

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR UNIVERSITY -

ANNABA-

UNIVERSITE BADJI MOKHTAR -ANNABA-



جامعة باجي مختار
- عنابة -

Faculté : Sciences de l'Ingénieur -Année 2018-

Département : Informatique

THESE

Présentée en vue de l'obtention du diplôme de **Doctorat en Sciences**

Déploiement et interrogation des services dans une grille de calcul (application sur la vidéo surveillance)

**Option : Intelligence
Artificielle**

Par

Dib Ahmed

Directeur de thèse	Sellami Mokhtar	Pr.	Université Badji Mokhtar Annaba, Algérie
Devant le jury			
Président	Khadir Mohamed Tarek	Pr.	Université Badji Mokhtar Annaba, Algérie
Examineur	Kazar Okba	Pr	Université de Biskra, Algérie
Examineur	Zarzour Hafed	MCA	Université de Souk-Ahras, Algérie

ملخص

تقدم هذه الوثيقة دراسة حول نشر واستجواب خدمات تطبيقات الوسائط المتعددة وخاصة تلك المخصصة لمراقبة الفيديو في الشبكة الحاسوبية. لقد تطور نشر الخدمات بشكل كبير مع ظهور تكنولوجيات جديدة مثل الحوسبة السحابية والشبكات الذكية. ونتيجة لذلك ، تتضمن الدراسة الحالية دمج حلول النشر الجديدة في بيئة الشبكة الحاسوبية. وهذا التكامل يساهم في الحفاظ على نفس بيئة تشغيل تطبيقات الوسائط المتعددة، ولكن مع المزيد من الحركة والاستغلال الأفضل للموارد الفيزيائية. وهذا يتيح تقديم تطبيقات الوسائط المتعددة كخدمة في بيئة الحوسبة الشبكية مثلما هو متاح في الحوسبة السحابية. ومع ذلك، فإن عددا من تطبيقات الوسائط المتعددة ليست مناسبة تماما للمعالجة المتوازية في بيئة موزعة مثل الشبكة والحوسبة السحابية. تعتمد هذه التطبيقات على المعالجة المتسلسلة لمكونات الفيديو المتتالية، مما يؤدي إلى تحديد الأداء وتدهور الدقة عند تنفيذ هذه التطبيقات. توفر هذه الدراسة حلاً لهذا النوع من الوظائف لتوفير خدمة للمراقبة بالفيديو تتلاءم تماماً مع البيئة الموزعة.

الكلمات المفتاحية

الشبكة الحاسوبية، الحوسبة السحابية، النظام الموزع، النشر، الخدمة، خدمة الشبكة الحاسوبية، حاوية الخدمة مراقبة الفيديو، معالجة الفيديو، معالجة الصور، MapReduce، Kubernetes، YARN، Apache Hadoop.

Résumé

Ce document présente une étude de déploiement et d'interrogation des services d'applications multimédias et spécialement celles dédiées à la vidéosurveillance dans une grille de calcul. Le déploiement des services a exponentiellement évolué avec l'apparition de nouvelles technologies telles que les nuages de calcul et les grilles intelligentes. Du fait, la courante étude consiste à l'intégration de nouvelles solutions de déploiement dans un environnement de grille de calcul. Cette intégration permet de garder le même environnement d'exécution d'applications multimédias, mais avec plus de dynamisme et une meilleure exploitation des ressources physiques. Cela a permis la fourniture d'applications multimédias comme services dans un environnement de grille de calcul en analogie avec les systèmes de vidéosurveillance comme service (VaaS) au niveau des nuages de calcul. Cependant, un nombre de fonctionnalités d'applications multimédias ne sont pas bien adaptées à un traitement parallèle dans un environnement distribué tels que les grilles et les nuages de calcul. Ces fonctionnalités sont basées sur le traitement séquentiel de consécutives composantes des vidéos, ce qui résulte une limitation de performances et une dégradation de l'exactitude lors de l'accomplissement de ces fonctionnalités. La présente étude fournit une solution pour ce type de fonctionnalité afin de fournir un service de vidéosurveillance bien adapté à un environnement distribué.

Mots clés

Grille de calcul, nuage de calcul, système distribué, déploiement, service, service de grille, conteneur de service, MapReduce, Apache Hadoop, YARN, Kubernetes, vidéosurveillance, traitement de la vidéo, traitement d'images.

Abstract

This document presents a study of deployment and interrogation of multimedia application services and especially those dedicated to video surveillance in a grid computing environment. Deployment of services has exponentially evolved with the advent of new technologies such as cloud computing and smart grids. Therefore, the current study involves the integration of new deployment solutions into a computing grid environment. This integration makes it possible to keep the same execution environment of the multimedia applications, but with more dynamism and a better exploitation of the physical resources. This allowed the provision of multimedia applications as a service in a computing grid environment in analogy with video surveillance systems as a service (VaaS) on cloud. However, a number of features of multimedia capabilities are not well suited to parallel processing in a distributed environment such as grids and computational clouds. These features are based on the sequential processing of consecutive video components, causing a performance limitation and degradation of accuracy when performing these features. This study provides a solution for carrying out this kind of functionality to provide a video surveillance service that is well suited to distributed environments.

Keywords

Grid computing, cloud Computing, Distributed System, Deployment, Service, Grid Service, Service Container, MapReduce, Apache Hadoop, YARN, Kubernetes, CCTV, Video Processing, Image Processing.

Table des matières

Introduction générale.....	1
1. Contexte de la thèse :	1
2. Problématique	3
3. Objectif du la thèse	4
4. Organisation de la thèse	4
Chapitre 1 : les grilles de calcul et les services	7
1. Introduction.....	7
2. Les Systèmes distribués	7
2.1. Modèles de systèmes distribués.....	9
2.1.1. Les clusters de calcul.....	9
2.1.2. Les grilles de calcul.....	9
2.1.3. L'informatique en nuage	11
3. Architecture d'une grille.....	15
3.1. Couche fabrique.....	15
3.2. Couche connectivité	17
3.3. Couche ressources	18
3.4. Couche collective	19
3.5. Couche application	20
4. Quelques Intergiciels pour les grilles.....	20
4.1. Globus.....	20
4.2. UNICOR.....	21
4.3. Legion.....	23
5. Les Services de grille et les technologies en relation	24
5.1. Architecture orientée Service (SOA).....	24
5.2. Transfert d'Etat Représentatif (REST).....	24
5.3. Architecture ouverte de grille de service (OGSA)	25

5.4. Les services de grille	27
6. Conclusion	28
7. Références.....	29
Chapitre 2 : déploiement et interrogation des services dans une grille.....	32
1. Introduction.....	32
2. Serveurs d'applications	32
3. Les conteneurs	34
3.1. Les conteneurs classiques Java/JEE	35
3.2. L'emprisonnement.....	36
3.3. Les Conteneurs Linux (LXC)	36
3.4. Docker	38
4. Orchestrateurs de conteneurs	40
4.1. Kubernetes	40
4.1.1. Architecture	42
4.2. Apache YARN.....	43
4.2.1. Architecture	43
4.2.2. Soumission d'application et déploiement dans YARN.....	44
5. Déploiement, interrogation, et exécution des services dans une grille	46
5.1. Travaux en relation	46
5.1.1. Globus	46
5.1.2. HAND	49
5.1.3. DynaGrid.....	51
5.1.4. UNICORE	52
5.1.5. Autres travaux	54
5.2. Critiques.....	54
6. Proposition de solution	55
6.1. Externalisation des conteneurs de services.....	56

6.2.	Utilisation des images pour la mise à jour des conteneurs de ressources.....	57
6.3.	Ajout d'un module extensible pour utiliser les ressources externes.....	59
7.	Conclusion	61
8.	Références.....	62
Chapitre 3 : la vidéo surveillance et les systèmes distribués.....		65
1.	Introduction.....	65
2.	Définition du domaine	67
2.1.	La surveillance.....	67
2.2.	La contre surveillance.....	68
2.3.	La vidéosurveillance.....	68
2.4.	La télésurveillance	69
2.5.	La vidéosurveillance IP	70
2.6.	La vidéosurveillance intelligente.....	70
2.7.	La vidéosurveillance en Algérie	71
3.	Les composantes d'un système de vidéosurveillance	72
3.1.	Les équipements d'acquisition	72
3.2.	La transmission.....	75
3.3.	Le traitement.....	77
4.	Architecture des systèmes de vidéosurveillance.....	78
4.1.	Architecture centralisée	78
4.2.	Architecture distribuée	79
4.2.1.	Utilisation des caméras intelligentes distribuées.....	79
4.2.2.	Utilisation des serveurs enregistreurs.....	82
5.	Serveurs de stockage des systèmes de vidéosurveillance	83
5.1.	Le stockage distribué des données de la vidéosurveillance.....	83
5.1.1.	SAN (Storage Area Network)	84
5.1.2.	NAS (Network-Attached Storage)	85

5.1.3.	DAS (Direct-Attached Storage)	87
5.1.4.	Le stockage dans le nuage	87
6.	Le traitement distribué des données de vidéosurveillance.....	89
6.1.	Les systèmes de vidéosurveillance traditionnels :.....	89
6.2.	La vidéosurveillance comme service (VSaaS) :	91
7.	Conclusion	94
8.	Références.....	95
Chapitre 4 : proposition de solution pour le traitement de vidéo dans un environnement distribué.....		
		98
1.	Introduction.....	98
2.	Composantes de vidéo	98
3.	Problématique	99
4.	Travaux en relation avec la détection des limites des shots.....	100
4.1.	Accélération des calculs de VSBD	101
4.2.	VSBD à base de GPU et multi CPU	101
4.3.	Traitement de vidéos à base de MapReduce	102
5.	Présentation de la méthode proposée	103
5.1.	Frameworks de base	103
5.1.1.	MapReduce.....	103
5.1.2.	Apache Hadoop	105
5.2.	Préparation des données vidéos	106
5.2.1.	Extraction des frames	107
5.2.2.	Génération des fichiers de séquence	108
5.2.3.	L'ingestion des fichiers de séquence sur le HDFS.....	108
5.3.	La phase Map.....	109
5.3.1.	Récupération des données d'entrée	109
5.3.2.	Extraction des caractéristiques des frames.....	109

5.3.3.	Mesure de similarité entre les frames.....	112
5.4.	La phase de réduction	114
5.4.1.	Regroupement et tri des frames ayant la même mesure de similarité intermédiaire	114
5.4.2.	Détection des limites des shots.....	115
6.	Expérimentation.....	118
6.1.	Environnement d'exécution.....	118
6.1.1.	Ressources physiques	118
6.2.	Ressources logicielles.....	119
6.3.	Evaluation de l'exactitude de la solution.....	120
6.3.1.	Base de vidéos utilisée	120
6.3.2.	Analyse de la mesure de similarité intermédiaire	122
6.3.3.	Métriques d'évaluation de l'exactitude	124
6.3.4.	Comparaison de l'exactitude de la méthode proposée avec des solutions existantes.....	124
6.3.5.	Test de la méthode VSBD proposée sur des données volumineuses	126
6.4.	Evaluation de la scalabilité de la solution	127
7.	Conclusion	130
8.	Références.....	131
	Conclusions et perspectives	135

Index des figures

Figure 1. Architecture en couche d'une grille informatique.....	15
Figure 2. Relation entre WSRF, OGSA et les Services Web.....	27
Figure 3. Architecture et conteneurs JavaEE.	35
Figure 4. Architecture du conteneur LXC.....	37
Figure 5. L'API Docker.	38
Figure 6.L'architecture de Docker.....	39
Figure 7. Schéma illustratif de la structure d'une POD.	42
Figure 8.Vue illustrative de l'architecture de Kubernetes.	43
Figure 9. Interaction entre les composantes de YARN pour l'accomplissement d'une application.	45
Figure 10. Exécution des services déployés dans Globus.....	48
Figure 11. Conteneur des services de grille externalisé dans un conteneur Docker et géré par Kubernetes.....	57
Figure 12. Vue architecturale de la solution proposée.	60
Figure 13.Architecture générique d'une caméra intelligente.....	81
Figure 15: Un système de vidéosurveillance distribué basé sur les enregistreurs.....	83
Figure 16: Un exemple d'un FCSAN.	85
Figure 17. Structure d'une vidéo.	99
Figure 18. Étapes de préparation des données et leur ingestion sur l'HDFS.....	107
Figure 19. Le flux des traitements dans la phase Map.	113
Figure 20. Cas de frames bien triés composants deux shots.	116
Figure 21. Cas de frame manquant dans un shot.....	116
Figure 22. Cas de frame isolé.....	116
Figure 23. Flux du processus de la phase de réduction.	117
Figure 24 : un résumé de vue des 300 premiers frames de la vidéo anni003.mpg	122

Figure 25: Variation de la similarité intermédiaire calculée sur un intervalle de 300 images de la vidéo Anni003.mpg.	123
Figure 26: Comparaisons de la précision entre le schéma proposé et d'autres approches.	125
Figure 27: Comparaisons de rappel entre le schéma proposé et d'autres approches.	125
Figure 28: Comparaisons de F-score entre le schéma proposé et d'autres approches.	125
Figure 29. Comparaison de F-score entre la méthode proposée et une méthode basée sur MapReduce classique.	127
Figure 30. Variation du temps d'exécution en fonction du nombre de blocs et des nœuds impliqués dans les calculs.	129

Index des tableaux

Tableau 1: Ressources physiques utilisées pour l'évaluation de l'exactitude de l'application.	119
Tableau 2: Ressources logicielles.	119
Tableau 3: Vidéos utilisées pour le test de l'exactitude de la solution.	121
Tableau 4. Données volumineuses utilisées pour le test de la méthode proposée.	126
Tableau 5. Caractéristique des nœuds de l'environnement.	128

Introduction Générale

Introduction générale

1. Contexte de la thèse :

Le besoin en ressources de calcul, de stockage et de mémoire est en croissance continue. Cela est due à l'apparition des équipements sophistiqués pour la capture du son, d'image et de vidéo en haute définition. De plus, le contenu multimédia est devenu un élément essentiel dans la vie personnelle comme professionnelle des gens, ce qui a conduit à une production d'une quantité énorme de ce contenu. Par conséquent, la gestion et l'analyse des données multimédias de haute définition et de quantité importante en ligne est devenu un défi pour les systèmes multimédias classiques vu le besoin de puissantes ressources de calcul et de stockage.

Les applications de la vidéosurveillance font partie des applications multimédias. Elles offrent une diversité de fonctionnalités liées aux contextes de leur utilisation telles que l'archivage des vidéos, la recherche, la détection, le tracking et la reconnaissance d'objets, d'évènements ou de personnes. Pour une énorme quantité de données multimédias, il est très difficile d'assurer un bon fonctionnement avec un respect des contraintes liées au temps de traitement et à la consommation de ressources de calcul. Les systèmes de vidéosurveillance les plus utilisés se basent sur l'utilisation d'un seul serveur dédié, ou sur un environnement de ressources limitées.

La puissance de calcul et de stockage sont des ressources indispensables pour l'accomplissement des tâches liées au traitement de données multimédias. Les grilles présentent un moyen de calcul très puissant, ils fournissent les ressources nécessaires pour le traitement d'une grande quantité de données multimédias. Ils regroupent des ressources hétérogènes de nature et géographiquement distribuées pour fournir les ressources requises. L'utilisation des grilles masque toutes hétérogénéités de ressources et laisse l'utilisateur concentré sur ses tâches principales d'administration et de développement du métier. L'utilisateur spécifie ses besoins en ressources dans un langage spécialisé, et lance ses tâches à base de ce qu'il a réservé comme ressources. Cependant, une couche logicielle appelée intergiciel positionnée sur les ressources de la grille permet d'assurer la

communication et la coopération entre les composantes de cette dernière. Un intergiciel est installé en dessus des systèmes d'exploitation et de plusieurs outils et APIs tels que la JVM, SSI, SSh, Shell, FTP, etc. Ces intergiciels fournissent des services indispensables pour le bon fonctionnement de la grille. Les services offerts permettent d'assurer :

- La gestion des données : les services des intergiciels assurent l'intégrité, l'intégralité, l'unicité et la disponibilité de données.
- La gestion des ressources : les grilles de calcul sont vues sous forme d'un ensemble de ressources de différentes natures. Un service pour gérer l'ensemble des ressources est indispensable pour la synchronisation, la coopération et le bon fonctionnement des opérations lancées sur une grille.
- La gestion de la sécurité : la sécurité de ressources et de données de la grille est une priorité pour les clients. En se basant sur des politiques, des standards et des protocoles normalisés, la grille offre des services assurant la sécurité des opérations sur l'ensemble des composantes de la grille.
- La gestion des informations : afin de bien utiliser une ressource physique, on bien connaitre ses capacités et son état actuel. Des services dédiés à l'extraction des informations de ressources sont fournies pour affecter la bonne tâche sur la bonne ressource pour une durée de temps bien déterminée.

Afin de bien bénéficier des services offerts par les grilles de calcul, un déploiement d'applications est nécessaire. Ce déploiement consiste à l'hébergement des applications sur les différents nœuds qui constituent la grille. Un meilleur déploiement permet une bonne découverte de ces applications, ainsi qu'un bon approvisionnement des ressources nécessaires pour un bon fonctionnement.

Avec l'apparition de nouvelles technologies de traitement distribué de l'information telles que le nuage de calcul, le brouillard de calcul et les grilles intelligentes, le déploiement au niveau des grilles de calcul peut être largement amélioré. Il peut bénéficier des techniques de déploiement utilisées dans les nouvelles technologies pour garantir une dynamique lors de la mise à jour des applications déjà déployées, ainsi qu'une élasticité dans l'approvisionnement des ressources. Cependant, leur interrogation garde les mêmes caractéristiques que celles dans un environnement de grille classique.

2. Problématique

Les grilles de calcul sont destinées aux applications qui traitent les données en batch et qui sont très gourmandes en termes de ressources pour des durées de temps importantes. Par contre, dans le cas des systèmes de vidéosurveillance, on n'a pas besoin de réserver énormément de ressources de calcul pour une longue durée. Par contre, on doit avoir un service prêt à l'utilisation en n'importe quel moment, et capable de fournir à la demande des ressources illimitées. Cela nous fait penser à la réponse à la question :

- 1- Pourquoi réserver énormément de ressources pour une longue durée si elles ne sont utilisées que périodiquement ? comment la facturation peut être équitable de points de vue clients et fournisseurs ?

Les services déployés sur une grille de calcul peuvent être fournis par divers fournisseurs et consommés par multiples clients. Dans le cas où une mise à jour d'un de ces services de grille, ou un redémarrage d'un serveur est nécessaire, on doit répondre aux questions suivantes :

- 2- Comment assurer une disponibilité permanente des services déployés sur une grille ?
- 3- Comment assurer qu'une instance d'un service fait référence à la dernière version déployée de ce dernier ?

Les systèmes de la vidéosurveillance représentent un type spécial des applications multimédias. L'utilisation de ces systèmes dans des environnements distribués nécessite une implémentation de fonctionnalités de traitement d'image, de parole, et de vidéo. Certaines de ces fonctionnalités ne sont pas adaptées à une exécution parallèle. Un exemple de ces fonctionnalités est la segmentation temporelle des vidéos qui est à la base de plusieurs autres opérations d'analyse du contenu vidéo. Cela est dû aux besoins de traiter les consécutifs frames d'une vidéo de manière séquentielle. Cela pose la problématique de les traiter séquentiellement et donc on perd en performance, ou de manière parallèle et donc on perd en exactitude de résultats. La question sur laquelle on doit répondre est la suivante :

- 4- Comment peut-on fournir un système de vidéosurveillance performant et d'une haute exactitude dans un environnement distribué ?

3. Objectif du la thèse

L'objet du travail que nous avons menu le long de cette thèse est de :

- 1- Réaliser un déploiement d'un service de grille destiné aux systèmes de vidéosurveillance. Le déploiement à proposer doit être dynamique, automatique et facile à gérer.
- 2- L'exécution des services déployés doit être performante et consomme les ressources de manière rationnelle.
- 3- Proposer une technique de traitement de vidéo adaptée au environnements distribués en assurant une scalabilité des opérations avec un taux d'exactitude important.

4. Organisation de la thèse

La démarche qu'on a adopté pour la réalisation de ce travail est reflétée dans l'organisation de ce document. Ce manuscrit est structuré en quatre chapitres.

Le premier est consacré à la présentation du principal domaine de recherche qui est représenté par les technologies en liaison avec les grilles de calcul. Une présentation des systèmes distribués, des services, et des services de grille ainsi que leurs caractéristiques est effectuée.

Le deuxième chapitre consiste dans un premier lieu en la présentation des nouvelles technologies liées avec le déploiement des services dans des systèmes distribués. Dans un second lieu, le chapitre réserve une partie pour la présentation des techniques classiques du déploiement adoptées par les intergiciels existants, ou par des études dans le domaine. Les techniques existantes sont critiquées à base d'un nombre de critères et une proposition d'une architecture modulaire est présentée sur base d'une hybridation entre les systèmes de déploiement classiques et entre des techniques récentes utilisées par de nouvelles technologies de big data.

Le troisième chapitre focalise sur le domaine de la vidéosurveillance, les dispositifs de capture de vidéos, les serveurs d'analyse et de traitement, la manière dont ces vidéos sont stockées, et la distribution des traitements et de stockages des vidéos dans des environnement de big data.

Le Quatrième chapitre est consacré à la solution proposée concernant le traitement distribué des vidéos à l'aide des frameworks existants dans le domaine big data. L'application sur laquelle porte le chapitre est la segmentation temporelle des vidéos. L'environnement distribué sur lequel la solution est testée est un système basé sur le modèle de programmation MapReduce qui permet un traitement parallèle des vidéos. Dans un deuxième lieu, on présente l'aspect expérimental. Les résultats de la méthode de traitements distribués à base de MapReduce sont présentés et discutés. Une comparaison entre la méthode proposée et des études de l'état de l'art est aussi présentée afin de montrer l'avantage de la technique proposée.

Chapitre 1 : les grilles de calcul et les services

Chapitre 1 : les grilles de calcul et les services

1. Introduction

L'informatique est un outil qui permet aux scientifiques du monde entier de repousser davantage chaque jour les limites de la connaissance. Les sciences actuelles telles que l'astrophysique, la physique des particules élémentaires, les sciences de la terre, la génétique et bien d'autres encore, ne cessent de réclamer encore plus de puissance informatique. Les besoins en calcul et en stockage ont été augmenté d'une manière importante. Pour relever ces défis scientifiques, les ingénieurs ont pensé à de nouveaux modèles informatiques. Fédérer plusieurs ordinateurs au sein d'une même entité virtuelle et regrouper tous ces ordinateurs afin d'en faire un super calculateur virtuel. C'est ainsi qu'est né le concept de grille informatique. Elle regroupe un ensemble de techniques visant une meilleure exploration de ressources à très grande échelle. Les grilles de calcul sont classées parmi les systèmes concurrents qui se basent sur des architectures avec plusieurs unités de traitement. Ces dernières coopèrent dans le but d'exécuter en parallèle une ou plusieurs tâches.

2. Les Systèmes distribués

Le développement continu des ordinateurs a mené à l'apparition de puissantes machines. Ces dernières sont dotées de performants processeurs composés de plusieurs cœurs. Elles achèvent de manière parallèle les tâches qui leurs sont affectées. D'un autre côté, l'avancement dans le domaine des réseaux a permis d'avoir des LANs qui offrent une bande passante à l'échelle de plusieurs Giga byte par seconde. Cela a permis de réduire le temps nécessaire de transmission des données au sein d'un réseau de plusieurs unités de traitement. Les LANs à haut débit ont ouvert les portes pour produire des supercalculateurs et des centres de calcul qui lisent des données en entrée, les traiter et produire d'autres données qui auront des destinations locales ou distantes. De l'autre côté, le développement des réseaux étendus (WAN) a offert la possibilité de produire des systèmes capables de partager les mêmes tâches malgré la distance géographique qui les sépare. Les centres de

calcul sont liés par des réseaux WAN haut débit pour former chacun une ressource accessible par un système dite distribué.

Un système distribué est une collection d'éléments de calcul autonomes qui apparaît à ses utilisateurs comme un seul système cohérent (van Steen et Tanenbaum, 2016). Cette dernière définition souligne deux axes importants :

- Une collection des éléments de calcul autonomes : ce point met l'accent sur une agrégation d'un ensemble de ressources de calcul, appelées aussi nœuds, qui peuvent être de différents types y compris des supercalculateurs, des ordinateurs de bureau, des smartphones, des machines virtuelles, ou même de simples capteurs déployés sur un environnement de travail et qui traitent de l'information. Chaque ressource de traitement autonome peut être utilisée dans un contexte privé ou collectif. L'ensemble de ressources est interconnecté via des supports de liaison physiques filaires ou sans fil.
- Un seul système cohérent : l'utilisation de l'ensemble des ressources distribuées est gérée par un système qui offre une vue simplifiée de l'ensemble tout en masquant la distribution et l'hétérogénéité des ressources. De plus, ce système doit fournir les outils nécessaires pour gérer les tâches, les données, et l'état global du système. La gestion des tâches qui s'exécutent sur des nœuds hétérogènes et distants inclut l'ordonnancement, la remontée des exceptions, l'affectation des tâches aux différents nœuds et la collecte des résultats de ces tâches. Cependant, la gestion des données inclut le transfert et la réplication de ces données.

Néanmoins, la supervision de toutes les opérations est assurée par un système d'information qui offre les éléments nécessaires pour le bon fonctionnement du système distribué. Il assure l'actualisation de l'état du système global par rapport aux changements fréquents de la topologie à chaque fois où un nœud est connecté ou déconnecté. De plus, il offre un support pour la gestion de la charge des éléments du système, ce qui aide à avoir un équilibrage de charge pour mener à bien les tâches en cours d'exécution.

D'un autre côté, la plupart des systèmes distribués sont dotés d'un module de visualisation d'états. Ce module assure l'interface entre le système et les utilisateurs en leur fournissant des outils de suivi de tâches, des données sur le support de stockage

distribué, de l'état global de la charge du système, et des fichiers journaux qui détaillent l'historique de toutes manipulations du système.

2.1. Modèles de systèmes distribués

Plusieurs modèles de système distribué ont été créés tout au long d'une vingtaine d'année pour des éventuelles amélioration en performance, une réduction des coûts et pour répondre à certains besoins fonctionnels.

2.1.1. Les clusters de calcul

Le cluster computing est l'un des modèles des systèmes distribués qui a été développé pour réduire le coût élevé de s'offrir des HPC¹ qui été limité aux riches et puissantes organisations. L'émergence des clusters a été mené par des projets académiques tels que Beowulf (Becker et al., 1995), Berkeley NOW (Culler et al., 1997), et HPVM (Anderson et al., 1995).

Un cluster est une collection d'ordinateurs parallèles et/ou distribués interconnectés entre eux à l'aide d'un réseau à haut débit. Ces ordinateurs se coopèrent pour accomplir une tâche donnée. Le cluster assure une haute disponibilité des ressources en adoptant la stratégie de duplication des nœuds, si l'un de ces nœuds tombe en panne, son dupliquant se charge de continuer les tâches interrompues. Quand un utilisateur lance une requête, il aura l'impression d'interagir avec un seul système cohérent. Cependant sa requête sollicite les ordinateurs interconnectés tout en masquant tous détails de distribution. La tâche de l'utilisateur est ensuite devisée et exécutée sur les nœuds du cluster en partageant la charge de travail et en offrant des performances élevées.

2.1.2. Les grilles de calcul

Une grille est un type de système parallèle et distribué qui permet le partage, la sélection et l'agrégation de ressources autonomes géographiquement distribuées de manière dynamique au moment de l'exécution en fonction de leur disponibilité, capacité, performance, coût et qualité de service (Chetty et Buyya, 2002). La grille de calcul se base sur un intergiciel pour diviser et répartir des morceaux d'un programme entre plusieurs

¹ HPC : en anglais 'High-performance computing', est une technologie basée sur un ordinateur appelé supercalculateur qui offre les performances les plus élevés possible.

ordinateurs. Le calcul en grille implique un calcul distribué, qui peut également impliquer l'agrégation de systèmes à base de grappes à grande échelle. La taille d'une grille peut varier d'un petit réseau de postes de travail informatiques au sein d'une entreprise à de grandes collaborations entre de nombreuses entreprises et réseaux.

Une des définitions des grilles de calcul qui a été donnée par son fondateur Ian Foster (Foster et Buyya, 2001) consiste à vérifier trois critères majeurs (Foster, 2002) :

- Coordination des ressources de contrôle non centralisées : Une grille intègre et coordonne les ressources et les utilisateurs de différents domaines de contrôle.
- Utilisation des protocoles et des interfaces standards, ouverts, et à usage général : Une grille est fondée à la base de protocoles et d'interfaces multi-usages qui abordent des problèmes fondamentaux tels que l'authentification, l'autorisation, la découverte et l'accès aux ressources.
- Délivrance des qualités de service non triviales : une grille de calcul doit assurer une meilleure qualité de service en termes de temps de réponse, bande passante, disponibilité des ressources et sécurité.

Les grilles de calcul reposent sur la notion des organisations virtuelles (VO). Une VO est un concept présenté par un ensemble d'individus et d'équipements informatiques qui partagent des ressources et des services soumis à des politiques de sécurité. Les utilisateurs groupés ont les mêmes besoins au sein d'une même VO. Cette dernière possède des règles et des stratégies de sécurité adaptées le mieux à ce groupe. La VO fournit une multitude de services selon les besoins de leurs utilisateurs et en suivant des stratégies de sécurité et des politiques définies par les concepteurs.

Un utilisateur peut être affecté à plusieurs organisations virtuelle. Pour se faire, il doit s'authentifier auprès d'une de ces VO, ensuite, il peut accéder aux autres VO dont il est membre, et donc a le droit de bénéficier des ressources et des services disponibles dans toutes les VO. La durée de vie des organisations virtuelles est variante. Ainsi leurs créations et leurs gestions ouvrent la voie vers d'autres travaux de recherches qui incluent les projets de localisation, d'affectation et de partage de ressources dans une VO.

Parmi les importantes caractéristiques de la grille (Robertazzi, 2017), on cite :

1. L'autonomie : Chaque site de l'ensemble constituant la grille possède le contrôle sur ses propres ressources.
2. L'hétérogénéité : Les composantes de la grille fournissent des interfaces standardisées pour surmonter la diversité d'un site.
3. La gestion des données : les données dans une grille sont considérées comme des ressources. Des services spéciaux offre des fonctionnalités de transfert, d'archivage et de réplication.
4. La personne clé pour les grilles est l'utilisateur, pas le propriétaire de la ressource.

2.1.3. L'informatique en nuage

Un nuage de calcul est une collection de ressources interconnectés et virtualisés. L'ensemble des ressources est dynamiquement provisionné et libéré avec un minimum d'effort de gestion et d'interaction avec les fournisseurs de services. D'après l'agence NIST², les nuages de calcul sont caractérisés par cinq principales caractéristiques, trois modèles de services et quatre modèles de déploiement (Mell et Grance, 2011). Les cinq principales caractéristiques sont les suivantes :

1. Libre-service : Un consommateur peut s'approvisionner de manière unilatérale des ressources fournies par des fournisseurs de services, selon les besoins, sans nécessité d'aucune intervention humaine avec chacun de ces fournisseurs.
2. Accès large réseau : Les ressources des nuages sont disponibles sur les réseaux de leurs fournisseurs et sont accessibles via des mécanismes et protocoles standards. Ces ressources sont accessibles aux utilisateurs par le biais de plates-formes hétérogènes (téléphones mobiles, tablettes, ordinateurs portables, montres intelligentes, etc.).
3. Agrégation de ressources : Les ressources informatiques des fournisseurs de services sont agrégées pour servir plusieurs consommateurs à l'aide d'un modèle à

² NIST : National Institute of Standards and Technology, une agence américaine qui a comme mission de promouvoir l'économie en développant des technologies, la métrologie et des standards de concert avec l'industrie.

locataires multiples, avec différentes ressources physiques et virtuelles. Ces ressources sont attribuées dynamiquement et réaffectées en fonction de la demande des consommateurs.

4. **Elasticité rapide** : Les capacités des ressources attribuées aux clients peuvent être dynamiquement ajustées aux besoins et à la demande de manière élastique. Cela produit une évolution rapide des performances de ces ressources en fonction de la demande. Cette évolution automatique des capacités des ressources donne l'impression que ces dernières sont illimitées, et peuvent être approvisionnées en n'importe quelle quantité et à tout moment. L'élasticité peut être horizontale ou verticale. Le premier cas est matérialisé par une augmentation du nombre des ressources requises pour offrir la capacité demandée par un client. Par exemple l'ajout d'une unité de calcul pour fournir une fréquence de CPU plus élevée ou une barrette mémoire pour atteindre la taille requise. D'un autre côté, l'élasticité verticale consiste en l'augmentation de la capacité d'une ressource sans modifier le nombre des ressources déjà attribuées ou de faire référence à d'autres ressources externes. Cela peut être vu comme la modification de la configuration d'utilisation d'une ressource afin de permettre à un utilisateur de l'exploiter avec les nouveaux paramètres.
5. **Service Mesuré** : la consommation des services de cloud peut être mesurée afin d'assurer une transparence lors de la facturation tant pour le fournisseur que pour les clients. Pour chaque type de ressource, une unité de mesure appropriée est associée afin de pouvoir contrôler et optimiser automatiquement l'utilisation de ces ressources.

Les trois modèles de service offerts par les cloud sont :

- **Software comme Service (SaaS)** : les fonctionnalités offertes au consommateur consistent à utiliser les applications du fournisseur fonctionnant sur une infrastructure de cloud. Les applications sont accessibles à partir de divers dispositifs clients par le biais de protocoles et des standards tels que HTTP, FTP, SSL, etc. Le consommateur ne gère ni ne contrôle l'infrastructure cloud sous-jacente.

- Platform comme Service (PaaS) : ce modèle fournit des instances de middlewares et de contexte d'exécution. Des exemples des services fournies sont les serveurs de Base de données, les conteneurs et les serveurs d'application. Un client peut bénéficier de ce modèle en déployant ses propres applications compatibles avec les APIs, les serveurs et les langages de programmation supportés par la platform offerte par le fournisseur. D'un autre côté, le client ne peut pas gérer l'infrastructure sous-jacente représentée par l'infrastructure du cloud. Windows Azure³, Heroku⁴, Google App Engine⁵, Sont des exemples de PaaS.
- Infrastructure comme Service (IaaS) : ce modèle offre aux consommateurs la possibilité de contrôler et gérer des ressources qui définissent le type de système d'exploitation, les performances des ressources que ce soit une puissance de calcul, taille de mémoire vive, capacité de stockage ou une bande passante d'une ressource réseau. Ce modèle est généralement réservé aux administrateurs systèmes et/ou réseaux. Amazon Web Services⁶ (AWS) et Microsoft Azure⁷ sont des exemples connus de IaaS.

Et pour finir avec les cloud, on site brièvement les quatre modèles de déploiement qui sont 1) le cloud privé, 2) le cloud communautaire, 3) le cloud public, et 4) le cloud hybride. Quel que soit le modèle de déploiement du cloud, son infrastructure peut-être soit on-premise ou non. Sa gestion peut être assurée par sa propre organisation qui l'exploite, par un acteur externe, ou par les deux ensembles.

La principale différence entre les modèles de déploiement des cloud est l'étendue de son exploitation. Le modèle privé est caractérisé par une utilisation limitée à un nombre de clients appartenant à une seule organisation. Le modèle communautaire étend cette utilisation à plusieurs organisations qui ont des préoccupations communes que ce soit commercial, éducatif, social ou autre. Le modèle de cloud public quant à lui est caractérisé

³ Windows Azure : disponible sur azure.microsoft.com, est la plate-forme applicative en nuage de Microsoft.

⁴ Heroku : disponible sur www.heroku.com, créé en 2007 et racheté par la suite par Salesforce.com, il offre une dizaine de contextes d'exécution qui facilitent le développement des applications en cloud en utilisant node.js, Java, Spring, Clojure, Python, Scala, API Facebook, PHP, etc.

⁵ Google App Engine : disponible sur console.cloud.google.com, offre une platform de conception et d'hébergement d'applications web basées sur les serveurs de Google.

⁶ Amazon Web Services : aws.amazon.com, fournit plusieurs services tels que EC2 et S3, le premier est un service qui offre un accès aux serveurs de calcul. Le second offre un service de stockage.

⁷ Microsoft Azure : son site officiel est azure.microsoft.com, composé de plusieurs éléments qui incluent Windows Azure, Hadoop On Azure et Windows Azure Media Services.

par un accès au grand public à un niveau d'exploitation intense. Cependant, le modèle de déploiement hybride est celui résultant de la combinaison de plusieurs autres modèles de déploiement. D'un autre côté, le niveau de sécurité de données et des opérations est différent tous en dépendant du modèle du cloud adopté. Ce niveau de sécurité est meilleur pour le modèle privé où l'accès aux données est limité à un ensemble de client de confiance, et les opérations sont traitées sur les serveurs internes ce qui réduit les risques d'attaque externe et d'altération des données échangées.

