Les applications, les plateformes et l'infrastructure restent disponibles en cas de défaillance des ressources.

### 1.4.4 Eléments constitutifs du Cloud Computing

Un environnement Cloud Computing ne peut être défini sans deux éléments importants qui sont les centres de calcul et la virtualisation qui a été la première pierre vers l'ère du Cloud Computing qu'on va donner plus de détails sur elle vu que notre travail est une proposition d'une politique de gestion de tolérance aux pannes dans un environnement virtualisé qui est le Cloud Computing:

## a) Centre de Calcul (Datacenter)

Un centre de traitement de données (Data-Center en anglais) est un site physique, sur lequel se trouvent regroupés des équipements constituant le système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.) (Figure 7). Il peut être interne et/ou externe à l'entreprise, il comprend en général un contrôle sur l'environnement (climatisation, système de prévention contre l'incendie, etc.), une alimentation d'urgence et redondante, ainsi qu'une sécurité physique élevée. Cette infrastructure peut être propre à une entreprise et utilisée par elle seule, ou à des fins commerciales. [19]

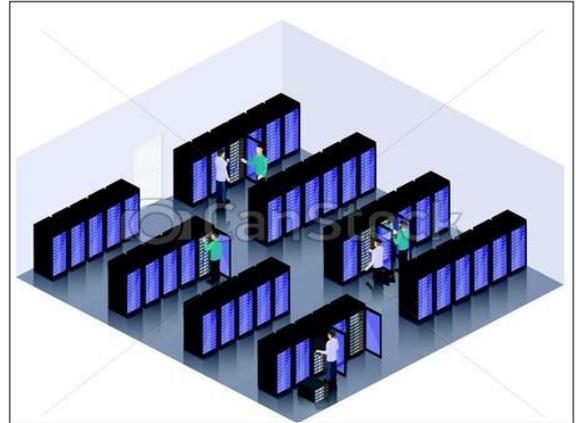


Figure 7 Data Center [59]

## b) La virtualisation

La virtualisation est tout simplement une technologie qui permet une gestion optimisée des ressources matérielles en disposant de plusieurs machines virtuelles sur une machine physique [16]. La Figure 8 représente un schéma d'architecture de la virtualisation.

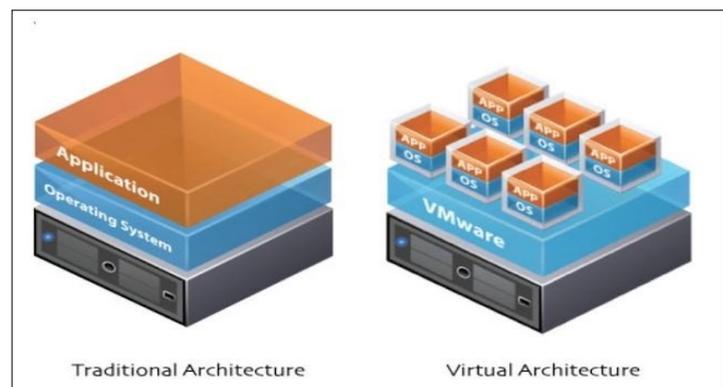


Figure 8 Architecture de la virtualisation [17]

Autrement dit, la virtualisation se définit comme l'ensemble des techniques matérielles et/ou logicielles qui fonctionnent sur une seule machine, plusieurs systèmes d'exploitation, appelées machines virtuelles (MVs), ou encore des systèmes d'exploitation invités [18], en se basant sur deux principes fondamentaux: [66]

- **Le cloisonnement:** chaque système d'exploitation fonctionne indépendamment des autres et ne peut pas interférer les autres ;

- **La transparence:** le fonctionnement en mode virtualisé ne change rien au fonctionnement du système d'exploitation ou des applications.

Le principe de virtualisation repose sur le mécanisme suivant:

- ✓ Un système hôte qui est un simple système d'exploitation installé sur un serveur physique unique, va servir afin d'accueillir plusieurs autres systèmes d'exploitation par un logiciel appelé hyperviseur.

✓ L'hyperviseur est donc un logiciel de virtualisation qui est directement installé sur le système d'exploitation principal ou hôte. Celui-ci va permettre de créer plusieurs environnements clos et indépendants qui pourront à leur tour héberger d'autres systèmes invités. Les environnements indépendants ainsi créés grâce à l'hyperviseur sont des machines virtuelles.

Donc, le système hôte va créer plusieurs environnements entièrement indépendants grâce à un hyperviseur. Ces machines virtuelles vont pouvoir communiquer avec les ressources physiques et le hardware du serveur physique (mémoire, espace disque ...etc.).

Les hyperviseurs peuvent être divisés en deux types :

### ✓ **Hyperviseur de type 01**

Noyau allégé et optimisé aussi appelé natif, est un logiciel qui s'exécute directement sur une plateforme matérielle, elle est considérée comme outil de contrôle de système d'exploitation. De ce fait un système d'exploitation secondaire peut être exécuté au-dessus du matériel. Parmi les hyperviseurs de type 01, on trouve : Xen, Hyper-V et ESXI Server ...etc.

### ✓ **Hyperviseur de type 02**

Dit hosted ou architecture invitée, est un logiciel qui s'exécute à l'intérieur d'un autre système d'exploitation, un système d'exploitation invité s'exécutera en troisième niveau au-dessus du matériel. L'intérêt qu'on peut trouver c'est le fait de pouvoir exécuter plusieurs hyperviseurs simultanément vu qu'ils ne sont pas liés à la couche matérielle.

Parmi les hyperviseurs de type 2 on trouve : VMware Player, VMware Workstation et Virtual Box ...etc.

## **1.4.5 Techniques de la virtualisation**

Il existe différentes techniques de virtualisation, nous citons par niveau d'abstraction décroissant : [66]

### **a) Virtualisation complète**

Virtualisation est dite complète lorsque le système d'exploitation invité n'a pas conscience d'être virtualisé, donc n'a aucun moyen de savoir qu'il partage le matériel avec d'autres OS. Ainsi, l'ensemble des systèmes d'exploitation virtualisés s'exécute sur un ordinateur unique, pouvant fonctionner de manière totalement indépendante les uns des autres et être vu comme des ordinateurs à part entière sur réseau.

### b) La paravirtualisation

Par opposition à la virtualisation, on parle de la paravirtualisation lorsque le système d'exploitation doit être modifié pour fonctionner sur un hyperviseur de paravirtualisation. Les modifications sont en fait des insertions de divers permettant de rediriger les appels système au lieu de les traduire (par des drivers backend et frontend).

### c) L'isolation

Consiste à mettre en place, sur un même noyau de système d'exploitation, une séparation forte entre différents contextes logiciels. Il s'agit de la technique de virtualisation la plus « légère » qui existe.

## 1.4.6 Types de la virtualisation

Il existe différents types de virtualisation, on va présenter les principaux types seulement et vu que notre travail consiste à proposer une politique pour assurer la tolérance aux pannes dans le Cloud Computing donc, on est intéressé par la virtualisation de stockage alors on va donner plus de détails sur elle et on présentera les autres types en bref comme suit: [66]

### a) Virtualisation du stockage

Consiste à regrouper l'ensemble des périphériques de stockage physique en un seul périphérique simulé centralisé. Ce dernier est géré depuis une console centrale. La manipulation du stockage dans le Cloud est surtout utilisée pour la sauvegarde, l'archivage et la récupération des données en masquant l'architecture de stockage réelle et physique complexe. Il existe plusieurs technologies de stockages, nous allons présenter quelques une permettant de mettre en place un réseau de stockage:

#### ✓ NAS et NFS

Un NAS, ou stockage réseau (Network-Attached Storage) est simplement un serveur fournissant leurs fichiers à d'autres serveurs par le réseau. NFS est le standard universel pour l'accès aux fichiers sur un réseau, c'est le protocole le plus utilisé dans les NAS. Dans le cadre d'un isolateur, il permet de stocker l'arborescence du serveur virtuel à distance. Dans le cadre d'une solution de virtualisation complète il permet de stocker à distance les fichiers contenant les disques durs de la machine virtuelle. Ce dernier cas est déconseillé hors des environnements de test : NFS n'est pas adapté à la lecture aléatoire dans un seul fichier. En revanche pour un isolateur, stocker les données en NFS est intéressant, et le deviendra encore plus avec les systèmes de fichier de nouvelle génération tels que ZFS, HAMMER ou BTRFS, qui permettent des snapshots instantanés, le versionnement des arborescences, et autres fonctionnalités pour l'instant réservées aux baies de stockage haut de gamme. En plus des matériels dédiés, la plupart des systèmes d'exploitation proposent une implémentation

serveur NFS, ce qui permet d'utiliser n'importe quel serveur comme serveur de stockage NFS. Ces derniers utilisent alors soit des disques locaux, soit leur propre réseau de stockage SAN. Les volumes NAS peuvent être créés sur les baies de disques RAID.

### ✓ SAN

L'autre technologie importante de stockage de données est le SAN. Un SAN, ou réseau de stockage (Storage Area Network), est un réseau sur lequel circulent les données entre un système et son stockage. Cette technique permet de déporter tout le stockage interne d'une machine vers un équipement dédié. Les SAN sont des équipements dédiés, qui ne travaillent qu'aux plus basses couches du stockage, la notion de fichier leur est inconnue ; ils travaillent simplement sur des blocs de données et les fournit par le réseau à des serveurs qui eux sauront les utiliser. Cependant, les SAN les plus hauts de gamme sont dotés de capacités avancées, tel que la prise de cliché, ou encore la copie rapide de volumes. Les deux principaux protocoles d'accès à un SAN sont iSCSI et Fibre Channel.

### ✓ iSCSI (Internet Small Computer System Interface)

iSCSI est un protocole d'accès disque fonctionnant sur un réseau Ethernet, il permet d'implémenter un réseau de stockage en profitant de la connectique et des équipements de commutation standards. Comme le NFS, il peut être soit implémenté par une baie de stockage dédiée, ce qui assure les meilleures performances, soit par un serveur classique disposant du logiciel adéquat, par exemple IET (iSCSI Enterprise Target) sous Linux.

### ✓ Fibre Channel

La solution la plus haut-de-gamme pour implémenter un réseau de stockage est l'utilisation d'une baie dédiée et du protocole Fibre Channel. Basé sur des fibres optiques il assure une latence et un débit bien meilleurs qu'iSCSI, à un prix bien sûr plus élevé. Son principe d'utilisation est le même qu'un SAN iSCSI.

### b) Virtualisation de matérielle

C'est le type le plus courant, dans ce cas un gestionnaire de machine virtuelle reflète les versions virtuelles des systèmes d'exploitation et les fusionne en un énorme serveur physique unique. Il aide à exécuter plusieurs systèmes d'exploitation sur un même périphérique en même temps.

### c) Virtualisation des applications

Une application peut être utilisée sur un autre hôte à partir duquel elle est installée dans différentes tailles. Ce type est également appelé virtualisation du service d'application.

### d) Virtualisation des postes de travail

L'environnement de bureau de travail est séparé de la machine physique et organisé en infrastructure de postes de travail virtuels. Elle aide les utilisateurs à accéder à leurs bureaux à partir de n'importe quel endroit.

### e) Virtualisation du réseau

C'est le processus reproduisant un réseau physique et ses différents composants (ports, routeurs...etc.), cette technique permet de mettre à disposition des réseaux isolés, cloisonnés mais fonctionnant sur une infrastructure mutualisée ; aussi aide à mieux superviser et à identifier l'utilisation des données. Elle assure également la sécurité en limitant le mouvement des fichiers sur plusieurs réseaux.

### f) Virtualisation de serveur

Par un principe d'émulation, une couche logicielle « hyperviseur » isole les ressources physiques des systèmes d'exploitation et les regroupent en un seul serveur. Par ce principe, plusieurs systèmes d'exploitation peuvent cohabiter sur une même machine, indépendamment l'un de l'autre.

## 1.4.7 Les modèles de services

La partie visible du Cloud Computing est composé de plusieurs catégories de services, les trois principales catégories sont: IaaS, PaaS et SaaS présentées dans la Figure 9.

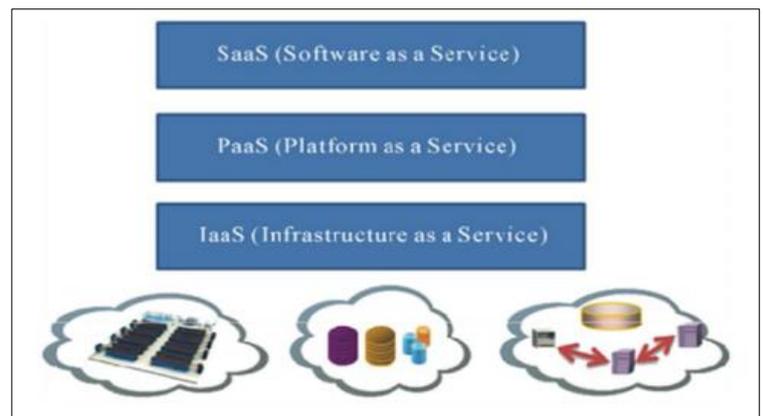


Figure 9 Les modèles de services [21]

### 1) IaaS (Infrastructure as a Service, des services virtuels disponibles à la demande)

Les services IaaS du Cloud Computing permettent de mettre à la disposition des utilisateurs, des ressources matérielles (réseau, stockage et systèmes d'exploitation) accessibles sous forme virtuelle. [20]

### 2) PaaS (Platform as a Service, des plateformes de développement prêtes à l'emploi)

Le PaaS, permet d'externaliser l'infrastructure matérielle (hardware) mais également l'environnement logiciel (software). Le PaaS s'adresse aux professionnels et en particulier aux développeurs puisqu'il offre la possibilité de garder la main sur l'installation et le développement des applications nécessaires à l'entreprise via un environnement Cloud. Le mode PaaS permet donc de louer l'exploitation des serveurs et les outils intégrés. Cette configuration est également appréciée

pour héberger un site web sans avoir à en gérer les couches système. Parmi les exemples de PaaS, on compte en outre Azure de Microsoft. [20]

### 3) SaaS (Software as a Service, des services métiers à la demande)

Les services SaaS du Cloud Computing permettent de mettre à la disposition des utilisateurs des applications prêtes à l'emploi. [21]

Le tableau1 présente des exemples d'offres de services Clouds existants sur le marché des leaders mondiaux. [16]

IaaS	PaaS	SaaS
-Amazon offre EC2 et AWS	-Microsoft offre Azur	- Google offre Google Apps (messagerie et bureautique)
-Microsoft offre Azur	- Google offre Google App Engine	- Salesforce CRM (Customer Relationship Management)

Tableau 1 Les leaders mondiaux du Cloud Computing [16]

Les avantages et les inconvénients de chaque modèle de services sont résumés dans le tableau2 [22]:

	avantage	inconvénient
<b>SaaS</b>	-pas d'installation -plus de licence -migration	-logiciel limité -sécurité -dépendances des prestataires
<b>PaaS</b>	-pas d'infrastructure nécessaire -pas d'installation -environnement hétérogène	-limitation des langages -pas de personnalisation dans la configuration des machines virtuelles
<b>IaaS</b>	-administration -personnalisation -flexibilité d'utilisation	-sécurité -besoin d'un administrateur système

Tableau 2 Avantages et Inconvénients des services Cloud. [22]

### 1.4.8 Les modèles de déploiement

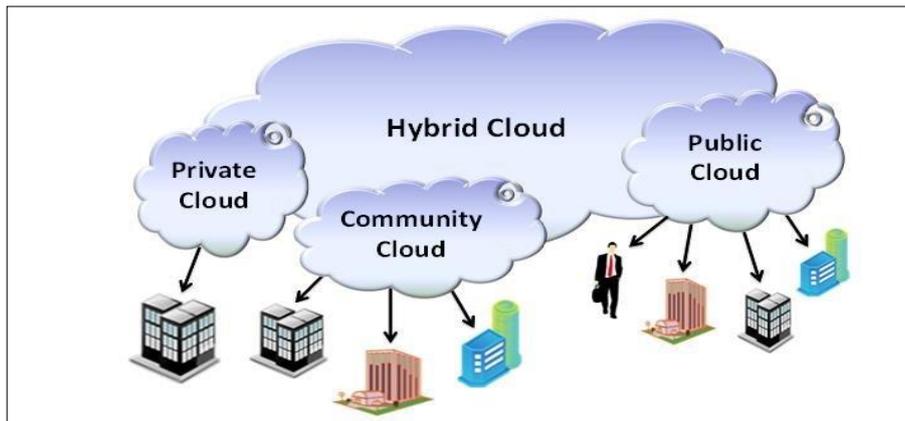


Figure 10 Les modèles de déploiement. [23]

Le NIST a défini quatre modèles de déploiement pour les services de Cloud présentés dans la Figure 10, quel que soit le modèle de service utilisé (SaaS, PaaS ou IaaS), avec des variantes qui répondent à des besoins spécifiques: [24]

#### 1) Cloud communautaire (Community Cloud)

L'architecture est dédiée à une communauté professionnelle spécifique, pour permettre de travailler de manière collaborative sur un même projet. À titre d'exemple: UnivCloud<sup>1</sup>.

#### 2) Cloud privé (Private Cloud)

- **Le Cloud privé interne:** Ce Cloud privé interne est à l'usage de plusieurs consommateurs appartenant à cette seule entreprise qui est propriétaire de l'infrastructure. Elle peut également être partagée de façon privée avec les filiales.
- **Le Cloud privé externe:** Le Cloud privé externe suit la même logique que le Cloud privé interne. La différence est que l'architecture est hébergée chez un prestataire. Elle est entièrement dédiée à l'entreprise et accessible via des réseaux sécurisés (VPN).

#### 3) Cloud publique (Public Cloud)

Un Cloud public est basé sur un modèle standard de Cloud Computing, dans lequel un prestataire de service met les ressources, tels que les applications, ou le stockage, à la disposition du grand public via internet. Le Cloud public peut être gratuit ou fonctionner selon paiement à la consommation.

L'avantage de ce genre d'architecture est d'être facile à mettre en place, pour des coûts relativement raisonnables. La charge du matériel, des applications, de la bande passante étant couverte par le fournisseur, ce modèle permet de proposer une souplesse et une évolutivité accrue

<sup>1</sup> <http://univcloud.fr/>

afin de répondre rapidement au besoin. Il n'y a pas de gaspillage de ressources car le client ne paye que ce qu'il consomme [25].

Bien que le Cloud public offre un maximum de flexibilité, il demande de lourds investissements pour le fournisseur de services. De plus, ce type de Cloud n'est pas **sécurisé**.

#### 4) Cloud hybride (Hybrid Cloud)

Comme nous venons de le voir avec le Cloud public, certains problèmes peuvent se poser, liés à la sécurité de l'information. Il est alors possible de mélanger les deux approches du Cloud, privé et public, en débouchant sur une plate-forme hybride comme le montre la Figure 11.

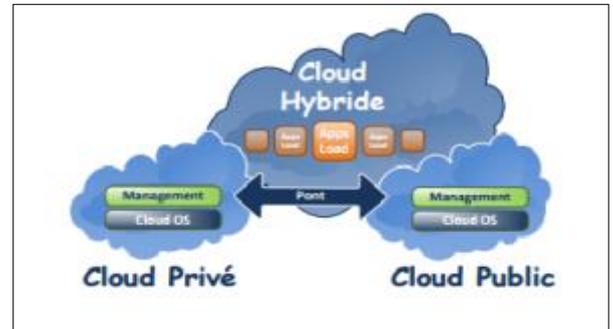


Figure 11 Cloud hybride [60]

### 1.4.9 Avantages et inconvénients du Cloud Computing

Les principaux avantages et inconvénients du Cloud Computing sont les suivants [26]:

#### ➤ Avantages

- ❖ **Flexibilité:** plusieurs choix des fournisseurs qui offrent des services fiables et à grande échelle aux entreprises;
- ❖ **La virtualisation:** c'est à dire pas besoin d'investir pour concevoir une plateforme de Cloud Computing ;
- ❖ **Réduction du coût:** dû à l'efficacité opérationnelle et déploiement plus rapide de nouveaux services aux entreprises ;
- ❖ **L'agilité pour l'entreprise:** résolution des problèmes de gestion informatique simplement sans avoir à vous engager à long terme ;
- ❖ **Un développement plus rapide des produits:** réduisant le temps de recherche pour les développeurs sur le paramétrage des applications ;
- ❖ **Pas de dépenses de capital:** plus besoin de locaux pour élargir vos infrastructures informatiques.

### ➤ Inconvénients

- ❖ **La bande passante peut faire exploser votre budget:** La bande passante qui serait nécessaire pour mettre cela dans le Cloud est gigantesque;
- ❖ **La sécurité des applications peut être amoindrie:** Un Cloud public n'assure définitivement pas la sécurité des applications;
- ❖ **La fiabilité du Cloud :** un grand risque lorsqu'on met une application qui donne des avantages compétitifs ou qui contient des informations clients dans le Cloud;
- ❖ **Taille de l'entreprise:** si votre entreprise est grande alors vos ressources sont grandes, ce qui inclut une grande consommation du Cloud. Vous trouverez peut être plus d'intérêt à mettre au point votre propre Cloud plutôt que d'en utiliser un externalisé. Les gains sont bien plus importants quand on passe d'une petite consommation de ressources à une consommation plus importante.

## 1.5 Conclusion

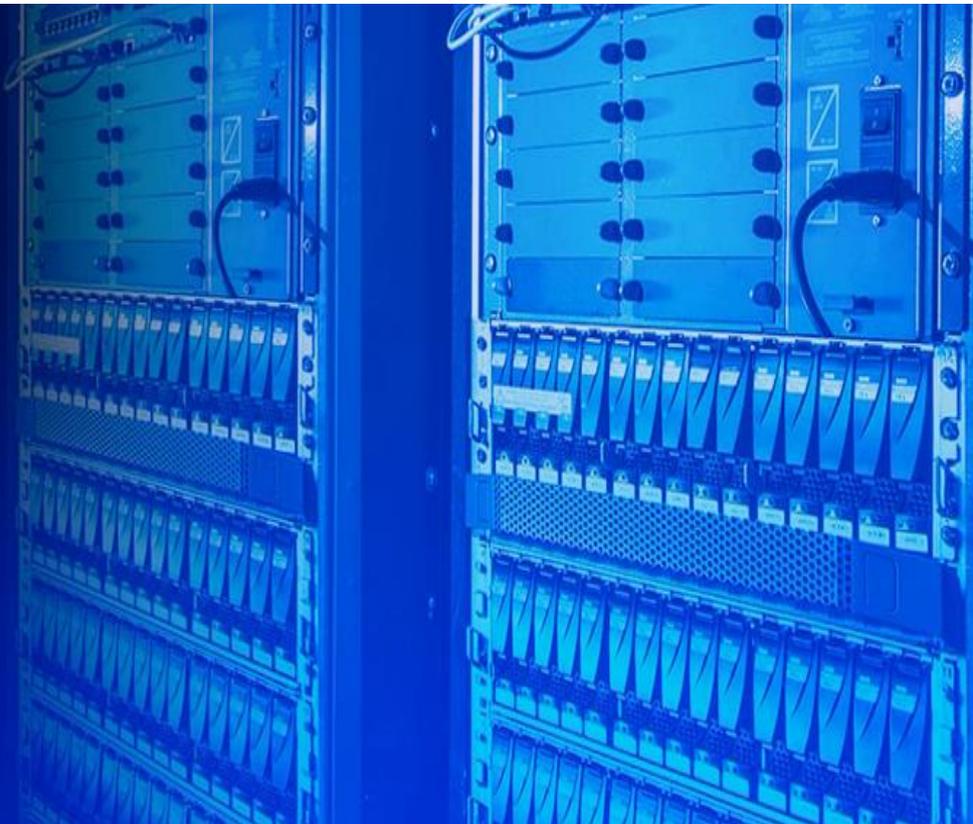
Tout au long de ce chapitre, nous avons abordé une vue générale sur les systèmes distribués en donnant quelques exemples après nous avons entamé le concept de Cloud Computing commençant par sa définition et son historique, puis nous avons spécifié ces caractéristiques fondamentales, ses composants ainsi que ses modèles de déploiements et de services après nous avons cité quelques avantages et inconvénients.

Le Cloud Computing est un nuage de services et de données, basé sur une supervision centralisée. Malgré tous ses avantages, les politiques de tolérance aux pannes actuelles ne sont pas parfaites, ce qui peut présenter toujours des risques d'entraîner des ruptures dans le fonctionnement global du système.

Puisque notre travail consiste à proposer une solution de tolérance aux pannes dans l'environnement Cloud, on va entamer ce domaine où nous allons introduire le concept de la tolérance aux pannes, ses types et ses techniques...etc dans le chapitre suivant.

## Chapitre 2 :

# Tolérance aux pannes



## 2.1 Introduction

Ces dernières années, des progrès importants ont été réalisés dans le domaine des systèmes informatiques, qui ont permis le développement d'environnements distribués. Avec ce développement, le système est devenu de plus en plus complexe, de nombreuses pannes se sont produites et la continuité de son fonctionnement n'est plus garantie. Pour résoudre ces problèmes, des techniques de tolérance aux pannes ont été mises en œuvre. La capacité d'un système à fonctionner même en cas de panne d'un de ses composants est appelée, tolérance aux pannes.

Dans ce chapitre, nous présenterons dans un premier temps le concept de la sûreté de fonctionnement ainsi que les classifications des pannes. Ensuite, nous exposerons quelques techniques et à la fin nous allons introduire quelques travaux connexes sur les techniques de tolérance aux pannes dans l'environnement Cloud Computing où nous nous sommes intéressés dans ce travail.

## 2.2 Notion de sûreté de fonctionnement

La sûreté de fonctionnement (SdF) d'un système est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre. [27]

Elle peut être présentée par trois paramètres: les attributs, les entraves et les moyens, décrits par la Figure 12 [28].

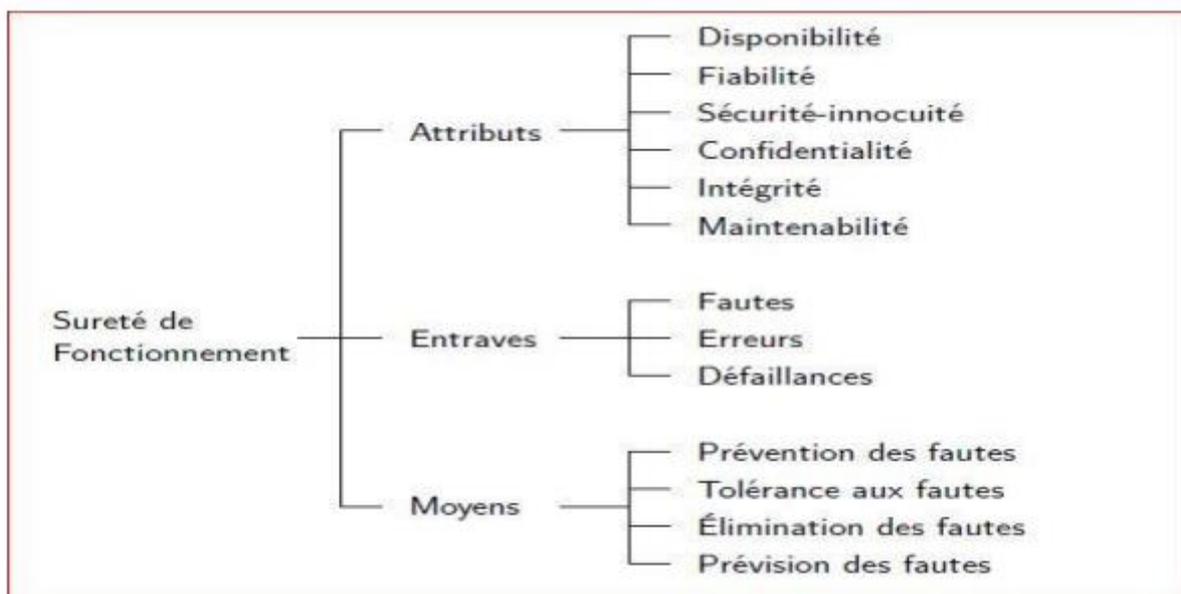


Figure 12 L'arbre de la sûreté de fonctionnement [28]

### 2.2.1 Attributs de la sûreté de fonctionnement

Les attributs de la sûreté de fonctionnement correspondent aux propriétés que doivent vérifier un système. Ces attributs permettent d'évaluer la QoS fournie par le système. Il existe six attributs de la sûreté de fonctionnement sont les suivants :

- ✓ **La disponibilité (Availability):** c'est la capacité de rendre disponible un service à tout instant;
- ✓ **La fiabilité (Reliability):** représente la probabilité d'un système à accomplir l'ensemble des fonctions spécifiées dans son cahier des charges, dans un environnement donné et pour un temps de fonctionnement donné;
- ✓ **La sécurité-innocuité (Security):** c'est l'absence de conséquences catastrophiques sur l'utilisateur ou son environnement;
- ✓ **La maintenabilité (Repairability):** c'est l'aptitude du système à être réparé ou amélioré ;
- ✓ **L'intégrité (Integrity):** c'est-à-dire garantir que les données sont bien celles que l'on croit être;
- ✓ **L'authenticité (Authenticity):** consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées;
- ✓ **La confidentialité (Privacy):** la confidentialité des données est la protection des communications ou des données stockées contre l'interception et la lecture par des personnes non autorisées [65].

