L'ECOSYSTEME HADOOP

Par Christophe Parageaud

IPPON
Digital . Technologies . Hosting

LIVRE BLANC
JANVIER 2016

IPPON

Digital . Technologies . Hosting

WWW.IPPON.FR
BLOG.IPPON.FR

PARIS
NANTES
BORDEAUX
WASHINGTON, DC
RICHMOND, VA
NEW YORK, NY

TABLE DES MATIÈRES

1 L'AUTEUR

Christophe PARAGEAUD est architecte Java EE et manager technique chez Ippon Technologies depuis 2007. Il travaille sur des missions d'architecture et de support. Fort de 17 ans d'expérience, dont 12 en Java, Christophe est spécialisé dans les architectures distribuées et les applications hautes performances.

SES ARTICLES

- TAMTAM – JAVA 8 FUNCTIONAL PROGRAMMING: don't neglect optimisations!
- MapReduce et les grilles de données ou Hadoop sans Hadoop
- JAVA ERGONOMICS: Faut-il encore tuner la JVM ?
- LIVRE BLANC IPPON: Grilles de données mémoire
- APACHE FLINK ET SPARK: redondance?
- MONGODB V3: la révolution?
- RETOUR SUR L'ANNÉE 2014: Big Data, Mobilité, Agilité, Cloud

2 A PROPOS D'IPPON TECHNOLOGIES

Ippon Technologies est une société innovante créée en 2003 par un sportif de Haut Niveau et un polytechnicien avec pour ambition de devenir le cabinet de conseil en technologies leader sur les solutions Digitales, Cloud et BigData.

Ippon accompagne les entreprises dans le développement et la transformation de leur système d'information avec des applications performantes et des solutions robustes. Ippon propose une offre de services à 360° pour répondre à l'ensemble des besoins en innovation technologique :

Conseil, Design, Développement, Hébergement et Formation.
Nos équipes tirent le meilleur de la technologie pour transformer rapidement les idées créatives de nos clients en services à haute valeur ajoutée.

Elles accompagnent à la fois les grands groupes (Axa, EDF, Société Générale, LVMH,...) et les champions français de l'industrie numérique (CDiscount, LesFurets.com, Aldebaran, Mirakl,....) dans leurs innovations.

CONSEIL

DESIGN

RÉALISATION

HÉBERGEMENT

Nous avons réalisé, en 2014, un chiffre d'affaires de 16 M€ en croissance organique de 15%. Nous sommes aujourd'hui un groupe international riche de 200 consultants répartis en France et aux USA.

2.1. Nos Solutions

Nous accompagnons nos clients sur la mise en oeuvre de projets Data ou NoSQL avec une offre structurée pour faciliter les 3 étapes d'un projet Data:

LAUNCH

- Poser une architecture Data pour répondre aux besoins métiers (Lambda, Kappa, SMACK, ...)
- Valider les hypothèses techniques et métiers de cette architecture
- Travailler de pair avec notre eco-système de partenaires éditeurs
- Démarche : Lean StartUp

BUILD

- Construire les projets Data en se basant sur nos REX et dans une logique Quick-Win
- Faciliter l'intégration des équipes de Data Scientist
- Former les équipes sur le changement de paradigme (NoSQL, Spark, moteurs d'indexation [Lucene, Solr, Elasticsearch], Scala...
- Démarche : Scrum/XP

RUN

- Outiller les projets dans une logique DevOps
- Industrialiser la plateforme pour pouvoir assurer un service 24/24 et 7/7 si besoin
- Accompagner les équipes Ops pour opérer ces nouvelles technologies
- Démarche : Kanban

2.2. Nous contacter

Vous pouvez retrouver toutes nos coordonnées sur
www.ippon.fr/contact,
nous joindre par mail à l'adresse
accueil@ippon.fr,
ou contacter une de nos agences directement par téléphone :

RICHMOND, VA
Ippon USA
(844-477-6687)

PARIS
01 46 12 48 48
43 avenue de la Grande Armée - 75116 Paris

NEW YORK, NY
Ippon USA
(844-477-6687)

NANTES
02 40 48 28 06
1 Rue Du Guesclin - 44000 Nantes

WASHINGTON, DC
Ippon USA
(844-477-6687)

BORDEAUX
05 35 54 62 26
61 cours de l'Intendance - 33000 Bordeaux

3 LICENCE

4 PRÉSENTATION DE L'ÉTUDE

4.1. Big Data

Le terme Big Data reflète l'augmentation exponentielle de la volumétrie des données d'un Système d'Information que l'on a pu connaître ces dernières années.
En outre, les nouveaux challenges auxquels sont confrontés les entreprises sont la vitesse et le coût du traitement de ces données.
En plus de ces challenges techniques, il faut ajouter une autre plus value essentielle qui est la valorisation de ces données.

C'est cette caractéristique économique qui va justifier la mise en œuvre des technologies du Big Data, elle mesure la plus value pour l'entreprise et le retour sur investissement.
Rapidement il a été constaté que les solutions traditionnelles ne pourraient pas convenir (en tout cas pour un coût mesuré).
Il a donc fallu inventer de nouveaux systèmes et c'est Hadoop qui a émergé en premier.

Le fait d'abandonner le mode relationnel des bases de données traditionnelles, pour un mode clé/valeur ou colonnes, a permis de diviser la volumétrie et les temps de traitements (on parle d'une division par deux du volume de stockage et par dix des temps de traitement).

4.2. Hadoop

En 2004, Google a publié un article présentant son algorithme de calcul à grande échelle, MapReduce, ainsi que son système de fichier en cluster, GoogleFS. Rapidement (2005) une version open source voyait le jour sous l'impulsion de Yahoo.

Aujourd'hui il est difficile de se retrouver dans la jungle d'Hadoop pour les raisons suivantes :

- Ce sont des technologies jeunes.
- Beaucoup de buzz et de communication de sociétés qui veulent prendre le train Big Data en marche.
- Des raccourcis sont souvent employés (non MapReduce ou un équivalent n'est pas suffisant pour parler d'Hadoop).
- Beaucoup d'acteurs différents (des mastodontes, des spécialistes du web, des start-up, ...).

Dans une distribution Hadoop on va retrouver les éléments suivants (ou leur équivalence) : HDFS, MapReduce, ZooKeeper, HBase, Hive, Oozie, Pig, Sqoop, ...

Ces solutions sont des projets Apache et sont donc disponibles unitairement mais l'intérêt d'un package complet est évident :

- **compatibilité entre les composants,**
- **simplicité d'installation,**
- **support, ...**

Dans cet article on évoquera les trois distributions majeures que sont Cloudera, Hortonworks et MapR, toutes les trois se basant sur Apache Hadoop.

On peut toutefois les distinguer en fonction de la distance qu'elles prennent avec cette base commune:

- **MapR : noyau Hadoop mais packagé et enrichi de solutions propriétaires.**
- **Cloudera : fidèle en grande partie sauf pour les outils d'administration.**
- **Hortonworks : fidèle à la distribution Apache et donc 100% open source.**

Il existe d'autres distributions, voire des offres cloud, mais qui n'offrent pas l'ensemble des fonctionnalités d'une plate forme Hadoop ou ne sont pas open source (ou a minima gratuites).

Reste le cas de la plateforme Pivotal HD qui offre une version communautaire.

Cette distribution est entièrement basée sur Apache Hadoop auquel il faut ajouter deux produits maison récemment rendus open source :

- HAWQ : Système d'interrogation Hadoop de type SQL.
- GemFire : Grille de données mémoire.

Cette solution ne sera pas étudiée ici.

5 PRÉSENTATION HADOOP

Ce chapitre a pour but de vous permettre de mieux appréhender Hadoop dans sa globalité et de mieux cerner les enjeux liés.

5.1. Les 5 V du Big Data

Même si les cas d'utilisation d'Hadoop/Big Data sont larges, les cinq définitions suivantes caractérisent les applications Big Data.

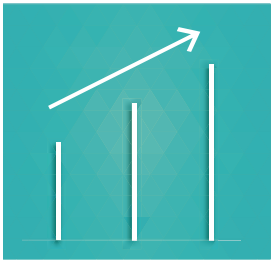
5.1.1. Volume

Les entreprises font face à une augmentation exponentielle des données (jusqu'à plusieurs milliers de téraoctets):

- **Logs,**
- **Réseaux sociaux,**
- **e-commerce,**
- **Catalogue produit,**
- **Analyse des données,**
- **Monitoring,...**

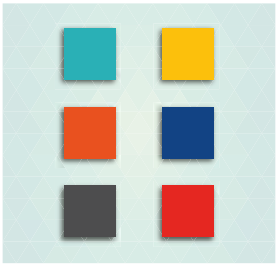
Les technologies traditionnelles (Business Intelligence, Bases de données) n'ont pas été pensées pour de telles volumétries.

Une des caractéristiques du Big Data est sa capacité à traiter d'énormes quantités de données pour un coût mesuré.



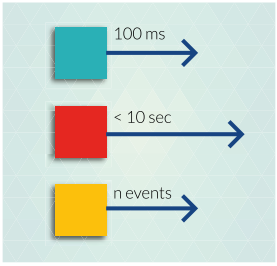
VOLUMETRIE

Une volumétrie importante qui ne peut être traitée par les solutions classiques (Téra octets, Péta octets).
De plus cette volumétrie est croissante.



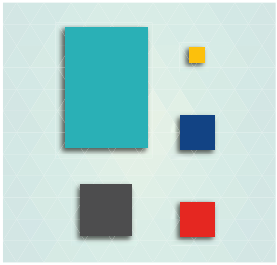
VARIABILITE

Des données hétérogènes, JSON, CSV, texte, ...
Des formats non encore connus.



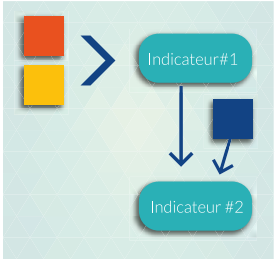
VELOCITE

Capacité à traiter des données en quasi temps réel.
Analyse prédictive/approximative sur un échantillon de données.



VERACITE

Toutes les données n'ont pas la même valeur, le même poids dans l'algorithme de traitement.



VALEUR

Les données sont transformées pour créer de la valeur pour les clients et l'entreprise.

5.1.2. Variété/variabilité

Les données à traiter dans une entreprise sont de natures multiples.
Exemple de données structurées:

- Flux RSS, XML,
- JSON,
- Bases de données.

Ce à quoi peuvent s'ajouter des données non structurées:

- Mails,
- Pages web,
- Multi-media (son, image, video, ...).

Ces données non structurées peuvent faire l'objet d'une analyse sémantique permettant de mieux les structurer et les classer, entraînant une augmentation du volume de données à stocker. La solution doit être évolutive car les formats de données ne sont pas tous actuellement connus (voir par exemple comment le format JSON a supplanté XML très rapidement).

5.1.3. Vélocité

Dans certains cas, l'accès et le partage des données doivent se faire en temps réel (on verra par la suite que ce n'est pas toujours vrai pour Hadoop). Toutes les entreprises n'ont pas la même échelle de temps entre traitements batch et traitements en temps réel : millisecondes, secondes, minutes, ...
La vitesse de traitement de certaines solutions permet d'offrir des capacités temps réel d'analyse et de traitements et donc un retour utilisateur plus rapide.

