Projet Fin d'étude 1ere version

Amini Makhlouf Ben Aissa Tinhinane

2020/2021

Table des matières

I État de l'art **5**

1 Big Data **6**
1.1 Historique et quelques statistiques sur le Big Data : 7
1.2 Définitions : 8
1.3 Intérêts du Big-Data : 9
1.4 Les contraintes du Big Data : 10
1.5 Caractéristiques des systèmes Big Data : 11
1.6 Choses qui viennent du Big Data (Exemples du Big Data) : 15
1.7 Les sources du Big Data : 16
1.7.1 Les données internes ou externes à l'entreprise 16
1.7.2 Les sources du Big Data selon leur provenances : 16
1.8 Les métiers du Big Data : 19
1.9 Les technologies du Big Data : 21
1.9.1 Les infrastructures : 21
1.9.2 Les technologies de traitements : 22
1.9.3 Les technologies de stockage : 26
1.10 Le future du Big Data 27

2 Le cloud computing : **29**
2.1 Définitions 29
2.2 Les composants clés du cloud computing : 30
2.2.1 Virtualisation : 30
2.2.2 Les Data Center : 30
2.3 Utilisations du Cloud Computing 32
2.4 Caractéristiques du Cloud 33
2.5 inconvénients du Cloud 33
2.6 Types de services Cloud 35
2.7 Types et modes de stockage dans le cloud computing : 37
2.7.1 Les types de stockage dans le cloud computing : 37
2.7.2 Les formats de stockage dans le cloud : 37

3 Les Objets connectés (IoT) : **39**
3.1 Définition : 40
3.2 L'histoire de l'IoT : 40
3.3 les composantes d'un réseau IoT : 41
3.4 Architecture de l'Internet des objets 41
3.5 La s´ecurité dans l'Internet des objets 41

3.6 Internet of Things : des applications pour tous ? 42
3.7 Le rôle de l'Internet des objets dans le Big Data 43

4 Le NoSql : 44
4.1 Rappel de gestion de base de données relationnelles : 45
4.1.1 Définition : 45
4.1.2 Les règles CODD : 46
4.1.3 Les contraintes des SGBDs relationnels (Propriétés ACID) : 47
4.1.4 Limite des bases de données relationnelles : 48
4.1.5 Exemples de bases de données relationnelles 50
4.2 Les bases de données NoSQL : 51
4.2.1 Définition : 51
4.2.2 Théorème de CAP : 52
4.2.3 Les propriétés de BASE : 53
4.2.4 Les types de bases de données NoSql : 54
4.2.5 Les avantages NoSQL : 57
4.2.6 Les inconvénients NoSQL : 58
4.2.7 Examples BDD NoSql : 59
4.3 Ver le NewSQL la base de données moderne 64
4.3.1 L'architecture NewSQL : 65
4.3.2 Les avantages de la solution NewSQL : 65
4.3.3 Les limites de la solution NewSQL : 65
4.3.4 Résumé : 66
4.3.5 Exemples des bases de données NewSQL : 66

Table des figures

1.1 1 minute d'Internet . . . 7
1.2 Le Big Data . . . 8
1.3 Les 3 V. . . . 11
1.4 Tableau évolution du chiffre d'affaires par région. . . . 14
1.5 Graphique évolution du chiffre d'affaires par région. . . . 14
1.6 Les 10 V. . . . 15
1.7 Les sources du Big Data . . . 16
1.8 Hadoop. . . . 22
1.9 Exemple de Word Count. . . . 23
1.10 Les composants d'Hadoop. . . . 24
1.11 Bases de données NoSql. . . . 25
1.12 Exemple de Word Count. . . . 27

2.1 La Virtualisation . . . 30
2.2 Exemple de data center (ceux de Facebook et Google) . . . 31
2.3 Architecture du data center . . . 31
2.4 Services cloud . . . 35

3.1 Internet of Things . . . 39

4.1 Éléments d'une table d'une BDDR. . . . 45
4.2 Problème lié aux propriétés ACID en milieu distribué . . . 48
4.3 Théorème de CAP . . . 52
4.4 ACID vs BASE . . . 53
4.5 base de donnée orientée clé-valeur . . . 54
4.6 base de donnée orientée colonne . . . 55
4.7 base de donnée orientée document . . . 55
4.8 base de donnée orientée graphe . . . 56
4.9 Logo Redis . . . 59
4.10 Logo Voldemort . . . 60
4.11 Logo HBASE . . . 60
4.12 Logo CASSANDRA . . . 61
4.13 Logo MONGODB . . . 61
4.14 Logo Neo4J . . . 62
4.15 Naissance du NewSQL à partir de 3 architectures . . . 64
4.16 L'architecture d'une base de données NewSQL populaire NueDB. . . . 65
4.17 Logo NuoDB. . . . 66
4.18 Logo VoltDB. . . . 66

4.19 Logo Clustrix. 66

Première partie

État de l'art

Chapitre 1

Big Data

Introduction

Avec la mise en place des services en ligne grâce à l'utilisation extensive d'Internet, le nombre de données générées qui transitent chaque jour sur le web, n'a fait que s'accroître de manière exponentielle, on parle ici de plus de 2,5 trillions d'octets générés quotidiennement, soit plus de 29.000 Giga-octets (Go) d'informations qui sont publiées dans le monde chaque seconde.

Ses données qui sont non pas que volumineuses mais aussi hétérogènes, viennent de toute part, la majeure partie de ses dernières proviennent de trois sources principales : les données sociales(les likes, les commentaires, les tweets, les photos/vidéos. . . etc), les données machines (Les capteurs tels que les appareils médicaux, les caméras routières, les satellites, les jeux et l'Internet des objets fournissent des données à haute vitesse, valeur, volume et variété) et les données transactionnelles (générées à partir de toutes les transactions quotidiennes qui ont lieu à la fois en ligne et hors ligne. Les factures, les ordres de paiement, les enregistrements de stockage, les reçus de livraison . . . etc).

Ses données qui sont utilisées par prêt de 6 milliards d'individus chaque jour, doivent être capturées, analysées, stockées, recherchées, partagées, visualisées, et transférées tout cela sans atteinte à la vie privée des utilisateurs, ce qui a poussé les chercheurs à trouver de nouvelles manières de réaliser tout cela étant donné que les outils traditionnels tels que le système de gestion de base de données relationnelles (SGBDR) et le SQL se retrouve dans l'incapacité de gérer ce nombre important et hétérogènes de données, et c'est ainsi qu'est né le **"Big Data"**.

En effet, comme chaque domaine de connaissance, la terminologie naissante "Big Data" et la science des données sont utilisées pour parler de ce phénomène, Nous allons lors de ce chapitre présenter les concepts et les définitions se rapportant au domaine du "Big Data" quelques statistiques ainsi que, les intérêts, contraintes et caractéristiques de ce dernier.

1.1 Historique et quelques statistiques sur le Big Data :

L'expression «Big Data» serait apparue en octobre 1997 selon les archives de la bibliothèque numérique de l'Association for Computing Machinery (ACM), dans un article scientifique sur les défis technologiques à relever pour visualiser les «grands ensembles de données».

Il apparaît depuis fréquemment dans la presse et dans les revues universitaires, et des programmes de «Data Science» ont vu le jour dans le monde universitaire au cours des six dernières années. Le 29 mars 2012, WHOSTP a annoncé la "Big Data Research and Development Initiative" qui s'appuie sur des initiatives fédérales "allant de l'architecture informatique et des technologies de mise en réseau aux algorithmes, à la gestion des données, à l'intelligence artificielle, apprentissage automatique, développement et déploiement de cyber infrastructures avancées".

Au cours des six dernières années, au moins 17 programmes de science des données ont commencé dans les principales universités de recherche américaines et Internet regorge de publicités pour des livres et des cours de science des données.

Selon l'étude Data Age 2025, la sphère de données mondiale passera de 33 zettaoctets en 2018 à 175 Zo d'ici 2025. Près de 30% des données mondiales devront être traitées en temps réel et le stockage réalisé sur le Cloud public représentera 49% du volume total de données.

Pour ce qui est des statistiques le moins qu'on puisse c'est qu'elles sont impressionnants voici une figure qui permet de représenter la quantité de données générée en 60 secondes d'Internet en 2020 *(Les données relevées ont été reprises de la compagnie Domo qui les a elle-même synthétisées à partir de nombreuses sources hétérogènes comme Business Insider, le New York Times, The Verge ou bien encore Hootsuite parmi d'autres et résumé par le site Visual Capitalist)* :

FIGURE 1.1 – 1 minute d'Internet

En analysant la **figure 1.1** on constate l'énorme quantité de données qui circule. En effet, durant cette minute sur internet alors que les utilisateurs de Facebook publient 147 000 photos, ceux d'Instagram partagent 347 222 stories et Twitter attire 319 nouveaux abonnés. Les plateformes de streaming ne sont pas en reste avec le SVOD Netflix notamment où les utilisateurs visionnent plus de 400 000 heures de vidéo en l'espace de 60 secondes et durant ce même laps de temps, 500 heures de vidéo sont publiées sur Youtube.

1.2 Définitions :

Mais alors qu'est-ce que le Big Data ?

Plusieurs définitions peuvent être données au Big Data, étant un objet complexe polymorphe, sa définition varie. Parmi elles nous citons :

Définition 1 : Le Big Data désigne l'ensemble des données numériques produites par l'utilisation des nouvelles technologies à des fins personnelles ou professionnelles. Cela regroupe les données d'entreprise ? Des contenus publiés sur le web, des transactions de commerce électronique, des échanges sur les réseaux sociaux, des données transmises par les objets connectés des données géolocalisées, ...etc.

Définition 2 : Le "Big Data" désigne les technologies et les initiatives qui impliquent des données trop diverses, en évolution rapide ou massives pour que les technologies, les compétences et les infrastructures conventionnelles puissent être traitées efficacement. Autrement dit, le volume, la vitesse ou la variété des données est trop important.

Définition 3 : Le terme Big Data fait référence aux données dont le coût de stockage, de gestion et d'analyse dans des systèmes de base de données traditionnels (relationnels et/ou monolithiques) serait généralement trop élevé. Habituellement, ces systèmes ne sont pas rentables, car ils ne disposent pas de la flexibilité nécessaire pour stocker des données non structurées (comme des images, du texte et des vidéos), pour accommoder des données "à haute vélocité" (en temps réel) ou pour s'adapter automatiquement à de très gros volumes de données (de l'ordre du pétaoctet).

FIGURE 1.2 – Le Big Data

1.3 Intérêts du Big-Data :

Dans tous les secteurs, les entreprises utilisent le Big Data engrangé dans leurs systèmes à différentes fins. Il peut s'agir d'améliorer les opérations, de proposer un meilleur service client, de créer des campagnes marketing personnalisées basées sur les préférences des consommateurs, ou tout simplement d'augmenter le chiffre d'affaires.

Grâce au Big Data, les entreprises peuvent profiter d'un avantage compétitif face à leurs concurrents n'exploitant pas les données. Elles peuvent prendre des décisions plus rapides et plus précises, s'appuyant directement sur les informations.

Par exemple, une entreprise peut analyser le Big Data pour découvrir de précieuses informations sur les besoins et les attentes de ses clients. Ces informations peuvent ensuite être exploitées pour créer de nouveaux produits ou des campagnes marketing ciblées afin d'accroître la fidélité client ou d'augmenter le taux de conversion. Une entreprise s'appuyant totalement sur les données pour aiguiller son évolution est qualifiée de " data-driven " (dirigée par les données).

On peut citer comme exemple : *Netflix, en effet En 2015, la lettre envoyée par Netflix à ses actionnaires a démontré que la stratégie Big Data portait ses fruits. Au premier trimestre 2015, 4,9 millions de nouveaux abonnés ont été enregistrés, contre quatre millions à la même période en 2014. De même, 10 milliards d'heures de contenu ont été diffusées pendant ce trimestre. Grâce à une utilisation intelligente du Big Data, l'influence de Netflix ne cesse de s'accroître.*

En outre, le Big Data est utilisé dans le domaine de la recherche médicale. Il permet notamment d'identifier des facteurs de risque de maladies, ou de réaliser des diagnostics plus fiables et plus précis. Les données médicales permettent aussi d'anticiper et de suivre les éventuelles épidémies.

Les mégadonnées sont utilisées dans presque tous les secteurs sans exception. L'industrie de l'énergie s'en sert pour découvrir des zones de forage potentielles et surveiller leurs opérations ou le réseau électrique. Les services financiers l'utilisent pour gérer les risques et analyser les données du marché en temps réel.

Les fabricants et les entreprises de transport, quant à eux, gèrent leurs chaînes logistiques et optimisent leurs itinéraires de livraison grâce aux données. De même, les gouvernements exploitent le Big Data pour la prévention du crime ou pour les initiatives de Smart City.

pour résumer, Le Big Data permet de construire de meilleurs modèles, qui produisent des résultats plus précis avec des approches extrêmement innovantes concernant la manière dont :

- Les entreprises se commercialisent et vendent leurs produits.
- La gestion des ressources humaines.
- La réaction au catastrophes naturelles.

Ces exemples ne sont finalement qu'une poignée des opportunités qu'offre le Big Data. Les entreprises, et pas seulement, devront faire preuve d'imagination, d'organisation et d'un énorme sens d'analyse pour prendre la pleine mesure du phénomène. De cette maîtrise découle de nouveaux usages qui bouleverse notre façon de concevoir Internet.

1.4 Les contraintes du Big Data :

L'intérêt du Big Data, c'est de pouvoir tirer profit de nouvelles données produites par tous les acteurs (les entreprises, les particuliers, les scientifiques et les institutions publiques) dans le but d'optimiser son offre commerciale, ses services, développer la recherche et le développement mais aussi créer des emplois. Il y a certes des avantages mais aussi des inconvénients du Big Data.

Certaines publications discutent des obstacles au développement d'applications de méga données. Les principaux défis sont énumérés comme suit :

❶ **Représentation des données :** De nombreux ensembles de données présentent certains niveaux d'hétérogénéité dans le type, la structure, la sémantique, l'organisation, la granularité et l'accessibilité. La représentation des données vise à rendre les données plus significatives pour l'analyse informatique et l'interprétation des utilisateurs. Néanmoins, une représentation incorrecte des données réduira la valeur des données originales et peut même empêcher une analyse efficace des données.

❷ **Réduction de la redondance et compression des données :** En général, il existe un niveau élevé de redondance dans les jeux de données. La réduction de la redondance et la compression des données sont efficaces pour réduire le coût indirect de l'ensemble du système en partant du principe que les valeurs potentielles des données ne sont pas affectées. Par exemple, la plupart des données générées par les réseaux de capteurs sont hautement redondantes.

❸ **Gestion du cycle de vie des données :** Par rapport aux progrès relativement lents des systèmes de stockage, la détection et le calcul omniprésents génèrent des données à des taux et des échelles sans précédent. Nous sommes confrontés à de nombreux défis urgents, dont l'un est que le système de stockage actuel ne peut pas supporter des données aussi massives. De manière générale, les valeurs cachées dans le Big Data dépendent de la fraîcheur des données.

❹ **Mécanisme analytique :** Le système analytique des méga données traitera des masses de données hétérogènes dans un temps limité. Cependant, les SGBDR traditionnels sont strictement conçus avec un manque d'évolutivité et d'extensibilité, ce qui ne pourrait pas répondre aux exigences de performance. Les bases de données non relationnelles ont montré leurs avantages uniques dans le traitement des données non structurées et ont commencé à se généraliser dans l'analyse des méga données. Même ainsi, il existe encore quelques problèmes de bases de données non relationnelles dans leurs performances et applications particulières. Des recherches supplémentaires sont nécessaires sur la base de données en mémoire et des échantillons de données basés sur une analyse approximative.

❺ **Confidentialité des données :** La plupart des fournisseurs ou propriétaires de services de méga données ne pouvaient actuellement pas maintenir et analyser efficacement des ensembles de données aussi énormes en raison de leur capacité limitée. Ils doivent s'appuyer sur des professionnels ou des outils pour analyser ces données, ce qui augmente les risques potentiels pour la sécurité. Par exemple, l'ensemble de données transactionnelles comprend généralement un ensemble de données d'exploitation complètes pour piloter les processus métier clés. Ces données contiennent des détails et certaines informations sensibles telles que les numéros de carte de crédit.

❻ **Gestion de l'énergie :** La consommation d'énergie des systèmes informatiques a beaucoup attiré l'attention du point de vue économique et environnemental. Avec l'augmentation du volume de données et des demandes analytiques, le traitement,

le stockage et la transmission de données massives consommeront inévitablement de plus en plus d'énergie électrique.

❼ **Expendabilité et évolutivité :** Le système analytique du Big Data doit prendre en charge les ensembles de données présents et futurs. L'algorithme analytique doit être capable de traiter des ensembles de données de plus en plus étendus et plus complexes.

❽ **Coopération :** L'analyse du Big Data est une recherche interdisciplinaire, qui nécessite la coopération d'experts dans différents domaines pour exploiter le potentiel du Big Data. Une architecture de réseau Big Data complète doit être mise en place pour aider les scientifiques et les ingénieurs dans divers domaines à accéder à différents types de données et à utiliser pleinement leur expertise, afin de coopérer pour atteindre les objectifs analytiques.