### 2.2.2 Entraves à la sûreté de fonctionnement

Dans le vocabulaire technique, on évite le mot « panne », à cause des ambiguïtés associées. Les entraves de la sûreté de fonctionnement sont réparties en 3 notions : les fautes (fault en anglais), les erreurs (errors en anglais) et les défaillances (failures en anglais) qui sont schématisées dans la Figure 13 qui présente chaîne fondamentale des entraves à la sûreté de fonctionnement. [30]

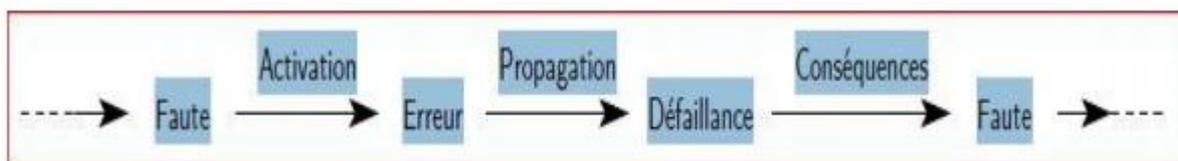


Figure 13 La chaîne fondamentale des entraves à la sûreté de fonctionnement [29]

L'activation d'une faute du système, provoque la propagation d'erreurs dans le système. Une défaillance a lieu lorsqu'une erreur est observée.

Un système est une combinaison de plusieurs composants, donc leur défaillance est une source d'erreur pour le système, dont elle peut lui provoquer un comportement anormal (il s'agit d'une faute pour le système). La chaîne faute, erreur et défaillance devient alors récurrentes.

- ✓ **Défaillance:** Un élément est dit défaillant lorsqu'il ne se comporte pas conformément à sa spécification ;
- ✓ **Erreur:** Une erreur est l'état anormal d'un élément, susceptible de causer une défaillance, mais cette dernière n'est pas encore déclarée ;
- ✓ **Faute:** Une faute peut être une action, un événement ou une circonstance qui peut entraîner une erreur.

Autrement dit, Une faute est la conséquence d'une défaillance (en général humaine) lors de la conception du système. Une erreur est un état atteint par le système qui n'avait pas été spécifié, un état que le système n'était pas supposé d'atteindre. On a une erreur parce qu'il y a une faute (ou plusieurs), on a une défaillance parce qu'il y a une erreur (ou plusieurs). [27]

### 2.2.3 Moyens de la sûreté de fonctionnement

Afin d'améliorer la fiabilité des systèmes sûrs de fonctionnement, et durant la phase de développement nous passons, quatre méthodes peuvent être utilisées : [27]

- **La prévention des fautes :** consiste à empêcher l'occurrence de faute et de diminuer la probabilité de leur apparition;
- **L'élimination des fautes:** c'est de détecter, identifier et enlever les fautes du système ;
- **La prévision des fautes:** consiste à faire une estimation probabiliste des fautes qui peuvent se produire et ayant une influence sur le système [30];
- **La tolérance aux pannes (appelée aussi, tolérance aux fautes) :** C'est de diminuer la probabilité de défaillance dans un système malgré la présence éventuelle de pannes (ce concept sera détaillé par la suite).

## 2.3 Classification des pannes

Les pannes sont classifiées selon différents critères. Une classification générale est montrée sur le schéma de la Figure 14 selon : le degré de gravité de la défaillance et le degré de la permanence de la panne.

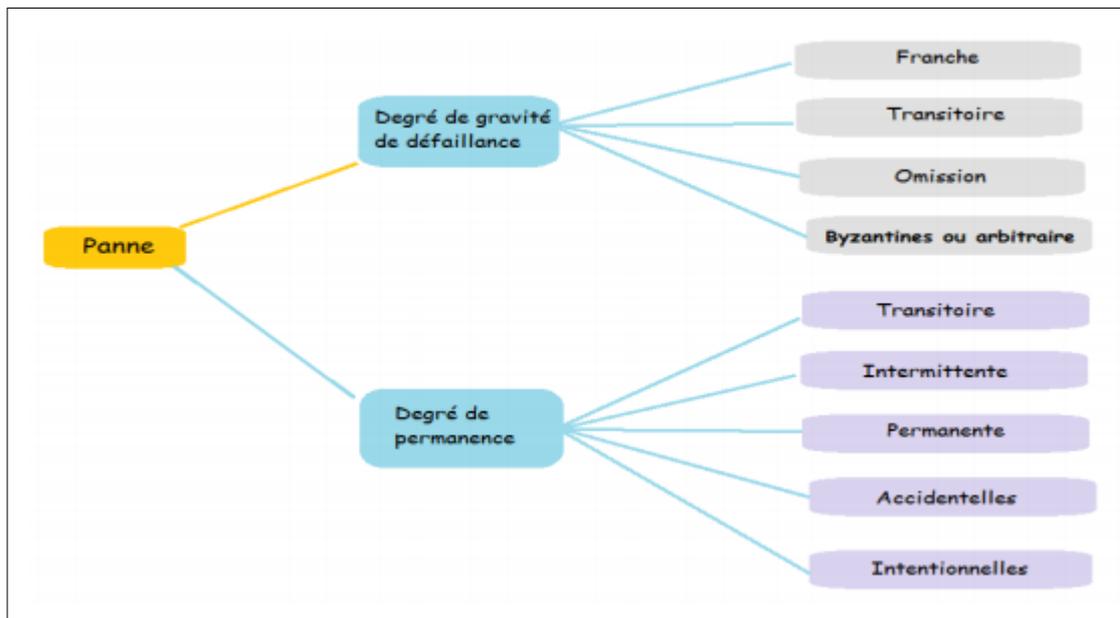


Figure 14 Classification générale des pannes [31]

### 2.3.1 Panne selon la gravité des défaillances

#### A. Panne franche (Crash, Fail-stop, Fail-silent)

Une panne franche est une panne permanente, c'est-à-dire qu'une fois le composant est en panne franche, il cesse immédiatement et de façon indéfinie de répondre à toute sollicitation ou de générer de nouvelles requêtes (jusqu'à une réparation) tel que certains types de programmes erronés (exemple de boucle). [31]

#### B. Panne par omission ou transitoire (transient, omission failure)

En réponse à un événement en entrée, un composant ne délivre jamais la réponse attendue (le composant ne peut pas fournir son service habituel pendant une certaine période => perte de quelques données). Ultérieurement, il peut répondre à nouveau de façon correcte. [31]

#### C. Panne temporelle (timing, performance failures)

C'est le comportement anormal du système par rapport à une période de temps, comme par exemple le temps de réponse qui dépasse les exigences des spécifications du système, plus simplement la panne temporelle est une panne franche mais non définitive. [32]

#### D. Panne byzantine ou arbitraire (malicious, byzantine failures)

Inclut tous les types de pannes, pour cela nous pouvons distinguer deux types de pannes byzantines, la Figure 15 montre la notion de panne byzantine. [31]

- ✓ **Panne byzantine naturelle** : consiste soit à une erreur matérielle ou une erreur logicielle;

- ✓ **Panne byzantine malicieuse:** c'est un comportement visant à faire échouer le système (virus, ver...).

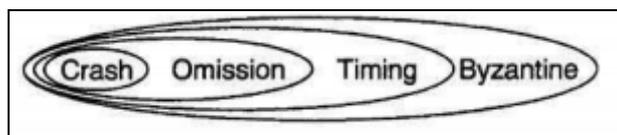


Figure 15 Panne byzantine [31]

### 2.3.2 Panne selon le degré de la permanence

Une seconde classification des pannes selon leur degré de permanence est également possible. Nous distinguons ainsi les pannes transitoires qui apparaît une seule fois puis disparaît, les pannes intermittentes qui apparaît plus ou moins périodiquement et les pannes permanentes qui persistent dès qu'elles apparaissent jusqu'à la réparation. Aussi, nous pouvons distinguer la nature des pannes selon qu'elles soient accidentelles (par exemple: installation d'un composant mal configuré) ou intentionnelles (sont des actions malveillantes, par exemple: les virus, Les attaques...).

## 2.4 Tolérance aux pannes

La tolérance aux pannes est une méthode de conception qui permet à un système de rester fonctionnel, éventuellement de manière réduite au lieu de tomber complètement en panne et rester plus ou moins opérationnel lorsque l'un de ses composants ne fonctionne plus correctement [33]. D'une autre façon le système ne s'arrête pas de fonctionner, s'il y ait une défaillance matérielle ou bien une défaillance logicielle. [34]

### 2.4.1 Les politiques de tolérances aux pannes

Ils existent deux types de politiques de la tolérance aux pannes standard qui sont les suivantes : [35]

#### ❖ Tolérance aux pannes proactive

Ici le système doit prédire les pannes avant leurs apparitions pour but de les empêcher, et ceci en utilisant des méthodes de développement pour assurer la prévention des pannes. En générale en se basant sur des solutions d'analyse de comportement de système d'une manière continue.

De cette manière, il peut commencer la récupération avant que les composants tombent en panne et évite ou au moins minimise le temps nécessaire pour la stabilisation. Par exemple, remplacer les composants soupçonnés par d'autres composants corrects, comme illustré dans la Figure 16 qui présente la ligne de temps de détection des pannes proactives. [35]

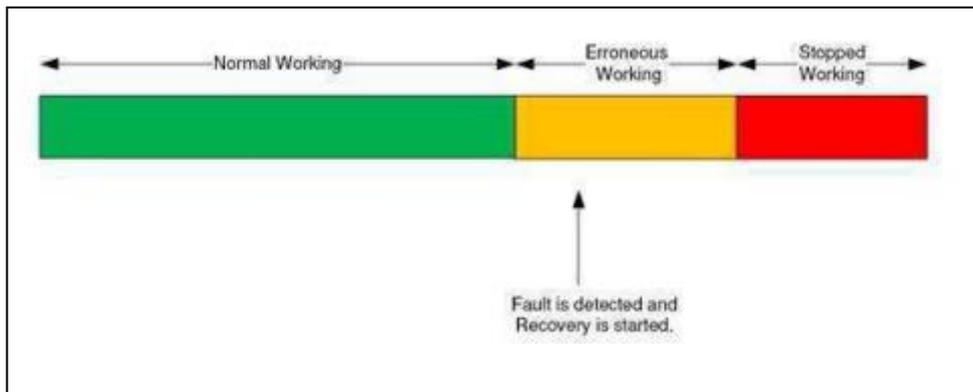


Figure 16 La ligne de temps de détection des pannes proactives [35]

Donc, la tolérance proactive suppose que les composants ont un comportement prédictif et le système illustre des symptômes de panne avant la panne réelle. Ce type de détection donne au système le temps nécessaire pour réagir avant la panne. La détection prédictive est favorable dans le cas des systèmes à temps réel. Pour que la prédiction de défaillance soit performante, il faut choisir les meilleurs mécanismes de prédiction qui s'adaptent avec les caractéristiques du système. [35]

❖ **Tolérance aux pannes réactive**

Cette politique est utilisée pour réduire l'impact des défaillances sur un système lorsqu'une défaillance se produit effectivement.

Le système détecte la faute après son occurrence et commence la récupération, il fournit une bonne solution de tolérance aux pannes pour les environnements informatiques généraux, mais il ne satisfait pas les contraintes de temps posées par les systèmes de contrôle de temps réel (par exemple: les systèmes de contrôle des patients dans les hôpitaux, ...), comme illustré dans la Figure 17 qui présente la ligne de temps de détection des pannes réactives. [35]

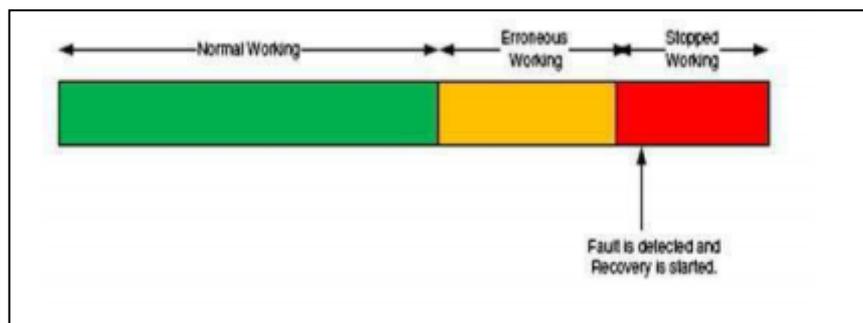


Figure 17 La ligne de temps de détection des pannes réactives [35]

## 2.5 Techniques utilisées pour assurer la tolérance aux pannes

Nombreuses techniques de base pour assurer la tolérance aux pannes sont connues, pour but d'éviter les échecs et d'empêcher les fautes introduisent lors de fonctionnement normale du système, nous citons par exemple:

### 2.5.1 Réplication

Parmi les techniques les plus utilisées pour l'assurance de tolérance aux pannes et afin de rendre l'exécution réussite, cette technique consiste à conserver de multiples copies de données ou d'objets, donc il nécessite deux fois l'espace de stockage, ces copies seront utilisées lorsque les pannes se produisent. [36]

On peut toujours confondre entre réplication et sauvegarde, ces deux termes sont quasiment différents, la sauvegarde des données ne change pas dans le temps tandis que la réplication est toujours synchronisée avec la copie d'origine. [32]

Cette technique est classée parmi les techniques de tolérance aux pannes les plus utilisées dans l'environnement Cloud Computing.

### 2.5.2 Resoumission des tâches (Task Resubmission)

Cette technique est basée sur la resoumission de tâches lorsqu'une erreur est détectée. Les processus de resoumission doivent être effectués sans interrompre le flux de travail du système. [36]

### 2.5.3 Migration

Un composant, serveur ou service est susceptible de tomber en panne lors de son exécution. Pour éviter l'arrêt du système, on migre vers un composant, serveur ou service plus robuste. A titre d'exemple, si une baie de stockage tombe en panne, la baie redondante prend le relai avec un temps d'arrêt minimal du système même chose en cas d'arrêt d'un serveur ou d'un service. [32]

### 2.5.4 Check-Point / Redémarrage

Cette approche est basée sur la sauvegarde périodique, qui est placée sur un support de stockage stable, persistant et fiable, en cas d'arrêt du système celui-ci sera redémarré à partir de ce point de sauvegarde appelé Check-Point. [32]

### 2.5.5 RAID

RAID (Redundant Array of Independent Disks) est une configuration de stockage mise en place par l'université Américaine de Berkeley dès 1987. Il est très utilisé dans les systèmes d'exploitation. Le but est alors de répartir les données sur différents disques durs afin d'améliorer la performance, la tolérance aux pannes...etc.

Par ailleurs, il existe beaucoup de configuration RAID différentes (plusieurs niveaux), pour que l'utilisateur puisse choisir le niveau qui satisfait ses besoins. L'amélioration des niveaux de cette technique permet à satisfaire plus d'un besoin sur un seul niveau (par exemple, dans certains niveaux, on trouve l'assurance du besoin de tolérance aux pannes + une optimisation de l'espace de stockage, cela apporte plus de bénéfices à l'utilisateur ==> minimisation des couts...etc.), et ça pour un serveur

ou un ordinateur personnel. C'est pour cela, le RAID a prouvé des meilleurs résultats et il est utilisé dès maintenant.

Afin d'assurer la tolérance aux pannes, les techniques citées sont largement utilisés dans les systèmes d'exploitation mais seules les quatre premières techniques (Réplication, Migration, Check-Point et Resoumission des tâches) sont parmi les techniques utilisées dans l'environnement Cloud.

### 2.6 Travaux connexes sur la tolérance aux pannes dans le Cloud

Beaucoup de travaux ont été fait dans le domaine de la tolérance aux pannes dans les infrastructures du Cloud Computing, dont il y a beaucoup de place de recherche disponible dans ce domaine jusqu'à ce moment-là. Les infrastructures du Cloud ont introduit des nouvelles questions liées à la tolérance aux pannes. Dans ce contexte on va exposer quelques travaux similaires.

Kong et. Coll, ont présenté un modèle de tolérance aux pannes et de performance dans les infrastructures virtuelles. Mais il n'est pas bien adapté à la tolérance aux pannes des applications dans le Cloud. [37]

Ravi Jhavar, Vincenzo Piuri, Marco Santambrogio, dans leurs article « A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing », décrit le Framework FTM (Fault Tolerance Manager). Il présente la gestion des techniques de fiabilité pour les utilisateurs dans laquelle il peut spécifier et appliquer le niveau désiré de tolérance aux panne sans avoir besoin d'aucune connaissance sur sa mise en œuvre à base sur une couche de service utilisateur dédié. [38]

Une autre approche, utilise un middleware tolérante aux pannes qui est basé sur la réplication a été proposée par W. Zhao, P. M. Melliar-Smith, et L. E. Moser, dans leur article « Fault Tolérance Middleware for Cloud Computing ». [39]

Dans « Approach for constructing a modular Fault tolerant protocol » l'article écrit par M.Hiltunen & R.D. Schlichting, propose des protocoles modulaires et en les combinant à un système à l'aide de techniques hiérarchiques. [40]

Une autre étude vise à fournir une meilleure compréhension des problèmes de la tolérance aux pannes et identifie les différents outils et techniques utilisés pour la tolérance aux pannes. Lorsque plusieurs instances d'une application sont en cours d'exécution sur plusieurs MVs et l'un des serveurs tombe en panne, il est nécessaire de mettre en œuvre une technique de tolérance aux pannes qui peut gérer ces types de pannes. Pour résoudre ce problème, Anju Bala<sup>1</sup>, Inderveer a proposé et mis en œuvre la HAProxy. Dans l'article « Fault Tolérance-Challenges, Techniques and Implémentation in Cloud Computing ». [41]

Rajesh.S, KannigaDevi.R, dans leur article « Improving Fault Tolerance in Virtual Machine Based Cloud Infrastructure ». La méthode qu'ils ont proposée utilise deux ensembles différents de nœuds, l'un est l'ensemble des MVs et l'autre est le nœud d'arbitrage principal (nœud serveur). Le serveur contient le temps checker qui est l'évaluateur de la fiabilité et le mécanisme de décision pour trouver la MV fiable qui s'occupera pour traiter la demande du client. Pour fournir la tolérance aux pannes, les données peuvent être stockées sur de multiples Cloud utilisant la technique de virtualisation. [42]

### 2.7 Conclusion

Dans ce chapitre, nous avons dressé un état de l'art sur la sûreté de fonctionnement, après nous avons donné un aperçu général sur les classifications des pannes. Par la suite, nous avons montré les deux politiques principales (proactive et réactive) de la tolérance aux pannes et quelques techniques utilisées. Nous avons terminé par certains travaux connexes sur cette notion dans l'environnement Cloud.

Notre but dans ce travail est de garantir la tolérance aux pannes réactive dans l'environnement Cloud Computing. Pour cela nous présentons dans le prochain chapitre une technique puissante appelée, RAID. Cette dernière est utilisée dans un large nombre de système d'exploitation pour assurer la tolérance aux pannes.

## Chapitre 3 :



# RAID

Redundant Array of Independent Disks

### 3.1 Introduction

Nous avons parlé brièvement dans le chapitre précédent sur les différentes techniques utilisées pour assurer la tolérance aux pannes, où chacune a ses propres caractéristiques. D'autre part, nous avons vu la puissance de la technologie RAID non seulement dans le contexte de tolérance aux pannes mais aussi dans le domaine d'optimisation de l'espace de stockage.

Le RAID est avant tout une description de la manière dont il est possible de stocker les données sur différents disques durs et de les utiliser comme s'il s'agissait d'un seul et unique disque. Il en existe plusieurs types, dits matériels ou logiciels. RAID est une technologie largement utilisée dans l'industrie informatique, le stockage et la maintenance des serveurs puisqu'il est assez standard dans la plupart des serveurs et les périphériques ce qui en fait un élément très utile d'une maintenance adéquate. Le RAID est principalement utilisé en raison de ses bonnes performances et de sa capacité de tolérance aux pannes.

Il existe une grande variété de niveaux RAID, chacun avec sa propre application idéale et sa propre méthode de tolérance aux pannes. La tolérance aux pannes consiste à aider les systèmes à minimiser les pertes de donnée qui peuvent être des données critiques et importantes (des données des serveurs, BD ...etc.).

### 3.2 Définition et historique

Le RAID (Redundant Array of Inexpensive Disks ou Redundant Array of Independent Disks) est une technique qui consiste à utiliser plusieurs disques durs simultanément en les faisant apparaître comme un seul. Le but est d'accroître la performance ainsi que la vitesse pour traiter plus de données en parallèle par l'ajout d'informations redondantes, la sécurité et la tolérance aux pannes.

Le concept du RAID fut lancé en 1987 à l'université de Berkeley. Ces configurations ont été numérotées de 0 à 5. On les désigne par des "niveaux RAID" ou "RAID levels": RAID0 à RAID5.

D'autres architectures RAID sont apparues par la suite (6 e 7) et des constructeurs proposent des RAID à plusieurs niveaux combinés: 1+0, 5+0 ... etc. [43]

### 3.3 Types de RAID

#### 3.3.1 Le RAID logiciel

Le RAID logiciel (Figure 18), est une solution logicielle qui dépend du système d'exploitation et utilise les ressources comme la mémoire et CPU du système, il s'agit d'un driver qui permet de

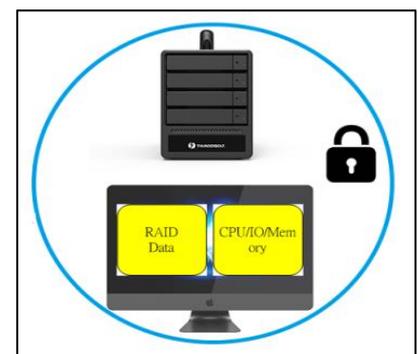


Figure 18 RAID logiciel [46]

construire des disques durs virtuels, et les combinent en un seul disque logique, mais cela permet de provoquer quelques inconvénients par exemple : la charge du CPU (overhead). [43]

### 3.3.2 Le RAID matériel

Le RAID matériel (Figure 19) est utilisé plus dans les industries, cette solution se fait au niveau du matériel indépendamment du système d'exploitation, à l'aide d'une carte contrôleur spécifique, équipée la plupart du temps d'un BIOS : cette carte va présenter au système d'exploitation les arrays (matrices) enregistrés dans sa mémoire comme étant des disques classiques : toute la charge due aux algorithmes sera donc cachée au CPU de la machine et sera traitée par la carte contrôleur. Ainsi, on a un système totalement indépendant et transparent pour le soft. C'est un système que l'on retrouve très communément dans les entreprises dans les baies des serveurs de fichiers. [44]

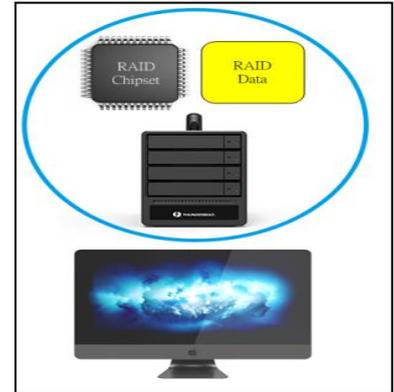


Figure 19 RAID matériel [46]

En comparant les deux types de RAID:

- **RAID logiciel**

+ Prise en charge minimale (pas de besoin d'autres supports matériels par exemple, les cartes contrôleurs, les disques durs sont virtualisés, donc pas de crash de contrôleur matériel ainsi que la possibilité de récupérer les disques sur tout système indépendamment du matériel installé...);

+ Pas de coûts supplémentaires;

- Faible performance du système;

- Surcharge élevée de processeur.

- **RAID matériel**

- Largement pris en charge;

- Coûteux;

+ Hautes performances;

+ Faible surcharge du processeur.

## 3.4 Les concepts du RAID

La technologie RAID dans le traitement des données est divisée en trois façons (Striping, Mirroring et Parity), chacune ayant un effet sur la performance et la façon dont les données sont conservées.

### 3.4.1 Stripage (Striping)

La baie de stockage (matrice RAID) qui contient plusieurs disques divise le flux de données en petites unités avec des zones spécifiques, puis les écrit une par une sur chaque disque, ce qui augmente les performances des disques de stockage de cette manière.

### 3.4.2 Miroir (Mirroring)

La baie de stockage stocke une copie des données sur tous les disques à la fois, ce qui augmente les performances des disques de stockage, en plus d'éviter les erreurs qui peuvent survenir sur ces disques pendant le fonctionnement.

### 3.4.3 Parité (Parity)

Il permet aux disques de stockage de traiter les données comme un seul disque, tandis que pendant que l'un d'eux est perdues, c'est-à-dire que l'un des disques cesse de fonctionner, les données perdues sont recalculées à partir des autres et puis redistribuées sur un nouveau disque qui va prendre le relais ou sur lui-même après sa maintenance, ça dépend le cas.

## 3.5 Les niveaux de RAID

Ils existent plusieurs modes (niveaux) de RAID, dont chaque niveau répondre à des besoins spécifiques, ici nous allons essayer de citer quelques-uns qui sont les plus connus et utilisés :

### 3.5.1 RAID0 (Striping)

Le RAID0, également connu sous le nom d'« entrelacement de disques » ou de « volume agrégé par bandes » (striping en anglais), est une configuration RAID permettant d'augmenter significativement les performances de la grappe en faisant travailler « n » disques durs en parallèle (avec  $n \geq 2$ ). La Figure 20 montre la répartition des données sur un RAID0.

Le RAID0 est basé sur les principaux points suivants : [45]

- ✓ Striping au niveau du bloc sans parité;
- ✓ Nécessite 2 disques durs au minimum ;
- ✓ Les données sont écrites sur chaque disque dans un ordre séquentiel ;

- ✓ Pas de redondance, si un disque tombe en panne, toutes les données sont perdues donc pas sécurité ;
- ✓ Très haute performance, puisque cette technique augmente la vitesse de transferts ;
- ✓ Idéal lorsque la performance est la seule préoccupation.

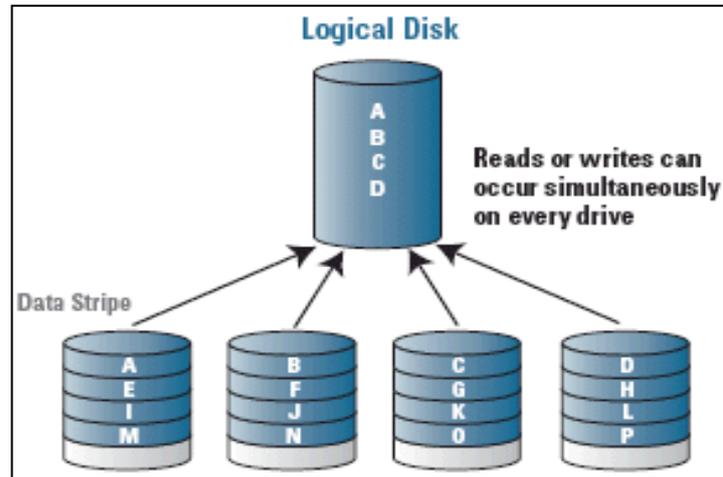


Figure 20 RAID0 [46]

### 3.5.2 RAID1 (Mirroring)

Le RAID1 consiste en l'utilisation de «n» disques redondants (avec  $n \geq 2$ ), chaque disque de la grappe contenant à tout moment exactement les mêmes données, d'où l'utilisation du mot « miroir ». La Figure 21 montre la répartition des données sur un RAID1.