En outre que les grilles et les grilles, les approches de calcul distribué ne cessent pas d'évoluer. De nouvelles approches connaissent du succès actuellement et occupent une part importante du marché des big data. Parmi ces nouvelles approches, on cite :

- La grille de bureau de calcul (Desktop Grid Computing) (Cérin et Fedak, 2012) : cette approche permet d'utiliser un ensemble d'ordinateur de bureau d'une organisation donnée comme des ressources de calcul. Ces ressources permettent d'assurer un support d'exécution distribué de tâches à faible priorité ou de tâches à priorité normal lorsque les ressources ne sont pas utilisées.
- Le calcul volontaire (volunteer computing) (Anderson, 2010) : elle repose sur les mêmes principes que la grille de bureau de calcul à l'exception de la provenance des ressources qui s'étale à l'échelle public. Cela offre plus de ressources en nombre et en performance, mais augmente les risques des attaques ou d'exploitation maléfique.
- Le brouillard de calcul (Fog computing - fogging) (Bonomi et al., 2012) : cette approche tente de surmonter les limites des nuages qui exigent une connexion fiable à Internet. Le brouillard fonctionne avec les nuages dans la partie des bords qui a une interaction directe avec le monde réel. Au niveau des bords, on trouve des capteurs de données de différents types selon le champ applicatif. Ces capteurs sont combinés avec des unités de traitement que ce soit des FPGA⁸ ou des processeurs spécialisés à faible consommation énergétique. Le brouillard permet la collecte et le pré-traitement des données en place, et transmet la synthèse de ces données au serveurs distant. Cette technique permet de réduire la quantité de données à

⁸ FPGA (Field-Programmable Gate Array) : des circuits intégrés en silicium reprogrammables.

transmettre aux serveurs, ce qui améliore les performances globales du système et assurant une réduction de la charge des serveurs et l'utilisation de la bande passante. De plus, il favorise l'autonomie de ses composants en déminuant la quantité de données échangée.

3. Architecture d'une grille

Une grille informatique est une infrastructure composée de ressources matérielles et logicielles, de mécanismes de communication et d'un ensemble de services fournis généralement par un Middleware. Ce dernier a comme objectif le contrôle des activités et des interactions dans la grille. Cette infrastructure peut être visualisée comme une architecture en couche qui illustre les principales composantes d'une grille. La figure 1 montre l'architecture en couche d'une grille.

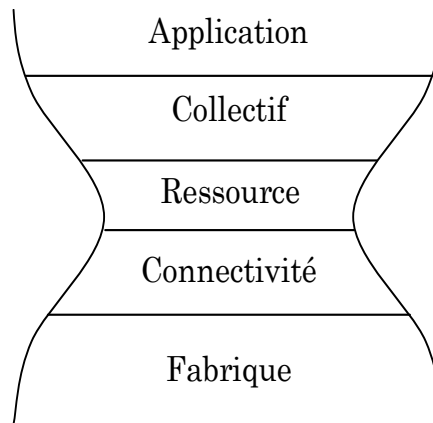


Figure 1. Architecture en couche d'une grille informatique (Foster, 2001 ; Foster, 2005).

Chaque couche représente une abstraction d'un ensemble de fonctionnalité du système. Elles peuvent communiquer directement avec les composants et services de n'importe quelle autre couche de niveau inférieure dans la hiérarchie.

3.1. Couche fabrique

Cette couche est liée directement aux matériels qui constituent la base de l'infrastructure de la grille et fournit toutes les ressources. A ce niveau, on distingue deux types de ressources, physique et logique. La première est représentée par l'ensemble des ressources matérielles telles que des unités de calcul, mémoire et des supports de transmission réseau. Les unités de calcul sont de différentes formes, des plus simples comme un simple processeur aux plus complexes comme des clusters ou des centres de

calcul. De l'autre côté, les ressources logiques sont présentées par des entités logicielles qui assurent la gestion de la grille. Les services d'annuaires, de bases de données, des systèmes de fichiers distribués, et des serveurs virtuels sont des exemples de ressources logiques fournies par la couche fabrique.

Lorsque l'une des couches supérieures fait appel à une ressource de la couche fabrique, des composantes logicielles de la couche agissent directement sur les ressources physiques et logiques. Elles lancent des opérations spécifiques selon la demande des utilisateurs. Les ressources eux-mêmes implémentent un mécanisme d'introspection pour fournir les informations qui décrivent leurs structures, leurs fonctionnalités et leurs états. Ces informations sont nécessaires pour la bonne exploitation des ressources sollicitées. Des fonctionnalités intrinsèques de la couche fabrique peuvent être citées pour différents types de ressources :

- Ressources de calcul : ce type de ressource implémente des fonctionnalités permettant de récupérer des informations sur les caractéristiques matérielles comme la fréquence et le modèle d'un processeur ou la taille d'un espace mémoire. De même, des informations sur les caractéristiques de la partie logicielle est aussi fournie en incluant la charge d'un système, l'état des files d'attente et la liste des ressources de calcul disponibles. De même, ces ressources sont dotées d'un ensemble de mécanismes capables de lancer des applications, de contrôler l'état des processus et de gérer les ressources à allouer.
- Ressources de stockage : Ces ressources permettent la transmission ou la récupération des fichiers et d'effectuer des opérations d'entrée/sortie sur des supports de stockage dédiés. De plus, les ressources de stockage fournissent un mécanisme de gestion de ressources impliquées par l'échange et la transmission des données comme la gestion de l'espace mémoire, la bande passante des disques, la bande passante réseau, la CPU et la charge du système. Des fonctions d'introspection sont utiles pour fournir des informations sur les caractéristiques logicielles et matérielles utilisées durant le transfert des données.
- Ressources réseau : ces ressources fournissent des fonctionnalités d'introspection pour déterminer les caractéristiques et la charge du réseau. Les interfaces des

routeurs sont principalement utilisées car elles représentent une source d'information fiable. Un routeur peut fournir des informations concernant la topologie du réseau et les algorithmes de routage utilisés grâce aux protocoles qu'il peut implémenter. Une ressource réseau peut tirer les meilleurs chemins entre deux points du réseau en terme du coût de transmission, de bande passante, ou du plus courts chemin.

3.2.Couche connectivité

Cette couche implémente les protocoles nécessaires pour l'authentification et l'interrogation des ressources dans la grille. Les protocoles de communication implémentés sont des standards connus comme TCP/IP, UDP, HTTP, DNS, etc. Avec une hétérogénéité croissante des ressources, un nombre important de protocoles a été implémentés, et il reste toujours la possibilité d'incrémenter ce nombre selon les besoins de la communauté et des nouvelles technologies. On note que le protocole Grid Security Infrastructure (GSI) (Web Services Architecture, 2018) est l'appui de toutes transactions faites sur la grille.

Les protocoles d'authentification sont implémentés au même niveau que les protocoles de communication. Leur utilité est d'authentifier les ressources de la couche fabrique, leurs origines et leurs propriétaires. De même, des opérations de cryptographie et une variété de techniques d'authentifications sont implémentées afin d'assurer une exploitation légale et sécurisée de l'ensemble des ressources offertes par l'infrastructure.

Cependant, la couche connectivité repose sur un ensemble de mécanismes pour assurer la sécurité d'échange des données, d'authentification, et d'exploitation des ressources, ci-dessous les principaux mécanismes :

- Single sign-on (SSO) : c'est un mécanisme qui simplifie la procédure d'authentification. Il permet une authentification unique pour l'exploitation de l'ensemble des ressources dont un utilisateur possède le droit d'y accéder. Les identifiants d'un client seront propagés sur l'ensemble des ressources de la grille. De même, le mécanisme SSO permet de fermer de manière automatique toutes les sessions ouvertes d'un utilisateur qui viens de se déconnecter.
- Délégation : les processus lancés par un utilisateur portent les mêmes autorisations et droits que leurs propriétaires. Ainsi que pour les applications et services

interrogés par ces processus prennent les mêmes droits. Ce mécanisme est connu par ‘délégation des privilèges’.

- Relation d’approbation : si un processus fait appel à plusieurs ressources de différents sites, et il n’est authentifié qu’auprès d’un seul, la relation d’approbation assure l’accès à toutes les ressources demandées en faisant confiance aux politiques de sécurité du premier site.
- Intégration : dans une grille, chaque site ou fournisseur de services implémente une solution interne pour sécuriser ses ressources. La politique de sécurité générale de la grille doit être capable de s’interfacer avec toutes les différentes solutions spécifiques adoptées par les sites.

3.3.Couche ressources

Cette couche fait appel aux services des deux couches précédentes pour gérer les ressources et collecter les informations qui les concernent de point de vue individuel. La gestion des ressources implique l’utilisation des services de contrôle, d’énumération et d’initialisation des états des ressources. A titre d’exemple, on peut citer le protocole (Grid Resource Access and Management -GRAM-) et le protocole GridFTP. Le premier est utilisé pour allouer et gérer les ressources de calcul. Cependant, le protocole GridFTP sert à gérer l’accès aux données et le transfert à très grande vitesse de ces données.

Ils existent deux classes principales des protocoles implémentés dans cette couche :

- Les protocoles d’informations : ils permettent l’accès aux informations des ressources que ce soit des informations statiques ou dynamiques. Statiques comme celles concernant un calculateur, sa puissance, sa capacité de stockage, sa configuration et sa stratégie de sécurité. Dynamique comme la charge globale d’un système, le nombre de processus lancés, l’espace disponible dans un média de stockage, etc.
- Les protocoles de gestion : permettent de négocier l’accès aux ressources partagées. Ils offrent la possibilité de spécifier des réservations, à voir les conditions de qualité des services et des opérations qui peuvent être effectuées (ex : la création de processus et l’accès aux données distantes). Cette classe de protocoles et de services

contrôle l'exécution des opérations et remonte les erreurs vers les services des couches supérieurs.

3.4.Couche collective

Contrairement à la couche précédente, cette couche possède une vue globale des ressources. Elle implémente des services et des protocoles capables de gérer un nombre important de ressources et gérer leurs interactions. Cette couche est responsable de l'ordonnancement et la co-allocation des ressources demandées. Elle possède plusieurs informations globales sur les ressources. Ces informations lui donnent la possibilité de sélectionner et d'allouer la ressource la plus appropriée pour la réalisation d'une tâche demandée. Elle s'occupe également de la réplication des données, la récupération et le traitement des alertes remontées par la couche ressources. En d'autres mots, la couche collective se charge de l'orchestration des ressources disponibles sur la grille. Comme services, on cite :

- Annuaire : Le service d'annuaire fournie par cette couche joue un rôle important dans la gestion des ressources de la grille. Il est représenté par une base de données contenant toutes les caractéristiques des ressources. Avec ces informations, le courtier localise la ressource la plus appropriée pour répondre à la requête lancée, il consulte l'annuaire et transmet les informations nécessaires pour la localisation physique de la ressource.
- Services d'allocation et d'ordonnancement : afin d'allouer une ressource à un processus, le service allocateur consulte l'annuaire pour trouver celle la plus appropriée. L'ordonnanceur prend l'initiative de décider quel processus doit être exécuté, et sur quelle ressource, et le moment exacte de son exécution selon la politique de l'ordonnanceur adoptée.
- Services de contrôle et de diagnostic : ils donnent à l'utilisateur la possibilité de contrôler l'état des ressources et s'assurer de leur bon fonctionnement.
- Services de gestion de données : les processus lancés ont besoin d'un ensemble de données pour accomplir leurs tâches. Les services de gestion de données permettent la récupération et le transfert fiable et sécurisé de ces données.

- Services de réplication de données : leur objectif est de maximiser les performances de la grille. Une réplication est effectuée dans le but de minimiser le coût et le temps d'accès à un bloc de données distant. Le répliquer en local ou à un endroit proche des opérations de calcul permet de réduire le temps d'accès à la ressource et d'augmenter sa disponibilité.

3.5.Couche application

Elle contient les applications déployées sur la grille. Cette couche peut interagir directement avec les autres couches inférieures. Les couches Collectif et Ressources sont sollicitées pour localiser les ressources demandées par une application. La couche Connectivité sera l'outil d'authentification. Et la couche Fabrique pour y accéder. Une application peut être un service spécifique ou une application cliente qui tente à exécuter une partie métier hébergé dans un serveur d'exécution.

4. Quelques Intergiciels pour les grilles

Un intergiciel de grille est une infrastructure logicielle qui permet une meilleure exploitation d'une grille et une meilleure virtualisation des ressources. Son objectif est de masquer toute hétérogénéité et d'assurer l'interopérabilité entre les composantes de la grille. Il repose sur des standards et fournit des applications, des services et des APIs⁹ pour implémenter les fonctionnalités nécessaires. Les intergiciels sont conçus de manière modulaire pour répondre à deux besoins de base qui sont la facilité de déploiement et l'utilisation à la demande. Toute amélioration ou évolution de fonctionnalités de la grille nécessite une augmentation dans un ou plusieurs modules de l'intergiciel. Parmi les intergiciels les plus reconnus dans le domaine des grilles sont :

4.1.Globus

Globus¹⁰ (Foster , 2001), il a été développé sur la base du réseau I-WAY¹¹ aux Etats-Unis dans le laboratoire Nationale de l'Argonne par l'équipe de Ian Foster. En plus de supporter des applications demandant des performances élevées, il peut gérer des

⁹ API : application programming interfaces, est un ensemble de définitions de routines, de protocoles et d'outils pour la construction de logiciels.

¹⁰ Projet disponible à l'adresse : <https://www.globus.org/>

¹¹ Le réseau I-WAY : Information Superhighway, est un réseau de communication mondial à haut débit capable de transporter des données, de la voix, de la vidéo et d'autres services à travers le monde en utilisant différentes technologies telles que les satellites, la fibre optique et les télécommunications cellulaires.

organisations virtuelles qui ont des préoccupations communes. Il fournit une infrastructure logicielle permettant aux applications de manipuler les ressources hétérogènes, géographiquement distribuées comme étant une machine virtuelle unique. La boîte à outil Globus offre des services de base de sécurité, d'informations, de communications et de gestion des données. De plus, elle implémente l'architecture OGSA¹² qui définit la manière de partage et de distribution de l'information, ce qui assure l'interopérabilité entre les ressources de la grille. Les principaux modules de Globus sont :

- GSI (Globus Security Infrastructure) : un module d'authentification basé sur un ensemble de standards tels que SSO, TLS, certificat X.509, et la cryptographie à clé publique ;
- MDS (Monitoring and Discovery Service) : responsable de la collection des informations des différentes ressources. Ce module est largement sollicité par la plupart des services de la grille afin de leur fournir les états actuels des ressources et la charge globale du système ;
- GRAM (Globus Resource Allocation Manager) : il est responsable du contrôle et d'allocation des ressources de calcul. Il est utilisé pour la gestion de l'exécution des tâches sur ces ressources. Cette gestion inclut le lancement, la reprise et la mise en fin des tâches.
- GridFTP (Grid File Transport Protocol) : chargé de la gestion des données, il permet le transfert sécurisé des fichiers. Il est à la base de fonctionnement d'autres services tels que RFT(Reliable File Transfert) et RLS (Replica Location Service).

4.2.UNICOR

C'est un projet Européen qui a pour objectif d'assurer un accès sécurisé et uniforme à des plateformes de calcul intense (Almond et Snelling, 1999).

UNICORE est un système à quatre niveaux : client, passerelle, services, et système cible. Tous les composants à l'exception du système cible sont implémentés en Java (Schuller et al., 2014). Les clients utilisent HTTPS pour se connecter à UNICORE via la

¹² OGSA : Open Grid Services Architecture, est un ensemble de normes qui étend les services Web et l'architecture orientée services à l'environnement de grille informatique.

passerelle. UNICORE fournit des clients en interface graphique et en ligne de commande. De plus, des applications personnalisées peuvent être créées en utilisant les API client Java de UNICORE.

La passerelle est un proxy inverse HTTPS qui permet un accès sécurisé à la couche service. Des informations sur l'identité des clients connectés sont aussi transmises à la couche service pour assurer la sécurité des transactions et des opérations. Un exemple des informations transmises est le certificat SSL généré par le client et qui contient d'autres informations personnelles et professionnelles tels que le nom, l'adresse email, le nom de domaine, le groupe de travail, etc. De plus, des informations concernant l'environnement de travail des clients sont aussi transmises. Elles incluent des informations sur la localisation géographique, l'adresse IP, les propriétés de connexion utilisée par le client, les informations sur l'application de connexion, etc.

Cependant, dans le noyau de UNICORE, on trouve le serveur UNICORE/X qui fournit le backend d'exécution et communique avec le niveau système cible, ainsi qu'avec un ensemble d'interfaces de services. Les services de base de la couche 'services' comprennent principalement un service de registre qui fournit des informations sur les services déployés sur l'infrastructure, les services de soumission, de gestion des tâches, d'accès et de transfert de fichiers qui font partie des services métier de l'intergiciel.

UNICORE/X reçoit les demandes/requêtes du client via la passerelle. Il vérifie les identifiants du client et les autorisations qui lui sont accordées, et appelle le service approprié. Les autres services de la couche 'Service' sont accessibles par UNICORE/X via des interfaces. Ces services sont déployés dans le conteneur de UNICORE qui est appelé 'environnement de services UNICORE' (USE). Le conteneur USE fournit un serveur Web, des outils de sécurisation, des APIs et des services de grilles y compris leurs fichiers de configuration et des classes de base sur lesquelles les services sont créés. Le serveur UNICORE/X peut s'interfacer avec d'autres systèmes externes pour fournir des fonctionnalités supplémentaires tels que l'authentification et l'autorisation des utilisateurs. La base de fonctionnement du serveur UNICORE/X est le moteur d'exécution XNJS (Schuller et al., 2006), qui communique avec le niveau système cible.

Le niveau système cible (Target System Interface -TSI-) est utilisé pour soumettre et gérer des Jobs aux ressources externes. De plus, il assure la transmission des données et toutes opérations d'entrée/sortie nécessaires pour l'exécution des Jobs soumis. Le TSI est un module qui s'exécute sur deux parties qui sont le cluster hébergeant le niveau des services, et le cluster qui représente la ressource externe. Cette dernière peut être soit un cluster de calcul ou de stockage.

Sur les clusters de calcul, le TSI joue le rôle d'un client de l'un des gestionnaires de ressources connus tels que YARN¹³, Slurm¹⁴, ou Torque¹⁵. Cependant, sur des clusters de stockage, il s'interface avec des espaces et des systèmes de stockage comme le cas de Amzon S3¹⁶ et HDFS¹⁷.

4.3.Legion

Legion (Natrajan et al., 2002) est un projet abandonné vu l'orientation des nouvelles technologies vers les cloud. Il est cité dans cette section car il a été l'un des intergiciels les plus connus dans le domaine de grille de calcul. Il a été considéré comme un méta-système à base d'objet développé à l'Université de Virginia. Il fournit aux utilisateurs une infrastructure logicielle permettant aux systèmes hétérogènes, distribués et performants d'interagir uniformément. Contrairement à Globus, l'Intergiciel Legion, ainsi que ses modules sont vus comme étant des objets. Cela facilite beaucoup la manipulation des relations d'instanciation, d'héritage et de polymorphisme. Legion s'interface entre le système d'exploitation des utilisateurs et les ressources distribuées de la grille. Chaque utilisateur aura l'impression de ne voir que ses propres ressources, mais en réalité, il accède aux différentes ressources réparties sur le réseau tout en implémentant les mécanismes et les protocoles de sécurité et d'accès définis par l'Intergiciel.

¹³ YARN, l'acronyme de Yet Another Resource Negotiator, est un gestionnaire de ressources intégrable avec les Framework de bigdata. Consultable sur le lien : <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>.

¹⁴ Slurm, Acronyme de Simple Linux Utility for Resource Management, est un ordonnanceur de tâche destiné principalement aux HPC. Disponible sur le lien : <https://slurm.schedmd.com/>.

¹⁵ Torque, un autre gestionnaire de ressources qui fournit un contrôle sur les travaux par lots et les ressources informatiques distribuées. Le projet est disponible sur <http://www.adaptivecomputing.com/products/torque/>.

¹⁶ Amazon Simple Storage Service (Amazon S3) : est un service web de stockage, évolutif, conçu pour la sauvegarde et l'archivage en ligne des données et des programmes d'application. Accessible sur le lien : <https://aws.amazon.com/s3/>.

¹⁷ HDFS (Hadoop Distributed File System), est le système de fichier distribué d'Apache Hadoop. La documentation de son architecture est disponible sur le lien : https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

5. Les Services de grille et les technologies en relation

Dans cette section, on va présenter brièvement les principales technologies et standards sur lesquels se basent les services opérants sur les grilles de calcul.

5.1. Architecture orientée Service (SOA)

La réalisation des systèmes à base de services est émergée et devenu la manière la plus répondeuse pour les systèmes distribués. Cela est soutenu par quelques caractéristiques telles que le faible couplage et la supporte de la variance d'implémentation de ces services. Cependant, l'architecture orientée service (SOA) définit la manière de conception des systèmes logiciels par l'intégration de nouveaux services ou de ceux qui existent déjà. Ainsi, SOA sert à offrir des composantes interopérables grâce à la nature faiblement couplée des services et l'ensemble des protocoles et standards sur lesquelles elle repose. Selon W3C¹⁸, SOA est présentée sous forme d'une architecture de système distribué caractérisée par une vue logique de ses composantes, une structure descriptive et un style de communication basée sur l'échange des messages (Web Services Architecture, 2018). De plus SOA repose en particulier sur le style d'architecture REST.

5.2. Transfert d'Etat Représentatif (REST)

Acronyme de 'Representational State Transfer', REST est un style d'architecture des applications hypermédias destinées aux systèmes distribués. Il a été adopté par les géants de l'informatique à cause de sa simplicité et de sa facilité à publier et à consommer par des clients. Les services qui implémentent REST sont appelés services RESTful. Ils sont des ressources à consommer comme n'importe quelle autre entité. Un nom unique appelé Uniform Resource Identifier (URI) est attribué à chaque ressource afin de la distinguer et de la localiser.

L'interaction avec les services RESTful est effectuée via le protocole http, ce dernier permet d'implémenter les opérations de base pour manipuler une ressources telles que la création, la lecture, la modification et la suppression. Les interactions dans REST sont sans états dans le sens où la signification d'un message ne dépend pas de l'état d'une conversation (suite d'interactions entre ressources). Du fait, les transactions à état sont

¹⁸ W3C : World Wide Web Consortium, est une internationale communauté où les organisations membres, un personnel qui travail à plein temps et le public travaillent ensemble pour développer des standards Web.

basées sur le principe de transmission explicite de l'état en se servant d'existantes techniques telles que l'utilisation des cookies et les champs d'information cachés dans les formulaires.

Cependant, l'utilisation des URIs et des Hyperliens permet de faciliter la découverte des services REST en utilisant les registres et les dépôts centralisés. De plus, un langage descriptive spécifique appelé WADL¹⁹ permet de décrire et de découvrir des services RESTful en analogue avec le langage descriptif WSDL²⁰ des services web.

5.3.Architecture ouverte de grille de service (OGSA)

OGSA est la vision de convergence décrite par la GGF²¹ des services Web et des services de grilles. Elle adopte l'architecture SOA afin de présenter les fonctionnalités des grilles sous forme d'un ensemble de services. Les services OGSA ou les services de grille sont une extension des services Web avec l'ajout de la notion d'état qui permet le control les services durant leurs cycles de vie. Cependant, afin d'implémenter cette notion de services à état, OGSA fait référence aux spécifications OGS²². La notion de service de grille a été introduite avec l'apparition de OGS, ils sont modélisés par des services Web avec une manière de description augmentée. OGS définit les mécanismes de création, de gestion et d'échange d'information entre les services de grille.

Pour qu'un Service Web se qualifie d'être un Service de grille, OGSA a défini trois conditions (Gannon et al.,2012) qui sont :

1. Il doit être une instance d'une implémentation de service ;
2. Il doit avoir un Grid Services Handle (GSH) qui est l'URI d'une instance d'un Service de grille. Il sert à la localisation du 'Grid Service Reference' (GSR) l'entité descriptive du service ;

¹⁹ WADL : web Application Description Language, est un langage basé sur XML permet de décrire des services proposés par une application REST.

²⁰ WSDL : Web Services Description Language, est un langage basé sur le standard XML permettant de décrire un service web.

²¹ GGF: Global Grid Forum, renommé à OGF acronyme de Open Grid Forum. Elle représente la communauté des utilisateurs, développeur et entreprises pour la standardisation des grilles de calcul

²² OGS : Open Grid Services Infrastructure, Publiée par la GGF pour fournir une infrastructure à OGSA capable d'ajouter la notion d'état aux services.

3. Il doit implémenter la méthode 'Grid Service' qui lui permet de collecter des informations de son environnement d'exécution, de définir la durée de vie d'une instance, ainsi que de pouvoir la détruire.

Cependant, les fonctionnalités que doit offrir un service de grille est de natures différentes comparées à celles offertes par un service Web. Cela a conduit au remplacement de OSGI par l'introduction des spécification Web Services Resource Framework (WSRF). Ce dernier est un ensemble de spécifications qui introduisent la notion des Ressources Services Web (Ressource-WS) utiles à la gestion des informations des états des ressources. Les spécifications Services Web sont :

- **Propriétés des Ressources-WS (WS-ResourceProperties) :** Elles décrivent le concept des Ressources-WS et comment des ressources à état sont associées à des Services Web. De même, elles décrivent comment les propriétés d'un service sont trouvées, modifiées et supprimées par l'utilisation des descriptions de ses données internes.
- **Durée de vie des Ressources-WS (WS-ResourceLifetime) :** C'est la période entre la création de l'instance de Ressource-WS et sa destruction. Les spécifications WS-ResourceLifetime tentent à standardiser la façon avec laquelle une instance d'une ressources-WS est détruite, ainsi que la façon dont sa durée de vie est gérée.
- **Notification-WS (WS-Notification) :** la notification est l'outil standard pour la communication inter-objets. Les WS se basent sur l'implémentation des échanges des messages standards. Cette spécification permet au WS d'être un producteur de notifications, et à certains d'autres d'être notifiés (consommateurs). De plus, elle fournit des standards pour notifier les consommateurs lors d'une modification d'une Ressource-WS.
- **Exceptions-WS (WS-BaseFaults) :** C'est un ensemble de spécifications permettant de standardiser la base des exceptions-WS qui peuvent être produites.
- **Groupe de Service Web (WS-ServiceGroup) :** Afin de construire un service de haut niveau qui sert à la gestion des autres services de l'environnement, on a besoins d'un ensemble de primitives. Ces derniers offrent des opérations basiques sur une

collection de services telles que l'ajout, la suppression ou la recherche d'un service dans un groupe.

- **Renouveau des références-WS (WS-RenewableReferences) :** C'est un ensemble de spécifications permettant de standardiser les mécanismes de renouvellement des références des instances quand elles ne seront plus valides.

5.4. Les services de grille

Un service de grille est un service défini à base de WSDL. Il est conforme à un ensemble de conventions liées à la définition de son interface et de ses comportements (Frey et al., 2002). Les Services WSRF sont le noyau des Services de grille, ils sont utilisés pour la création, la destruction et la gestion des Ressources-WS à état. La figure 2 montre la relation entre WSRF, OGSA et les Services Web pour construire un Service Web à état.

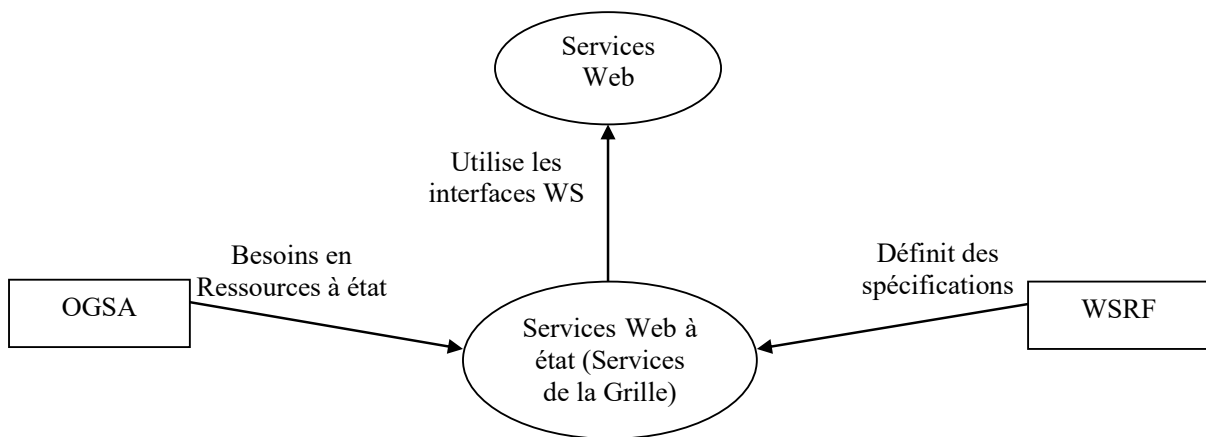


Figure 2. Relation entre WSRF, OGSA et les Services Web (Borja et Lisa, 2006).

6. Conclusion

Les grilles de calcul ont été créés pour surmonter les limites des anciens systèmes distribués et fournir un niveau d'abstraction pour la bonne exploitation des ressources hétérogènes et géographiquement distribuées. Cela a permis de mettre au premier plan des notions existantes, mais qui n'ont pas pris leur bonne place au sommet des technologies telles que la notion de l'architecture orientée services SOA et les spécifications WSRF. Cependant, d'autres nouvelles technologies ont apparus telles que les cloud et les smart grid pour fournir une solution plutôt souple que celle offerte par les grilles de calcul. Une partie des technologies ont été reprises et améliorées, et d'autres notions sont restées restreintes aux grilles. Certaines solutions telles que les cloud ont veillé à réutiliser les grilles en intégrant une couche de virtualisation dans leur architecture. Cette couche permet de virtualiser n'importe quelle ressource même toute une grille afin de fournir des ressources ajustées à la demande des applications et des clients.

De plus, La grille reste une structure réutilisable par les nouvelles technologies, mais la priorité est donnée à l'externalisation des ressources. D'un autre côté, les grilles restent compatibles avec les nouvelles solutions grâce à son aspect modulaire qui permet la mise à jour et l'amélioration partielle selon les cas.

7. Références

- Almond, J., & Snelling, D. (1999). UNICORE: uniform access to supercomputing as an element of electronic commerce. *Future Generation Computer Systems*, 15(5-6), 539-548.
- Anderson, D. P. (2010). Volunteer computing: the ultimate cloud. *ACM Crossroads*, 16(3), 7-10.
- Anderson, T. E., Culler, D. E., & Patterson, D. (1995). A case for NOW (networks of workstations). *IEEE micro*, 15(1), 54-64.
- Becker, D. J., Sterling, T., Savarese, D., Dorband, J. E., Ranawak, U. A., & Packer, C. V. (1995, August). BEOWULF: A parallel workstation for scientific computation. In *Proceedings, International Conference on Parallel Processing* (Vol. 95, pp. 11-14).
- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16). ACM.
- Borja Sotomayor & Lisa Childers. "Globus® Toolkit 4: Programming Java Services". Morgan Kaufmann publisher. 2006.
- Cérin, C., & Fedak, G. (Eds.). (2012). *Desktop grid computing*. CRC Press.
- Chetty, M., & Buyya, R. (2002). Weaving computational grids: how analogous are they with electrical grids?. *Computing in Science & Engineering*, 4(4), 61-71.
- Culler, D. E., Arpaci-Dusseau, A., Arpaci-Dusseau, R., Chun, B., Lumetta, S., Mainwaring, A., ... & Wong, F. (1997, May). Parallel computing on the Berkeley NOW. In *9th Joint Symposium on Parallel Processing*.
- D. Booth, H. Haas, F. McCabe, et al., Working Group Note 11: Web Services Architecture, www.w3.org/TR/2004/NOTE-ws-arch-20040211/ (accessed 03/05/2018)
- Foster, I. (2001, May). The globus toolkit for grid computing. In *ccgrid* (p. 2). IEEE.
- Foster, I. (2002). What is the grid?-a three point checklist. *GRIDtoday*, 1(6).
- Foster, I. (2005, November). Globus toolkit version 4: Software for service-oriented systems. In *IFIP international conference on network and parallel computing* (pp. 2-13). Springer, Berlin, Heidelberg.
- Foster, I., & Kesselman, C. (Eds.). (2003). *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier.
- Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 15(3), 200-222.
- Frey, J., Graham, S., & Kesselman, C. (2002, October). Grid service specification. In *Open Grid Service Infrastructure WG. Technical Report, Global Grid Forum, Draft* (Vol. 2).
- Gannon, D., Chiu, K., Govindaraju, M., & Slominski, A. (2012). A revised analysis of the open grid services infrastructure. *Computing and informatics*, 21(4), 321-332.
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- Natrajan, A., Nguyen-Tuong, A., Humphrey, M. A., Herrick, M., Clarke, B. P., & Grimshaw, A. S. (2002). The legion grid portal. *Concurrency and Computation: Practice and Experience*, 14(13-15), 1365-1394.
- Robertazzi, T. G. (2017). Grids, Clouds, and Data Centers. In *Introduction to Computer Networking* (pp. 113-127). Springer, Cham.

- Schuller, B., Menday, R., & Streit, A. (2006, August). A versatile execution management system for next-generation unicore grids. In *European Conference on Parallel Processing* (pp. 195-204). Springer, Berlin, Heidelberg.
- Schuller, B., Rybicki, J., & Benedyczak, K. (2014). High-performance computing on the web: Extending uncore with restful interfaces. In *The Sixth International Conference on Advances in Future Internet* (No. FZJ-2014-06228). Jülich Supercomputing Center.
- van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, 98(10), 967-1009.

Chapitre 2 : Déploiement et interrogation des services dans une grille de calcul

Chapitre 2 : déploiement et interrogation des services dans une grille de calcul

1. Introduction

Le déploiement des services dans une grille est une étape indispensable pour la mise en disponibilité de ces services. Le noyau d'un déploiement est le conteneur des services. Dans le présent chapitre, on va présenter la notion des conteneurs de services, des conteneurs de ressources, ainsi que leur orchestration. Dans un deuxième lieu, on présente les solutions de déploiement et d'interrogation de services existantes, et comment les jobs sont pris en charge par les ressources des grilles. Par la suite, on soulève les insuffisances marquées sur les solutions classiques. Et à la fin, on propose une meilleure solution qui prend en considération les critiques signalées des travaux existants et l'amélioration de déploiement des services dans les grilles et par conséquent leur invocation.

2. Serveurs d'applications

Un serveur d'application est une composante logicielle utilisée pour héberger des applications Web y compris des services web et des EJB (Entreprise Java Beans). Il définit un environnement d'exécution connu par le nom 'Conteneur'. Ce dernier est un niveau d'abstraction qui s'interface entre un composant et la fonctionnalité spécifique à la plateforme de niveau inférieur et qui prend en charge ce composant (J2EE Containers, 2018). Ce dernier peut être un service web, Servlet, JSP, JSF, EJB, Applet, etc. Ils existent plusieurs serveurs d'applications très connus, on cite Apache TomEE (Apache TomEE, 2018), Websphere (Application Server | IBM Cloud, 2018), JBoss (JBoss Developer, 2018), et Weblogic (Oracle WebLogic Server Technical Information, 2018).

Un serveur d'application fournit des services indispensables pour l'exécution des composants logiciels. Il offre un environnement d'exécution du code utilisateur sur une couche modulaire configurable qui assure des services de base tels que :

- Le regroupement de connexion (connection pooling) : ce service permet de réutiliser les connexions déjà ouvertes aux bases de données. Cela optimise les performances des applications en évitant l'ouverture et la fermeture d'une connexion à chaque requête d'accès à la base de données. Une nouvelle ouverture de connexion est effectuée seulement si aucune n'existe déjà.

- Gestion des transactions : Une transaction est une unité d'activité, au sein de laquelle plusieurs mises à jour de ressources peuvent être rendues atomiques. Elle peut être locale ou globale. Dans le premier cas, la transaction est prise en charge par le gestionnaire de ressource qui peut être à titre d'exemple une unité de traitement ou une base de données. Dans le second cas, la transaction peut atteindre plusieurs gestionnaires de ressources. Cela exige que la gestion de la transaction globale soit gérée par un gestionnaire externe. Dans ce contexte, un serveur web est un gestionnaire de ressources qui peut coordonner entre les transactions globales, comme il peut fournir un environnement dans lequel les transactions locales du gestionnaire de ressources peuvent s'exécuter (IBM Knowledge Center, 2018).
- Messagerie : le serveur d'applications assure la gestion des messages pour parvenir à remonter les messages issus des différents niveaux architecturaux de l'application. De plus, elle peut être utile pour avoir une trace d'exécution des applications qui sert à l'analyse des opérations effectuées par l'application, les erreurs, et l'historique de manière générale de tout ce qui est passé au niveau des applications lancées sur le serveur web.
- Distribution de charge : la distribution de charge permet d'affecter les requêtes des clients aux différentes unités de traitement afin d'optimiser l'utilisation des ressources, la tolérance aux pannes, et optimiser le temps de réponse du système. Cela peut être réalisé en définissant un ensemble de conteneurs sur un ensemble de ressources physiques. Les conteneurs délivrés par les serveurs web sont une virtualisation de ressources software et hardware. Un conteneur peut être lancé en spécifiant une taille maximale d'un espace mémoire, une fréquence de processeur, un nombre de cœur CPU ou une bande passante précise. Cela est valable sur le même hôte comme sur des hôtes distants. Le serveur d'applications collecte des informations des états des différents conteneurs lancés. Pour une nouvelle charge, il redirige la requête vers le conteneur le plus approprié de point de vue ressources physiques et logiques. Si les ressources d'un conteneur sont épuisées, le serveur d'applications consulte un autre conteneur moins chargé et répond aux exigences de l'application à lancer.

- Services de Clustering : Le clustering de serveurs est une méthodologie de regroupement de serveurs indépendants pour offrir une meilleure flexibilité, évolutivité et disponibilité. Un serveur d'application est doté d'un service qui supporte le regroupement de plusieurs ressources physiques afin de résoudre les problèmes de manque d'espace de stockage, de bande passante et de surcharge des unités de traitement.
- Sécurité : un serveur d'application doit supporter les protocoles standards de sécurité tels que SSL, SSH, TLS, HTTPS, SSO, etc. afin de sécuriser les données et les applications hébergées sur ce serveur. Cela rend possible des opérations de sécurité de base telles que l'authentification et la délégation des droits.