5.1.4. Valeur

C'est un point essentiel du Big Data car il va permettre de monétiser les données d'une entreprise. **Ce point n'est pas une notion technique mais économique. On va mesurer le ROI de la mise en œuvre du Big Data et sa capacité à s'auto financer par les gains attendus pour l'entreprise.**

5.1.5. Véracité

C'est la capacité à disposer de données fiables pour le traitement.
On va s'intéresser à la provenance des données afin de déterminer s'il s'agit de données de confiance.
En fonction du critère de confiance, on accordera plus ou moins d'importance à la donnée dans les chaînes de traitement. Le critère de confiance ne mesure pas uniquement la méfiance vis à vis de la source mais surtout l'importance que l'on souhaite lui donner.
Toutefois, parmi les données dont il faut éventuellement se méfier on trouve les données des réseaux sociaux dont la provenance et l'objectivité est difficile à évaluer.

5.2. Hadoop versus SGBDR

Parce qu'au sein d'un SI, un seul type de solution ne peut convenir à tous les besoins, il est important de connaître les grandes différences entre le monde SQL et le monde Hadoop.
De plus, on observe un rapprochement entre le monde SQL traditionnel et le monde NoSQL au travers des éditeurs qui proposent de plus en plus ces deux solutions ainsi que des points d'accès entre ces deux mondes.

Enfin le monde Hadoop cherche de plus en plus à s'interconnecter au SI et donc au monde SQL afin de minimiser le coût de la conduite du changement

La matrice suivante présente les grandes différences entre les technologies des bases de données et Hadoop.

	SGBD-R	HADOOP
VOLUMÉTRIE	Giga octets	Péta octets
TYPE D'ACCÈS	Temps réel et batch	Batch
MISES À JOUR	Lectures et écritures multiples	Une seule écriture, lectures multiples
STRUCTURE	Schéma pré défini	Schéma évolutif
INTÉGRITÉ	Forte	Faible
SCALABILITÉ	Non linéaire	Linéaire

5.3. Hadoop versus NoSQL

Cette comparaison est inévitable même si elle n'a pas de sens tant les deux technologies sont complémentaires.

Tout d'abord, ces deux technologies font parties de la famille Big Data et ont de nombreux points communs :

- Scalabilité
- Capacités à traiter des formats hétérogènes
- Tire profit des «commodity hardware» : serveurs basse ou moyenne gamme
- Capacité à traiter la donnée là ou elle est stockée (« Data Locality »)

5.3.1. Points différenciants

5.3.1.1. Vitesse de traitement

NoSQL est plus adapté pour des traitements rapides de type temps réel ou des traitements interactifs.
Hadoop intègre une base NoSQL (HBase) qui est souvent utilisée pour ses capacités d'analyse. Le socle Hadoop étant utilisé pour le stockage et la transformation de données.

5.3.1.2. Framework de traitements

Les solutions NoSQL offrent peu ou pas de capacités de traitements natives à la plateforme (MapReduce est parfois possible ainsi que des fonctions d'agrégations)mais on est loin de la variété offerte par la plateforme Hadoop. En clair NoSQL est une technologie de stockage alors qu'Hadoop est une solution de stockage et de traitement.

5.3.1.3. Capacités de stockage

Même si NoSQL peut stocker des volumétries très importantes, Hadoop conserve un avantage car il dispose d'un système spécifiquement conçu pour ces volumes (HDFS).
De plus, sa scalabilité est linéaire, ce qui n'est pas le cas de toutes les solutions NoSQL dont certaines ont une architecture centralisée.

5.3.2. Conclusion

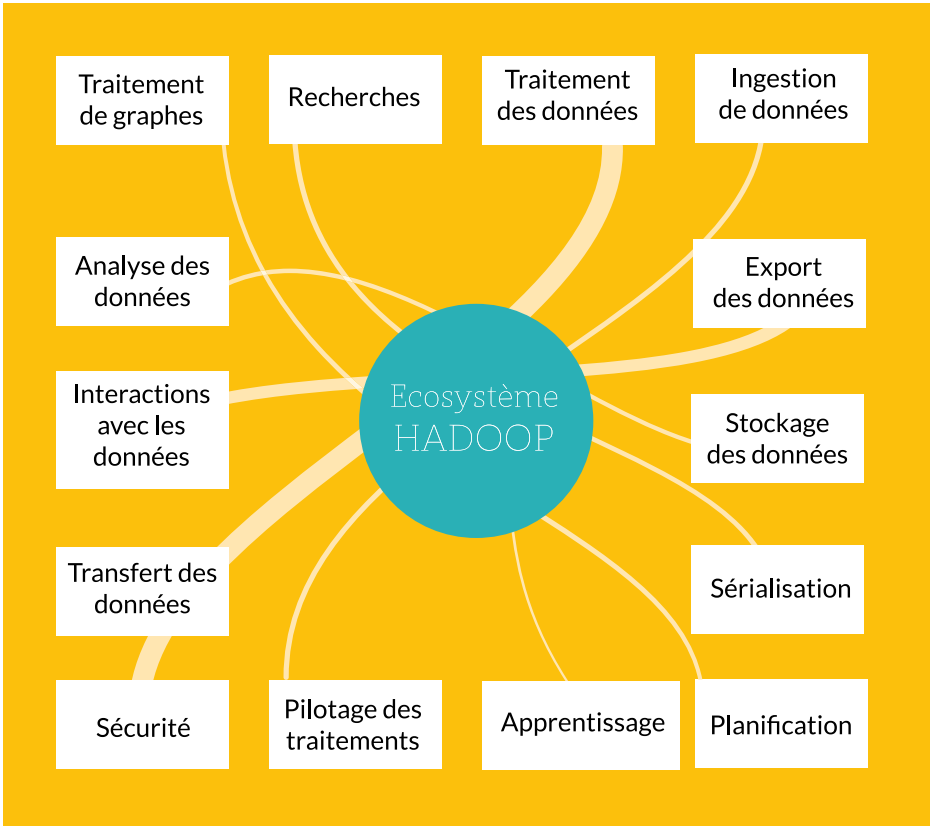
En réalité, ces deux technologies sont complémentaires et l'on va souvent les trouver associées.

- **NoSQL pour les traitements temps réel.**
- **Hadoop pour les traitements batchs et le stockage.**
- **Eventuellement une consolidation entre les deux systèmes (architecture lambda).**

5.4. Catégories des solutions Hadoop

Le schéma suivant permet de se rendre compte de la complexité et du nombre de solutions offertes par une plateforme Hadoop.
Il est rare que tous ces besoins soient présents dans une entreprise mais il est rassurant de démontrer la complétude de la plateforme.

En effet la plateforme Hadoop est parmi les plus complètes du marché du Big Data, de nombreuses solutions sont compatibles et peuvent étendre les fonctionnalités initiales de stockage et de traitement.


Traitement de graphes
Recherches
Traitement des données
Ingestion de données
Analyse des données
Export des données
Interactions avec les données
Ecosystème HADOOP
Stockage des données
Transfert des données
Sérialisation
Sécurité
Pilotage des traitements
Apprentissage
Planification

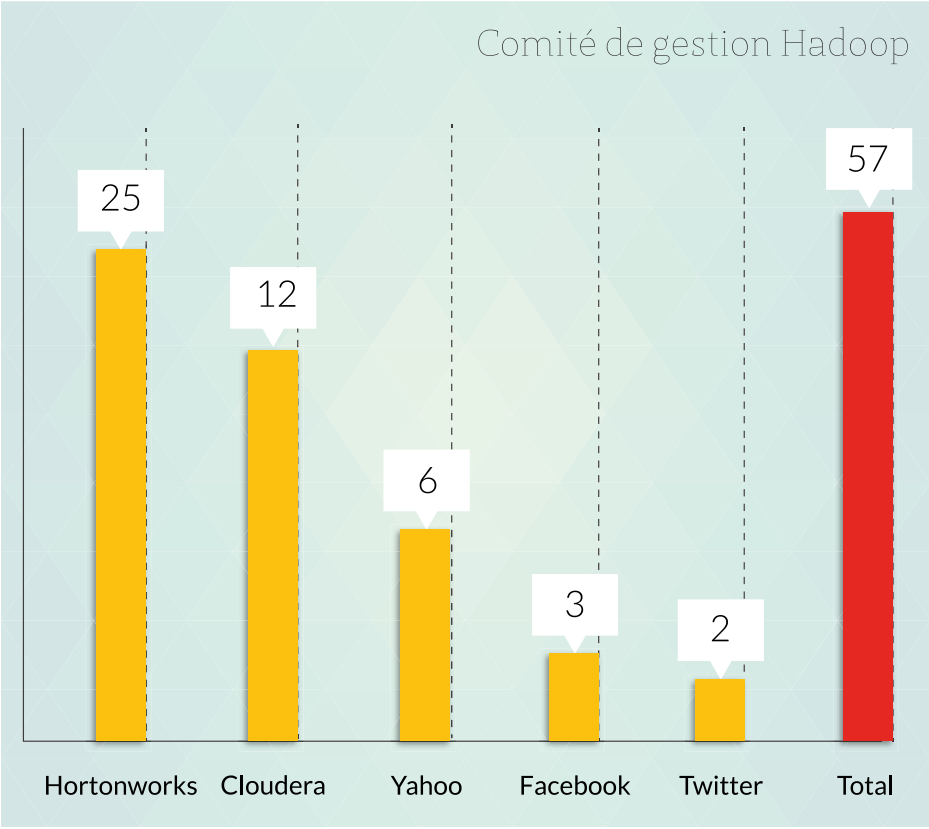
5.5. Liste des membres actifs Hadoop

Hadoop est géré, comme tout projet Apache, par un comité constitué des personnes les plus influentes de l'écosystème.

Les schémas suivants démontrent l'implication des différentes sociétés.

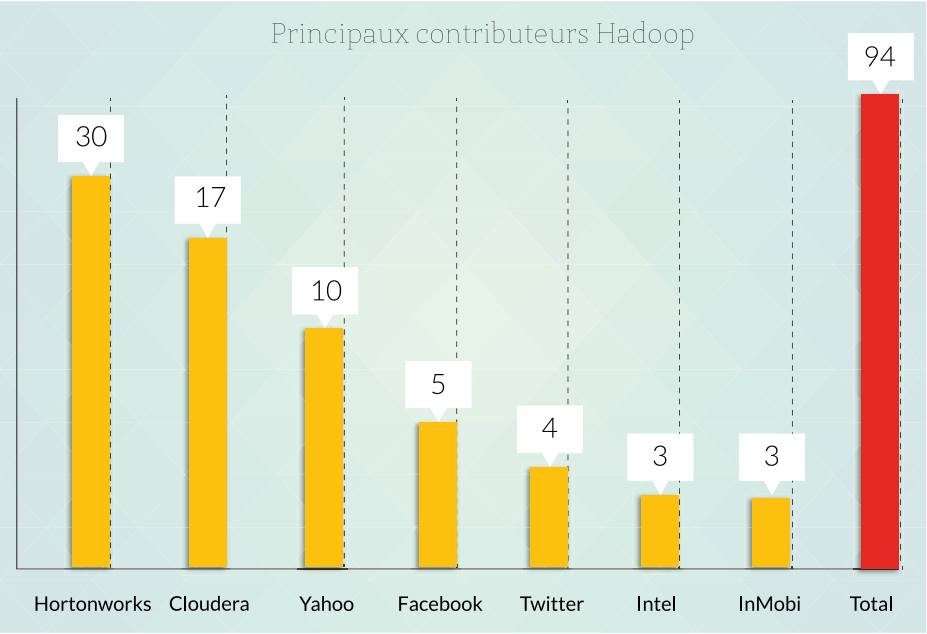
5.5.1. Responsables du projet Hadoop

Liste des principaux responsables de la vision, de l'orientation et du contenu de la distribution Hadoop.