Les principales notions de RAID1 sont les suivantes: [45]

- ✓ Mirroring au niveau du bloc (redondant);
- ✓ Nécessite 2 disques durs au minimum;
- ✓ Les données sont écrites sur les deux disques;
- ✓ Redondant, si un disque tombe en panne, le système continue de fonctionner ;
- ✓ Des performances en lecture et en écriture plus lentes que celles du RAID0;
- ✓ Idéal lorsque la tolérance aux fautes est une priorité et que les performances peuvent être compromises;
- ✓ Permet une défaillance d'un seul disque.

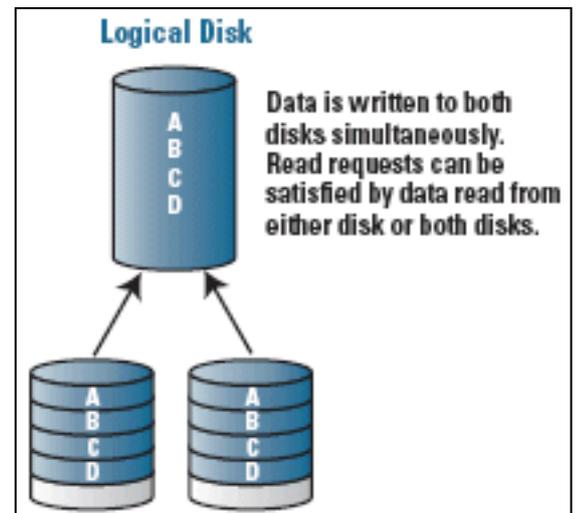


Figure 21 RAID1 [46]

### 3.5.3 RAID5 (Striping with distributed parity)

Le RAID5 combine la méthode du volume agrégé par bandes (striping) à une parité répartie. La parité, se retrouve répartie circulairement sur les différents disques. Chaque bande est donc constituée de «n» blocs de données et d'un bloc de parité (ou bloc de données de contrôle). Ainsi, en cas de panne de l'un des disques de la matrice RAID, pour chaque bande il manquera soit un bloc de données soit le bloc de parité. Si c'est le bloc de parité, ce n'est pas grave, car aucune donnée ne manque. Si c'est un bloc de données, on peut calculer son contenu à partir des «n-1» autres blocs de données et du bloc de parité. L'intégrité des données de chaque bande est préservée. Donc non seulement la grappe est toujours en état de fonctionnement. La Figure 22 illustre la répartition des données ainsi que les données de parités (obtenues par l'opération xor) sur une matrice RAID5.

Le RAID5 repose sur les principaux points suivants: [45]

- ✓ Striping au niveau des blocs avec parité répartie (redondant) ;
- ✓ Nécessite 3 disques durs au minimum ;
- ✓ Les données sont écrites sur les trois disques ;
- ✓ Redondance en cas de défaillance d'un disque, le système continue de fonctionner ;
- ✓ Le meilleur des deux modes (RAID0 & RAID1): excellentes performances en lecture et en écriture avec redondance ;
- ✓ Idéal lorsque la tolérance aux pannes et les performances de lecture/écriture du disque sont requises;
- ✓ Permet la défaillance d'un seul disque.

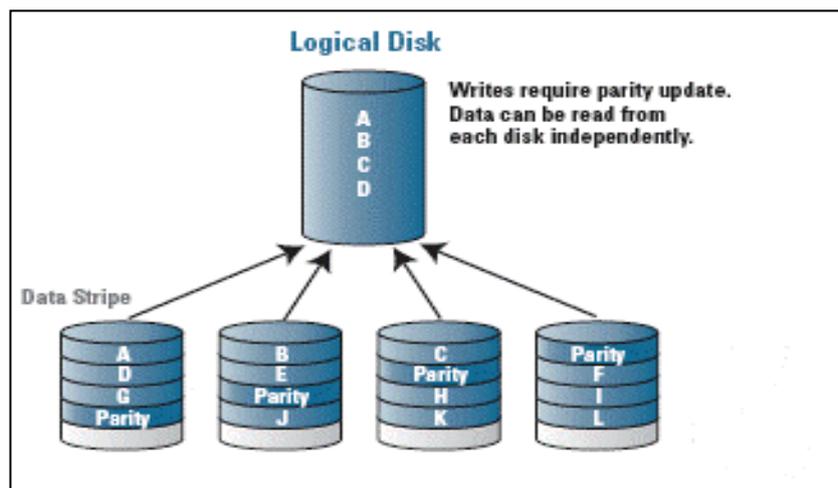


Figure 22 RAID 5 [46]

Comme nous avons dit, bien que les fichiers stockés sur le RAID5 restent intacts en cas de défaillance d'un disque dur, des données peuvent être perdues si un deuxième disque dur tombe en panne avant que le RAID ne soit reconstruit avec le disque dur de remplacement.

Le RAID5 offre des performances qui peuvent s'approcher du RAID0. Le grand avantage du RAID5 par rapport au RAID0 est la protection des données. De plus, il vous reste environ 75% de la capacité de stockage d'une baie RAID0 (en fonction du nombre total de disques durs et des capacités de stockage disponibles). [47]

### 3.5.4 RAID6 (Striping with dual parity)

Le RAID6 est un RAID à double parité P et Q. L'objectif de double parité est d'aider le RAID6 à résister à une double panne de deux disques durs simultanément. Par contre, cette double parité affecte la performance en écriture. Une étude sur le RAID6 a constaté une chute de performance de l'ordre de 30% par rapport au RAID5. La Figure 23 illustre la répartition des données ainsi que les données de parités sur un RAID6.

Le RAID6 est basé sur les principaux points suivants: [45]

- ✓ Striping au niveau des blocs avec deux bits de parité répartis (généralement on les appelle q et p);
- ✓ Nécessite 4 disques durs au minimum;
- ✓ Les données sont écrites sur les quatre disques ;
- ✓ Permet la défaillance de 2 disques;
- ✓ Idéal lorsque la tolérance aux pannes et les performances de lecture/écriture du disque sont requises;

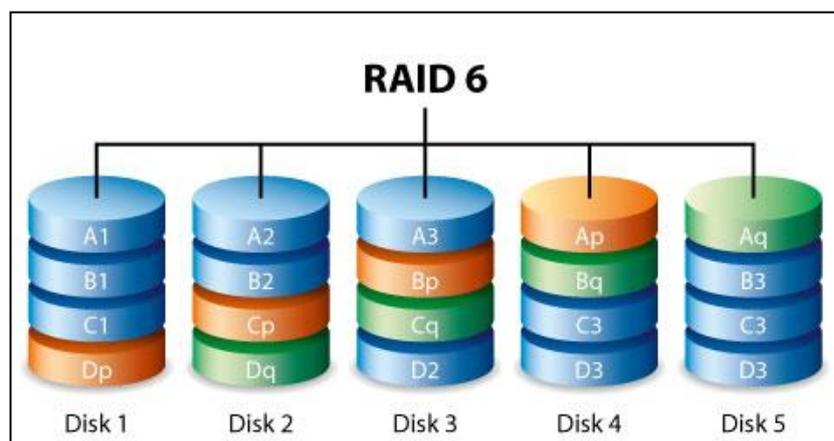


Figure 23 RAID6 [63]

Comme nous l'avons dit auparavant que certains constructeurs proposent des combinaisons pour les systèmes RAID, nous mentionnerons quelques modes les plus connus :

### 3.5.5 Le RAID 1+0 (RAID10)

RAID10 (appelé aussi Striped-Mirror) permet d'obtenir un volume agrégé par bande avec une bonne fiabilité (puisque basé sur des grappes répliquées). Chaque grappe contenant au minimum deux éléments et un minimum de deux matrices de disques étant nécessaire, il faut au minimum quatre unités de stockage pour créer un volume RAID10. La Figure 24 montre la répartition des données sur un RAID10.

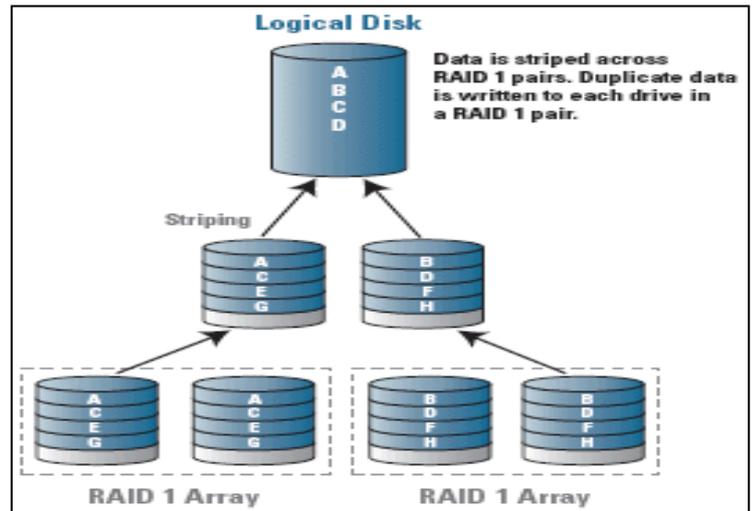


Figure 24 RAID10 [46]

Le RAID10 est basé sur les principaux points suivants: [45]

- RAID10 fait le Mirroring après le Striping ;
- Minimum de 4 disques car les données doivent passer par deux étapes : le Mirroring et le Striping;
- Le RAID10 permet une bonne tolérance aux pannes - tout disque peut fonctionner tant qu'il a une copie de lui (le mirroring).

### 3.5.6 Le RAID 5+0 (RAID50)

Le RAID50 combine le striping du RAID0 avec la parité du RAID5. En raison de la vitesse de stripage du RAID0, le RAID50 améliore les performances du RAID5, en particulier lors de l'écriture. Il offre également plus de protection qu'un seul niveau de RAID. Un minimum de six disques durs est nécessaire pour une le RAID50 [47] La Figure 25 montre la répartition des données sur un RAID50.

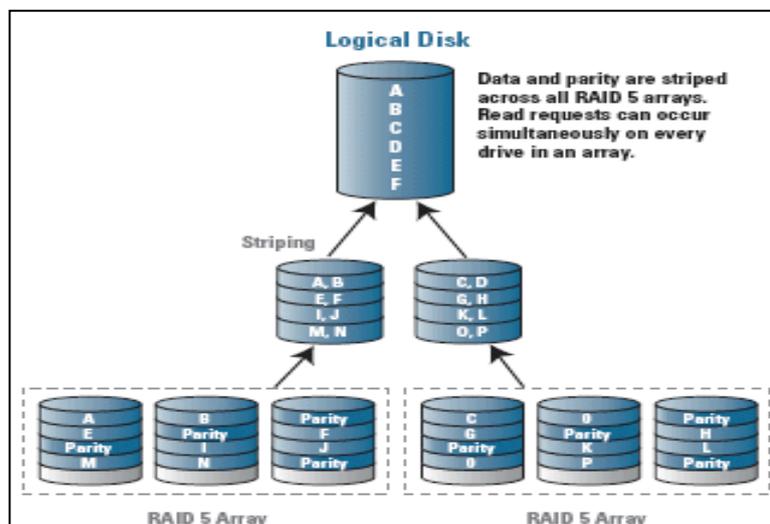


Figure 25 RAID50 [46]

### 3.5.7 Le RAID 6+0 (RAID60)

Le RAID60 combine le RAID0 à bandes (qui fait le Striping) et le RAID6 à double parité. En raison de la rapidité du striping du RAID0, le RAID60 améliore les performances du RAID6. Il offre également plus de protection qu'un seul niveau de RAID. Un minimum de huit disques durs est nécessaire pour un RAID60 [47] La Figure 26 montre la répartition des données sur un RAID60.

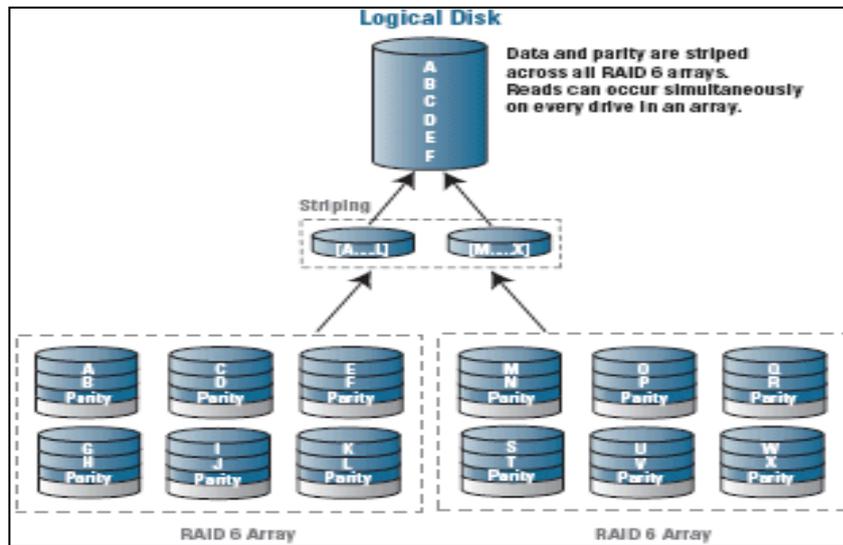


Figure 26 RAID 60 [46]

Dans le tableau3, nous citons quelques caractéristiques comme la performance, le coût et le niveau de protection des niveaux RAID les plus couramment connus et utilisés.

RAID	Nombre minimum de disques	Protection des données	Performance en lecture	Performance en écriture	Coût
0	2	Aucune	Haute	Haute	€
1	2	Un seul disque en panne	Haute	Moyenne	€€€€
5	3	Un seul disque en panne	Haute	Faible	€€
6	4	Deux disques en panne	Haute	Faible	€€€

Tableau 3 Caractéristiques de certains niveaux de RAID [48] [61]

## 3.6 Avantages de RAID

Bien que chaque niveau de RAID a des avantages mais aussi a des inconvénients. Mais puisque cette technologie a plus d'avantage que d'inconvénient, dans cette section nous allons citer quelques-uns: [49]

### 3.6.1 Tolérance aux pannes

C'est le principal avantage de cette technologie qui réside dans certains niveaux, qui permette aux disques durs de survivre aux nombreux problèmes auxquels sont confrontés (disque qui tombe en panne, ...), donc à minimiser les pertes de donnée dans les disques durs.

### 3.6.2 Performance

Haute performance et une grande vitesse de lecture et d'écriture si on compare un disque dur parmi l'ensemble de disques qui utilisent RAID par rapport à un autre disque dur qui ne l'utilise pas.

### 3.6.3 Capacité

Espace de stockage, qui vous permet de collecter des disques séparés de petites entités en une seule entité avec un grand espace de stockage, ce qui est la fonctionnalité de base que nous avons mentionnée précédemment.

## 3.7 Conclusion

Notre travail consiste à garantir la tolérance aux pannes dans le Cloud, mais malheureusement les défaillances des machines virtuelles ou physiques sont très probables dans cet environnement, donc la nécessité de disposer d'une solution qui permette à la fois de récupérer les données après une panne et d'optimiser l'espace de stockage consommé. C'est pour cela, dans ce chapitre, nous avons proposé une brève revue de la littérature sur une fameuse technologie dans le monde de la tolérance aux pannes qui est le RAID ainsi que ses différents niveaux.

Dans le prochain chapitre nous présentons la partie conception de notre solution proposée, où nous allons adapter le RAID niveau5 - présenté dans ce chapitre - dans l'objectif d'assurer une meilleur tolérance aux pannes dans un environnement Cloud.

**PARTIE 2 :**

**CONCEPTION,  
IMPLEMENTATION ET  
SIMULATION DE LA  
SOLUTION PROPOSEE**

## **Chapitre 4 :**

# **CONCEPTION DE LA SOLUTION PROPOSEE**

## 4.1 Introduction

La réalisation de notre contribution doit être impérativement précédée d'une conception qui a pour objectif de formaliser les étapes préliminaires de son développement. Pour la conception du système proposé, nous allons utiliser le langage UML ainsi que quelques algorithmes permettent de décrire le processus de fonctionnement de notre approche proposée, appelée « CRAID5 ».

## 4.2 Description et architecture de l'approche proposée

### 4.2.1 Description

On se focalise dans ce travail sur la tolérance aux pannes réactive dans un environnement Cloud, en prenant en compte l'espace de stockage qui doit être optimisé.

Pour cela, nous proposons CRAID5, une politique de gestion de tolérance aux pannes réactive basée sur un niveau spécifique d'un concept puissant connu sous le nom de RAID décrit dans le chapitre précédent, qui est le «RAID5» que nous l'avons présenté en détails dans la Section 4.2.3 de ce chapitre. Notre approche permet de récupérer les ressources perdues de Cloud tel que les MPs et les MVs. La contribution majeure de la solution proposée réside dans l'optimisation d'espace du stockage en assurant le service de tolérance aux pannes.

### 4.2.2 Architecture de système

La Figure 27 décrit l'architecture du système proposé ainsi que les interactions entre ses différentes entités.

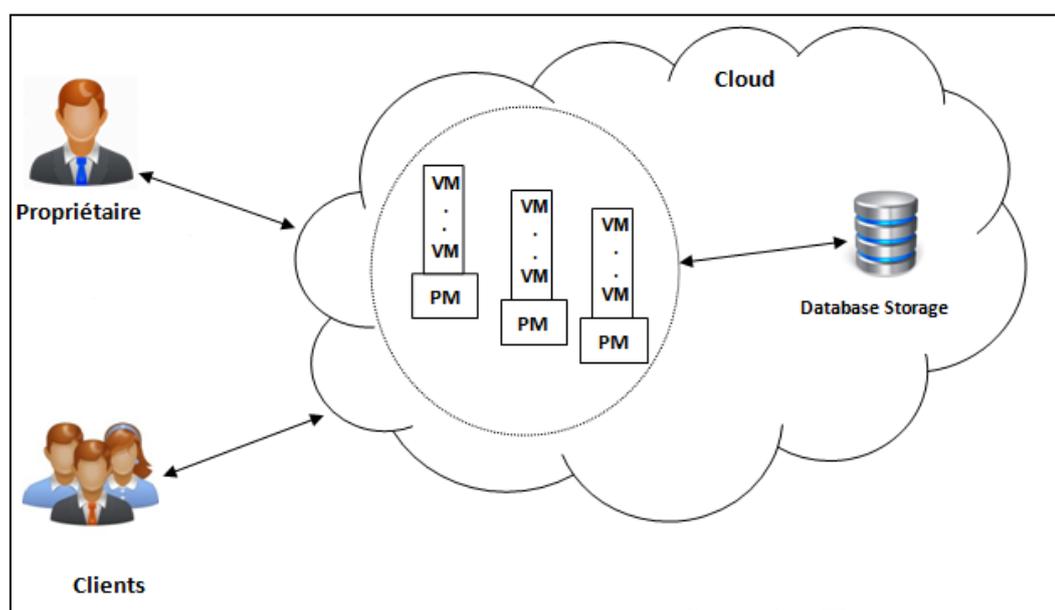


Figure 27 Architecture du système

Notre système sera constitué des entités suivantes :

- Un ensemble de ressources stocké sous forme de fichiers (File Storage).
- Une BD, qui contient toutes les informations sur les utilisateurs ainsi que des informations sur les ressources créés.
- Un ensemble d'utilisateurs (administrateur ou propriétaire de Cloud et les clients).

Pour mieux décrire et comprendre la solution proposée, on va présenter quelques notions importantes sur le RAID5, notamment ce qu'est concerne la création des backups, la récupération de données... etc.

### 4.2.3 L'utilisation de RAID5 et la reconstruction des données

Afin de satisfaire le critère de tolérance aux pannes nous proposons l'utilisation de RAID5, qui est une solution de stockage qui tient compte la tolérance aux pannes. Prenez un RAID5 composé de 4 disques par exemple. P0 est l'information de parité de D0, D1 et D2 et ainsi de suite. Il est supposé que le RAID5 ne sauvegarde pas seulement les données, mais stocke les données et les informations de parité correspondantes dans chaque disque du RAID5, elles sont stockées de manière distincte dans différents disques. Lors de la défaillance d'un disque, une opération logique à partir des blocs de parité permet de régénérer les blocs de données perdus correspondants.

Selon le principe du striping, les données sont segmentées en blocs et réparties en bande sur l'ensemble des disques durs. Chaque bande comporte en outre un bloc de parité, dont la valeur est calculée (par une opération logique XOR ou OU exclusif) à partir des autres blocs de donnée de la bande en question comme illustré dans la Figure 28.

En général, Pour un RAID5 constitué de N disques durs, chaque bande est donc constituée d'un bloc de parité et de N-1 blocs de données. En outre, les blocs de parité sont répartis sur l'ensemble des disques durs du système afin de ne pas être concentrés sur un seul d'entre eux.

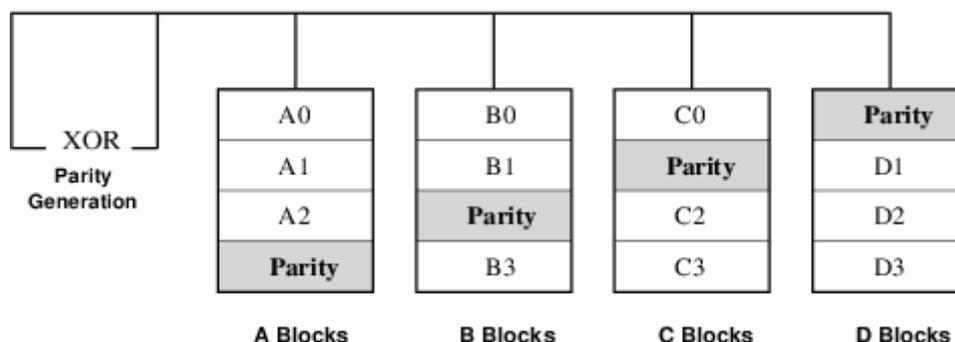


Figure 28 RAID5 et le principe de parité

- **Principe de l'algorithme de RAID5**

$P0 = D0 \text{ XOR } D1 \text{ XOR } D2 \dots \text{ XOR } Dn$  ( $D0, D1, D2 \dots Dn$  sont des blocs de données,  $P0$  est la parité, XOR est l'opération logique).

Le principe de l'algorithme de parité du XOR (Exclusive OR) est présenté dans le tableau4:

Valeur de A	Valeur de B	Valeur de XOR
0	0	0
1	0	1
0	1	1
1	1	0

Tableau 4 Tableau de vérité de l'opération XOR

La valeur A et la valeur B correspondent à deux bits. On peut trouver si A est identique à B, et le résultat XOR est 0 et si A est différent de B, le résultat XOR est 1. Si le résultat XOR et A ou B sont connus, l'autre valeur peut être calculée. Par exemple, si A est égal à 1 et que le résultat XOR est 1, alors B doit être égal à 0. Et si A est égal à 1 et que le résultat XOR est 0, alors B doit être égal à 1. C'est le principe de base du codage XOR et de la parité.

Donc, Les informations de parité celles qui permettent la reconstruction de données perdues ou endommagées.

- **Le stockage des données dans RAID5**

Une recommandation majeure de RAID5 consiste à utiliser des disques de même taille, si non le RAID va traiter tous les disques de la même manière comme s'il s'agissait du plus petit disque.

- **Exemple 1:** 10 disques durs de 1 To correspondent à 9 To de stockage.
- **Exemple 2:** 9 disques de 1 To et un disque de 500 Go représentent 4,5 To de stockage (chaque 1 To est traité comme un 500 Go).

## 4.3 Processus de l'approche proposée et les algorithmes utilisés

### 4.3.1 Processus de fonctionnement

La technique RAID5 recommande d'utiliser des disques de même taille avec un nombre total de disques utilisés qui doit être d'au moins trois.

Cette recommandation est prise en compte dans notre architecture, en divisant les ressources du Cloud en ensembles (sets en anglais) des ressources de tailles différentes, qui vont être divisés en sous-ensembles (bande ou sub-sets en anglais) de même taille, où une MV ou MP est l'équivalent d'un disque dans la solution RAID de base. Deux types de ressources sont pris en compte dans l'architecture de système. Le premier est le type de ressource MV, tandis que le second est le type de ressource MP.

L'approche proposée est composée de trois fonctionnalités essentielles :

- Ajout de ressources.
- Suppression de ressources.
- Récupération de ressources.

#### ❖ Ajout de ressources

Lorsqu'une nouvelle MV ou MP est ajoutée à l'ensemble des ressources du Cloud, l'événement d'ajout est déclenché par le système qui va commencer par attribuer un identifiant R unique à la nouvelle ressource et enregistre toutes les informations concernant la nouvelle ressource ajoutée dans une BD. Il utilise ensuite toutes les informations sauvegardées afin de trouver le sous-ensemble le mieux adapté (SEid), et intègre ensuite la nouvelle ressource dans ce sous-ensemble. Enfin, un tuple (RType; Rid; SEid) est construit comme suit:

**RType:** Représente le type de ressource avec plusieurs valeurs possibles. Par exemple, une valeur de type peut être MP pour représenter MP, MVSmall pour représenter une petite MV, ou MVLarge pour représenter une large MV, etc.

**Rid:** Représente un identifiant unique qui permet d'identifier une ressource dans le système.

**SEid:** Représente un identifiant unique utilisé pour identifier le sous-ensemble qui comprend Rid.

#### ❖ Suppression de ressources

La suppression de MV ou MP est un autre événement qui peut apparaître. Comme un événement d'ajout de MV/MP, quand l'événement de suppression de MV/MP est détecté par le système, il identifie la ressource supprimée par son identifiant, le retire de son sous-ensemble et vérifie si la taille du sous-ensemble après l'enlèvement est inférieur à trois. Si tel est le cas, il supprime l'ensemble du sous-ensemble, enfin, il met à jour la BD.

### ❖ Récupération de ressources

Le troisième événement qui peut apparaître est la panne des ressources. Une ressource est considérée comme une ressource en panne lorsqu'elle ne répond pas aux demandes des utilisateurs, ce qui peut généralement se produire si la ressource n'est pas accessible.

L'approche CRAID5 proposée suit le RAID concept. Chaque fois qu'on a un échec de ressource, CRAID5 ne cherchera pas à copier de la MP/MV perdue. Il reconstruira plutôt la copie perdue de la ressource du sous-ensemble qui contient MV/MP. Dans CRAID5, les ressources sont codées par une séquence de bits qui représentent les données des utilisateurs.

En supposant qu'un sous-ensemble  $SE_i$  contient des ressources  $p$  de même taille noté  $(R_1; R_2 ; \dots; R_p)$ , avec  $p$  égal ou supérieur à trois et  $R_p$  représentent les valeurs de parité comme indiqué en (1)

$$R_p = R_1 \oplus R_2 \oplus R_3 \oplus \dots \oplus R_{p-1} \quad (1)$$

Pour la phase de reconstruction, elle se fait avec un simple XOR entre toutes les ressources qui composent le sous-ensemble correspondant, sauf  $R_p$ . Après la reconstruction de la ressource, le système mettra à jour la BD.

### 4.3.2 Description des algorithmes utilisés

#### ❖ Ajout de ressources

L'ajout est fait par la recherche de sous-ensemble qui ne contient aucune MV hébergée dans la même MP avec la nouvelle MV (c'est-à-dire, chercher un  $SE_i$  qui ne contient aucune MV dans la MP qui va héberger la nouvelle MV), si aucun sous-ensemble n'est trouvé, un nouveau sous-ensemble est créé la nouvelle MV est ajoutée à ce sous-ensemble.