3. Les conteneurs

Un conteneur est un niveau d'abstraction logiciel qui sert à découpler les applications de leur contexte d'exécution. Il s'interface avec le système d'exploitation afin de fournir les ressources et les dépendances nécessaires à l'exécution des applications. Les ressources peuvent être logiques ou physiques. Les ressources logiques sont des composantes intégrables au conteneurs utilisées à la demande sous forme de modules tels que l'espace des noms fourni par le SE, la prise en charge des logs, d'un protocole particulier de sécurité, d'un serveur mail spécifique, etc. De l'autre côté, les ressources physiques sont présentées par le processeur, la mémoire vive, les supports de stockage, etc. Le recours aux conteneurs, en générale, permet d'assurer les caractéristiques suivantes pour les applications conteneurisées (Hightower et al., 2017) :

- Vitesse : la technologie des conteneurs permet une rapidité de déploiement des applications et des services de sorte qu'ils restent toujours disponibles malgré leurs opérations de maintenances et de mise à jour.
- Passage à l'échelle : les applications et les services conteneurisés sont facilement scalables le fait que les conteneurs sont répliquables et donc ils peuvent offrir leurs services à plus grande échelle. D'une autre part, le passage à l'échelle de l'équipe de développement et de test des services est aussi possible grâce à l'architecture modulaire de ces services.

Ci-dessous, on cite les conteneurs les plus connus.

3.1. Les conteneurs classiques Java/JEE

La technologie JEE définit plusieurs types de conteneurs qui dépendent de la nature de la composante. La figure 3 montre les différents conteneurs définis par JEE :

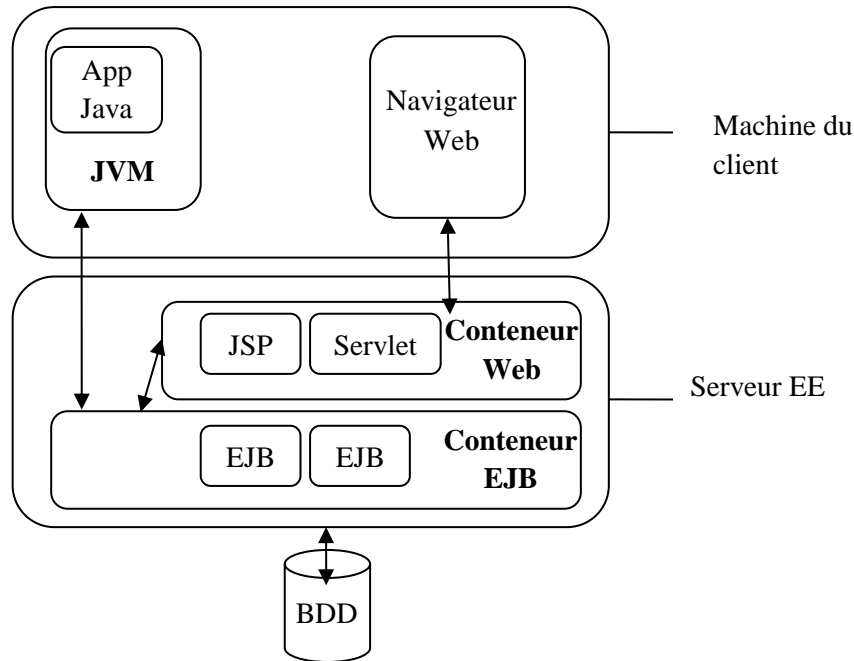


Figure 3. Architecture de conteneur JavaEE ("Java EE Containers - The Java EE 5 Tutorial", 2018).

- La machine virtuelle Java (JVM) : elle assure une abstraction entre les programmes écrits en Java et l'environnement d'exécution y compris le système d'exploitation. La JVM accède aux ressources physiques et logiques de la machine hôte via l'OS et les présente aux applications chargées et lancées sur le conteneur.
- Le conteneur Web : ce type de conteneur est fourni par des serveurs Web tel qu'Apache Tomcat ou des serveurs d'applications tel que JBoss. Il gère l'exécution des Servlets et des JSP dans une application JEE ainsi que leurs accès aux bases de données.
- Le conteneur JavaBeans Entreprises (EJB) : il permet l'exécution des EJBs en assurant leurs connexions aux SGBDs et aux composantes lancées sur le conteneur web.
- Conteneur d'Applet : l'applet est un programme Java qui s'exécute sur un navigateur web. Le conteneur des Applets est le contexte qui permet l'exécution de

ces derniers. Il doit permettre leur téléchargement sur la machine du client, vérification des droits d'exécution, et le lancement proprement dit des Applets.

3.2.L'emprisonnement

Appelé aussi 'Jail', il vise à fournir un environnement d'exécution isolé (Kamp et al., 2000). Cette technique permet à l'OS de tromper les applications en cours d'exécution et leur croire qu'elles fonctionnent sur la racine de système de fichiers '/', et en réalité elles fonctionnent sur des dossiers chrootés²³. Cela donne la possibilité d'isoler ou d'emprisonner n'importe quel processus et même un OS complet dans un coin du système de fichiers. L'avantage de cette technique est qu'elle est très légère et ne consomme aucune ressource supplémentaire pour la mettre en place. De l'autre côté, les ressources utilisées par ces processus ne peuvent pas être gérées de manière isolée et sont considérées comme n'importe quel autre processus de la machine hôte.

3.3.Les Conteneurs Linux (LXC)

Les conteneurs LXC (Rosen, 2014a) héritent de la technique Jail du fait qu'elle se repose sur les dossiers chrootés pour créer un contexte d'exécution isolé du reste de la machine hôte. Cependant, ils surmontent les limites de Jail en utilisant principalement la technologie Cgroups, et le cloisonnement des espaces de nommage qui sont détaillés ci-dessous :

- Cloisonnement des espaces de nommage (Kernel namespaces), L'isolation de l'espace de noms permet de séparer des groupes de processus. Cela garantit qu'ils ne peuvent pas voir les ressources dans d'autres groupes. Différents espaces de noms sont utilisés pour l'isolation des processus, les interfaces réseau, l'accès à la communication interprocessus, les points de montage ou pour isoler les identifiants de noyau et de version (Rosen, 2013).
- Cgroups, où les groupes de contrôle gèrent et limitent l'accès aux ressources pour les groupes de processus. Ils permettent de partager les ressources matérielles disponibles entre les conteneurs et, si nécessaire, de définir des limites et des contraintes d'accès et d'exploitation de ces ressources. Parmi les possibles

²³ Un dossier Chrooté est un emplacement dans le système de fichier sur lequel une commande chroot a été exécutée.

fonctionnalités que peut offrir Cgroups, on cite (Rosen, 2014b) : (1) l'affectation d'un ensemble de CPUs à un ensemble de processus ; (2) Créez un processus avec une quantité de mémoire limitée ; (3) établir des rapports d'utilisation de CPU par différents processus ; (4) Mettre en pause des processus ; (5) assigne une priorité à un trafic réseau ; (6) marquer un trafic réseau donné.

De plus, les conteneurs Linux se basent sur les modules 'Application Armor' (AppArmor) (Bauer, 2006) et Security-Enhanced Linux (SELinux) (McCarty, 2005). Ils fournissent un support sécurisé pour l'isolation des processus. Ils sont une implémentation de module de sécurité Linux (LSM²⁴). La fonction la plus attractive est la possibilité d'isoler un attaquant pour qu'il ne puisse pas accéder au système d'exploitation. Ils permettent d'étiqueter les processus avec un libellé lié à un contexte de sécurité. De plus, ils permettent de surveiller ces processus en vérifiant qu'ils respectent certains modes d'accès qui incluent le rôle²⁵, l'utilisateur virtuel²⁶, et le contexte de sécurité²⁷.

La figure 4 présente l'ensemble des composantes liées à l'exécution des applications en utilisant le conteneur Linux.

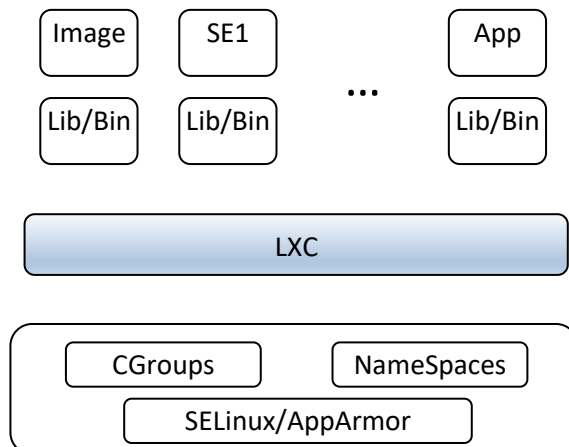


Figure 4. Architecture du conteneur LXC.

²⁴ Linux Security Module (LSM) : un Framework qui permet au noyau Linux de supporter une variété de modèles de sécurité informatique tout en évitant le favoritisme envers une implémentation de sécurité unique.

²⁵ Un rôle dans SELinux est une attribution d'un droit à un processus ou un utilisateur tels que l'accès en lecture ou en écriture.

²⁶ Un utilisateur virtuel est une attribution d'un ensemble de rôle à un acteur défini par SELinux.

²⁷ Contexte de sécurité SELinux : défini par un nom, un rôle, et un type qui représente la nature de l'objet surveillé par SELinux tels qu'un fichier, un port réseau, etc.

Les objets déployables sur un conteneur LXC sont des applications simples, des systèmes d'exploitation complet, ou des images d'applications/systèmes. Chacun de ces objets est lié à ses bibliothèques et à ses dépendances indispensables pour leur exécution. Ces Lib et Bin sont des fichiers précompilés, des exécutables, et des fichiers de configuration.

3.4.Docker

Docker (Merkel, 2014) est une technologie de conteneurisation basée principalement sur CGroups, le noyau et les conteneurs Linux. A partir de sa version 0.8, libContainer ("Opencontainers/runc", 2015) est ajouté à côté de LXC, il est écrit en langage Go²⁸ afin de permettre l'accès aux APIs du noyau Linux tels que CGroups et namespaces sans aucune autre dépendance externe. En autre mot, il est une implémentation native de l'environnement d'exécution de conteneur Linux (Marmol et al., 2015). De plus de LXC et libcontainer, Docker peut être utilisé avec d'autres outils d'isolation et de virtualisation tels que OpenVZ²⁹ ("Open source container-based virtualization for Linux", 2018), libvirt³⁰ ("libvirt: The virtualization API", 2018), etc.

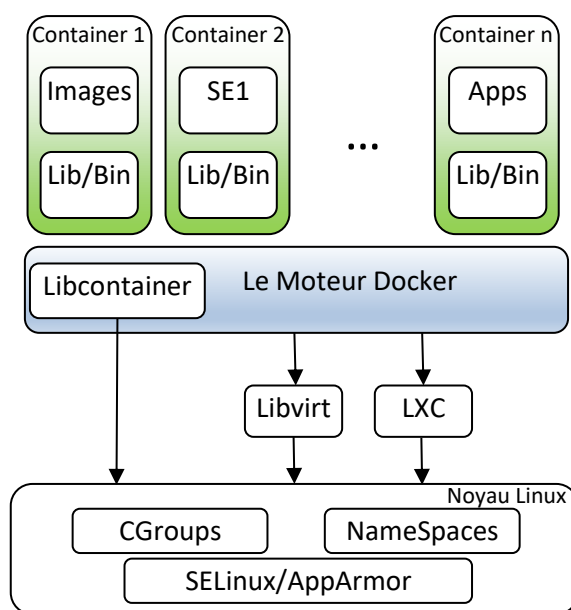


Figure 5. L'API Docker.

²⁸ Langage Go : ou Golang, est un langage utilisé initialement par Google. Il offre des fonctionnalités avancées telles que la collecte de garbage Mémoire, politique de sécurité, etc.

²⁹ OpenVZ : fournit des conteneurs Linux isolés et sécurisés sur le même serveur physique en permettant une meilleure utilisation du serveur et garantissant que les applications n'entrent pas en conflit.

³⁰ LibVirt : est un pilote qui gère les plateformes de virtualisation, il supporte des plateformes telles que LXC, OpenVZ, VMware, VirtualBox, etc.

Docker peut isoler des applications ainsi que leurs dépendances dans des conteneurs qui peuvent être exécutés sur n'importe quel serveur en facilitant leur gestion, leur migration, et leur redémarrage à froid ou à chaud. Il peut gérer plusieurs conteneurs LXC ou libcontainer à la fois et leurs permettre de partager les ressources matérielles et logicielles de l'hôte. La figure 5 présente les principales composantes de Docker.

Les avantages de Docker par rapport à lxc sont :

- La modularité : docker est basé sur la notion de micro-service, cette dernière permet une gestion partielle des applications sans affecter l'ensemble des fonctionnalités ni leurs déploiements. Ainsi, elle favorise la réutilisation partielle, la maintenance, l'évolutivité et le passage à l'échelle.
- La notion des couches : les images manipulées par docker sont sous forme de plusieurs couches. A la base, on trouve un OS tels que Ubuntu ou Fedora dans la première couche. Une modification dans l'image entraîne une création d'une autre couche qui sera traduite par des instructions enregistré dans un fichier docker. L'utilisation des couches permet de bien gérer les versions des images, leur réutilisation par plusieurs conteneurs, le gain en espace disque et la restauration des anciennes versions des images.

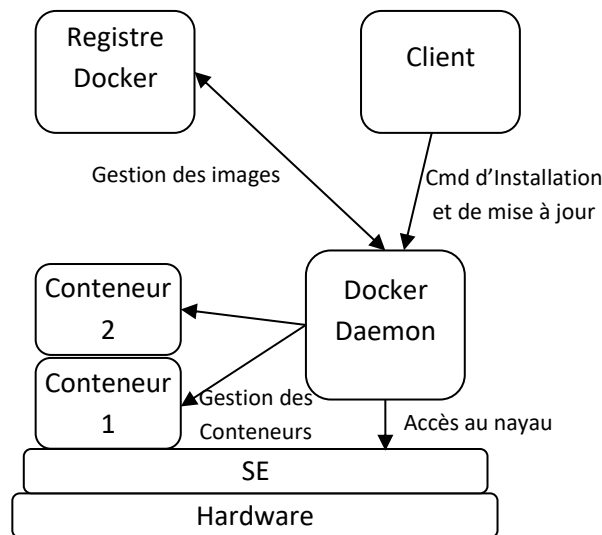


Figure 6.L'architecture de Docker.

Docker est d'une architecture client-serveur. Le client Docker manipulé par l'utilisateur s'interface avec le daemon Docker. Ce dernier est chargé d'exécuter les

commandes du client telles que `apt-get install/update`, `exec`, `run`, etc. de plus, le daemon Docker crée des images à chaque fois sollicité par le client et il peut les gérer dans le registre d'images en utilisant les fonctions `push`, `pop`, et `search`. Le registre Docker est utilisé pour le stockage des images cumulées utilisables par le Moteur Docker. D'un autre côté, le daemon Docker se charge du démarrage, de l'arrêt et de l'inspection des conteneurs. La figure 6 montre l'architecture de Docker.

D'autres conteneurs existants tels que Warden³¹ sont utilisés dans le domaine de cloud et sont moins réponsifs comparé à Docker.

4. Orchestrateurs de conteneurs

Les outils d'orchestration gèrent la manière dont plusieurs conteneurs sont créés, mis à jour, et rendus hautement disponibles. L'orchestration contrôle également la manière dont les conteneurs sont connectés pour créer des applications sophistiquées à partir de plusieurs micro-services déployés dans de différents conteneurs. Les environnements d'exécution des conteneurs peuvent faire des opérations de `push` ou de `pull` d'une image pour gérer un conteneur. Par contre si on fait face à plusieurs conteneurs, ces environnements d'exécution vont échouer. L'objectif d'utiliser L'orchestrateur des conteneurs est de laisser les environnements d'exécution de ces conteneurs focaliser sur la gestion de ses états y compris la création, le démarrage, l'arrêt et le chargement des images. Ainsi, laisser les orchestrateurs gérer l'infrastructure sous-jacente en permettant la portabilité, la sécurité et la synchronisation entre conteneurs.

4.1. Kubernetes

Kubernetes ("Production-Grade Container Orchestration – Kubernetes", 2018) est un orchestrateur open source dédié au déploiement d'applications conteneurisées. Il a été à l'origine développé par Google, inspiré par une décennie d'expérience dans le déploiement de systèmes évolutifs et fiables dans des conteneurs via des APIs orientées applications (Burns et al., 2016). Kubernetes fournit le software nécessaire pour un déploiement fiable et scalable des applications distribuées (Hightower et al., 2017). Il peut être utilisé comme 'hosted Kubernetes-As-A-Service' avec les grands fournisseurs des cloud tels que Google

³¹ Warden : ou Garden pour Warden écrit en Go, offre la gestion des conteneurs du PaaS Cloudfoundry. Le projet est accessible sur le site : <https://github.com/cloudfoundry-attic/warden>.

Cloud Platform³², Microsoft Azure³³, Amazon Web Service³⁴ (AWS), et OpenShift³⁵ de RedHat.

Kubernetes fournit des abstractions de haut niveau pour gérer des groupes de conteneurs. Il est désigné pour automatiser et faciliter les opérations suivantes :

- Le déploiement des applications qui tournent sur plusieurs conteneurs : grâce à l'architecture modulaires des applications conteneurisées qui est basée sur les micro-services, ce type de déploiement est possible. Kubernetes se charge de déployer ces applications, les maintenir en fonction et maintenir leur synchronisation les unes aux autres sans aucun effort des utilisateurs.
- Passage à l'échelle des applications conteneurisées : Kubernetes permet la gestion des ressources physiques lors de passage à l'échelle des applications. Le redimensionnement des applications se fait de manière automatique en distribuant la charge de travail et provisionner ou libérer les ressources physiques de manière rapide.
- Déploiement de nouvelles versions des applications conteneurisées sans aucun temps d'arrêt : Kubernetes permet la mise à jour des versions des applications en upgrade ou downgrade de manière fiable et rapide.
- Assure des fonctionnalités de base indispensables pour la gestion des conteneurs telles que le networking pour permettre aux conteneurs d'échanger les données, la découverte de services, et un stockage persistant.
- Fonctionne indépendamment de l'environnement d'exécution : Kubernetes fonctionne sur tous systèmes supportant la technologie de conteneurisation que ce soit Windows, Linux, des cloud privés ou publiques, des machines virtuelles ou physiques, et sur des clusters qui mixte tous les environnements précédemment cités.

³² Google Cloud Platform : Accessible sur : <https://cloud.google.com>.

³³ Microsoft Azure : Accessible sur : <https://azure.microsoft.com/>

³⁴ Amazon Web Service : Accessible sur : <https://aws.amazon.com/>

³⁵ OpenShift : un environnement de la communauté RedHat qui offre une plateforme complète de contrôle de conteneurs. Accessible sur : <https://www.openshift.com/>

4.1.1. Architecture

L'architecture de Kubernetes est une architecture client-serveur. Le nœud maître contrôle les nœuds esclaves. Ces derniers hébergent et exécutent les conteneurs en utilisant des PODs.

Une POD peut grouper un ou plusieurs conteneurs déployés sur le même nœud en leur permettant de partager les mêmes ressources y compris le même nom d'hôte, adresse IP, mécanisme IPC³⁶. L'utilisation des PODs permet aussi de séparer la partie conteneurs de l'infrastructure sous-jacente ce qui simplifie leur gestion. Une réplication des PODs assure une scalabilité horizontale pour une application ou un service donné. La figure 7 montre un schéma illustratif d'une POD.

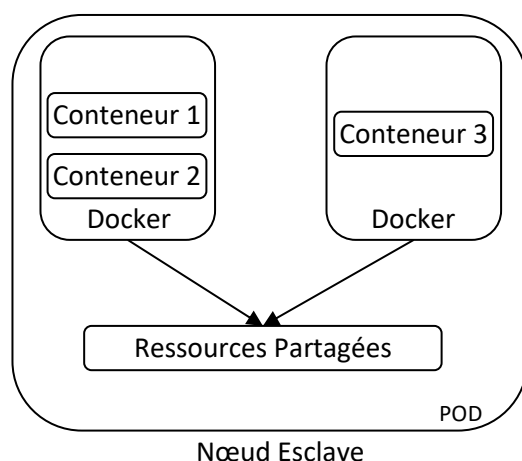


Figure 7. Schéma illustratif de la structure d'une POD.

Un autre composant de l'architecture de Kubernetes est le Kubelet. Ce dernier est exécuté sur les nœuds esclaves, sa principale fonctionnalité est de surveiller le bon lancement des conteneurs sur les nœuds de déploiement et la collecte des informations concernant la charge des ressources de ces nœuds. Ainsi, le nœud maître peut assurer l'équilibrage de charge et la bonne exploitation des ressources. Une autre fonctionnalité de maître concernant les PODs est d'assurer leurs résiliences. Si une POD est à défaut de fonctionnement, le Kubelet transmet l'information au maître, ce dernier assure de manière automatique l'exécution du POD sur un autre nœud qui possède les performances requises.

³⁶ Les mécanismes d'IPC permettent de faire communiquer et/ou de synchroniser n'importe quel couple de processus locaux.

Un dernier composant de l'architecture de Kubernetes est la notion de service. Ce dernier permet de jouer le rôle d'un point d'entrée pour des PODs identiques ou répliqués. Il redirige la requête d'un client au bon POD en vue de sa charge de travail. Un service détient toutes les références des PODs sur l'ensemble des nœuds disponibles. Les services peuvent correspondre à des front end applicatif tels qu'un service de gestion de base de données, un serveur Web, un serveur de fichier, ...

La figure 8 montre l'architecture de l'orchestrateur Kubernetes.

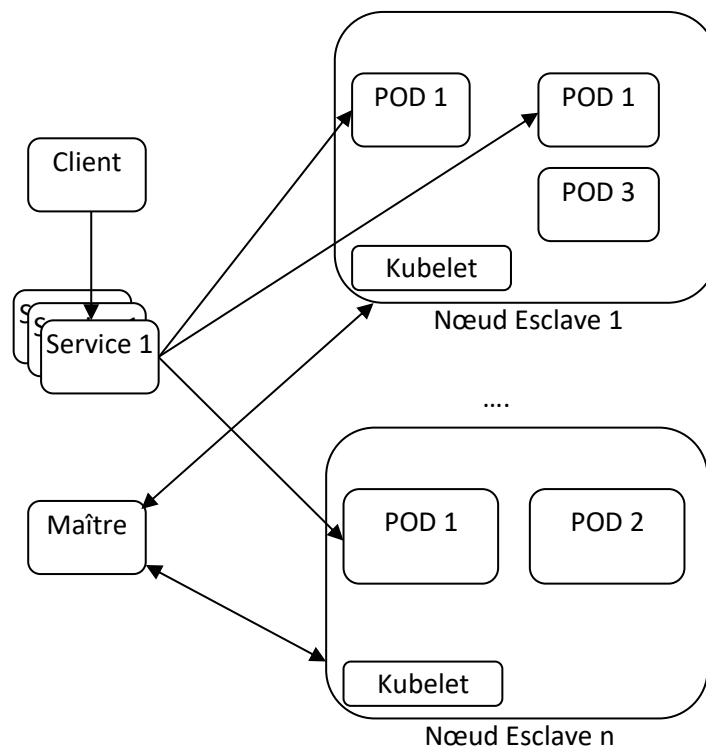


Figure 8. Vue illustrative de l'architecture de Kubernetes.

4.2. Apache YARN

YARN (Yet Another Resource Negotiator) (Vavilapalli et al., 2013) est principalement un négociateur de ressource et utilisé aussi comme un orchestrateur de conteneurs. Il est à la base de fonctionnement de plusieurs Framework de big data tels qu'Apache Hadoop et Apache Spark. Son principe de fonctionnement est de partager la gestion des ressources et des Jobs sur plusieurs nœuds.

4.2.1. Architecture

YARN est d'une architecture centralisée dans laquelle un seul composant logique, le gestionnaire de ressources (RM), alloue des ressources aux Jobs soumis au cluster. Un

Job soumis peut être atomique ou sous forme de DAG est pris en charge par l'orchestrateur d'applications (Application Master - AM). Ce dernier fait appel aux gestionnaires des nœuds (Node Manager - NM) qui assurent la disponibilité des ressources pour l'exécution des applications. Les composantes de YARN sont en nombre de trois :

4.2.1.1. Le gestionnaire de nœud (NM)

Le NM est un processus démon qui s'exécute sur chacun des nœuds du cluster sur lequel est déployé YARN. Sa fonction est de surveiller la disponibilité des ressources sur les nœuds, de signaler les pannes et de gérer les conteneurs lancés sur le cluster (Karanasos et al., 2018).

4.2.1.2. Le gestionnaire de ressources (Resource Manager-RM)

Le gestionnaire de ressources harmonise la consommation des ressources entre plusieurs applications concurrentes. Plusieurs RMs peuvent être utilisés pour assurer la scalabilité des applications et la haute disponibilité des ressources, l'un entre les RMs lancés étant le maître. Les NMs informent périodiquement les RMs de leurs statuts faisant partie des éléments de l'état du cluster. Le RM gère également les demandes de ressources de toutes les applications y compris la disponibilité des ressources, les priorités de planifications et des stratégies de partage de ressource (Karanasos et al., 2018). Le planificateur effectue la mise en correspondance des demandes d'applications et des conteneurs disponibles.

4.2.1.3. Le moniteur d'application (AM)

Il est chargé de négocier les ressources à partir du RM et de travailler avec le(s) NM(s) pour exécuter et surveiller les conteneurs et leur consommation de ressources. Il est responsable de négocier les conteneurs de ressources appropriés présentés par le RM et de suivre leurs états. Un AM doit généralement exploiter les ressources de plusieurs nœuds pour exécuter un Job. Pour obtenir des conteneurs, l'AM émet des demandes de ressources au RM. Lorsque le planificateur affecte une ressource ou un conteneur à un AM, ce dernier la présente aux NMs pour l'exploiter.

4.2.2. Soumission d'application et déploiement dans YARN

La figure 9 illustre les interactions entre les modules de YARN lors de soumission d'une application. Ces interactions sont décrites par ordre chronologique comme suit :

- Le client soumet une application ainsi que ses spécifications qui sont utiles pour son exécution. L'application est supposée gourmande et soumise via un client YARN (front-end de YARN).
- Le RM vérifie si les ressources requises par l'application sont disponibles. Le RM localise les conteneurs fournissant les ressources nécessaires gérées par les NMs, et lance le moniteur d'application.
- Sur les nœuds qui hébergent les conteneurs cibles, les NMs reçoivent les spécifications nécessaires pour le lancement des conteneurs requis pour l'exécution de l'application.
- Le RM reçoit les actualités sur l'état des conteneurs à partir des NMs, et les nouvelles demandes de ressources à partir de l'AM. Cependant, l'utilisateur reçoit des mises à jour de progression de l'application en cours d'exécution à partir des AMs.
- Une fois l'application terminée, le AM se désenregistre auprès du RM et s'arrête.

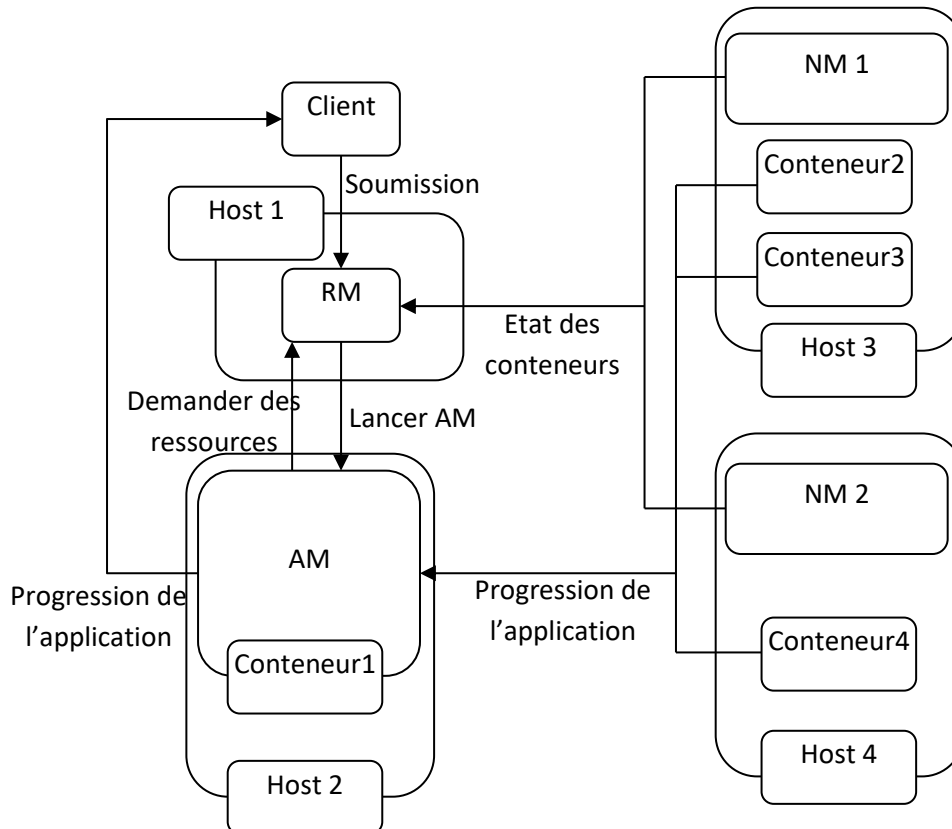


Figure 9. Interaction entre les composants de YARN pour l'accomplissement d'une application.

D'autres orchestrateurs de conteneurs sont aussi reconnus autant que Kubernetes tels que Docker Swarm³⁷ et Apache Mesos ("Apache Mesos", 2018).

5. Déploiement, interrogation, et exécution des services dans une grille

Le déploiement des services de grille dépend de l'intergiciel qui assure le fonctionnement de cette grille. Comme la technologie des grilles de calcul est désapprouvée en faveur des technologies plus récentes telles que les clouds, on trouve très peu d'intergiciels qui supporte encore les grilles tels que l'intergiciel Unicore ("UNICORE | Distributed computing and data resources", 2018) et Globus. Cependant, un bon déploiement doit être dynamique afin d'assurer que les services déployés sont bien en dernière version, efficacement gérable, hautement disponible et présente une très bonne qualité de service. Cependant, au point de vue des utilisateurs, un déploiement dynamique désigne la possibilité qu'un client distant charge et déploie un nouveau service, ou annuler le déploiement d'un service déjà existant dans un conteneur existant sans affecter le bon fonctionnement du reste des services (Qi et al., 2007).

5.1.Travaux en relation

5.1.1. Globus

Le conteneur Globus est l'un des environnements d'hébergement Java pour les services Web compatibles WSRF. Le conteneur Globus est exécuté sur chaque hôte et permet aux services d'utiliser les ressources sous-jacentes de l'hôte, telles que le processeur, la mémoire et le stockage. Une étape typique pour exécuter une application sur un hôte spécifique consiste à implémenter l'application en tant que service compatible WSRF et à le déployer dans le conteneur de l'hôte. Les clients accèdent ensuite au conteneur pour exécuter les fonctions du service avec SOAP et http (Byun et al., 2009).

Les services de grilles sont à la base des services web avec l'adoption des spécifications WSRF. Un service de grille est composé d'un ensemble de fichiers, chacun d'eux joue un rôle bien défini. Les rôles existants sont : la découverte, la description du

³⁷ Docker Swarm : <https://docs.docker.com/get-started/part4/>

contenu du service, la description du déploiement du service et la partie métier qui est l'implémentation des fonctionnalités que le service doit assurer.

Les services de grilles dans Globus sont déployés comme de simple services web. Cependant, l'exécution des instructions de la partie métier est pris en charge par le gestionnaire des ressources connus par GRAM5³⁸ (Grid Resource Allocation and Management). Un conteneur de service Web/grille fait référence à trois éléments principaux qui sont le moteur SOAP, le serveur d'application, et le serveur Web. Un serveur d'application peut contenir le serveur web et le moteur SOAP.

Le serveur web permet aux clients d'interroger le service, ou de répondre via des messages conformes au protocole http. Ces messages sont le seul moyen de communication avec le service. Ils sont emballés dans des enveloppes SOAP par le moteur SOAP. Le client ne peut localiser les services qu'en envoyant une requête au serveur web. Cela invoque un autre service de localisation qui a la fonction de fouiller parmi les fichiers descripteurs le service qui répond le mieux à la demande de client. Les services sont inscrits au près d'un registre généralement décentralisé en analogue au registre UDDI. La description des services de grille se fait avec un langage basé XML qui hérite de WSDL, et qui s'appelle GSDL pour 'Grid Service Description Language'.

Une fois le service est découvert, les requêtes du client seront mises dans des enveloppes SOAP et transmises via le serveur Web jusqu'à le point final traduit par l'URI du service. Le serveur d'application peut par la suite exécuter la partie métier du service interrogé et retourne la réponse via le même chemin. La partie métier du service sert à lancer un Job gourmand en termes de ressources physiques. Un Job dans Globus est un programme qui nécessite une description, une autorisation, et une gestion de son exécution.

La description des Jobs est réalisée en utilisant le langage descriptif 'Resource Specification Language (RSL)'. Il sert à décrire les ressources et tous ce qui est nécessaire pour la bonne exécution du Job y compris l'espace mémoire et le nombre et la puissance des CPUs. Les divers composants de l'architecture de gestionnaire de ressources incluent un co-allocateur et manipulent le langage RSL. Ce dernier fournit la syntaxe nécessaire pour

³⁸ GRAM5 : est un ensemble de service du Globus Toolkit qui permet la gestion des Jobs sur des ressources de grille. Disponible à l'adresse : <http://toolkit.globus.org/toolkit/docs/5.0/5.0.0/execution/gram5/>.

composer des descriptions de ressources complexes en se basant sur des paires <attributs, valeur>.

Le gestionnaire des ressources GRAM5 vérifie l'existence du service demandé et la possibilité de son utilisation par le client. L'authentification des utilisateurs de Globus est assurée par le module GSI. Il implémente une politique de délégation des droits afin que la session ouverte par les utilisateurs soit valide au niveau de tous les autres modules de l'intergiciel. Selon le profil de l'utilisateur connecté, GRAM5 accord ou non l'exécution du service demandé. Ainsi, il peut définir les ressources physiques qui peut les inclure dans l'exécution du Job issue du service lancé.

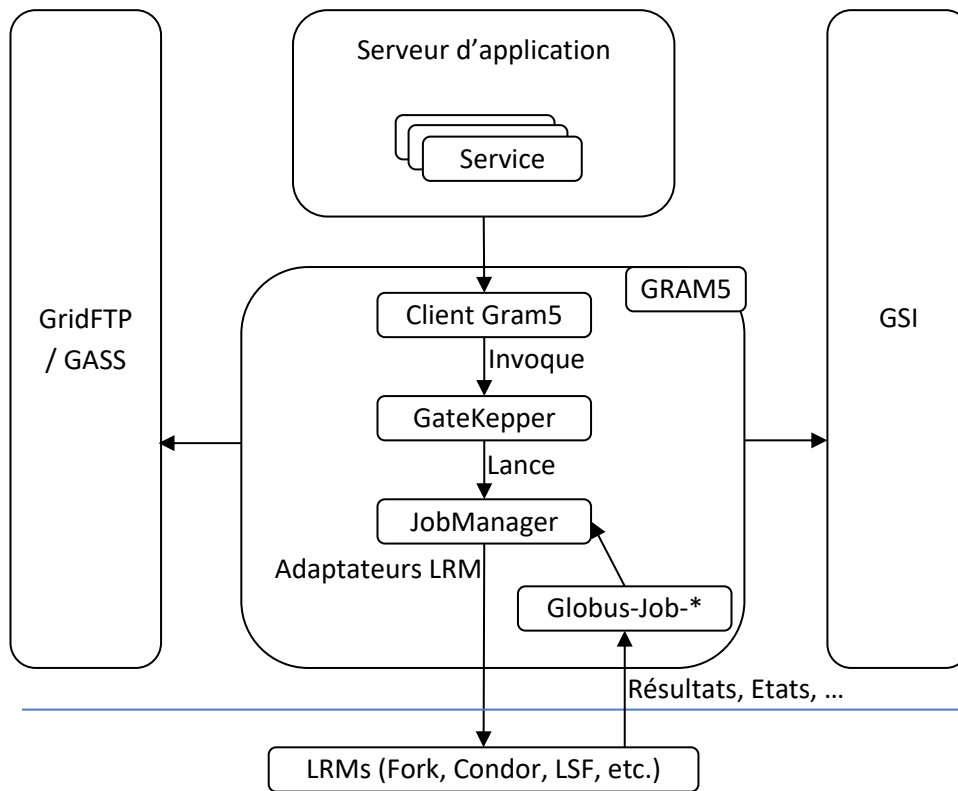


Figure 10. Exécution des services déployés dans Globus.

La gestion de l'exécution des Jobs concerne l'allocation des ressources, le lancement des fichiers exécutables et la collecte des résultats. L'exécution d'un service est pris en charge par le Globus-gatekeeper qui est responsable principalement d'autoriser et d'exécuter les services dans GRAM5. Globus-gatekeeper démarre des instances pour globus-job-manager. Ce dernier lance et contrôle les Jobs à l'aide de l'un des gestionnaires

de ressources locales LRM comme Torque³⁹, Oracle GridEngine⁴⁰ et Condor⁴¹ (Basney et Livny, 2000).

GRAM5 permet aussi l'exécution des Jobs dans des nœuds distants. Un transfert des fichiers exécutables ainsi que les données qui leurs dépendent se fait via le protocole GridFTP par le mécanisme 'Global Access to Secondary Storage' (GASS). Ce dernier, offre à la grille un support de systèmes de fichiers distribués qui facilite l'accès aux ressources distantes par des standards comme http, FTP, HPSS⁴², SRB⁴³, ... La figure 10 montre l'interaction entre les principales composantes de l'architecture de gestion de ressources dans Globus.