1.5 Caractéristiques des systèmes Big Data :

Les méga-données sont un terme générique utilisé pour désigner toute collection de données volumineuse et complexe qui peuvent dépasser la capacité de traitement des systèmes et techniques de gestion de données conventionnels. Les applications du Big Data sont infinies.

Les méga-données sont souvent caractérisées par le volume extrême des données, la grande variété de types de données et la vitesse à laquelle les données doivent être traitées. (Ces caractéristiques sont dites les 3V)

Ces caractéristiques ont été identifié pour la première fois par l'analyste Douglas Laney's membre du Gartner 10 dans un rapport publié en 2001. Plus récemment, plusieurs autres caractéristiques (autres V) ont été ajoutées aux descriptions des méga-données, notamment la véracité, la valeur et la variabilité. Bien que les méga-données ne correspondent à aucun volume de données spécifique, le terme est souvent utilisé pour décrire des téraoctets, des péta-octets et même des exa-octets de données capturées au fil du temps.

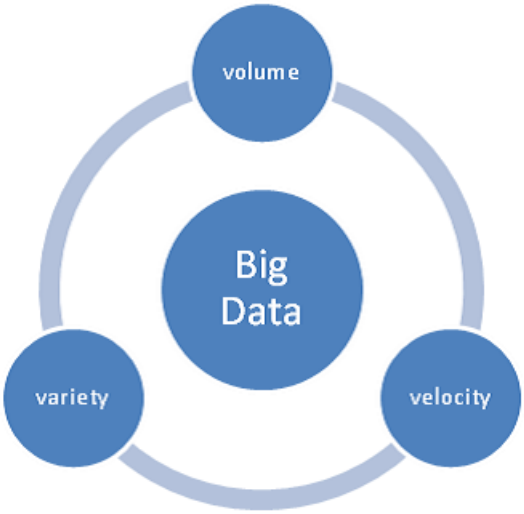

FIGURE 1.3 – Les 3 V.

Certaines personnes attribuent encore plus de V aux Big Data, les data scientiste et les consultants ont créé diverses listes contenant entre sept et 10 V. On donne dans ce qui suit 10 caractéristiques "10V" sur les méga-données sachant que Les 3 premier critères basiques du Big Data sont le volume la vitesse ainsi que la variété :

1. Le Volume :

Le caractère volume est certainement celui qui est le mieux décrit par le terme Big de l'expression Big Data. Volume fait référence à la quantité d'informations, trop volumineuse pour être acquise, stockée, traitée, analysée et diffusée par des outils standards. Ce caractère peut s'interpréter comme le traitement d'objets informationnels de grande taille ou de grandes collections d'objets.

Exemple : *les utilisateurs d'Instagram partagent 347 222 stories en 60secondes.*

2. La Variété :

Elle fait référence aux différentes formes toujours croissantes que les données peuvent prendre, en effet les données Big Data ne sont pas seulement des nombres, des dates et des chaînes. Les méga-données englobent également une grande variété de types de données, y compris des données structurées dans des bases de données SQL et des entrepôts de données, des données non structurées, telles que des fichiers texte et document conservés dans des clusters Hadoop ou des systèmes NoSQL, et des données semi-structurées, telles que des journaux de serveur Web ou la diffusion des données à partir de capteurs.

Exemple : *Un projet d'analyse des méga-données peut tenter d'évaluer le succès d'un produit et les ventes futures en corrélant les données de ventes passées, les données de retour et les données de révision des acheteurs en ligne pour ce produit.*

3. La vélocité :

Dernière dimension, tout aussi importante que les précédentes, la vélocité traduit la capacité à produire rapidement les données et à les transformer en temps utile pour leurs utilisateurs. L'exercice, déjà difficile dans un contexte "classique", prend toute sa valeur lorsqu'il doit être applique à d'immenses volumes de données de toutes sortes.

Exemple : *Google traite en moyenne plus de "40 000 requetés de recherche par seconde", ce qui représente environ 3,5 milliards de recherches par jour.*

4. La Véracité :

Elle fait référence aux biais, au bruit et aux anomalies dans les données. Ou, mieux encore, il fait référence aux incertitudes et à la fiabilité des données souvent incommensurables.

Exemple : *Dans le cadre d'un sondage réalisé par IBM, 27% des entreprises interrogées avouent ne pas être certaines de l'exactitude des données qu'elles collectent. De même, un chef d'entreprise sur trois utilise les données pour prendre des décisions, mais n'a pas vraiment confiance. Ce manque de véracité et de qualité des données coûte environ 3,1 trillions de dollars par an aux États-Unis.*

5. La Valeur :

Toutes les données collectées n'ont pas une valeur commerciale réelle et l'utilisation de données inexactes peut affaiblir les informations fournies par les applications d'analyse. Il est essentiel que les organisations utilisent des pratiques telles que le nettoyage des données et confirment que les données sont liées à des problèmes commerciaux pertinents avant de les utiliser dans un projet d'analyse de Big Data.

On peut dire que les autres caractéristiques du Big Data n'ont pas de sens si on ne tire pas de valeur commerciale de ces données. Les Données massives offrent une valeur substantielle : comprendre mieux les clients. Les cibler en conséquence, optimiser les processus

et améliorer les performances de la machine ou de l'entreprise. Avant de se lancer dans une stratégie Big Data, on doit comprendre le potentiel et les caractéristiques les plus difficiles.

Exemple : *La mise en place d'une analyse Big Data a permis à la société de développement d'éoliennes Vestas 11 d'optimiser son processus d'identification des meilleurs emplacements pour implanter ses éoliennes. Le traitement Big Data a engendré une augmentation de la performance de production d'électricité et une réduction des coûts énergétiques associés.*

Remarque : Certaines personnes attribuent encore plus de V aux Big Data ; les scientifiques des données et les consultants ont créé d'autres listes en ajoutant la variabilité, la validité, la visualisation, la volatilité ainsi que la vulnérabilité.

6. La Variabilité :

La variabilité dans le Big Data fait référence à plusieurs sens. Dans un premier temps elle désigne le nombre d'incohérences dans les données. Celles-ci doivent être détectées par des techniques de détection d'anomalies et de valeurs aberrantes pour faciliter la création d'analyse significative.

Les méga-données sont également variables en raison de la diversité de dimensions résultant de multiples types et sources de données. La variabilité peut également faire référence à la vitesse incohérente à laquelle les données volumineuses sont chargées dans la base de données.

Exemple : *L'équipe d'IBM 12 fait participer Watson 13 au célèbre jeu télévisé américain Jeopardy, un jeu ou les candidats doivent trouver les réponses à des questions posées. Watson devait "être capable de comprendre l'énoncé des questions, buzzer pour prendre la main, disséquer une réponse dans son sens pour déterminer quelle était la bonne question". Les mots n'ont pas de définitions statiques et leur signification peut varier énormément dans le contexte.*

7. La Validité :

Similaire à la véracité, la validité fait référence à la précision et à la correction des données pour l'usage auquel elles sont destinées. Selon Forbes 14, environ 60% du temps d'un scientifique est consacré au nettoyage de ses données avant de pouvoir effectuer une analyse. L'avantage de l'analyse des données massives est aussi primordial que celui des données sous-jacentes. On doit donc avoir de bonnes pratiques de gouvernance des données pour garantir une qualité des données cohérente, des définitions communes et des métadonnées.

Exemple : *La date d'une transaction est 02/07/1994 alors que l'activité de la société a débuté en 2000.*

8. La Volatilité :

On se pose les questions : *'quel âge doivent avoir les données pour qu'elles soient considérées comme non pertinentes, historiques ou obsolète ?', 'Combien de temps faut-il conserver les données ?'* Avant l'ère du Big Data, en général, on stockait les données indéfiniment. Quelques téraoctets de données ne pouvaient pas engendrer de dépenses de stockage élevées.

En raison de la vitesse et du volume de ces données massives, leur volatilité doit être soigneusement prise en compte. Il est maintenant fondamental d'établir des règles pour la disponibilité et à la mise à jour des données a de garantir une récupération rapide des informations en cas de besoin.

Exemple : *Une entreprise e-commerce peut ne pas souhaiter conserver un historique des achats client d'un an. Parce qu'après un an la garantie par défaut sur leur produit expire, il n'y a donc aucune possibilité de restaurer ces données.*

9. La Visualisation :

Une autre caractéristique du Big Data est la difficulté à les visualiser. Les logiciels de visualisation de données volumineuses actuels sont confrontés à des problèmes techniques en raison des limitations de la technologie en mémoire, de leur faible évolutivité, de leur fonctionnalité et de leur temps de réponse. Il est impossible de se fier aux graphiques traditionnels lorsqu'on essaye de tracer un milliard de points de données. Il est donc nécessaire d'avoir différentes manières de représenter des données. Telles que la mise en cluster de données ou l'utilisation de cartes d'arbres, de sunbursts, de coordonnées parallèles, de diagrammes de réseau circulaires ou de cônes.

Si on associe cela avec la multitude de composante résultant de la variété et de la vélocité des données massives et des relations complexes qui les lient, il est possible de voir qu'il n'est pas si simple de créer une visualisation significative.

Prenons l'exemple : *du tableau suivant qui fait apparaître deux séries de chiffres : le chiffre d'affaires en France et le chiffre d'affaires du reste du monde. La lecture de ce tableau et sa signification ne sont pas immédiates.*

kEur	Janv	Fevr	Mars	Avril	Mai	Juin	Juillet	Aout	Sept	Oct	Nov	Déc
France	10000	12000	14000	13000	15444	17028	15000	15804	18000	17500	19000	20958
Reste du monde	3444	3816	4038	3558	3864	4074	3558	2000	3594	3498	3612	4140
	13444	15816	18038	18558	19308	21102	18558	17804	21594	20998	22812	25098

FIGURE 1.4 – Tableau évolution du chiffre d'affaires par région.

Mais si nous représentons les séries de chiffres sous forme graphique (ci-dessous), on comprend en un coup d'œil que le chiffre d'affaires en France progresse et que le chiffre d'affaires du Reste du monde stagne.

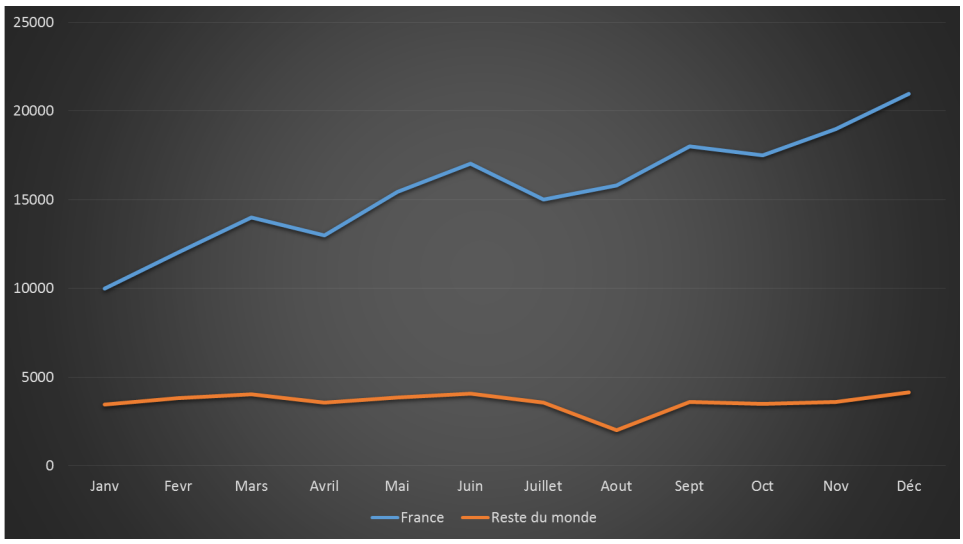

FIGURE 1.5 – Graphique évolution du chiffre d'affaires par région.

10. La Vulnérabilité :

Le Big Data apporte de nouveaux problèmes de sécurité. Malheureusement, il y a quotidiennement des violations de données massives.

Exemple : *Rapporté par CRN15 : en mai 2016, un pirate informatique appelé Peace a posté des données sur le dark web pour les vendre, qui auraient inclus des informations sur 167 millions de comptes LinkedIn et 360 millions d'e-mails et de mots de passe pour les utilisateurs de MySpace 16.*

FIGURE 1.6 – Les 10 V.

Remarque : Il existe d'autres sources qui parlent de 54 Vs pour les caractéristiques du Big Data tel que la venue (le lieu), valence, vocabulaire, imprécision...etc.

1.6 Choses qui viennent du Big Data (Exemples du Big Data) :

Le concept de big data est une gestion groupée de différentes formes de données générées par divers appareils (Android, iOS, etc.), d'applications (applications musicales, applications Web, applications de jeux, etc.) ou d'actions (recherche par SE, navigation à travers des types similaires de pages Web, etc.). Voici la liste de certains champs de données couramment trouvés qui sont sous l'égide du Big Data :

Données sur les boîtes noires : Les données des boîtes noires sont un type de données recueillies à partir d'hélicoptères, d'avions et de jets privés et gouvernementaux. Ces données comprennent la capture des sons de l'équipage de conduite, l'enregistrement séparé du microphone ainsi que des écouteurs, etc.

Données boursières : Les données boursières comprennent diverses données préparées sur « l'achat » et la « vente » de différentes décisions brutes et bien prises.

Données sur les médias sociaux : Ce type de données contient des informations sur les activités des médias sociaux qui incluent des messages soumis par des millions de personnes dans le monde entier.

Données sur les transports : Les données sur les transports comprennent les modèles de véhicules, la capacité, la distance (d'une source à l'autre) et la disponibilité de différents véhicules.

Données des moteurs de recherche : récupérez une grande variété d'informations non traitées stockées dans les bases de données SE.

1.7 Les sources du Big Data :

Pour réussir avec le Big Data, il est important que les entreprises disposent du savoir-faire pour passer en revue les différentes sources de données disponibles et classer en conséquence leur convivialité et leur pertinence. Deux classifications des sources du Big Data sont données dans ce qui suit :

1. Selon qu'elles soient internes ou externe à l'entreprise.
2. Selon leur provenance.

1.7.1 Les données internes ou externes à l'entreprise

Les données sont internes si une entreprise les génère, les possède et les contrôle.
Exemple :

- Module ERP d'entreprise.
- Capteurs, contrôleurs.
- Centres d'appels internes.
- Journaux de site Web.

Les données externes sont des données publiques ou des données générées en dehors de l'entreprise ; en conséquence, la société ne la possède ni ne la contrôle. Exemple :

- Médias sociaux.
- Statistiques officielles.
- Ensembles de données accessibles au public pour l'apprentissage automatique.

1.7.2 Les sources du Big Data selon leur provenances :

Les données les plus volumineuses sont utilisées aujourd'hui par les organisations et les entreprises dans le seul but d'effectuer des analyses. Toutefois, avant de pouvoir extraire des informations et des renseignements précieux à partir de données importantes, ces dernières doivent connaître plusieurs sources de données importantes. Les données, comme nous le savons, sont massives et existent sous diverses formes. Si elles ne sont pas bien classées ou si elles ne proviennent pas d'une source sure, elles peuvent finir par faire perdre du temps et des ressources précieuses. Afin de réussir avec les données volumineuses, il est important que les entreprises aient le savoir-faire nécessaire pour faire le tri entre les différentes sources de données disponibles et classer en conséquence leur utilité et leur pertinence.

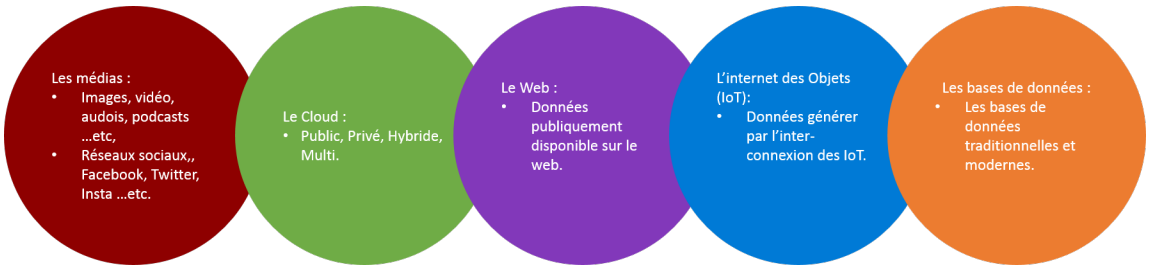

FIGURE 1.7 – Les sources du Big Data

Les médias :

Les médias sont la source la plus populaire de méga données, car ils fournissent des informations précieuses sur les préférences des consommateurs et l'évolution des tendances. Ces données sociales proviennent des mentions J'aime, des tweets et des retweets, des commentaires, des téléchargements de vidéos et des médias généraux qui sont téléchargés et partagés via les plateformes de médias sociaux préférées au monde. Ce type de données fournit des informations inestimables sur le comportement et le sentiment des consommateurs et peut être extrêmement influent dans les analyses marketing.

Exemple :*Les médias comprennent les médias sociaux et les plates-formes interactives, telles que Google, Facebook, Twitter, YouTube, Instagram, ainsi que des supports génériques tels que des images, des vidéos, des audio et des podcasts qui fournissent des informations sur l'interaction utilisateur.*

Le Cloud :

Aujourd'hui, avec la croissance accélérée du volume de données utilisé par les applications, de nombreuses organisations ont déplacé leurs données vers des serveurs Cloud pour fournir des services évolutifs, ables et hautement disponibles [12].