L'ajout de ressources est fait selon le pseudo algorithme suivant illustré dans la Figure 29:

---

**Algorithme 1 Ajout de ressource**


---

```

1 : Entrée : PMj, VMi ;
2 : Sortie: SE_id ;
3 : Pour SEi ∈ SE Faire
4 :     SB : Pour VM ∈ SEi Faire
5 :         Si VM est hébergé dans PMj Alors
6 :             break SB ;
7 :     Fin.
8 : Fin.
9 : Si (SEi == True) //SEi est trouvé
10 :    Ajouter VMi au SEi ;
11 : Sinon
12 :    Créer nouveau SE ;
13 :    Ajouter VMi au nouveau SE ;
14 : Fin.

```

---

Figure 29 Pseudo algorithme "Ajout de ressources"

#### ❖ Récupération de ressources

Deux cas de panne peuvent apparaître, le premier consiste en une panne MP, tandis que le second consiste en une panne du MV.

Dans le premier cas, toutes les MVs qui ont été hébergées sur la MP défaillante sont identifiées et l'ensemble des MVs est reconstruit de la même manière que dans le cas d'une panne de MV. Dans le second cas, la MV perdue est reconstruite en utilisant le processus de récupération défini auparavant.

La Figure 30 présente l'algorithme utilisé pour récupérer les MVs ou MPs une fois qu'une panne s'est produite.

---

**Algorithme 2 Récupération de ressources**


---

```

1 : Entrée : R1, R2,..... , Rp ;
2 : Sortie: RPerdue ;
3 : Si (RPerdue est PM) Alors
4 :     Pour Ri ∈ PM Faire
5 :         Reconstruire(Ri) ;
6 :     Fin
7 : Sinon
8 :     Reconstruire(RPerdue) ;
9 : Fin.
10 : Fonction Reconstruire(RPerdue)
11 :     Trouver SE_id dont RPerdue ∈ SE_id
12 :     Reconstruire RPerdue ;
13 : Fin Fonction.

```

---

Figure 30 Pseudo algorithme "Récupération de ressources"

## 4.4 Conception des diagrammes

Pour décrire les différents besoins fonctionnels de notre système, on va faire une conception avec des diagrammes afin d'obtenir une vue globale sur les exigences de notre système. Nous avons opté pour le langage UML pour plusieurs raisons:

- Langage servant à décrire des modèles d'un système (réel ou logiciel) basé sur des concepts orientés-objets ;
- Système de notations pour modéliser les systèmes en utilisant des concepts orientés objets ;
- Représente un standard de modélisation, une notation : il s'agit donc d'un outil et non d'une méthode ; chacun peut utiliser la méthode la plus appropriée à ses besoins ;
- Il permet de représenter l'aspect traitement du système aussi bien que l'aspect donné.

Dans ce chapitre, nous présentons la conception des différents éléments du système réalisé. Nous utilisons pour cela les diagrammes suivants:

- Diagramme de cas d'utilisation.

- Diagramme de séquence.
- Diagramme de classes.

Dans notre système, nous pouvons distinguer les acteurs (rôles) suivants:

**1- Administrateur** : c'est le propriétaire de Cloud (fournisseur de services) qui gère les MPs et les MVs des clients ainsi que leurs comptes.

**2- Client** : c'est le demandeur de service qui peut gérer ses propres MVs.

#### 4.4.1 Diagramme cas d'utilisation globale

Un diagramme de cas d'utilisation permet de décrire l'interaction entre l'acteur et le système. Un acteur représente un rôle joué par une entité externe (utilisation humaine, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Le diagramme cas d'utilisation global montré dans la Figure 31 regroupe toutes les fonctionnalités qui peuvent être établies par l'utilisateur du système. Il est suivi d'une description détaillée pour chacune d'elle, illustrée dans le tableau5.

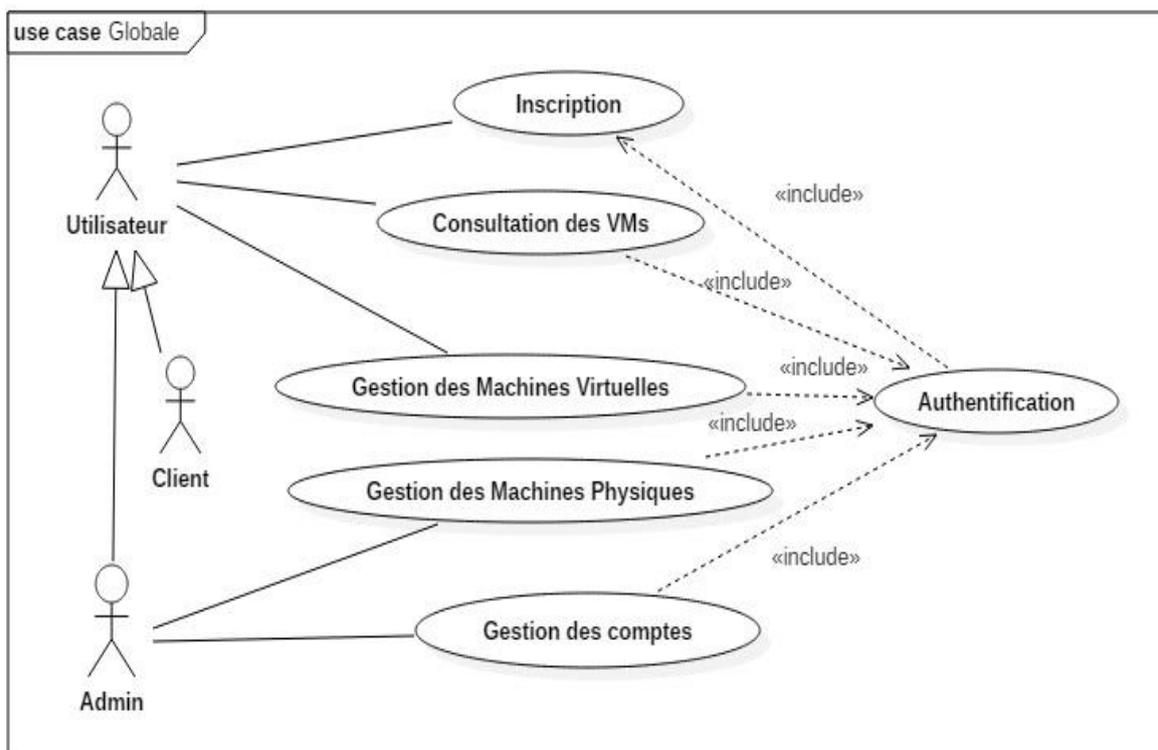


Figure 31 Diagramme cas d'utilisation global.

Cas d'utilisation	Acteur	Description des cas d'utilisation
Inscription	Utilisateur	Chaque utilisateur peut s'inscrire au système.
Authentification	Utilisateur	Chaque utilisateur peut s'authentifier au système afin d'accéder à son propre espace.
Gestion des comptes	Administrateur	Permet d'ajouter ou supprimer un compte client.
Gestion des machines physiques	Administrateur	Créer, supprimer ou récupérer une machine physique.
Gestion des machines virtuelles	Client	Permet de créer, supprimer ou récupérer ses propres machines virtuelles.
	Administrateur	Permet de créer, supprimer ou récupérer des machines virtuelles pour n'importe quel client.
Consultations des machines virtuelles	Client	Le client peut consulter toutes ses propres machines virtuelles qui sont déjà créées auparavant.
	Administrateur	Permet à l'administrateur de consulter toutes les machines virtuelles de tous les clients.

Tableau 5 Description des cas d'utilisation

#### 4.4.2 Diagramme de séquence

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Dans ce qui suit, nous allons détailler les principales fonctionnalités dans notre système :

- 1- Gestion de MPs,
- 2- Gestion de MVs.

##### A. Diagramme de séquence « Gérer les machines physiques »

Les MPs sont des éléments essentiels dans le système pour qu'il soit sous un environnement Cloud, donc elles nécessitent une gestion. Elles sont gérées par le propriétaire qui représente l'administrateur dans notre système.

Les diagrammes suivants présentés dans les Figures suivantes : Figure 32, Figure 33 et Figure 34, montrent les principales fonctions dans la gestion des MPs (ajout, suppression et récupération) respectivement.

❖ Ajout de MPs

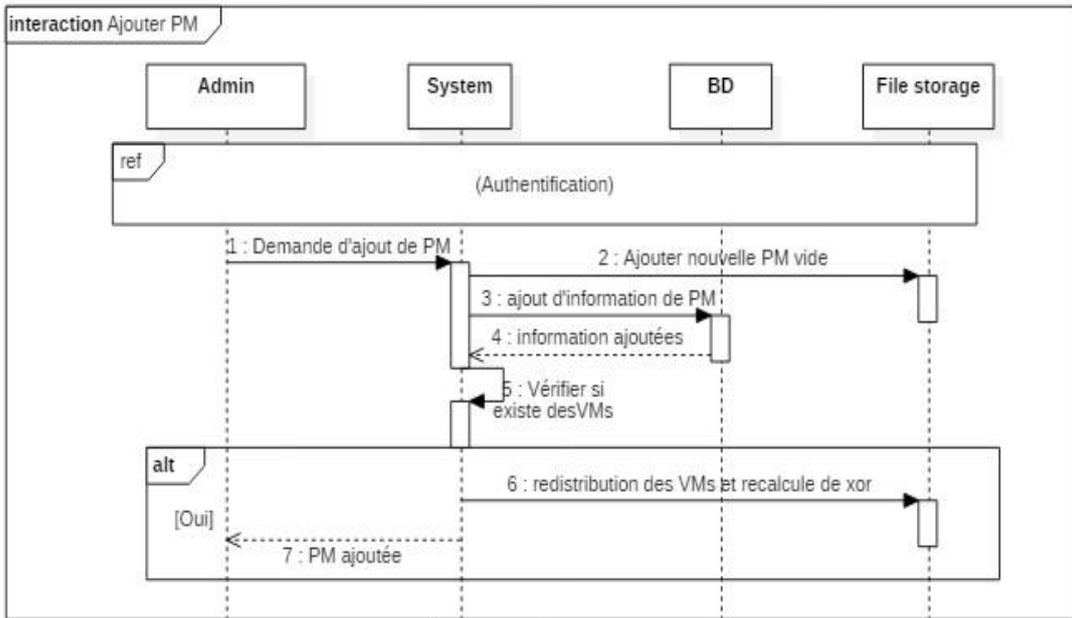


Figure 32 Diagramme de séquence " Ajouter MP"

❖ Suppression de MPs

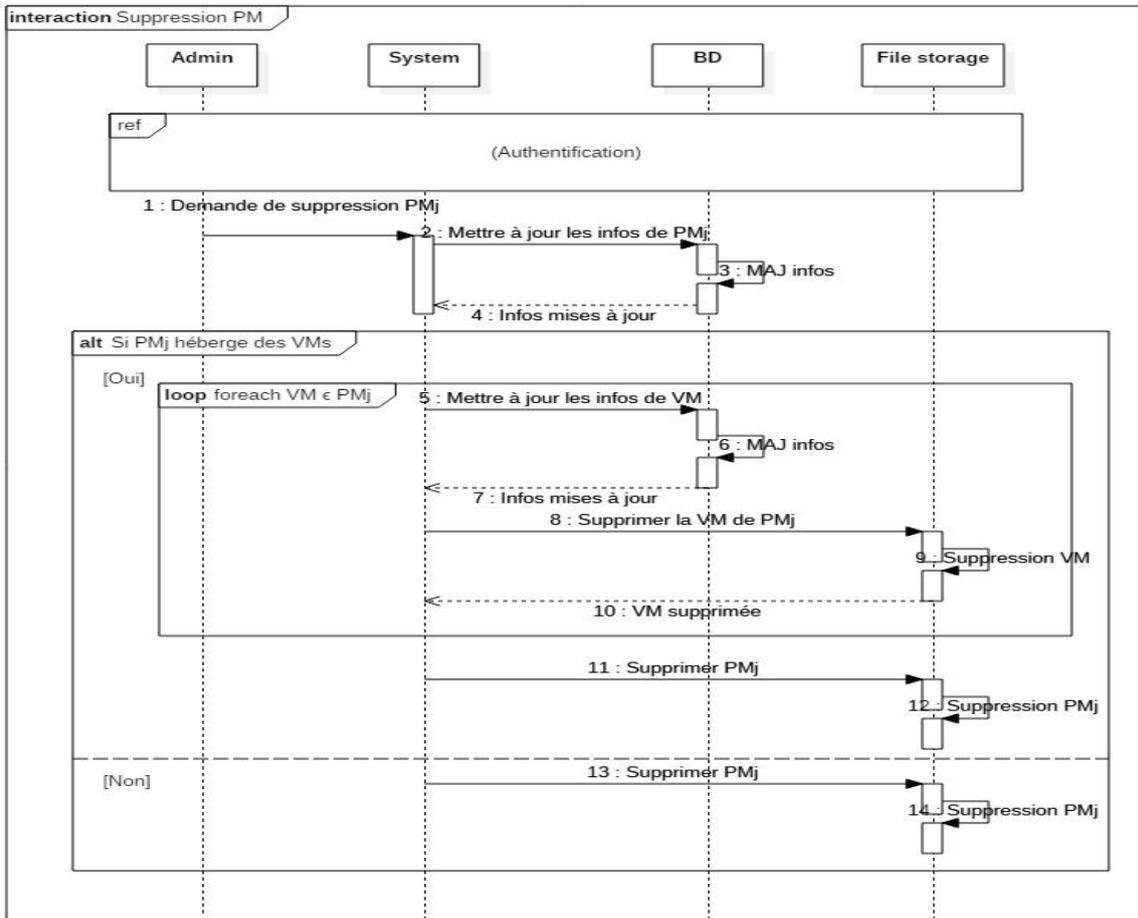


Figure 33 Diagramme de séquence " Supprimer MP"

❖ Récupération de MPs

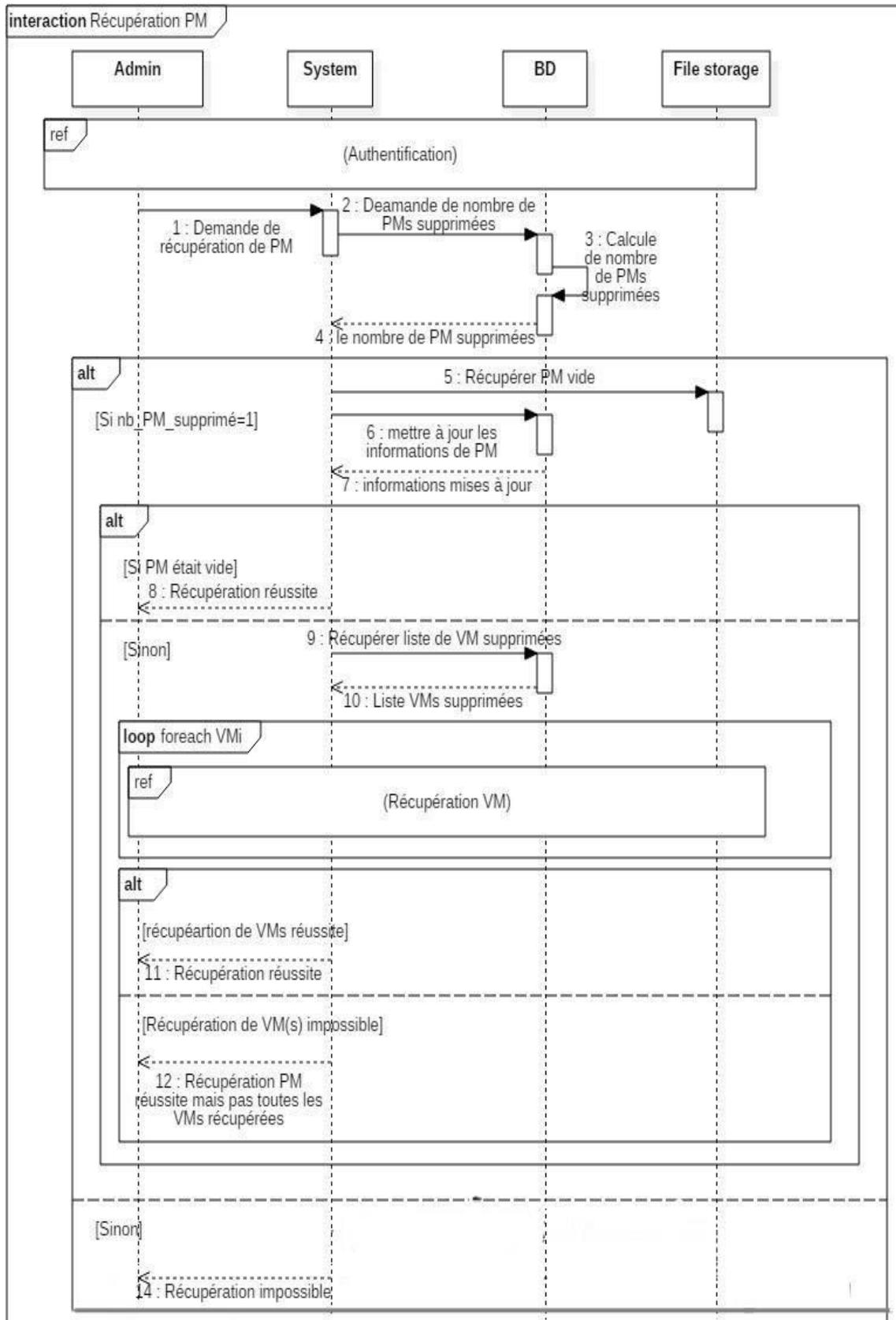


Figure 34 Diagramme de séquence " Récupérer MP"

**B. Diagramme de séquence « Gérer les machines virtuelles »**

Comme les MPs, les MVs aussi constituent des éléments importants dans le Cloud et qui sont gérées par les clients et/ou le propriétaire.

Dans notre système, le client peut ajouter, supprimer et récupérer ses propres MVs, alors que l'administrateur peut ajouter, supprimer et récupérer des MVs pour n'importe quel client. Les diagrammes suivants présentés dans les Figures suivantes : Figure 35, Figure 36 et Figure 37, montrent les principales fonctions dans la gestion des MVs (ajout, suppression et récupération) respectivement.

➤ Ajout de MVs

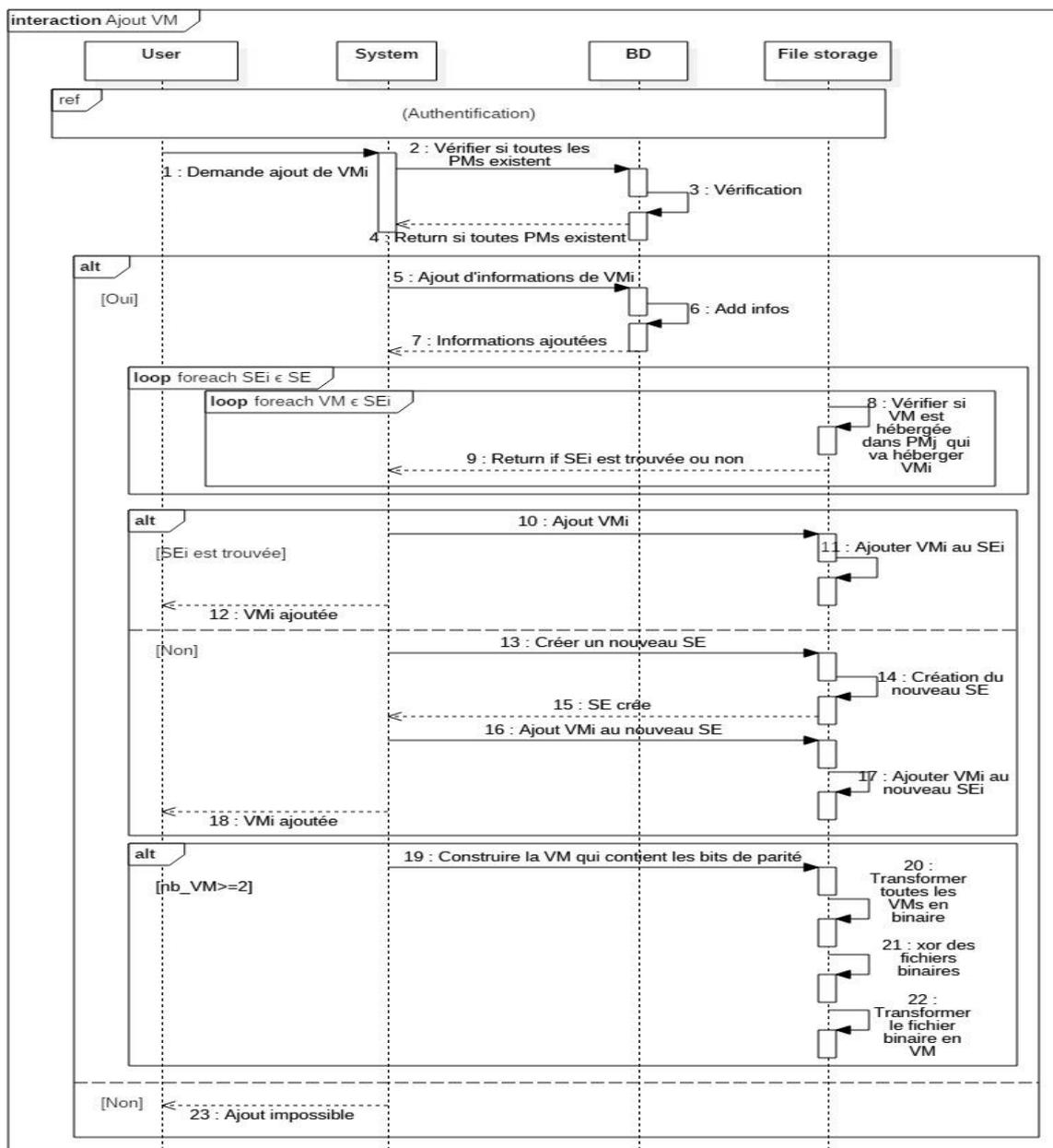


Figure 35 Diagramme de séquence "Ajouter MV"

➤ Suppression de MVs

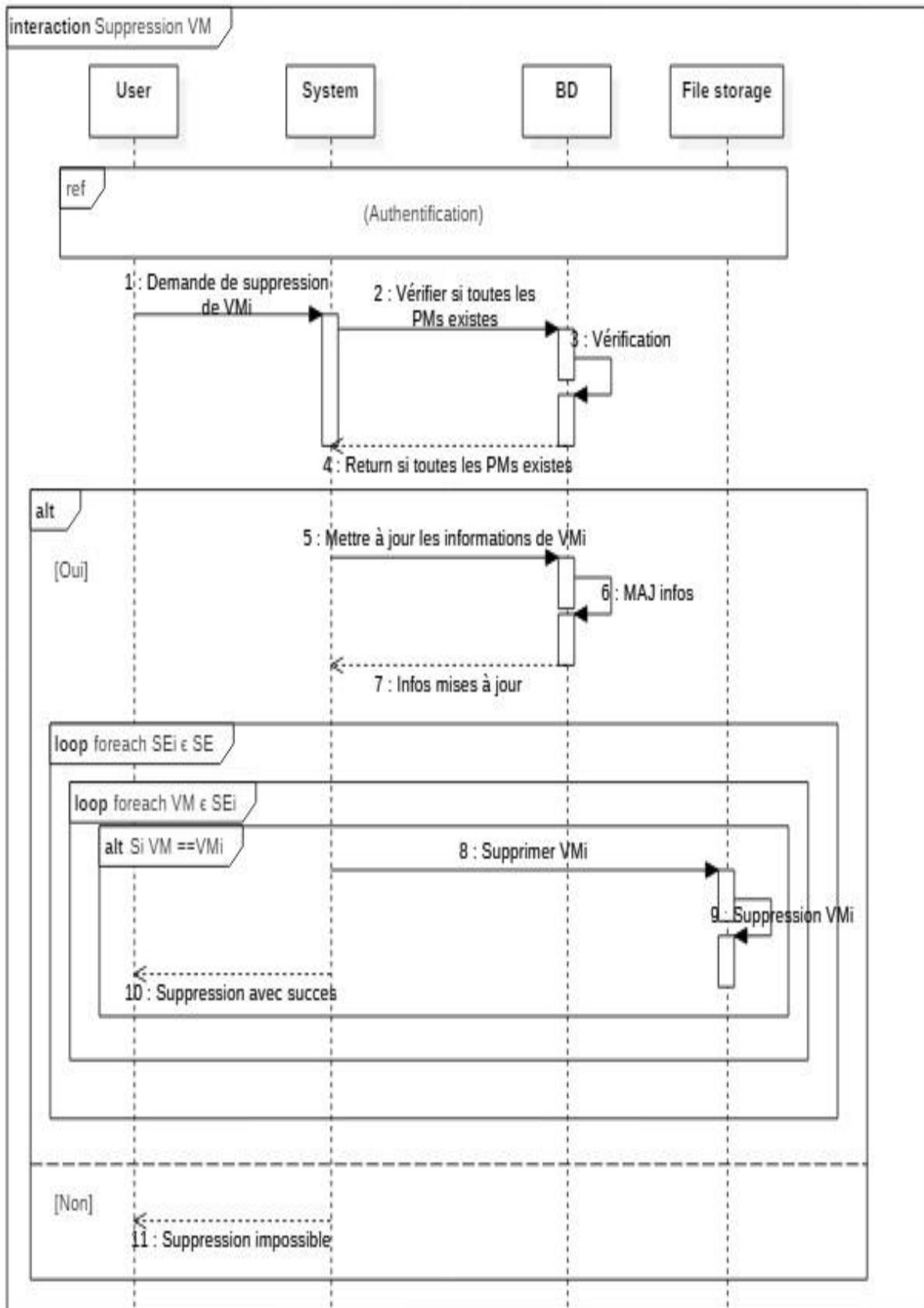


Figure 36 Diagramme de séquence " Supprimer MV"

➤ Récupération de MVs

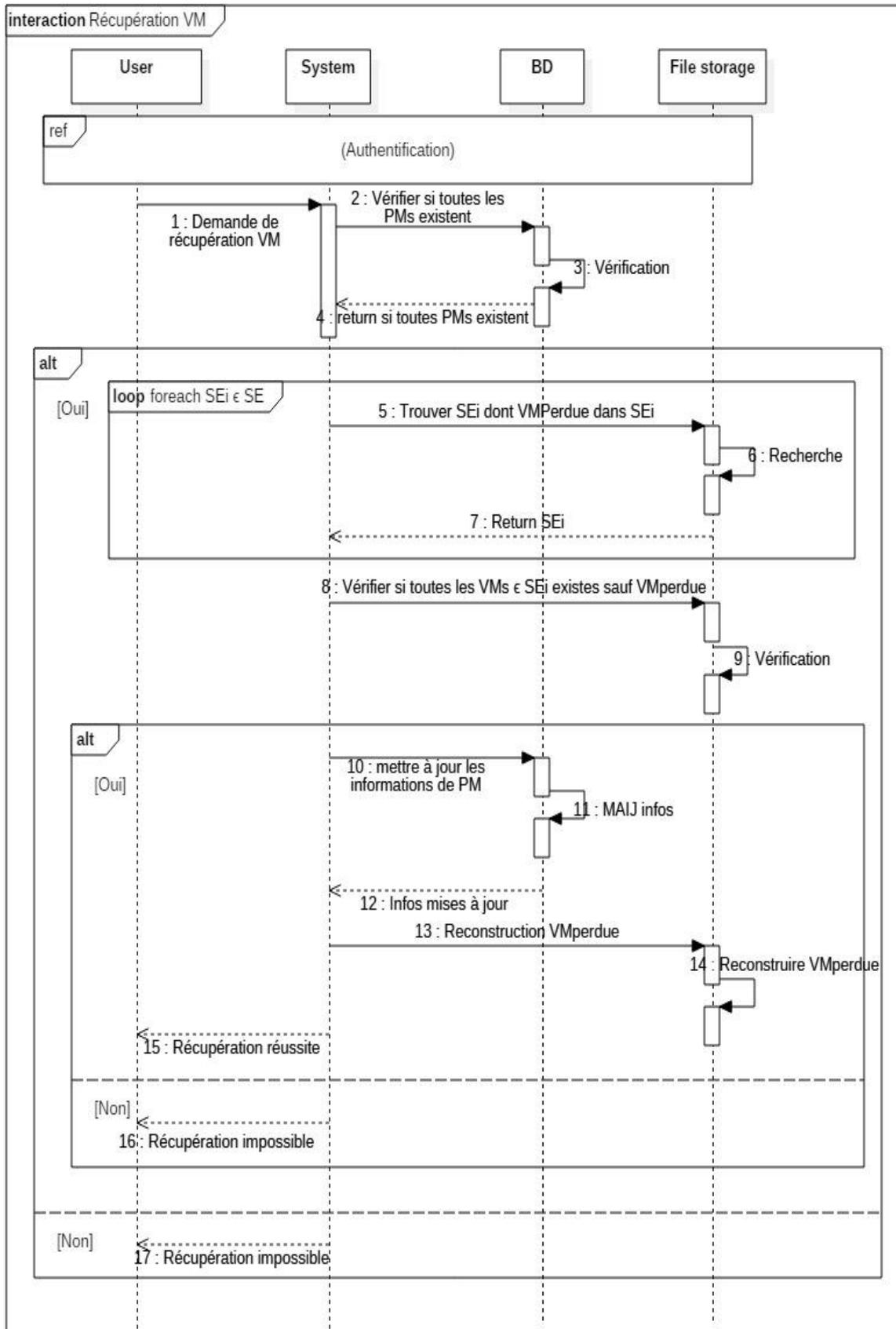


Figure 37 Diagramme de séquence " Récupérer MV"

### 4.4.3 Diagramme de classes

Le diagramme de classes présente les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d’UML car il fait abstraction des aspects temporels et dynamiques.