5.1.2. HAND

Afin de fournir la possibilité d'un déploiement dynamique dans un conteneur GT, les auteurs de l'étude réalisée dans (Qi et al., 2007) ont proposé une solution appelée HAND acronyme de (highly available dynamic deployment infrastructure). L'objectif principale de la solution est d'utiliser les ClassLoader pour rafraîchir la nouvelle liste des services déployés de manière dynamique sans le redémarrage de tout le conteneur. La solution HAND a défini six critères pour un déploiement dynamique réussit dont on cite :

- 1- Un conteneur qui reçoit des demandes de déploiement dynamique doit, si possible, terminer les requêtes utilisateur existantes ;
- 2- Les requêtes utilisateur reçues lors de l'exécution d'une procédure de déploiement dynamique doivent être gérées correctement si possible ;
- 3- La procédure de déploiement devrait être décomposée en étapes plus petites afin de réduire le risque deadlock ;

³⁹ Torque est un gestionnaire de ressources open source fournissant un contrôle sur les Jobs par lots et les nœuds de calcul distribués. Disponible sur le lien : <http://www.adaptivecomputing.com/products/torque/>.

⁴⁰ Oracle GridEngine: un scheduler de Job à la propriété d'Oracle, accessible sur <http://gridscheduler.sourceforge.net/>

⁴¹ Condor, un scheduler de Job supporté par GRAM5, accessible sur : <http://research.cs.wisc.edu/htcondor/>

⁴² HPSS : High Performance Storage System de IBM, est un système de stockage très performant et hautement scalable et connait des améliorations continues, accessible sur <http://www.hpss-collaboration.org/>

⁴³ SRB : Storage Ressource Broker du super-computer de l'université de San diego.

- 4- Des approches de redondance multiple doivent être fournies aux utilisateurs distants et locaux afin de réduire l'indisponibilité.

Pour atteindre ces objectifs, les auteurs ont proposé l'utilisation de deux types de déploiements sur deux niveaux distincts qui sont le niveau conteneur et le niveau service. Le module déployeur dynamique est composé principalement des parties suivantes :

- 1- Le déployeur du fichier GAR : ce composant est chargé d'invoquer les actions de déploiement. Ces dernières peuvent être exécutées en toute sécurité une fois que tous les services sont arrêtés, mais avant qu'un nouveau service ClassLoader est lancé. Ce service est responsable des bibliothèques de base utilisées pour exécuter le conteneur, telles que le moteur SOAP, la journalisation et la sécurité.
- 2- Le validateur : ce module est chargé de vérifier l'exactitude du fichier GAR en cours de déploiement. Il empêche le déploiement des fichiers GAR non valides ou mal formés.
- 3- Le module de journalisation : il est utilisé pour enregistrer l'historique détaillé de l'exécution des actions de rechargement.
- 4- Le Restorer : ce composant est un mécanisme de sauvegarde. En cas d'erreur, il peut aider le conteneur à restaurer son état de fonctionnement précédent.

Le module déployeur dynamique proposé dans l'étude présentée a permis d'améliorer la qualité de déploiement. Dans le premier type qui concerne le déploiement niveau service, il ne cause pas le redémarrage de tout le conteneur. Cependant, il affecte le service redéployé en le désactivant s'il est déjà actif et en fonction. Cette solution permet de réduire le temps de l'indisponibilité de service, par contre sa désactivation peut avoir des conséquences fatales pour des applications critiques en les rendant indisponible même pour une durée très limitée. Le deuxième cas concernant le déploiement dynamique réalisé au niveau conteneur. Le module déployeur impose le redémarrage de tout le conteneur pour le déploiement ou le redéploiement d'un service. Cela est due aux dépendances inter-services et de relations père-fils qui exige la désactivation d'un groupe de services qui dépend fonctionnellement par un autre service ou un autre groupe de services. Cette deuxième solution n'est toujours pas optimale car elle tolère le redémarrage du conteneur, ce qui

entraîne une non-disponibilité de tous les services déployés sur ce conteneur même pour une période très limitée.

5.1.3. DynaGrid

Dans l'étude (Byun et al., 2009), les auteurs proposent un autre Framework appelé DynaGrid qui traite le problème de déploiement dynamique, l'adaptativité d'approvisionnement des ressources, et l'invocation des services. Pour le déploiement dynamique, les auteurs proposent l'utilisation d'un service spécifique appelé Dynamic Service Launcher (DSL). Ce service s'exécute au niveau de toutes ressources de la grille et offre la possibilité de gérer la demande d'exécution de service Web sur n'importe quel conteneur Globus. Outre le déploiement dynamique, DSL effectue également la réplique des ressources à états (ServiceRessource) et la journalisation des requêtes.

Dans la solution proposée par les auteurs, le déploiement d'un nouveau service passe par les étapes suivantes :

- 1- Un service spécifique appelé 'ServiceDoor' cherche une ressource appropriée pour l'exécution du service via le service d'information de la grille.
- 2- Le serviceDoor transfère les fichiers du service vers la ressource physique et déploie le service via DSL.
- 3- Le DSL crée une méta service ressource qui contient des informations sur le service déployé y compris l'ID, la configuration et le ClassLoader du service.
- 4- Le DSL inscrit le endpoint reference (EPR) du méta service ressource au serviceDoor qui sera utilisé pour l'accès au service déployé par la suite.

Le conteneur est à ne pas redémarrer pour l'activation du nouveau service déployé. Lorsqu'il reçoit la demande de lancement du nouveau service, il utilise l'EPR passé par le client. Le DSL reçoit l'EPR du méta service ressource liée au service à exécuter afin d'extraire les informations à base de quelles il va modifier le contexte de ClassLoader. Le DSL change le contexte d'exécution selon les informations tirées de méta service ressource y compris les paramètres de configuration du service à lancer et procède à son exécution.

Cependant, la requête de l'utilisateur passe bien par un proxy. Une entité ClientProxy est générée afin d'accéder au corps du framework. Chaque client de service doit charger une instance de Client Proxy pour une ressource spécifique afin de rediriger la requête du client vers le DSL approprié en passant par le serviceDoor. Ce dernier possède tous les EPR de tous les services déployés. Il est utilisé pour localiser le méta service ressource du service demandé afin de le retourner au client. Les EPRs localisés une fois par le serviceDoor sont mis en cache du système pour des futures utilisations. Cela évite aux clients de le recontacter à chaque invocation du même service. Par contre, un EPR du service n'est toujours pas le même à cause des opérations de redéploiement ou de suppression de déploiement. Dans ces cas, les requêtes des clients doivent repasser par le serviceDoor afin de pouvoir recharger les nouveaux EPRs à jour.

Bien que cette solution offre bien un déploiement dynamique, et une gestion des requêtes d'interrogation de services par un générateur de proxy client, la gestion de la plupart des modules du framework est compliquée. Pour une future modification, amélioration, ou mise à jour, toute l'architecture va être mise en question notamment pour la gestion des versions des services, de leurs réplicants et des EPRs.

5.1.4. UNICORE

UNICORE connaît des améliorations continues et des modifications importantes dans chaque nouvelle version. Dans sa dernière version 7 qui est apparue en 2014, il a connu des modifications majeures notamment dans la prise en charge de nouveaux analyseurs de données et technologies de stockage. Le serveur UNICORE/X supporte bien la soumission de Jobs via YARN, ce qui rend possible l'utilisation des frameworks d'analyse de big data tels que l'écosystème Apache Hadoop⁴⁴, Apache Spark⁴⁵, MapReduce⁴⁶ et Apache Flink⁴⁷. De plus, UNICORE 7 intègre des éléments de stockage utilisés par les frameworks d'analyse de big data. Par conséquent, l'intégration de l'HDFS (Hadoop Distributed File System) (Honnutagi, 2014) est possible. De même, des interfaces de stockage sur le cloud

⁴⁴ Apache Hadoop, un Framework de big data basé sur YARN, MapReduce, et un système de fichier distribué appelé HDFS. Le projet Hadoop est accessible sur le lien : <http://hadoop.apache.org/>

⁴⁵ Apache Spark, est un moteur d'analyse pour le traitement de données à grande échelle. Le projet est accessible sur le lien : <https://spark.apache.org/>

⁴⁶ MapReduce : un modèle de programmation basé sur deux primitives : Map et Reduce, il est à la base de plusieurs Framework de big data.

⁴⁷ Flink, est un Framework pour le traitement distribué des flux de données. Le projet est accessible sur le lien : <https://flink.apache.org/>.

ont été intégrés à UNICORE 7 comme l'interface CDMI (Cloud Data Management Interface)⁴⁸ et Amazon S3⁴⁹, ce qui rend possible l'utilisation d'autres source de stockage (Benedyczak et al., 2016).

L'intergiciel implémente une architecture client/passerelle/serveur de trois tiers, plus le TSI (Target System Interface) qui sert d'interface avec les ressources externes. De plus, il intègre un proxy inverse HTTPS dans sa couche passerelle en permettant un accès sécurisé à la couche service. Les jobs sont transmis à travers l'infrastructure en tant que fichiers sérialisés connus par AOJ (Abstract Job Objects). Ces derniers contiennent des données et des descriptions de tâches de calcul. Le premier tiers de l'architecture 'client' offre une interface graphique ou de ligne de commande. Il est le moyen de lancement des jobs et de récupération de leurs résultats. Le second tiers, la passerelle, permet d'assurer la sécurité des interactions ainsi que la transmission des AOJs à la grille. Cependant, le troisième tiers permet l'exécution des tâches sur des plateformes et des sites locaux ou externes.

Le déploiement des services de grilles dans UNICORE se fait au niveau de la couche 'services' via un service spécialisé appelé 'Registre'. Ce dernier permet aux clients de découvrir les services disponibles hébergés par un ou plusieurs serveurs UNICORE/X. un client fait recours au service de registre pour localiser un service de grille déployé. Le registre retourne le fichier descriptif du service localisé et permettre au client de l'interroger. Tous les services dans UNICORE respectent les spécifications WSRF, et peuvent être RESTful ou non. Cependant, un orchestrateur de services opérant sur la couche 3 est chargé d'exécuter les tâches individuelles dans un workflow, ainsi que de gérer leurs états. Différentes stratégies de courtage sont mises en œuvre afin de repérer les ressources les mieux adaptées pour chaque étape du flux de travail. Garce à la nature faiblement couplée de l'intergiciel, il est possible de personnaliser la partie workflow en adoptant des stratégies de courtage plus adaptées ou en intégrant un orchestrateur de services répondant le mieux aux besoins des applications des clients.

⁴⁸ CDMI, définit l'interface que les applications utiliseront pour créer, récupérer, mettre à jour et supprimer des éléments de données du cloud. Accessible sur le lien <https://www.snia.org/cdmi>.

⁴⁹ Amazon Simple Storage Service (Amazon S3) : est un service web de stockage, évolutif, conçu pour la sauvegarde et l'archivage en ligne des données et des programmes d'application. Accessible sur le lien : <https://aws.amazon.com/s3/>.

5.1.5. Autres travaux

D'autres travaux ont été réalisés en vue d'améliorer le déploiement dans son aspect dynamique et/ou distant automatique. Parmi ces travaux on cite Nimbus (Keahey et al., 2009), hot depl srv. (Frieese et al., 2004), WSPeers (Harrison et Taylor, 2005), VMPlants (Krsul et al., 2004), l'étude (Kecskemeti et al., 2011), etc.

5.2. Critiques

Les insuffisances soulevées sur le déploiement des services dans une grille ainsi que leur exécution sont en nombre de cinq :

- 1- Absence de déploiement dynamique : Globus et Unicore offrent une solution classique de déploiement comme celle existante dans un conteneur JavaEE. Un déploiement d'un nouveau service consiste à son inscription dans la liste des services déployés et qui est gérée par le conteneur. Les modifications de la liste des services déployés est prise en considération une fois le conteneur est redémarré. Le déploiement et le redéploiement nécessite le redémarrage du conteneur, ce qui implique l'altération des tâches qui sont déjà en cours d'exécution. Cette altération rend indisponible, pour un certain temps, tous les autres services déployés dans le conteneur en question.
- 2- Services non mesurables : l'utilisation des ressources par un service n'est pas facilement mesurée. Les tâches lancées sont exécutées sur des ressources approvisionnées à base de la description de ces tâches. Si une demande de ressources supplémentaire ou inférieure à ce que c'était prévue est passée, une intervention humaine est exigée pour modifier les paramètres d'approvisionnements précédentes, et donc la facturation des services fournis. La facturation des ressources est une opération très importante pour les clients comme pour les fournisseurs. L'intervention humaine pour la facturation est très compliquée et insuffisante face à un grand nombre de services à facturer.
- 3- Approvisionnement non adaptatif des ressources : lorsqu'un service lance un ou plusieurs Jobs, il spécifie les ressources requises pour le bon accomplissement des tâches demandées. Cela cause une non adaptabilité de la consommation des ressources réservées en raison d'absence de la notion d'élasticité rapide. Cette

dernière permet d'allouer et de libérer de façon rapide et fiable les ressources selon la consommation réel des Jobs lancés.

- 4- Module d'exécution non extensible : on remarque que l'intégration de nouveaux gestionnaires de tâche est délicate afin d'exploiter une multitude de type de ressource de calcul et de stockage. L'ajout de nouveaux gestionnaires de ressources nécessite une modification de l'un des composantes des solutions précédentes. Au niveau de Globus à titre d'exemple, cet ajout se fait en intégrant une interface à Gram5 pour prendre en considération le nouveau LRM à ajouter. Cette intégration nécessite plusieurs mises à jour à plusieurs niveaux notamment l'aspect de sécurité et d'adaptation de la représentation des Jobs à soumettre pour que le nouveau LRM les reconnait. Au niveau de Unicore, le TSI se charge d'interfaçage entre le serveur UNICORE/X et les gestionnaires de ressources externes. Néanmoins pour prendre en charge un nouveau gestionnaire, on doit modifier le niveau TSI et l'adapter aux spécificités du nouvel élément.
- 5- Absence de réplication des conteneurs : les conteneurs peuvent être vus comme une collection d'outil qui permet d'inscrire un service, l'exécuter, le désinscrire, et le rendre découvrable par d'autres entités externes. L'échec d'un conteneur cause l'altération des services déployés. Pour cela, un mécanisme de réplication de conteneur est très bénéfique pour le bon fonctionnement de la grille. Si un conteneur est interrompu ou endommagé pour une raison ou pour une autre, ça ne doit pas empêcher le bon déroulement des services déployés sur ce conteneur. Un conteneur de service doit être répliqué pour contenir la dernière bonne version de l'original. Et il doit être prêt à prendre le relai à tous moments sans affecter l'environnement d'exécution des services, ni les demandes de leur invocation.

6. Proposition de solution

La solution proposée doit couvrir les cinq insuffisances rencontrées avec les solutions adoptées dans l'état de l'art. pour se faire, on propose l'ajout des trois fonctionnalités suivantes pour toutes architecture d'un intergiciel ou de Framework de Grille :

- 1- Externaliser les conteneurs du serveur d'application.

- 2- Utiliser la technologie des systèmes et applications en images pour la mise à jour des conteneurs après chaque opération de déploiement, redéploiement et annulation de déploiement.
- 3- Ajout d'un module extensible qui adopte la notion des plugins pour son évolution. Ce module va être l'appui pour supporter l'externalisation des traitements vers différents gestionnaires de tâches. Ainsi de supporter l'interfaçage avec les systèmes de stockage les plus récents.

6.1.Externalisation des conteneurs de services

Ce point est dédié principalement à la résolution des problèmes liés aux déploiement dynamiques et automatiques des services. Isoler le conteneur du reste de système d'exécution des services permet une gestion flexible de ce conteneur.

L'externalisation diffère selon le conteneur en question. Physiquement, les conteneurs de services sont un ensemble de fichiers et parfois une structure dans une base de données. Un service déployé est traduit par la copie de ses fichiers composantes dans des répertoires spécifiques. L'arborescence des conteneurs inclut différents emplacements pour chaque type de fichiers tels que les descripteurs d'interfaces des services, fichiers de journalisation, fichiers de configuration, fichiers du code métier, etc. De plus, le registre des services déployés qui représente une entité principale pour le déploiement, est aussi sous forme de fichiers qu'on peut manipuler déplacer et copier si on a les autorisations suffisantes. Cependant, le conteneur est lié avec un ensemble de services de découverte, de sécurité et autres. En outre, le conteneur est lié aussi à un contexte d'exécution qui concerne tous les services déployés. Par conséquent, l'externalisation du conteneur se fait en suivant les étapes suivantes :

- Créer un conteneur de ressource physique avec docker, cela permet d'avoir des ressources physiques propres au conteneur des services. Ainsi, sa gestion avec Kubernetes ou autre orchestrateur de conteneur sera possible.
- Installer un système d'exploitation approprié dans le conteneur de ressources physique créé. Ce système va héberger le conteneur en tant qu'un ensemble de fichier et d'applications.

- Copier les fichiers du conteneur des services, ainsi que toutes applications qui peut lui être utile dans le nouveau système installé. Les applications qui peuvent être utiles sont les gestionnaires de fichiers et de bases de données qui contiennent les données du conteneur.
- Configurer le serveur d'application et les services liés au conteneur de service de façon à modifier les paramètres utiles pour y accéder. Cette configuration concerne principalement la connexion entre le conteneur des services et les autres modules.

La figure 11 illustre la vue architecturale du conteneur des services externalisé.

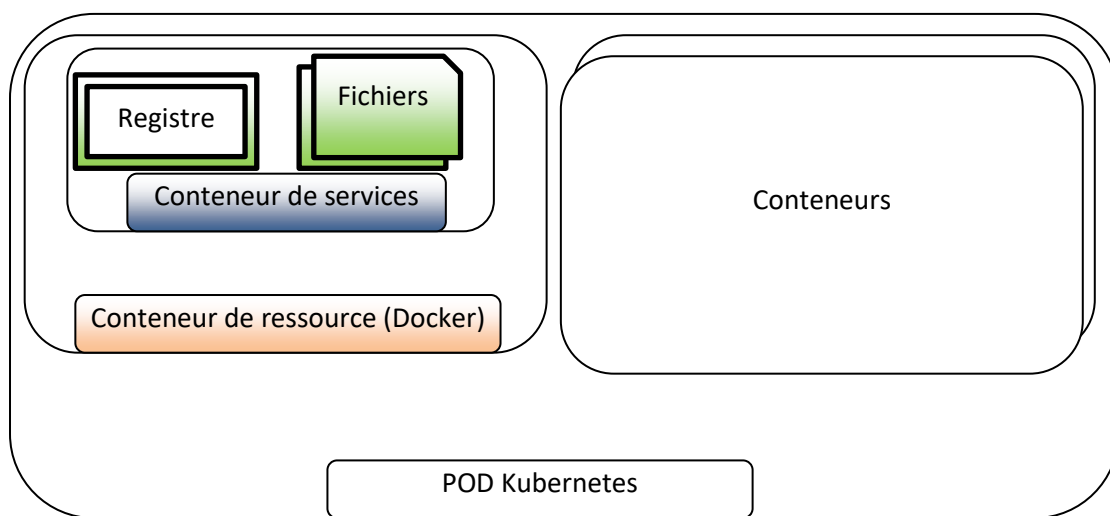


Figure 11. Conteneur des services de grille externalisé dans un conteneur Docker et géré par Kubernetes.

Le conteneur externalisé est considéré comme un système qui gère un ensemble de fichiers et de tables. Ce système est consultable par les services de base de la grille comme le service d'information et de sécurité. Lorsque le serveur cherche un service demandé par une autre entité, ce serveur accède au registre et au fichier du conteneur externalisé. Le conteneur lui répond avec une réponse négative ou positive. Dans le dernier cas, le conteneur externalisé retourne le EPR du service demandé afin qu'il puisse être lancé par le serveur. Avec cette technique, les problèmes liés au déploiement de nouveau service et de redéploiement vont être surmontés comme montré dans la section suivante.

6.2. Utilisation des images pour la mise à jour des conteneurs de ressources

Les images d'applications sont une manière efficace de déploiement et de gestion des mises à jour du software. L'externalisation du conteneur de service installé sur le

conteneur de ressource créé dans l'étape précédente permet d'effectuer des opérations de clonage, de push et de pull. Ces fonctionnalités vont permettre un déploiement dynamique souple et une réplication du conteneur de service de façon fiable avec une possibilité de gestion des versions. Ci-dessous l'impact de l'utilisation des images pour le déploiement des services de grille :

- Les images sont facilement gérables par des serveurs d'images tels que docker et NATS⁵⁰.
- Le déploiement d'un nouveau service, ou son redéploiement ou toutes autres modifications du conteneur de service, résulte la création d'une nouvelle version d'une image déployable.
- Une nouvelle version d'image possède les mêmes propriétés de connexion que l'image de la version antérieure. Les propriétés concernées sont l'adresse IP, l'adresse MAC, et le nom de l'hôte.
- Le déploiement d'une nouvelle version d'image est assuré par le serveur d'image. Il exécute une fonction push pour l'insérer et de pull pour la retirer et restaurer la plus ancienne qui est supposée plus stable. Cela permet de lancer une nouvelle version du conteneur contenant toutes les modifications portées sur les services.
- La restauration d'une ancienne version d'une image du conteneur de services est réalisable en lançant une opération pull. Avec cette fonction, on peut revenir vers une version antérieure du conteneur afin de rectifier des erreurs ou restaurer des fichiers endommagés ou perdus.
- Les opérations push et pull sont effectuées d'une manière transparente. La nouvelle image prend le relais avec l'application des modifications apportées sur le conteneur des services. Le serveur d'images redirige la communication avec l'ancienne image vers celle déployée récemment. Les erreurs qui peuvent être produits à cause du changement de l'état du conteneur des services sont remontées et traitées, si possible, par le serveur d'images.

⁵⁰ NATS, un serveur d'image pour Windows et Linux. Le projet est accessible via le lien : <https://nats.io/>

6.3. Ajout d'un module extensible pour utiliser les ressources externes

L'exécution des Jobs issues des services doit pouvoir être faite sur n'importe quelle infrastructure externe qui fournit les ressources d'exécution nécessaires. En analogue avec le module TSI de Unicore, le nouveau module ajouté doit pouvoir être extensible en plugins et s'interface avec des orchestrateurs de conteneurs, des gestionnaires de ressources, et des systèmes de stockage de données.

L'interfaçage avec les orchestrateurs de conteneurs tels que Kubernetes, Docker Swarm, apache Mesos, et des gestionnaires de ressources tels que YARN et Slurm permet de faciliter la communication avec plusieurs fournisseurs de services comme Amazon et Azure car il adopte les applications conteneurisées de manière générale. Cela offre la possibilité d'approvisionner les ressources de calcul nécessaires et bénéficier de la propriété de l'élasticité rapide avec laquelle est caractérisée un cloud. De même, les systèmes de stockage distribués sont d'une importance élevée pour la bonne exécution des Jobs. Les données requises pour l'accomplissement des tâches sont copiées et répliquées à travers le système de fichier distribué. Afin de pouvoir utiliser ces systèmes, un interfaçage est nécessaire pour transmettre toutes les données de l'host local dans la grille vers le système de stockage externe.

L'aspect architectural des plugins est structuré de trois modules principaux :

- 1- Proxy : afin d'assurer la sécurité des transactions entre le plugin et la ressource externe, le plugin est identifié grâce à un certificat SSL. Le proxy joue le rôle d'intermédiaire entre la grille et la ressource externe. Il commence par recevoir une requête d'exécution d'un Job ou de transmission de données vers une ressource externe. Ensuite, il redirige cette requête vers le bon destinataire qui fait confiance au proxy grâce au certificat existant entre les deux. Par conséquent, le plugin arrive à contacter les gestionnaires de ressources externes.
- 2- Parseur : ce module joue le rôle d'interpréteur qui transforme la requête émise par la grille à une autre transmise aux ressources externes. Généralement, les Jobs et les ressources de la grille utilisent le langage descriptif 'Resource Specification Language (RSL)' pour spécifier ses propriétés qui sont le fichier source du job, les

performances requises, etc. Le parseur transforme ce fichier descriptif basé sur XML à un autre compris par le gestionnaire des ressources externes.

- 3- Lanceur de commande : ce module représente la partie cliente des gestionnaires des ressources externe. Il lance des commandes en tant qu'un utilisateur reconnu par les gestionnaires grâce au module proxy. Les commandes à lancer sont en forte liaison avec le gestionnaire de ressource externe qui exige l'utilisation de ses propres commandes et sa propre syntaxe.

La figure 12 récapitule toute l'architecture de la solution proposée.

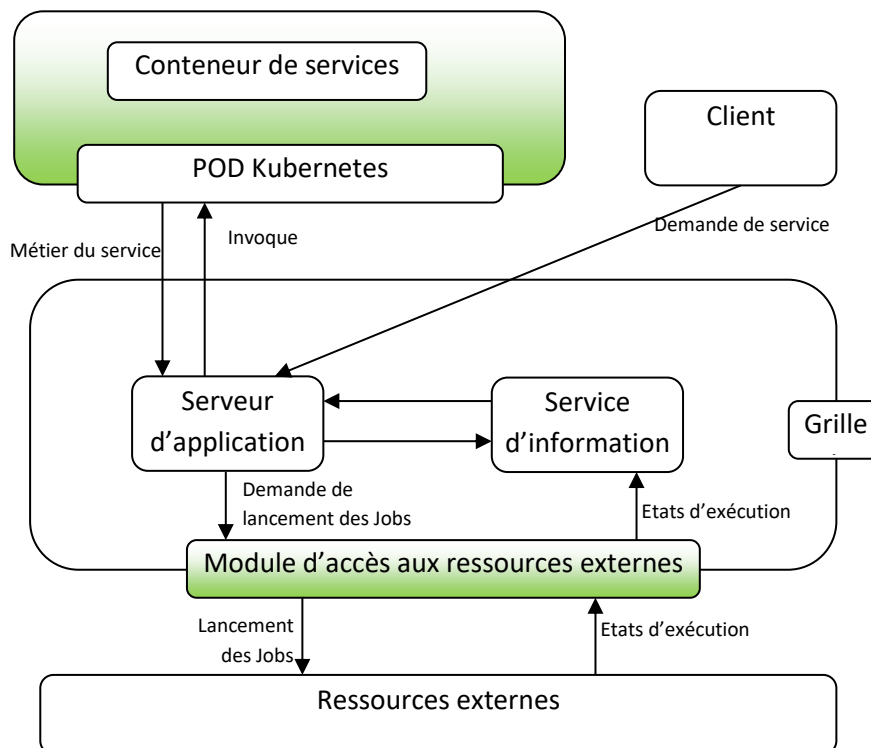


Figure 12. Vue architecturale de la solution proposée.

7. Conclusion

Dans le courant chapitre, on a présenté un état de l'art de déploiement des services dans les grilles de calcul, ainsi que la manière dont ils sont interrogés via l'exposition des mécanismes des intergiciels existants, et des études qui ont proposé des solutions dans le même contexte. Ensuite, on a soulevé les insuffisances marquées et on a proposé une solution pour les corriger.

Dans la solution présentée on a proposé de réutiliser les techniques de conteneurisation pour surmonter les problèmes de dynamisme et automatisme des déploiements. Cette solution est inspirée des mécanismes utilisés par les cloud et elle a montré son efficacité et sa capacité de fournir une qualité de service optimale. Cependant, dans la partie interaction avec les ressources externes on a choisi une solution modulaire basée sur la notion de plugin afin de s'interfacer avec le maximum d'éléments externes sans modifier le noyau de gestion des tâches de la grille. Les techniques basées sur les plugins sont adoptées de plus en plus en génie logiciel car elle a connu du succès dans les applications qui nécessitent une croissante évolution poussée par la variation des demandes des utilisateurs en fonctionnalité, en nombre et en qualité.

8. Références

- Apache Mesos. (2018). Retrieved from <http://mesos.apache.org/>
- Application Server | IBM Cloud. (2018). Retrieved from <https://www.ibm.com/cloud/websphere-application-platform>
- Basney, J., & Livny, M. (2000). Managing network resources in Condor. In *High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on* (pp. 298-299). IEEE.
- Bauer, M. (2006). Paranoid penguin: an introduction to Novell AppArmor. *Linux Journal*, 2006(148), 13.
- Benedyczak, K., Schuller, B., Petrova-El Sayed, M., Rybicki, J., & Grunzke, R. (2016, July). UNICORE 7—Middleware services for distributed and federated computing. In *High Performance Computing & Simulation (HPCS), 2016 International Conference on* (pp. 613-620). IEEE.
- Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, omega, and kubernetes. *Queue*, 14(1), 10.
- Byun, E. K., & Kim, J. S. (2009). Dynagrid: An adaptive, scalable, and reliable resource provisioning framework for wsrp-compliant applications. *Journal of Grid Computing*, 7(1), 73-89.
- Codrops, L. (2018). Apache TomEE. Retrieved from <http://tomee.apache.org/>
- Friese, T., Smith, M., & Freisleben, B. (2004, November). Hot service deployment in an ad hoc grid environment. In *Proceedings of the 2nd international conference on Service oriented computing* (pp. 75-83). ACM.
- Harrison, A., & Taylor, I. (2005, February). Dynamic web service deployment using WSPeer. In *Proceedings of 13th Annual Mardi Gras Conference-Frontiers of Grid Applications and Technologies* (pp. 11-16).
- Hightower, K., Burns, B., & Beda, J. (2017). *Kubernetes: Up and Running: Dive Into the Future of Infrastructure*. " O'Reilly Media, Inc."
- Honnutagi, P. S. (2014). The Hadoop distributed file system. *IJCSIT*, 5, 6238-6243.
- IBM Knowledge Center. (2018). Retrieved from https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/cjta_trans.html
- J2EE Containers. (2018). Retrieved from https://docs.oracle.com/cd/E17802_01/j2ee/j2ee/1.4/docs/tutorial-update6/doc/Overview3.html
- Java EE Containers - The Java EE 5 Tutorial. (2018). Retrieved from <https://docs.oracle.com/javase/5/tutorial/doc/bnabo.html>
- JBoss Developer. (2018). Retrieved from <http://www.jboss.org/>
- Kamp, P. H., & Watson, R. N. (2000, May). Jails: Confining the omnipotent root. In *Proceedings of the 2nd International SANE Conference* (Vol. 43, p. 116).
- Karanasos, K., Suresh, A., & Douglas, C. (2018). Advancements in YARN Resource Manager. *Encyclopedia Of Big Data Technologies*, 1-9. doi: 10.1007/978-3-319-63962-8_207-1
- Keahey, K., Tsugawa, M., Matsunaga, A., & Fortes, J. (2009). Sky computing. *IEEE Internet Computing*, 13(5), 43-51.
- Keckskemeti, G., Terstyanszky, G., Kacsuk, P., & Neméth, Z. (2011). An approach for virtual appliance distribution for service deployment. *Future Generation Computer Systems*, 27(3), 280-289.
- Krsul, I., Ganguly, A., Zhang, J., Fortes, J. A., & Figueiredo, R. J. (2004, November). Vmplants: Providing and managing virtual machine execution environments for grid

- computing. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing* (p. 7). IEEE Computer Society.
- libvirt: The virtualization API. (2018). Retrieved from <https://libvirt.org/>
- Marmol, V., Jnagal, R., & Hockin, T. (2015). Networking in containers and container clusters. *Proceedings of netdev 0.1, February*.
- McCarty, B. (2005). *Selinux: Nsa's open source security enhanced linux* (Vol. 238). O'Reilly.
- Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux Journal, 2014(239), 2*.
- Open source container-based virtualization for Linux. (2018). Retrieved from <https://openvz.org/>
- Opencontainers/runc. (2018). Retrieved from <https://github.com/opencontainers/runc>
- Oracle WebLogic Server Technical Information. (2018). Retrieved from <http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html>
- Production-Grade Container Orchestration - Kubernetes. (2018). Retrieved from <https://kubernetes.io/>
- Qi, L., Jin, H., Foster, I., & Gawor, J. (2007, February). Hand: Highly available dynamic deployment infrastructure for globus toolkit 4. In *15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'07)(PDP)*, Naples, Italy, 2007, pp. 155-162.
- Rosen, R. (2013). Resource management: Linux kernel namespaces and cgroups. *Haiflux, May, 186*.
- Rosen, R. (2014a). Linux containers and the future cloud. *Linux J, 240(4), 86-95*.
- Rosen, R. (2014b). *Linux kernel networking*. New York: Apress.
- Tierney, B., Lee, J., Chen, L. T., Herzog, H., Hoo, G., Jin, G., & Johnston, W. E. (1994, October). Distributed parallel data storage systems: A scalable approach to high speed image servers. In *Proceedings of the second ACM international conference on Multimedia* (pp. 399-405). ACM.
- UNICORE | Distributed computing and data resources. (2018). Retrieved from <https://www.unicore.eu/>
- Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., ... & Saha, B. (2013, October). Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing* (p. 5). ACM.

Chapitre 3 : La vidéosurveillance et les systèmes distribués

Chapitre 3 : la vidéo surveillance et les systèmes distribués

1. Introduction

Avec l'évolution et l'apparition de nouvelles menaces sur la sécurité des personnes et des entreprises, et avec les besoins accumulés de la supervision des routines publiques à grande échelle, la vidéosurveillance s'avère de plus en plus un élément indispensable et crucial pour le développement de l'infrastructure de sécurité de l'Algérie. Son utilité s'étale sur de nombreux domaines d'applications tels que la reconnaissance, la recherche de personne ou d'objet, la prévention contre des attaques de multi-origines, la supervision des incendies, ainsi que plusieurs autres activités dans le domaine militaire et autres.

Sur le plan juridique, l'émergence de la vidéosurveillance a présenté pour certains une violation de la vie privée des personnes, tant dit que pour d'autres été juste une réduction de la sphère de la vie privée au profit d'un niveau de sécurité collectif estimé meilleur. En 1949, la notion de 'Big Brother' (Whitaker, 2001) a fait surface dans le roman nommé 'Nineteen Eighty-Four' (Orwell, 2009) de George Orwell en Angleterre. Le roman est connu aussi sous le nom de 'Big Brother vous regarde', il trace déjà une vue du futur où chaque individu est surveillé de façon inaperçue, rassurante et au même temps sévère. Conformément avec la notion de Big Brother, des entreprises informatiques telles que Facebook, Google et Microsoft sont considérées comme trop envahissantes à cause du suivi des activités de ses utilisateurs. Cependant, des lois et des commissions spécialisées ont été créées dans le monde pour définir les principes à respecter lors de la collecte, du traitement et de la conservation des données personnelles. Cela est effectué tout en offrant une certaine flexibilité pour l'exploitation de la technologie liée aux données privées telle que le réseau Internet et la vidéosurveillance.

Plusieurs événements sur le plan de la technologie ont marqué l'histoire de la vidéosurveillance. La création de la télévision dans les années 40 du siècle précédent, à accélérer le processus d'apparition de la vidéosurveillance qui a eu lieu en Allemagne nazie en 1942 par la compagnie SIEMENS. L'objet de cette première utilisation été l'observation de plus près le lancement des missiles balistiques. La technologie utilisée a permis la capture des images en noire et blanc par une caméra et les transmettre via un câble à un moniteur pour les visualiser, ce qui a mis au point les éléments fondamentaux de la technologie

CCTV⁵¹ (TV en circuit fermé). Le deuxième évènement technologique été guidé par les besoins excessifs d'enregistrer les séquences vidéo capturées. Cela a conduit à l'apparition des caméras capables de stocker les images captées dans des supports d'enregistrement de type bande magnétique. Cette invention a eu lieu en 1954 par la compagnie RCA⁵² Victor (Radio Corporation of America). Plusieurs autres évènements technologiques ont véhiculé parallèlement le développement de la vidéosurveillance dont on cite :

- Le passage de la représentation analogique des données vers la représentation numérique : cela a influencé sur la façon de transfert et de stockage des séquences vidéo captées. Les données passent par des étapes de codage et décodage dans les deux extrémités du support de transmission, i.e. au niveau de la caméra et au niveau d'un ordinateur connecté à cette caméra. Les données sont stockées sur des supports de stockage numériques tels que des disques durs et non plus sur des bandes magnétiques.
- L'apparition et l'évolution des réseaux : ce qui a permis d'ouvrir les portes sur la vidéosurveillance IP grâce à laquelle l'accès distant à des ressources de capture vidéo, la transmission, la réception et le partage de flux multimédia est devenu possible.
- L'évolution des unités de calcul et des ordinateurs : plus l'unité de calcul est puissante, et plus la mémoire vive de cette unité est considérable, plus elle est capable de réaliser rapidement et avec peu ou sans anomalies plus de traitement. La capacité ou la performance de l'unité de calcul reste toujours un facteur important pour mesurer la performance de réalisation d'une tâche.
- L'évolution de la technologie de calcul parallèle et distribué : le calcul parallèle et distribué a fourni une autre dimension au traitement des données multimédias en se servant de multiples unités de calcul, des grappes ou même des centres de calcul. Cela a permis le passage de traitement sur mono processeur vers le traitement sur multi cœurs et multi processeur, et le passage de la petite échelle vers la grande échelle où le nombre des unités de traitement qui peuvent être impliqués dans la réalisation d'une tâche est énorme et théoriquement illimité.

⁵¹ Closed-Circuit Television : appellation commune des caméras analogiques

⁵² Radio Corporation of America : ancienne entreprise électronique spécialisée dans la conception et la fabrication de systèmes de vidéo et d'image numérique appartient aujourd'hui à Technicolor.

- L'évolution des appareils d'acquisition : le remplacement des caméras analogiques par des caméras numériques a permis d'acquérir des données multimédia de meilleures qualités. Cette amélioration a été visible que ce soit au niveau résolution des images et au niveau de la quantité et la qualité d'information incluse dans une unité de présentation de la donnée. L'augmentation de la qualité de données a causé l'augmentation de leur taille, ce qui a posé de nouveaux défis concernant leur transmission et les performances de leur traitement.
- L'avancement sur les techniques de traitement des données multimédia : l'évolution des algorithmes de l'intelligence artificielle, ainsi que les techniques de détection et d'extraction des caractéristiques des vidéos ont marqué l'évolution de la vidéosurveillance. Ils ont conduit à la réalisation avec beaucoup plus de précision des opérations qui étaient de la fiction dans une époque de temps tels que le suivi de mouvement et la recherche d'une personne dans une ville en appliquant une reconnaissance faciale ou autres applications.