Le stockage Cloud héberge des données structurées et non structurées et fournit aux entreprises des informations en temps réel et des informations à la demande. Le principal attribut du Cloud computing est sa flexibilité et son évolutivité. Comme les méga données peuvent être stockées et sourcées sur des Clouds publics ou privés, via des réseaux et des serveurs, le Cloud constitue une source de données efficace et économique car les ressources sont fournis à la demande et on ne paye que les ressources utilisées.

Le Web :

Le Web public constitue une source répandu et facilement accessible pour le Big Data. Les données sur le Web ou "Internet" sont généralement disponibles pour les particuliers et les entreprises. De plus, des services Web tels que Wikipédia fournissent des informations gratuites et rapides à tout le monde. Le Web public est également une autre bonne source de données sociales, et des outils comme Google Trends peuvent être utilisés à bon escient pour augmenter le volume de données.

L'énormité du Web garantit sa polyvalence et est particulièrement bénéfique pour les start-ups et les PME, car elles n'ont pas à attendre pour développer leur propre infrastructure et référentiels de Big Data avant de pouvoir tirer parti du Big Data.

L'Internet Des Objets (IoT) :

Les données machine sont définies comme des informations générées par des équipements industriels, des capteurs installés dans des machines et même des journaux Web qui suivent le comportement de l'utilisateur. En logistique, il peut s'agir de capteurs qui servent à la traçabilité des biens pour la gestion des stocks et les acheminements.

En domotique, l'IoT recouvre tous les appareils électroménagers communicants, les capteurs, les compteurs intelligents et systèmes de sécurité connectés des appareils de type box domotique. Le phénomène IoT est également très visible dans le domaine de la santé et du bien-être avec le développement des montres connectées, des bracelets connectés et d'autres capteurs surveillant des constantes vitales.

Exemple : *En Tennis, la marque Zepp propose un système de capteur connecté qui analyse vos performances sur le terrain. Il suffit de fixer le capteur sur le manche de n'importe*

quelle raquette de tennis et de se mouvoir pour obtenir un retour d'information instantané sur iPhone, iPad ou appareil Android.

Les bases de données :

Les entreprises préfèrent aujourd'hui utiliser une fusion de bases de données traditionnelles et modernes pour acquérir des méga données pertinentes. Cela permet d'ouvrir la voie à un modèle de données hybride avec un faible coût d'investissement. De plus, ces bases de données sont également déployées à plusieurs ns de Business Intelligence. Le processus d'extraction et d'analyse de données parmi de vastes sources de méga données est un processus complexe qui peut être frustrant. Ces complications peuvent être résolues si les organisations englobent toutes les considérations nécessaires des méga données, et prennent en compte les sources de données pertinentes et les déploient d'une manière bien adaptée à leurs objectifs organisationnels.

Exemple : *Les bases de données les plus populaires Microsoft SQL Server, MongoDB, Oracle, MySQL et IBM Db2 ...etc.*

1.8 Les métiers du Big Data :

Le marché du big data est à l'origine d'un nombre croissant de métiers Les entreprises de tous les secteurs cherchent désormais à exploiter les données à leur disposition pour aiguiller leur stratégie et leur développement. Toutefois, pour être en mesure d'exploiter ces données, les entreprises doivent s'appuyer sur des compétences et du savoir-faire de professionnels hautement qualifiés capables d'utiliser les technologies analytiques. Ainsi, le Big Data a donné naissance à de nombreux nouveaux métiers :

1. LE CHIEF DATA OFFICER

Le chief data officer se charge de gouverner la data, qui constitue un capital vital pour l'entreprise. Il a pour mission de trier les masses de données disponibles afin de faciliter l'accès à l'information pertinente permettant des prises de décision adaptées. Pour ce faire, il doit constamment vérifier la fiabilité des informations recueillies et s'appuyer sur des éléments objectifs provenant de données statistiques. Le chief data officer intervient dans la construction et la mise en application d'une stratégie de gouvernance des données et travaille en collaboration avec d'autres professionnels tels que les data scientists, les spécialistes en business intelligence et les statisticiens.

2. LE DATA ENGINEER

Le data engineer (ou ingénieur de données) est un professionnel spécialisé dans la gestion des données. Sa mission principale consiste à recueillir, croiser, trier et réaliser des opérations de nettoyage des données. Il doit aussi gérer leur stockage dans différentes bases de données et exploiter des masses d'information sous divers formats.

3. LE DATA SCIENTIST

Le data scientist, appelé également analyste en big data, est un spécialiste de l'analyse des données massives. Sa mission prend effet après celle de l'ingénieur de données : le data engineer intervient dans la gestion des données alors que le data scientist assure le traitement de ces données pour en extraire de la valeur. Pour ce faire, le data scientist se charge de développer des algorithmes statistiques afin de tirer des informations pertinentes permettant de classifier des données, d'anticiper un comportement ou encore de préconiser des actions appropriées. Il doit donc avoir de solides connaissances en informatique, en statistiques et en management. Le data scientist doit également maîtriser les techniques du datamining et les outils de traitement des bases de données tels que Hadoop, MapReduce, Java, BigTable et NoSQL.

Les analystes en big data interviennent dans divers domaines d'activité : ils développent dans l'e-commerce et les réseaux sociaux les algorithmes de recommandation de pages, de profils et de produits

4. L'ARCHITECTE BIG DATA

Data architect en anglais, l'architecte big data a pour mission principale d'organiser des données brutes. C'est un métier plus conceptuel que technique, qui assure la création des infrastructures de stockage et conçoit des solutions de gestion des données massives. Il propose également aux décideurs la cartographie des outils Hadoop à mettre en place. L'architecte big data travaille en étroite collaboration avec le data scientist, tout en lui

fournissant les données brutes à traiter. Il intervient également dans l'étude de la faisabilité technique et la mise en place des outils et la configuration des machines.

5. LE DÉVELOPPEUR BIG DATA

Le développeur big data maîtrise les différents langages informatiques notamment Java et Python. Il assure la cohérence du système, la gestion des pannes et garantit la continuité du service. Les données massives sont en effet au centre des préoccupations du métier de développeur big data. Ce profil travaille aussi en collaboration avec le data scientist : alors que ce dernier intervient dans la conception des algorithmes facilitant la prise de décision, le développeur big data assure leur mise en marche. Il fait partie des rares profils du big data à pouvoir gérer toutes les catégories des outils d'Hadoop pour des objectifs d'évaluation.

6. LE GROWTH HACKER

Le growth hacker n'est pas simplement un métier, c'est surtout un état d'esprit qui permet de développer plusieurs techniques webmarketing. C'est un profil à la croisée du marketing, du développement logiciel et du big data, qui a pour mission d'accélérer la croissance d'un produit ou d'un service propre à la structure qui l'embauche. Il utilise pour cela des solutions digitales innovantes et des pratiques de pointe afin d'accroître le revenu de son entreprise. Pour ce faire, le growth hacker cherche à développer, à partir d'Hadoop, de nouveaux produits et de nouvelles fonctionnalités. Il utilise également les outils de base de données (SQL) et les langages d'abstraction. De plus, comme tous les professionnels du marketing, il est en recherche constante de clients. Le growth hacker est très prisé par les start-up et les entreprises qui souhaitent se réinventer constamment.

7. LE DATA MINER

Le data miner assure la transmission des connaissances utiles à la progression de l'entreprise. Il dégage ainsi les tendances relatives à la consommation des clients pour en sortir avec une stratégie marketing réalisable sur le terrain. Pour se positionner sur le marché, il prend en compte les habitudes de consommation et les tarifs appliqués par la concurrence. De plus, il assure le tri des informations potentiellement exploitables, analyse les données après les avoir formatées et nettoyées. Il réalise aussi des rapports d'analyse, des tableaux de visualisation des données et compare les performances de l'entreprise pour les ajuster aux objectifs et prévisions. Le data miner a de grandes capacités d'observation et d'analyse de données.

8. L'ADMINISTRATEUR BIG DATA

L'administrateur joue un rôle primordial dans la structure informatique d'une entreprise. Il intervient dans la conception, l'optimisation et la configuration des infrastructures de stockage des données massives. Il assure également la sécurisation des données ainsi que l'attribution des autorisations et des droits d'accès aux différents utilisateurs. L'administrateur big data maîtrise les langages de programmation, les outils d'administration Hadoop et les protocoles de sécurité. Il vérifie la disponibilité de l'information à tout moment et apporte les modifications nécessaires sur les bases de données.

1.9 Les technologies du Big Data :

Cette technologie est importante pour présenter une analyse plus précise qui conduit l'analyste d'affaires à prendre des décisions très précises, assurant ainsi une efficacité opérationnelle plus considérable en réduisant les coûts et les risques commerciaux. Maintenant, pour implémenter de telles analyses et détenir une telle variété de données, il faut avoir besoin d'une infrastructure qui puisse faciliter et gérer et traiter d'énormes volumes de données en temps réel. De cette façon, le Big Data est classé en deux sous-catégories, le Big Data opérationnel qui comprend des données sur des systèmes et le Big Data analytique qui comprend des systèmes.

Nous décrivons ensuite ici tous les composants qui font partie des solutions Big Data sous de nombreux angles : matériel, méthodologies, logiciels et applications de base, etc.

Pour mieux catégoriser ces concepts, nous les avons a répartis en différentes sections selon l'objectif visé par chacun. Ces catégories sont : l'infrastructure, le stockage, le traitement et les composants de haut niveau.

1.9.1 Les infrastructures :

Le développement du Big Data commence avec les clusters Big Data qui exécutent en parallèle les instructions d'un logiciel de haut niveau. Le cluster est artitionné en deux types de nœuds selon la fonction principale exercée :

- Nœuds de données ou esclaves (informatique).
- Nœuds de gestion ou maîtres (gestion).

Outre leur fonction, le maître et les esclaves peuvent être différenciés par leurs capacités de calcul et leur quantité dans le champ de nœuds.

Les esclaves sont chargés de surveiller les données partitionnées, de traiter et d'interroger les données locales. Les unités de données et de traitement doivent être aussi proches que possible pour éviter les retards introduits par les mouvements entre les partitions. Les nœuds de données sont gourmands en disque et standard en termes de capacités de calcul et de mémoire.

Les maîtres reçoivent et transforment les programmes des applications clientes en instructions parallèles qui peuvent être comprises par les esclaves. Une fois que les applications clientes ont atteint le démon maître, elles finissent par démarrer ou réveiller plusieurs processus dans les esclaves qui retournent finalement une sortie suivant la direction opposée. Parmi l'ensemble des responsabilités approuvées pour les nœuds de gestion figurent :

- La récupération après défaillance.
- La gestion des ressources.
- La planification des travaux.
- La surveillance ou la sécurité.

Pour accomplir ces tâches, les maîtres nécessitent une puissance de calcul et de mémoire élevée. Dans les clusters Big Data standard, il suffit de garder deux maîtres supports qui se surveillent mutuellement.

Les deux types de nœuds sont connectés via une connexion réseau, généralement LAN (Ethernet ou InfiniBand). Certaines configurations permettent également de connecter les maîtres de déférents centres de données sur un réseau WAN pour éviter facilement les défaillances du système. Dans chaque centre de données, le maître et les esclaves sont interconnectés en privé pour ingérer des données, déplacer des données entre les nœuds et effectuer des requêtes. Il existe également un autre réseau public qui sert de façade entre le client et le service de gestion (SSH, VNC, interface web,..)

1.9.2 Les technologies de traitements :

FIGURE 1.8 – Hadoop.

Dans cette section nous parlerons de l'arrivée des technologies de traitement ajustées, plus spécialement sur la mise au point de modes de calcul à haute performance (MapReduce), nous parlerons de (Hadoop) une solution de Big Data très largement utilisée pour effectuer des analyses sur de très grands nombres de données.

❶ Map-reduce :

MapReduce a été introduit par Google et décrit en détails dans la publication « MapReduce : Simplified Data Processing on Large Clusters » publiée en 2004 et ça pour faciliter la mise en œuvre de ses workfows de traitement parallèle. L'objectif principal était de remplacer la programmation complexe et non intuitive sur l'informatique distribuée (préalablement abordée par les plateformes HPC) par une plateforme transparente moderne avec seulement deux fonctions : Map et Reduce. Ces deux fonctions définies par l'utilisateur permettent aux utilisateurs d'utiliser les ressources distribuées sans se plaindre du réseau, de la planification, de la récupération après défaillance, etc.

Un modèle aussi intuitif que le MapReduce ne nécessite pas d'expertise concernant le parallélisme et les systèmes distribués. Son Framework Plug-and-Play embarque tous les détails pour implémenter les systèmes de calcul parallèle, la persistance et la résilience, l'optimisation et l'équilibre des ressources.

✓ **Principe de MapReduce :** MapReduce applique le principe dit « diviser pour distribuer pour régner », la stratégie mise en place pour exécuter un calcul sur des données massives consiste a :

- **Découper** les données en sous ensemble de plus petite taille, appelés lors ou fragments.
- **Affecter** chaque lot a une machine permettant ainsi leur traitement en parallèle.
- **Agréger** l'ensemble des résultats intermédiaires obtenu pour chaque lot pour construire le résultat final.

Ce dernier se repose sur deux fonctions : Map qui permet aux différents points du cluster distribué de distribuer leur travail. La fonction Reduce qui permet de réduire la forme finale des résultats des clusters en un seul résultat.

Le rôle du développeur d'applications distribuées c'est de réfléchir en MapReduce :

- Choisir une manière de découper les données afin que l'opération MAP soit parallèlisable.
- Choisir la clé a utilisé pour le problème ciblé.
- Ecrire le code de la fonction pour l'opération MAP.
- Ecrire le code de la fonction pour l'opération REDUCE.

Exemple d'utilisation de MapReduce : *Nous allons tester un programme MapReduce grâce à un exemple très simple, le WordCount, l'équivalent du Hello-World pour les applications de traitement de données. Le Wordcount permet de calculer le nombre de mots dans un fichier donné, en décomposant le calcul en deux étapes : L'étape de Mapping, qui permet de découper le texte en mots et de délivrer en sortie un flux textuel, où chaque ligne contient le mot trouvé, suivi de la valeur 1(pour dire que le mot a été trouvé une fois) et l'étape de Reducing, qui permet de faire la somme des 1 pour chaque mot, pour trouver le nombre total d'occurrences de ce mot dans le texte.*

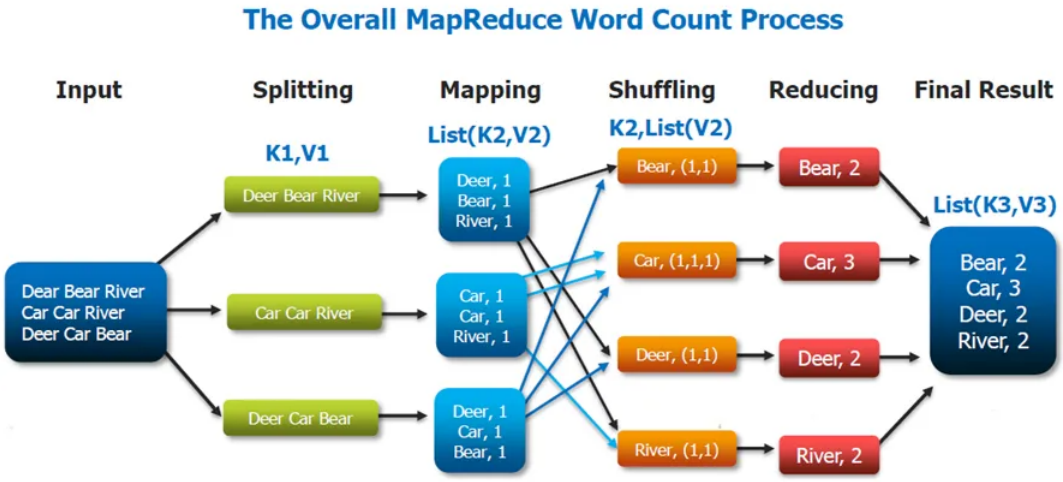

FIGURE 1.9 – Exemple de Word Count.

✓ **Les avantages de MapReduce :** Parmi les avantages de la programmation MapReduce nous citons

- La scalabilité.
- La flexibilité.
- La sécurité et l'authentification.
- Le traitement parallèle.
- La disponibilité.
- Un modèle simple de programmation.

❷ HADOOP :

MapReduce a tout son intérêt dans le Big Data car il permet le passage à l'échelle des traitements sur de gros volumes de données, cependant il faut une infrastructure logiciel dédiée qui permettent d'exécuté le schéma MapReuce de manière distribué sur les clusters machine. C'est là qu'intervient Hadoop.

Hadoop est un framework logiciel open source permettant de stocker des données, et de lancer des applications sur des grappes (cluster) de machines standards. Cette solution offre un espace de stockage massif pour tous les types de données, une immense puissance de traitement et la possibilité de prendre en charge une quantité de tâches virtuellement illimitée. Basé sur Java, ce framework fait partie du projet Apache, sponsorisé par Apache Software Foundation.