Vu que dans l’implémentation de notre approche on a prévu d’utiliser des images statiques de type (jpg, png, bmp...etc.) comme des MVs dans notre système réalisé (cela est présenté en détails dans le chapitre suivant), deux champs (height et width) sont ajoutées qui contiennent des informations détaillées sur les images statiques.

Le diagramme de classes du système et la description des classes sont illustrés respectivement dans la Figure 38 et le tableau6:

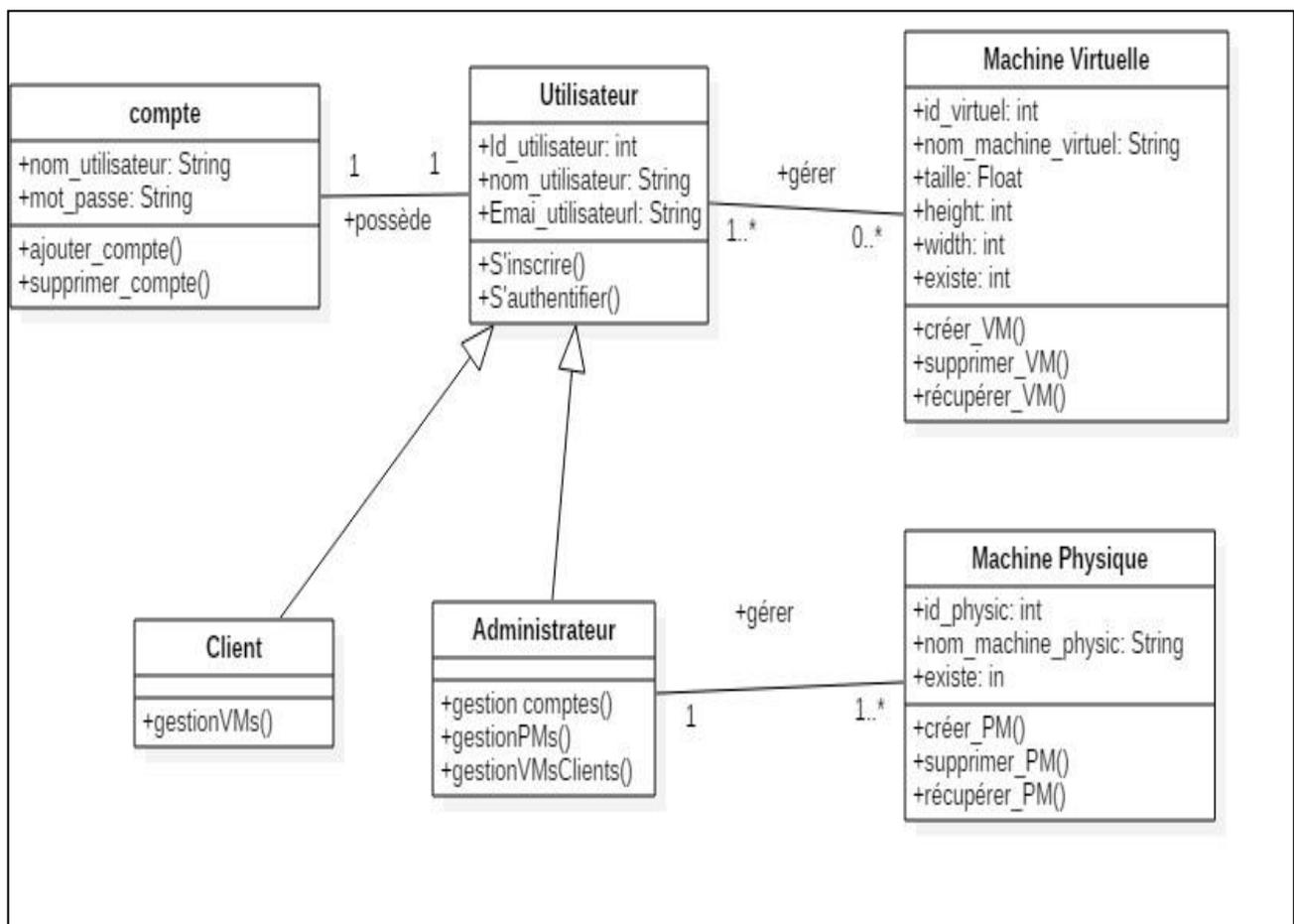


Figure 38 Diagramme de classes

Classe	Attribut	Type	Désignation	Méthode
Compte	Nom_utilisateur	String	Nom de l'utilisateur.	Créer_compte() Supprimer_compte()
	Mot de passe	String	Mot de passe de l'utilisateur.	
Utilisateur	Id_utilisateur	Int	Identifiant de l'utilisateur	S'inscrire() S'authentifier()
	Nom_utilisateur	String	Nom de l'utilisateur.	
	Email_utilisateur	String	Email de l'utilisateur.	
Machine Virtuelle	Id_virtuel	Int	Identifiant de la machine virtuelle	Créer_MV() Supprimer_MV() Récupérer_MV()
	Nom_machine_virtuel	String	Le nom de la machine virtuelle	
	Taille	Float	La taille de la machine virtuelle	
	Height	int	La hauteur de la machine virtuelle	
	Width	Int	La largeur de la machine virtuelle	
	Existe	Int	Si la machine virtuelle existe ou non (1 ou 0).	
Machine Physique	Id_physic	Int	Identifiant de la machine physique	Créer_MP() Supprimer_MP() Récupérer_MP()
	Nom_machine_physic	String	Nom de la machine physique	
	Existe	Int	Si la machine physique existe ou non (1 ou 0).	

Tableau 6 Tableau descriptif des classes

## 4.5 Conclusion

Après avoir présenté le contexte général de notre contribution où nous nous sommes concentrés sur la description et la conception de notre approche proposée, nommée CRAID5. Nous avons présenté deux algorithmes permettant de décrire le fonctionnement de notre approche. Ainsi que quelques diagrammes qui ont été formalisés à l'aide du langage UML, ce qui nous permet de mieux comprendre les différentes étapes à suivre dans la partie implémentation de notre solution.

Nous passons au chapitre suivant, où nous présentons les détails de notre implémentation, ainsi que les différents résultats obtenus.

## Chapitre 5 :

**IMPLEMENTATION**

**&**

**SIMULATION**

## 5.1 Introduction

Nous avons vu dans les chapitres précédents en quoi consiste notre proposition, et dans ce dernier chapitre, nous arrivons à la description de l'aspect pratique de notre travail. Nous présentons brièvement l'environnement de travail utilisé ainsi que les différents outils utilisés pour la réalisation de notre système CRAID5. En effet, Nous avons essayé de concevoir une interface intuitive et pratique, ainsi qu'une évaluation de notre système avec une autre technique très utilisée dans le domaine de tolérance aux pannes dans le Cloud, technique de réplication.

## 5.2 Contexte de l'implémentation

Dans notre implémentation de l'approche proposée, nous avons optés pour le scénario où chaque MV héberge uniquement une seule image, ensuite les différentes opérations définies dans la partie conception vont être appliqués sur le fichier image, car l'image est la partie statique de la MV, et la gestion des fichiers de type image est plus facile, ce qui nous permet de faire nos différents tests d'une façon plus ergonomique et nous permet de voir les résultats de notre approches plus clairement, en plus les opérations sur des images statiques est plus rapide.

Nous constatons que si notre approche fonctionne correctement sur les images statiques de type (png, jpg,...etc.), elle va systématiquement fonctionner sur des images de n'importe quel type (iso par exemple), car les différentes opérations ont été faites des bits alors que les MVs en réalité aussi sont stockées sous formes de bits.

## 5.3 L'environnement de travail

L'approche proposée dans ce travail a été implémentée et testée dans un environnement possédant les caractéristiques suivantes :

**5.3.1 Le système d'exploitation:** Nous avons utilisé le système d'exploitation Windows 10-64 bits.

**5.3.2 Caractéristiques de l'ordinateur:** L'ordinateur qui a été exploité possède la configuration présentée dans le tableau7:

Unité	Caractéristiques
Processeur	Intel(R) Core(TM) i3-5010 U CPU @ 2.10Ghz 2.10Ghz
Mémoire RAM	4.00 Go
Disque dur	500 Go
Écran	15 pouces

Tableau 7 Caractéristiques de l'ordinateur

## 5.4 Architecture, langages et environnements de développement

### 5.4.1 Le design pattern MVC (Model, Vue, Contrôle)

Dans le cadre de notre application, nous avons opté pour le design pattern MVC comme une architecture pour notre système (Figure 39).

MVC est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données. Comme son nom l'indique, le design pattern MVC est composé de trois parties:

- ❖ Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée "Modèle" (le M de MVC);
- ❖ Tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (mise en forme, affichage) doit être contenu dans la couche nommée "Vue" (le V de MVC);
- ❖ Tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans la couche nommée "Contrôle" (le C de MVC).

En pratique, le modèle correspond aux objets java, la vue correspond à la page JSP, et le contrôleur correspond au Servlet. [50]

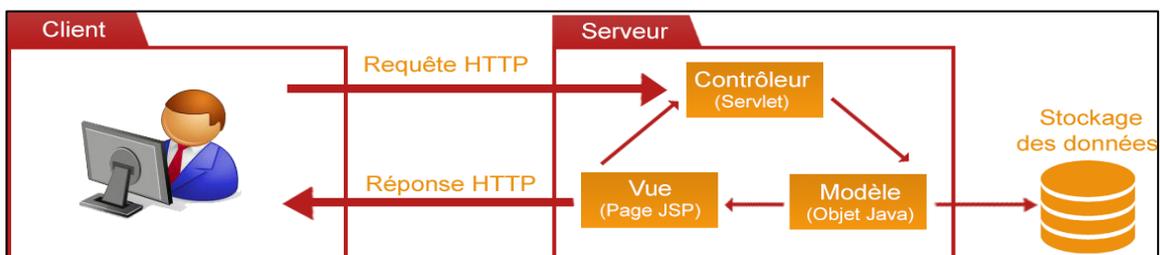


Figure 39 Modèle MVC

### 5.4.2 Java entreprise édition (Java EE)

Le terme Java EE anciennement raccourci en J2EE signifie Java Enterprise Edition, il fait référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE à quoi s'ajoute un grand nombre de bibliothèques remplissant différentes fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de JavaEE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Java EE regroupe un ensemble d'API chacun est destiné à faire un traitement spécifique. Dans le cadre de notre application, nous avons choisi d'utiliser les API suivants :

- **Servlet** : Une servlet est une classe Java, qui a la particularité de permettre le traitement de requêtes et la personnalisation de réponses ;
- **JSP (Java Server Pages)** : Les pages JSP sont une des technologies de la plateforme Java EE les plus puissantes, simples à utiliser et à mettre en place. Elles se présentent sous la forme d'un simple fichier au format texte, contenant des balises respectant une syntaxe à part entière. Le langage JSP combine à la fois les technologies HTML, XML, servlet et JavaBeans en une seule solution permettant aux développeurs de créer des vues dynamiques;
- **JDBC (Java DataBase Connectivity)** : C'est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis vos applications Java à des données rangées sous forme de tables. Dans la très grande majorité des cas, il s'agira bien entendu de bases de données stockées dans un SGBD.

Java EE permet de coder son application comme on le souhaite, néanmoins, il est recommandé d'adopter une architecture plus ou moins standard, que tout développeur peut reconnaître, c'est-à-dire dans laquelle tout développeur sait se repérer. [50]

### 5.4.3 Le serveur Apache Tomcat

Apache Tomcat est le serveur d'application Java du projet Jakarta de la fondation Apache. Ce serveur libre, sous licence Apache permet d'exécuter des applications web développées avec les technologies java (Servlet, JSP . . .etc.). [51]

### 5.4.4 Eclipse

Est un IDE libre, extensible développé en JAVA. Il permet de créer des projets de développement sous n'importe quel langage de programmation. Ceci permet à ses utilisateurs de l'adapter selon leurs besoins. [57]

### 5.4.5 Xampp

Est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (Apache, MySQL, Perl, PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [52]

### 5.4.6 MySQL

Le système de gestion de base de données relationnelle (SGBDR) open source Oracle MySQL s'appuie sur le langage de requête structuré SQL (Structured Query Language). Il est compatible avec presque toutes les plateformes notamment Linux, Unix et Windows. Utilisé pour toutes sortes d'applications, il est le plus souvent associé aux applications Web et à la publication de contenus en ligne. [53]

### 5.4.7 HTML5

C'est le langage universel utilisé sur les pages web lisibles par tous les navigateurs Web (Internet Explorer, Google chrome, Mozilla, etc.). Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue. [54]

### 5.4.8 CSS3

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. [55]

### 5.4.9 JavaScript

Le JavaScript est un langage informatique utilisé sur les pages web. Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Communications Corporation, Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activés côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

La particularité du JavaScript consiste à créer des petits scripts sur une page HTML dans le but d'ajouter une petite animation ou un effet particulier sur la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur, mais certains scripts sont peu utile et servent surtout à ajouter un effet esthétique à la page. L'intérêt du JavaScript est d'exécuté un code sans avoir à recharger une nouvelle fois la page. [58]

### 5.4.10 Bootstrap

Kit CSS créé en 2011 par les développeurs de Twitter, il permet de construire rapidement et facilement des sites web esthétiques et responsives. Bootstrap offre aussi des plugins jQuery de qualité pour enrichir les pages du site. [56]

### 5.4.11 jQuery

Est un framework Javascript sous licence libre qui permet de faciliter des fonctionnalités communes de Javascript. L'utilisation de cette bibliothèque permet de gagner du temps de développement lors de l'interaction sur le code HTML d'une page web. [62]

## 5.5 Présentation de l'application

Notre projet consiste à réaliser un système sous forme d'une application web, fonctionne dans un réseau local. Cette application est une implémentation de l'approche de tolérance aux pannes proposée, nommée CRAID5.

Tel que décrit dans la Section 5.2 de ce chapitre, dans l'implémentation du notre système, on a proposé d'utiliser les fichiers de type image statique pour représenter les MVs, et les répertoires comme étant des MPs.

### ❖ Les interfaces graphiques

Notre système réalisé contient plusieurs fenêtre, chaque fenêtre a ses propres fonctionnalités, la description des fenêtres est la suivante:

- **Interface « Accueil » (Index) et « A propos »**

La Figure 40 illustre l'interface « Accueil » qui représente la vitrine de notre système, il s'agit de la première page qui apparait à l'utilisateur, elle joue un rôle de portail. Cette fenêtre contient quatre boutons, le premier bouton permet d'afficher la fenêtre d'accueil, le deuxième bouton nous permet d'avoir quelques informations sur la page « A propos », le troisième nous permet de s'authentifier pour pouvoir accéder à l'espace d'utilisateur et le quatrième bouton pour s'inscrire.

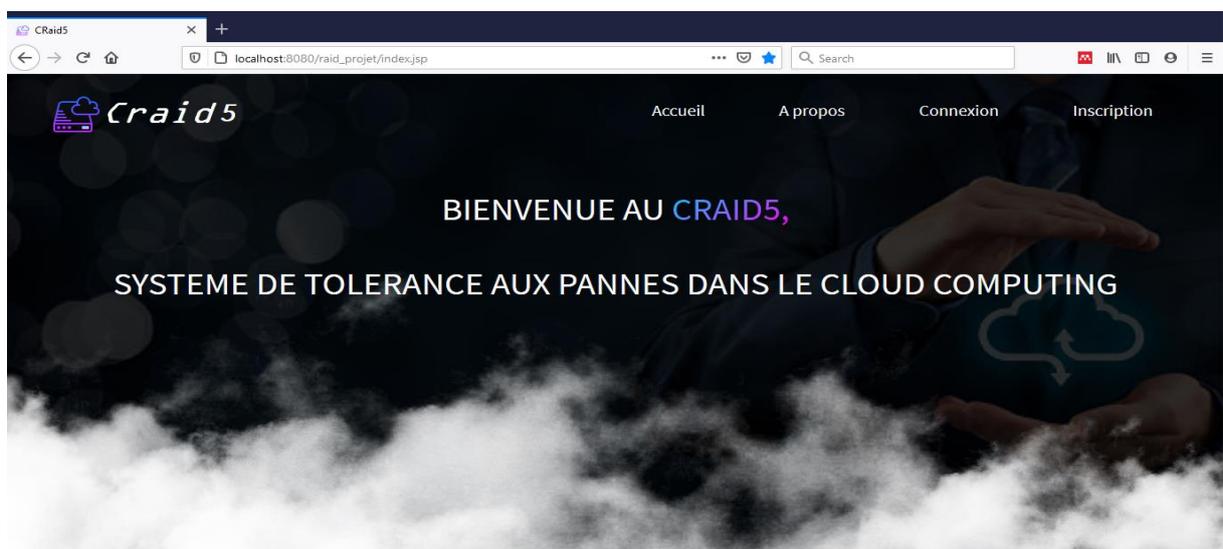


Figure 40 Interface "Accueil"

La Figure 41 illustre la page « A propos » donne une petite description et des informations sur notre système.

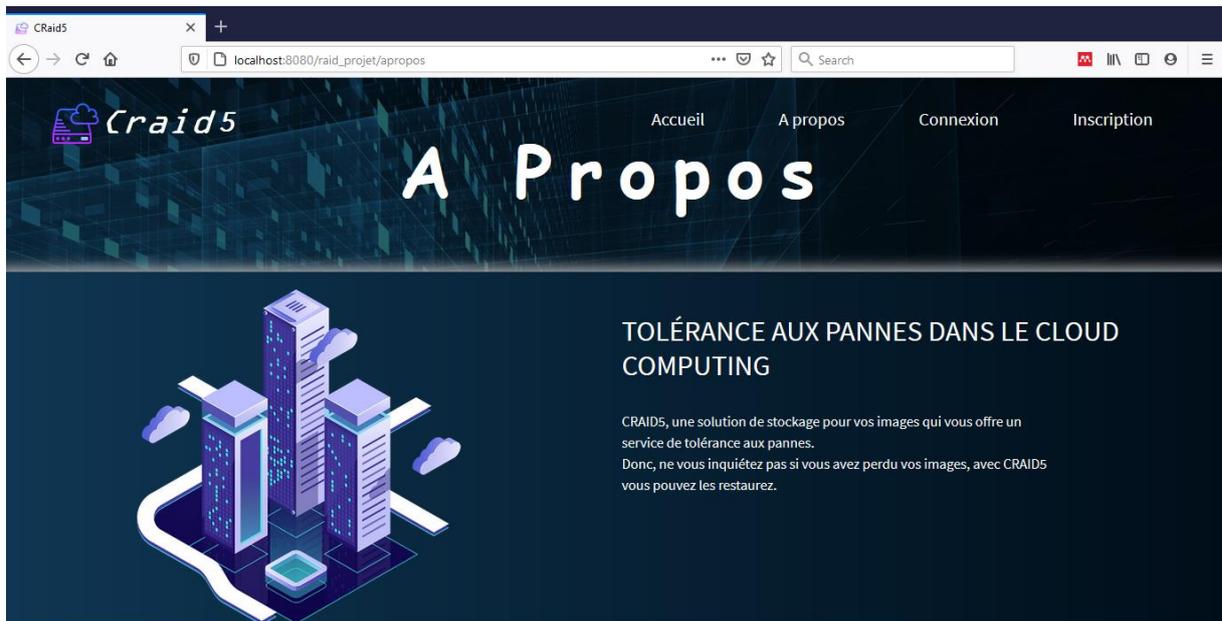


Figure 41 Interface "A propos"

- **Interface « Inscription »**

L'utilisateur doit entrer son pseudo, mot de passe et email afin de s'inscrire dans l'interface montrée par la Figure 42, après la validation d'inscription, automatiquement ses informations sauvegarderaient dans la BD où le mot de passe de l'utilisateur va être stocké d'une façon chiffré pour des raisons de sécurité comme représenté dans la Figure 44 et l'utilisateur va recevoir un email comme illustré dans la Figure 43 et la page va s'orienter directement vers la page d'« Authentification » pour qu'il puisse authentifier et accéder à son propre espace, sinon une erreur d'inscription s'affichera et dans ce cas-là il faut réinsérer les informations correctement.

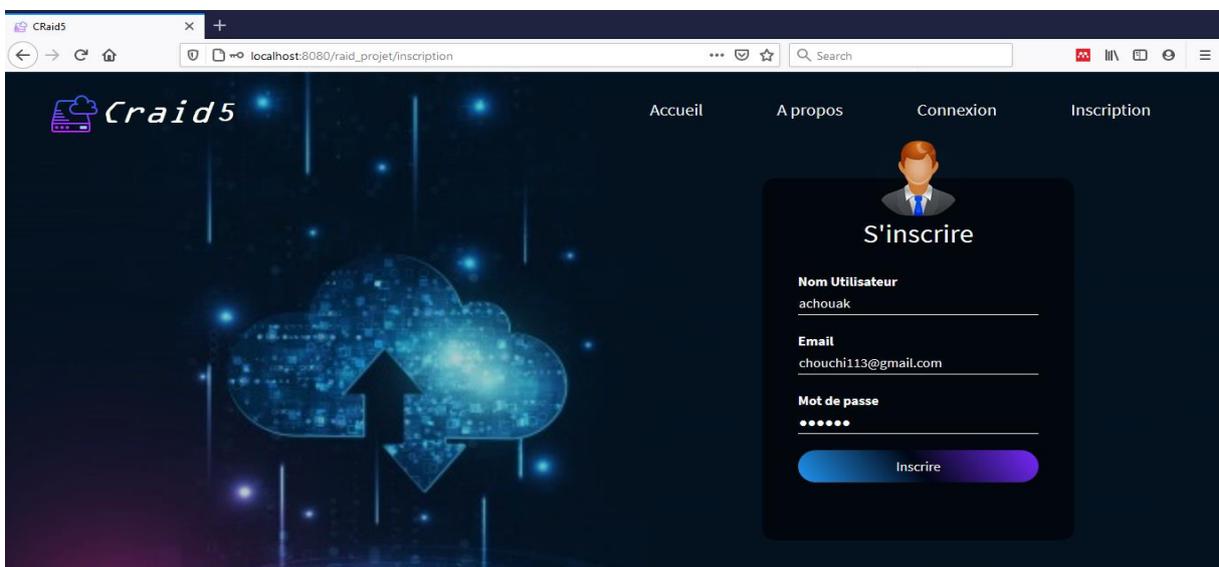


Figure 42 Interface "Inscription"



Figure 43 Message reçu après l'inscription

+ Options			IdRegister	Utilisateur	Email	Mot_passe
<input type="checkbox"/>	Edit  Copy  Delete		1	admin	admin@gmail.com	21232f297a57a5a743894a0e4a801fc3
<input type="checkbox"/>	Edit  Copy  Delete		2	achouak	chouchi113@gmail.com	89e8492e1a0f28e2e29bed6dbb9c547b

Figure 44 Informations des utilisateurs enregistrées sur la BD

- **Interface « Authentification »**

Le client doit taper son pseudo et son mot de passe correctement afin d'accéder à son propre espace qui contient dans la fenêtre principale un affichage pour ses propres MVs, autrement une erreur d'authentification s'affichera et dans ce cas-là il faut réinsérer les informations d'une manière correcte comme représenté dans la Figure 45:

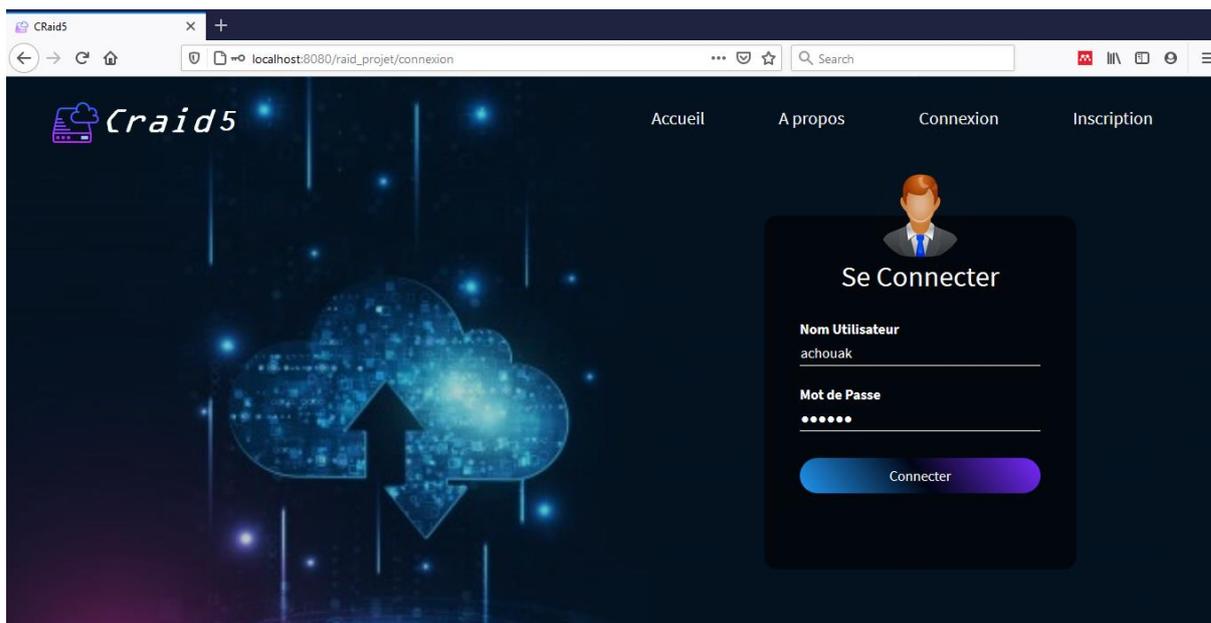


Figure 45 Interface "Authentification"

- **Interface « Status de Cloud »**

Cette interface représente l'état de Cloud. Après l'authentification, le propriétaire de Cloud peut voir toutes les MVs existantes de tous les clients avec toutes ses informations (nom de MV, la taille de stockage, sa hauteur, sa largeur ainsi que son id). Autrement, si c'était un client qui a accédé à son compte, son status de Cloud lui affichera ses propres MVs existantes qui ont été créés par lui auparavant comme le montre la Figure 49, sinon un tableau vide qui va être affiché pour montrer qu'il n'y a aucune MV créée dans le Cloud. Dans le tableau des MVs, les lignes présentées en rouge

réfèrent aux MVs non disponibles dans le Cloud. On remarque aussi le menu à gauche qui s'ouvre en passant la souris ou en cliquant sur « Menu », qui affiche le nom de l'utilisateur en haut avec quatre boutons, le premier représente le bouton de status Cloud de la page actuelle, le deuxième pour la page gestion de Cloud, le troisième est conçu pour naviguer vers la gestion des clients (ce bouton est affiché seulement pour le propriétaire), et le dernier pour la déconnexion si un utilisateur veut quitter l'application. Les deux menus pour l'admin ou le client sont illustrés respectivement dans les Figures 47 et 46.