Dans le reste du chapitre on va présenter des terminologies liées à la vidéo surveillance dans la section 2, ensuite on va donner une vue profonde des composants d'un système de vidéosurveillance en section 3, suivi par un listing des architectures de ces systèmes et leur relation avec la distribution des traitements et de stockage en section 4. A la fin, la section 5 conclut le chapitre.

2. Définition du domaine

Dans cette section, on va présenter les différentes notions du domaine de la vidéosurveillance pour mettre en évidence les points de leurs intersections et de leurs différences.

2.1.La surveillance

Quel que soit le domaine dans lequel la surveillance est défini, on trouve les verbes d'observation en commun tels que 'superviser', 'observer', 'regarder'. Cependant, la définition qui s'associe bien au contexte de ce mémoire est celle trouvée dans l'encyclopédie 'Collins Discovery' ('Surveillance. (n.d.) Collins Discovery Encyclopedia, 1st edition', 2005). La définition originale citée en anglais est : « close observation or supervision maintained over a person, group, etc., esp one in custody or under suspicion ». Avec la définition citée et qui date de l'année 2005, après les attentats du 11 septembre, on

peut ressentir une manœuvre pour rediriger la surveillance vers la protection des biens. Cela implique de surveiller de plus près, de façon discrète ou non, les individus, et les groupes soupçonnés par l'ordre de l'état.

2.2.La contre surveillance

La contre surveillance est toute activité qui a pour but de renforcer la sécurité privée afin d'éviter la surveillance, ou au moins la rendre difficile. Dans le domaine informatique, beaucoup de techniques et de méthodes ont été développées pour arriver à ce but. A titre d'exemple, on cite la création de nouveaux protocoles d'échange sécurisés de l'information, l'avancement dans la cryptographie, le renforcement de la sécurité des systèmes d'exploitation et des systèmes de gestion de bases de données, l'utilisation des pare-feux que ce soit matériel ou logiciel, tous font partie de la contre surveillance. Mais malgré tous les efforts de la communauté, protéger la vie privée reste un défi majeur dont tous les acteurs doivent faire face.

2.3.La vidéosurveillance

Le mot vidéosurveillance fait référence à l'ensemble des équipements logiciels et matériels qui permettent à une autorité, société, individu ou groupe d'individus de mettre en place un système qualifié pour effectuer des tâches de surveillance à distance dans des lieux publics, ou privés. Les tâches de surveillance à distance sont effectuées à l'aide d'un dispositif de capture d'image et de son. Ces derniers seront transmis à un équipement de contrôle pour les traiter, les archiver, ou juste les afficher aux écrans dans le cas des images, ou le faire écouter à partir d'une sortie voix pour le son. Les données multimédias capturées par la vidéosurveillance constituent plus de 33% des données du big data en 2014, ce qui est énorme en incluant des jours, des mois et même des années de données de la vidéosurveillance. Le taux de cette grande quantité de données va être réduit jusqu'à 14% en 2020 au profit des données issues d'autres domaines de la technologie d'information tels que les systèmes embarqués, les téléphones intelligents, etc. (Zwolenski & Weatherill, 2014).

Avoir le droit d'installer un système de vidéosurveillance nécessite une autorisation par les autorités en vigueur. En Algérie, le domaine de la vidéosurveillance été considéré sensible à partir de 2009, seul la Direction Générale de la Sûreté Nationale (DGSN) est en

mesure d'attribuer des autorisations, et des contrats avec des prestataires de service pour ce genre de secteur considéré sensible. Cependant, chaque pays prend les mesures de sécurité qui lui convient, en France une commission indépendante CNIL (Commission nationale de l'informatique et des libertés) a été créée par une loi qui date de 1978. Cette commission assure la protection de la vie privée des individus en acceptant ou non des nouvelles lois, tel que la loi de 'pasqua'⁵³ qui date de 1995 et traite la problématique d'enregistrement des séquences de la vidéosurveillance ; et accorde ou non le droit d'utiliser la vidéosurveillance en tenant en considération la nature des activités, les endroits, et le nombre de caméra de surveillance qui seront déployées.

2.4.La télésurveillance

La télésurveillance est un système de surveillance efficace contre les intrusions, la détection des incendies et des accidents dans les locaux de travail en temps réel. Elle est souvent dotée d'un centre d'alarme distant dirigé par un personnel formé et qui travail 24/24 et 7jours/7, des caméras de surveillance, une sirène, des haut-parleurs, un système d'activation/désactivation, et un ensemble de capteur de différente nature selon les besoins tels que des capteurs de mouvement, de la fumée, de comportement suspect, etc. Lors de la détection d'un événement suspect, une alarme est déclenchée au niveau du centre d'alarme, le personnel se branche sur l'endroit du fait, une procède à une vérification et confirmation de l'événement afin d'identifier la(s) personne(s) en cas d'une intrusion, ou identifier l'accident ou l'endroit de l'incendie. Après l'identification de l'événement déclenchant l'alarme, les forces de l'ordre seront notifiées pour prendre les mesures de sécurité nécessaires.

La télésurveillance ne passe à l'enregistrement des séquences vidéo que si l'alarme est déclenchée, ce qui est considéré comme un avantage en profit de la vie privée des personnes contrairement à la vidéosurveillance. Une combinaison d'un système de vidéosurveillance et de télésurveillance est envisageable dans le cas d'une supervision, surveillance d'une activité suspecte ou d'un incendie en temps réel. Notamment la liaison d'un système de vidéosurveillance à un ou plusieurs centres de télésurveillance est possible dans le cas de besoins.

⁵³ Loi française n° 95-73 du 21 janvier 1995 d'orientation et de programmation relative à la sécurité

2.5.La vidéosurveillance IP

La vidéosurveillance est dite IP en analogie avec la vidéosurveillance analogique connue par son utilisation des caméras analogiques qui produisent une bonne qualité d'images ce qui est favorable pour les tâches d'identification. La vidéosurveillance IP offre plein d'avantage en se servant de sa liaison réseau, différents types de connexion peuvent être utilisés tels que les réseaux sans fil, les réseaux locaux ou tout simplement par le réseau internet. La mise en place d'un système de vidéosurveillance IP peut avérer coûteux dans le cas d'absence d'un réseau, autrement, il peut avoir un coût très réduit si une installation réseau est déjà présente. L'utilisation de la vidéosurveillance IP offre non seulement l'accès à distance aux ressources utilisées, mais elle permet le streaming en temps réel ou non, à partir de n'importe quel appareil doté d'un software capable de recevoir du flux vidéo, un PDA ou une simple tablette peut être utile.

Un système de vidéosurveillance IP nécessite un serveur doté de logiciels spécifiques afin de contrôler et de configurer à distance les caméras de surveillance déployées, ainsi qu'à enregistrer, sécuriser, gérer et partager les séquences vidéo capturées.

Le système de vidéosurveillance IP est souvent confondu avec une caméra IP, cette dernière est considérée comme sophistiquée en offrant des fonctionnalités intégrées de luxes telles que la capture de mouvement, la reconnaissance faciale, la capture thermique, comptage, etc.

2.6.La vidéosurveillance intelligente

La vidéosurveillance intelligente (VSI) est un système d'analyse d'images créé pour assister le personnel de la sécurité à détecter les événements suspects. Avec l'œil de l'être humain, on ne peut pas distinguer tous les détails dans une scène en temps réel là où il y a une activité suspecte qui se produit. Des études ont montré qu'un surveillant ne peut suivre attentivement 9 à 12 caméras plus de 15 minutes (Hampapur et al., 2003), alors, la VSI est créée pour réduire le personnel et le libérer de la surveillance continue par faute de manquer un événement critique. La VSI met à la disposition du personnel de la vidéosurveillance un ensemble d'outils logiciels, et/ou matériels qui se base sur des techniques de l'intelligence artificielle. Le serveur de la vidéosurveillance reçoit les images capturées par les caméras de surveillance, des traitements sur ces images sont possibles est

configurables par l'administrateur du système. Une fois que les images sont arrivées au niveau du serveur, leur analyse est faite par des techniques basées sur des algorithmes de l'intelligence artificielle. Dans le cas où les images sont affichées sur des moniteurs, ces logiciels peuvent encadrer une partie de l'image qui correspond à un incendie, une activité suspecte, un objet suivi, un objet suspect détecté, etc. de façon à indiquer aux agents derrière les moniteurs les détails qui peuvent correspondre à des fraudes ou à des accidents. Dans le cas où les images ne sont pas affichées sur des moniteurs, une alerte sera lancée si un événement spécial est souligné.

2.7.La vidéosurveillance en Algérie

Un système national de vidéosurveillance est réalisé à l'échelle du territoire national, il est orienté plutôt vers la vidéo protection et assure des tâches de base soulignées par un décret présidentiel promulgué au journal officiel le 23 août 2015. Le système de télésurveillance est défini par ce décret comme étant un outil technique de connaissance et d'anticipation qui a pour objectif de contribuer à la lutte contre le terrorisme, la prévention des actes criminels, la protection des personnes et des biens, et la préservation de l'ordre public. Il a également pour objectifs la régulation de la circulation routière, la sécurisation des édifices et sites sensibles ainsi que la gestion des situations de crise ou de catastrophes naturelles.

En 2004, la DGSN a mis en place un système de vidéosurveillance constitué en plusieurs caméras qui couvre les principales artères de la capitale, 3000 caméras sont déjà opérationnelles seulement à Alger en 2015. Les grandes agglomérations à l'instar d'Annaba, Oran, Blida, Constantine et Ghardaïa ont reconnu la mise en place des systèmes de vidéosurveillance, et des études pour élargir ce réseau sur d'autres Wilaya du pays est en cours de réalisation. Chaque système de vidéosurveillance est lié à un centre de télésurveillance dirigé par la DGSN installé au niveau des wilayas concernées, ces derniers sont raccordés à leurs tours à un centre principal dit national situé à Alger et dirigé par le centre de commandement et de contrôle de la DGSN.

Une gamme de caméras de surveillance ultrasophistiquée, un système satellitaire de navigation BDS⁵⁴ intelligent, et des hélicoptères sont déployés dans des régions

⁵⁴ Beidou Navigation Satellite System : un système développé et opéré par la Chine doté de 23 satellites utilisé pour la navigation à l'instar de système GPS des Etats-Unis ou le système GALILEO de l'Union Européenne

publiques telles que les agglomérations urbaines, zones suburbaines, grands axes routiers et tronçons à forte circulation, ports, aéroports et grandes enceintes sportives, grandes entreprises économiques et tout lieu pouvant recevoir beaucoup ou voir transiter beaucoup de monde. Derrière tous ces équipements se trouve le facteur humain composé de personnel formé et qualifié, des ingénieurs et des techniciens en informatique et en électronique doté d'une expérience dans le domaine de télésurveillance assure la maintenance des équipements, la supervision des opérations sur le terrain, la synchronisation entre les différentes hiérarchies de l'équipe d'intervention, et surtout la perspicacité lors de l'analyse des séquences vidéo capturées en temps réel.

3. Les composantes d'un système de vidéosurveillance

Un système de vidéosurveillance est composé principalement d'un ensemble d'éléments figés et indispensables, ensuite ils sont augmentés par des éléments plus au moins optionnels dans le sens non critique, mais ils font la différence entre un système de vidéosurveillance basique ou sophistiqué, performant ou non. Les éléments les plus communs dans un système de vidéosurveillance sont les équipements d'acquisition, la transmission et le traitement des images transmises.

3.1. Les équipements d'acquisition

Il existe plusieurs modèles et plusieurs formes de caméras de surveillance, elles sont structurées en deux parties principales : analogiques et numériques. L'existence de multitude de formes de caméras est due d'un côté au constructeur qui essaie d'utiliser un bon design à la mode, et d'un autre côté, elle est due aux besoins et aux conditions prévues à l'utilisation de ces caméras. Parmi les types de caméras qui existent on cite :

- **Caméra fixe** : la caméra fixe comme son nom l'indique est la caméra non mobile et non motorisée, elle est fixée dans un endroit visible en général devant l'entrée des structures publiques pour donner l'impression que les lieux sont surveillés. C'est le genre classique de caméra, elle n'offre pas des opérations avancées pour le suivi et la détection du mouvement, elle est en général en noir et blanc et de qualité d'image réduite.

- Caméras PTZ⁵⁵ (panoramique/inclinaison/zoom) : ce type de caméra est considéré comme sophistiqué, il offre des options avancées telles que la rotation horizontale et verticale, la vision nocturne basée sur les rayons infrarouges pour permettre la surveillance même dans les endroits sombres. Ces caméras peuvent être motorisées et contrôlées à partir d'autres appareils comme les Smartphones et les ordinateurs. Ainsi, elles peuvent effectuer plusieurs autres opérations utiles dans la vidéosurveillance comme le zoom, le suivi des objets, la détection de mouvement. Et grâce à la technologie de la vidéosurveillance IP, les caméras PZT peuvent être connectées à n'importe quel type de réseau que ce soit Ethernet, Wifi ou Internet ce qui les rend accessible et complètement contrôlable à distance. L'enregistrement des séquences vidéo avec les caméras PTZ peut être activé ou désactivé, planifié à des horaires du jour, ou activé seulement lors de vérification de condition qui est en générale la détection d'un évènement ou d'un mouvement. L'enregistrement conditionnel des vidéos issues de la caméra joue un rôle important pour réduire l'utilisation de la bande passante, économiser des heures d'enregistrement par jour et donc l'utilisation des supports de stockage.
- La caméra dôme : peut remplacer une dizaine de caméras fixes, sa version motorisée est capable de faire une rotation horizontale de 360° et verticale de 180°. Connue par sa robustesse, solidité, et sa forme discrète. Une caméra dôme peut être en même temps une caméra PTZ et IP, elle est utilisée en Algérie surtout par les forces de l'ordre au niveau des plafonds des lieux de détention temporaires.
- La caméra panoramique : ce type de caméra offre une couverture étendue entre 180° et 360° horizontalement en une même image. Plusieurs techniques permettent la capture de photos panoramiques, la première consiste à combiner les images issues de la rotation latérale d'une optique. La deuxième technique est celle adoptée par les caméras numériques, elles utilisent un capteur linéaire qui parcourt successivement les directions incluses dans le champ de vision, et assemble automatiquement les portions successives des images prises pendant le mouvement. L'utilisation d'autres techniques basées sur la lentille telles que l'œil de poisson, la lentille à miroirs, et la lentille panomorphe est aussi envisageable.

⁵⁵ PZT : Pan-Tilt-Zoom

- La caméra thermique : en se basant sur les ondes infrarouges, elle enregistre à l'aide des capteurs les différents rayonnements émis par les corps en fonction de leur température. A chaque degré est affectée une couleur, et l'assemblage de tous les états de chaleur des capteurs permet de donner la vue réfléchi par la température des objets filmés. Ils existent deux types de technologies thermiques : sans refroidissement et avec refroidissement qui est aussi appelée cryogénisée. Cette dernière se refroidit à des températures négatives afin d'améliorer sa précision et sa résolution, elle arrive à détecter une variation de température de 0.1° à une distance moyenne de 300 mètres, ce qui n'est pas possible avec le type de caméra sans refroidissement. Les caméras thermiques sont utilisées en générale dans des lieux sombres et peu dégagé, elles sont très répondues dans plusieurs domaines d'applications de la protection civiles et le domaine militaire.
- La caméra à vision nocturne : aussi appelé amplificateur de la lumière résiduelle (ALR), son principe est de capturer les particules de la lumière (photons) à l'aide d'une plaque capteur appelée photocathode, et les transformer à des particules électriques représentés par des électrons. Ce type de caméra amplifie de près de 50 000 fois la moindre lumière captée de différentes sources telles que la lumière de la lune, des étoiles, et aussi des infrarouges. L'image issue de la plupart des caméras à vision nocturne sont colorées avec la couleur verte, ce qui est expliqué par l'utilisation d'une plaque phosphorique qui émet de la lumière verte au contact des électrons. Récemment, de nouvelles technologies très sensibles à la lumière permettent d'afficher les images captées la nuit avec leurs couleurs originales et avec beaucoup moins de bruit et plus de précision telles que les caméras X27 dotée d'algorithmes spécifiques de Sierra Pacific Innovations, ME20F-SH de Canon, et A7S III de Sony. Les caméras de vision nocturne sont pratiques pour filmer dans l'obscurité dans un rayon pouvant aller jusqu'à 30 mètres pour se protéger contre les voleurs et les cambriolages.
- La caméra anti-vandalisme : à mettre dans les endroits sensibles, résistante en cas de volonté de l'endommager ou de l'arracher. Elle peut offrir les mêmes fonctionnalités que les autres types de caméras.

- La caméra d'espionnage : caractérisée par sa forme camouflée et par sa petite taille, elle peut être portable ou fixée dans des endroits non suspects, elle peut avoir plusieurs formes typiques telles que la forme d'un stylo.
- La caméra embarquée : ce type de caméra est utilisé surtout dans les environnements mobiles tels que dans les moyens de transport. Généralement, ces caméras sont dotées d'une unité de traitement qui les qualifie de réaliser des opérations de comptage, de détection et de suivi.

3.2.La transmission

Pour la plupart des systèmes de vidéosurveillance et de télésurveillance, la transmission des vidéos capturées est une tâche fondamentale pour le reste du système. Elle est utile pour l'affichage sur des moniteurs, l'enregistrement et les traitements possibles. Les techniques de transmission dépendent du type des caméras utilisées, on distingue la transmission filaire et la transmission sans fil.

- La transmission filaire de la vidéo analogique : le câble coaxial KX6 est le plus utilisé, il est composé d'un câble d'alimentation (12v ou 24v) et d'un câble vidéo pour la transmission des images et du son. La distance maximum que peut parcourir le signal vidéo avec ce type de câble est de 250 mètres, une utilisation de répéteurs, boîtiers coaxiaux, des kits complet de transmetteurs et de récepteurs est possible afin d'étendre la distance maximum en se basant sur l'amplification du signal. De même, l'utilisation d'autre type de câble de transmission analogique de la vidéo est possible comme le câble KX8 qui permet une transmission à une distance de 500 mètres, et le câble IDC100 qui permet une transmission du signal à une distance de 1000 mètres.
- La transmission filaire de la vidéo numérique : tous les moyens de transmission de la vidéo analogique sont utilisables pour transmettre la vidéo numérique avec l'ajout d'un module de codage/décodage. Ce module permet de convertir un signal vidéo de la forme analogique vers la forme numérique et vis-vers-ça. L'utilisation de la solution de module de codage/décodage est souvent accompagnée par une réduction de la qualité du signal transmis. Cependant, la transmission de la vidéo numérique est effectuée en utilisant des câbles torsadés blindés afin de protéger le signal vidéo transmis des différentes altérations possibles. Ainsi, l'utilisation de réseau Ethernet

est possible dans le cas de la vidéosurveillance IP, elle offre un débit théorique qui peut aller jusqu'à 1 Gbits/s et pour de très grandes distances en présence de commutateurs et de répéteurs pour régénérer le signal une fois affaibli. D'autre part, pour la transmission des vidéos numériques dans des réseaux dite multimédia, l'utilisation de la CPL⁵⁶ est possible pour transférer le signal vidéo via le courant porteur en utilisant le réseau électrique domestique ou dans une entreprise. Les protocoles de transport quant à eux sont différents de ceux utilisés dans les réseaux de données. On trouve les protocoles basés sur UDP qui fonctionnent en mode non connecté tel que RTP, RTCP, RTSP, RSVP, ou encore des modèles tels qu'IntServ et DiffServ. La caractéristique en commun entre les protocoles multimédia c'est qu'ils favorisent le streaming en temps réel tout en essayant d'offrir la meilleure qualité de service.

- La transmission sans fil du signal vidéo : Que ce soit pour la vidéo numérique ou analogique, la transmission sans fil nécessite un émetteur et un récepteur du signal. Ce type de transmission est utile lorsque l'utilisation des câbles n'est pas possible ou très coûteuse. L'émetteur dans le cas de la transmission de la vidéo analogique utilise l'une des trois fréquences 900 MHz, 2.4 GHz ou 5.8 GHz, la transmission est précédée des opérations de codage, multiplexage et modulation des composantes audio et vidéo. Les trois éléments de base de la couleur, plus l'audio mono ou stéréo sont fusionnés et transmis sur une fréquence. Au côté récepteur, le signal est démultiplexé et décodé afin de retrouver les signaux de base : RGB⁵⁷ plus l'audio. Le codage du signal de la vidéo numérique est différent de celui analogique, l'image est prise pixel par pixel de gauche à droite, de haut en bas, et codée de manière à présenter les éléments de base d'un pixel qui sont la couleur, la luminosité et le contraste, d'autres éléments peuvent être ajoutés tels que la transparence ainsi que des informations globales sur l'image comme sa dimension et sa qualité. La transmission du signal audio/vidéo numérique est la transmission des informations déjà codées en termes de paquets sur des fréquences qui dépendent de l'étendu. L'avantage de la transmission de la vidéo numérique par rapport à celle analogique est de pouvoir envoyer la vidéo en qualité haute définition grâce aux techniques de

⁵⁶ CPL : courants porteurs en ligne

⁵⁷ RGB : Rouge, Vert et Bleu

la compression pour des distances allant jusqu'à 2000 mètres dans un espace dégagé et dans des conditions satisfaisantes.

3.3.Le traitement

Le traitement du flux vidéo capturé par une caméra de surveillance peut être effectué au niveau de l'unité de traitement aussi appelée serveur, et au niveau de quelque type de caméras intelligentes. Ils existent plusieurs modèles de serveur qui peuvent être listés selon l'ordre de leur apparition comme suit : on trouve l'enregistreur de la vidéo numérique (DVR⁵⁸), l'enregistreur numérique réseau (NVR⁵⁹), l'enregistreur de vidéo hybride (HVR⁶⁰), et le super enregistreur de la vidéo numérique (SDVR⁶¹). Tous ces serveurs offrent des fonctionnalités de base qui permettent de recevoir le flux vidéo capturé, l'analyser, le compresser, l'enregistrer, le visualiser et l'acheminer sur un réseau si cette option est disponible. La qualité des fonctionnalités offertes par les serveurs de traitement de la vidéo diffère d'un serveur à un autre et peut être résumé dans les points suivants :

- Le type de caméra avec laquelle le serveur est compatible (analogique, numérique ou les deux ensemble),
- La qualité de compression et d'enregistrement des vidéos,
- Coût et facilité d'installation et de configuration.

En outre des serveurs matériels cités plus haut, une solution logicielle est possible pour permettre d'analyser la vidéo capturée. Ces logiciels sont à installer sur des ordinateurs serveurs et des plateformes performantes afin de bénéficier de leur puissance de traitement et de stockage. Les performances élevées des ordinateurs serveurs sont indispensables pour l'analyse des vidéos reçues en permanence et dont le résultat peut être crucial et donc nécessite un traitement en temps réel.

Que ce soit la solution matérielle ou logicielle, les traitements de base doivent être offerts et augmentés en cas de besoins. Les opérations les plus courantes incluses par l'analyse de la vidéo de surveillance sont les suivantes :

⁵⁸ DVR : Digital Video Recorder, utilisé lorsque le flux de la vidéo en entré est analogique.

⁵⁹ NVR : Network Video Recorder, utilisé lorsque la vidéo en entré est en HD ce qui est le cas des caméras IP.

⁶⁰ HVR : Hybrid Video Recorder, utilisé pour les deux types de vidéos : numériques et analogiques.

⁶¹ SDVR : Super Digital Video Recorder, une combinaison de DVR, HVR, et NVR.

- La détection : cette opération offre la possibilité de détecter un mouvement, un incendie et d'autres événements. Dans les caméras intelligentes, PTZ et dôme, cette fonction est traitée par la caméra mêmes, et la détection entraîne le déclenchement d'un événement qui sera traité par l'unité de traitement.
- La reconnaissance : la reconnaissance des objets, des formes et des personnes représente une opération cruciale et une grande partie des besoins de la vidéosurveillance. La reconnaissance automatique dans le domaine de la vidéosurveillance ouvre la porte devant une multitude d'axe de recherche en combinant les algorithmes de l'intelligence artificielle avec la technologie des données énormes pour le traitement de la grande masse de vidéos issues des systèmes de vidéosurveillance.
- Le suivi du mouvement : cette opération est le complément de la détection du mouvement et la reconnaissance des formes. Le suivi de l'objet en mouvement peut être réalisé à base de plusieurs techniques, la vitesse et la direction du mouvement peuvent être calculées en suivant les points d'intérêts dans la séquence des images constituant la vidéo. Dans le cas de suivi d'un objet à travers plusieurs caméras de surveillance, cela nécessite une reconnaissance de l'objet suivi en utilisant une des techniques disponibles qui s'appuient sur des modèles probabilistes, ou des techniques de l'intelligence artificielle.

4. Architecture des systèmes de vidéosurveillance

Les systèmes de vidéosurveillance composés de plusieurs caméras, peuvent être classés selon leurs architectures en deux catégories : les systèmes centralisés et les systèmes distribués.

4.1. Architecture centralisée

Dans le premier type d'architecture, toutes les caméras déployées sur un espace, une entreprise, ou une cité, envoient les séquences de vidéos capturées à une unité de traitement pour les analyser, les enregistrer et les afficher. La liaison des caméras avec le serveur est réalisée par un réseau IP si la vidéosurveillance IP est supportée. Autrement la connexion des caméras doit être directe avec l'unité de traitement.

Pour des opérations de traitement avancées tels que la détection, la reconnaissance et le suivi des objets, les enregistreurs ne peuvent pas traiter les vidéos issues d'un nombre

élevé de caméras, une utilisation d'un serveur sophistiqué est essentielle pour offrir les ressources nécessaires aux opérations gourmandes de traitement de vidéo.

D'un autre côté, dans le cas d'utilisation d'un réseau pour la transmission des vidéos, l'utilisation de la bande passante doit être gérée et partagée entre toutes les caméras déployées. Aussi le choix de la qualité des vidéos capturées et le choix des équipements réseau doit être étudié afin de maintenir un rapport coût/QoS raisonnable.

4.2. Architecture distribuée

Dans un système de vidéosurveillance distribuée, le stockage des données multimédia et les traitements peuvent être partagés entre plusieurs éléments. Ainsi, Les dispositifs de capture peuvent fournir des ressources de calcul supplémentaires pour effectuer une partie des traitements. Les avantages de la vidéosurveillance distribuée sont résumés dans ce qui suit :

- Possibilité d'enregistrer des caméras distantes en temps réel, sans délai ;
- Réduction massive de l'utilisation du réseau car seuls les flux vidéo en cours de visualisation sont extraits du serveur distant ;
- Si un serveur du réseau tombe en panne, tous les autres continuent à enregistrer ;
- Augmentation massive des performances : aucun réseau n'est saturé. Les ressources sont partagées entre des réseaux physiques distincts et des périphériques matériels ;
- Permet la surveillance des caméras géographiquement diverses à partir d'un seul emplacement.

4.2.1. Utilisation des caméras intelligentes distribuées

Les caméras intelligentes distribuées (DSC⁶²) sont des dispositifs autonomes qualifiées pour effectuer quelques tâches précises sans le passage par les serveurs, leurs tâches est de produire des données, reconnaître et rapporter des objets et activités d'intérêt, plutôt que des capturer des images (Rinner et Wolf, 2008). Les DSCs sont des systèmes intégrés distribués qui effectuent une vision par ordinateur en temps réel à l'aide de plusieurs caméras. Elles sont le résultat de convergence de l'évolution de la vision par ordinateur, la technologie de l'intégration à très grand échelle (VLSI⁶³), et l'informatique embarquée.

⁶² DCC: Distributed smart cameras

⁶³ VLSI: Very Large-Scale Integration

Cependant, elles améliorent de façon remarquable les performances globales du système de vidéosurveillance distribué sur lequel elles sont déployées comme cité ci-dessous :

- Rationnaliser l'utilisation de la bande passante : le traitement de la vidéo capturée par les dispositifs de capture permet l'analyse des scènes et la transmission de celles qui représentent un évènement à traiter par le serveur. Ce qui permet d'utiliser la bande passante seulement pour transmettre les scènes utiles et donc libérer la bande passante pour les autres dispositifs de capture du même système.
En outre, la bande passante nécessaire pour la transmission d'une scène est réduite grâce à l'utilisation des algorithmes de compression tels que la transformé cosinus discrète DCT⁶⁴ ou la modulation par impulsion et codage DPCM⁶⁵ qui utilise la prévision compensée du mouvement entre trames.
- Prendre des décisions en temps réel : la réaction à des évènements peut être effectuée en temps réel sans faire recours aux serveurs. Une caméra de surveillance intelligente peut exécuter des actions en présence de conditions de déclenchement. Dotées de programmes intelligents, elles peuvent détecter une tentative d'effraction, un incendie, comptage de personnes, de détection des stationnements anormaux, etc. Ce qui leurs permet de prendre des actions en temps réel face aux événements déclenchant telles que le lancement d'une alarme ou d'une notification aux forces de droit.
- Rationnaliser l'utilisation des ressources de stockage : en détectant les événements d'importance, les caméras intelligentes transmettent le flux à stocker seulement aux besoins, ce qui réduit la quantité des vidéos enregistrées.
- Réduire la charge des traitements sur les serveurs : les traitements effectués au niveau des caméras intelligentes jouent un rôle important dans l'équilibrage de charge du système de vidéosurveillance distribué. En assurant une partie des traitements par les dispositifs de capture, les serveurs de traitements peuvent bien réaliser leurs missions avec des performances appropriées.

Une caméra intelligente combine la capture de la vidéo, le traitement et la communication en une seule plateforme embarquée comme montré par la figure 1.

⁶⁴ DCT: Discrete Cosine Transform

⁶⁵ DPCM: Differential Pulse Code Modulation

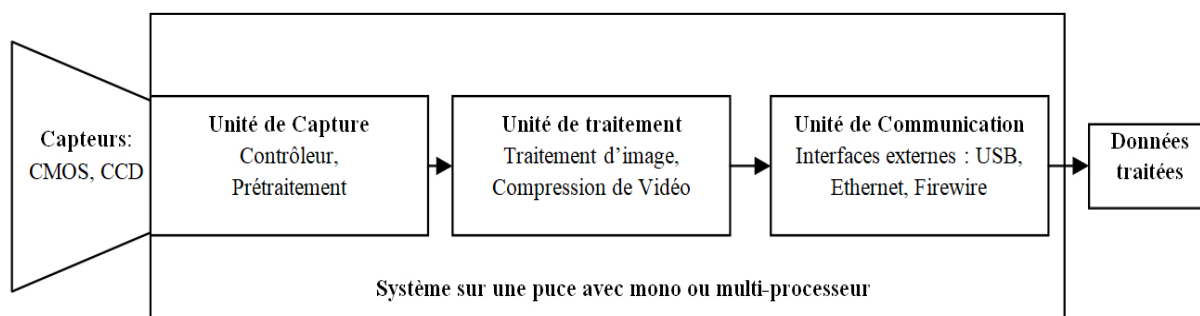


Figure 13. Architecture générique d'une caméra intelligente.

Les capteurs CCD⁶⁶, CMOS⁶⁷, superCCD, etc. sont des sources de données pour les caméras, ils sont des capteurs d'images qui assurent la transformation des signaux lumineux en signaux électriques. Ces derniers sont traités par l'unité de capture qui est responsable d'effectuer des opérations de prétraitement représentées principalement de l'élimination du bruit, la transformation couleurs et niveau de gris, et la régulation de la luminosité, de même l'unité de capture est responsable du contrôle des capteurs en ce qui concerne la sensibilité à la lumière, le focus, le zoom, le taux de capture frame/seconde... Une fois l'image est capturée, elle est transmise à l'unité de traitement qui manipule les données reçues en temps réel. Elle est composée d'une mémoire ROM qui contient les programmes implémentant les algorithmes de traitement des images et des vidéos, la mémoire de traitement charge à la demande les données reçues pour les traiter par l'unité de calcul. Cette dernière est composée d'un mono ou multiprocesseur avec des performances qui varient selon plusieurs critères tels que la consommation de l'énergie, le modèle de la caméra et la mission affectée à cette caméra. Une unité de traitement réalise une variété d'algorithmes de traitement d'image comme la détection du mouvement, le suivi, la segmentation, la reconnaissance d'objet, etc. et produisent des caractéristiques géométriques et de couleur, objets segmentés, ou des décisions de haut niveau tel que la conduite à l'inverse d'une direction, un objet suspect, ou un incendie détecté. Ces décisions sont transmises à l'utilisateur final par un ajout d'information aux flux vidéo transmis en affichant un message d'avertissement ou en encerclant l'objet d'intérêt avec une couleur distinguée. Les unités de traitement les plus populaires et qui offre un bon compromis entre performance, consommation énergétique, et flexibilité sont les réseaux de portes

⁶⁶ CCD: Charged Coupled Device.

⁶⁷ CMOS: Complementary Metal Oxide Semiconductor

programmables (FPGAs⁶⁸), les processeurs de signal numérique (DSPs⁶⁹) et/ou les microprocesseurs. A la fin des traitements, les données résultantes sont transmises à l'unité de communication qui contient des interfaces externes permettant la transmission des données aux destinations supportant une variété de standards tels qu'Ethernet, USB, Firewire, Wifi, etc.

Afin d'assurer de bonnes performances et le passage à l'échelle des systèmes de caméra de surveillance distribués, les traitements sont effectués en impliquant plusieurs nœuds (processeurs) en constituant un réseau ad hoc où chaque nœud ne connaît pas à priori l'état des autres nœuds. Par conséquent un échange d'information est effectué par messages afin de détecter l'identité et l'état de charge des voisins d'un nœud donné. Comme la transmission des données entre les nœuds est très coûteuse en terme consommation énergétique, le nœud qui envoie les données doit bien cibler les nœuds impliqués dans les traitements en communs à effectuer. Les nœuds qui peuvent être ciblés sont des nœuds qui partagent une connaissance gestuelle, ou des nœuds qui offrent une puissance de traitement nécessaire à l'accomplissement d'une tâche. Le premier cas met l'accent sur la connaissance gestuelle distribuée qui est représentée par la présence d'une connaissance dans plusieurs nœuds qui possèdent une partie d'une image, des angles de vue différents pour un objet d'intérêt, une partie d'une scène, etc.

4.2.2. Utilisation des serveurs enregistreurs

Un système de vidéosurveillance distribué basé sur les enregistreurs intelligents est composé principalement d'un réseau, un nombre de caméras de nature hétérogène que ce soit analogique ou numérique, et un ensemble d'enregistreurs qui peuvent être à leurs tours hétérogènes tout en dépendant des caméras utilisées. La figure 2 montre un cas de déploiement possible d'un système de caméras de surveillance distribué basé sur des enregistreurs. Un réseau IP est nécessaire pour accéder aux enregistreurs liés aux caméras d'un côté et au réseau Ethernet de l'autre côté. Les agents au centre de télésurveillance peuvent accéder aux enregistreurs via le réseau utilisé et visualiser les vidéos en direct ou celles déjà sauvegardées. L'inconvénient majeur de cette architecture est limitation du

⁶⁸ FPGA: Field-programmable gate arrays

⁶⁹ DSP : digital signal processors

nombre des caméras à quelques dizaines et qui est causé par la limitation de la bande passante nécessaire pour la transmission de toutes les vidéos via le réseau.

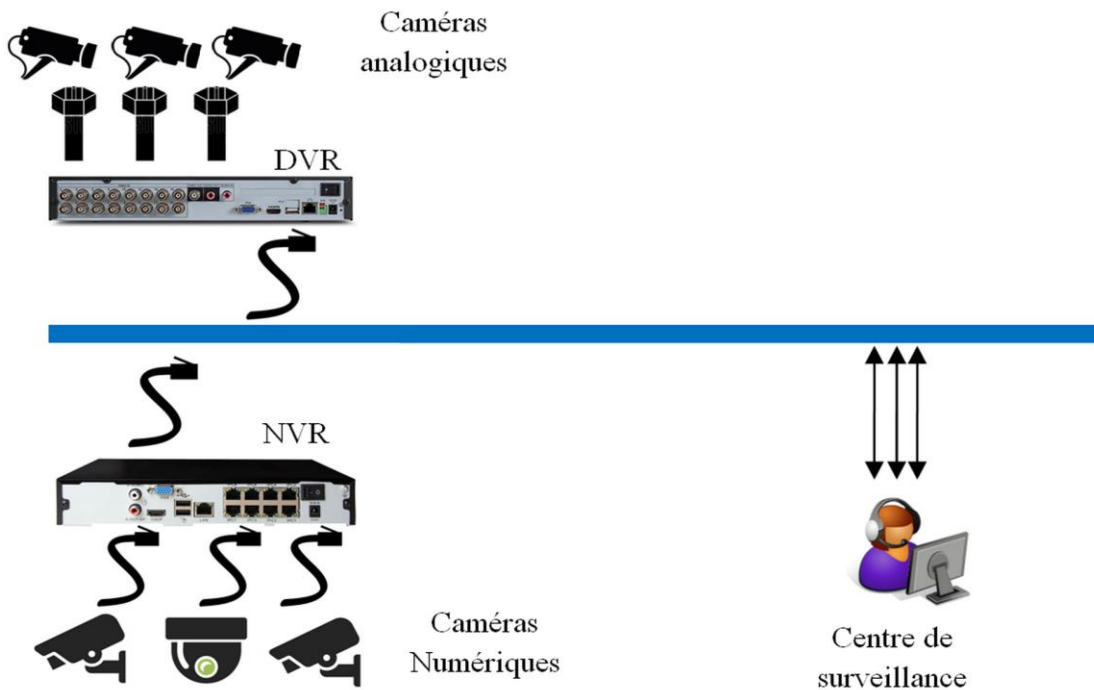


Figure 14: Un système de vidéosurveillance distribué basé sur les enregistreurs.