Grâce a MapReduce, il permet de traiter les immenses quantités de données. Plutôt que de devoir déplacer les données vers un réseau pour procéder au traitement, MapReduce permet de déplacer directement le logiciel de traitement vers les données.

Hadoop se compose essentiellement de :

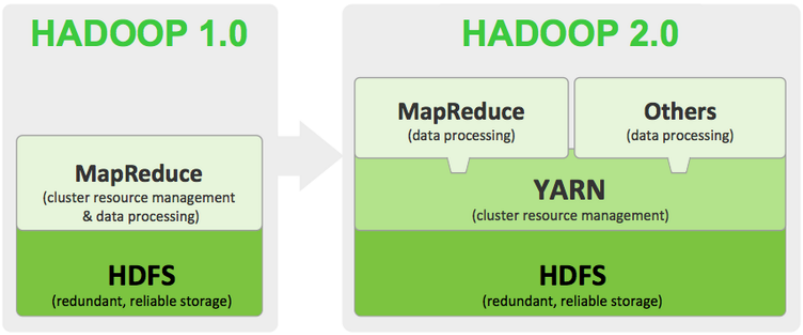

FIGURE 1.10 – Les composants d'Hadoop.

- ✓ **Système de gestion de fichiers HDFS (Hadoop Distributed File System) :** HDFS est un système de fichiers distribué, extensible et portable Inspiré par GFS et écrit en Java. Il est conçu pour être un système de stockage distribué, évolutif et résilient, conçu pour interagir facilement avec MapReduce. Il fournit une bande passante d'agrégation importante tout au long du réseau. Comme pour GFS, un réseau HDFS est composé d'un nœud maître appelé Namenode et des serveurs de données appelés Datanodes, de grande taille par défaut 64 Mo pour optimiser les temps de transfert et d'accès. Il est toutefois possible de monter à 128 Mo, 256 Mo, 512 Mo voire 1 Go.
- ✓ **Modèle de programmation Map-reduce :** Le framework MapReduce permet de traiter les immenses quantités de données. Plutôt que de devoir déplacer les données vers un réseau pour procéder au traitement, MapReduce permet de déplacer directement le logiciel de traitement vers les données. Ce modèle fera l'objet de la deuxième technologie que nous allons présenter.
- ✓ **YARN (Yet Another Ressource Negiciator)**à été ajoutée comme une fonctionnalité clé depuis Hadoop 2.0 déployée en 2013, Avant son ajout, Hadoop ne pouvait exécuter que des applications MapReduce. YARN a donc beaucoup augmenté les cas d'usage potentiels du framework. En découplant la gestion des ressources et la planification du composant de traitement de données de MapReduce, YARN a également permis à Hadoop de prendre en charge davantage d'applications et de types de traitement différents.

Hadoop 3 sortie le 13 décembre 2017 apporte son lot de nouveautés qu'il convient de présenter :

- La première différence provient de la gestion des conteneurs. La troisième apporte davantage d'agilité grâce à l'isolation des paquets de Docker.
- Le coût d'utilisation d'Hadoop 3 est également plus faible. La deuxième version demande plus d'espace de stockage. Là où la troisième version demande 9 blocs de stockage.
- Autre différence majeure, Hadoop 2 ne gère qu'un seul namenode. Cet outil de gestion de l'arborescence du système de fichiers, les métadonnées des fichiers et les répertoires. La version suivante peut en gérer plusieurs, ce qui permet d'augmenter de manière exponentielle la taille des infrastructures.
- Enfin, cette dernière version du système HDFS ouvre de nouvelles perspectives pour les concepteurs d'algorithmes de machine learning et de deep learning.

❸ Les bases de données NoSQL :

De nos jours, l'ubiquité de la connexion Internet est une réalité (les voitures que nous conduisons, les montres que nous portons, nos petits appareils médicaux domestiques, nos réfrigérateurs et congélateurs, nos Smartphones et ordinateurs portables). De plus, les données numériques produites par les êtres humains, dont les séquences vidéo, les photos et autres, atteignent des volumes importants de plusieurs EO par jour. Ces données actuellement stockées dans des bases qui leur ont été conçues spécifiquement sont gérés par des logiciels de gestion de bases de données volumineuses, jouant le rôle d'intermédiaires entre les bases de données d'un côté et les applicatifs et leurs utilisateurs de l'autre. On parle ici des bases de données non-relationnelles, dites NoSQL.

Concrètement une base de données NoSQL est une approche de la conception des bases et de leur administration particulièrement utile pour de très grands ensembles de données distribuées. Elle englobe une gamme étendue de technologies et d'architectures, afin de résoudre les problèmes de performances en matière d'évolutivité et de Big Data que les bases de données relationnelles ne sont pas conçues pour affronter. De plus elle est particulièrement utile lorsqu'une entreprise doit accéder, à des fins d'analyse, à de grandes quantités de données non structurées ou de données stockées à distance sur plusieurs serveurs virtuels du Cloud.

FIGURE 1.11 – Bases de données NoSql.

1.9.3 Les technologies de stockage :

1. **Stockage "In-Memory" :** Pour des analyses encore plus rapides, les traitements directement en mémoire sont une solution. Une technologie bien qu'encore trop coûteuse il est vrai pour être généralisée. Les bases de données "In Memory" sont généralement construites comme des base relationnelles. Elles sont conformes aux exigences ACID (Atomicity, Consistency, Isolation, Durability) qui garantissent l'intégrité des transactions. Les données contenues en mémoire sont volatiles par principe. Un système de sauvegarde périodique par image disque, snapshot, permet de sauvegarder la base. Ce système est complété d'une historisation des transactions afin de remettre la base en état en cas de coupure de courant.
2. **Le Cloud computing :** C'est une solution d'externalisation capable de louer de puissants moyens de calcul. Ceux-ci sont dotés de larges capacités de stockage extensibles et adaptés aux traitements des Big Data. Au cœur des réflexions sur les infrastructures IT, de nouvelles offres Big data as a Service (BDaaS).

   C'est un terme général employé pour désigner la livraison de ressources et de services à la demande par Internet. Il désigne le stockage et l'accès aux données par l'intermédiaire d'Internet plutôt que via le disque dur d'un ordinateur. Il s'oppose ainsi à la notion de stockage local, consistant à entreposer des données ou à lancer des programmes depuis le disque dur.

1.10 Le future du Big Data

Ses dernières années, plusieurs articles parlent de la fin du Big Data à cause de certains singes, comme le fait qu'Apache est décidé d'abandonné une dizaine de projets Hadoop.

Pris individuellement, le fait de retirer un projet peut sembler négligeable. Cependant, il est ici question de 13 logiciels, ce qui est un événement assez significatif. Voici, par ordre alphabétique, la liste des projets Apache Hadoop ayant fait l'objet d'un retrait : **Apex, Chukwa, Crunch, Eagle, Falcon, Hama, Lens, Marmotta, Metron, PredictionIO, Sentry, Tajo et Twill**. Alors que ces programmes sont liés au big data, la société a annoncé le 1er avril le retrait d'au moins 19 projets open source de sa réserve. Parmi ces derniers, une dizaine figure dans l'écosystème Hadoop.

Gartner considère quant à lui que les entreprises qui reposent sur de larges quantités de données historiques ont réalisé avec la pandémie que la plupart de leurs modèles ne sont plus pertinents. « La pandémie a tout changé, rendant beaucoup de données inutiles » expliquent les analystes.

Dépassées sont les techniques traditionnelles d'IA reposant sur des données historiques massives ! L'avenir est à des technologies d'analyse et d'IA qui requièrent moins de données, mais davantage de diversité. Il est temps de passer du « Big Data » au **« Small & Wide Data »**.

C'est l'une des grandes tendances mises en lumière par le nouveau rapport Gartner sur les 10 tendances *« Data & Analystics »* pour 2021. *« Ces tendances peuvent aider les organisations et la société à faire face aux changements perturbateurs, à l'incertitude et aux possibilités qu'elles offrent pour les trois années à venir »* explique Rita Sallam, VP Analyst chez Gartner.

FIGURE 1.12 – Exemple de Word Count.

Conclusion :

Chapitre 2

Le cloud computing :

Introduction

De plus en plus utilisé par les entreprises de toutes les industries, le Cloud Computing est la nouvelle forme de stockage de données du 21ème siècle.

2.1 Définitions

Définition 1 : « Le Cloud », littéralement le nuage, est un terme hérité du jargon technique. Aux débuts d'Internet, les diagrammes techniques représentaient souvent les serveurs et l'infrastructure réseau qui composent Internet sous la forme de nuages. Alors que de plus en plus de processus informatiques étaient déplacés vers cette partie 'serveurs et infrastructures' d'Internet, l'expression « passer dans le nuage » était une manière abrégée de désigner l'endroit où les processus informatiques se déroulaient. Aujourd'hui, « le cloud » est un terme largement accepté pour ce type d'accès.

Définition 2 : Le cloud computing est la fourniture de services informatiques notamment des serveurs, du stockage, des bases de données, la gestion réseau, des logiciels, des outils d'analyse, l'intelligence artificielle, via Internet (le cloud) dans le but d'offrir une innovation plus rapide, des ressources flexibles et des économies d'échelle.

Définition 3 : Un paradigme de calcul distribué émergeant dans lequel les données et les services sont disponibles dans des data centers extensibles et peuvent être accédés de manière transparente depuis des appareils (ordinateurs, téléphones, grappes, ...) connectés par Internet.

De manière générale, on parle de Cloud Computing lorsqu'il est possible d'accéder à des données ou à des programmes depuis internet, ou tout du moins lorsque ces données sont synchronisées avec d'autres informations sur internet. Il suffit donc pour y accéder de bénéficier d'une connexion internet.

2.2 Les composants clés du cloud computing :

Pour comprendre le fonctionnement du Cloud, il faut connaitre ses deux composants essentiels, à savoir la notion de Virtualisation et de Data center.

2.2.1 Virtualisation :

C'est l'ensemble des techniques matérielles et/ou logiciels qui permettent de faire fonctionner sur une seule machine, plusieurs systèmes d'exploitation (appelées machines virtuelles (VM), ou encore OS invité).

Même s'il existe plusieurs types de virtualisation, la forme la plus populaire de virtualisation dans le cloud est la virtualisation des serveurs qui consiste à dématérialiser le comportement et les données d'un serveur ou d'une machine, de façon à faire tourner plusieurs de ces instances dématérialisées sur un même serveur physique.

De cette façon, les diffiérentes instances créées se partagent les ressources du serveur physique. Cela permet une plus grande modularité dans la répartition des charges, une facilité dans l'administration des serveurs et une réduction des coûts.

FIGURE 2.1 – La Virtualisation

2.2.2 Les Data Center :

Un data center ou centre de données, c'est un site physique qui a une infrastructure composée d'un réseau d'ordinateurs et d'espaces de stockage, Il peut être interne ou externe à l'entreprise. Ces sites sont des salles remplies de baies de stockage, utilisées par de nombreuses entreprises et autres organisations gouvernementales

— **Les composants du Data Center :** Un centre de données basique regroupe des serveurs, des sous-systèmes de stockage, des commutateurs de réseau, des routeurs, des firewalls, et bien entendu des câbles et des racks physiques permettant d'organiser et d'interconnecter tout cet équipement informatique.

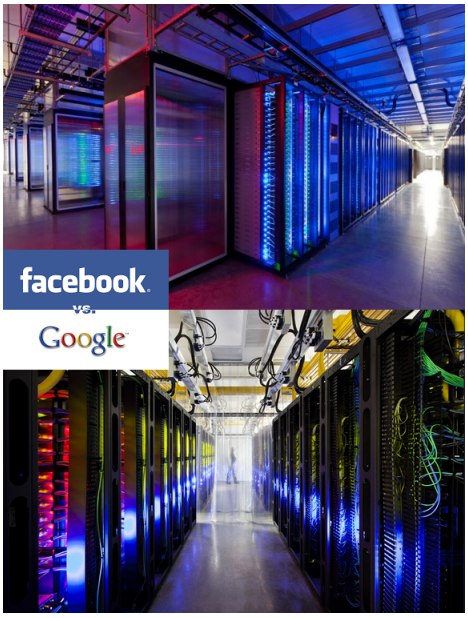

FIGURE 2.2 – Exemple de data center (ceux de Facebook et Google)

— **L'architecture du data center :** Théoriquement, n'importe quel espace suffisamment vaste peut servir de Data Center. Cependant, le design et l'implémentation d'un data center nécessite de prendre plusieurs précautions. Par-delà les problèmes basiques du coût et des taxes, les sites sont sélectionnés sur de nombreux critères, comme la localisation géographique, la stabilité météorologique, l'accès aux routes et aux aéroports, la disponibilité énergétique, les télécommunications ou encore l'environnement politique.

Pour fonctionner correctement, un Data Center doit aussi abriter l'infrastructure adéquate : un système distribution d'énergie, un commutateur électriques, des réserves d'énergie, des générateurs dédiés au backup, un système de ventilation et de refroidissement, et une puissante connexion internet. Une telle infrastructure nécessite un espace physique suffisamment vaste et sécurisé pour contenir tout cet équipement.

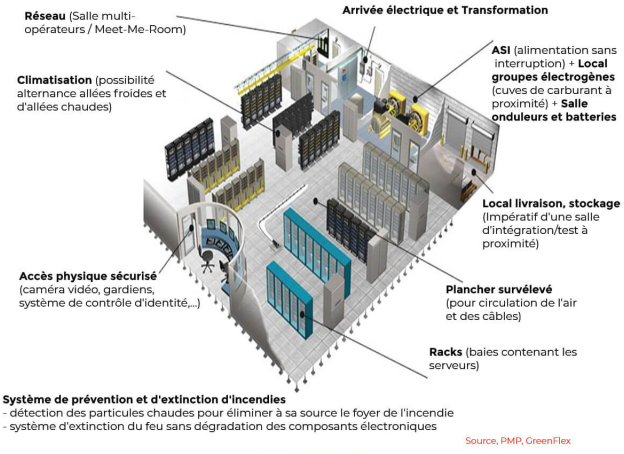

FIGURE 2.3 – Architecture du data center

2.3 Utilisations du Cloud Computing

Vous utilisez probablement en ce moment même le cloud computing sans le savoir. Si vous utilisez un service en ligne pour envoyer des courriers électroniques, modifier des documents, regarder des films ou regarder la télévision, jouer à des jeux ou stocker des images ou autres fichiers, il est probable que le cloud computing intervienne dans les coulisses. Les premiers services de cloud computing n'ont pas encore dix ans, mais un grand nombre d'organisations, par exemple des start-ups, des multinationales, des services administratifs ou des ONG, adopte cette technologie pour de nombreuses raisons.

Ici quelques exemples des possibilités d'utilisation des services cloud d'un fournisseur de cloud :

- Créez des applications cloud natives Créez, déployez et mettez à l'échelle rapidement des applications (web, mobiles et API). Tirez parti des technologies et approches cloud natives, telles que les conteneurs, Kubernetes, l'architecture de microservices, la communication pilotée par des API et DevOps.
- Tester et générer des applications Réduisez les coûts et délais de développement d'applications en utilisant des infrastructures cloud dont l'échelle peut être facilement adaptée.
- Stocker, sauvegarder et récupérer des données Protégez vos données à moindre coût et à grande échelle en les transférant via Internet vers un système de stockage cloud hors site, accessible à partir de tout emplacement et appareil.
- Analyser des données Unifiez vos données entre les équipes, les divisions et les emplacements dans le cloud. Utilisez ensuite des services cloud, par exemple de Machine Learning et d'intelligence artificielle, pour extraire des insights qui vous permettent de prendre des décisions éclairées.
- Diffuser du contenu audio et vidéo Communiquez avec votre public en tout lieu, en tout temps et sur tout appareil via un système audio et vidéo haute définition mondialement distribué.
- Incorporer de l'intelligence Utilisez des modèles intelligents pour interagir avec les clients et fournir des insights à partir des données capturées.
- Diffuser des logiciels à la demande Également appelés logiciel en tant que service les logiciels à la demande vous permettent d'offrir à vos clients les versions et mises à jour les plus récentes des logiciels, à tout moment et en tout lieu.

2.4 Caractéristiques du Cloud

Cette technologie offre plusieurs caractéristiques qui sont très avantageuse pour les utilisateurs professionnels et les utilisateurs finaux. Selon le NIST, le cloud computing doit posséder 5 caractéristiques essentielles qui sont :

- **Accès réseau universel :** L'ensemble des ressources doit être accessible et à disposition de l'utilisateur universellement et simplement à travers le réseau.
- **Libre service à la demande :** Permet à l'utilisateur d'utiliser et de libérer des ressources distantes en temps réel en fonction des ses besoins, sans nécessiter d'intervention humaine du côté fournisseur.
- **Ressources partagées :** Les ressources matérielles du fournisseur sont partagées entre les différents utilisateurs.
- **Élasticité :** Les ressources allouées aux utilisateurs peuvent être augmentées ou diminuées selon l'usage.
- **Service mesurable et facturable (pay-as-you-use) :** Les utilisateurs paieront pour les ressources qu'ils ont utilisées et pour la durée de leur utilisation.