5.5.2. Principaux contributeurs

Liste des principaux contributeurs actifs à l'ensemble des solutions Hadoop.



6 LE CŒUR : HADOOP KERNEL

Apache Hadoop est un framework qui va permettre le traitement de données massives sur un cluster allant de une à plusieurs centaines de machines. Apache Hadoop est un projet open source (Apache v2 licence).

Hadoop est écrit en Java et a été créé par **Doug Cutting** et **Michael Cafarella** en 2005 (après avoir créé le moteur de recherche Lucene, Doug travaillait alors pour Yahoo sur son projet de crawler web Nutch).

Apache Hadoop va gérer la distribution des données au cœur des machines du cluster, leurs éventuelles défaillances mais aussi l'agrégation du traitement final.

L'architecture est de type « Share nothing » : aucune donnée n'est traitée par deux nœuds différents même si les données sont réparties sur plusieurs noeuds (principe d'un noeud primaire et de noeuds secondaires).

Apache Hadoop est composé de quatre éléments :

- **Hadoop Common** : Ensemble d'utilitaires communs aux autres éléments.
- **Hadoop Distributed File System (HDFS)** : Un système de fichiers distribué pour le stockage persistant des données.
- **Hadoop YARN** : Un framework de gestion des ressources et de planification des traitements.
- **Hadoop MapReduce v2** : Un framework de traitements distribués basé sur YARN.

6.1. HDFS (Hadoop Distributed File System)

HDFS est un système de fichiers Java utilisé pour stocker des données structurées ou non sur un ensemble de serveurs distribués.
C'est un système distribué, extensible et portable développé par le créateur d'Hadoop à partir du système développé par Google (GoogleFS).
Écrit en Java, il a été conçu pour stocker de très gros volumes de données sur un grand nombre de

HDFS s'appuie sur le système de fichiers natif de l'OS pour présenter un système de stockage unifié reposant sur un ensemble de disques et de systèmes de fichiers hétérogènes.

Un cluster HDFS repose sur deux types de composants majeurs :

1. **NameNode** : Ce composant gère les fichiers et les répertoires du cluster de manière centralisée.
Il est unique mais dispose d'une instance de backup afin d'assurer la continuité du fonctionnement du cluster Hadoop en cas de panne.

2. **DataNode** (nœud de données) : ce composant stocke et restitue les blocs de données (données primaires) et abrite des copies des autres instances.
Par défaut, les données sont stockées sur trois nœuds différents : dans deux nœuds proches (même machine ou rack) et l'autre sur un nœud plus distant.
Le RAID est par conséquent inutile sur un cluster HDFS.

La consistance des données est basée sur la redondance. Une donnée est stockée sur au moins X volumes différents.

1. Node (Master/slave) : Dans une architecture Hadoop chaque membre pouvant traiter des données est appelé Node (Noeud).

2. Un seul d'entre eux peut être master même s'il peut changer au cours de la vie du cluster, il s'agit du NameNode.

3. Le NameNode est responsable de la localisation des données dans le cluster.

4. Le NameNode est donc un Single Point Of Failure (SPOF) dans un cluster Hadoop. C'est pourquoi il est conseillé d'en démarrer au moins deux.

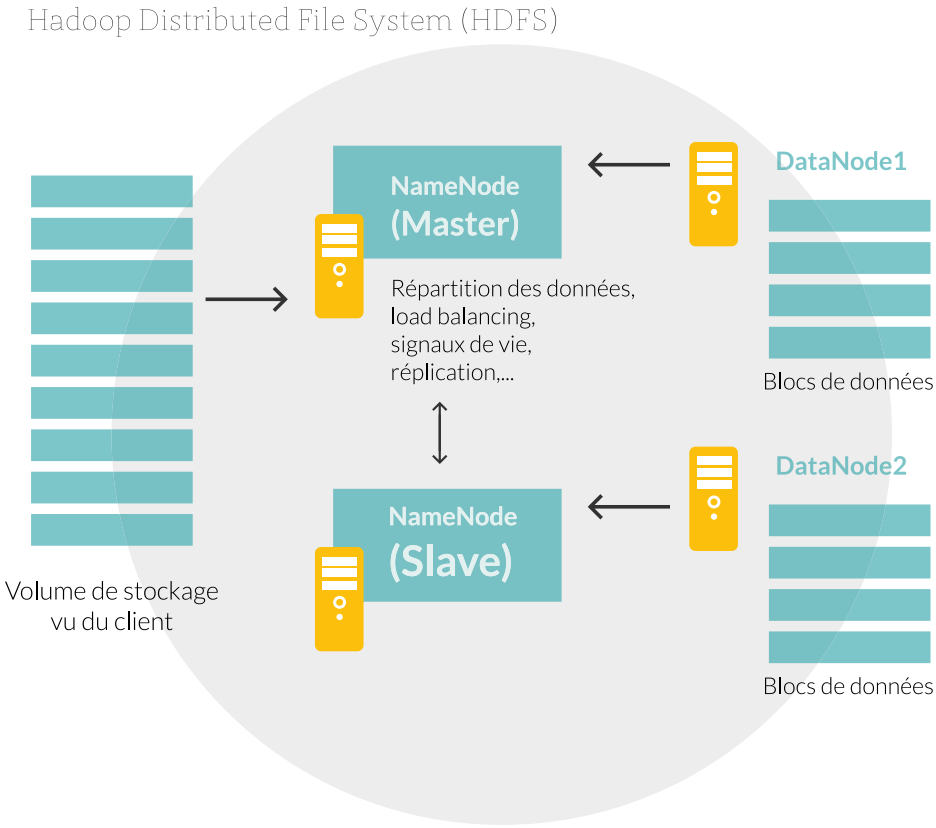
5. Depuis Hadoop 2.0, la "promotion" est automatique en cas de défaillance du NameNode principal.

6. Les autres nœuds, stockant les données sont des slaves appelés DataNode.

6.1.1. Rôle des NameNode et des DataNode

Au sein du cluster, les données sont découpées et distribuées en blocs selon les deux paramètres suivants :

- Blocksize : Taille unitaire de stockage (généralement 64 Mo ou 128 Mo). C'est à dire qu'un fichier de 1 Go (et une taille de bloc de 128 Mo) sera divisé en 8 blocs.
- Replication factor : C'est le nombre de copies d'une donnée devant être réparties sur les différents noeuds du cluster (souvent 3, c'est à dire un primaire et deux secondaires).



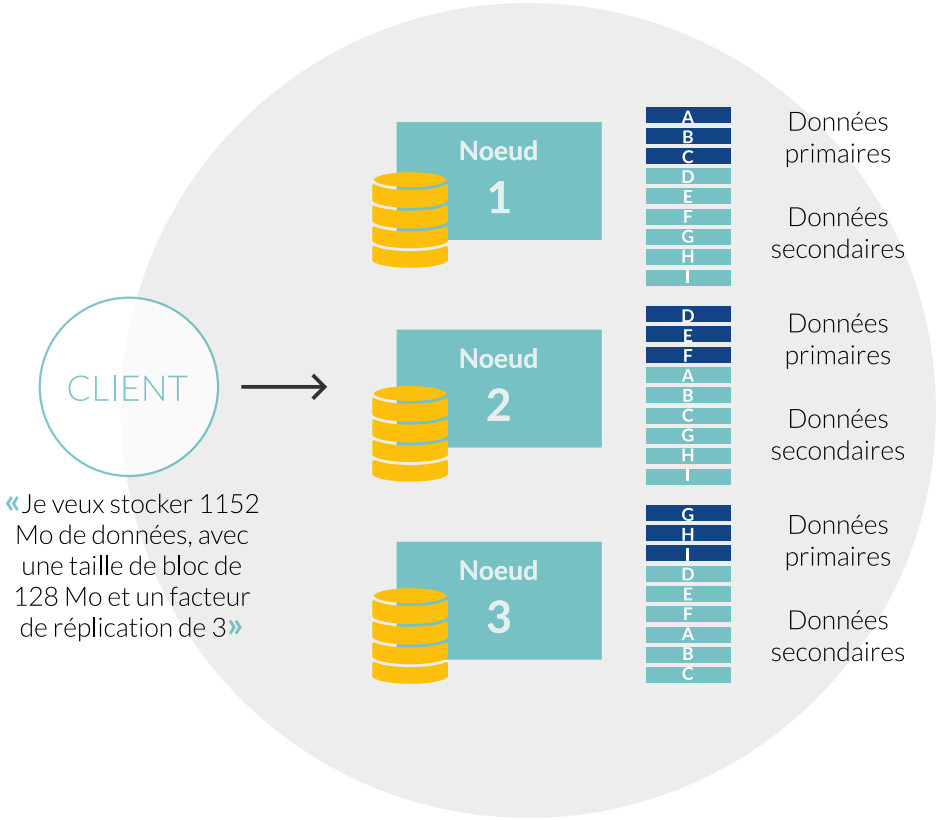
Répartition des données (DataNode)
Hadoop Distributed File System (HDFS)

Enfin, un principe important d'HDFS est que les fichiers sont de type «write-once» car dans des opérations analytiques, on lit la donnée beaucoup plus qu'on ne l'écrit. C'est donc sur la lecture que les efforts ont été portés.
Ce qui signifie que l'on ne modifie pas les données déjà présentes, on les remplace entièrement.

Un principe lié est qu'à partir du moment où un fichier HDFS est ouvert en écriture, il est verrouillé pendant toute la durée du traitement.

Il est donc impossible d'accéder à des données ou à un résultat tant que le job n'est pas terminé et n'a pas fermé le fichier (et un fichier peut être très volumineux avec Hadoop).

6.1.2. Système de répartition des données au sein des DataNode



6.1.3. Résumé et caractéristiques de HDFS

Résumé de Hadoop Distributed File System :

- Open Source.
- Développé en Java.
- Supporte de très gros volumes.
- Tolérant à la faute.
- Un ensemble de disques de taille modeste.
- Possibilité d'OS hétérogènes (c'est une surcouche).
- Blocs de 64 Mo à 1 Go (un fichier est donc découpé en n blocs, le prendre en compte dans les traitements).
- De type write-once (impossible de modifier a posteriori un enregistrement).
- Le verrou est posé sur tout le bloc pendant la phase d'écriture (impossible de lire).
- Plus lent que l'OS : ± 25 % en lecture, ± 50 % en écriture.
- Haut débit (volume).
- Scalable.
- Localisation des disques dans une machine/un rack.

Politique de réplication :

- 3 replicas dont 1 replica sur la machine locale et 2 replicas sur un rack distant.

HDFS permet donc de stocker de manière fiable des volumes de données très importants et sous leur forme brute.

6.1.4. Amélioration

Afin de minimiser ces dégradations, HDFS propose depuis Hadoop 2.3 un cache de disque afin d'accélérer les temps d'accès aux données.
Chaque DataNode peut utiliser une partie de la zone Off-heap pour y stocker une partie des fichiers.
Ce système n'est pas généralisé et doit se faire par déclaration : le contenu d'un répertoire ou d'un fichier doit être déclaré « cacheable ».

Quelles sont les améliorations possibles du système de fichiers HDFS ?