5. Serveurs de stockage des systèmes de vidéosurveillance

L'utilisation des ordinateurs serveurs puissants pour la vidéosurveillance distribuée peut être impliquée au niveau stockage comme au niveau traitement de données.

5.1. Le stockage distribué des données de la vidéosurveillance

Des technologies distinctes offrent la possibilité d'effectuer un stockage avec des performances élevées qui peuvent être utilisées dans des différents domaines notamment celui de stockage des données des systèmes de vidéosurveillance. Les solutions existantes sont évaluées en fonction des critères ci-dessous ("NAS, DAS and SAN: What's the Best Archive for IP Surveillance?", 2018) :

- **Évolutivité** : Possibilité d'ajouter rapidement et facilement de la capacité de stockage au fur et à mesure de la croissance des besoins en nombre de caméras de surveillance et aussi en technologie. A titre d'exemple, on cite le cas de besoins d'ajouter une nouvelle baie de stockage de technologie différente à celle du système qui existe déjà.

- **Facilité d'utilisation** : Possibilité de stocker, d'indexer et de récupérer des vidéos sans avoir besoin d'une expertise technique spécialisée pour faire fonctionner le système.
- **Abordabilité** : Elle est présentée par la relation entre le coût d'une solution et le pouvoir de paiement de l'entreprise qui a sollicité une solution sur un axe de temps.

Parmi les technologies de stockage qui existent, on cite :

5.1.1. SAN (Storage Area Network)

C'est un réseau dédié au stockage de données permettant de mutualiser des ressources de stockage. Les serveurs renvoient les données à stocker sur le réseau SAN (Tate et al., 2016) qui les achemine jusqu'aux baies de stockage. Les données à stocker sont routées via des Switch spécialisés Cisco reliés entre eux et qui forment une structure appelée 'Fabrique'. Un réseau de stockage comporte au moins deux fabriques indépendantes. L'adoption de plusieurs fabriques pour le même réseau a pour objectif d'avoir une haute disponibilité des données stockées, ainsi que de permettre une tolérance aux pannes et des interventions de maintenance matérielles transparentes.

L'échange des données à travers un réseau de stockage est basé sur les standards SCSI (Small Computer System Interface) pour la connexion et le transfert des données entre les serveurs et les périphériques de stockage. On distingue deux catégories de réseau de stockage selon la nature des fabriques utilisées. La première catégorie est FCSAN (Fibre Channel Storage Area Network). Un exemple de sa structure est montré par la figure 3. Elle utilise le protocole FCP (Fibre Channel Protocole) afin de permettre la transmission haut débit des données à travers un réseau de routeurs et Switchs interconnectés par des fibres optiques. Tandis que pour la seconde catégorie, iSCSI SAN, elle utilise le protocole iSCSI (internet Small Computer System Interface) qui permet la transmission haute débit des données en se basant sur le protocole IP. Cela lui permet d'utiliser les réseaux Ethernet existants. Les deux catégories présentent des avantages l'un par rapport à l'autre. L'utilisation de FCP permet d'offrir une bande passante qui peut dépasser les dix Giga/Sec grâce aux fibres optiques qui constituent le moyen de liaison des réseaux de la fabrique. Quant à l'utilisation du protocole iSCSI offre l'avantage en fonction du coût car il exploite les réseaux Ethernet déjà mis en place.

Les baies de stockages sont utilisées pour conserver les bits de données. Cependant, le système de fichiers est localisé au niveau du serveur qui effectue des opérations d'Entrée/Sortie en termes de blocs de données et non pas de fichier. Une des fonctions des SANs est la création de LUN (Logical Unit Number). Il permet de découper les espaces de stockage en plusieurs unités logiques afin de gérer plus efficacement l'espace de stockage de la baie. Un LUN peut être répliqué en mode synchrone ou asynchrone vers une autre baie afin d'assurer la disponibilité des données de LUN et de permettre une tolérance aux pannes.

Néanmoins, la technologie basée sur les SANs est très couteuse en terme d'acquisition du système comparée avec d'autres solutions d'un côté. De l'autre côté, elle n'est pas facile à gérer et ne favorise pas l'évolutivité.

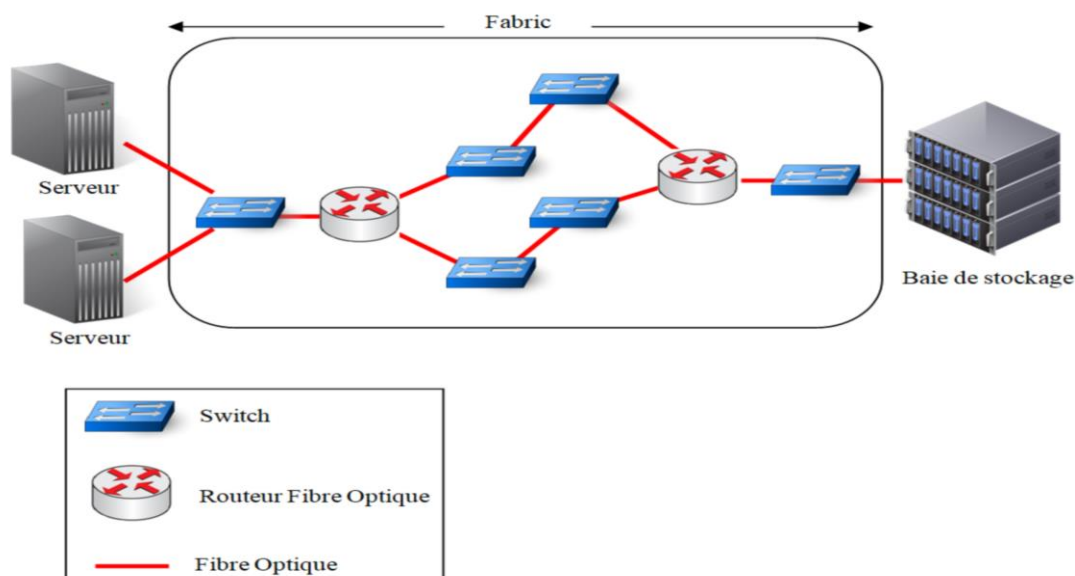


Figure 15: Un exemple d'un FCSAN.

5.1.2. NAS (Network-Attached Storage)

Appelé aussi stockage en réseau. Contrairement à SAN, NAS est un serveur de fichier autonome qui contient un système d'exploitation dédié à la gestion des données tels que FreeNAS⁷⁰, HNAS⁷¹, NASLite⁷², etc. NAS est doté d'un logiciel de configuration, de son propre système de fichier, ainsi qu'un ensemble de disques indépendants. Le NAS est

⁷⁰ FreeNas : Un serveur de fichiers NAS dérivé du système d'exploitation FreeBSD.

⁷¹ HNAS : Une plate-forme NAS d'Hitachi Data Systems.

⁷² NASLite : Un serveur de fichiers NAS basé sur une mini-distribution d'un système Linux.

attaché à un réseau local afin de servir des serveurs de fichiers. Ces serveurs sont considérés comme des clients hétérogènes du système NAS dans le réseau, et qui peuvent accéder au même fichier simultanément. Cette caractéristique rend NAS très adapté aux applications qui sollicitent le système de fichier de façon intense tels que les opérations d'archivage, les transactions e-commerce, les applications Big Data et les applications multimédia. L'échange de données ne se fait pas par blocs comme dans le cas des SANs, mais en fichier. La transmission des données se fait via l'utilisation des protocoles qui dépendent des systèmes d'exploitation des serveurs de fichiers utilisés tels que CIFS⁷³ et SMB⁷⁴ pour Microsoft Windows, NFS⁷⁵ pour Linux et Unix, et AFP⁷⁶ pour Apple.

Les systèmes NAS offrent la combinaison idéale de l'évolutivité, facilitée d'utilisation et abordabilité. Grâce à l'ensemble des standards sur les quels est basée, l'évolutivité ne pose plus aucun problème en terme d'espace disque. La capacité de stockage peut passer de quelques Téra à des Péta Octet grâce à la possibilité de remplacer et d'ajouter des disques durs à chaud. De plus, la caractéristique d'évolutivité des réseaux d'accès au NAS est assurée. Les réseaux d'accès sont facilement déployables et offrent la possibilité de personnaliser leur structure et de les améliorer. On peut par exemple ajouter de nouveaux équipements ou des liaisons plus sophistiquées tout en gardant un rapport entre coût et performance. D'un autre côté, la centralisation des données permet de rendre le système plus facile à utiliser et accessible par plusieurs utilisateurs de plusieurs endroits en même temps. La gestion d'un système NAS se fait par des utilisateurs privilégiés sur des systèmes d'exploitation connus.

Les systèmes NAS offrent une haute sécurité des données en adoptant la technologie RAID⁷⁷. Cette dernière existe en plusieurs variantes et permet l'amélioration des performances globales du système. Elle permet une réplication des données sur d'autres disques durs et baies afin d'augmenter leurs disponibilités, favoriser le partage de charge, restaurer les données en cas de perte et améliorer la tolérance aux pannes.

⁷³ CIFS : Common Internet File System, une implémentation particulière du protocole SMB, créé par Microsoft.

⁷⁴ SMB : Server Message Block, un protocole de partage de fichiers qui a été inventé par IBM, il permet aux ordinateurs de lire et d'écrire des fichiers sur un hôte distant via un réseau local (LAN).

⁷⁵ NFS : Networked File System, développé par Sun Microsystems et sert le même but que SMB, utilisé pour Unix et Linux.

⁷⁶ AFP : Apple Filing Protocol, un protocole de partage de fichier utilisé pour Mac sous OS X.

⁷⁷ RAID: Redundant Array of Independent/Inexpensive Disks.

5.1.3. DAS (Direct-Attached Storage)

Le terme DAS a été créé après l'apparition des systèmes NAS et SAN afin de différencier entre les systèmes de stockage basés ou non sur les réseaux. Un DAS est un système de stockage directement connecté à un dispositif tel qu'un serveur, une caméra, un téléphone portable ou juste un simple ordinateur. Les supports de stockage utilisés peuvent être de différents types tels qu'un disque dur, une carte mémoire, CD/DVD ou un flash disque. L'accès à ces supports de stockage se fait en mode bloc par un seul serveur à la fois.

Dans le domaine de la vidéosurveillance, les DASs sont présentés par les DVRs. Ils sont liés directement aux caméras de surveillance sans passer par un réseau comme dans le cas des NVRs, et aussi sans avoir besoin d'un support de stockage intégré à ces caméras. Un DVR peut être lié directement à plusieurs caméras pour former une VSM (video management system). Il fournit un système complet de gestion de vidéosurveillance en incluant le stockage des données.

Un système DAS présente l'avantage d'être abordable en terme de coût. Mais présente aussi l'inconvénient qu'il n'est pas évolutif et offre un nombre limité de slots pour ajouter des supports de stockage en plus. D'un autre côté, la mise en place d'un système de vidéosurveillance basé sur les systèmes DAS est difficile vu la nécessité de connecter directement les caméras de surveillance à l'enregistreur. Cette méthode pose des complications pour le branchement, une chose qui est facilement évitable en utilisant le stockage en réseau qui nécessite seulement d'atteindre un des points d'entrée à ce dernier.

5.1.4. Le stockage dans le nuage

Le service de stockage dans le nuage offre la possibilité de stocker à distance et de n'importe quel endroit la vidéo capturée en utilisant une connexion à Internet. Le nombre de caméras qui peuvent être impliquées dans un système de vidéosurveillance basé sur le stockage sur le nuage dépend de la bande passante. La performance du réseau définit la quantité et la qualité des flux vidéo à échanger. Cependant, le nuage est une virtualisation des ressources de différents types y compris celles de stockage. L'architecture d'un nuage de stockage est composée de deux parties principales comme définies par Kulkarni et al. (2012) : l'avant plan (front-end), et l'arrière-plan (back-end). L'avant plan est la partie visible par les utilisateurs. Elle offre des APIs pour l'accès aux différents services offerts

par le nuage, notamment le service de stockage des données appelé StaaS (Storage as a Service). Il est basé sur l'utilisation de protocoles compatibles avec la technologie des services Web. Au-delà de la partie avant plan, on trouve la partie métier qui représente l'arrière-plan du système. Elle est utilisée précisément par les fournisseurs de services. Il est composé de différents serveurs, ordinateurs, systèmes de stockage de données, des conteneurs et des machines virtuelles.

Les systèmes de stockage dans les nuages sont de différents types. On distingue les systèmes de stockage objet, les systèmes de stockage de base de données relationnelle (RDS⁷⁸), et les systèmes de stockage distribués basés sur les fichiers. Ce dernier type est le plus adopté pour le stockage des vidéos. Il offre l'accès aux données partagées simultanément à partir de plusieurs hosts via le réseau Internet. L'infrastructure physique de stockage combine différentes technologies telles que les SANs, les NASs, et les DASs.

Le service de stockage offre des manières différentes pour l'accès aux données stockées dans le nuage comme l'accès à base de bloc de données pour les SANs et les DAS, l'accès basé sur les fichiers pour les NASs, l'accès à travers des services Web, et l'accès basée WebDAV⁷⁹ (Goland et al., 1999). Les accès basés sur les fichiers et les blocs de données offrent plus de performance, disponibilité et sécurité de données. L'accès via des APIs des services Web est communément utilisé car il supporte l'architecture des applications Web. Les services concernés sont basés sur des protocoles et des standards tels que SOAP⁸⁰ et REST⁸¹ qui lui permettent d'accéder aux ressources de stockage sur le cloud. Beaucoup de fournisseurs de service de stockage utilisent les APIs REST tels qu'Amazon S3⁸² de Amazon Web Service, Windows Azure de Microsoft, Mezeo pour offrir une plateforme de stockage en nuage. Cependant, l'accès basé sur WebDAV est le plus approprié pour le stockage de données en nuage le fait qu'il rend beaucoup plus rapide la mise en nuage des données (upload). De plus, il rend possible l'édition simultanée de ces données directement sur le serveur.

⁷⁸ RDS : Relational Database Storage Systems.

⁷⁹ WebDAV : Web-based Distributed Authoring and Versioning, une extension du protocole HTTP qui permet l'édition des données d'une manière distribuée.

⁸⁰ SOAP : Simple Object Access Protocol, un protocole de transmission de message basé sur XML.

⁸¹ REST : Representational State Transfer, permet d'ajouter la notion d'état à un service Web.

⁸² Amazon S3 : Amazon Simple Storage Service

La vidéosurveillance basée sur le stockage en nuage est estimée être la meilleure solution en tenant compte de l'évolutivité, du coût et de la facilité d'utilisation. Beaucoup de prestataires de services de vidéosurveillance sur les cloud offrent des applications spécifiques et efficaces. Ces applications sont conçues avec des interfaces ergonomiques et des fonctionnalités qui facilitent leur utilisation et la gestion d'un nombre important et évolutif de caméras de surveillance. Ces services sont offerts avec des prix compétitifs et qui ne dépassent en aucun cas ceux proposés par d'autres applications issues d'autres technologies. Le stockage en nuage bénéficie des avantages des NASs, SANs, et DASs, et donc offre la meilleure combinaison de solutions pour ces utilisateurs.

6. Le traitement distribué des données de vidéosurveillance

Le traitement des données vidéo est une opération gourmande en termes de ressources de calcul et de mémoire vive. Ces conditions réduisent les performances globales des systèmes de vidéosurveillance face à des caractéristiques limitées de l'environnement et une croissance importante en nombre et en qualité des vidéos. Les systèmes de traitement distribués permettent de couvrir ces besoins en offrant un système évolutif avec une distribution de charge de calcul. De plus, ils offrent la possibilité d'exécuter des opérations sur les vidéos en temps réel ce qui est très utile pour beaucoup d'applications de vidéosurveillance. Parmi les architectures des systèmes de vidéosurveillance distribués qui existent, on cite :

6.1. Les systèmes de vidéosurveillance traditionnels :

La plupart des anciens systèmes de vidéosurveillance distribués sont basés sur l'utilisation des NVRs ou la liaison à un ordinateur générique. Ces unités de traitement offrent des logiciels d'analyse du contenu vidéo appelés (VCA⁸³). Les fonctionnalités VCA sont des offres de fonctionnalités avancées et sont fournies généralement séparément avec un prix élevé. Cela affecte de façon importante le coût global de la mise en place de la solution envisagée. De plus, cette architecture de vidéo surveillance manque d'évolutivité car il n'est pas permis de doubler les coûts du système à chaque fois qu'on doit intégrer un nombre supplémentaire et limité de caméras, à voir des unités de traitement. Cependant, il existe une variété de logiciels qui fournissent une VCA gratuite et en source libre pour une

⁸³ VCA : Video Content Analysis.

configuration en mono-poste tels que ZoneMinder (Coombes, 2007) et OpenVSS (Suvonvorn, 2008).

Un autre critère important pour l'évaluation de l'architecture des systèmes de vidéosurveillance est l'aspect maintenance et entretien du système en cas de panne ou en cas d'évolution. La solution basée sur les systèmes classiques nécessite la présence d'un ou de plusieurs agents qualifiés pour la maintenance et la mise à jour de la configuration des systèmes après une modification de leur structure. Cela entraîne des frais supplémentaires, des complications techniques et une perte d'un temps critique qui peut être fatal pour tout l'organisme.

D'autre part, il existe une architecture des systèmes de vidéosurveillance distribuée composée de plusieurs composants et de plusieurs couches de traitement afin de partager les tâches de la VCA. Par exemple, l'architecture proposée dans Yuan et al. (2003) est constituée de plusieurs caméras liées à des ordinateurs appelés terminaux dont la tâche est d'effectuer des opérations de calcul légères. Un terminal est connecté via un réseau local à une ou plusieurs caméras, qui selon leurs types, peuvent effectuer ou non des opérations de base telles que le prétraitement, la détection et le suivi des objets. Tous les terminaux sont liés à leur tour à un serveur unique qui correspond au centre de contrôle dans le cas général.

Une autre architecture distribuée a été proposée dans (San Miguel et al., 2008) composée d'une partie physique et d'une autre logique. La première est vue comme un ensemble d'équipements connectés à un réseau Ethernet (100Mb/s). Ces équipements sont des caméras de différents types, des ordinateurs, des serveurs de stockage, des applications, des Switchs et de périphériques d'affichage. De l'autre côté, la partie logique sert à déterminer les opérations de chaque type d'équipement. On distingue les couches suivantes :

- La couche d'acquisition : elle est composée de l'ensemble des caméras de surveillance. Elle support l'ajout de nouvelles caméras avec un minimum d'effort de configuration.
- La couche de communication : la communication entre les éléments du système est basée sur le modèle client/serveur point à point assurée par cette couche.

- La couche des modules de traitement : les modules s'exécutent séparément sur des unités de calculs différentes. Une demande de réalisation d'une opération VCA lance un ou plusieurs modules chacun est responsable d'une tâche bien précise et l'ensemble a pour objectif d'accomplir l'opération demandée. La couche de traitement s'interface avec la couche d'acquisition pour lire les données vidéos à traiter, et la couche de gestion de données qui sera expliquée un peu en bas pour le stockage des résultats de traitement.
- La couche de gestion de données : les données sont catégorisées en deux types, le premier est celui concernant les résultats d'analyse, et le second est celui des informations contextuelles. Le stockage de ces données se fait à l'aide de bases de données situées dans un ou plusieurs serveurs de stockage.

Des travaux similaires se basent sur des architectures modulaires ont été proposées dans Wang et al. (2012) et Han et al. (2014) assurent la distribution des traitements des systèmes de vidéosurveillance.

6.2.La vidéosurveillance comme service (VSaaS) :

La vidéosurveillance en tant que service est un système de sécurité sans fil hébergé sur le Web. Elle permet aux utilisateurs de stocker, gérer, enregistrer et lire à distance des vidéos de surveillance entièrement dans le cloud. La séquence vidéo n'est pas stockée sur place, les utilisateurs n'ont pas besoin de logiciel pour le faire, seulement un système de caméra IP, une connexion Internet et un fournisseur VSaaS sont nécessaires (Neal et Rahman, 2012). Elle offre aux utilisateurs un niveau de contrôle et de sécurité sans précédent grâce au cloud. Elle offre également une maintenance simple et minimale et une approche de sécurité abordable et sûre pour les intégrateurs de systèmes qui recherchent des alternatives fiables et évolutives. De plus, la capacité de stockage attribuée aux VSaaS répond bien aux besoins en offrant la taille de stockage demandée par les utilisateurs avec la prise en compte de possibles futur évolutions.

Les différences entre la VSaaS et les systèmes de vidéosurveillance traditionnels peuvent être extraits de la différence entre un système de cloud et un autre classique. Ces différences ont été résumées dans l'étude menée par Zhou (2016). Ces différences sont

inspirées des cinq caractéristiques essentielles de l'informatique en nuage définies par l'Institut national des normes et de la technologie (NIST⁸⁴) :

- Self-service à la demande : c'est un service de nuage qui met en disposition des ressources dont un client ou une application a besoin à la demande. Dans le cas des VSaaS, cette caractéristique est utile en cas de besoins supplémentaires en ressources de stockage ou de calcul. Les applications forment automatiquement les besoins en ressources et les fournisseurs de services leur répondent aussi de manière automatique sans aucune intervention humaine.
- Large accès au réseau : cela concerne les données offertes de plusieurs endroits en ligne, hébergées dans un nuage et qui sont accessibles par une variété de dispositifs. Cette caractéristique permet l'accès aux systèmes VSaaS de n'importe quel point relié à Internet. L'accès à ces systèmes est utile pour la transmission des données vidéos du et vers le VSaaS des caméras installées n'importe où dans le monde, et des serveurs de traitement de données aussi géographiquement distants.
- La mise en commun des ressources : c'est l'une des opérations fondamentales des nuages. Elle consiste à regrouper un ensemble de ressources afin de fournir un service fiable de calcul, de stockage, ou de réseau. Ce groupement permet aux fournisseurs de VSaaS de donner l'impression que les ressources sont infinies et immédiatement disponibles.
- Elasticité rapide : c'est le pouvoir d'adaptation aux besoins de la VSaaS. Ces besoins sont en termes de ressource de calcul, de stockage et de bande passante. L'élasticité dans ce contexte est présentée par deux manières : horizontale et verticale. La première est représentée par l'augmentation de nombre de serveurs ou de machines virtuelles impliqués pour accomplir une tâche précise. La seconde manière d'élasticité est l'augmentation des ressources propres aux machines virtuelles telles que les CPU, RAM, HDD, etc.
- Service mesuré : Dans un service mesuré, les caractéristiques du service sont contrôlées et surveillées par le fournisseur. Ceci est très utile à la tâche de facturation, le contrôle d'accès, l'optimisation des ressources et la planification de la capacité des ressources à offrir.

⁸⁴ NIST : est une agence du département du Commerce des États-Unis. Sa mission est de promouvoir l'innovation et la compétitivité industrielle.

Le traitement des données multimédia dans les VSaaS diffère selon le déploiement et le type des dispositifs de capture utilisé, on distingue les unités de traitement suivantes : les dispositifs de capture, le bord finaux de nuage et le nuage. Dans le cas où il y a une nécessité d'analyser en temps réel le flux capturé, les caméras intelligentes sont les plus appropriées car elles sont chargées d'exécuter les opérations dont elles sont responsables telles que la détection des objets, du mouvement, le suivi, etc. Dans ce cas, la caméra intelligente offre une grande partie de ressources de calcul.

Néanmoins, la plupart des caméras intelligentes sont sous la forme de capteur mobile qui nécessite une énergie pour ses opérations de calcul et de transmission du flux capturé. Afin de réduire la consommation de l'énergie, les caméras intelligentes sont liées à des bords finaux de nuage. Le cas d'utilisation de brouillard (fogging) est envisageable. Ces bords terminaux de nuage reçoivent le flux envoyé par les caméras de surveillance, que ce soit intelligente ou non. Ils effectuent les opérations dont ils sont responsables, ils envoient le flux vidéo modifié et les résultats de son traitement aux serveurs en nuage pour être analysés, stockés et archivés.

L'utilisation des cloud pour le traitement de la vidéo capturée est aussi une solution envisageable mais elle ne répond pas aux besoins d'analyse en temps réel. La transmission du flux vidéo de l'ensemble des dispositifs de capture vers le nuage nécessite une connexion à internet fiable, sûre et à grand débit. Ces exigences ne sont pas toujours vérifiées permanentement. Une fois le flux arrivé au nuage, il sera traité par les applications dédiées aux traitements et la gestion de vidéos. Dans ce contexte, on peut faire référence à Cloudatastructure⁸⁵ ("Cloudatastructure Inc - Building Automation Solutions", 2018) qui propose un DVR pour récupérer le flux capturé de plusieurs caméras, le transmettre via le réseau Internet, l'analyser et l'enregistre au niveau du nuage.

⁸⁵ Cloudatastructure : est l'un des principaux fournisseurs de solutions intégrées d'automatisation du bâtiment pour la vidéosurveillance, les contrôles d'accès aux portes et l'intégration IoT.

7. Conclusion

La vidéosurveillance joue un rôle crucial dans la sécurité des individus, des propriétés et des nations. Sa propagation dans tous les pays est en la preuve. Maitriser cette technologie est un facteur clé pour le développement, cela nécessite la mise à jour des équipements matériels, de même pour la partie logicielle, et de la qualification des ressources humaines. Cependant les systèmes de vidéosurveillance sont multiples, riches de fonctionnalité, et de fiabilité variable. La mise en place d'un système de sécurité nécessite une étude de l'environnement à surveiller, du type d'action à prévenir et des objets à détecter.

D'un autre côté, la technologie de la vidéosurveillance ne cesse d'évoluer. En passant par les points cités dans ce chapitre, on trouve que la variété de ces technologies est un avantage en même. Elles offrent une solution pour divers champs d'application et leurs l'hybridation offre l'optimale des solutions.

D'autre part, la technologie de la vidéosurveillance basée sur les nuages a ouvert beaucoup de voies de recherche. Elle offre l'ensemble de ressources requis pour le bon fonctionnement des opérations de surveillance. Parmi les axes de recherche de la vidéosurveillance sur nuage, on cite : les applications de traitement de vidéos en temps réel, le passage à grande échelle des traitements, la gestion d'énergie des dispositifs de capture, la gestion des bibliothèques multimédia et la gestion des ressources.

8. Références

- Bramberger, M., Doblander, A., Maier, A., Rinner, B., & Schwabach, H. (2006). Distributed embedded smart cameras for surveillance applications. *computer*, 39(2), 68-75.
- Cloudatastructure Inc. (2018). Cloudatastructure Inc - Building Automation Solutions. [online] Available at: <http://cloudatastructure.com/> [Accessed 3 Feb. 2018].
- Coombes, P. (2007). ZoneMinder: A Linux-based camera monitoring and analysis tool. open source software.
- Goland, Y., Whitehead, E., Faizi, A., Carter, S., & Jensen, D. (1999). HTTP Extensions for Distributed Authoring--WEBDAV(No. RFC 2518).
- Hampapur, A., Brown, L., Connell, J., Pankanti, S., Senior, A., Tian, Y. "Smart Surveillance: Applications, Technologies and Implications" dans 2003 Joint Conference of the Fourth International Conference on Information, Communications & Signal Processing and Fourth Pacific-Rim Conference on Multimedia, pp. 1133- 1138, vol.2. New York, État-Unis, December 15-18, 2003.
- Han, J., Choi, N., Chung, T., Kwon, T. T., & Choi, Y. (2014). A target-centric surveillance system based on localization and social networking. *Multimedia tools and applications*, 73(1), 241-265.
- Kulkarni, G., Waghmare, R., Palwe, R., Waykule, V., Bankar, H., & Koli, K. (2012, October). Cloud storage architecture. In *Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on* (pp. 76-81). IEEE.
- NAS, DAS and SAN: What's the Best Archive for IP Surveillance?. (2018). Retrieved from <https://www.securityinfowatch.com/whitepaper/12292320/nas-das-and-san-whats-the-best-archive-for-ip-surveillance>
- Neal, D., & Rahman, S. M. (2012, December). Video surveillance in the cloud-computing?. In *Electrical & Computer Engineering (ICECE), 2012 7th International Conference on* (pp. 58-61). IEEE.
- Orwell, G. (2009). *Nineteen eighty-four*. Everyman's Library.
- Rinner, B., & Wolf, W. (2008). An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10), 1565-1575.
- San Miguel, J. C., Bescós, J., Martínez, J. M., & García, Á. (2008, May). Diva: a distributed video analysis framework applied to video-surveillance systems. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on* (pp. 207-210). IEEE.
- Surveillance. (n.d.) Collins Discovery Encyclopedia, 1st edition. (2005). Retrieved September 16 2017 from <http://encyclopedia2.thefreedictionary.com/surveillance>.
- Suvonvorn N (2008) A video analysis framework for surveillance system. In: *Proceedings of the 2008 IEEE 10th workshop on multimedia signal processing*, pp 867–871
- Tate, J., Beck, P., Ibarra, H. H., Kumaravel, S., & Miklas, L. (2016). *Introduction to storage area networks*. IBM Redbooks.
- Wang, G., Tao, L., Di, H., Ye, X., & Shi, Y. (2012). A scalable distributed architecture for intelligent vision system. *IEEE transactions on Industrial Informatics*, 8(1), 91-99.
- Whitaker, R. (2001). *Big Brother. com: la vie privée sous surveillance*. Presses Université Laval.
- Yuan, X., Sun, Z., Varol, Y., & Bebis, G. (2003, July). A distributed visual surveillance system. In *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on* (pp. 199-204). IEEE.
- Zhou, L. (2016). *Design and implementation of a cloud based intelligent surveillance system* (Doctoral dissertation, Auckland University of Technology).

Zwolenski, M., & Weatherill, L. (2014). The digital universe: Rich data and the increasing value of the internet of things. *Australian Journal of Telecommunications and the Digital Economy*, 2(3), 47.

Chapitre 4 : Proposition de solution pour le traitement de vidéo dans un environnement distribué

Chapitre 4 : proposition de solution pour le traitement de vidéo dans un environnement distribué

1. Introduction

Le traitement de la vidéo possède une multitude d'applications dans notre vie quotidienne, notamment le traitement de la vidéosurveillance qui assure une protection des biens et des personnes à l'échelle nationale comme à l'échelle internationale. L'inconvénient majeur que peut rencontrer ce type d'application est la taille importante des vidéos à traiter, ce qui nous force à mettre en place des exigences et des conditions à vérifier afin d'assurer une qualité de service acceptable. Le traitement de vidéo dans un environnement distribué et à grand échelle permet de résoudre la plupart des problèmes liés aux manque de ressources nécessaires pour l'accomplissement des tâches. Les ressources que peut offrir un environnement distribué sont illimitées, ce qui est bénéfique pour assurer une scalabilité horizontale de ces ressources.

Dans ce chapitre, on va présenter une méthode originale de traitement de vidéo dans un environnement distribué que ce soit basé sur les grilles de calcul ou sur les nuages de calcul. La solution proposée dans Dib et Sellami (2018) surmonte l'inconvénient de traitement des vidéos de façon partielle sur des nœuds distants. Cette façon de traitement peut ne pas être fiable pour les vidéos qui ne supportent pas le découpage. La méthode a été réalisée sur une opération basique de traitement de vidéo qui est la détection des limites des shots (plans/clichés). Cette détection de shots est aussi connue comme la segmentation temporelle des vidéos. Elle est nécessaire avant toutes autres opérations telles que la segmentation, l'annotation, l'indexation et la recherche à base de contenu. Selon le mode de transition entre les shots adjacents, les limites sont classées en coupures abruptes ou en transition progressive (fades, wipes, et dissolves).

2. Composantes de vidéo

La vidéo est un enregistrement d'images et de sons, généralement sous forme de fichier numérique. Elle décrit une séquence d'activités et le processus dynamique des événements. La séquence d'images est projetée sur un dispositif d'affichage avec un ordre et une fréquence suffisante et nécessaire pour que la vidéo soit bien présentée.

Parmi les nombreuses représentations vidéo possibles, le shot est l'élément sémantique le plus fondamental, constitué d'une série d'images consécutives interreliées prises en continu par une seule caméra et représentant une action continue dans le temps et dans l'espace. La vidéo est composée d'un ensemble de scènes. Une scène vidéo est définie comme un ensemble de shots sémantiquement liés et temporellement adjacents décrivant et transmettant un concept ou une histoire de haut niveau (Puri et al., 2018).

Un frame, ou une image, est l'unité atomique de la structure de la vidéo. Une collection de frames représente un shot. Les deux frames d'extrémités d'un shot sont appelés 'limites de shot'. Dans le domaine d'annotation de vidéo, les limites de shots sont parfois considérées comme les frames clés qui marque une vidéo. La figure 16 montre les principales composantes d'une vidéo.

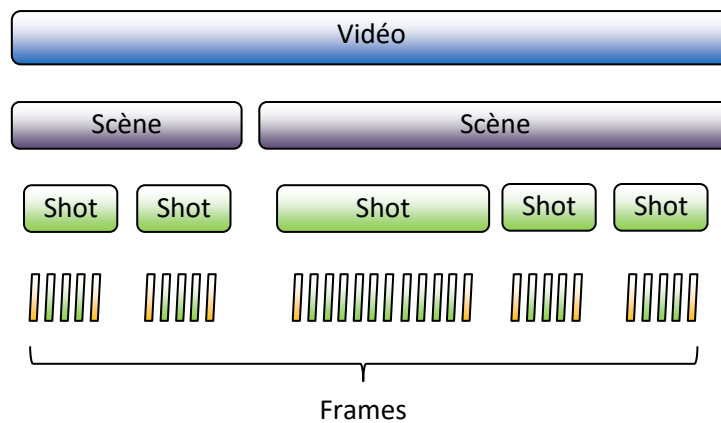


Figure 16. Structure d'une vidéo.

3. Problématique

Le traitement des vidéos sur des environnements distribués nécessite l'implication d'un ensemble de nœuds traitants. Dans le cas où la vidéo à analyser est de petite taille, le traitement de la vidéo est pris en charge par un seul Job sur un seul nœud, ce qui ne pose pas de problématique. Par contre si la vidéo à analyser est de taille volumineuse, qui est le cas le plus courant, le traitement de cette vidéo est réalisé en trois actions principales :

- 1- Un découpage de la vidéo en parties (split).
- 2- Une réplique de ces parties sur l'ensemble du système de fichiers distribué.
- 3- Exécuter les traitements demandés sur les parties de la vidéo.

Cette façon de faire dans un environnement distribué peut être bien fonctionnelle pour certaines tâches comme la recherche à base de contenu et la détection et la reconnaissance des objets. Cela est due à l'indépendance des frames composantes la vidéo l'un par rapport à son successeur. A titre d'exemple, si on veut chercher l'apparition d'une personne dans une vidéo de surveillance, on peut commencer du début de la vidéo, comme du milieu. En fin de compte on traite tous les frames de la vidéo. Par contre, cette technique de découpage en parties peut causer de faux résultats pour d'autres type d'opérations telles que le suivi du mouvement des objets et la détection des limites des shots (VSBD) qui sera l'objet d'étude de la solution proposée. L'échec de la technique de découpage dans ce type d'opération est dû à la dépendance temporelle des frames de la vidéo. Si on souhaite tracker le mouvement d'un objet dans une vidéo, on doit commencer le traitement à partir du point d'apparition de l'objet, ensuite appliquer des algorithmes spécifiques sur les frames qui succède celle qui marque l'entrée de cet objet.

Dans la détection des limites des shots, les frames d'une vidéo doivent être traités de manière séquentielle. Chaque frame est comparé avec celui qui le précède afin de calculer la variation de leurs descripteurs. Une importante variation indique à priori une présence d'une limite de shot. Le traitement séquentiel des frames représente une entrave pour la recherche des limites des shot dans un environnement distribué. Cela est causé par le traitement partiel des vidéos car on perd l'ordre de ses parties une fois planifié d'être réalisé en parallèle indépendamment l'un de l'autre. De plus, on perd l'information de présence ou non de limites de shot dans les extrémités des parties.

À savoir qu'une vidéo divisée en parties peut être traité par plusieurs nœuds distants, la récupération des informations de présence ou non des limites de shot aux extrémités des parties est coûteuse. Avec la présente solution, on permet la comparaison de plusieurs frames localisés sur des nœuds distants avec une possibilité de traitement des frames dans un ordre aléatoire.

4. Travaux en relation avec la détection des limites des shots

Pour une utilisation pratique des applications multimédias sur de grands ensembles de données, de nombreuses études ont été menées afin d'accélérer le processus de VSBD. La plupart des études portent sur les méthodes et les algorithmes à utiliser. Ils inspectent les variations de mesure des images consécutives pour vérifier l'existence d'une limite. Ci-

dessous des travaux qui ont portés sur l'accélération du processus de calcul mais toujours dans un domaine séquentiel non distribué.

4.1.Accélération des calculs de VSBD

(Shekar et al., 2016) ont proposé une nouvelle technique VSBD basée sur le résidu spectral d'un frame de vidéo. Ce coefficient est obtenu en analysant le spectre logarithmique du frame qui donne la nouveauté en comparaison avec le précédent. Les limites sont détectées lorsqu'une variation significative se produit dans des régions innovantes des frames adjacents.

D'autres techniques proposées dans la littérature se concentrent sur les mesures extraites d'éléments de base des frames telles que les différences d'histogrammes de couleurs et la différence entre pixels des frames adjacents comme dans les travaux de (Patel et al., 2013 ; Janwe et Bhojar, 2013 ; Shekhar et Shukla, 2016), les valeurs propres dans (Benni et al., 2015) et de nombreuses autres mesures comme présentées dans (SenGupta et al., 2015).

Dans (Birinci et Kiranyaz, 2014), une nouvelle méthode VSBD basée sur les règles de perception humaine et le fameux « Information Seeking Mantra » est proposée pour éviter les traitements inutiles et obtenir des calculs plus rapides.