2.5 inconvénients du Cloud

Cette technologie offre plusieurs avantages et bénéfices pour les utilisateurs professionnels et les utilisateurs finaux. Les trois principaux avantages sont l'approvisionnement en libre-service, l'élasticité, et le paiement à l'utilisation.

Pour de nombreuses personnes, le stockage local utilisé pendant les dernières décennies demeure aujourd'hui supérieur au Cloud Computing. Ces personnes considèrent qu'un disque dur permet de garder les données et les programmes physiquement proches, autorisant un accès rapide et simplifié pour les utilisateurs de l'ordinateur ou du réseau local.

Faire confiance aux opérateurs

C'est le principal reproche émis à l'égard du Cloud. Les télécoms, les entreprises de médias et les FAI contrôlent l'accès. Faire entièrement confiance au Cloud signifie également croire en un accès continu aux données sans aucun problème sur le long terme. Un tel confort est envisageable, mais son coût est élevé. De plus, ce prix continuera d'augmenter à mesure que les fournisseurs de Cloud trouvent un moyen de faire payer plus cher en mesurant par exemple l'utilisation du service. Le tarif augmente proportionnellement à la bande passante utilisée.

En dehors de ce problème de confiance, de nombreux autres arguments s'opposent au Cloud Computing. Le cofondateur d'Apple, Steve Wozniak, a ainsi critiqué le Cloud en 2012 en présageant de nombreux problèmes de grande envergure dans les cinq années à venir. On peut par exemple redouter des crashs. Durant l'été 2012, Amazon a rencontré ce type de problème. En tant que fournisseur d'entreprises comme Netflix ou Pinterest, l'entreprise américaine a ainsi provoqué la mise hors service des plateformes de ces clients. En 2014, Dropbox, Gmail, Basecamp, Adobe, Evernote, iCloud et Microsoft ont rencontré des problèmes similaires. En 2015, ce fut le tour de Apple, Verizon, Microsoft, AOL, Level 3, Google et Microsoft. Ces désagréments ne durent généralement que quelques heures, mais représentent une perte d'argent colossale pour les entreprises affectées.

La question de la propriété intellectuelle

Par ailleurs, Wozniak a exprimé ses inquiétudes concernant la propriété intellectuelle. Il est en effet difficile de déterminer à qui appartiennent les données stockées sur internet. On peut prendre pour exemple les nombreuses controverses survenues au sujet des changements de conditions d'utilisation de sites dérivés du Cloud comme Facebook ou Instagram. Ces réseaux sociaux créent la polémique en s'octroyant des droits sur les photos stockées sur leurs plateformes. Il y a également une différence entre les données mises en ligne et les données créées directement au sein du Cloud. Un fournisseur pourrait aisément revendiquer la propriété de ces dernières. La propriété est donc un facteur à prendre en compte.

Aucune autorité centrale ne gouverne l'usage du Cloud pour le stockage et les services. L'Institute of Electrical and Electronics Engineers (IEEE) tente de devenir cet organe régulateur. En 2011, il a créé l'IEEE Cloud Computing Initiative, visant à établir des standards pour l'utilisation, particulièrement dans le domaine des entreprises. Pour l'heure, les règles sont encore floues et les problèmes se règlent au cas par cas.

2.6 Types de services Cloud

La plupart des services de cloud computing peuvent être classés en trois grandes catégories : IaaS (infrastructure as a service), PaaS (platform as a service), SaaS (software as a service). On les appelle parfois « pile » de cloud computing, car elles s'empilent les unes sur les autres. Si vous savez en quoi elles consistent et en quoi elles sont différentes, vous pourrez plus facilement atteindre vos objectifs. Ses derniers temps se rajoute a ses 3 services deux autres qu'on va présenter ci-dessous.

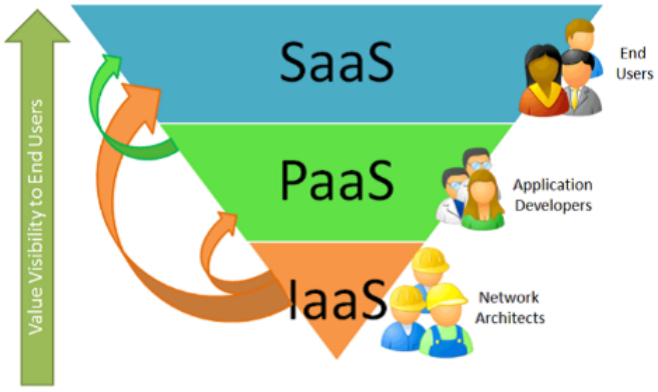

FIGURE 2.4 – Services cloud

Service	Définition	Exemple
Infrastructure as a service (IaaS)	La catégorie la plus basique des services de cloud computing. Avec l'IaaS, vous louez une infrastructure informatique (serveurs, machines virtuelles, stockage, réseaux, systèmes d'exploitation) auprès d'un fournisseur de services cloud, avec un paiement en fonction de l'utilisation.	• Amazon EC2 Elastic Compute Cloud. • S3 / Simple Storage Service d'Amazon. • Google drive, DropBox qui sont gratuits.
Platform as a service (PaaS)	L'expression plateforme en tant que service (PaaS, Platform-as-a-Service) qualifie les services de cloud computing qui offrent un environnement à la demande pour développer, tester, fournir et gérer des applications logicielles. PaaS est conçu pour permettre aux développeurs de créer rapidement des applications web ou mobiles sans avoir à se préoccuper de la configuration ou de la gestion de l'infrastructure de serveurs, de stockage, de réseau et de bases de données nécessaire au développement.	• App Engine de Google qui se limite à Java et Python. • Windows Azure de Microsoft permet de travailler avec les langages comme .NET, PHP, Python, Ruby et Java.

Software as a service (SaaS)	Le logiciel en tant que service (SaaS, Software-as-a-Service) est une méthode de diffusion d'applications logicielles via Internet, à la demande et en général sur abonnement. Avec le SaaS, les fournisseurs de services cloud hébergent et gèrent les applications logicielles et l'infrastructure sous-jacente, et gèrent la maintenance, par exemple la mise à niveau des logiciels et l'application des correctifs de sécurité. Les utilisateurs se connectent à l'application via Internet, en général par l'intermédiaire d'un navigateur web sur leur téléphone, leur tablette.	• Google apps avec Google Docs, Calendar et Gmail qui sont gratuites. • Facebook, Linkdin qui sont gratuits. • Offices 365 de Microsoft propose des applications web (Word,Excel, PowerPoint, Publisher...).
DBaaS (DataBase as a service)	Un tel modèle fournit des mécanismes transparents pour créer, stocker, accéder et mettre à jour des bases de données. De plus, le fournisseur de services de base de données assume l'entière responsabilité de l'administration de la base de données, garantissant ainsi la sauvegarde, la réorganisation et les mises à jour de version. L'utilisation de ce service permet aux fournisseurs de répliquer et de personnaliser leurs données sur plusieurs serveurs, qui peuvent être physiquement séparé [16].	• Amazon Web Services. • RackSpace. • IBM, Microsoft, Oracle.
Informatique serverless	Se chevauchant avec PaaS, l'informatique Serverless se concentre sur la création de fonctionnalités applicatives sans perte de temps en lien avec la gestion permanente des serveurs et de l'infrastructure requise à cette fin. Le fournisseur de cloud se charge de la configuration, de la planification de la capacité et de l'administration du serveur à votre place. Les architectures serverless sont hautement scalables et basées sur des événements. Elle n'utilisent des ressources que quand une fonction ou un déclencheur spécifiques s'activent.	• Google apps avec Google Docs, Calendar et Gmail qui sont gratuites. • Facebook, Linkdin qui sont gratuits. • Offices 365 de Microsoft propose des applications web (Word,Excel, PowerPoint, Publisher...).

TABLE 2.1 – Les services Cloud

2.7 Types et modes de stockage dans le cloud computing :

2.7.1 Les types de stockage dans le cloud computing :

Tous les clouds ne sont pas identiques et aucun type de cloud computing ne convient à tout le monde. Plusieurs modèles, types et services différents ont évolué pour nous aider à trouver la solution adaptée à vos besoins

- **Cloud public :** est détenu et exploité par un fournisseur de cloud tiers, qui propose des ressources de calcul, telles que des serveurs et du stockage, via Internet. Microsoft Azure est un exemple de cloud public. Dans ce dernier, tout le matériel, tous les logiciels et toute l'infrastructure sont la propriété du fournisseur du cloud. Vous accédez à ces services et vous gérez votre compte par l'intermédiaire d'un navigateur web.
- **Cloud privé :** est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local des entreprises, dans les quelles paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé qui est un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau privé.
- **Cloud hybride :** regroupe des clouds publics et privés, liés par une technologie leur permettant de partager des données et des applications. En permettant que les données et applications se déplacent entre des clouds privé et public, un cloud hybride offre à l'entreprise une plus grande flexibilité, davantage d'options de déploiement et une optimisation de l'infrastructure, de sécurité et de conformité existantes.
- **Multicloud :** est un type de déploiement cloud qui implique l'utilisation de plusieurs clouds publics. Autrement dit, une organisation disposant d'un déploiement multi-cloud loue des serveurs et des services virtuels auprès de plusieurs fournisseurs externes. Les déploiements multi-cloud peuvent aussi être des clouds hybrides, et vice-versa.

2.7.2 Les formats de stockage dans le cloud :

Il existe trois formats de stockage de données dans le cloud :

- **Stockage en mode bloc :** Le stockage en mode bloc permet de diviser un seul volume de stockage (par exemple, un nœud de stockage dans le cloud) en plusieurs instances individuelles appelées blocs. Il s'agit d'une solution de stockage rapide à faible latence, idéale pour les charges de travail hautes performances.
- **Stockage en mode objet :** Le stockage en mode objet implique d'attribuer à chaque donnée des identifiants uniques, que l'on appelle les métadonnées. Compte tenu du fait que les objets ne sont ni compensés ni chiffrés, ils sont rapidement accessibles à très grande échelle. Il s'agit donc d'une solution idéale pour les applications cloud-native.
- **Stockage en mode fichier :** Le stockage en mode fichier est le plus utilisé sur les systèmes NAS. Il permet d'organiser les données et de les présenter aux utilisateurs. Sa structure hiérarchique permet de parcourir les données du haut vers le bas en toute simplicité, mais augmente le temps de traitement.

Conclusion :

Chapitre 3

Les Objets connectés (IoT) :

Introduction

Depuis la fin des années 1980, Internet a évolué de manière spectaculaire. La dernière étape est l'utilisation de ce réseau mondial pour la communication avec des objets ou entre objets, évolution nommée Internet des Objets (IoT pour Internet of Things).

L'évolution de l'IoT est rapide et n'a pas de limites, elle constitue la prochaine étape de la révolution numérique. En effet, depuis 2014, le nombre d'objets connectés est supérieur au nombre d'humains connectés et il est prévu que 50 milliards d'objets seront connectés en 2020.

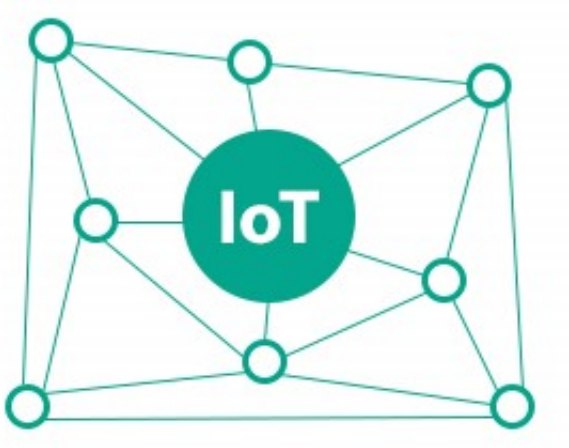

FIGURE 3.1 – Internet of Things

3.1 Définition :

Définition 1 : L'IoT (Internet of Things, pour Internet des Objets) est un système d'interconnexion entre des dispositifs informatiques, des machines, des objets, des animaux et même des personnes, munies d'identifiants uniques (UID) avec la capacité de transférer des données sur un réseau. Et ce, sans interaction d'humain à humain ou d'humain à ordinateur.

Globalement, il s'agit de tout « objet » naturel ou artificiel auquel on peut attribuer une adresse IP et qui peut transférer des données sur un réseau. De plus en plus d'entreprises, quelque soit le secteur, utilisent l'IoT pour fonctionner plus efficacement et mieux comprendre leurs clients, afin d'offrir de meilleurs services. Mais aussi pour améliorer la prise de décision et accroître la valeur de l'entreprise.

Définition 2 : L'internet des objets peut aussi être défini selon l'UIT, comme étant une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution.

3.2 L'histoire de l'IoT :

C'est lors d'une présentation faite à Procter & Gamble, en 1999, que Kevin Ashton, co-fondateur de l'Auto-ID Center au MIT, a mentionné pour la première fois l'Internet des Objets. Il souhaitait attirer l'attention des directeurs de P& G sur les puces RFID (identification par radiofréquence). Cependant, il a nommé sa présentation « Internet of Things » pour intégrer la nouvelle tendance de l'année : Internet. Toutefois, l'idée d'appareils connectés existe depuis les années 1970. Ainsi, le premier objet connecté était une machine à Coca, à l'Université de Carnegie Mellon, au début des années 1980.

Via le Web, les développeurs pouvaient vérifier l'état de la machine, et déterminer si une boisson froide était disponible pour eux. L'IoT a ensuite évolué avec des technologies sans fil (Wifi par exemple), des MEMS (systèmes microélectromécaniques), des microservices et d'Internet. Ainsi, la technologie opérationnelle (OT – Operational Technology) et la technologie de l'information (IT) se sont rapprochées. Ceci a permis d'analyser des données non structurées, générées par des machines, pour en tirer des axes d'amélioration.

L'Internet des Objets utilise comme base la connectivité M2M (Machine to Machine). C'est-à-dire que des machines se connectent entre elles, via un réseau, sans interaction humaine. L'IoT est donc un réseau de milliards de noeuds (capteurs ici), qui connectent des personnes, des systèmes et d'autres applications pour collecter et partager des données. Néanmoins, le concept d'écosystème de l'Internet of Things ne s'est concrétisé qu'au milieu des années 2010. Une avancée que l'on doit au gouvernement chinois, qui a déclaré qu'il ferait de l'IoT une priorité stratégique, notamment dans sa stratégie de reconnaissance faciale et de fichage du peuple chinois.

3.3 les composantes d'un réseau IoT :

3.4 Architecture de l'Internet des objets

3.5 La s´ecurité dans l'Internet des objets

3.6 Internet of Things : des applications pour tous ?

Il existe de nombreuses applications de l'internet des objets. Cela va de l'IoT pour les industries de la grande consommation et de l'IoT entreprise à l'IoT manufacturier ainsi qu'à l'IoT industriel (IIoT). Ces applications couvrent de nombreux secteurs verticaux, notamment l'automobile, les télécommunications et l'énergie.

1. Particuliers, bâtiments intelligents et sécurité publique

Dans le segment des consommateurs, on peut citer les maisons intelligentes avec la domotique. Elles sont équipées de thermostats, d'appareils électroménagers, de chauffage, d'éclairage ou encore d'appareils électroniques intelligents. Ils peuvent tous être connectés et commandés à distance, via des ordinateurs, smartphones et autres appareils mobiles. Les bâtiments intelligents peuvent même réduire les coûts énergétiques grâce à des capteurs qui détectent le nombre d'occupants d'une pièce.

Du côté de la sécurité publique, des dispositifs portatifs permettent d'améliorer les délais d'intervention des permiers secours, en cas d'urgence, lors d'incendie par exemple ou lors d'une crise cardiaque d'une personne portant in pace maker. ou encore grâce à des itinéraires optimisés pour l'intervention des policiers ou du SAMU. Ou encore en suivant les signes vitaux des travailleurs de chantier ou des pompiers, sur des sites où leur vie est en danger. Dans le domaine de la santé, l'IoT permet de suivre les patients de plus près.

2. Hôpitaux, smart cities et entreprises

Les hôpitaux les utilisent aussi pour la gestion des stocks de produits pharmaceutiques et les instruments médicaux. En agriculture, l'Internet des Objets permet, par exemple, de surveiller la luminosité, la température, le taux d'humidité dans l'air et dans les sols des champs cultivés. Dans une smart city, ou ville intelligente, l'IoT se déploie à travers les réverbère intelligents et des compteurs intelligents. Ils permettent notamment de réduire la circulation et améliorer l'assainissement. Mais aussi réaliser des économies d'énergie et répondre aux préoccupations environnementales.

Pour les organisations, l'Internet des objets offre de multiples avantages, comme améliorer l'expérience client. Mais aussi, surveiller l'ensemble de leurs processus opérationnels, intégrer et adapter des modèles commerciaux et prendre de meilleures décisions. Ou encore améliorer la productivité des employés, économiser du temps et de l'argent, et générer plus de revenus. Globalement, l'IoT encourage les entreprises à repenser la manière dont elles abordent leurs activités, leurs industries et leurs marchés. Et il leur donne les outils nécessaires pour améliorer leurs stratégies commerciales.

3.7 Le rôle de l'Internet des objets dans le Big Data

A mesure que le nombre d'objets connectés augmente, le volume de données générées par l'internet des objets explose. Ainsi, pour pouvoir les prendre en charge et les analyser en temps réel, il est nécessaire de s'en remettre aux outils analytiques Big Data.