Différentes catégories :

1. Cache de disques,
2. Système de fichiers distribués,
3. Systèmes NoSQL.

6.1.4.1. Caches de disques

Les caches de disques vont permettre d'améliorer les performances de traitements sur le cluster tout en restant compatibles avec Hadoop.

Tachyon

Tachyon (produit AMPLab) est un cache de disque, dont le but est d'améliorer les temps d'accès aux données.
Si, de manière générale, un accès disque est peu performant comparé à un accès mémoire, le système de persistance Big Data (file system distribué) l'est encore davantage à cause de la redondance des informations pour gérer le failover.
Toutefois, Tachyon n'est pas un système autonome ou en tout cas si l'on veut de la résilience pour la donnée.
Il faut donc le coupler avec un autre système comme HDFS, S3, etc.
En conséquence, il est plus long en écriture (car en plus du cache il faut alimenter le système sous jacent).
Tachyon est donc parfait pour l'axiome Hadoop : « write once, read many ».

Ignite

La société GridGain a fait don de sa solution In Memory Data Grid en 2015 qui est devenue Apache Ignite.

En ce qui concerne Hadoop, deux composants permettent d'accélérer les traitements sur Hadoop tout en restant compatibles :

- IGFS - In-Memory FileSystem,
- Plugin Hadoop Accelerator.

IGFS (IGnite File System)

Implémentation de HDFS qui utilise Ignite et donc la mémoire pour le stockage des données.
Contrairement à Tachyon qui va accélérer les lectures sur disque, IGFS va aussi permettre d'accélérer les écritures en servant de tampon.
C'est IGFS qui va décider des données à placer en mémoire et si la synchronisation avec HDFS se fait de manière synchrone ou asynchrone.

Plugin Hadoop Accelerator

Ce composant va permettre de tirer profit de la mémoire à la fois pour les données mais aussi pour les traitements.
Notamment grâce à une implémentation maison du système de suivi des traitements (job tracker).

Distributions Hadoop supportées :

- Apache Hadoop,
- Hortonworks HDP,
- Cloudera CDH.

Il existe d'autres systèmes de fichiers en remplacement de HDFS.

Gluster FS

- Solution Open Source supportée par RedHat.
- Ecrit en C.
- Compatible Posix et NFS.
- Disponible sur les plateformes RedHat, Debian, MacOS, NetBSD.
- OpenSolaris.
- Compatible Swift (OpenStack).
- Très performant.

Ceph FS

- Existe depuis 2009.
- Solution Open Source supportée par Inktank.
- Ecrit majoritairement en C++ mais aussi en Perl, C et Python.
- Intégré au noyau Linux depuis la version 2.6.34 (mai 2010).
- Compatible S3 (Amazon) et Swift (OpenStack).
- Grande scalabilité et failover.

MapR-FS

En mai 2011, MapR a annoncé une alternative au système HDFS. Ce système permet d'éviter le SPOF qu'est le NameNode.
Ce système n'est pas inconnu car il s'agit de HBase, dont elle propose une version propriétaire.

Kudu (Cloudera)

Cloudera Kudu est un projet de Cloudera (actuellement en version bêta) développé en collaboration avec Intel.
C'est un système NoSQL orienté colonnes et natif Hadoop (tout comme HBase).
Contrairement à ce dernier Kudu supporte à la fois les accès aléatoires rapide (comme HDFS) et les écritures en temps réel (comme HBase).
C'est une solution unique, efficace pour ces deux types de besoin tout en restant compatible Hadoop.
Actuellement sous licence Apache il sera reversé à la fondation dans le futur.

HBase (Apache)

HBase est un sous-projet d'Hadoop, c'est un système de gestion de base de données non relationnelles distribué, écrit en Java, disposant d'un stockage structuré pour les grandes tables.
HBase est inspirée des publications de Google sur BigTable. Comme BigTable, c'est une base de données orientée colonnes.
Le mode standalone d'HBase utilisant le système de fichier natif à la place d'HDFS, il n'est pas distribué.

Mais HBase est souvent utilisé conjointement au système de fichiers HDFS, ce dernier facilitant la distribution des données de HBase sur plusieurs noeuds.
HBase est alors un moyen supplémentaire d'interroger les données construites avec Hadoop.
Contrairement à HDFS, HBase permet de gérer les accès aléatoires read/write pour des applications de type temps réel.

Cassandra (Apache)

Cassandra est une base de données orientée colonnes développée sous l'impulsion de Facebook (qui l'a abandonnée au profit de HBase en 2010).
Cassandra supporte l'exécution de jobs MapReduce qui peuvent y puiser les données en entrée et y stocker les résultats en retour (ou bien dans un système de fichiers).
Cette option n'est disponible qu'avec l'édition DataStax Enterprise Edition qui intègre un cluster Hadoop avec Cassandra.
Cassandra, comparativement à HBase est meilleure pour les écritures alors que ce dernier est plus performant pour les lectures.
Tout comme HBase, Cassandra est alors un moyen supplémentaire d'interroger les données construites avec Hadoop mais ne remplace pas complètement HDFS.

Accumulo (Apache)

Tout comme HBase et Cassandra, Accumulo est inspirée des publications de Google sur BigTable. Comme BigTable, c'est une base de données orientée colonnes.
A l'origine un projet de la NSA (2008), Accumulo est un projet Apache depuis 2011.
Tout comme HBase et contrairement à Cassandra, Accumulo est intimement lié à Hadoop et utilise Thrift et ZooKeeper.
Le contrôle d'accès est très fin puisqu'il est possible au niveau le plus élémentaire : la cellule.
Cette capacité est notamment très utile pour les architectures multi-tenant.

6.1.4.4. Accès au Cloud

Le cloud est un complément idéal au monde Hadoop, en offrant des possibilités de stockage et de traitement extensibles.
Il est donc possible d'utiliser un système de fichiers situé dans le cloud pour le stockage des données et l'exécution des traitements.

Solutions supportées :

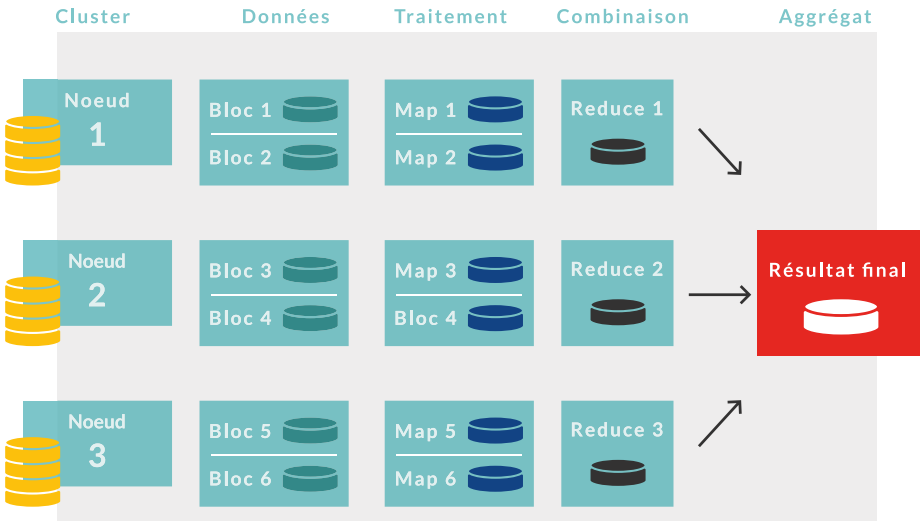
- **Amazon S3,**
- **Azure Blob Storage,**
- **OpenStack Swift.**

6.2. MapReduce

C'est un framework de traitements parallélisés, créé par Google pour son moteur de recherche web.
C'est un framework qui permet la décomposition d'une requête importante en un ensemble de requêtes plus petites qui vont produire chacune un sous ensemble du résultat final : c'est la fonction Map.

L'ensemble des résultats est traité (agrégation, filtre) : c'est la fonction Reduce.

Colocalisation des données et traitements



6.2.1. Les acteurs

Dans un traitement MapReduce, différents acteurs vont intervenir :

- Workers : Liste de nœuds Hadoop capables de traiter des tâches MapReduce.
- Master : Un worker dédié à la gestion des tâches.
- Client : Lance le traitement MapReduce (souvent nommé driver).

6.2.2. Les différentes phases

Initialisation : Le client/driver charge un/des fichiers dans HDFS et soumet un traitement MapReduce à la grille.

Split : Les données en entrée sont éventuellement divisées en blocs (16-64Mo).

Affectation : Le master affecte les tâches (Map et Reduce) aux workers. La configuration définit le nombre de tâches de type Map et Reduce supportées par chacun des noeuds.

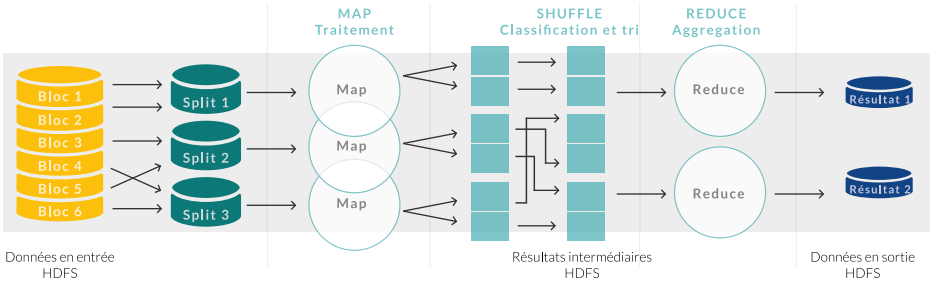
Map : Lecture des splits qui sont transmis à la fonction Map. Les ensembles clé/valeur produits par la fonction sont d'abord stockés en mémoire avant d'être périodiquement écrits localement (pas sur HDFS).

Shuffle : Les résultats des fonctions Map sont agrégés par la valeur de la clé pour produire une liste de valeurs traitées par le Reducer.

Reduce : Le master distribue au Reducer la liste des données à traiter. Les résultats sont envoyés au flux de sortie (HDFS, web services, ...).

Combiner : Optimisation, utilise les résultats intermédiaires du Map en entrée pour un traitement qui est généralement équivalent au Reducer (pas de garantie de passage).

Fin : Le master redonne la main au programme client.



6.2.3. Alternatives

Tout d'abord il faut bien comprendre le rôle de YARN dans la plateforme Hadoop.

YARN apporte une séparation claire entre les problématiques suivantes :

- Gestion de l'état du cluster et des ressources,
- Gestion de l'exécution des jobs.

Son adoption par Hadoop avec la version 2.0 a permis d'ouvrir la plate-forme à d'autres traitements que MapReduce.

Les alternatives sont très nombreuses et dépassent largement le cadre de ce livre blanc, elles ont pour point commun de favoriser l'utilisation de la mémoire plutôt que les accès disques.

Nous ne citerons que les plus connues :

- Apache Spark : Batch et micro batch.
- Apache Storm : Flux de données temps réel.
- Apache Impala : Streaming SQL.
- Apache Flink : Streaming itératif.

Offres plus spécialisées :

- Analyse interactive : Apache Drill,
- Traitement des graphes : Apache Giraph,
- ...

6.2.4. Bilan MapReduce

Selon Google voici la définition de MapReduce :
MapReduce est un modèle de programmation où l'utilisateur définit une fonction Map qui traite un ensemble de clés/valeurs afin de générer des clés/valeurs intermédiaires ainsi qu'une fonction Reduce qui agrège ces résultats selon la clé intermédiaire.