D'autres approches basées sur le regroupement, comme les techniques de prétraitement (Li et al., 2009b) ou de saut de frame (Gao et al., 2011) sont utilisées pour sélectionner uniquement les frames candidats où une limite possible peut être présente. Par conséquence, les traitements sont réduits et le temps de calcul est amélioré.

4.2.VSBD à base de GPU et multi CPU

Les solutions séquentielles pour les problèmes de traitement multimédia ne sont pas bien adaptées pour les datasets des vidéos à l'échelle du Web. De nouveaux défis sont apparus pour répondre aux exigences concernant le stockage et le traitement de ces vidéos.

La mise en œuvre d'algorithmes efficaces doit être accompagnée par une solution évolutive basée sur de puissantes technologies informatiques. En 1990, en architecture informatique, les microprocesseurs ont été équipés de capacités de multithreading qui partagent des ressources. Actuellement, avec l'avancement des circuits intégrés, et afin

d'améliorer les performances globales des systèmes, deux processeurs ou plus peuvent être connectés pour former un processeur multicœur. Cette technique offre ainsi un traitement simultané plus efficace. De plus, la technologie des unités de traitement graphique (GPU) a fourni une puissance supplémentaire au traitement d'image en raison de sa structure hautement parallèle.

En accord avec les technologies multiprocesseurs et GPU, (Toharia et al., 2012) ont proposé une architecture hétérogène basée sur le multi-CPU et multi-GPU pour intégrer les GPUs à une implémentation des Moments de Zernike pour la détection des limites des shots.

(Lee et al., 2014) ont proposé une solution basée sur les GPUs pour la VSBD. Ils ont comparé la luminosité des pixels et les données d'histogramme globales calculées en parallèle entre les blocs de frames. L'utilisation de GPU et de multi-CPU a fourni une meilleure performance, mais pas une solution évolutive à faible coût en raison du matériel utilisé et la consommation énergétique, en particulier pour les périphériques intégrant les GPUs.

Des études similaires ont été menées sur les applications media-mining basées sur des multi processeurs comme dans (Saleem et al., 2011 ; Li et al., 2009a). Mais ces systèmes devraient être augmentés par un mécanisme de gestion des données pour la réplication et le transfert sécurisé à grande échelle.

4.3. Traitement de vidéos à base de MapReduce

L'apparition des nouvelles technologies et frameworks des big data a émergé pour remplacer des schémas traditionnels. Ces derniers reposent généralement sur une interface de transmission de messages (MPI) telle que le calcul haute performance (HPC) et les grilles de calcul. De nombreux frameworks ont été développés afin de fournir le meilleur contrôle des ressources informatiques et de stockage sur l'infrastructure des cloud. La caractéristique commune de la plupart des frameworks des cloud est leur implémentation du modèle de programmation MapReduce (Dean et Ghemawat, 2010). Introduit par Google, MapReduce est une infrastructure logicielle parallèle simple et puissante. Il permet de développer facilement des applications évolutives, qui traitent de grandes quantités de données sur un cluster de machines de manière fiable et tolérante aux pannes. De nombreuses études ont

porté sur MapReduce pour améliorer les performances de traitement des données multimédias sur des datasets à grande échelle.

Pour effectuer une annotation vidéo à l'échelle de YouTube, (Morsillo et al., 2010) ont proposé un algorithme basé sur MapReduce pour construire un arbre vectoriel distribué de plus proche voisins.

(Cheng et al., 2013) ont proposé un algorithme basé sur MapReduce pour effectuer un traitement parallèle sur plusieurs serveurs. Ils combinent des classificateurs faibles pour accélérer l'algorithme de soustraction d'arrière-plan d'image.

Dans une autre application de recherche d'informations à l'intérieur des vidéos basée sur MapReduce, (Gu et al., 2013) ont proposé l'utilisation du Locality-sensitive hashing (LSH) pour calculer la similarité entre les clips vidéo. De nombreux autres travaux se concentrent sur l'utilisation de MapReduce pour le traitement vidéo, mais nous n'avons vu aucune étude traitant le VSBD basée sur le modèle de programmation MapReduce. Cependant, il peut bien l'améliorer.

5. Présentation de la méthode proposée

Dans cette section, on va présenter l'approche proposée de traitement de vidéo dans un environnement distribué que ce soit une grille de calcul ou un nuage. Cette approche permet de résoudre le problème des opérations qui nécessitent un traitement séquentiel des consécutifs frames dans un système distribué et parallèle.

5.1. Frameworks de base

La solution proposée est basée sur le modèle de programmation MapReduce. Ce modèle est adopté par la plupart des outils et des frameworks supportés par les intergiciels de grille de calcul. Apache Hadoop est le framework référence d'implémentation de MapReduce. Il peut être inclut dans une grille de calcul à base de l'intergiciel Unicore à titre d'exemple.

5.1.1. MapReduce

Le framework MapReduce a été initialement mentionné dans un article de Google (Dean et Ghemawat, 2008). Il a été défini comme un modèle de programmation pour le traitement et la génération de grands ensembles de données de manière parallèle et

distribuée. En tant que structure de programme, MapReduce est composé de deux primitives principales : Map et Reduce. En outre, une troisième primitive appelée ‘Shuffle’ est fournie pour des opérations de regroupement, de filtrage et de tri. Le framework MapReduce prend les données en entrée d’une application, les divise en blocs indépendants et les répartit sur l’ensemble des nœuds du cluster. Les tâches Map (mappeurs) qui s’exécutent de manière complètement parallèle consomment exclusivement des paires clé/valeur. Ils les traitent et, si nécessaire, produit un ensemble d’autres paires clé-valeur intermédiaires ou de sorties. La fonction Map est donnée par :

$$\text{Map (Key1, Value1)} \rightarrow \text{List (Key2, Value2)}.$$

Les sorties intermédiaires sont triées et regroupées efficacement par les tâches ‘Shuffle’ avant d’être utilisées par la fonction Reduce. Toutes les valeurs intermédiaires associées aux mêmes clés intermédiaires sont regroupées et envoyées aux tâches de réduction spécifiques (réducteurs) en fonction de leur clé. La fonction Shuffle est donnée par :

$$\text{Shuffle (List (Key2, Value2))} \rightarrow (\text{Key2}, \text{List (Value2)}).$$

A la fin, le réducteur reçoit des paires clé-valeur intermédiaires ayant la même clé provenant de plusieurs mappeurs ou shuffleurs. Il les fusionne et produit zéro ou une nouvelle paire clé-valeur. La fonction de réduction est donnée par :

$$\text{Reduce (Key2, List (Value2))} \rightarrow \text{List (Key3, Value3)}.$$

En combinant les tâches Map, Shuffle et Reduce, des tâches complexes peuvent être réalisées avec des performances élevées de manière évolutive et fiable, ce qui est très difficile à assurer autrement.

Chaque étape et chaque Job dans MapReduce peut être exécuté sur plusieurs nœuds. En tant que framework, l’architecture de MapReduce est de type maître/esclave. Il est composé d’un seul maître ‘job-tracker’ et d’un seul esclave ‘task-tracker’ par nœud. Le rôle principal du job-tracker est d’assurer la gestion des tâches assignées aux task-trackers. La gestion inclut le suivi des états d’exécution des tâches sur les nœuds esclaves, et la prise en charge de la tolérance aux pannes en déplaçant les tâches échouées vers d’autres task-trackers.

Lorsqu'un job-tracker distribue des tâches sur les task-trackers, il veille à maintenir les calculs le plus près possible des données. Cela évitera l'exécution des gourmandes tâches supplémentaires tels que le déplacement des données d'un nœud vers un autre. Ce déplacement de données est coûteux car il produit une charge supplémentaire sur les ressources réseaux, mémoires, et processeurs. Le Job-Tracker planifie d'abord les tâches Map en premiers lieu. Les réducteurs doivent attendre les données intermédiaires issues de l'exécution des mappers. La disponibilité des données est un facteur important pour les task-trackers, la priorité est donnée aux nœuds où les données résident et aux nœuds les plus proches des données (appartenant au même rack réseau).

5.1.2. Apache Hadoop

Hadoop est un projet open-source de Apache Software Foundation. C'est un framework qui fonctionne à travers un cluster composé de plusieurs milliers de nœuds. Il est destiné à stocker et traiter des données volumineuses de manière distribuée. Il a été créé par Doug Cutting le créateur d'Apache Lucene⁸⁶ et Mike Cafarella⁸⁷ à la fin de l'année 2004. Hadoop a été conçu en suivant le modèle d'architecture de Google et il est devenu la référence de l'implémentation de MapReduce. Le cœur du projet Hadoop est constitué du système de fichiers distribué appelé (HDFS) et du modèle de calcul distribué MapReduce géré par le gestionnaire YARN.

HDFS est une implémentation open source de Google GFS (Ghemawat et al., 2003). Il fournit un système de stockage distribué, évolutif et fiable d'une grande quantité de données. HDFS a une architecture maître/esclave. Il consiste en un seul name-node (maître) et un certain nombre de Data-Nodes (esclaves). Dans le but de contrôler les opérations d'accès aux fichiers, les métadonnées sont gérées par le name-node en mode centralisé. Tandis que les données sont divisées en blocs de taille fixe, également appelés Chunks (paramètre configurable généralement de 64 Mo). Ces blocs sont répliqués à travers le HDFS afin de fournir une tolérance aux pannes et d'augmenter la disponibilité des données.

⁸⁶ Apache Lucene : un projet open source qui offre un serveur d'indexation et de recherche en hautes performances, disponible en ligne à l'adresse : <https://lucene.apache.org/>

⁸⁷ Mike Cafarella : un informaticien spécialisé dans les systèmes de gestion de bases de données. Il est professeur agrégé à l'Université du Michigan.

5.2.Préparation des données vidéos

Les données sont le noyau de toutes applications. Leur présentation en un format approprié est cruciale pour assurer la fiabilité de leur consommation. Les données requises pour la solution VSBD proposée sont un ensemble de vidéos divisés en consécutifs frames. La taille de chaque image est significativement plus petite que la taille d'un chunk HDFS typique (64 méga octet par default). La taille d'un frame dépend de la qualité de la vidéo et du contexte de sa capture. Pour les frames extraits des vidéos de surveillance, la taille varie en fonction des conditions de l'endroit surveillé et du dispositif de capture utilisé. Dans un endroit fermé et pour des caméras de surveillance mobile, les vidéos sont généralement capturées à faible qualité. Cela est dû aux faibles détails présents dans une scène tels que le cas des caméras d'intérieur. Comme peut être le résultat des contraintes exigées par les dispositifs comme la gestion de l'énergie qui est un facteur important pour définir la qualité de la vidéo capturée. La consommation de l'énergie dans les caméras mobiles est partagée entre le traitement, l'analyse et la transmission de données. D'un autre côté, les vidéos de surveillance capturées par les autorités sont généralement de bonne qualité. Cela est justifié par le besoin de capturer le moindre détail dans des endroits ouverts, dynamiques et riches en informations.

La taille considérablement petite des frames par rapport à la taille des chunks favorise une charge de mémoire supplémentaire pour le name-node lorsqu'il traite d'énormes quantités de petits fichiers. Afin de détourner ce problème, l'utilisation des fichiers de séquence est favorable.

Un fichier de séquence est un fichier plat constitué de paires clé/valeur binaires. Ce type de fichier sera discuté dans la section 5.2.2. Les clés et les valeurs des données de la solution VSBD proposée sont respectivement les noms des frames et leur contenu binaire. Une fois les données vidéo converties en un ou plusieurs fichiers plats, elles seront chargées sur le HDFS dans des blocs de données. Cette opération est également appelée processus d'ingestion. La figure 17 montre les différentes étapes de processus de préparation des données vidéo. Les étapes ci-dessous sont appliquées sur les données avant d'être présentées aux mappers.

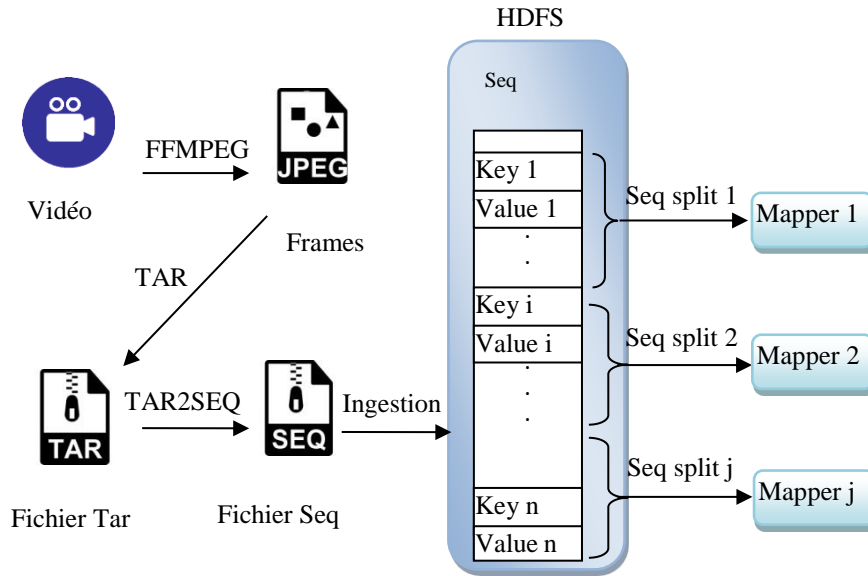


Figure 17. Étapes de préparation des données et leur ingestion sur l'HDFS.

5.2.1. Extraction des frames

Le découpage des vidéos en une séquence de frames consécutives est une étape importante pour toute application de traitement de vidéo. Ils existent plusieurs outils et bibliothèques dans ce contexte qui peuvent être utiles. FFmpeg⁸⁸, VLC⁸⁹, et VirtualDub⁹⁰ sont des outils puissants offerts gratuitement et en licence libre pour effectuer des traitements sur des données multimédia telles que la conversion, le fractionnement, l'indexation, le dumping et d'autres fonctionnalités.

L'extraction des frames d'une vidéo est effectuée en spécifiant le paramètre de débit pour produire un nombre de frame par seconde (FPS). Ce paramètre est connu comme l'un des déterminants de facteur important de qualité de vidéos. Il offre un compromis entre qualité et taille de vidéo. Un débit moyen d'extraction de frames doit être entre 24 et 30 fps en fonction de la norme⁹¹ utilisée. Le débit d'extraction de frames des vidéos original est maintenu parce qu'il est hors de portée de cette étude. Sinon, il peut être utilisé pour réduire le nombre de tâches de Map traitant les frames extraits, tout en conservant le minimum requis de qualité vidéo qui garantit les meilleures performances de détection des limites des

⁸⁸ FFmpeg: site officiel: <http://ffmpeg.org>

⁸⁹ VLC : VideoLAN Client, site officiel <https://www.videolan.org/>

⁹⁰ VirtualDub : site officiel : <http://www.virtualdub.org/>

⁹¹ Débit fps normalisé à 24 au cinéma, à 25 par le système européen PAL (Phase Alternating Line), et 30 pour la norme NTSC (National Television System Committee) adopté aux états unis et au Japon.

shots. Lors de l'extraction des frames, un numéro séquentiel leur sont attribué afin de permettre leur dénotation lorsqu'ils seront dispersés sur le HDFS.

5.2.2. Génération des fichiers de séquence

Le problème de surcharge de mémoire au niveau des name-nodes est causée par le traitement d'énormes quantités de petits fichiers. Ce problème peut être surmonté en utilisant certains formats tels que Hadoop Archives⁹² (fichiers HAR), Hipi Image Bundle (HIB) pour le projet HIPI (Hadoop Image Processing Interface project) (Sweeney et al., 2011) et les fichiers de séquence (Seq). Le but d'utiliser ce type de fichier est de regrouper une quantité de petits fichiers dans un seul fichier plat, ce qui réduit l'utilisation du disque et accélère le chargement des données dans MapReduce. Au lieu de lancer plusieurs tâches d'accès aux supports de stockage pour la lecture des fichiers de données de petite taille, il les prend tous à la fois en respectant un format bien précis. Les petits fichiers sont présentés dans le fichier de séquence sous la forme d'un ensemble de paires clé/valeur consécutives. Dans le cas général, lorsqu'il s'agit d'une application de traitement de vidéo ou d'image, les paires clé/valeur sont constituées respectivement des noms de frame et de leur contenu binaire. La conversion de petits fichiers en un fichier de séquence peut être effectuée par programmation via des APIs spécialisés, ou en utilisant l'un des outils existants. Tar2Seq ("A Million Little Files", 2018) est une bibliothèque open source libre qui permet la génération des fichiers Seq. Elle prend comme paramètre un fichier compressé avec l'algorithme TAR et produit un fichier plat.

5.2.3. L'ingestion des fichiers de séquence sur le HDFS

La dernière étape de la préparation des données est l'ingestion en batch des fichiers de séquence générés à partir de l'étape précédente. L'ingestion des données n'est que leur remontée sur le système de fichier distribué. Chaque fichier est stocké comme une séquence de blocs de la même taille, à part le dernier qui prend ce qui reste des données. Chaque bloc de données est stocké dans un fichier distinct sur le DataNode. Ainsi, pour améliorer la fiabilité et les performances d'accès aux données, chaque bloc est répliqué sur différents DataNodes.

⁹² Fichiers HAR : un fichier plat conforme à JSON.

5.3. La phase Map

Une fois que les données vidéo sont divisées en parties et remontées sur le HDFS, les mappers les prennent un par un et les lisent comme un ensemble de paires clé-valeur. Un Job MapReduce est attribué à chaque bloc de données. Un ensemble de tâches Map est affecté à chaque Job et lancé sur des conteneurs de ressources. Cela reflète la notion de Maître/esclave, où le Job-tracker vu comme maître lance les nœuds esclaves pris en charge par les task-trackers. Il supervise l'exécution des tâches jusqu'à la fin des traitements. Pour chaque paire clé/valeur, le Mapper effectue les opérations suivantes :

5.3.1. Récupération des données d'entrée

Les données vidéo sont stockées sous forme de frames représentés par des paires clé/valeur dans un fichier de séquence sur le HDFS. L'élément 'value' dans la paire clé-valeur est une donnée binaire qui peut être récupérée et convertie en un format d'image approprié. La conversion de la valeur est effectuée par une simple conversion de la forme binaire de l'image à sa forme équivalente et requise. Plusieurs bibliothèques de traitement d'images sous plusieurs langages de programmation offrent des formats d'image différents. Des exemples de formats d'images disponibles sur des bibliothèques libres sont :

- 1- Mat pour OpenCV⁹³ en Java, .Net, C++ et Python ;
- 2- Bitmap pour Accord.Net⁹⁴ sous C# et Computer Vision Sandbox⁹⁵ ;
- 3- MBFImage pour OpenImaj (Hare et al., 2011) en Java.

5.3.2. Extraction des caractéristiques des frames

Les caractéristiques localisées dans une image peuvent être une structure spécifique tels qu'un point, un bord, une forme ou un objet. Elles peuvent être locales ou globales. Le premier type est d'une structure de bas niveau, et il est défini comme un modèle d'image

⁹³ OpenCV, accessible en ligne sur <https://opencv.org/>, est une bibliothèque open source qui regroupe plus de 2500 algorithmes de vision par ordinateur. Elle est disponible sur plusieurs systèmes d'exploitation et sous plusieurs langages de programmation tels que C, Java, Python, et C# sous le nom Emgucv.

⁹⁴ Accord.Net : écrit en C#, accessible en ligne sur : <http://accord-framework.net/>, est un framework de machine learning combiné avec des bibliothèques de traitement d'image et d'audio. Il est destiné à des applications de vision et d'audition par ordinateur, de traitement de signal, et des statistiques.

⁹⁵ Computer Vision Sandbox : accessible en ligne sur : <http://www.cvsandbox.com/>, est une bibliothèque disponible en C# qui permet de résoudre des problèmes liés aux caméras de surveillance, détection des objets, traitement parallèle des vidéos, et d'autre application de traitement d'image et de vidéo.

qui diffère de son voisinage immédiat. Les caractéristiques locales sont également liées à la couleur, l'intensité et la texture qui décrivent certains points intéressants des images. Les caractéristiques locales intéressantes sont celles qui sont faiblement ou ne sont pas affectées par des complications telles que le bruit, le zoom, la rotation et dites invariantes à l'échelle et à l'orientation.

Néanmoins, les fonctionnalités locales utilisées seules ne peuvent pas donner une vue globale de l'aspect, d'où la nécessité des caractéristiques globales pour décrire l'ensemble du contenu de l'image. De nombreuses caractéristiques globales ont été proposées dans la littérature. On fait référence à l'utilisation d'histogrammes de couleurs, discutés en détail dans (Ljubovic et Supic, 2013), les moments de couleur (Flickner et al., 1995), l'histogramme de direction des bords (Park et al., 2000), etc.

Il existe un certain nombre de méthodes pour détecter et décrire les caractéristiques locales telles que SIFT (Lowe, 1999), Speeded Up Robust (SURF) (Bay et al., 2006), FAST orienté et BRIEF pivoté (ORB) (Rublee et al., 2011), etc. La détection de caractéristiques consiste à identifier les points d'intérêt, appelées aussi points clés, telles que les régions, les courbes et les indices visuels saillants. Divers détecteurs de caractéristiques peuvent être utilisés tels que Canny et Deriche pour la détection des bords, Haris et Shi-Tomasi pour la détection des coins, Laplacien de Gaussian (LoG) et Determinant of Hessian (DoH) pour la détection des blobs, etc. Généralement, les points clés les plus intéressants dans une image sont ceux ayant un contraste élevé et sont bien situés sur un bord. Une sélection appropriée des points clés est très utile pour surmonter certains problèmes associés à l'identification des objets d'intérêt, comme dans le cas de changements d'échelle d'image, de l'angle de vue, d'éclairage, de qualité d'image, etc. (Li et al., 2015).

Cependant, la description des caractéristiques, comme leur détection, est une étape importante dans la vision par ordinateur. Elle vise à assigner un vecteur de description à chaque point clé détecté. SIFT est l'une de plusieurs méthodes qui fournit un détecteur de caractéristiques. Il calcule la magnitude du gradient et l'orientation du gradient pour créer un ensemble d'histogrammes sur une fenêtre centrée sur chaque point clé identifié. Inspiré par SIFT, des méthodes avancées ont été développées telles que SURF, l'histogramme des gradients orientés (HOG), et l'emplacement du gradient et l'histogramme d'orientation (GLOH). BRIEF (Calonder et al., 2012) est un autre descripteur de caractéristiques basé sur

des comparaisons d'intensité de pixel, également décrit comme un descripteur binaire. Il est devenu populaire grâce à sa rapidité et sa faible consommation de mémoire. BRISK (Leutenegger et al., 2011), FREAK (Alahi et al., 2011) et ORB sont également inspirés par BRIEF. Ils diffèrent les uns des autres dans la façon dont les paires de pixels sont spatialement échantillonnées dans le patch d'image entourant un point clé donné.

Outre l'utilisation des caractéristiques locales, les caractéristiques globales représentent une bonne alternative, elles peuvent être calculées à partir de plusieurs modèles globaux tels que GIST (Oliva et Torralba, 2001) et Sac de Caractéristiques (BoF) (Sivic et Zisserman, 2003). Le premier est un modèle qui simule les yeux humains pour recevoir une information instantanée et l'identifier. Il résume la magnitude du gradient et l'orientation du gradient pour différentes parties d'une image. BoF est un autre descripteur d'objet visuel, qui est inspiré par la représentation du Sac de Mots (BoT) pour la classification du texte. Il est largement utilisé pour sa performance et sa possibilité de passage à grand échelle.

Cependant, au niveau de la solution proposée, le Mappeur doit implémenter une méthode de détection et de description de caractéristiques. Pour cela, on a choisi la méthode GIST pour fournir les descripteurs globaux utiles à la comparaison entre les frames. GIST fournit une description approximative du contenu du frame en utilisant les informations de gradient de différents endroits de l'image. Il encode un ensemble de dimensions perceptives (naturalité, ouverture, rugosité, expansion, robustesse) qui représente la structure spatiale dominante d'une scène (Oliva et Torralba, 2001). Une fois les caractéristiques sont extraites, leur description est réalisée en effectuant une convolution de chaque bande du frame. Trois bandes d'image sont utilisées pour assurer une meilleure exploitation des informations visuelles présentes dans l'image d'entrée qui est redimensionné à (128 x 128) pixels, avec 32 filtres Gabor à 4 échelles et 8 orientations, produisant 32 cartes de caractéristiques. Les caractéristiques calculées sont également divisées en 16 blocs, et la moyenne est prise pour chacun d'eux. Il en résulte 1536 descripteurs GIST pour une seule image (32 cartes de caractéristiques x 16 blocs x 3 bandes). Cela présente un vecteur de grande dimension qui peut ralentir les tâches de calcul pour une grande quantité de frames dans un environnement non distribué. Il est à noter qu'une utilisation de descripteurs de caractéristique locales peut produire beaucoup plus de données de description. Des méthodes de réduction de dimension

peuvent être utilisées telles que l'Analyse à base de Composante Principale (ACP) et L'Analyse des Correspondances Multiples (ACM) (Abdi et Valentin, 2007).

5.3.3. Mesure de similarité entre les frames

Les frames d'entrée sont traités séparément les uns des autres. C'est-à-dire que deux frames peuvent être traitées séparément par deux mappers différents sur des nœuds différents. Par conséquent, comparer entre deux frames sur des nœuds distants n'est pas une tâche évidente. La mesure de similarité entre les frames est effectuée en comparant leurs descripteurs GIST. Le but de cette étape est de définir une mesure simple qui peut être utilisée parfaitement pour déterminer la similarité entre les images distantes. Ainsi, on propose d'utiliser la transitivité comme une relation intuitive. Dans ce contexte, la relation de transitivité signifie que lorsqu'un frame f_1 est similaire à un frame f_2 , et f_2 est à son tour similaire à un frame f_3 , alors f_1 est également similaire à f_3 . Ainsi, la relation de transitivité peut être appliquée comme suit : si un Mapper 1 (M_1) analyse f_1 , et si un Mapper 2 (M_2) analyse f_2 , alors un élément intermédiaire f_0 peut être utilisé par M_1 et M_2 , où le rapport entre f_1 et f_0 calculé par M_1 peut être comparé avec le rapport entre f_2 et f_0 calculé par M_2 . Cependant, le rapport utilisé, et appelé similarité intermédiaire (IntSim), est présenté par la mesure de similarité donnée par :

- Au niveau de Mapper1 :

$$IntSim1 = 1 - distance(GistVector(f1), GistVector(f0))$$

- Au niveau de Mapper2 :

$$IntSim2 = 1 - distance(GistVector(f2), GistVector(f0))$$

Pour prendre une décision sur la similitude entre f_1 et f_2 dans une étape ultérieure, la différence absolue de leur similarité intermédiaire IntSim1 et IntSim2 est utilisée.

La figure 2 montre les principales étapes de la phase Map, et l'algorithme 1 détaille le cœur de la fonction Map.

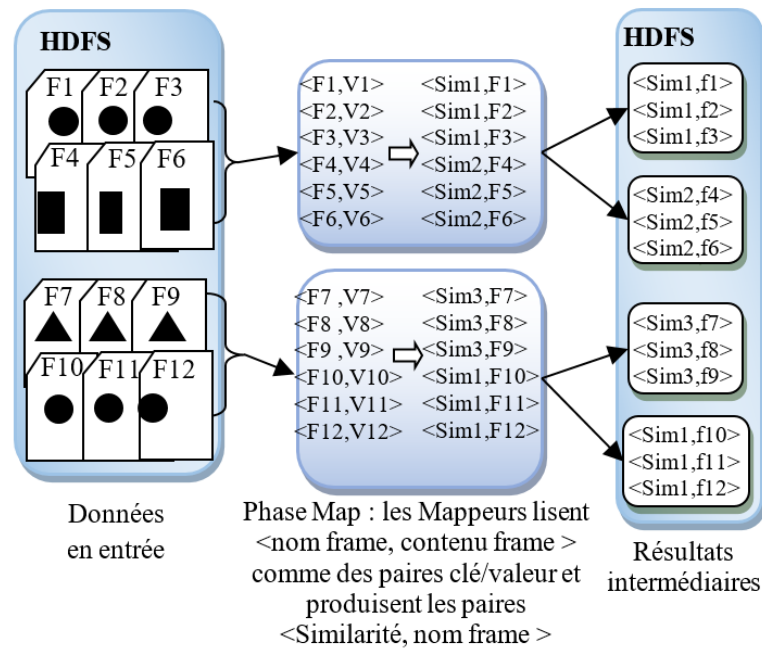


Figure 18. Le flux des traitements dans la phase Map.

Algorithm 1 : Le corp de la fonction Map

```

//Input : 1) key : le nom du frame; 2) value: le contenu du frame
//Output : le couple clé/valeur intermédiaire qui représente respectivement
la valeur de la similarité intermédiaire et le nom du frame.
map (Text key, BytesWritable value)
    Image frame = read_and_cast (value); // récupérer le contenu de
    l'image.
    Vector v = gist(frame); // vecteur de descripteurs GIST.
    Vector v0 = new vector(v.size()); // v0: vecteur intermédiaire ayant
    la meme taille que v.
    Double int_sim = 1 - CHI_SQUARE.compare(v, v0); // int_sim: la
    similitude entre l'actuel et l'intermédiaire frame, chi
    square est utilisé pour mesurer la distance entre les
    vecteurs.
    int_sim = around (int_sim, _threshold); // approximation de la valeur
    de similarité.
    emit-Intermediate (int_sim, key); // écrire le nouveau couple
    clé/valeur calculé dans le contexte Hadoop, la nouvelle
    clé est la similarité intermédiaire approximative, et la
    valeur est l'ancienne clé, qui représente le nom du
    frame.
  
```

Chaque mappeur écrit ses résultats sous forme de couples clé/valeur dans le contexte du framework Hadoop. La clé produite est la similarité intermédiaire approchée, tandis que la valeur est le nom de l'image traitée. L'approximation de la clé est effectuée

pour regrouper les frames ayant une mesure de similarité intermédiaire proche l'une de l'autre. Selon la valeur de la clé, les paires clé/valeur produites sont envoyées aux Réducteurs. Deux paires avec des clés différents signifie qu'il existe deux shots différents. Par contre, deux paires ayant les mêmes valeurs de clé signifie que les deux frames correspondants appartiennent probablement au même shot. La décision exacte sera prise par le réducteur dans la phase suivante.

5.4. La phase de réduction

Les jobs Réducteurs accèdent au HDFS pour trouver les paires clé/valeur intermédiaires produites par les mappeurs. Chaque réducteur fonctionne sur des paires ayant le même élément clé. La fonctionnalité principale du réducteur consiste à détecter les limites des shots en passant par les étapes suivantes :

5.4.1. Regroupement et tri des frames ayant la même mesure de similarité intermédiaire

Les résultats intermédiaires sur le HDFS sont prêts à être traités par des réducteurs. Les paires clé/valeur ayant la même clé sont automatiquement redirigées pour être traitées par le même réducteur. Pour expliquer le processus de groupement, le même exemple illustré sur la figure 18 est repris. Trois valeurs différentes de similarité intermédiaire se distinguent. Les paires sont groupées par clé pour former trois sous-ensembles présentés aux trois réducteurs, un pour chaque groupe. Pour l'exemple utilisé dans la figure 18, les réducteurs traitants les couples <intSim, Nom-frame> sont :

- Réducteur 1 traite les paires suivantes : <IntSim1, f1>, <IntSim1, f2>, <IntSim1, f3>, <IntSim1, f10>, <IntSim1, f11>, <IntSim1, f12>.
- Réducteur 2 traite les paires suivantes : <IntSim2, f4>, <IntSim2, f5>, <IntSim2, f6>.
- Réducteur 3 traite les paires suivantes : <IntSim3, f7>, <IntSim3, f8>, <IntSim3, f9>.

Le résultat de la phase de groupement est présenté par les lignes suivantes sur chaque réducteur :

- Au niveau du réducteur 1, le résultat produit est : 'IntSim1 : f1, f2, f3, f10, f11, f12'.
- Au niveau du réducteur 2, le résultat produit est : 'IntSim2 : f4, f5, f6'.
- Au niveau du réducteur 3, le résultat produit est : 'IntSim3 : f7, f8, f9'.

La liste des frames regroupés est créée en fonction de l'ordre d'arrivée des paires clé-valeur. L'ordre du numéro d'image est également important. Il est utile de définir les images consécutives appartenant aux shots. Pour trier les listes, il est possible d'ajouter une étape Shuffle entre Map et Reduce, ou implémenter un type de liste triée disponible dans n'importe quel langage de programmation pris en charge. On appelle 'int-Sim/ligne-shots' la ligne composée de la valeur de similarité intermédiaire et de la liste de frames triées (ex : 'IntSim1 : f1, f2, f3, f10, f11, f12').

5.4.2. Détection des limites des shots

La détection des limites des shots est d'abord effectuée en détectant ces shots. Chaque réducteur fonctionne sur une liste triée d'index tirés de la liste des noms des frames. Pour extraire les shots de la liste des index d'images, les réducteurs implémentent un algorithme pour trouver des séries de nombres consécutifs présentés par les indexes des images. Trois cas possibles peuvent être rencontrés lors de l'extraction des shots.

- Le premier concerne les listes des indexes bien triées, chacune correspondant à un seul shot comme montré par la figure 19.
- Le deuxième cas est lorsqu'un nombre d'éléments manque dans la liste des indexes. Les éléments manquants ne doivent pas affecter la détection du shot. Ce cas se produit quand une ou plusieurs images du même shot possèdent des descripteurs hautement variables qu'à ceux de leurs voisins. Cette différence peut être le résultat d'une brusque et forte variation de contraste ou de luminosité entre deux consécutives frames. La figure 20 illustre ce cas.

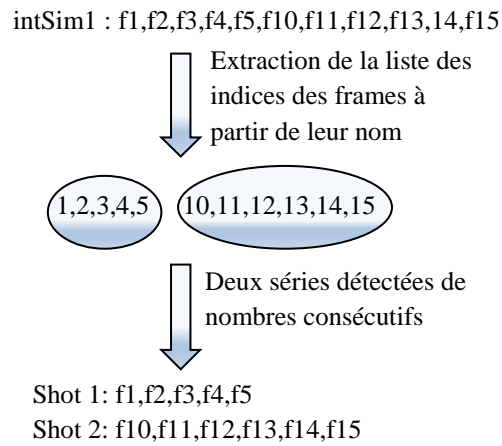


Figure 19. Cas de frames bien triés composants deux shots.

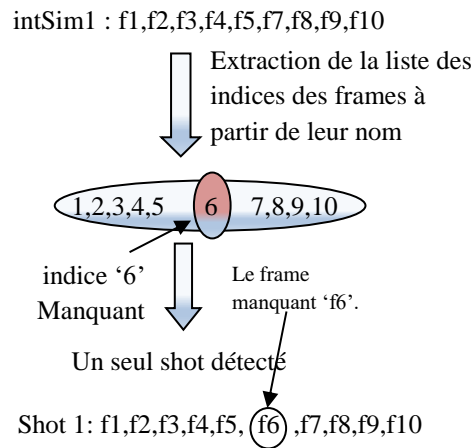


Figure 20. Cas de frame manquant dans un shot.

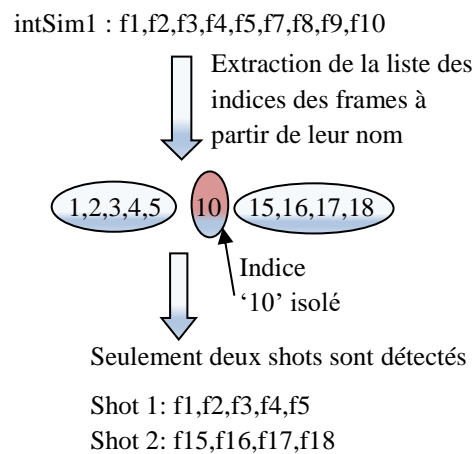


Figure 21. Cas de frame isolé.

- Le dernier cas est lorsqu'un ou plusieurs éléments se trouvent isolés. Les éléments manquants discutés dans le cas précédent deviennent des éléments isolés dans une ou plusieurs autres séries de listes d'index. La figure 21 illustre ce cas.

Une fois que les shots sont détectés, les limites de ces shots peuvent également être définies. Le résultat final, qui contient les limites de shots détectées est écrit sur le HDFS. L'algorithme 2 décrit les principales étapes des fonctions des réducteurs. La figure 22 illustre les résultats produits après l'exécution de la phase de réduction sur le même exemple présenté sur la figure 18.

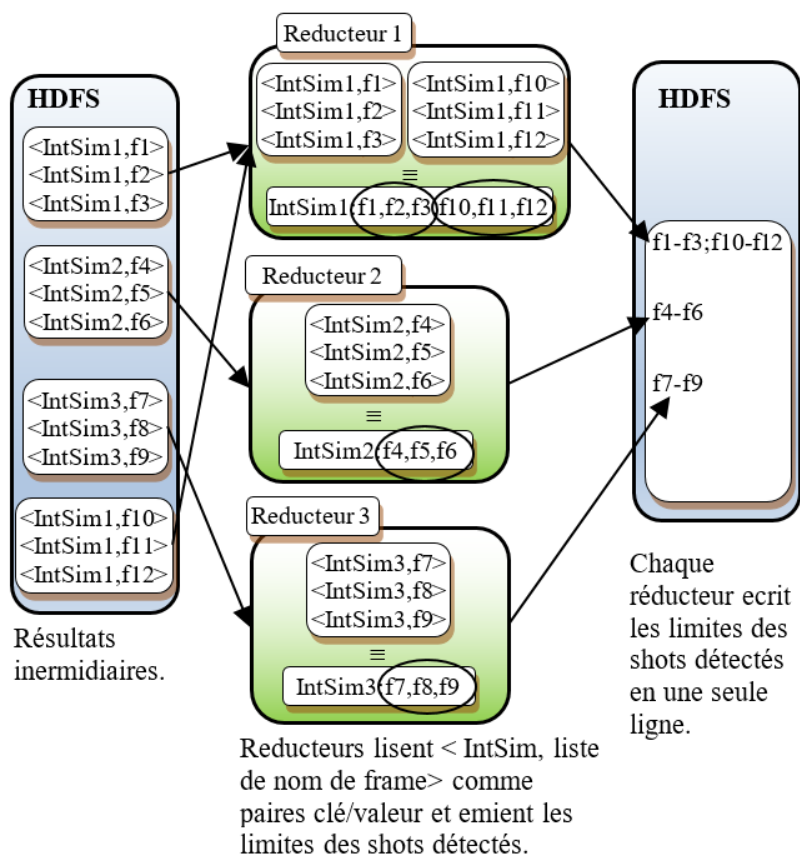


Figure 22. Flux du processus de la phase de réduction.