Ces outils ont la capacité de traiter rapidement les larges volumes de données générées en continu par les appareils IoT, et d'en extraire des insights exploitables. Le machine learning permet notamment de repérer des modèles de données. Avec ces patterns, une entreprise peut notamment mettre en place la maintenance prédictive sur ses machines industrielles.

Exemples de cas d'usage :

Pour illustrer la corrélation entre IoT et Big Data, on peut prendre l'exemple des sociétés de transport. Ces dernières utilisent les données collectées par des capteurs et les outils d'analyse Big Data pour améliorer leur efficience, économiser de l'argent, et réduire leur impact sur l'environnement.

Ou les véhicules de livraison embarquent des capteurs qui permettent de surveiller l'état du moteur, le nombre d'arrêts, la vitesse de déplacement, le nombre de kilomètres parcourus ou encore la quantité de carburant consommée.

Chapitre 4

Le NoSql :

Introduction

De nos jours, l'ubiquité de la connexion Internet est une réalité (les voitures que nous conduisons, les montres que nous portons, nos petits appareils médicaux domestiques, nos réfrigérateurs et congélateurs, nos Smartphones et ordinateurs portables). De plus, les données numériques produites par les êtres humains, dont les séquences vidéo, les photos et autres, atteignent des volumes importants de plusieurs EO par jour.

Ces données actuellement stockées dans des bases qui leur ont été conçues spécifiquement sont gérés par des logiciels de gestion de bases de données volumineuses, jouant le rôle d'intermédiaires entre les bases de données d'un côté et les applicatifs et leurs utilisateurs de l'autre. On parle ici des bases de données non-relationnelles, dites NoSQL.

4.1 Rappel de gestion de base de données relationnelles :

Au début de l'ère des bases de données, chaque application stockait ses données au sein de sa propre structure unique. Lorsque les développeurs voulaient créer des applications pour utiliser ces données, ils devaient en savoir beaucoup sur la structure spécifique des données afin de trouver les données dont ils avaient besoin. Ces structures de données étaient inefficaces, difficiles à gérer et à optimiser pour obtenir de bonnes performances pour les applications. Le modèle de base de données relationnelle a été conçu pour résoudre le problème que présentent plusieurs structures de données arbitraires.

4.1.1 Définition :

Le modèle relationnel a été introduit pour la première fois par Ted Codd du centre de recherche d'IBM en 1970 dans un papier désormais classique, et attira immédiatement un intérêt considérable en raison de sa simplicité et de ses fondations mathématique.

Dans ce modèle, les données sont représentées par des tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle relationnel, d'autre part ces données sont gérées par un système de gestion de bases de données relationnelle (relational database management system, en anglais), qui représente un système intégré pour la gestion unifiée des bases de données relationnelles, il est constitué d'un composant de stockage et d'un composant de gestion de données.

L'interface standard pour une base de données relationnelle est le langage SQL (Structured Query Language) considéré comme le langage de manipulation des données relationnelles le plus utilisé aujourd'hui. Il est devenu un standard de fait pour les SGBD relationnels. Il possède des caractéristiques proches de l'algèbre relationnelle (jointures, opérations ensemblistes) et d'autres proches du calcul des tuples (variables sur les relations). SQL est un langage redondant qui permet souvent d'écrire les requêtes de plusieurs façons différentes.

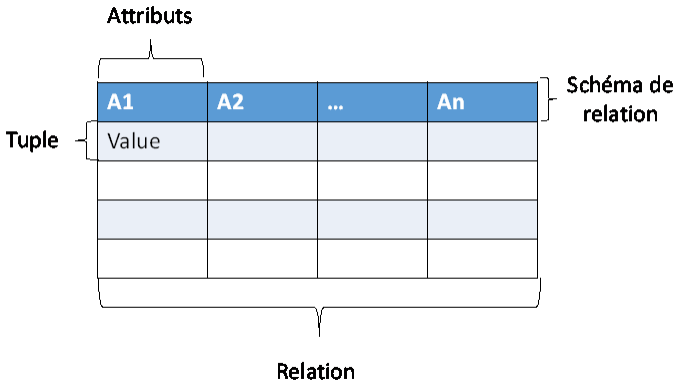

FIGURE 4.1 – Éléments d'une table d'une BDDR.

4.1.2 Les règles CODD :

Les douze (12) règles de Codd sont un ensemble de règles édictées par Edgar F.Codd afin de définir les caractéristiques que doit présenter un système de gestion de base de données (SGBD) afin d'être considéré comme relationnel (SGBDR). On considère parfois une règle 0, qui stipule que l'intégralité des fonctions du SGBDR doit être accessible par le modèle relationnel.

1. **Unicité :** toutes les informations sur les données sont représentées au niveau logique (valeurs dans des colonnes de tables) et non physique.
2. **Garantie d'accès :** les données sont accessibles sans ambiguïté uniquement par la combinaison du nom de la table, de la clef primaire et du nom de la colonne.
3. **Traitement des valeurs nulles :** une valeur spéciale doit représenter l'absence de valeur, une information manquante ou une information inapplicable (valeur NULL).
4. **Catalogue lui-même relationnel :** la description de la base de données doit être accessible comme les données ordinaires (un dictionnaire des données est enregistré dans la base).
5. **Sous langage de données :** un langage doit permettre de définir les données, définir des vues (visions particulières de la base, enregistrées comme des relations), manipuler les données, définir les contraintes d'intégrité, des autorisations et gérer des transactions.
6. **Mise à jour des vues :** toutes les vues pouvant théoriquement être mises à jour doivent pouvoir l'être par le système.
7. **Insertion, mise à jour, et suppression de haut niveau :** le langage doit comporter des ordres effectuant l'insertion, la mise à jour et la suppression de données, aussi bien pour des lots de tuples issues de plusieurs tables que juste pour un tuple unique issu d'une table unique.
8. **Indépendance physique :** indépendance vis à vis de l'implantation physique des données.
9. **Indépendance logique :** indépendance vis à vis de l'implantation logique des données (tables, colonnes, etc.).
10. **Indépendance d'intégrité :** les contraintes d'intégrité doivent pouvoir être définies dans le langage relationnel et enregistrées dans le dictionnaire des données (catalogue).
11. **Indépendance de distribution :** indépendance de la répartition des données sur divers sites.
12. **Règle de non subversion :** on ne peut jamais contourner les contraintes (d'intégrité ou de sécurité) imposées par le langage du SGBD en utilisant un langage de programmation de plus bas niveau.

Remarque : *L'ensemble de ces règles indique la voie à suivre pour les systèmes de gestion de bases de données relationnelles. Elles ne sont jamais totalement implémentées, à cause des difficultés techniques que cela représente.*

4.1.3 Les contraintes des SGBDs relationnels (Propriétés ACID) :

Selon la théorie des bases de données, les propriétés ACID sont les quatre principaux attributs d'une transaction de données. Il s'agit là d'un des concepts les plus anciens et les plus importants du fonctionnement des bases de données, il spécifie quatre buts à atteindre pour toute transaction. Ces buts sont les suivants :

1. **Atomicity (Atomicité) :** Lorsqu'une transaction est effectuée, toutes les opérations qu'elle comporte doivent être menées à bien : en effet, en cas d'échec d'une seule des opérations, toutes les opérations précédentes doivent être complètement annulées, peu importe le nombre d'opérations déjà réussies. En résumé, une transaction doit s'effectuer complètement ou pas du tout.

   Exemple : *une transaction qui comporte 3000 lignes qui doivent être modifiées, si la modification d'une seule des lignes échoue, alors la transaction entière est annulée. L'annulation de la transaction est toute à fait normale, car chaque ligne ayant été modifiée peut dépendre du contexte de modification d'une autre, et toute rupture de ce contexte pourrait engendrer une incohérence des données de la base.*

2. **Consistancy (Cohérence) :** Avant et après l'exécution d'une transaction, les données d'une base doivent toujours être dans un état cohérent. Si le contenu final d'une base de données contient des incohérences, cela entraînera l'échec et l'annulation de toutes les opérations de la dernière transaction. Le système revient au dernier état cohérent. La cohérence est établie par les règles fonctionnelles.

   Exemple : *Un système doit être capable de reconnaître qu'une facture est liée à un client et aux éléments factures. Il doit être capable d'éviter, par exemple, la suppression d'un client s'il existe encore des factures pour ce client, et la suppression d'une facture qui a des éléments associées.*

3. **Isolation (Isolation) :** La caractéristique d'isolation permet à une transaction de s'exécuter en un mode isolé. En mode isolé, seule la transaction peut voir les données qu'elle est en train de modifier, c'est le système qui garantit aux autres transactions exécutées en parallèle une visibilité sur les données antérieures. Ce fonctionnement est obtenu grâce aux verrous système posés par le SGBD.

   Exemple : *Prenons l'exemple de deux transactions A et B : lorsque celles-ci s'exécutent en même temps, les modifications effectuées par A ne sont ni visibles, ni modifiables par B tant que la transaction A n'est pas terminée et validée.*

4. **Durability (Durabilité) :** Toutes les transactions sont lancées de manière définitive. Une base de données ne doit pas afficher le succès d'une transaction pour ensuite remettre les données modifiées dans leur état initial. Pour ce faire, toute transaction est sauvegardée dans un fichier journal afin que, dans le cas où un problème survient empêchant sa validation complète, elle puisse être correctement terminée lors de la disponibilité du système.

4.1.4 Limite des bases de données relationnelles :

Les bases de données existent maintenant depuis environ 56 ans et le modèle relationnel depuis environ 46 ans, pendant plusieurs décennies, ce modèle bien très puissant, représentait la solution parfaite pour les différents acteurs dans le domaine de gestion des données, néanmoins ces architectures ont atteints leurs limites pour certains services ou sites manipulant de grandes masses de données, tels que Google, Facebook, etc. En effet ce genre de sites possède plusieurs millions voire des milliards d'entrées dans leurs bases de données et tout autant de visites journalières, en conséquence les données sont distribuées sur plusieurs machines, de plus pour des raisons de fiabilité ces bases de données sont dupliquées pour que le service ne soit pas interrompu en cas de panne.

Malheureusement le modèle relationnel présente quelques problèmes liés à ce passage à l'échelle tel que :

1. Problème lié à l'application des propriétés ACID en milieu distribué :

Une base de données relationnelle est construite en respectant les propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité), ses propriétés bien que nécessaires à la logique du relationnel nuisent fortement aux performances et en particulier la propriété de cohérence.

En effet, la cohérence est très difficile à mettre en place dans le cadre de plusieurs serveurs (environnement distribué), car pour que celle-ci soit respectée tous les serveurs doivent être des miroirs les uns des autres, de ce fait deux problèmes apparaissent :

- Le coût en stockage est énorme car chaque donnée est présente sur chaque serveur.
- Le coût d'insertion/modification/suppression est très grand, car on ne peut valider une transaction que si on est certain qu'elle a été effectuée sur tous les serveurs et le système fait patienter l'utilisateur durant ce temps.

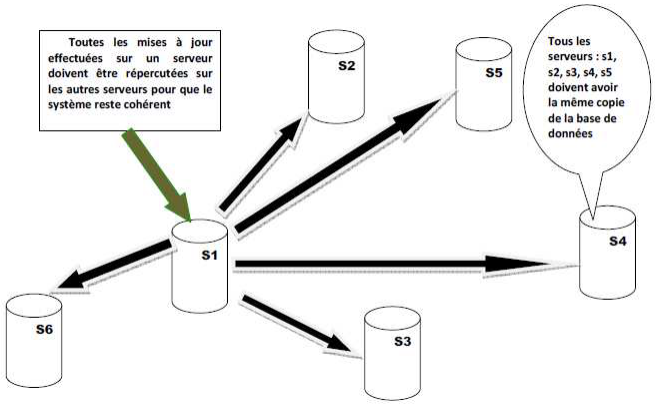

FIGURE 4.2 – Problème lié aux propriétés ACID en milieu distribué

2. Pas de schéma de base de données hiérarchique :

Contrairement aux bases de données orientées objet, les bases de données relationnelles n'offrent pas la possibilité d'implémenter des schémas de base de données avec des classes hiérarchiquement structurées. Des concepts tels que les entités subordonnées qui héritent de propriétés d'entités supérieures ne peuvent pas être implémentés avec elles. Par exemple, on ne peut pas créer de sous-tuples avec eux. Tous les tuples d'une base de données relationnelle se trouvent au même niveau hiérarchique.

3. Problème de requête non optimale dû à l'utilisation des jointures :

Imaginons une table contenant toutes les personnes ayant un compte sur Facebook, soit 1.55 milliards d'utilisateurs actifs par mois les données dans une base de données relationnelle classique sont stockées par lignes, ainsi si on effectue une requête pour extraire tous les amis d'un utilisateur donné, il faudra effectuer la jointure entre la table des usagers et celle des amities (chaque usagr ayant au moins un ami) puis parcourir le produit cartésien de ces deux tables. De ce fait, on perd énormément en performances en raison du temps consommé pour stocker et parcourir une telle quantité de données.

4. Problème lié à la gestion des objets hétérogènes :

« Le stockage distribué n'est pas la seule contrainte qui pèse à ce jour sur les systèmes relationnels» disait Carl STROZZI. Au fur et à mesure du temps, les structures de données manipulées par les systèmes sont devenues de plus en plus complexes en contrepartie les moteurs de stockage évoluant peu. Le principal point faible des modèles relationnels est l'absence de gestion d'objets hétérogènes ainsi que le besoin de déclarer au préalable l'ensemble des champs représentant un objet.

D'autre part le modèle relationnel est fondé sur un modèle mathématique solide s'appuyant sur des concepts simples qui font sa force en même temps que sa faiblesse.

Nous expliquerons quelques limites :

1. **Surcharge sémantique :** Le modèle relationnel s'appuie sur un seul concept (la relation) pour modéliser à la fois les entités et les associations entre ces entités. Il existe donc un décalage entre la réalité et sa représentation abstraite.
2. **Types de données :** Ces modèles sont limités à des types simples (entiers, réels, chaînes de caractères), les seuls types étendus se limitant à l'expression de dates ou de données financières, ainsi que des conteneurs binaires de grande dimension (BLOB, pour Binary Large OBjects) qui permettent de stocker des images ainsi que des fichiers audio ou vidéos. Ces BLOBs ne sont toutefois pas suffisants pour représenter des données complexes (pas de structure), les mécanismes de contrôle BD sont inexistants, et le langage de requêtes (SQL) ne possède pas les opérateurs correspondant aux objets stockés dans ces BLOBs.

5. Le partitionnement de données :

L'un des problèmes de la normalisation dans un SGBDR concerne la distribution des données et du traitement. S'il y a des données stockées ayant un rapport entre elles, comme des clients, des commandes, des factures, des lignes de facture, etc., dans des tables différentes, des problèmes surgiront en cas de partitionnement de ces données. Pour y remédier, il faut alors s'assurer que les données en rapport les unes avec les autres se trouvent sur le même serveur.

4.1.5 Exemples de bases de données relationnelles

Les systèmes de gestion de bases de données relationnelles (SGBDR) les plus couramment utilisés sont donnés comme suit :

- **Db2 :** Est l'un des SGBD relationnelles propriétaire d'IBM 3 disponible aux utilisateurs sous licence commerciale.
- **Microsoft SQL Server :** Le système de gestion de base de données de Microsoft 4 en langage SQL est disponible sous une licence par utilisateur payante.
- **MySQL :** Est le SGBDR open source 5 le plus utilisé dans le monde. Depuis son acquisition par Oracle, MySQL est commercialisé sous une double licence. La communauté des développeurs d'origine poursuit le projet sous le nom de MariaDB.
- **PostgreSQL :** avec PostgreSQL, les utilisateurs peuvent accéder gratuitement à un système de gestion de base de données relationnel-objet (SGBDRO). Le développement ultérieur est effectué par une communauté open source.
- **Oracle Database :** le système de gestion de base de données relationnelle de la société du même nom Oracle 6 est commercialisé sous licence propriétaire contre rémunération.
- **SQLite :** est une bibliothèque appartenant au domaine public contenant un système de gestion de bases de données relationnelles.

Résumé : Le schéma relationnel des bases de données est clair, mathématiquement solide et a fait ses preuves dans la pratique depuis plus de 40 ans. Pourtant, le stockage des données dans des tables structurées ne répond pas à toutes les exigences des technologies modernes de l'information.

4.2 Les bases de données NoSQL :

Depuis les années 70, la base de données relationnelle était l'incontournable référence pour gérer les données d'un système d'information. Toutefois, face aux 3V (Volume, Velocity, Variety), le relationnel peut difficilement lutter contre cette vague de données. Le NoSQL s'est naturellement imposé dans ce contexte en proposant une nouvelle façon de gérer les données, sans reposer sur le paradigme relationnel, d'où le "Not Only SQL". Cette approche propose de relâcher certaines contraintes lourdes du relationnel pour favoriser la distribution (structure des données, langage d'interrogation ou la cohérence).