Il y a beaucoup de magie dans l'utilisation de MapReduce qui est en réalité gérée de manière interne :

- Distribution des traitements au plus près de la donnée Hadoop,
- Parallélisation,
- Découpage des données en splits,
- Reprise automatique sur erreurs,
- ...

MapReduce est idéal pour les traitements batchs et les gros fichiers, mais il n'est pas itératif ni adapté aux petites volumétries par défaut.

Données en entrée/sortie :

- HDFS (fichiers et répertoires),
- Disques locaux/réseaux,
- JDBC,
- Socket,
- HBase,
- ...

6.3. Tolérance à la panne

La tolérance à la panne dans Hadoop repose sur plusieurs principes :

Défaillance du système de gestion des tâches

Depuis Hadoop v2 il existe un système de failover du composant de planification des tâches (NameNode passif en attente).

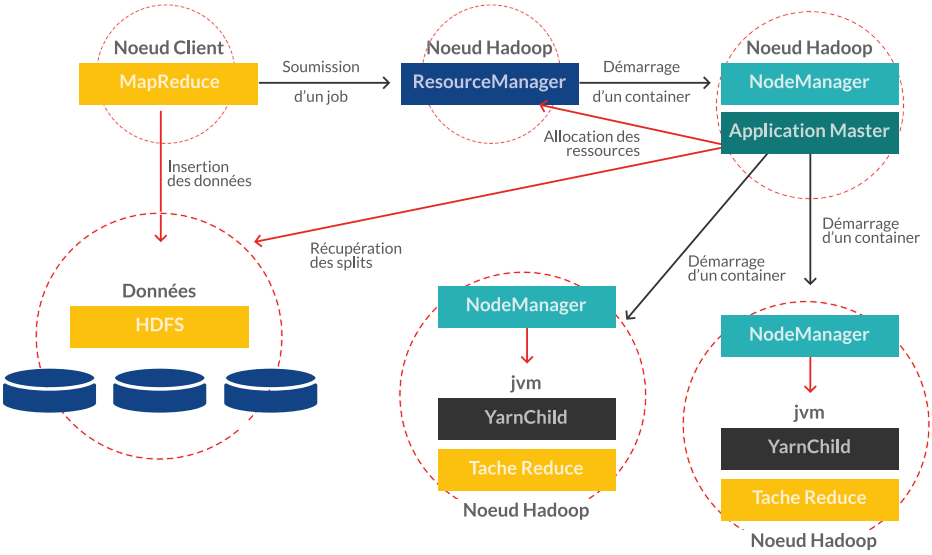
Suivi des workers par le “job tracker”

L’éventuel crash d’un noeud est détecté classiquement par un signal de vie (60 secondes par défaut). Le job tracker connaît les tâches affectées à un noeud et peut donc les relancer sur un autre.

« Speculative execution »

Lorsqu’un nœud est inactif, le job tracker lui affecte des tâches même si elles sont déjà affectées. Le premier qui termine la tâche “gagne”. Avec ce système on anticipe une éventuelle défaillance ou un engorgement.

YARN



6.4. Format de stockage

Le format de stockage est au cœur du système Hadoop car il s'agit du format de sérialisation des données.
Il est utilisé par HDFS pour le stockage des données mais aussi par MapReduce comme format d'échange entre les nœuds.
Les formats supportés sont :

- Format texte de type CSV,
- JSON,
- Binaire (Hadoop Serialization),
- Binaire (Avro, Thrift, Parquet, ...).

Les formats texte et JSON vont présenter l'avantage d'être compréhensibles par des humains mais l'inconvénient d'être peu optimisés pour le stockage et les performances.
Les formats binaires vont au contraire présenter l'avantage d'être plus performants.

6.4.1. Zoom sur Apache Avro

Apache Avro est un système de sérialisation de données qui a été conçu par Doug Cutting comme indépendant des langages utilisés.
Il existe une implémentation pour les langages suivants : C, C++, C#, Java, PHP, Python et Ruby.
Le fait de ne pas être lié à un langage (et à ses types) permet une évolution indépendante du type de données supporté et donc une évolutivité accrue.
Les schémas Avro sont généralement écrits au format JSON et stockés au format binaire.
De façon générale, c'est une alternative aux projets Apache Thrift et Google Protocol Buffer.

6.4.2. Synthèse

FORMAT	AVANTAGES	INCONVENIENTS
FORMAT TEXTE	Lisible par les humains	Simpliste (difficile de représenter les objets liés entre eux)
JSON	Flexible, Lisible par les humains	Performances
HADOOP SERIALIZATION	Performances	Pas de compatibilité ascendante prévue
BINAIRE (THRIFT, AVRO, ...(.	Performances et flexible.	Encore un nouveau format !

7 LES EXTENSIONS

Autour du cœur d'Hadoop, des solutions sont nées afin de couvrir les besoins non pris en compte ou bien pour faire le lien avec les systèmes existants. Aujourd'hui c'est un écosystème complet qui est proposé, capable de répondre à toutes les problématiques d'une entreprise.

7.1. Requêtage des données : Hive (Facebook)

Hive est à l'origine un projet Facebook qui permet de faire le lien entre le monde SQL et Hadoop.
Il permet l'exécution de requêtes de type SQL sur un cluster Hadoop en vue d'analyser et d'agréger les données.
Le langage SQL utilisé est nommé HiveQL. C'est un langage de visualisation uniquement, c'est pourquoi seules les instructions de type "Select" sont supportées pour la manipulation des données.

Dans certains cas, les développeurs doivent faire le mapping entre les structures de données et Hive.
Hive utilise un connecteur JDBC/ODBC.

Il existe une phase de transformation qui va transcrire les requêtes HiveQL en traitements MapReduce ou Tez (framework MapReduce de nouvelle génération par Hortonworks) dans les versions plus récentes.
Afin d'améliorer les performances, il existe des projets de migration de MapReduce vers Spark (présent en version bêta depuis Hive 1.1).

7.2. Scripting sur les données : Pig (Yahoo)

Pig est à l'origine un projet Yahoo qui permet le requêtage des données Hadoop à partir d'un langage de script.
Contrairement à Hive, Pig est basé sur un langage de haut niveau, PigLatin, qui permet de créer des programmes de type MapReduce ou Tez.

Contrairement à Hive, Pig ne dispose pas d'interface web.
Afin d'améliorer les performances, il existe des projets de migration afin d'utiliser Spark à la place de MapReduce/Tez.
Ces projets sont directement intégrés au projet Pig mais sont encore en cours de finalisation.

7.3. Intégration SGBD-R : Sqoop (Cloudera)

Sqoop permet le transfert des données entre un cluster Hadoop et des bases de données relationnelles.
C'est un produit développé par Cloudera.
Il permet d'importer/exporter des données depuis une base SQL vers Hadoop (et inversement).
Pour la manipulation des données Sqoop utilise MapReduce et des drivers JDBC.

La version Sqoop v2 (2013) apporte quelques améliorations :

- Meilleure utilisation du paradigme MapReduce (avec Sqoop v1 seuls les traitements Map étaient utilisés).
- Meilleure sécurité.
- Déploiement : Sqoop v1 nécessitait de déployer Sqoop ainsi que les drivers sur le client, avec Sqoop2 l'approche est différente puisque tout est installé sur le serveur Sqoop.

7.4. Ordonnanceur : Apache Oozie (Yahoo)

Oozie est une solution de workflow (au sens ordonnanceur d'exploitation) utilisée pour gérer et coordonner les tâches de traitement de données à destination de Hadoop.

Oozie s'intègre parfaitement avec l'écosystème Hadoop puisqu'il supporte les types de jobs suivants :

- MapReduce (Java et Streaming),
- Pig,
- Hive,
- Sqoop,
- Autres tels que programmes Java ou scripts de type Shell.

7.5. Gestion des clusters Hadoop

Oozie est une solution de workflow (au sens ordonnanceur d'exploitation) utilisée pour gérer et coordonner les tâches de traitement de données à destination de Hadoop.

Oozie s'intègre parfaitement avec l'écosystème Hadoop puisqu'il supporte les types de jobs suivants :

- MapReduce (Java et Streaming),
- Pig,
- Hive,
- Sqoop,
- Autres tels que programmes Java ou scripts de type Shell.

7.5.1. Clustering

7.5.1.1. Apache ZooKeeper

ZooKeeper est un service de coordination des services d'un cluster Hadoop. En particulier, le rôle de ZooKeeper est de fournir aux composants Hadoop les fonctionnalités de distribution.
Pour cela, il centralise les éléments de configuration du cluster Hadoop, propose des services de clusterisation et gère la synchronisation des différents éléments (événements).

ZooKeeper est un élément indispensable au bon fonctionnement de nombreuses solutions :

- HBase,
- Spark,
- YARN,

- ....

7.5.2. Supervision

7.5.2.1. Apache Ambari (Hortonworks)

Ambari est un projet Apache initié par Hortonworks et développé en partenariat avec Pivotal. Il est destiné à la supervision et à l'administration de clusters Hadoop.
C'est un outil web qui propose un tableau de bord. Cela permet de visualiser rapidement l'état d'un cluster.

Ambari dispose d'un tableau de bord dont le rôle est de fournir une représentation :

- **De l'état des services.**
- **De la configuration du cluster et des services.**
- **De l'exécution des jobs.**
- **Des métriques de chaque machine et du cluster.**

Ambari dispose aussi de fonctions d'administration telles que les sauvegardes et d'un système d'alertes en cas de dysfonctionnement.

De plus, Ambari inclut un système de gestion de configuration permettant de déployer des services d'Hadoop ou de son écosystème sur des clusters de machines.
Ambari se positionne en alternative à Chef ou Puppet pour les solutions génériques ou encore à Cloudera Manager pour le monde Hadoop.

Ambari ne se limite pas à Hadoop mais permet de gérer également tous les outils de l'écosystème.

Les solutions Hadoop annoncées sont :

- Hadoop,
- HDFS,
- MapReduce,
- Hive,
- Zookeeper,
- Oozie,
- HBase,
- Ganglia, Nagios.

De plus Ambari offre une API RESTful pour l'intégration avec les systèmes de pilotage du SI (monitoring).

7.6. Autres composants de l'écosystème Hadoop

7.6.1. Apache Flume (Cloudera)

Flume est une solution de collecte et d'agrégation de fichiers de logs, destinés à être stockés et traités par Hadoop.
Il a été conçu pour s'interfacer directement avec HDFS au travers d'une API native.
Flume est à l'origine un projet Cloudera, reversé depuis à la fondation Apache. Parmi les alternatives, on peut citer Apache Chukwa.

7.6.2. Apache Mahout

Apache Mahout est un projet de la fondation Apache depuis 2011 visant à créer des implémentations d'algorithmes d'apprentissage automatique (Machine Learning) et de Data Mining.
Même si les principaux algorithmes d'apprentissage se basent sur MapReduce, il n'y a pas d'obligation à utiliser Hadoop, Apache Mahout ayant été conçu pour pouvoir fonctionner sans cette dépendance.

Précurseur historique, Mahout fait face à de nouvelles librairies soit plus adaptées aux algorithmes itératifs, telles que MLlib de Spark (devenu Spark ML), soit provenant du monde des Data Scientists telles que Scikit-learn ou bien le langage R.

7.6.3. Apache Drill (MapR)

Initié par MapR, Drill est un système distribué permettant d'effectuer des requêtes sur de larges données. Il implémente les concepts exposés par le projet Google Dremel.
Drill permet d'adresser le besoin temps réel d'un projet Hadoop, MapReduce étant plutôt conçu pour traiter de larges volumes de données en batch sans objectif de rapidité et sans possibilité de redéfinir la requête à la volée.