Deux boutons en bleu en haut dans l'espace d'admin pour voir l'évaluation par implémentation ou la visualisation des résultats obtenus par simulation. La Figure 48 montre le status de Cloud de l'admin avec toutes les MVs qui appartiennent aux différents clients.



Figure 46 Menu "Client"

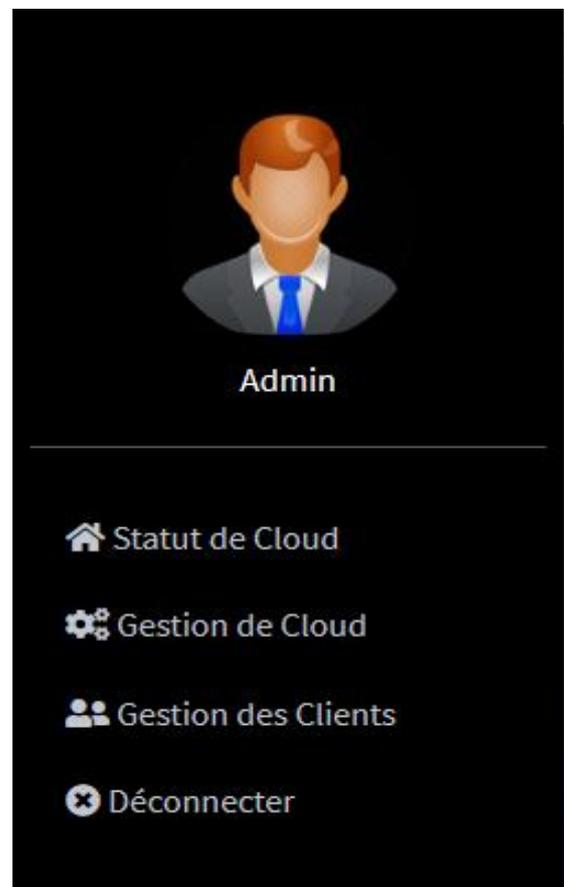


Figure 47 Menu "Admin"

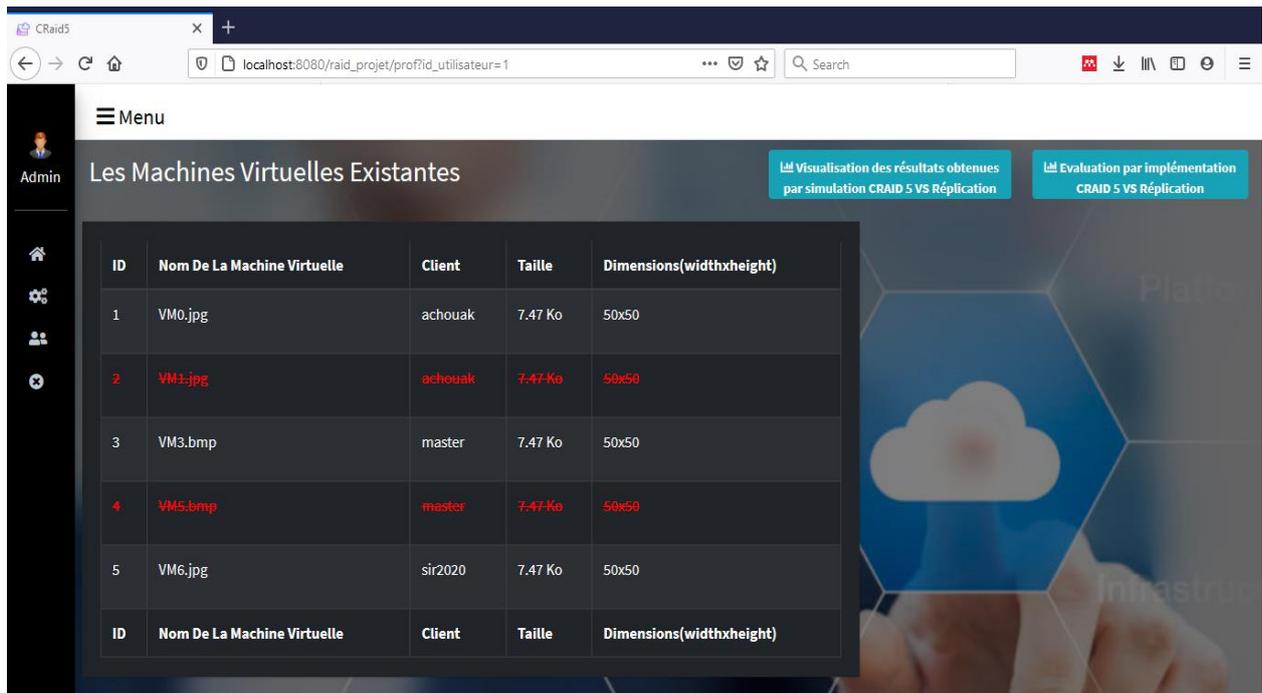


Figure 48 Interface "Status de Cloud" pour l'admin

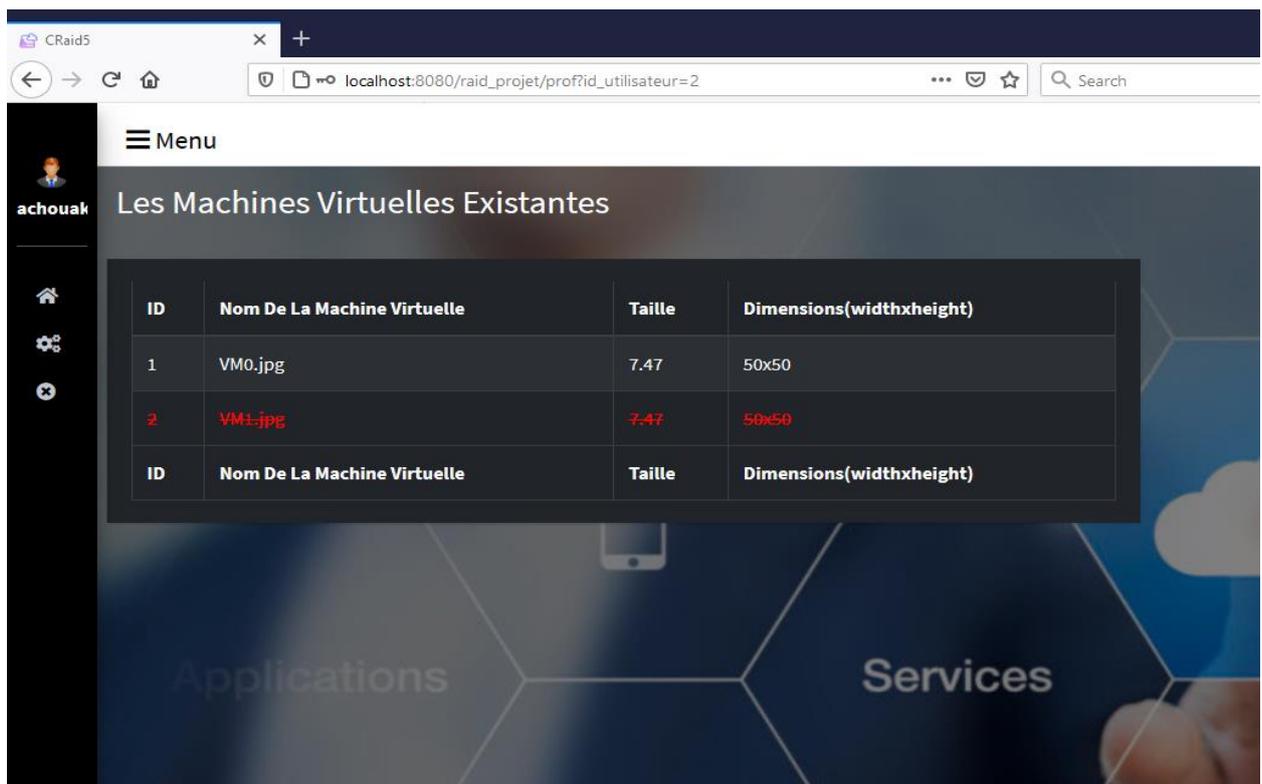


Figure 49 Interface "Status de Cloud" pour le client

- **Interface « Gestion des comptes clients »**

Dans cette interface, le propriétaire de Cloud peut supprimer ou ajouter de nouveaux clients. On distingue quelques boutons, un pour le retour vers « le status de Cloud », un autre pour rafraichir la

page, «supprimer client » pour la suppression de client, et le dernier bouton pour l’ajout d’un nouveau client. L’interface « Gestions des comptes clients » est illustrée dans la Figure 51.

En cliquant le bouton « Ajouter Client » une fenêtre s’affichera où il peut taper les coordonnées de client après il clique sur ajouter pour ajouter le nouveau client, comme le montre la Figure 50.

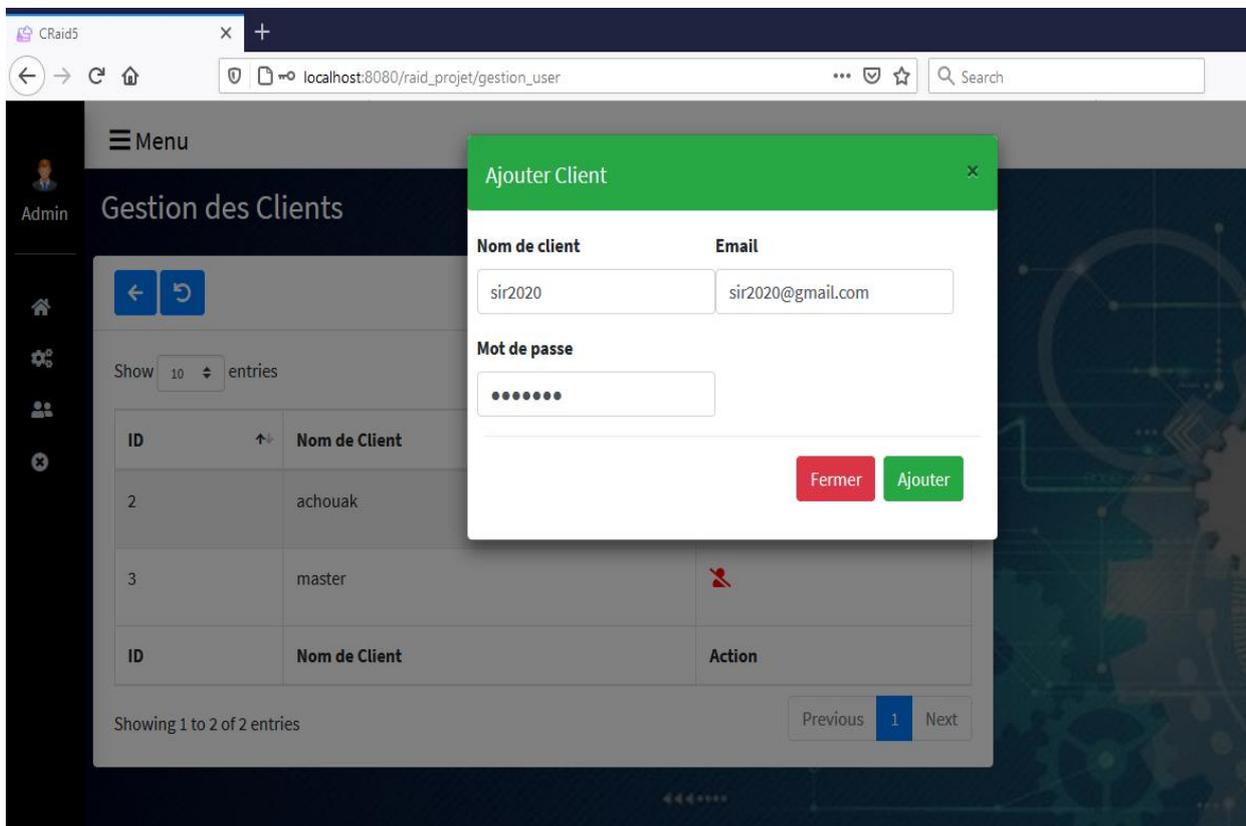


Figure 50 Ajout Client par l’admin

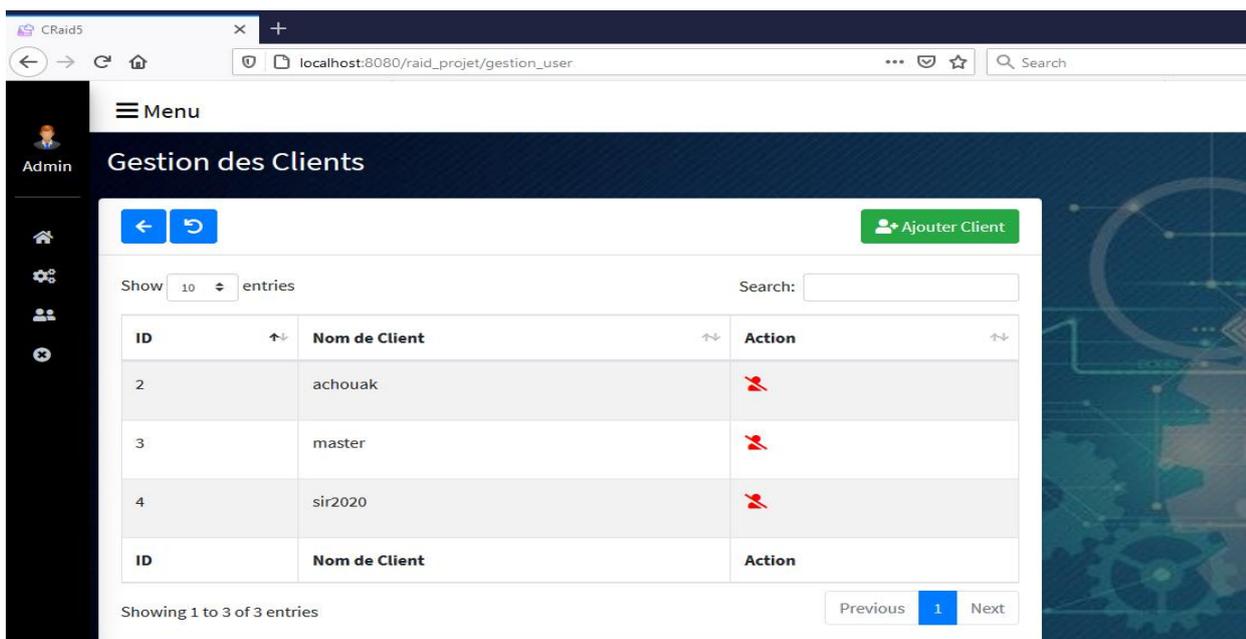


Figure 51 Interface "Gestion des Comptes Clients"

- **Interface « Gestion de Cloud »**

En cliquant sur le deuxième bouton dans le menu, et si c'était l'espace de propriétaire, une interface de gestion des MPs qui s'affichera, sinon, si c'était un client on passe directement vers la gestion de ses MVs.

### 1. Gestion de MPs

En accédant à cette interface par l'admin, le système affichera les MPs existantes qui ont été créés par lui auparavant, sinon au départ le Cloud ne possède pas de MPs et un tableau vide qui va être affiché pour montrer qu'il n'y a aucune MP créée dans le Cloud et le bouton « Gestion MVs » est désactivé jusqu'à l'ajout des MPs. Ici l'admin peut ajouter, supprimer ou récupérer les MPs supprimées, qui sont gérées par lui seulement.

Cette interface possède plusieurs boutons, un bouton pour l'ajout des MPs, un autre « Gestion MVs » pour naviguer vers la page de gestion des MVs, le bouton bleu « récupérer MP » permet la récupération de MP perdue qui est désactivé si la MP existe. Un bouton pour la suppression des MPs et le dernier « actualiser » pour rafraîchir la page, et le bouton « gestion MV » pour passer à la page de gestion des MVs et le dernier bouton « initialiser le Cloud » permet à l'initialisation du Cloud (mettre le Cloud à nouveau c'est-à-dire la suppression de toutes les ressources MPs et MVs existantes sur le Cloud), en appuyant sur lui, une petite fenêtre qui s'affichera pour confirmer l'initialisation de Cloud ou non comme mentionné dans la Figure 52.

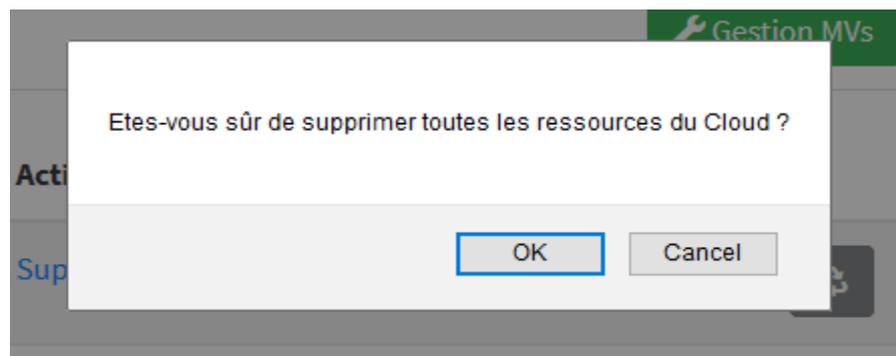


Figure 52 Fenêtre de confirmation d'initialisation de Cloud

#### A. Ajout de MPs

En appuyant sur le bouton « ajouter MP » une petite fenêtre s'affichera illustrée dans la Figure 53 qui contient une liste de choix où le propriétaire peut choisir le nombre de MPs à ajouter selon ses besoins. Vu que notre système est basé sur RAID5, le propriétaire ne peut pas créer moins de trois machines physiques.

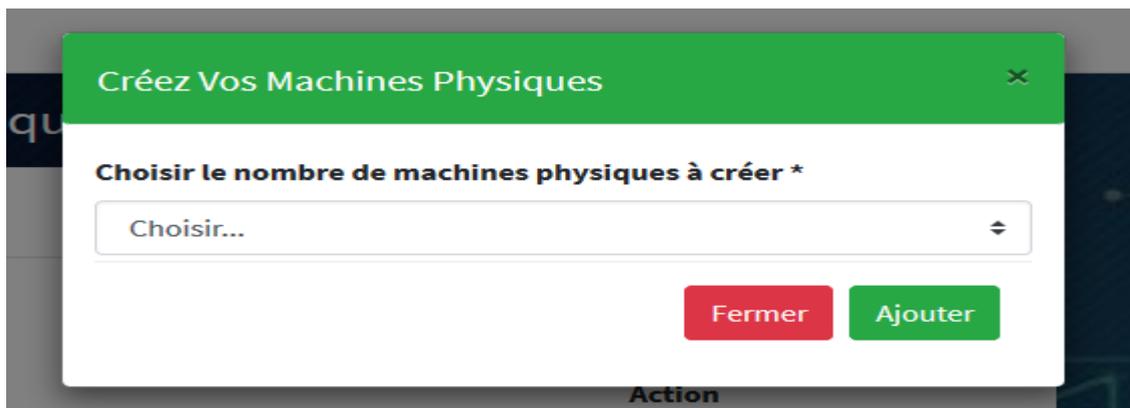


Figure 53 Fenêtre Ajouter MPs

Les MPs ajoutées seront stockées comme étant des sous-répertoires dans le répertoire racine « admin » comme l'illustre la Figure 54 et leurs informations seront envoyés à la BD.

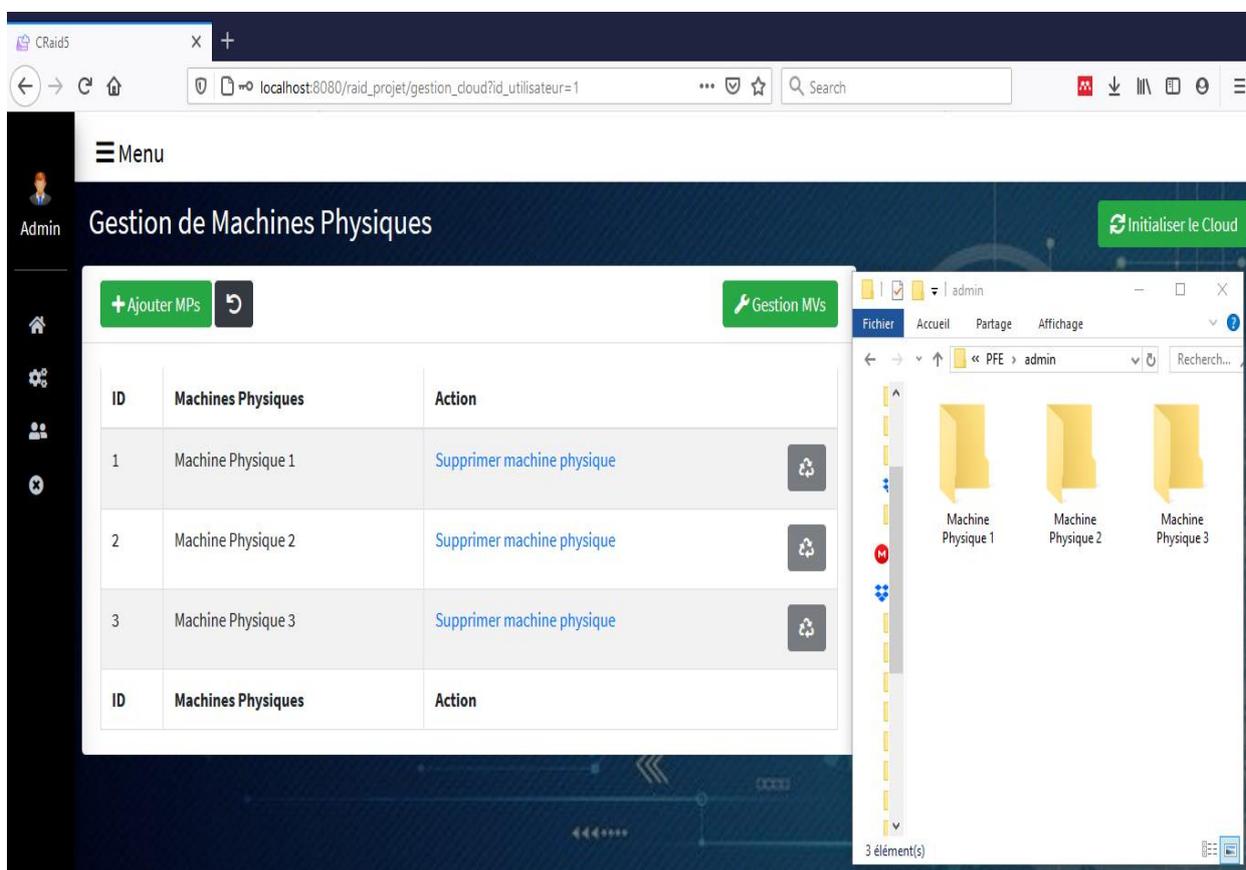


Figure 54 Interface "Gestion de Cloud" après l'ajout de MPs

### B. Suppression de MPs

S’il clique sur le bouton «supprimer machine physique » pour supprimer une machine ciblé, les MPs sont supprimées n’auront plus disponibles dans le Cloud, la ligne sur le tableau qui la représente va être barrée, en rouge et le bouton « récupérer machine physique » qui a été désactivé avant la suppression, va être activé pour qu’il puisse récupérer la machine physique perdue avec toutes les

MVs incluses si y'en avait, ainsi que le bouton « Gestion MVs » aussi va être désactivé d'où les clients ou l'admin ne peuvent pas ajoutées des MVs, donc la panne de MP entraine la panne de tout le système comme le montre la Figure 55.

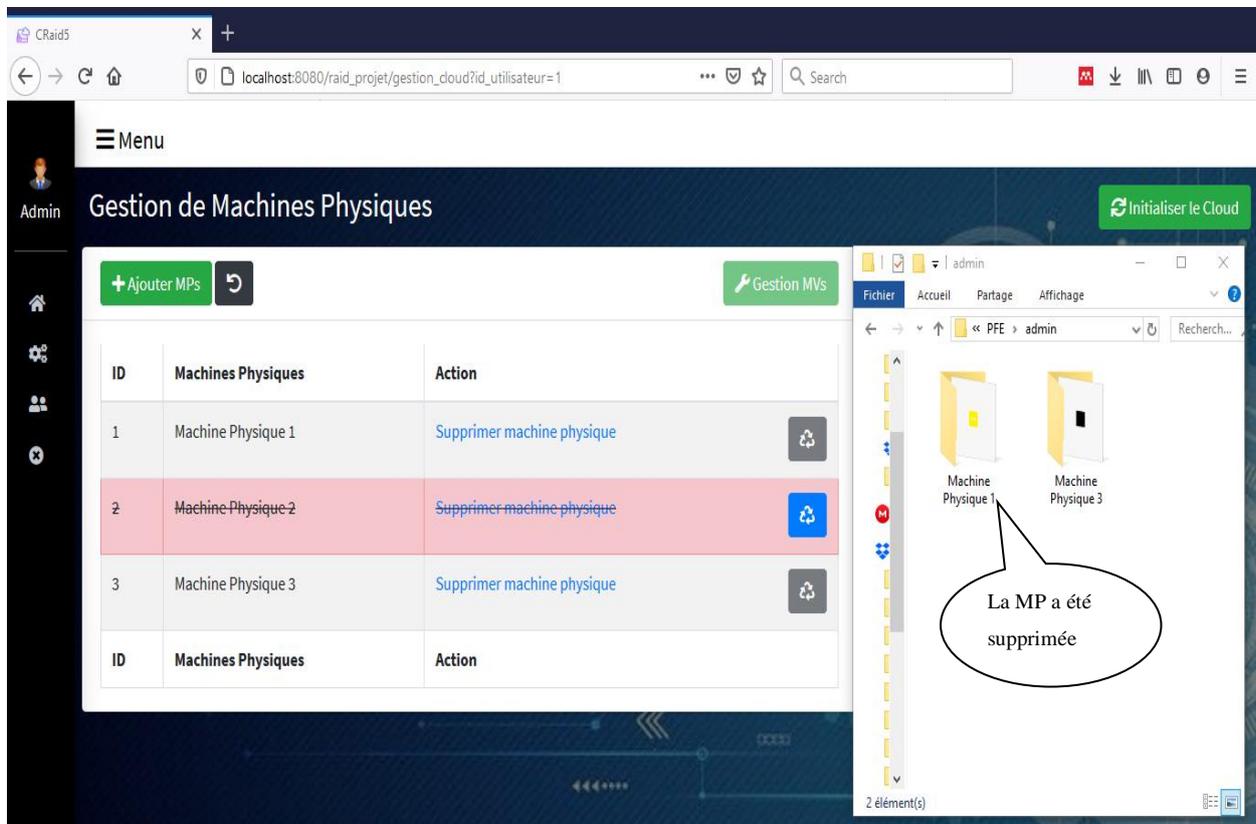


Figure 55 Interface "Gestion de Cloud" après la suppression de MPs

### C. Récupération de MPs

Cette étape vient après une panne de MPs qui peut être causée par une suppression de celles-ci afin de récupérer la MP avec les MVs hébergées si y'en avait. Elle peut être faite par l'admin afin de récupérer les MPs perdues.

A la récupération, un message va être affiché pour nous informer sur trois cas qui peuvent être trouvés, le premier la récupération est réussite où le système réussit à récupérer toutes MVs résidentes dans la MP perdue, le deuxième, la récupération de la MP est réussite mais pas toutes les MVs sont récupérées et le dernier cas, si y'en a plus qu'une MP perdue sur le Cloud donc c'est impossible de récupérer la MP perdue et un message disant que la récupération de MP perdue est impossible et c'est impossible à continuer à utiliser CRAID5, donc il faut une réinitialisation du Cloud afin de faire d'autre tests. Si la récupération est réussite, le bouton « Gestion MVs » sera activé et la ligne représentante sur le tableau n'est plus en rouge, sinon la machine physique sera supprimée carrément.