4.2.1 Définition :

En informatique et en bases de données, NoSQL désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. L'explicitation la plus populaire de l'acronyme est Not only SQL (« pas seulement SQL » en anglais) et c'est en effet ce que ce modèle de base de données veut être : non pas une contrepartie, mais bien un enrichissement et complément utile des bases de données SQL relationnelles traditionnelles. Ce faisant, les bases de données NoSQL dépassent les limites des systèmes relationnels et exploitent un modèle de base de données alternatif. Cela ne veut toutefois pas dire qu'aucun système SQL n'est utilisé. Il existe de nombreuses variantes combinées au sein desquelles les deux solutions peuvent être utilisées et qui restent toutefois englobées sous l'étiquette NoSQL.

La définition exacte de la famille des SGBD NoSQL reste sujette à débat. Le terme se rattache autant à des caractéristiques techniques qu'à une génération historique de SGBD qui a émergé autour des années 2010. D'après Pramod J. Sadalage et Martin Fowler, la raison principale de l'émergence et de l'adoption des SGBD NoSQL serait le développement des centres de données et la nécessité de posséder un paradigme de bases de données adapté à ce modèle d'infrastructure matérielle.

- **Distinction des autres termes :**
  - **Non SQL :** Le terme « bases de données non SQL » est trompeur, pour ne pas dire faux. Il prédit que ces bases de données sont sans aucune utilisation de SQL et ce n'est pas toujours le cas dans NoSQL. Le terme bases de données non SQL existe, mais il est plutôt utilisé comme un terme vague que comme une expression professionnelle. Cela signifie que les données sont traitées par un autre langage que SQL, par ex. XQuery pour les bases de données XML.
  - **Distributed storage :** Liés à NoSQL, il existe des termes tels que « stockage distribué » ou « stockage structuré distribué ». Il est assez difficile de distinguer ces termes de NoSQL, car ils décrivent exactement une caractéristique de NoSQL. Le stockage distribué est un terme générique pour un système qui prétend être un stockage unique mais qui est en réalité une collection de nombreuses unités informatiques stockant des parties des fichiers. Certaines bases de données NoSQL prétendent être un système de stockage distribué, par exemple BigTable de Google.

4.2.2 Théorème de CAP :

Le théorème de CAP est l'acronyme de « Coherence », « Availability » et « Partition tolerance », aussi connu sous le nom de théorème de Brewer. Ce théorème, formulé par Eric Brewer en 2000 et démontré par Seth Gilbert et Nancy Lych en 20025, énonce une conjecture qui définit qu'il est impossible, sur un système informatique de calcul distribué, de garantir en même temps les trois contraintes suivantes :

- **« Coherence » (Cohérence) :** Tous les clients du système voient les mêmes données au même instant.
- **« Availibility » (Haute disponibilité) :** Un système est dit disponible si toute requête reçue par un noeud retourne un résultat. Bien évidemment le noeud en question ne doit en aucun cas être victime de défaillance.
- **« Partition tolerance » (Tolérance à la partition) :** Un système est dit tolérant à la partition s'il continue à répondre aux requêtes de manière correcte même en cas de panne autre qu'une panne totale du système.

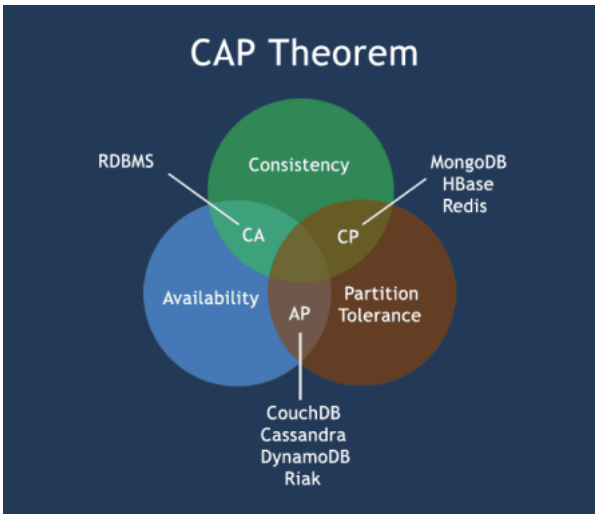

FIGURE 4.3 – Théorème de CAP

Seules deux des trois contraintes peuvent être respectées en même temps. Un des premiers buts des systèmes NoSQL est de renforcer la « scalabilité » horizontale, il faut pour cela que le principe de tolérance au partitionnement soit respecté, ce qui exige l'abandon soit de la cohérence, soit de la haute disponibilité.

Haute disponibilité et tolérance à la partition : (AP)

Les bases ne sont pas forcément cohérentes dans le temps mais la multiplicité des bases sur le réseau permet de garantir une réponse quoi qu'il arrive.

En effet, cela voudrait dire que les utilisateurs n'ont pas forcément tous la même vue à un moment donné. Dans certains cas cela peut poser des problèmes, mais bien souvent ce n'est pas une nécessité. Prenons l'exemple de deux utilisateurs qui sont amis sur Facebook, Marcel et Pierre. Si Marcel partage une photo sur son « mur » et que Pierre au même instant ne la voit pas dans son « fil d'actualité » elle apparaîtra dans les prochaines secondes. Dans ce cas précis, est-ce un problème pour Pierre d'attendre environ une minute pour voir la photo que vient de publier son ami Marcel ?

Cohérence et tolérance à la partition : (CP)

Un tel système de base de données stocke les données dans les nœuds distribués, mais assure également la cohérence de ces données, mais le support n'est pas assez bon pour la disponibilité.

La plupart du temps, les systèmes NoSQL qui garantissent une forte cohérence des données sont architecturés en maître/esclave. Cela veut dire que toutes les écritures doivent être faites sur le serveur maître, et en cas de panne de ce dernier, les opérations d'écriture, de modification et de suppression ne deviennent plus disponibles. Seule la lecture des données sur les serveurs esclaves est possible.

Cohérence et Haute disponibilité (CA) :

Tous les clients du système voient les mêmes données au même instant. Notamment les deux propriétés respectées par les bases de données relationnelles.

4.2.3 Les propriétés de BASE :

Dans la première partie consacrée aux bases de données relationnelles nous avons vu les propriétés ACID auxquelles doivent répondre les SGBD de type relationnel. Les SGBD NoSQL qui, selon le théorème CAP, privilégient la disponibilité ainsi que la tolérance à la partition plutôt que la cohérence, répondent aux propriétés de BASE.

Le principe de BASE est le fruit d'une réflexion menée par Eric Brewer (Théorème de CAP). Les caractéristiques de BASE sont fondées sur les limites que montrent les SGBD relationnelles. Voici sa description :

- **Basically Available :** quelle que soit la charge de la base de données (données/requêtes), le système garantie un taux de disponibilité de la donnée.
- **Soft-state :** La base peut changer lors des mises à jour ou lors d'ajout/suppression de serveurs. La base NoSQL n'a pas à être cohérente à tout instant.
- **Eventually consistent :** À terme, la base atteindra un état cohérent.

FIGURE 4.4 – ACID vs BASE

Ainsi, une base NoSQL relâche certaines contraintes, telles que la synchronisation des réplicas, pour favoriser l'efficacité. Le parallèle ACID / BASE repris du domaine de la chimie permet d'appuyer là où ça fait mal : la concurrence. L'enfer des transactions gérées par les bases de données relationnelles est transformé en paradis pour le temps de réponse en relâchant cette contrainte impossible à maintenir.

4.2.4 Les types de bases de données NoSql :

Dans la mouvance NoSQL, il existe une diversité d'approches classées en quatre catégories. Ces différents systèmes NoSQL utilisent des technologies forts distinctes. Les différents modèles de structure sont décrits comme suit :

Les bases de données orientées clé-valeur :

Souvent assimilé à une « hashmap » distribuée, le système de base de données de type clé/valeur est probablement le plus connu et le plus basique que comporte la mouvance NoSQL. Son principe est extrêmement simple : chaque objet (valeur) est identifié par une clé unique. Celle-ci représente la seule manière de solliciter l'objet.

La communication avec la base de données se résume aux opérations basiques que sont PUT, GET, UPDATE et DELETE. La plupart des bases de données de type clé/valeur disposent d'une interface HTTP REST qui permet de procéder très facilement à des requêtes depuis n'importe quel langage de développement. Ces systèmes affichent des performances exceptionnellement élevées en lecture et en écriture, ainsi qu'une « scalabilité » horizontale étendue, cela vient du fait que ces types de bases sont réduits à un simple accès disque.

Du fait que les opérations possibles soient basiques (simple CRUD), le besoin en « scalabilité » verticale est fortement réduit. Ces systèmes sont souvent utilisés comme dépôts de données si toutefois les besoins en termes de requêtes restent de niveau simple et que l'intégrité relationnelle des données est non significative.

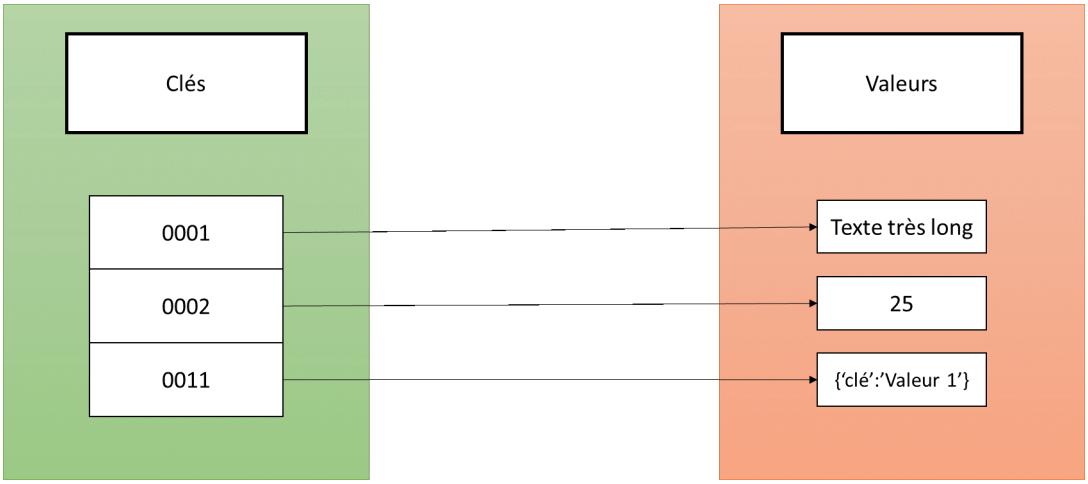

FIGURE 4.5 – base de donnée orientée clé-valeur

Exemple d'applications : *Détection de fraude en temps réel, IoT, e-commerce, gestion de cache, transactions rapides.*

Les bases de données orientées colonnes

La représentation orientée colonnes est celle qui se rapproche le plus des tables dans une base de données relationnelles. Elles permettent d'être beaucoup plus évolutive et flexible puisqu'on peut disposer de colonnes différentes pour chaque ligne. Elles peuvent évoluer dynamiquement en nombre et en nom et contrairement à une table relationnelle (pas de champ « NULL »).

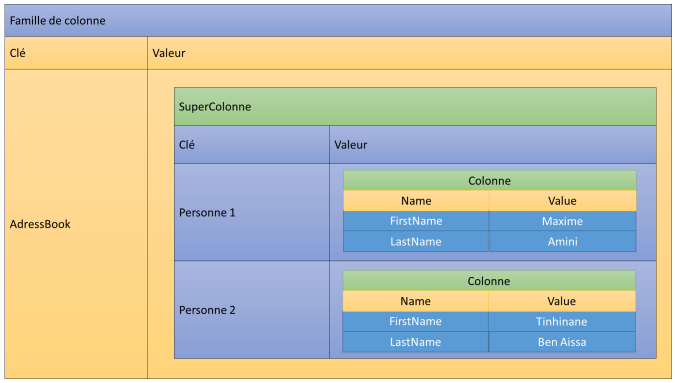

FIGURE 4.6 – base de donnée orientée colonne

- **Column :** c'est l'entité de base qui représente un champ de données. Toutes les colonnes sont définies par un couple clé/valeur
- **Super column :** c'est une colonne qui contient d'autres colonnes
- **Column family :** elle est considérée comme un conteneur de plusieurs colonnes ou super-colonnes

Exemple d'applications : *Comptage (vote en ligne, compteur, etc), journalisation, recherche de produits dans une catégorie, reportage à large échelle.*

Les bases de données orientées documents

Les systèmes de type documentaire sont composés de collections de documents. La représentation en document est une sorte d'extension du concept clé/valeur. La valeur est représentée sous forme de document, ces documents ont une structure arborescente : il contient une liste de champs, un champ est associé à une valeur qui peut, elle même être une liste. Ces documents sont principalement de type JSON ou XML.

Ces bases sont dites « **Schemaless** » ce qui signifie sans schéma défini. Cela veut tout simplement dire qu'il n'est pas nécessaire de définir au préalable les champs dans le document : on peut très bien en rajouter en cours de développement. Les documents peuvent être très différents les uns des autres au sein de la base. Le fait que les documents soient structurés permet d'effectuer des requêtes sur le contenu des objets.

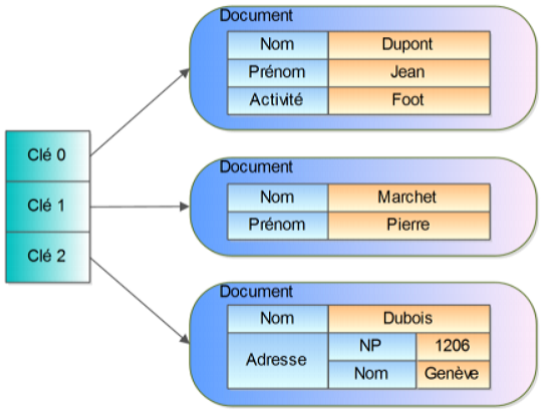

FIGURE 4.7 – base de donnée orientée document

Exemple d'applications : *Gestion de contenu (bibliothèques numériques, collections de produits, dépôts de logiciels, collections multimédia, etc.), framework stockant des objets, collection d'événements complexes, gestion des historiques d'utilisateurs sur réseaux sociaux.*

Les bases de données orientées graphes

Une base de données graphe établit des relations entre les données à l'aide de nœuds et d'arêtes. Le réseau de relation des données est organisé par les points nodaux et leurs connexions les uns avec les autres. Dans le cas de volumes de données aux informations fortement interconnectées, les bases de données graphiques NoSQL présentent une performance considérablement supérieure à celle des bases de données SQL relationnelles.

Elles sont principalement utilisées dans le domaine des réseaux sociaux, pour représenter, par exemple, les relations entre les abonnés sur Twitter ou Instagram.

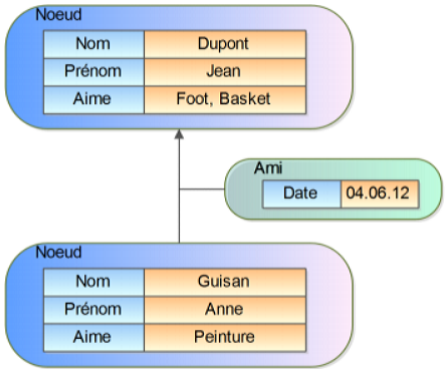

FIGURE 4.8 – base de donnée orientée graphe

Exemple d'applications : *Réseaux sociaux (recommandation, plus court chemin, cluster...), réseaux SIG 12 (routes, réseau électrique, fret...), web social (Linked Data).*

Autres base de données

Il existe plusieurs autres bases de données suivant une approche non relationnelle, mais elles ne sont pas considérées comme des bases de données NoSQL de base, mais plutôt comme des bases de données NoSQL logicielles :

Bases de données d'objets : Les bases de données d'objets utilisent l'idée d'objets dans les langages de programmation et la transforment en systèmes de bases de données.

Exemple : *db4o est un système de gestion de base de données orientée objet Open Source pour des applications Java et .Net 13.*

Bases de données XML : Les bases de données XML sont vraiment des bases de données non SQL. Le langage de requête est XQuery, XPath ou XUpdate et non plus SQL. Ces bases de données permettent de stocker des données au format XML. Par conséquent, la structure de la base de données est hiérarchique. XML est largement utilisé dans les applications Internet, donc une transformation de (anciennement) SQL en XML n'est plus nécessaire.

Exemple : *eXist est une base de données XML basée sur Java open source prenant en charge XQuery et XPath et possède comme CouchDB une interface HTTP RESTful.*

4.2.5 Les avantages NoSQL :

Les bases de données NoSQL ont été créées en réponse aux limitations de la technologie de base de données relationnelle. Comparées aux bases de données relationnelles, les bases de données NoSQL sont plus évolutives et offrent des performances supérieures, et leur modèle de données corrige plusieurs faiblesses du modèle relationnel. Les avantages de NoSQL sont notamment :

— **Gros volume de données « Big data » :** le NoSql est capable de gérer un volume important de données structurées, semi-structurées et non structurées, en effet il est devenu quasi impossible, pour un unique serveur de base de données relationnelle, de répondre aux exigences des entreprises en terme de performance. Aujourd'hui, ces gros volumes de données ne sont plus un problème pour les SGBD de type NoSQL, même le plus grand des SGBD relationnel ne peut rivaliser avec une base NoSQL.

— **Rapidité :** NoSQL n'est pas relationnelle. Pas de schéma de bases avec les contraintes sur les champs. Cela apporte de la flexibilité dans la gestion des données et la rapidité.