Drill est donc un système distribué qui permet l'analyse interactive des données. Ce n'est pas un remplacement de MapReduce mais un complément qui est plus adapté pour certains besoins.

7.6.4. Apache Tez (Hortonworks)

Tez est un nouveau framework basé sur YARN et qui se veut le successeur de MapReduce.

Son but est d'améliorer les temps de traitement par une utilisation réduite du réseau et des accès disque.
Il fait partie de cette deuxième génération de framework comme Spark ou Flink, plus axés sur le "temps réel". La faible latence est en effet un pré requis à l'exploration interactive des données stockées sur un cluster Hadoop.

Des versions de Pig, Hive et Cascading utilisent déjà Tez à la place de MapReduce.

7.6.5. Apache Hue (Cloudera)

Hue est une application Web qui propose des interfaces utilisateurs nettement plus élaborées (et intuitives) que celles fournies de base par Apache Hadoop. **L'idée proposée par Hue est de simplifier l'utilisation d'Hadoop et de ses sous-projets afin d'en favoriser l'adoption.**

Il est donc possible depuis Hue de :

- Naviguer dans le système de fichiers HDFS.
- Visualiser et d'exécuter un job MapReduce.
- Visualiser les données dans Hbase.
- Saisir des requêtes pour Apache Hive.
- Saisir des requêtes pour Apache Impala.
- Saisir des requêtes pour Apache Pig.
- Saisir des requêtes pour Apache Sqoop2.
- Bénéficier d'un tableau de bord et éditeur Apache Oozie.
- Construire de manière interactive des requêtes Apache Solr.
- Naviguer dans le système Apache ZooKeeper.
- Editer des traitements Spark.
- Editer des traitements SQL.

7.6.6. Apache Kafka (LinkedIn)

Apache Kafka est un Message Broker qui s'appuie sur le système de fichiers pour gérer les événements reçus.
La file de messages est donc persistée (et les données distribuées) et reconsultable à tout moment.
Contrairement à d'autres systèmes similaires, Kafka ne maintient pas l'état des files coté serveur (quel est le message courant pour ce consommateur ?). Cette problématique est traitée coté client.
Couplé à une approche sans maintien d'état, l'arrêt d'un nœud de Kafka n'induit pas de perte de messages.

En raison de sa scalabilité et de ses performances, Kafka est le système privilégié par de nombreux systèmes (Apache Spark, Flink, …) pour assurer la reprise sur erreurs.

Dans Hadoop, Kafka va permettre de persister le stockage des résultats intermédiaires (stockés sur le disque local et donc inaccessibles aux autres nœuds avec MapReduce) afin de faciliter la reprise sur erreur.
Il peut aussi servir comme source ou stockage des traitements sur le cluster Hadoop.

7.6.7. Apache Solr

Apache Solr est une des solutions d'indexation de contenus hétérogènes de la fondation Apache.

- Création en 2004 par CNET Networks.
- Début 2006, projet rendu open source et confié à la fondation Apache.
- En mars 2011, les projets Lucene et Solr sont fusionnés. Les versions sont harmonisées autour d'un tronc commun mais la distribution des deux solutions reste séparée.

Apache Solr est une évolution du projet Apache Lucene auquel il ajoute certaines fonctionnalités :

- Faceting,
- Interface web d'administration.

Contrairement à Lucene qui est une pure API Java qu'il faut intégrer à son projet, Solr est un moteur de recherche de type serveur qui expose une API Web pour les accès à ses services.
L'intégration avec Hadoop se fait au travers d'un connecteur qui permet le stockage de l'index sur HDFS.
L'alimentation de ce dernier s'effectue par des traitements MapReduce et donc en mode batch.

7.6.8. DataFu (LinkedIn)

DataFu est à l'origine un projet LinkedIn et fournit une bibliothèque de traitements MapReduce basée sur leur plateforme d'analyse de données.

Il est actuellement en incubation au sein de la fondation Apache.

Liste des fonctions disponibles :

- Statistiques (median, variance, ...),
- Quantities,
- Sampling,
- PageRank,
- Sessionization,
- ...

DataFu permet donc l'analyse et la valorisation des données d'un cluster Hadoop.
Concrètement, il s'agit de fonctions Pig personnalisées.

Un projet lié nommé Hourglass permet d'étendre MapReduce afin de proposer des traitements itératifs.

Le traitement itératif est un manque important de MapReduce et est particulièrement adapté pour les traitements d'analyse de données afin d'éviter un recalcul complet des informations.

7.7. Sécurité

7.7.1. Un démarrage raté

La plateforme Hadoop a été conçue pour traiter de très larges volumes de données.
La sécurité avait été volontairement délaissée et il a fallu attendre 2009 pour que les choses bougent.
En effet, c'est l'année où Kerberos a été retenu comme mécanisme d'authentification pour la plateforme.

La sécurité dans Hadoop est rendue difficile car :

1. Hadoop utilise un nouveau système de fichiers,
2. De multiples solutions composent Hadoop,
3. Le principe même d'Hadoop est un système réparti sur plusieurs nœuds.

C'est pourquoi ce sont d'abord les éditeurs qui ont proposé des solutions propriétaires au détriment des normes.

7.7.2. Les enjeux de la sécurité dans Hadoop

Qu'est ce que la sécurité dans Hadoop :

- Contrôle de l'accès aux données (HDFS),
- Authentification d'accès aux clients HTTP (supervision, monitoring),
- Authentification de lancement des tâches,
- Chiffrement des échanges réseau (données, logs, ...),
- Audit,
- ...

Et ce pour tous les éléments composants la plateforme Hadoop.

7.7.3. Solutions de sécurité natives à la plateforme

Apache Hadoop est par défaut dans un mode non sécurisé. L'activation du mode sécurisé permet de bénéficier de :

1. l'authentification.
2. l'autorisation au niveau des services.
3. l'authentification au niveau des consoles web.
4. la confidentialité des données.

7.7.3.1. Authentification basique

Tous les clients de la plateforme Hadoop doivent être authentifiés par Kerberos.
Un mapping entre le système d'exploitation et Hadoop est nécessaire pour l'authentification.

7.7.3.2. Autorisation au niveau des services et des consoles web

Une fois les clients authentifiés, il faut leur associer un rôle au sein du cluster.

Trois groupes sont proposés :

- HDFS : Accès aux données
- YARN : Accès à la planification des ressources et traitements
- MapReduce : Lancement de traitements

Le mapping est réalisé soit au niveau du système d'exploitation, soit par un annuaire LDAP.
Il faut définir les permissions entre chaque groupe et les différents points d'accès au cluster.

7.7.3.3. Confidentialité des données

L'activation du chiffrement des données entre les services Hadoop et les clients (protocole RPC), se fait au moyen d'un simple paramètre.

Chiffrement des données entre les services Hadoop : Les DataNodes n'utilisent pas le protocole RPC d'Hadoop, les données ne sont donc pas chiffrées malgré l'activation du chiffrement cité ci-dessus.
Il est toutefois possible de chiffrer spécifiquement ces échanges (3DES ou RC4).

Enfin, il est possible de chiffrer les données entre les consoles web et les clients.
Le protocole HTTPS (SSL) permet de sécuriser ces échanges.

7.7.3.4. Chiffrement HDFS

Depuis la version 2.6, HDFS offre la possibilité de chiffrer les données de manière transparente.
Il est possible de configurer certains répertoires HDFS pour que les lectures retournent des données chiffrées et que les écritures n'acceptent que des données chiffrées.
Le mécanisme est entièrement géré par le client, c'est à dire que le cluster Hadoop n'a aucune connaissance des clés utilisées et n'est pas en mesure de déchiffrer les données.

Possibilités offertes :

- Chiffrement au niveau de la base de données (à l'exception des index qui ne peuvent être chiffrés).
- Chiffrement au niveau du système de fichiers HDFS.

Il est à noter que Cloudera et Hortonworks travaillent sur un chiffrement de plus bas niveau (disque) qui, outre des performances accrues, offre une meilleure protection.

Globalement, la sécurisation d'une plateforme Hadoop nécessite de nombreux changements dans la configuration de base.
Elle doit être mise en place à différents niveaux et il manque une solution globale qui permettrait de sécuriser un cluster Hadoop.

7.7.4. Solutions de sécurité globales

On va trouver différents types de solutions :

- Les solutions centralisées telles qu'Apache Knox,
- Les solutions spécifiques à un élément de la plateforme (stockage, authentification, chiffrement, ...) comme ACL (Access Control List) pour HDFS,
- Les frameworks qui doivent être intégrés aux solutions devant être sécurisées comme Apache Sentry ou Apache Ranger.

7.7.4.1. Apache Knox

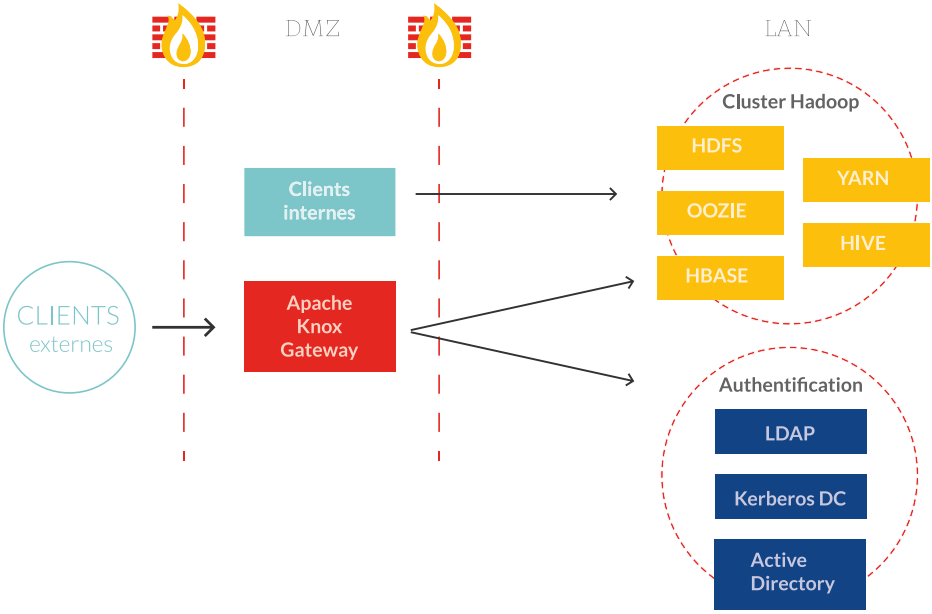
Apache Knox agit comme un proxy qui est un passage obligé afin d'accéder à un cluster Hadoop.
On va distinguer les accès en interne (Hadoop) des accès externes (clients).
Apache Knox est lui même un serveur HTTP qui peut être clusterisé.
Apache Knox est stateless et les clients accèdent au cluster Hadoop au travers d'API Rest.

Fonctionnalités offertes :

- Authentification : Gestion des utilisateurs et des groupes (LDAP, Active Directory, KDC (Kerberos Key Distribution Center).
- Fédération/SSO : Possibilité d'intégrer un header HTTP contenant les informations de fédération.
- Autorisation : Au travers des Access Control Lists (ACL).
- Audit : Tous les accès à Knox sont tracés.

Compatibilité :

- HDFS (WebHDFS),
- Oozie,
- HBase,
- Hive,
- YARN.