Algorithm 2: Reducer algorithm

```
//Input : 1) key : valeur rapprochée de la similarité intermédiaire ; 2)
value : liste de nom de frame.
//Output : les limites des shots détectées
reduce (double key, Iterable<Text> value)
    List<frame-indexes> lfi = extract_frame-indexes(value) ; // crée la
        liste des indexes des frames.
    Collection c= Collections.sort(lfi); // trier la liste des indexes des
        frames.
    List<shot> ls = extract_shots(c) ; // détecter des shots à partir de la
        liste des indexes consécutives.
    For each shot s in ls
        Emit (s.first, s.last); // extraire et émettre des limites de
            shots sur le Contexte de Hadoop.
```

6. Expérimentation

La présentation d'une solution à une problématique doit être accompagnée par une évaluation qui montre l'amélioration et l'ajout apporté à ce qu'il existe comme solutions antérieurs. Dans cette section, on va présenter les outils utilisés pour parvenir à implémenter la solution proposée y compris les frameworks, les APIs et les systèmes d'exploitation. Ainsi, on discute l'évaluation de la précision de la solution et des performances liées à l'exécution de l'application dans un environnement distribué.

6.1. Environnement d'exécution

Dans cette partie, on va présenter l'environnement utilisé pour l'exécution de l'application développée dans le cadre de cette étude.

6.1.1. Ressources physiques

Les ressources physiques requises à ce niveau de test ne sont pas très exigeantes en termes de puissance de calcul et de mémoire vive. Ci-dessous, les ressources physiques utilisées montrées par le tableau 1.

Tableau 1: Ressources physiques utilisées pour l'évaluation de l'exactitude de l'application.

Ressources	Description
Processeur	
CPU	Intel Core i7 (6th Gen) 6820HQ / 2.7 GHz
Max Turbo Speed	3.6 GHz
Nombre de cœur	Quad-Core
Cache	8 MB
Mémoire	
RAM	32 GB
Technology	DDR4 SDRAM - non-ECC

6.2.Ressources logicielles

Les ressources logicielles incluent tous frameworks, APIs et plateformes. Le tableau 2 montre les logicielles utilisées pour l'évaluation de la solution.

Tableau 2: Ressources logicielles.

Ressources	Description
Système d'exploitation	Linux Ubuntu 14.04 (64 bit)
Oracle VM VirtualBox	Version 5.2.1
JVM	JRE 8 mise à jour 161
Hadoop	Version 2.7.1
OpenImaj	Version 1.3.1
Ganglia	Version 3.7.2

Pour l'évaluation de l'exactitude de l'application, Hadoop est installé sous l'un des deux modes suivants :

- 1- Mode Pseudo-distribué : ce mode permet de mettre en place un nœud de cloud à base de Hadoop en simulant la distribution des traitements. Le Job-tracker et le Node-tracker pointent tous deux sur les mêmes ressources physiques mais en utilisant la technologie des threads pour assurer le traitement distribué. L'utilisateur aura l'impression de travailler sur un environnement réellement distribué. Ce type

d'installation est utile dans le cas de mesure de résultats de précision comme le cas de notre premier test.

- 2- Mode complètement distribué : ce mode d'installation permet de fournir un cloud installé sur plusieurs ressources physiques ou virtuelles. L'inconvénient de travailler avec des ressources physiques est le besoin de machines et de ressources réseaux très performantes. Cela peut facilement être surmonté en virtualisant les ressources physiques avec les machines virtuelles ou les conteneurs. A savoir, la capacité de mémoire et le nombre de cœurs attribué à chaque machine virtuelle doit être étudié selon les besoins d'utilisation et les caractéristiques de la machine réelle.

6.3.Evaluation de l'exactitude de la solution

Dans cette section, on mesure les critères d'évaluation de la méthode VSBD proposée. Ces mesures correspondent aux valeurs de l'exactitude des résultats basés sur le calcul de la précision, le rappel et le F-score. Pour chaque vidéo de test, on compare le nombre de limites de shot détecté en utilisant la méthode proposée, avec le nombre de limites de shots qui existent réellement. Les résultats obtenus sont de même comparés avec les résultats d'autres études.

6.3.1. Base de vidéos utilisée

La performance de l'approche de VSBD proposée est évaluée à travers une variété de vidéos de test comme illustrées dans le tableau 3. Ces vidéos de test sont téléchargées depuis OPEN VIDEO PROJECT (Geisler and Marchionini, 2000) et utilisées avec leur qualité originale. Cet ensemble de données de test fournit des clips vidéo provenant de diverses sources. Il représente un bon support pour mesurer l'exactitude de la méthode proposée en raison de la diversité de leurs contextes, contenus et conditions de prise des vidéos. Les points ci-dessous citent les principales raisons de choix de la base de données indiquée :

- Diversité des contextes : les vidéos sont capturées dans différents contextes y compris à l'intérieur des locaux et de building, aux endroits de travail et à l'extérieur dans la nature ou avec une forte présence d'activité. Cela fournit une variété de modèles et d'activités dans le contenu des vidéos.

- Diversité des conditions d'enregistrement : les conditions de l'endroit et le temps de captures d'un vidéo affectent sa qualité. Il est important de prendre des mesures de précision d'une solution dans différentes conditions avec différentes intensités de couleur, de luminosité et présence de bruit.
- Diversité du contenu : afin d'évaluer la robustesse du schéma proposé, il est nécessaire d'inclure des cas particuliers dans les vidéos de test tels que la variation de la luminosité, le déplacement rapide de caméra, le changement rapide de la forme des objets et le brusque changement.

Les vidéos sélectionnées sont :

- 'Anni003.mpg', elle fait partie du documentaire du 25ieme anniversaire de la NASA⁹⁶. Elle contient à la fois des transitions abruptes et progressives. La vidéo comprend des transitions entre des visages humains et certains objets en mouvement et en rotation. Ce vidéo présente l'avantage de pouvoir tester la sensibilité des variations entre les frames, il contient des transitions entre shots avec très peu de changement. Cependant, rendre le détecteur de variation sensible au changement peut produire de fausses détections de limites de shots.

Tableau 3: Vidéos utilisées pour le test de l'exactitude de la solution.

Vidéo	Durée	Nombre de Frames	Nombre d'abrupte transition
Anni003.mpg	142 s.	4267	24
UGS03_009.mpg	62 s.	1881	4
Newport_4.mpeg	60 s.	1798	16
Hcil2001_01.mpg	139 s.	4204	20

- 'UGS03_009.mpg', est une autre vidéo de test documentaire utilisée. Elle inclut à la fois des transitions abruptes et progressives. De plus, elle contient différentes formes en mouvement rapide, différents niveaux de couleur et de luminosité. Elle se compose de quelques scènes génériques, naturelles et professionnelles. Cette vidéo a l'avantage de permettre le test de l'application dans différents contextes. Un

⁹⁶ NAZA : National Aeronautics and Space Administration, est l'agence gouvernementale qui est responsable de la majeure partie du programme spatial civil des États-Unis

contexte variable permet d'exposer l'application aux tests intenses concernant son bon fonctionnement.

- 'newport_4.mpeg', elle consiste en une partie d'une publicité sur la cigarette. Elle contient un ensemble de scène de la vie quotidienne prise à l'intérieure des maisons. Elle n'inclut que des transitions abruptes.
- La dernière vidéo de test 'hcil2001_01.mpg' fait partie de la conférence qui inclut des prises de vue au travail avec un mouvement de caméra rapide. Les transitions entre les shots sont à la fois abruptes et progressives.

6.3.2. Analyse de la mesure de similarité intermédiaire

La méthode de VSBD proposée est basée sur la mesure de similarité intermédiaire. Par conséquent, la performance des résultats finaux est affectée par la qualité des valeurs int-sim calculées. La figure 24 montre les int-sim calculées pour les 300 premières images de la vidéo anni003.mpg résumées dans la figure 23. Quatre groupes sont perçues, chacune présentant un shot. La transition abrupte se produit lorsque la différence entre deux consécutives int-sim est supérieure à un certain seuil.

Deux seuils sont utilisés afin d'assurer la détection des shots. Dans le cas où l'objectif est de vérifier l'appartenance d'un frame à un shot, un seuil de faible valeur libellé Se1 est utilisé en comparant la différence entre les descripteurs des frames avec le seuil Se1. D'une autre part, si l'objectif est de détecter les transitions entre les shots, un seuil Se2 de valeur plus grande est utilisé en comparant la différence entre les descripteurs des frames. Si cette différence est supérieure au seuil Se2, on marque la présence d'une abrupte transition.

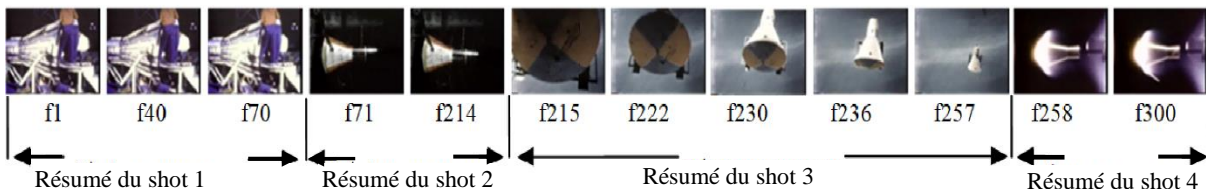


Figure 23 : un résumé de vue des 300 premiers frames de la vidéo anni003.mpg

Dans les premiers et quatrième groupes illustrés sur la figure 25 (de l'image 1 à l'image 70, et de l'image 258 à l'image 300). Toutes les images possèdent une proche int-

similarité l'une de l'autre, ce qui signifie qu'il y a une ressemblance significative entre les différentes images.

Dans le groupe 2 (de l'image 71 à l'image 214), certains points isolés sont détectés. Cela signifie qu'il existe une différence significative entre certaines consécutives similarités. Cela est due à une importante variation de la luminosité ou de la valeur de contraste entre deux frames consécutifs. Dans ce cas, si l'écart dépasse le seuil Se_2 , alors le réducteur peut vérifier s'il s'agit d'un point isolé ou d'une limite de shot correspondante à une abrupte transition.

Dans le groupe 3 (entre la 215ème et la 257ème frame), on observe une diminution des valeurs de la similarité intermédiaire avec un taux stable. Dans le cas de la vidéo de test actuellement utilisée, cela est dû à l'éloignement de l'objet au niveau du 3ème shot. Mais généralement, il peut aussi être causé par d'autres types de transition entre les shots.

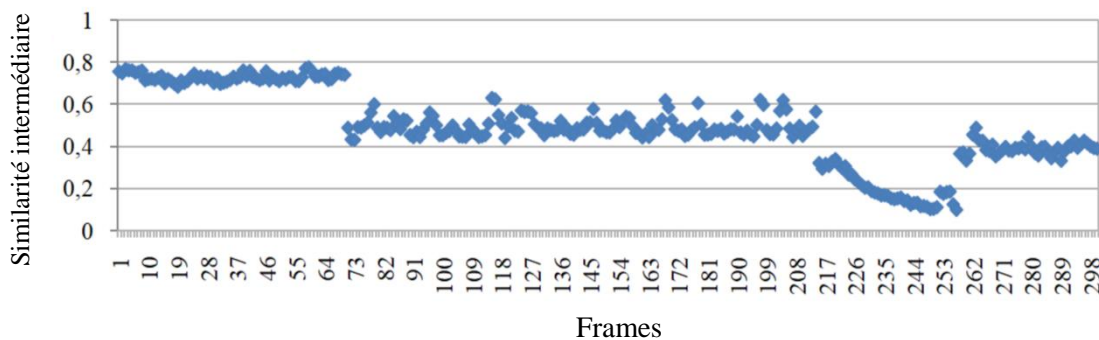


Figure 24: Variation de la similarité intermédiaire calculée sur un intervalle de 300 images de la vidéo Anni003.mpg.

Dans le premier groupe, chaque valeur de la similarité intermédiaire est comprise entre '0,68' et '0,77'. Mais, en passant vers l'image 71, qui représente la première limite du second shot, une valeur différente : '0,49' est observée. Cet important écart entre deux similarité intermédiaire mesurées signifie qu'il y a eu une grande variation dans les images consécutives, ce qui implique la présence a priori d'une transition abrupte.

6.3.3. Métriques d'évaluation de l'exactitude

Pour évaluer la performance de la méthode proposée, des mesures classiques de recherche d'information sont utilisées : la précision, le rappel et le F score définis comme suit :

$$\text{Précision} = \frac{\text{Correcte}}{\text{Correcte} + \text{faux}} \quad (1)$$

$$\text{Rappel} = \frac{\text{Correcte}}{\text{Correcte} + \text{manqué}} \quad (2)$$

$$F - \text{score} = \frac{2 * \text{Précision} * \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (3)$$

La précision est le nombre de limite de shot correctement détecté rapporté au nombre de limite de shot total trouvé y compris les fausses détections.

Le rappel est le ratio du nombre limite de shot correctement détecté par le nombre total de limite de shot total qui existe réellement.

Le F-score est une mesure de compromis entre précision et rappel. Elle combine la précision et le rappel est leur moyenne harmonique.

6.3.4. Comparaison de l'exactitude de la méthode proposée avec des solutions existantes

Dans cette section expérimentale, la performance de la méthode proposée est comparée à d'autres méthodes bien connues. Deux méthodes de références sont utilisées qui sont la différence de pixels (Luo et al., 2004) et l'approches basée sur la différence d'histogrammes des couleurs (Mas and Fernandez, 2003). Les Figures 25, 26 et 27 montrent respectivement les mesures de précision, de rappel et de F-score obtenues. Dans chaque figure, les mesures extraites de la méthode de VSBD proposée sont mises côte à côte avec les résultats des méthodes de référence. Les résultats montrent que la méthode proposée offre une exactitude de détection plus élevées par rapport aux autres méthodes. Cela la qualifier d'être utilisée par des application basées sur la segmentation temporelle dans un environnement distribué.

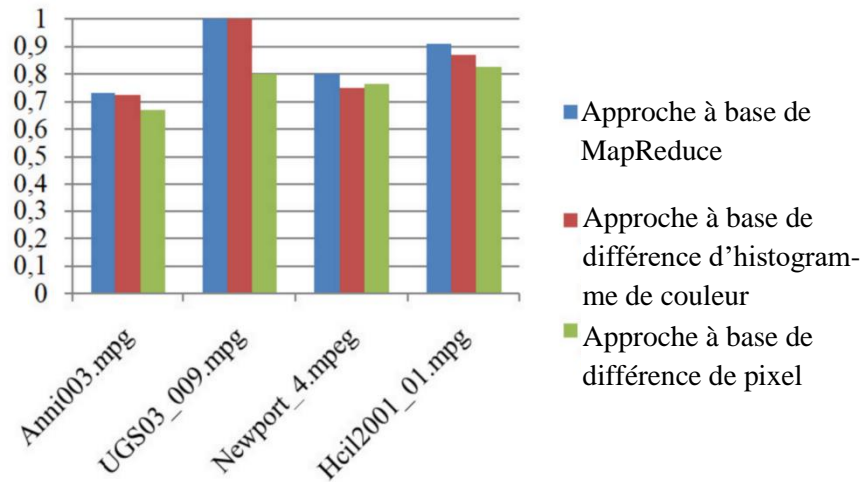


Figure 25: Comparaisons de la précision entre le schéma proposé et d'autres approches.

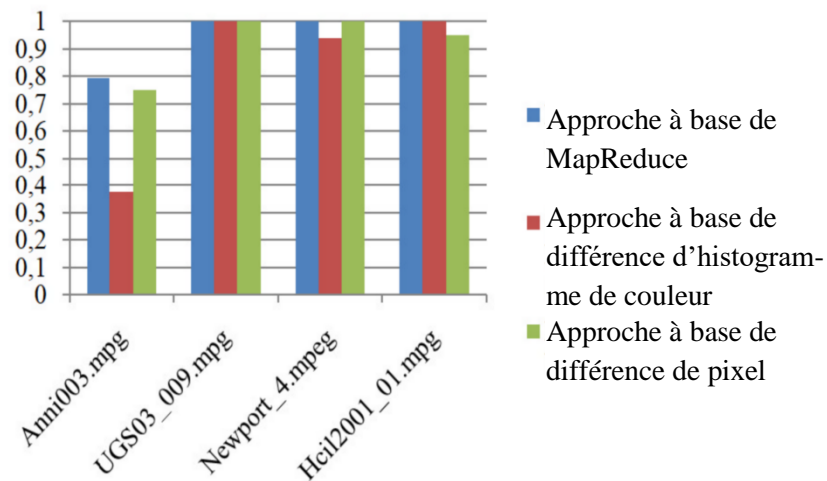


Figure 26: Comparaisons de rappel entre le schéma proposé et d'autres approches.

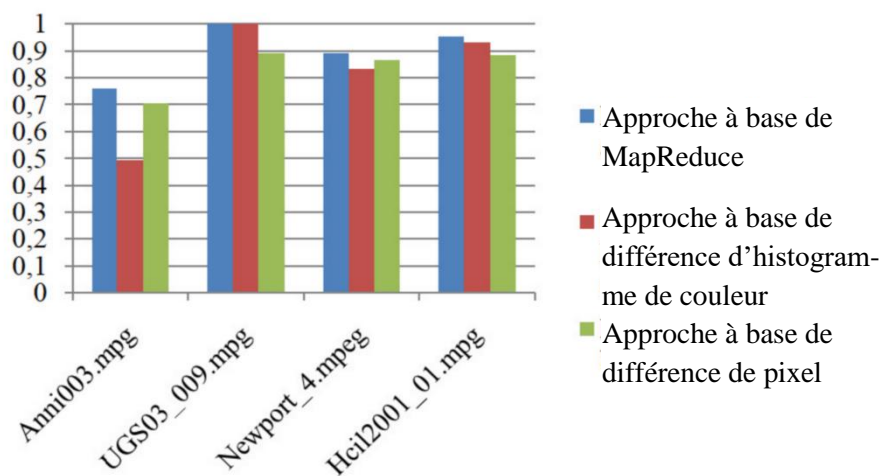


Figure 27: Comparaisons de F-score entre le schéma proposé et d'autres approches.

6.3.5. Test de la méthode VSBD proposée sur des données volumineuses

Dans cette section, on compare les résultats de la méthode VSBD basée sur le calcul des similarités intermédiaires avec une autre classique. Les deux méthodes appliquent MapReduce pour la distribution des traitements dans le même framework Hadoop et sur le même environnement.

La méthode de traitement VSBD classique, ou autres opérations multimédia procède comme suit :

- Diviser les données en blocs de taille fixe.
- Charger les données sur le HDFS.
- Appliquer l'opération Map et Reduce sur les données.
- Ecrire les résultats finaux sur le HDFS.

Les deux méthodes à comparer dans cette section suivent les mêmes étapes et implémentent les mêmes paramètres concernant les seuils à l'exception de l'utilisation de la similarité intermédiaire. Dans la méthode classique, un Job Map est lancé pour consommer un bloc de données. Il prend la vidéo frame par frame, et calcul la variation des descripteurs GIST pour extraire les shots du bloc de données pris en charge. Cette opération est effectuée sur tous les blocs des données et les résultats sont regroupés par les réducteurs et écrits sur le HDFS. Le tableau 4 montre les données vidéos sur lesquelles est appliquée la comparaison. Cependant, la figure 28 illustre F-score des résultats issus d'une part de la méthode qui implémente int-sim et d'autre part de la méthode classique. Les résultats F-score des deux méthodes sont mis côte à côte afin de bien visualiser la différence entre les deux.

Tableau 4. Données volumineuses utilisées pour le test de la méthode proposée.

Vidéo	Description	Durée	Taille	# d'abrupte transition
Vid1.mpg	Documentaire	2763 s.	725 MO.	52075
Vid2.mpg	Documentaire	2502 s.	561 MO.	32568
Vid3.mpg	Documentaire	5711 s.	1965 MO.	110267
Vid4.mpg	Documentaire	5438 s.	2306 MO.	81285

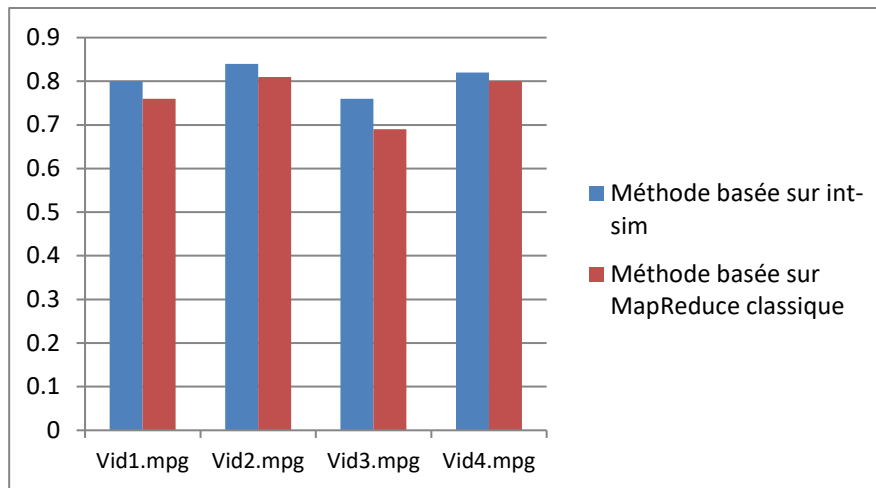


Figure 28. Comparaison de F-score entre la méthode proposée et une méthode basée sur MapReduce classique.

A partir de la figure 28, on trouve que l'exactitude de la méthode proposée dépasse celle qui utilise MapReduce de manière classique. La différence réside dans le traitement des bords des blocs. La méthode proposée prend les frames des blocs de manière aléatoire et les compare au niveau des réducteurs après un tri de leurs index. Cette manière de faire permet de traiter les bords des blocs comme n'importe quel autre frame avec les mêmes performances. De l'autre côté, les méthodes classiques prennent les frames des blocs et comparent entre eux de manière séquentielle. Les frames des bords des blocs ne sont pas traités car leurs successeurs se trouvent sur un autre bloc qui n'est pas accessible par la tâche Map en cours.

6.4.Évaluation de la scalabilité de la solution

Dans cette partie, on va présenter l'évaluation de scalabilité de la méthode proposée sur un ensemble de nœuds. Une mesure de temps d'exécution est étudiée afin de vérifier le passage à l'échelle de l'application et l'amélioration de ce temps d'exécution en variant le nombre de nœuds impliqués dans les traitements. L'évaluation est effectuée sur l'ensemble des vidéos de test du tableau 4.

Une application est dite scalable si les ressources physiques sont provisionnées de manière automatique en fonction des besoins de cette application. Si la propriété de scalabilité est vérifiée, on va trouver une amélioration de temps d'exécution en impliquant un nombre supplémentaire de ressources. Des outils de mesure de temps d'exécution et de

consommation de ressources peuvent être utilisés dans cette section. L'outil ganglia⁹⁷ est un logiciel de mesure de consommation de ressources tels que le CPU, la mémoire vive, l'énergie, etc. Ganglia est installé sur tous les nœuds de l'environnement distribué afin de calculer les statistiques demandées.

Les nœuds de l'environnement distribué sont des machines virtuelles avec des performances acceptables. Les machines virtuelles partagent les performances de la machine réelle décrites dans le tableau 1. Le tableau 5 montre les caractéristiques des machines virtuelles utilisées. Le framework Hadoop est installé en mode complètement distribué en spécifiant le nœud maître et les nœuds esclaves. La figure 29 montre la variation du temps d'exécution en fonction du nombre des nœuds impliqués dans les calculs. Le temps de copie et de réplication des données vidéo est négligé. Cependant, Les données sont divisées en blocs de 256 MO chacun et le nombre de blocs impliqués est varié.

Tableau 5. Caractéristique des nœuds de l'environnement.

Nom du nœud	Type	CPU	RAM
Machine 1	Maître/Esclave	1 cœur - 2.7 GHZ	4 GO
Machine 2	Esclave	1 cœur - 2.7 GHZ	4 GO
Machine 3	Esclave	1 cœur - 2.7 GHZ	4 GO
Machine 4	Esclave	1 cœur - 2.7 GHZ	4 GO

⁹⁷ Ganglia : outil de mesure de performance dans des systèmes distribués. L'outil est accessible sur le lien : <http://ganglia.sourceforge.net/>

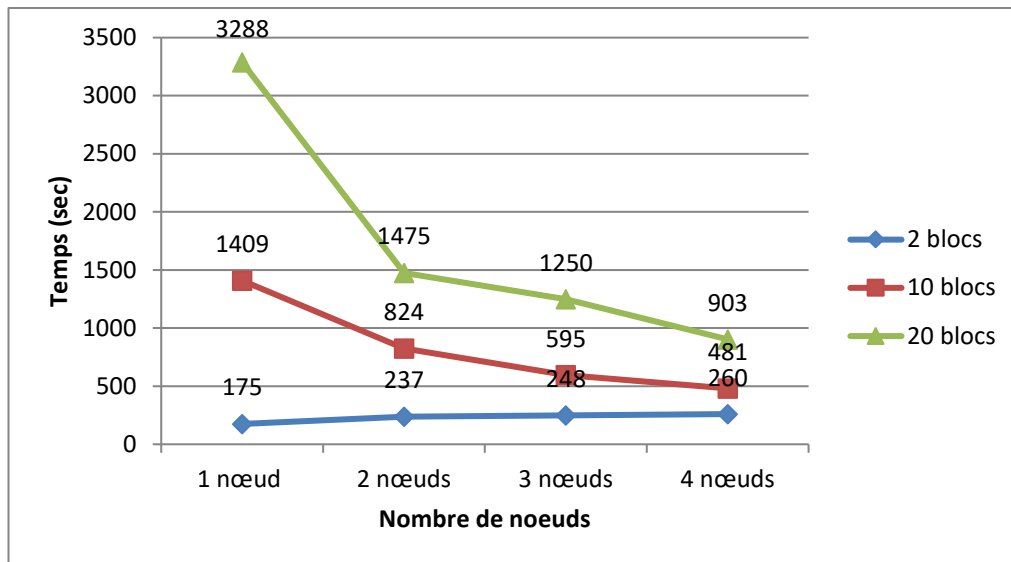


Figure 29. Variation du temps d'exécution en fonction du nombre de blocs et des noeuds impliqués dans les calculs.

La figure 29 montre bien que l'augmentation du nombre de ressources impliquées dans l'extraction des limites des shots permet de réduire le temps d'exécution moyen pour un nombre de blocs important. Par contre, l'ajout de plus de ressources à certain temps ne va plus améliorer le temps de calcul car celles déjà présentes fournissent les performances requises. Le graphe de temps d'exécution pour deux blocs en est un cas. Au contraire, on remarque une augmentation de ce temps avec l'implication de ressources supplémentaires ce qui est dû à la consommation du gestionnaire de l'environnement distribué.

7. Conclusion

Dans ce chapitre, on a présenté l'idée principale sur laquelle est basée la méthode de traitement de vidéo dans un environnement distribué. L'idée proposée est basée sur une relation intuitive, qui est la transitivité. Elle permet de rendre possible la comparaison entre deux éléments localisés sur des nœuds géographiquement distants. Cela a permis d'augmenter le taux d'exactitude et d'ajouter la notion de scalabilité aux méthodes déjà existantes pour le VSBD et les opérations en relation dans un environnement distribué.

L'environnement de test utilisé peut être largement amélioré avec la présence de réels machines performante à l'échelle de centaine de cœurs. Cela aidera à simuler un environnement distribué performant que ce soit pour les grilles ou pour les cloud pour tester le maximum de fonctionnalités. Cependant, des centres de calcul en Algérie fournissent des serveurs avec des gestionnaires de ressources spécifiques tels que Slurm pour soumettre les Jobs et ne permet pas la configuration personnalisée des sessions et des environnements.

8. Références

- Abdi, H., & Valentin, D. (2007). Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, 651-657.
- Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint, *Computer vision and pattern recognition (CVPR)*. 2012 IEEE conference on (pp. 510-517). Ieee.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. *European conference on computer vision* (pp. 404-417). Springer Berlin Heidelberg.
- Birinci, M., Kiranyaz, S. (2014) 'A perceptual scheme for fully automatic video shot boundary detection', *Signal Processing: Image Communication*, vol. 29, no 3, p. 410-423.
- Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., and Fua, P. (2012). BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7), 1281-1298.
- Cheng, S. T., Chen, Y. J., Wang, Y. T., Chen, C. F. (2013). Integration of MapReduce with an Interactive Boosting Mechanism for Image Background Subtraction in Cultural Sightseeing. *International Conference on Web-Based Learning*. Springer Berlin Heidelberg, p. 180-191.
- Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.
- Dean, J., & Ghemawat, S. (2010). MapReduce: a flexible data processing tool. *Communications of the ACM*, 53(1), 72-77.
- Dib, A., & Sellami, M. (2018). A Novel Solution for Scaling Video Shot Boundary Detection Based on Hadoop. *International Journal of Distributed Systems and Technologies (IJDST)*, 9(3), 39-52.
- Digital Digressions by Stuart Sierra. (2017). A Million Little Files'. [online] Available at: <http://stuartsierra.com/2008/04/24/a-million-little-files> [Accessed 25 Oct. 2017].
- Ffmpeg.org. (2017). FFmpeg. [online] Available at: <http://ffmpeg.org> [Accessed 25 Oct. 2017].
- Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., and Steele, D. (1995). Query by image and video content: The QBIC system. *Computer*, 28(9), 23-32.
- Gao, Y., Yong, J.H., Cheng., F.F. (2011) 'Video shot boundary detection using frame-skipping technique', [online] available on http://is.ifmo.ru/research/_video_shot_boundary_detection_using_frame-skipping_technique.pdf
- Geisler, G., and Marchionini, G. (2000). The open video project: research-oriented digital video repository. *Proceedings of the fifth ACM conference on Digital libraries*, 258-259, ACM.

Ghemawat, S., Gobioff, H., and Leung, S. T. (2003). The Google file system. *ACM SIGOPS operating systems review*, 37 (5), 29-43. ACM.

Gu, X., Zhang, Y., Zhang, L., Zhang, D., & Li, J. (2013). An improved method of locality sensitive hashing for indexing large-scale and high-dimensional features. *Signal Processing*, 93(8), 2244-2255.

Hare, J. S., Samangoei, S., & Dupplaw, D. P. (2011, November). OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th ACM international conference on Multimedia* (pp. 691-694). ACM.

Janwe, N. J., & Bhoyar, K. K. (2013, December). Video shot boundary detection based on JND color histogram. In *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on* (pp. 476-480). IEEE.

Lee, J. G., Kim, S., You, B. M., & Hwang, D. (2014). Parallel Design and Implementation of Shot Boundary Detection Algorithm. *Journal of the Institute of Electronics and Information Engineers*, 51(2), 76-84.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2548-2555). IEEE.

Li, W., Tong, X., Wang, T., Zhang, Y. Chen, Y.K. (2009a) 'Parallelization strategies and performance analysis of media mining applications on multi-core processors', *Journal of Signal Processing Systems*, vol. 57, no 2, p. 213-228.

Li, Y., Wang, S., Tian, Q., & Ding, X. (2015). A survey of recent advances in visual feature detection. *Neurocomputing*, 149, 736-751.

Li, Y.N., Lu, Z.M., Niu, X.M. (2009b) 'Fast video shot boundary detection framework employing pre-processing techniques', *IET image processing*, vol. 3, no 3, p. 121-134.

Ljubovic, V., & Supic, H. (2013, May). Comparative study of color histograms as global feature for image retrieval. In *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on* (pp. 1059-1063). IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). Ieee.

Luo, M., DeMenthon, D., & Doermann, D. (2004). Shot boundary detection using pixel-to-neighbor image differences in video. In *TRECVID 2004 Workshop Notebook Papers*.

Mas, J., & Fernandez, G. (2003). Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST*.

- Morsillo, N., Mann, G., Pal, C.,(2010) ‘Youtube scale, large vocabulary video annotation’, *Video Search and Mining. Springer Berlin Heidelberg*, p. 357-386.
- Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3), 145-175.
- Park, D. K., Jeon, Y. S., & Won, C. S. (2000, November). Efficient use of local edge histogram descriptor. *In Proceedings of the 2000 ACM workshops on Multimedia* (pp. 51-54). ACM.
- Patel, U., Shah, P., & Panchal, P. (2013). Shot detection using pixel wise difference with adaptive threshold and color histogram method in compressed and uncompressed video. *International Journal of Computer Applications*, 64(4).
- Puri, R., Prakash, A., & Tiwary, S. (2018). U.S. Patent Application No. 15/379,318.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. *In Computer Vision (ICCV), 2011 IEEE international conference on* (pp. 2564-2571). IEEE.
- Saleem, A., Chubbs, A., Gunn, N., Davidson, J., & Smirnov, S. (2011). Multimedia processing in parallel multi-core computation architectures. U.S. Patent No. 8,081,191. Washington, DC: U.S. Patent and Trademark Office.
- SenGupta, A., Thounaojam, D. M., Singh, K. M., Roy, S. (2015) ‘Video shot boundary detection: A review’, *IEEE International Conference on Electrical Computer and Communication Technologies (ICECCT) IEEE*, p. 1-6.
- Shekar, B. H., Uma, K. P., & Holla, K. R. (2016, September). Shot boundary detection using correlation based spectral residual saliency map. *In Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on* (pp. 2242-2247). IEEE.
- Shekhar, M. C. and Shukla, D. (2016). Detection of Scene Change in Video. *International Journal of Science and Research (IJSR)*, 5(3), pp.1187-1201.
- Sivic, J., & Zisserman, A. (2003, October). Video Google: A text retrieval approach to object matching in videos. *In Proceedings Ninth IEEE International Conference on Computer Vision* (p. 1470). IEEE.
- Sweeney, C., Liu, L., Arietta, S., & Lawrence, J. (2011). HIPI: a Hadoop image processing interface for image-based mapreduce tasks. Chris. University of Virginia.
- Toharia, P., Robles, O. D., Suárez, R., Bosque, J. L., & Pastor, L. (2012). Shot boundary detection using Zernike moments in multi-GPU multi-CPU architectures. *Journal of Parallel and Distributed Computing*, 72(9), 1127-1133.

Conclusions et perspectives

Conclusions et perspectives

Conclusions

L'utilisation des grilles de calcul est un outil puissant qui offre les ressources nécessaires pour l'accomplissement des tâches gourmandes telles que les applications de la vidéosurveillance. Néanmoins, le déploiement et l'interrogation de ces applications nécessitent de très grands efforts pour modifier et gérer des copies des instances des services sur différents conteneurs.

Une solution permettant une efficacité de déploiement dans les grilles, est la réutilisation des services de déploiement offerts par les frameworks des cloud. Cela fournit une solution souple à intégrer, conforme, et évolutive. L'intégration à ce niveau est réalisée de manière intuitive à partir des grilles en appelant le service à interroger qui est localisé sur des conteneurs virtualisés pour l'exécuter sur des conteneurs de ressources.

Les ressources fournies aux application multimédias par les services des grilles sont très **couteuses** vu leur réservation avec les meilleures performances pour une très longue durée. La réservation des ressources physiques avec la meilleure performance est due aux besoins de calculs intenses et aux besoins de réponse en temps réel pour quelques types d'opérations. Le besoin de réservation des ressources pour une longue durée est dû à la nature des applications multimédias qui est liée au contexte et à la raison de leur création.

Pour surmonter le problème du cout élevé de réservation de ressources pour les applications multimédias, on fait recours aux services des cloud qui permettent une facturation selon l'utilisation réelle des ressources. Cela est possible grâce aux caractéristiques de l'élasticité rapide et de la scalabilité qui offrent la possibilité de réserver, libérer et de réallouer les ressources selon la demande des clients et de leurs applications.

Cependant, les systèmes de vidéos surveillance effectuent des opérations de traitement d'images et de vidéos pour de diverses fonctionnalités. Parmi ces fonctionnalités, il y a ceux qui ne sont pas bien adaptées avec des environnements distribués telles que les

grilles de calcul et les cloud. Cela peut être résolu en étudiant ces fonctionnalités un par un, et proposant une solution adéquate comme celle de la segmentation temporelle.

Perspectives

Les futures études liées avec cet axe de recherche peuvent être classés en deux parties :

a- Perspectives liées aux systèmes distribués :

- 1- Créer des conteneurs standards intégrables aux systèmes distribués en général que ce soit cloud, grilles ou autres.
- 2- Unifier le langage de description des jobs afin de faire face à l'hétérogénéité des gestionnaires de ressources pour répondre aux besoins de performance de ces jobs.
- 3- Créer une couche de référence qui prend en charge les demandes de soumission des jobs sur les gestionnaires des ressources.

b- Perspectives liées aux traitements de vidéos dans des environnements distribués :

- 1- Etendre la technique basée sur la similarité intermédiaire sur d'autres fonctions de traitement de vidéo telle que la recherche, l'indexation, et la segmentation temporelle de différents types.
- 2- Adopter d'autres frameworks dédiés aux cloud, et qui offrent plus de performance de traitement telle qu'Apache Spark et Apache Storm.