— **Modèle de données flexible :** Changer le modèle de données est une vraie prise de tête dans une base de données relationnelle en production. Les systèmes NoSQL sont plus souples en termes de modèles de données, comme dans les catégories clé/valeur et documentaire. Même les modèles un peu plus stricts comme dans la catégorie orientée colonne permettent d'ajouter une colonne sans trop de problème.

— **Solution économique :** Les bases de données NoSQL ont tendance à utiliser des serveurs bas de gamme dont le coût est moindre afin d'équiper les « clusters », tandis que les SGBD relationnels, eux, tendent à utiliser des serveurs ultra puissants dont le coût est extrêmement élevé. De ce fait, les systèmes NoSQL permettent de revoir à la baisse les coûts d'une entreprise en termes de gigabytes ou de transactions par seconde. Cela permet de stocker ainsi que de manipuler plus d'informations à un coût nettement inférieur.

4.2.6 Les inconvénients NoSQL :

Il faut néanmoins être conscient que les avantages apportés par ces systèmes ne sont pas sans contreparties, aucun système n'étant parfait. Les principaux inconvénients de NoSQL sont les suivants :

— Les bases de données NoSQL n'ont pas les fonctions de fiabilité des bases de données relationnelles (ne

— prennent pas en charge les propriétés ACID).

— Afin de soutenir ACID, les développeurs devront implémenter leur propre code, ce qui rendra leurs systèmes plus complexes.

— Support limité : due à la jeunesse des bases de données NoSQL, le support de la communauté est parfois limité.

— Manque de standardisation : Pas de langage "NoSQL" standard sur les différentes bases de données.

— Aucune contrainte et validation à exécuter dans la base de données.

— Interopérabilité : Le passage d'une base de données NoSQL vers une autre n'est pas transparent pour une application.

— Impossibilité d'avoir en même temps la cohérence, disponibilité et tolérance de partition.

4.2.7 Examples BDD NoSql :

A l'heure actuelle, il y a plus de 122 solutions NoSQL tous types confondus (Clé/valeur, Document, Colonne et Graphe) sur le marché. La plupart d'entre elles sont soit sous licence Open Source ou soit proposées en SaaS. Les principaux contributeurs (Facebook, Google, Linkedin,...) de ces produits ont développé leurs outils à l'interne avec pour unique but de répondre à leurs propres besoins et non pas dans le but de les commercialiser. Lorsqu'ils se sont trouvés dans un état d'avancement très important (mise en production à l'interne), ces fameux SGBD ont été libérés et mis à disposition du grand public. Il ne faut pas oublier que la fondation Apache joue un grand rôle dans ces divers projets.

On citera quelques-unes classés selon leurs catégories (Pour avoir un plus large aperçu, le site nosql-database.org répertorie tous les produits de type NoSQL) :

BDDs orientés clé-valeur :

- **REDIS**.
- ORACLE NOSQL.
- **VOLDEMORT**.
- RIAK.
- INFINISPAN.
- HAZELCAST.

1. Redis : Est un projet Open Source de type clé/valeur sous licence BSD. Il dispose de plus de fonctionnalités que la majeure partie des autres solutions du même type. Redis prend en charge certains langages C++, PHP, Ruby, Python, Perl, Scala, etc. Redis est fait en langage C.

En revanche, un des seuls points négatifs à noter est que Redis ne fournit pas de réel mécanisme de partitionnement, mais uniquement une réplication de type maître/esclave. Malgré cela Redis reste très performant. Il faut aussi ajouter que le projet est sponsorisé par la société VM Ware, ce qui rend la solution crédible et pérenne.

Redis offre les caractéristiques suivantes :

- ✓ Basculement automatique (sans intervention humaine).
- ✓ Conserve sa base de données entièrement en mémoire.
- ✓ Les transactions sont un moyen d'envoyer en une seule opération un ensemble d'actions.
- ✓ Répliquer les données à un nombre quelconque d'esclaves.
- ✓ Possibilité de contrôler la durée de vie d'une donnée dans la base.
- ✓ Prise en charge de la publication / abonnement.

FIGURE 4.9 – Logo Redis

2. Voldemort : Est un projet Open Source de type clé/valeur sous licence Apache 2.0. Il a été nommé d'après un personnage du célèbre film « Harry Potter » et offre les caractéristiques suivantes :

- ✓ Les données sont automatiquement répliquées sur de nombreux serveurs
- ✓ Les données sont automatiquement partitionnées afin que chacun des serveurs obtienne un sous-ensemble d'un ensemble de données
- ✓ L'échec d'un serveur est géré de manière transparente
- ✓ Les données sont « versionnées »
- ✓ Chaque noeud est indépendant des autres. De plus, chaque noeud du système atteint de hauts niveaux de performance de l'ordre des 20Ko d'opérations par seconde.

Le projet Voldemort développé par les ingénieurs de LinkedIn a été mis en production à l'interne en 2009. Chez LinkedIn, Voldemort est utilisé pour certaines opérations de stockage de gros volumes de données, ce qui exige des performances accrues auxquelles les solutions de stockage simple ne peuvent répondre.

FIGURE 4.10 – Logo Voldemort

BDDs orientés colonnes :

- **HBASE.**
- **CASSANDRA**
- ACCUMULO.
- HYPERTABLE.

1. HBASE : HBase est une base de données distribuée et non relationnelle qui est conçue pour la base de données BigTable par Google. L'un des principaux objectifs de HBase est d'héberger des milliards (de lignes x millions de colonnes). On peut ajouter des serveurs à tout moment pour augmenter la capacité. Et de multiples nœuds maîtres assureront une haute disponibilité des données. Il est composé en Java 8. Il est autorisé sous Apache et est accompagné d'une API Java simple d'utilisation pour l'accès des clients.

Caractéristiques :

- ✓ Prise en charge de l'échec automatique.
- ✓ Linéairement évolutif.
- ✓ Permet la réplication des données.
- ✓ S'intègre à Hadoop, à la fois comme source et comme destination.

FIGURE 4.11 – Logo HBASE

2. CASSANDRA : Cassandra est une solution de type orienté colonnes. Mise en Open Source par Facebook en 2008, c'est la fondation Apache qui a repris le flambeau et l'a mise sous licence Apache 2.0. Les caractéristiques de cette solution sont les suivantes :

- ✓ flexibilité du schéma : chaque « table » est libre de suivre sa propre structure.
- ✓ représentation spatiale de la donnée (cela permet de traiter un grand nombre de lignes)
- ✓ « scalabilité » horizontale : s'il y a besoin de plus de puissance, on rajoute un serveur au cluster.
- ✓ plusieurs niveaux de stratégie de cohérence des données en écriture.
- ✓ réplication multi data-center.

Cassandra est utilisée par de nombreux acteurs de premier plan comme Facebook, Twitter et Cisco. Ceci garantit une certaine pérennité du produit. De plus une vaste communauté d'utilisateurs fait partager ses connaissances du produit et le font évoluer.

FIGURE 4.12 – Logo CASSANDRA

BDDs orientés documents :

- **MONGODB.**
- **COUCHDB.**
- RAVENDB.
- JERRASTORE.

1. MONGODB : MangoDB est un système de la catégorie orientée documents, le plus populaire de sa catégorie. Il est écrit en C++ et Open Source sous licence AGPL v3.0. Cette solution a été développée par la société 10gen. Voici les caractéristiques du produit :

- ✓ flexibilité du schéma : chaque document est libre de suivre sa propre structure.
- ✓ format de document JSON.
- ✓ garantit la « scalabilité » horizontale (réplication et « sharding »).
- ✓ support de recherche full-text, géo-spatiale et Map-Reduce.
- ✓ requêtes sur le contenu des documents.
- ✓ une large palette de pilotes est disponible pour divers langages de programmation.
- ✓ bonne documentation.

FIGURE 4.13 – Logo MONGODB

2. COUCHDB : CouchDB est une base de données NoSQL Open Source qui utilise JSON pour stocker des informations et JavaScript comme langage de requête. CouchDB applique un type de système de contrôle multi-versions pour éviter le blocage du fichier DB pendant l'écriture. C'est autorisé sous Apache. Il est classé 1er sur la liste Best NoSQL Database 2016 pour sa popularité. Caractéristiques :

- ✓ Cartographier/réduire la liste et afficher.
- ✓ Assurer la sécurité au niveau de la base de données.
- ✓ L'authentification s'ouvre via un cookie de session comme une application web.
- ✓ JSONP gratuitement.
- ✓ Suivre le stockage des documents.
- ✓ Prise en charge des propriétés ACID.
- ✓ Fournir la forme la plus simple de réplication.
- ✓ Interface utilisateur graphique basée sur un navigateur pour gérer vos données, vos autorisations et votre configuration.

BDDs orientés graphes :

- **Neo4J**
- INFINATEGRAPHE
- INFOGRID.
- HYPERGRAPH DB
- ALLEGROGRAPH.

1. Neo4J : Neo4j est un système de base de données orienté graphe. C'est un projet Open Source sous licence GPLv3. Sa première version est sortie en 2007. Ce type de solution est utilisé dans le monde des réseaux sociaux (Ex : amis sur Facebook). Voici quelques caractéristiques du produit :

- ✓ Transaction complètement ACID.
- ✓ Disponibilité haute.
- ✓ Possibilité d'avoir plus de 64 milliards de noeuds/relations/propriétés sur une JVM.
- ✓ Solution robuste (7ans en production sans interruption).
- ✓ Intégration d'API (librairie Java).

Neo Technologie, qui est la société de développement de la base Neo4j, propose un excellent support technique ainsi qu'une documentation complète.

FIGURE 4.14 – Logo Neo4J

D'autres base de données peuvent proposées plusieurs services en même temps comme :

Amazon DynamoDB : DynamoDB utilise un modèle de base de données NoSQL, qui n'est pas relationnel, ce qui permet d'avoir des documents, des graphiques parmi ses modèles de données. Chaque requête DynamoDB est exécutée par une clé primaire identifiée par l'utilisateur, qui identifie de manière unique chaque élément. Il libère également les clients du fardeau de l'exploitation et de la mise à l'échelle d'une base de données distribuée. Ainsi, le provisionnement matériel, l'installation, la configuration, la réplication, le patch logiciel, la mise à l'échelle des clusters, etc. sont gérés par Amazon.

Caractéristiques :

- ✓ Haute évolutivité.
- ✓ Plage de hachage pour l'indexation d'une plage de valeurs.
- ✓ Stockage des données dans les partitions.
- ✓ Utilise JSON comme protocole de transport et non comme format de stockage.

4.3 Ver le NewSQL la base de données moderne

NewSQL est un stockage distribué et potentiellement entièrement en mémoire et pouvant être requêté classiquement par une interface SQL. NewSQL est tiré du monde NoSQL mais reste différent. Comme NoSQL il s'agit d'une nouvelle architecture logicielle qui propose de repenser le stockage des données. Elle profite des architectures distribuées, des progrès du matériel et des connaissances théoriques depuis 35 ans. Mais contrairement à NoSQL elle permet de conserver le modèle relationnel au coeur du système.

NewSQL est né de la rencontre de 3 types d'architecture, relationnelle, non-relationnelle et grille de données appelée également cache distribué, comme indiqué dans la Figure ci-dessous. En effet il se positionne comme un stockage distribué conçu dans le prolongement des architectures NoSQL, pour des accès transactionnels à fort débit, au moyen d'une interface SQL. D'un point de vue évolutivité, il se situe en tant que concurrent direct des solutions NoSQL. Mais contrairement à ces solutions il conserve une interface relationnelle via le SQL, ce qui est l'une de ses forces. Enfin la plupart des solutions NewSQL proposent un stockage en mémoire. Ce stockage en mémoire distribué sur plusieurs machines sous forme de grille de données est largement utilisé depuis une dizaine d'années dans les environnements où une faible latence est critique, notamment dans certaines applications des banques d'investissement. Les solutions NewSQL partagent ainsi un positionnement intermédiaire entre les solutions NoSQL et les grilles de données.

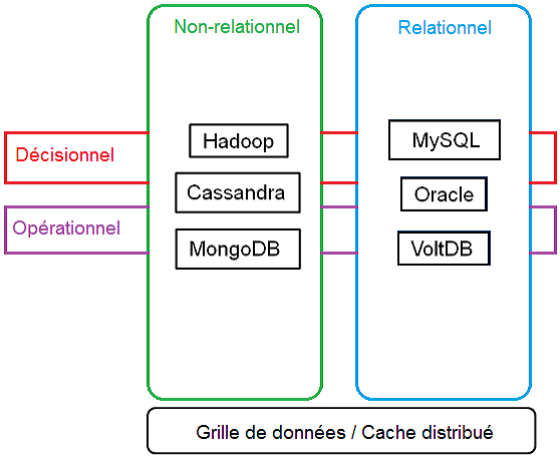

FIGURE 4.15 – Naissance du NewSQL à partir de 3 architectures

4.3.1 L'architecture NewSQL :

L'architecture NewSQL reprend des expériences antérieures du SQL relationnel et du NoSQL plusieurs caractéristiques, tout en ayant certaines particularités en termes de choix et d'avantages :

1. Le choix d'une interface SQL et d'un schéma relationnel.
2. Le schéma relationnel avec des limitations pour faciliter la distribution des données et des traitements.
3. La distribution et la réplication des données pour assurer l'évolutivité et la résilience.

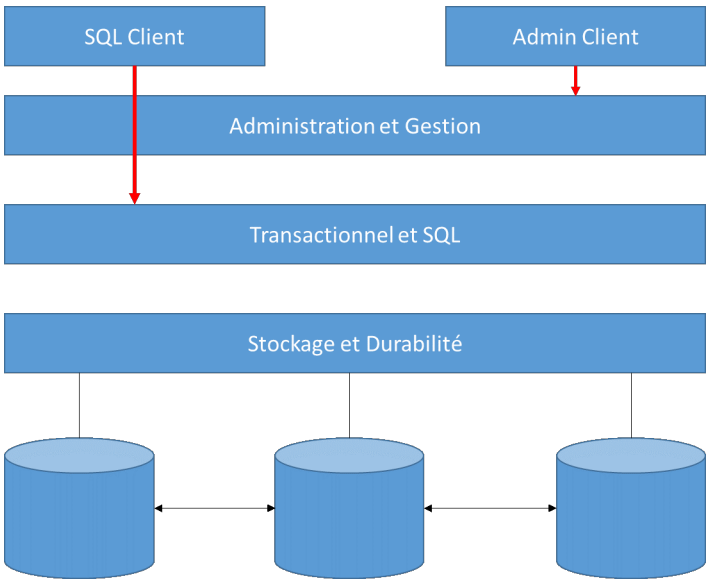

FIGURE 4.16 – L'architecture d'une base de données NewSQL populaire NueDB.

4.3.2 Les avantages de la solution NewSQL :

La solution NewSQL présente des avantages intéressants en termes de performances par rapport à ses prédécesseurs :

1. Elle utilise le SQL comme langage commun de requêtes.
2. Elle présente une architecture qui a de meilleures performances par noeud que les solutions classiques
3. type SGBD relationnel.
4. Elle minimise la complexité des applications tout en améliorant la consistance des données et en fournissant un support transactionnel complet.
5. Elle est compatible avec les outils de travail du standard SQL.
6. Elle fournit des analyses plus riches des traitements.
7. Elle est compatible avec l'architecture distribuée des Clusters.
8. Elle fournit un traitement plus performant des données en mémoire.

4.3.3 Les limites de la solution NewSQL :

La solution NewSQL est confrontée à quelques limitations l'empêchant d'atteindre un niveau de maturité suffisant :

1. Son architecture de calcul en mémoire, dit In-Memory, ne fonctionne pas au-delà de quelques TO de données.
2. Cette architecture nécessite un matériel spécifique avec des capacités importantes de stockage en mémoire, ce qui revient très onéreux.
3. Malgré son attitude à vouloir intégrer le modèle relationnel, le NewSQL fournit un accès limité aux outils de travail du standard SQL.

4.3.4 Résumé :

Une base de données NewSQL conserve la structure classique en colonnes mais fait appel à différents procédés pour conserver la rapidité même sur de larges volumes. En revanche, cette technologie n'a pas le recul suffisant pour aller plus loin et n'a pas encore pu suffisamment faire ses preuves. Par conséquent les entreprises sont encore réticentes à l'adoption de cette toute nouvelle architecture.

4.3.5 Exemples des bases de données NewSQL :

NuoDB : C'est un SGBD distribué qui pourrait être décrit comme une base de données à faible latence. Il permet à l'utilisateur d'interagir avec lui de manière transactionnelle en prenant en charge les opérations ACID et la réplication des données.

FIGURE 4.17 – Logo NuoDB.

VoltDB : Il s'agit d'une base de données distribuée en mémoire qui est considérablement plus rapide que les bases de données SQL. C'est une base de données flexible qui prend en charge le stockage JSON. Il est le mieux adapté aux applications à lecture fréquente et à faible fréquence d'écriture.

FIGURE 4.18 – Logo VoltDB.

Clustrix : c'est une base de données distribuée qui prend en charge l'analyse en temps réel, elle est optimisée pour les transactions massives. Il prend en charge certains outils de BI 23 et une récupération rapide des données.

FIGURE 4.19 – Logo Clustrix.

Conclusion :

Bibliographie