7.7.4.2. Apache Ranger (Hortonworks)

Apache Ranger (anciennement Apache Argus) n'est pas une solution clé en main mais un framework de gestion et de surveillance de la sécurité. Il doit être intégré à la solution à sécuriser.
Il permet de gérer à la fois les autorisations et l'audit du cluster.
Il est à noter qu'il permet de gérer très finement ces autorisations (rôle 1 sur le produit A et rôle 2 sur le produit B, alors que d'autres ne permettent de définir qu'un seul rôle pour l'ensemble des solutions du cluster).

Actuellement, il est intégré avec les solutions suivantes :

- Apache Hadoop,
- Apache Hive,
- Apache HBase,
- Apache Storm,
- Apache Knox.

7.7.4.3. Apache Sentry (Cloudera)

Apache Sentry est à l'origine un projet Cloudera et est en Incubation au sein de la fondation Apache.
Tout comme Apache Ranger, c'est un framework qui va permettre de gérer les autorisations au sein du cluster Hadoop.
Son approche est résolument orientée SQL puisqu'il va permettre d'attribuer des droits pour des ordres de type « select », « insert » et « transform » sur les données au sein du cluster.

Actuellement, il est intégré avec les solutions suivantes :

- Apache Solr,
- Apache Hive,
- Apache Impala,
- Apache Sqoop2

7.7.4.4. Cloudera Navigator

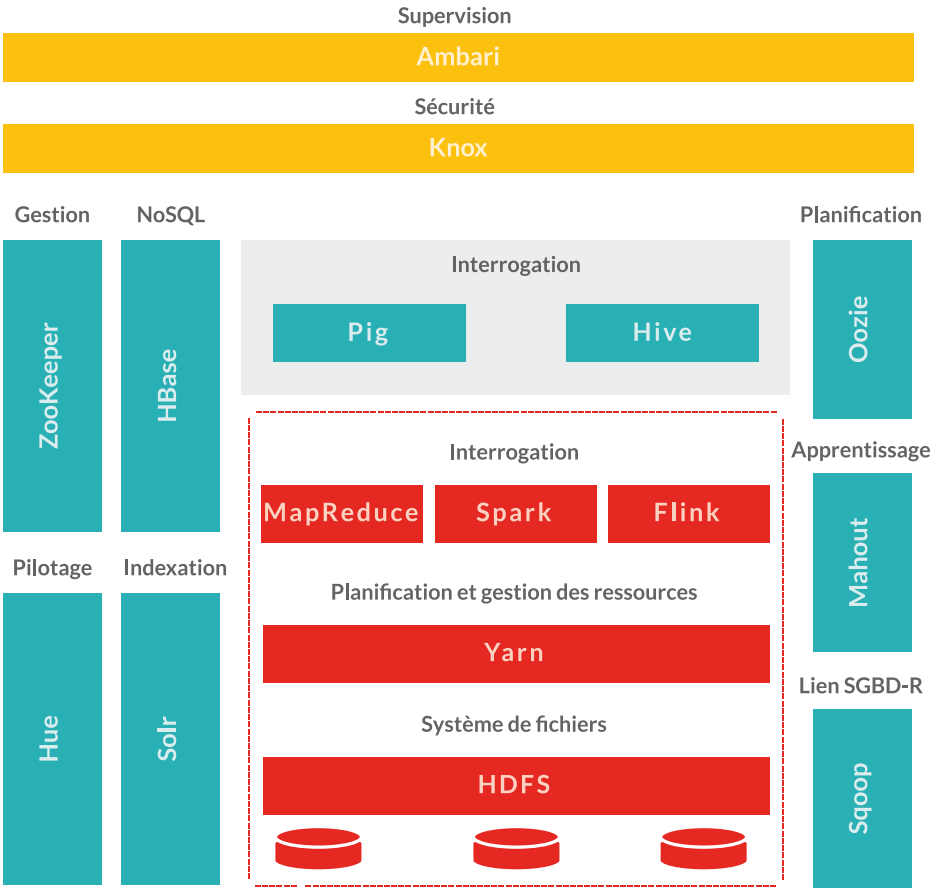
Cloudera Navigator (uniquement avec une licence) permet d'auditer un cluster Hadoop et donc d'enregistrer tous les accès et modifications (ce qu'il nomme activité).

Ce n'est donc pas une solution complète de sécurisation d'un cluster Hadoop.

Une activité se présente sous la forme suivante :

- Un timestamp,
- L'objet accédé,
- Opération sur l'objet,
- L'utilisateur ayant réalisé l'opération (ainsi que son adresse IP),
- Le service ayant permis l'accès.

8 VUE D'ENSEMBLE DE LA PLATE-FORME HADOOP



COMPOSANT	DÉTAILS	ORIGINE
HDFS	File System distribué	Yahoo, 2005 Projet Apache depuis 2007
MAPREDUCE	Framework de traitement parallélisé	Yahoo, 2005 Projet Apache depuis 2007
YARN	MapReduce 2.0	Yahoo, 2007 Projet Apache depuis 2008
SQOOP	Permet le transfert des données entre Hadoop et des bases de données relationnelles	Cloudera Projet Top-Level Apache depuis mars 2012
ZOOKEEPER	Service de coordination pour les applications distribuées	Yahoo, 2005 Projet Apache depuis 2007
HBASE	Base de données NoSQL (accès read/write aléatoires)	Powerset (Microsoft), 2007 Projet Apache depuis 2008
PIG	Scripting et requêtage Hadoop	Yahoo, 2006 Projet Apache depuis 2007
HIVE	Requêtage de type SQL	Hive a été initialement développé par Facebook. Projet Apache depuis 2008
AMBARI	Supervision	Initialement développé Hortonworks. Projet Apache depuis 2012
OOZIE	Workflow et planification de jobs Hadoop	Projet Apache depuis 2011
TEZ	Framework de traitement utilisant YARN, il remplace MapReduce afin de fournir des requêtes dites “temps réel”	Initialement développé par Hortonworks. Projet Apache depuis 2015
MAHOUT	Mahout est une librairie Java qui permet d’implémenter différents algorithmes de Machine Learning sur un cluster Hadoop	Projet Top Level Project Apache depuis 2010
HUE	Interface web d’accès au cluster Hadoop	Projet Cloudera (Cloudera Desktop) depuis 2009 Projet Apache depuis 2010
SOLR	Moteur d’indexation Apache	Initialement développé par CNET Networks Projet Apache depuis 2006

9 OPEN DATA PLATFORM

9.1. Présentation

Conscients que de la diversification des solutions et des distributions Hadoop étaient un frein à l'adoption de ces technologies, certains acteurs ont décidé de travailler ensemble à sa standardisation.

L'Open Data Platform lancée en février 2015 vise donc à proposer une normalisation afin de disposer d'un socle Hadoop standard dans le but de faciliter son adoption.

Ce n'est pas pour autant la fin des distributeurs de solutions Hadoop ainsi que des offres Cloud puisque la plateforme ne vise que les composants de base.Actuellement une application développée pour une distribution

Ce que propose l'Open Data Platform c'est l'assurance de pouvoir changer de distribution plus facilement qu'aujourd'hui.

Hadoop particulière n'est pas assurée de pouvoir s'exécuter sur une autre distribution.

De plus, les composants d'Hadoop évoluent très rapidement et il très important d'avoir une plateforme évolutive avec des composants interchangeables.

9.2. Membres de l'alliance

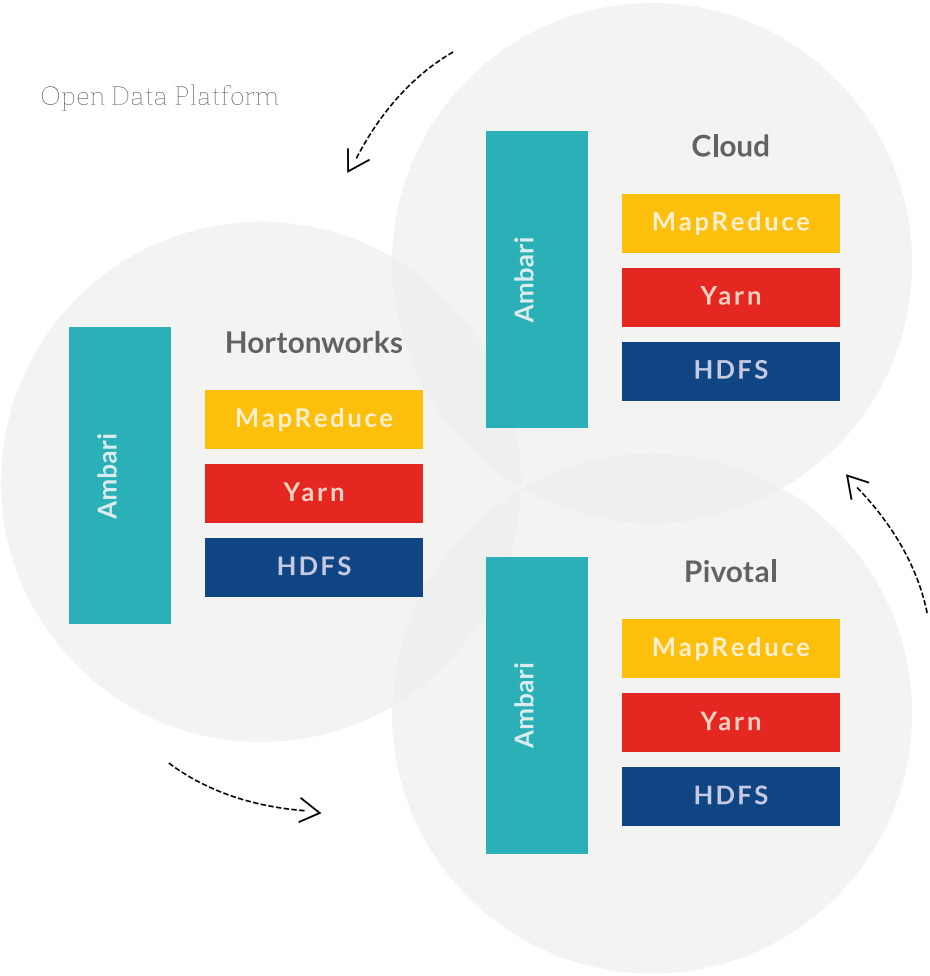
Une quinzaine d'acteurs compose actuellement cette alliance. Parmi eux, on retrouve des grands noms tels que :

Pivotal, Hortonworks, IBM, SAS, VMware, EMC, InfoSys, Capgemini et Teradata.

Certains distributeurs comme Cloudera ou MapR ont refusé de participer de peur de perdre l'avantage concurrentiel offert par leurs solutions maisons.

9.3. Contenu de la plateforme

1. Stockage HDFS,
2. Gestion des ressources YARN,
3. Traitement MapReduce 2.0,
4. Outil d'administration Ambari.


Open Data Platform
Ambari
Hortonworks
MapReduce
Yarn
HDFS
Ambari
Cloud
MapReduce
Yarn
HDFS
Ambari
Pivotal
MapReduce
Yarn
HDFS

10 LES DISTRIBUTIONS

10.1. Hortonworks

10.1.1. Présentation

Hortonworks a été formé en juin 2011 par des membres de l'équipe Yahoo en charge du projet Hadoop.
Leur but est de faciliter l'adoption de la plate-forme Hadoop d'Apache, c'est pourquoi tous les composants sont open source et sous licence Apache.
Le modèle économique d'Hortonworks n'est pas de vendre des licences mais essentiellement du support et des formations (même si on voit apparaître quelques produits avec licence comme SmartSense).
Cette distribution est la plus conforme à la plate-forme Hadoop d'Apache et Hortonworks est le gros contributeur Apache Hadoop.