La récupération des MPs est illustrée dans la Figure 56.

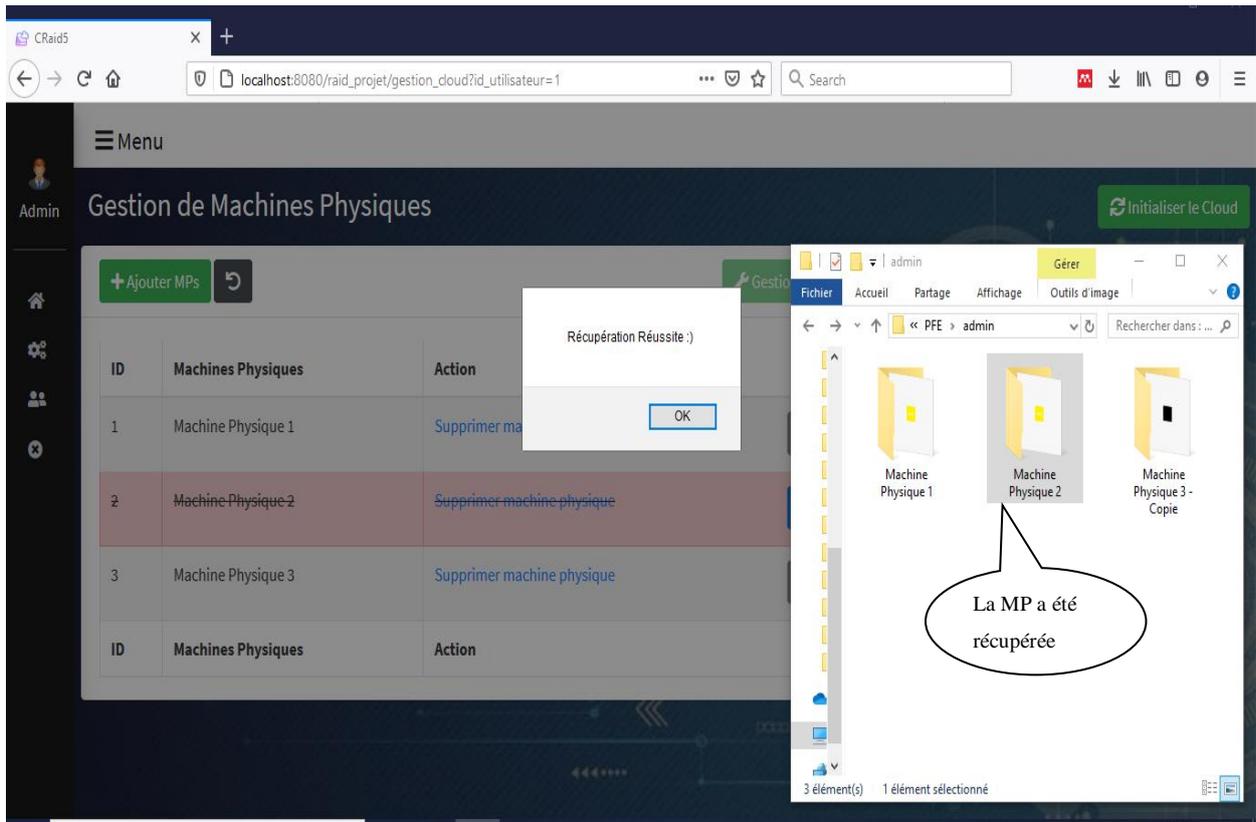


Figure 56 Récupération réussite de MPs

## 2. Gestion de MVs

Cet onglet est affiché pour le client et le prestataire pour qu'ils puissent gérer les MVs. Le client peut gérer ses MVs en ajoutant, supprimant ou récupérant les MVs perdues, un bouton pour la suppression des MVs, le bouton bleu « récupérer MV » permet la récupération de MV perdue. Un bouton pour le retour vers la page de status Cloud et un autre pour actualiser la page. Le dernier bouton est le bouton ajouter. En cliquant sur ce bouton, une petite fenêtre qui s'affiche, où le client ou l'admin peut uploader les MVs, afin de les stocker dans le Cloud.

### A. Ajout de MVs

Elle se fait soit par un client ou par le propriétaire de Cloud.

- **Espace Admin** : comme le montre la fenêtre 57, le propriétaire de Cloud peut ajouter des MVs pour n'importe quel client en choisissant son nom parmi les noms des clients dans la liste de choix.

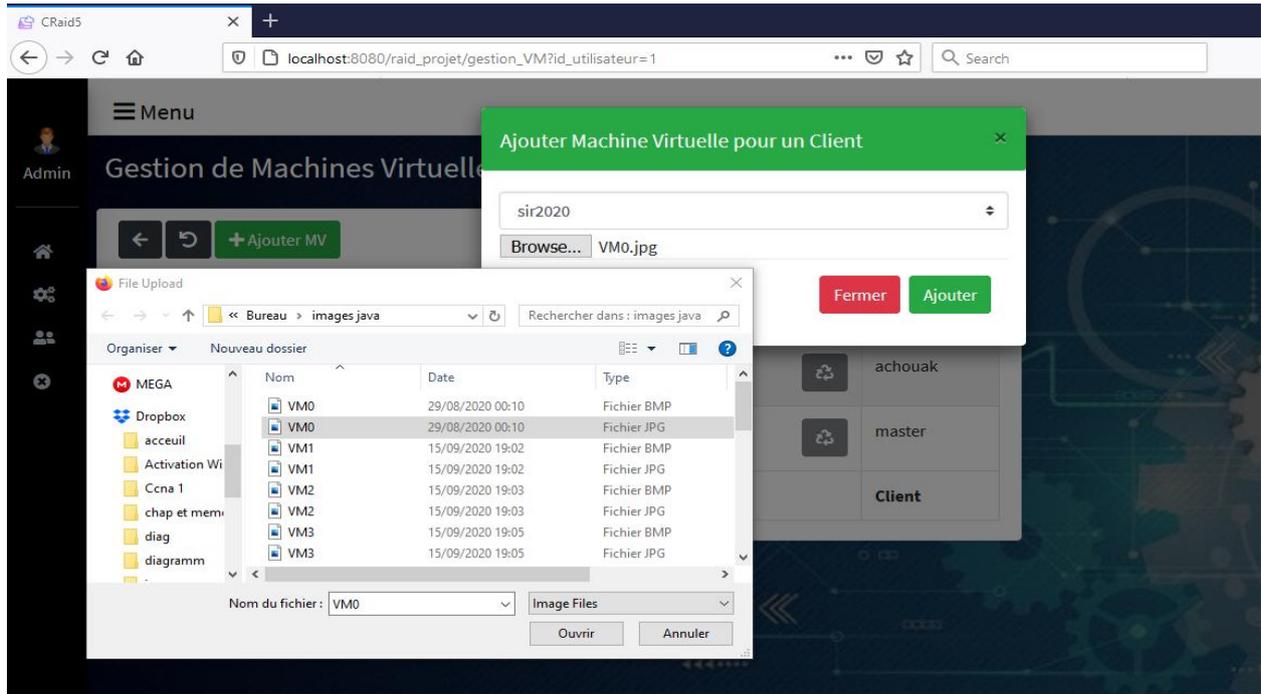


Figure 57 Ajout de MV pour le client "sir2020" par l'admin

Les MVs ajoutées seront stockées dans les MPs ainsi que leurs MVs contenant leurs bits de parité (vu dans le chapitre précédent) comme l'illustre la Figure 58 et leurs informations sur la BD.

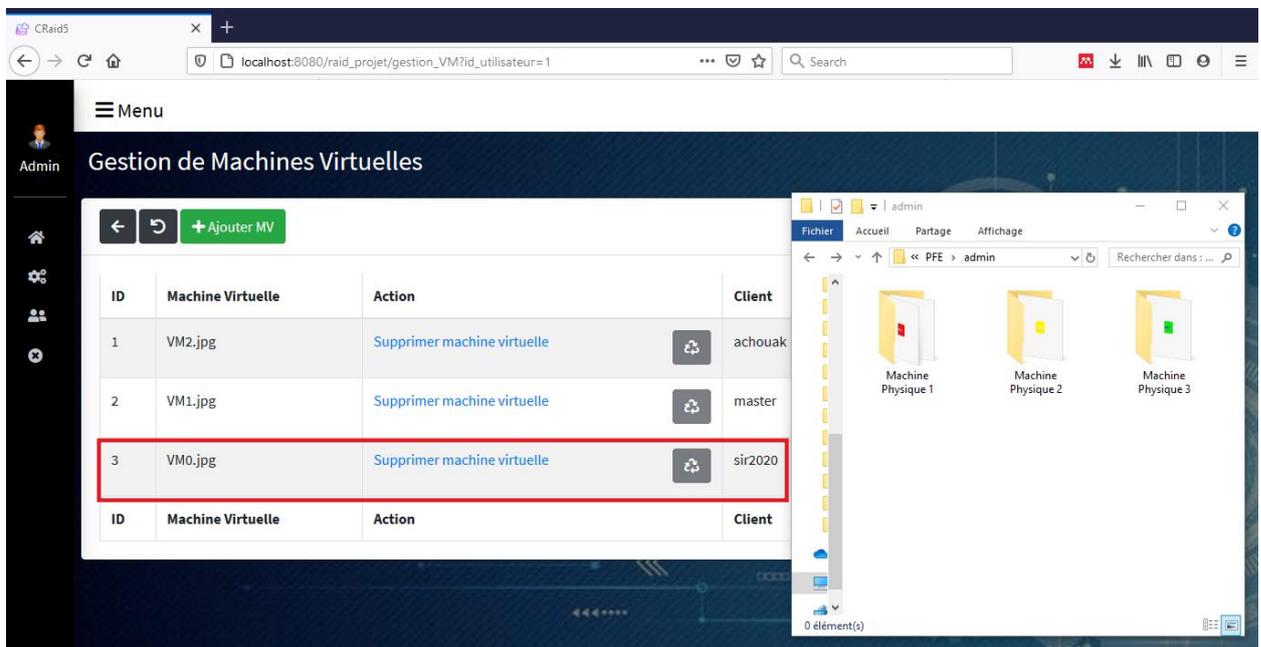


Figure 58 MV de client a été ajoutée

- **Espace Client :** Ici le client peut ajouter ses propres MVs sur son espace client en ajoutant la MV désirée dans la fenêtre qui s'affiche dans la Figure 59.

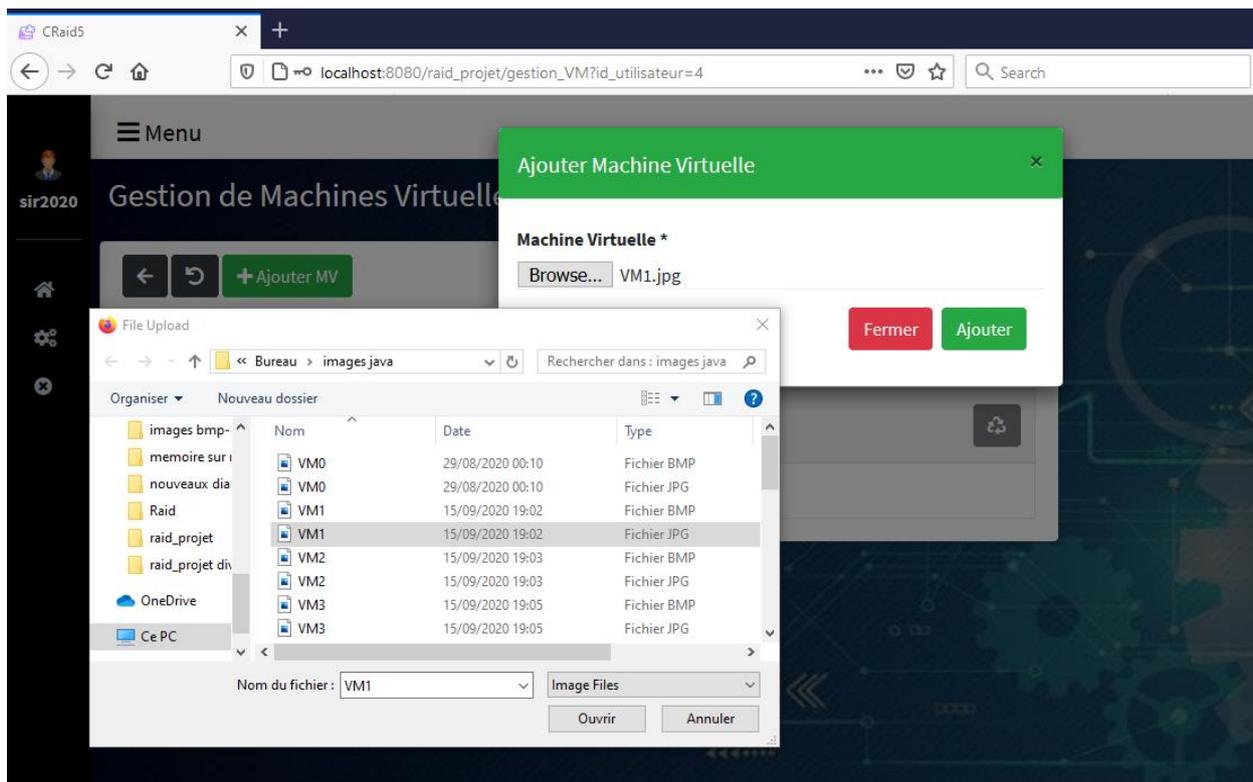


Figure 59 Interface « Gestion MVs » avec un ajout de MV par le client "achouak"

Pareil pour le client, les MVs ajoutées seront stockées dans les MPs existante comme l'illustre la Figure 60 et leurs informations sur la BD.

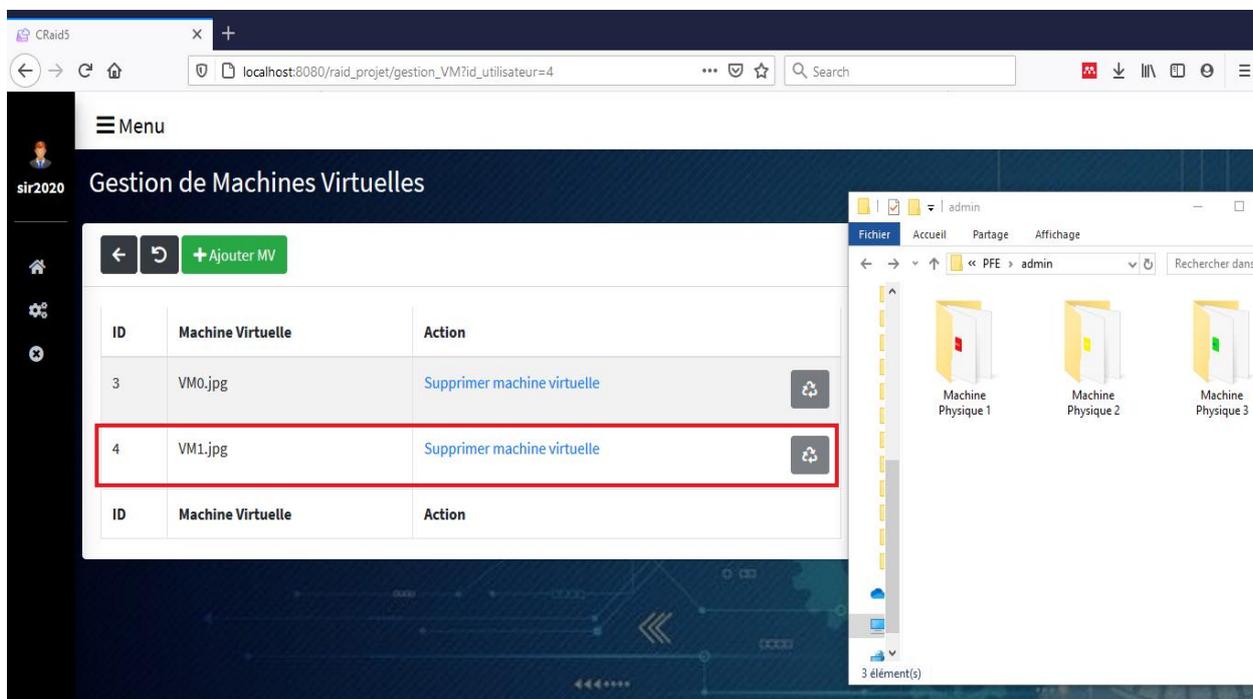


Figure 60 MV de client "achouak" a été ajoutée

**B. Suppression de MVs**

Aussi elle se fait soit par un client ou par le propriétaire de Cloud. Quand les MVs sont supprimées elles n'auront plus disponibles dans le Cloud, elles vont être barrées et en rouge et le bouton de récupération qui a été désactivé sera activé pour pouvoir faire la récupération de MV perdue.

➤ **Espace Admin :** Comme l'ajout, l'admin peut supprimer n'importe quel MV pour n'importe quel client. La Figure 61 illustre la suppression des MVs de clients par l'admin.

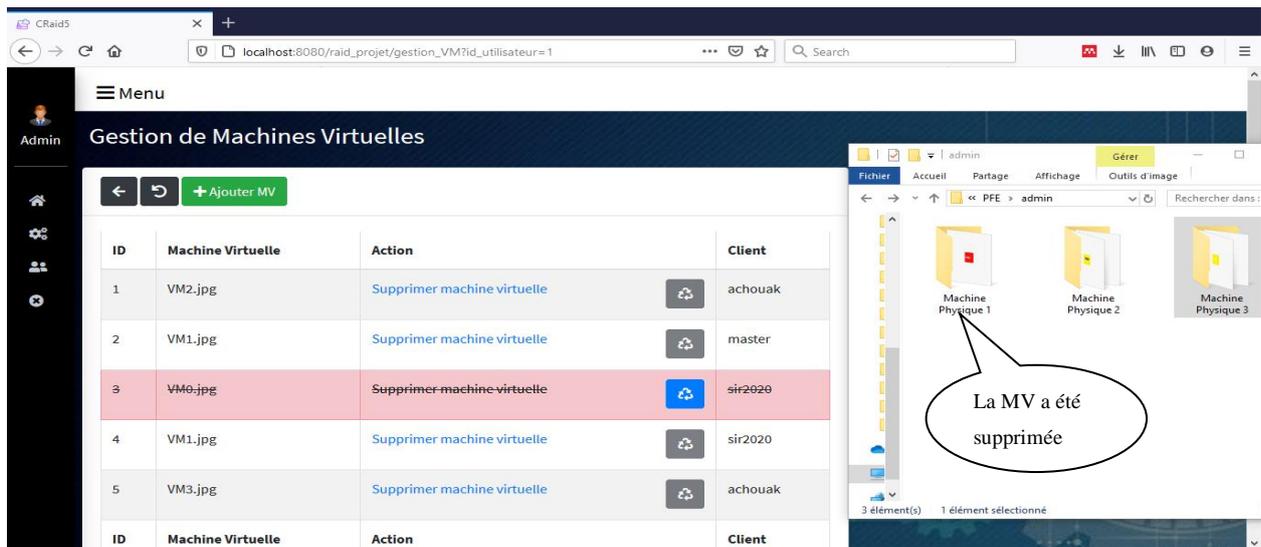


Figure 61 Interface « Gestion MVs » après une suppression de MVs par l'admin

➤ **Espace Client :** Le client a la possibilité de supprimer ses propres MVs existantes sur son espace comme le montre la Figure 62.

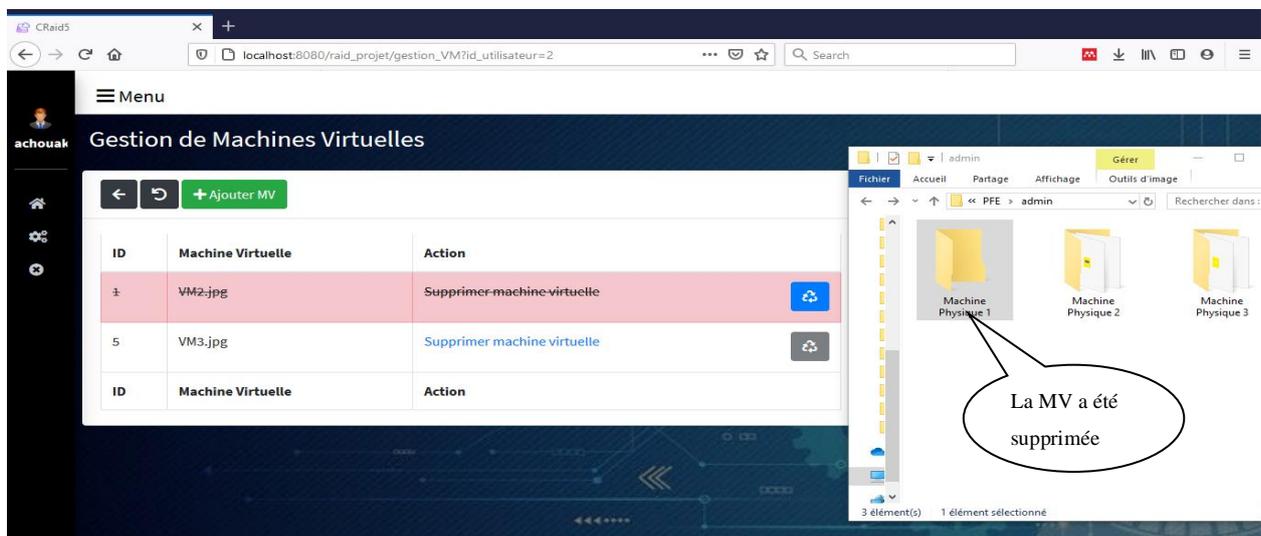


Figure 62 Interface « Gestion MVs » après une suppression de MVs par le client "achouak"

### C. Récupération de MVs

Comme la récupération des MPs, cette partie vient après une indisponibilité de ressources MVs causée par une suppression de celles-ci afin de rendre le système à son état normale après la panne. Elle peut être faite par les clients pour récupérer ses MVs ou par l'admin afin de récupérer les MVs des clients.

A la récupération, un message va être affiché pour informer est-ce que la récupération est faite ou c'est impossible de récupérer la donnée perdue. Si la récupération est réussite, la ligne représentante sur le tableau n'est plus en rouge, sinon la machine physique sera supprimée définitivement. La récupération de MVs par un client est illustrée dans la Figure 63, le même processus pour l'admin.

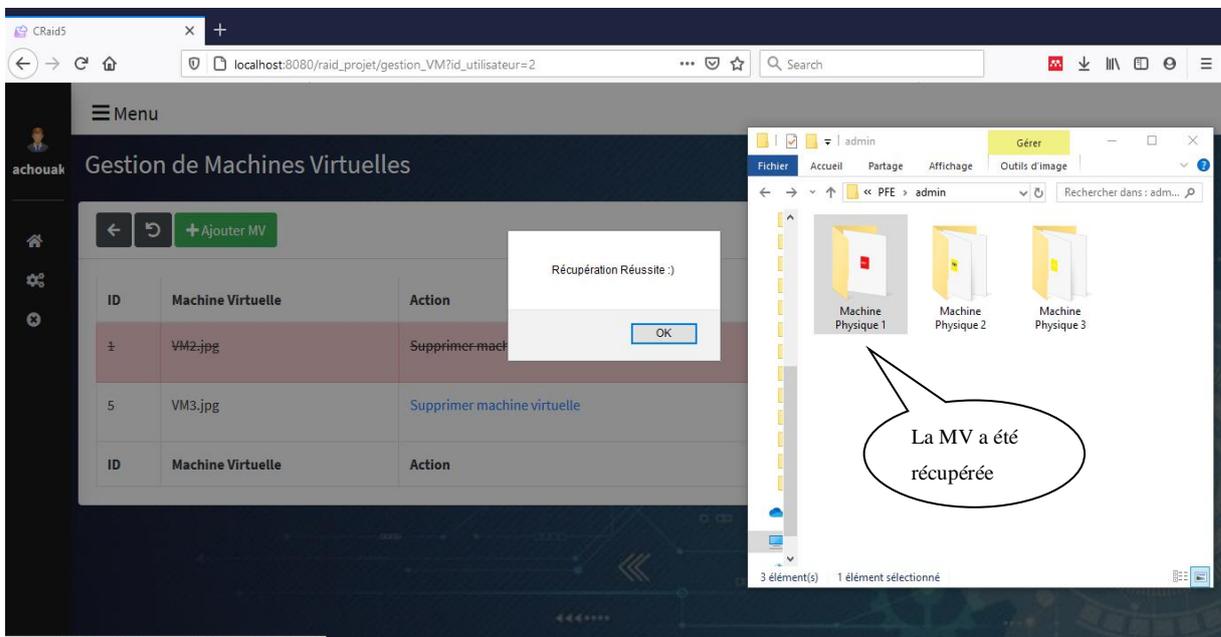


Figure 63 Récupération réussite de MV par le client "achouak"

## 5.6 Tests et Evaluations

Nous présentons dans cette partie quelques résultats des différents tests expérimentaux que nous avons obtenus par implémentation et par simulation pour comparer les approches et discuter les principaux résultats obtenus.

Afin de faire une évaluation de notre approche, on a choisi la technique de réplication comme référence, et qu'est considérée comme l'une des techniques les plus utilisée dans les plateformes distribué tel que le Cloud.

Afin de calculer l'espace totale consommé par chaque méthode, nous avons adapté les formules utilisées dans [36] et on a obtenu les formules suivantes :

✓ **La réplication** :  $CTotalSize = CStorageSize * 2.$  (1)

✓ **CRAID5** :  $CTotalSize = CStorageSize + \sum_{i=1}^p RSize_i$  (2)

Dans les formules (1) et (2), les variables sont :

- **CStorageSize** : espace total consommé par le Cloud à l'exclusion de l'espace de stockage utilisé pour la sauvegarde.
- **CBackupSize** : représente l'espace utilisé pour la sauvegarde.
- **CTotalSize** : représente l'espace total utilisé par le Cloud ( $CTotalSize = CStorageSize + CBackupSize$ ).
- **p** : représente le nombre total des sous-ensembles.
- **RSize<sub>i</sub>** : représente la taille des MVs hébergées dans le sous-ensemble.

### 5.6.1 Evaluation par implémentation

Dans l'implémentation de note approche proposée, on a pris le cas d'un seul ensemble et de multi sous-ensembles. Dans cette partie, nous supposons que nous avons un nombre de MP varie dans l'ensemble {3, 4, 5} et le nombre maximum de MV hébergées sur les MPs {6, 9, 12} respectivement.

Les trois cas sont représentés par des tableaux qui sont les suivants : Tableau8, Tableau9 et Tableau10. Chaque tableau a une représentation graphique illustrée dans les Figures suivantes : Figure 64, Figure 65 et Figure 66 respectivement. Pour les tests des trois, cas on a proposé que toutes les MVs (images statiques) de l'ensemble soient de taille 7.47 Ko avec des dimensions (width x height de 50x50).

#### ✚ Cas 1 (3 MPs)

Nb MV Stockage utilisé (Ko)	1	2	3	4	5	6
CRAID5	7.47	22.41	29.88	44.82	52.29	67.23
Réplication	14.94	29.88	44.82	59.76	74.7	89.64

Tableau 8 Comparaison CRAID5 vs Réplication "Cas1"

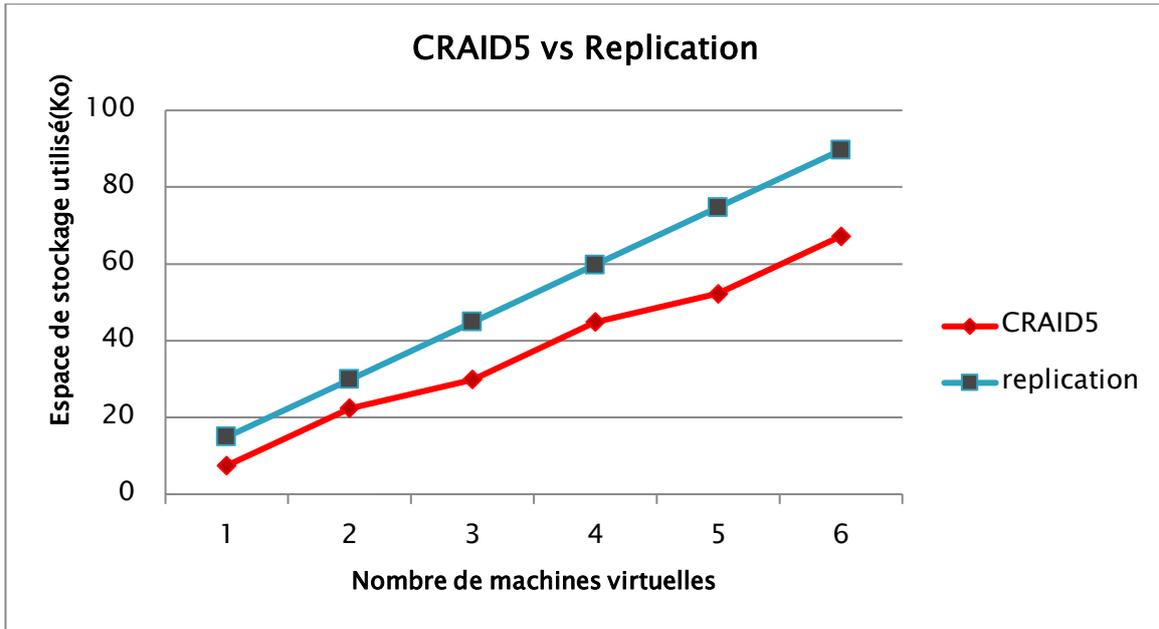


Figure 64 Comparaison CRAID5 vs Réplication "Cas1"

✚ Cas 2 (4 MPs)

Nb MV / Stockage utilisé (Ko)	1	2	3	4	5	6	7	8	9
CRAID5	7.47	22.41	29.88	37.35	52.29	59.76	67.23	82.17	89.64
Réplication	14.94	29.88	44.82	59.76	74.7	89.64	104.58	119.52	134.46

Tableau 9 Comparaison CRAID5 vs Réplication "Cas2"

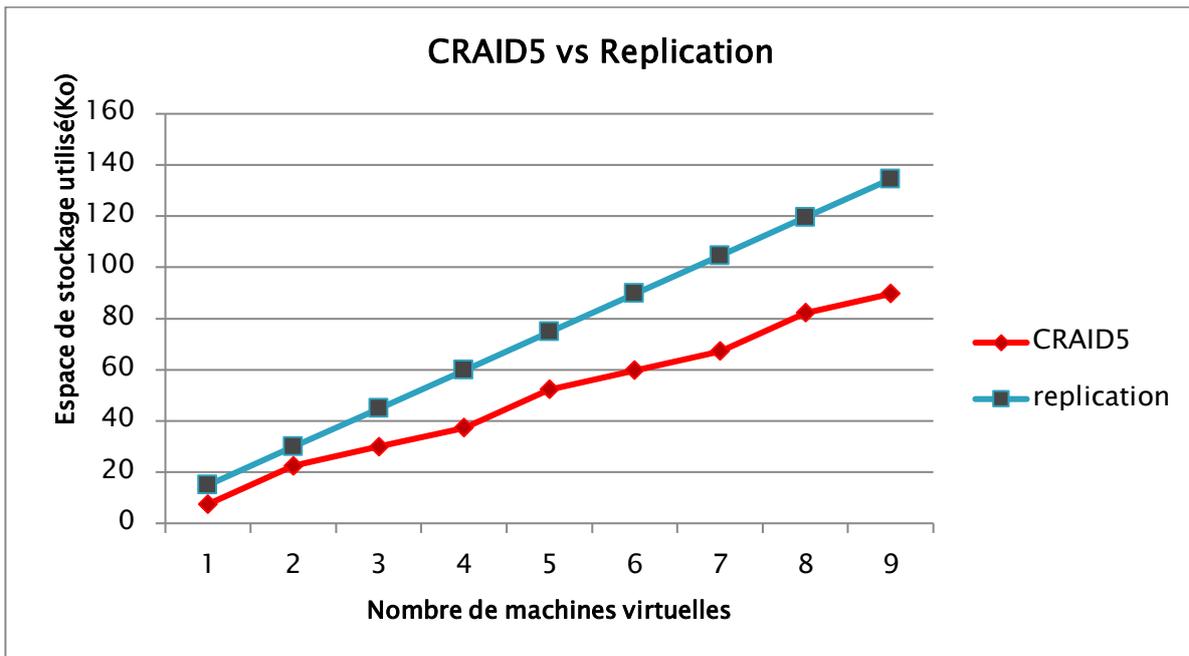


Figure 65 Comparaison CRAID5 vs Réplication "Cas2"

✚ Cas 3 (5 MPs)

Nb MV Stockage utilisé (Ko)	1	2	3	4	5	6	7	8	9	10	11	12
CRAID5	7.47	22.41	29.88	37.35	44.82	59.76	67.23	74.7	82.17	97.11	104.58	112.05
Réplication	14.94	29.88	44.82	59.76	74.7	89.64	104.58	119.52	134.46	149.4	164.34	179.28

Tableau 10 Comparaison CRAID5 vs Réplication "Cas3"

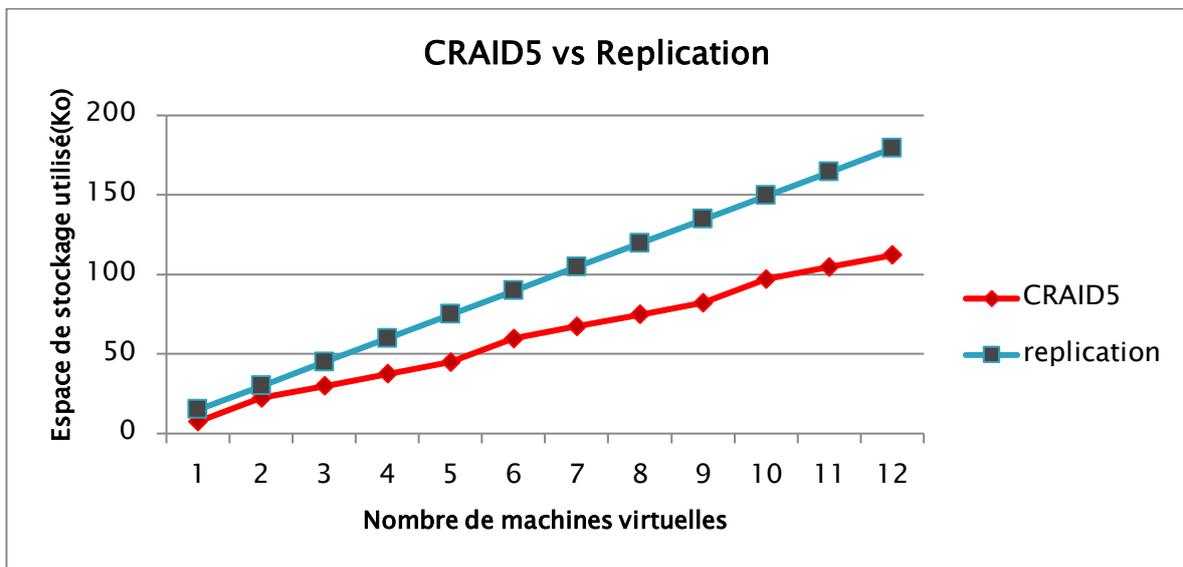


Figure 66 Comparaison CRAID5 vs Réplication "Cas3"

### 5.6.2 Evaluation par simulation

Dans la validation par implémentation de notre approche, on a pris le cas d'un seul ensemble et de multi sous-ensembles. Afin de faire une étude comparative avec un nombre d'ensembles/sous-ensembles plus large, on propose dans cette section de la faire par simulation. Dans cette partie, nous supposons que nous avons 5 types de MV en fonction de leurs tailles {200Mo, 500Mo, 1Go, 2Go, 4Go}, qui sont minuscules, petites, moyenne, grande et large MV respectivement. Il est considéré qu'il n'y a pas de contrainte sur l'espace de stockage disponible sur la MP, c'est-à-dire que l'espace est suffisant pour toutes les MV et la sauvegarde. A des fins d'évaluation, le nombre de MP varie dans l'ensemble {5, 10, 25, 50, 100}. Le nombre de MV hébergées sur chaque MP est généré de manière aléatoire. Les paramètres d'évaluation sont l'espace de stockage consommé par chaque politique et l'espace économisé par l'approche CRAID5 par rapport à la politique de réplication, tous ces détails sont mentionnés dans le tableau 11:

Nombre MP	MVs de taille 200Mo	MVs de taille 500Mo	MVs de taille 1Go	MVs de taille 2Go	MVs de taille 4Go	Somme de toutes les MVs
5	3	4	3	6	4	20
10	5	20	15	7	3	50
25	30	40	16	6	8	100
50	45	65	55	20	15	200
100	255	345	200	150	50	1000

Tableau 11 Paramètres de comparaison CRAID5 vs Réplication par simulation

Après avoir calculé l'espace de stockage consommé par chaque méthode, nous récapitulons les résultats obtenus pour les deux approches sur le Tableau 12.

	5MPs/20MVs	10MPs/50MVs	25MPs/100MVs	50MPs/200MVs	100MPs/1000MVs
CRAID5	44260	62876	96008	209612	954164
Réplication	68688	105968	174880	400440	1880600

Tableau 12 Résultat obtenus de comparaison CRAID5 vs Réplication par simulation

Ces résultats sont représentés sous forme graphique dans l'objectif d'améliorer la visualisation des résultats obtenus, elles sont illustrés sur la Figure 67.

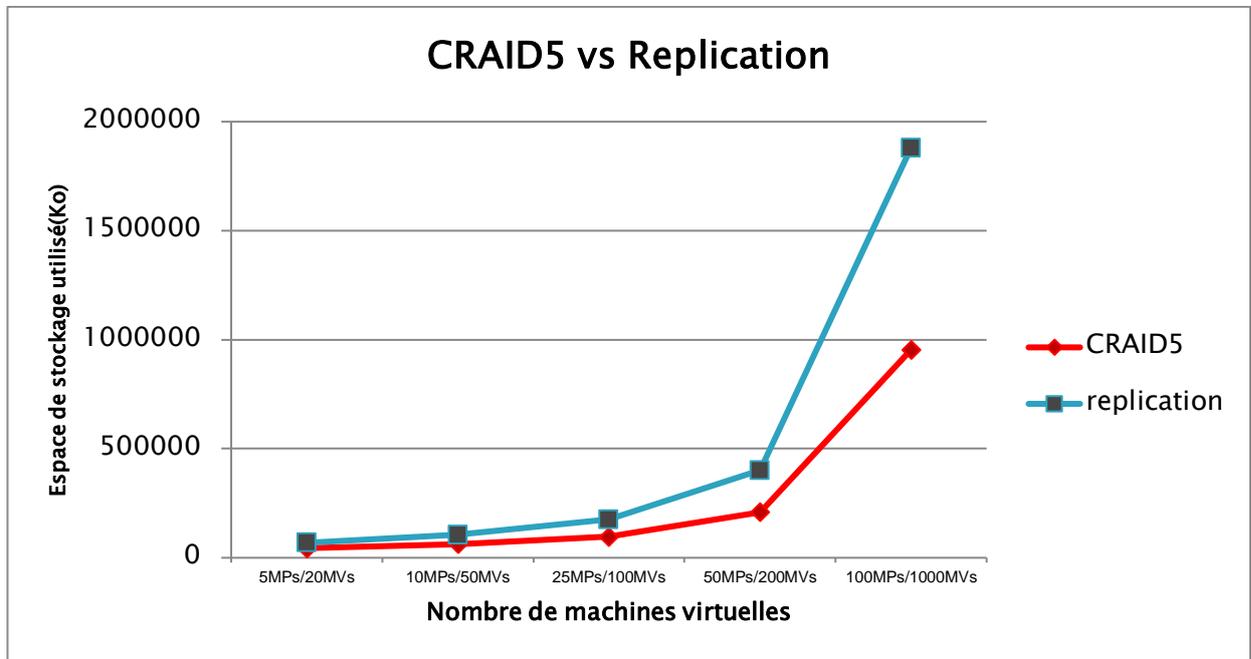


Figure 67 Comparaison CRAID5 vs Réplication par simulation

### 5.6.3 Discussion

Dans le cas de simulation, on a fait des tests sur cinq types de MV en fonction de leurs différentes tailles {200Mo, 500Mo, 1Go, 2Go, 4Go}, le nombre de MP varie dans l'ensemble {5, 10,

25, 50, 100}, le nombre maximum de MV hébergées sur les MPs était {20, 50, 100, 200, 1000} respectivement et le nombre de MV hébergées sur chaque MP a été généré aléatoirement. Alors que dans notre réalisation, on a fait des tests sur un seul ensemble de MV (des images de taille 7.47Ko avec des dimensions 50 x 50), le nombre de MP varie dans l'ensemble {3, 4, 5} et le nombre maximum de MV hébergées sur les MPs est dans l'ensemble {6, 9, 12} respectivement.

La représentation des résultats a été faite par des tableaux, une représentation graphique a accompagné chaque tableau. Dans les deux cas (par simulation et par implémentation), on constate que les courbes en rouge représente l'approche CRAID5 et en bleue représente l'approche de réplication. On remarque toujours que la courbe CRAID5 est au dessous de celle de réplication, d'où on conclut que l'espace de stockage de CRAID5 est optimisé par rapport à la réplication, donc les résultats obtenus par implémentation et par simulation montrent que l'approche CRAID5 minimise beaucoup de l'espace de stockage par rapport à la technique de réplication.

On remarque aussi que l'approche CRAID5 devient de plus en plus importante avec l'augmentation de nombre de ressources sur le Cloud.

Pour la réplication, la taille totale de l'espace utilisé pour la sauvegarde des MVs est exactement égale à la somme des tailles de toutes les MV. En revanche, pour CRAID5, l'espace de stockage dépend du nombre total de MV et MP.

Nous pouvons citer comme points faibles de CRAID5 :

- L'impossibilité de récupérer les ressources qui ne sont pas attachées à aucun SE, tel que le cas d'une seule ressource car le CRAID5 ne fonctionne qu'à partir de deux MVs par SE,
- L'impossibilité de récupérer les ressources dans le cas d'apparitions de deux ou plus craches simultanés dans le même SE.

## 5.7 Conclusion

Au fil de ce dernier chapitre, nous avons présenté la mise en œuvre de notre solution proposée dans un environnement Cloud. Le système réalisé est une implémentation entre plusieurs consommateurs de Cloud et un fournisseur de Cloud qui fournit le service de tolérance aux pannes, dans le but d'aider le client récupérer ses ressources en cas de pannes. On a assuré ce concept avec une optimisation de l'espace de stockage dans le Cloud à l'aide d'un niveau spécifique de RAID, spécialement RAID5.

Les résultats obtenus montrent que notre approche a réussi d'atteindre les objectifs désirés, les performances de notre approche sont meilleures par rapport à la technique de réplication.

## 1. Conclusion générale

L'informatique dans les nuages (Cloud Computing) est une tendance actuelle majeure pour répartir les traitements et les données de façon virtuelle sur des environnements d'exécution paramétrables. Il repose sur des principes d'élasticité et d'hétérogénéité de données, donc un nombre important de ressources de stockage sont utilisées et de grand volume de données sont stockées et hébergées tel que les MPs et les MVs. Ces pannes peuvent entraîner une perte de données et peuvent également coûter très cher car les services du Cloud sont, généralement, payants. Cette problématique nous a ramené à proposer une solution traitant la tolérance aux pannes dans le Cloud Computing, ce principe qui consiste à aider les systèmes à minimiser les pertes de données (des données des serveurs, BD ...etc.). Il y a deux types de tolérance aux pannes : réactive (le système détecte la panne après son occurrence et commence la récupération) et proactive (le système suppose que les composants ont un comportement prédictif et le système illustre des symptômes de panne avant la panne réelle).

Ce travail consiste à concevoir une solution efficace et performante de gestion des pannes réactive dans un réseau distribué de type Cloud, tout en optimisant l'espace de stockage dédié à la sauvegarde de données.

Afin d'atteindre nos objectifs, une nouvelle politique de tolérance aux pannes pour gérer les pannes de ressources dans un environnement Cloud a été proposée, nommée « CRAID5 ». « CRAID5 » hérite ses concepts de la technique RAID5, par conséquent, héritant des avantages offerts par RAID5, en particulier, en termes d'optimisation de l'espace de stockage par rapport aux systèmes de sauvegarde standard. Et afin de réaliser ce que nous avons étudié, nous avons implémenté un système permettant la tolérance aux pannes avec une minimisation de l'espace. Afin de mieux contourner toutes les fonctionnalités de chaque entité du système, on a fait une conception qui était nécessaire, dans ce cadre nous avons utilisé JavaEE comme un langage de programmation pour ses multiples avantages, nous avons fait des tests qui ont montré des résultats satisfaisants.

On a pu évaluer notre système à travers une comparaison avec la technique de réplication en termes de l'espace de stockage consommé par les deux méthodes. Notre comparaison a été faite par une implémentation et par une simulation. Dans les deux méthodes d'évaluation, les expériences montrent que notre solution proposée a de bonnes performances.

Les résultats de ce travail constituent les bases d'un travail à poursuivre et à améliorer pour une étude beaucoup plus approfondie qui pourra faire l'objet d'une thèse de doctorat.

## 2. Perspectives

Plusieurs perspectives peuvent être proposées afin d'améliorer notre approche (CRAID5), nous donnons comme exemples :

- Intégration de la notion de temps, et ceci pour permettre au système de tolérance aux pannes que nous avons proposé, de prévoir le temps de la reconstruction des machine crachées, et par conséquence permettre à l'utilisateur de connaitre le temps d'arrêt de ses MVs/Services.
- Il est utile d'adapter d'autres niveaux de RAID et étudier leurs performances, pour avoir un vue plus exacte sur les avantages/inconvénients de chaque niveau, comme il est possible d'utiliser une combinaison de plusieurs niveaux de RAID.
- Dans notre implémentation, nous avons pris on considération qu'un seul ensemble, il est envisageable d'élargir notre système réalisé pour qu'il supporte la gestion de multi-ensembles. Ainsi qu'il est utile de l'améliorer pour qu'il soit fonctionnel même dans le cas de panne de plus d'une ressource dans le même SE.
- Un autre travail futur à envisager, consiste à réaliser une véritable mise en place de la solution CRAID5 sur un environnement Cloud réel, et d'étudier ses performances afin de toucher mieux aux résultats.
- Dans le travail présenté dans ce document, nous avons focalisé sur un système distribué de type Cloud, il est intéressant de le rendre plus générique en l'adaptant sur autres environnements distribués, tel que l'Edge, HPC, Fog Computing ...etc.

**Bibliographie**

- [1] Onfais, B. Confais and B. C. Conception d' un système de partage de données adapté à un environnement de Fog Computing To cite this version. *Thèse de Doctorat*. 2018.
- [6] KAHANWAL, Dr, SINGH, Dr TP, *et al.* The distributed computing paradigms: P2P, grid, cluster, cloud, and jungle. *arXiv preprint arXiv:1311.3070*, 2013.
- [8] GUBBI, Jayavardhana, BUYYA, Rajkumar, MARUSIC, Slaven, *et al.* Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 2013, vol. 29, no 7, p. 1645-1660.
- [11] RAJAN, Sameer et JAIRATH, Apurva. Cloud computing: The fifth generation of computing. In : *2011 International Conference on Communication Systems and Network Technologies*. IEEE, 2011. p. 665-667.
- [12] K. Abdeslam, H. Mustapha. “ *Rapport Cloud Computing, université de henri poincaré* ”. 2011.
- [14] SIMMON, Eric. Evaluation of Cloud Computing Services Based on NIST SP 800-145. *NIST Spec. Publ*, 2018, vol. 500, p. 322.
- [15] S. Bennett, M. Bhuller, and R. Covington. Architectural strategies for cloud computing. White Paper in Enterprise Architecture : Oracle. AUG 2009.
- [16] O.ouchene, A. Chereit and. “Solution à base de broker dans un environnement d'inter -Cloud de type SaaS,”. 2019.
- [17] MESHARAM, Anjali D., SAMBARE, A. S., et ZADE, S. D. Fault tolerance model for reliable cloud computing. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2013, vol. 1, no 7, p. 600-603.
- [18] BUYYA, Rajkumar, VECCHIOLA, Christian, et SELVI, S. Thamarai. *Mastering cloud computing: foundations and applications programming*. Newnes, 2013.
- [21] Y.meslem S.debbas “*Ordonnancement et réplication de données dans le Cloud Computing*.”.2016
- [22] MICHON, Étienne, GOSSA, Julien, GENAUD, Stéphane, *et al.* Schlouder: A broker for IaaS clouds. *Future Generation Computer Systems*, 2017, vol. 69, p. 11-23.

- [24] RANGANATHAN, Kavitha et FOSTER, Ian. Identifying dynamic replication strategies for a high-performance data grid. In : *International Workshop on Grid Computing*. Springer, Berlin, Heidelberg, 2001. p. 75-86.
- [26] Deghbouch.H,Nedjadi.L, "Une stratégie de réplication de données pour gérer la tolérance aux pannes et l'équilibrage de charge dans le cloud computing",2018
- [27] G.Rubino, "Introduction à la sûreté de fonctionnement ( SdF )",2006.
- [28] AVIZIENIS, Algirdas, LAPRIE, J.-C., RANDELL, Brian, *et al.* Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 2004, vol. 1, no 1, p. 11-33.
- [29] L.Saïd, "Gestion De Tolerances Aux Fautes Dans Les Cloud Computing Avec CloudSim,". 2012, pp. 1–93.
- [30] MEROUFEL, Bakhta. *Tolérance aux pannes dans les grilles de données*. 2011. Thèse de doctorat. Université d'Oran1-Ahmed Ben Bella.
- [31] FLORIN, G. *La tolérance aux pannes dans les systèmes répartis*. Technical report, Laboratoire CEDRIC CNAM, 1996.
- [32] MEZOUER, Leila et MEKKIOUI, Nadjet. *Tolérances aux Pannes dans les infrastructures Cloud Computing*. Thèse de doctorat. 16-05-2018.
- [33] P.JALOTE.*Fault Tolerance in Distributed Systems, Prentice Hall these doctorate*. 1994.
- [34] NARASIMHAN, Priya, DUMITRAȘ, T. A., PAULOS, Aaron M., *et al.* MEAD: support for Real-Time Fault-Tolerant CORBA. *Concurrency and Computation: Practice and Experience*, 2005, vol. 17, no 12, p. 1527-1545.
- [35] NAIK, Namita. Stimulating proactive fault detection in distributed systems. 2007.
- [36] KHIAT, Abdelhamid. Cloud-RAIR: A Cloud Redundant Array of Independent Resources. *CLOUD COMPUTING 2019*.
- [37] KONG, Xiangzhen, HUANG, Jiwei, LIN, Chuang, *et al.* Performance, fault-tolerance and scalability analysis of virtual infrastructure management system. In : *2009 IEEE International Symposium on Parallel and Distributed Processing with Applications*. IEEE, 2009. p. 282-289.

- [38] JHAWAR, Ravi, PIURI, Vincenzo, et SANTAMBROGIO, Marco. A comprehensive conceptual system-level approach to fault tolerance in cloud computing. In : *2012 IEEE International Systems Conference SysCon 2012*. IEEE, 2012. p. 1-5.
- [39] ZHAO, Wenbing, MELLIAR-SMITH, P. M., et MOSER, Louise E. Fault tolerance middleware for cloud computing. In : *2010 IEEE 3rd International Conference on Cloud Computing*. IEEE, 2010. p. 67-74.
- [40] HILTUNEN, Matti et SCHLICHTING, R. An approach to constructing modular fault-tolerant protocols. In : *Proceedings of 1993 IEEE 12th Symposium on Reliable Distributed Systems*. IEEE, 1993. p. 105-114.
- [41] BALA, Anju et CHANA, Inderveer. Fault tolerance-challenges, techniques and implementation in cloud computing. *International Journal of Computer Science Issues (IJCSI)*, 2012, vol. 9, no 1, p. 288.
- [42] RADHAKRISHNAN, Ganesan. Adaptive application scaling for improving fault-tolerance and availability in the cloud. *Bell Labs Technical Journal*, 2012, vol. 17, no 2, p. 5-14.
- [44] Rémy, R. “ Réplication de données sur plate-forme hébergeant des applications métiers (NSS Gen2) ”. 2008, pp. 1–59.
- [66] Smile, «virtualisation & cloud principes mise en œuvre et outils opensource,» Smile, décembre 2012.

## Webographie

- [2] <https://zestedesavoir.com/tutoriels/2238/introduction-aux-systemes-distribues/>
- [3] <https://www.memoireonline.com/02/17/9572/Mise-en-oeuvre-dun-systeme-distribue-pour-lidentification-et-le-suivi-du-casier-judiciaire.html>
- [4] <https://www.lebigdata.fr/cluster-definition>.
- [5] <https://www.lebigdata.fr/hpc-definition>.
- [7] <http://www.metz.supelec.fr/metz/personnel/vialle/course/Mineure-HPC/index.htm>
- [9] <http://lhcatome.web.cern.ch/about/what-volunteer-computing>
- [10] <https://www.itforbusiness.fr/qu-est-ce-que-le-fog-computing-19728>
- [13] <https://fr.slideshare.net/ficel-hemza/introduction-au-cloud-computing-53416344>
- [19] [https://www.memoireonline.com/01/13/6713/m\\_Etude-et-mise-en-place-dune-solution-cloud-computing--privee-dans-une-entreprise-moderne-cas8.html](https://www.memoireonline.com/01/13/6713/m_Etude-et-mise-en-place-dune-solution-cloud-computing--privee-dans-une-entreprise-moderne-cas8.html).
- [20] <https://bluebearsit.com/iaas-paas-saas-definitions-et-differences>.
- [23] <https://www.webbazar.co.in/what-is-cloud-deployment-model/>.
- [25] <https://www.memoireonline.com/07/15/9194/Etude-sur-la-securite-du-cloud-computing.html>
- [43] <http://www.courstechinfo.be/Hard/Raid.html>
- [45] <https://prezi.com/xhojtwv0onhh/raid-redundant-array-of-independent-disks/>
- [46] <https://www.thinkmate.com/inside/articles/what-is-raid>
- [47] <https://www.lacie.com/gb/en/manuals/lrm/raid/>
- [48] <http://www.e-novatic.fr/tout-savoir-sur-le-stockage/>
- [49] <https://arabhardware.net/articles/raid-redundant-array-of-independent-disks>
- [50] <https://openclassrooms.com/fr/courses/626954-creez-votre-application-web-avec-java-ee>
- [51] <https://www.editions-eni.fr/open/mediabook.aspx?idR=98db25384380c51b79e46e170641bf45>
- [52] <https://desgeeksetdeslettres.com/web/xampp-plateforme-pour-heberger-son-propre-site-web>

- [53] <https://www.lemagit.fr/definition/MySQL>
- [54] <http://www.clashinfo.com/dico/definition-h/art45-html.html>
- [55] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203277-css-cascading-style-sheets-definition-traduction/>
- [56] [https://books.google.dz/books/about/Prenez\\_en\\_main\\_Bootstrap.html](https://books.google.dz/books/about/Prenez_en_main_Bootstrap.html)
- [57] <https://docplayer.fr/12281105-Familiarisation-avec-eclipse-netbeans.html>
- [58] <http://glossaire.infowebmaster.fr/javascript/>
- [59] <https://fr.dreamstime.com/illustration/datacenter.html>
- [60] [https://www.memoireonline.com/07/15/9194/m\\_Etude-sur-la-securite-du-cloud-computing5.html](https://www.memoireonline.com/07/15/9194/m_Etude-sur-la-securite-du-cloud-computing5.html)
- [61] <https://slideplayer.fr/slide/2452810/>
- [62] <http://glossaire.infowebmaster.fr/jquery/>
- [63] <https://www.linuxhelp.com/how-to-configure-raid6-in-centos-7>
- [64] <https://jaxenter.com/edge-cloud-content-delivery-race-147194.html>
- [65] <http://www.marche-public.fr/Terminologie/Entrees/confidentialite-des-donnees.htm>