Parmi les projets reversés, il y a :

- YARN,
- Tez,
- Ambari,
- Falcon,
- Knox,
- ...

10.1.2. Composants de la distribution

Les éléments suivants composent la plate-forme Hortonworks Data Platform (HDP) et sont des composants Apache Hadoop :

- HDFS : stockage distribué.
- MapReduce : Traitements parallélisés.
- HBase : Base NoSQL orientée colonnes sur HDFS.
- Pig : plate-forme de scripts d'interrogation HDFS.
- Hive : Requêtage et Méta-données HDFS.
- Oozie : Planification de traitements.
- ZooKeeper : Coordination du cluster.
- Ambari : Gestion et supervision.
- WebHDFS : Accès web aux données.
- Ingestion de données : Talend Open Studio for Big Data.
- Sqoop : Interactions avec les SGBD.
- Flume : Gestion distribuée des logs.
- Mahout : Apprentissage.

Composants non Apache Hadoop :

- Hortonworks Cloudbreak (licence Apache) : Solution agnostique de dimensionnement, de gestion et de monitoring d'un cluster HDP, compatible avec Microsoft Azure, Amazon AWS, Google Cloud Platform, OpenStack pour les offres cloud ainsi qu'Apache Ambari, Docker, Swarm et Consul.
- Hortonworks SmartSense (nécessite une licence) : Maintenance proactive d'un cluster HDP, recommandations, optimisation de l'utilisation des ressources.
- Solr on YARN (licence Apache) : C'est une version entièrement compatible avec Hadoop, ce qui permet de déployer Solr au sein du cluster et évite de déployer des machines dédiées à Solr.

10.1.3. Vision d'ensemble de la distribution

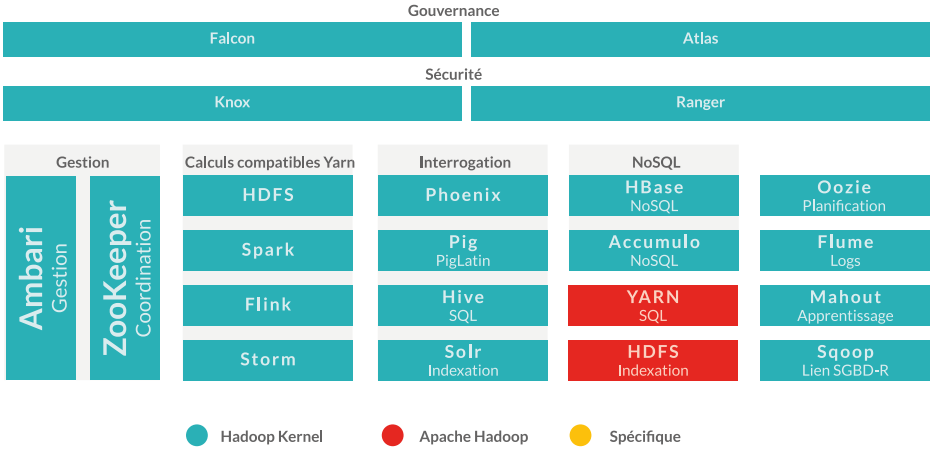
Les éléments suivants composent la plate-forme Hortonworks Data Platform (HDP) et sont des composants Apache Hadoop :

- HDFS : stockage distribué.
- MapReduce : Traitements parallélisés.
- HBase : Base NoSQL orientée colonnes sur HDFS.
- Pig : plate-forme de scripts d'interrogation HDFS.
- Hive : Requêtage et Méta-données HDFS.
- Oozie : Planification de traitements.
- ZooKeeper : Coordination du cluster.
- Ambari : Gestion et supervision.
- WebHDFS : Accès web aux données.
- Ingestion de données : Talend Open Studio for Big Data.
- Sqoop : Interactions avec les SGBD.
- Flume : Gestion distribuée des logs..
- Mahout : Apprentissage.

Composants non Apache Hadoop :

- Hortonworks Cloudbreak (licence Apache) : Solution agnostique de dimensionnement, de gestion et de monitoring d'un cluster HDP, compatible avec Microsoft Azure, Amazon AWS, Google Cloud Platform, OpenStack pour les offres cloud ainsi qu'Apache Ambari, Docker, Swarm et Consul.
- Hortonworks SmartSense (nécessite une licence) : Maintenance proactive d'un cluster HDP, recommandations, optimisation de l'utilisation des ressources.
- Solr on YARN (licence Apache) : C'est une version entièrement compatible avec Hadoop, ce qui permet de déployer Solr au sein du cluster et évite de déployer des machines dédiées à Solr.

HortonWorks Hadoop platform (HDP)



10.2. Cloudera

10.2.1. Présentation

Cloudera se veut comme la compagnie commerciale Hadoop.
Fondée par des experts Hadoop en provenance de Facebook, Google, Oracle et Yahoo.
Si leur plate-forme est en grande partie basée sur Hadoop d'Apache, elle est complétée avec des composants maison pour la gestion du cluster.
A noter aussi que la version d'Apache Hadoop distribuée est la dernière version stable complétée de patchs critiques ainsi que de quelques fonctionnalités de la version de développement.
Le modèle économique de Cloudera est la vente de licences mais aussi du support et des formations. Cloudera propose une version entièrement open source de leur plate-forme (Licence Apache 2.0).

De grands investisseurs ont misé récemment sur Cloudera :
1. Intel
2. EMC

10.2.2. Composants de la distribution Cloudera Express

Composants de la distribution Cloudera Express :
Composants Apache Hadoop :

- HDFS : File System distribué.
- MapReduce : Framework de traitement parallélisé.
- HBase : Base de données NoSQL (accès read/write aléatoires).
- Hive : Requêtage de type SQL.
- Pig : Scripting et requêtage Hadoop.
- Oozie : Workflow et planification de jobs Hadoop.
- Sqoop : Intégration de bases SQL.
- Flume : Exploitation de fichiers (logs) dans Hadoop.
- ZooKeeper : Service de coordination pour les applications distribuées.
- Mahout : Framework d'apprentissage et de Data Mining pour Hadoop.

Composants d'origine Cloudera :

- Hadoop Common : Un ensemble d'utilitaires.
- Hue : Interfaces utilisateur (web) pour les applications Hadoop.
- Whirr : Librairies et scripts pour l'exécution d'Hadoop et de services liés dans le cloud (abandonné depuis par Apache).
- Oryx 2 : Apprentissage (Machine Learning) de nouvelle génération, s'appuyant sur l'architecture Lambda et spécialisé dans les moteurs de recommandations.

Composants non Apache Hadoop :

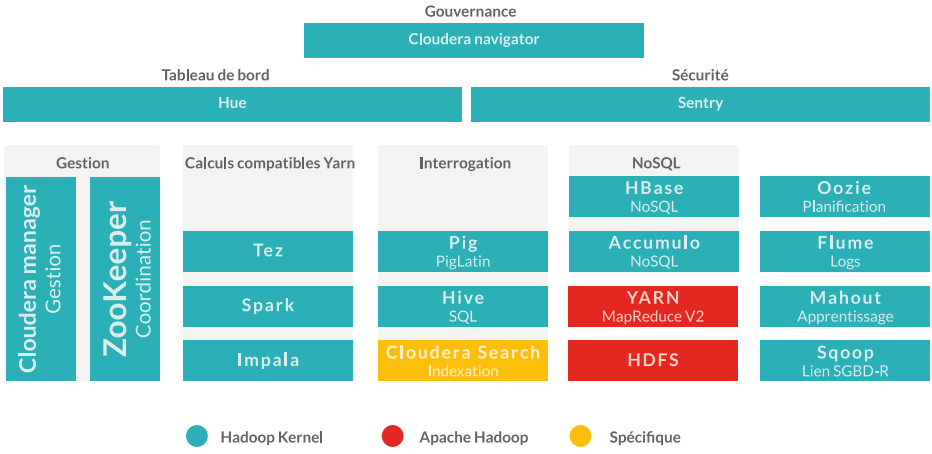
- Cloudera Impala : Moteur temps réel de requêtage SQL parallélisé de données stockées dans HDFS ou HBase. Contrairement à Hive de Hadoop, Impala n'utilise pas le framework MapReduce qui exige que les résultats de recherche soient écrits sur le disque, ce qui lui permet d'exécuter les requêtes plus rapidement. La consultation des données peut être interactive. Licence : ASLv2.
- Cloudera Manager, Director, Navigator : Déploiement et gestion des composants Hadoop.

A noter que Cloudera Manager n'est pas entièrement Open Source mais dispose d'une version gratuite avec quelques restrictions :

- La version gratuite est limitée à 50 noeuds.
- Certaines fonctionnalités sont uniquement disponibles sur la version commerciale (comme le monitoring, les sauvegardes et les mises à jour automatiques).
- Support uniquement pour la version payante.

10.2.3. Vision d'ensemble de la distribution

Cloudera Distribution for Hadoop Platform (CDH)



10.3. MapR

10.3.1. Présentation

MapR a été fondée en 2009 par d'anciens membres de Google.
Bien que son approche soit commerciale, MapR contribue à des projets Apache Hadoop comme HBase, Pig, Hive, ZooKeeper et surtout Drill.
MapR se distingue surtout de la version d'Apache Hadoop par sa prise de distance avec le cœur de la plate-forme. MapR propose ainsi son propre système de fichiers distribué ainsi que sa propre version de MapReduce : MapR FS et MapR MR.

Trois versions de leur solution sont disponibles :

- M3 : version open source.
- M5 : ajoute des fonctions de haute disponibilité et du support.
- M7 : environnement HBase optimisé.

MapR a remporté de beaux succès commerciaux depuis sa création.

- Un partenariat avec EMC pour la création et le support d'une version spécifique à la plate-forme Hadoop d'EMC.
- MapR est à l'origine de la version cloud de MapReduce d'Amazon : Elastic Map Reduce (EMR).
- Enfin ils ont été retenus par Google pour l'offre Big Data de Google Compute Engine (GCE).

Le fond d'investissement de Google a massivement investi dans MapR lors d'une levée de fonds en juin 2014 (80 M$).

10.3.2. Contenu de la distribution MapR M3

Composants Apache Hadoop :

- HBase,
- Pig,
- Hive,
- Mahout,
- Cascading,
- Sqoop,
- Flume.

MapR propose son propre système en remplacement de HDFS qui est une version maison de HBase (performance et fiabilité améliorées). Il est composé de MapR FS et de MapR Control System.

Avantages :

- Système plus adapté au mode read/write que HDFS.
- MapR intègre un serveur NFS (Network File System) pour l'intégration au SI de l'entreprise.
- Simplification de mise en oeuvre (surcouche du File System de l'OS et non remplacement comme HDFS).
- Sécurité.

MapR FS reste compatible avec les API MapReduce/HDFS et Hbase.
MapR propose son propre système en remplacement de MapReduce d'Apache.

Avantages :

- MapR annonce de meilleures performances.
- Entièrement optimisé pour HBase.

MapR Control System (MCS) permet la gestion et la supervision du cluster Hadoop. C'est un outil web permettant de gérer à la fois les ressources du cluster (CPU, RAM, disque) que les services et les jobs.
MCS permet de définir des alarmes sur des seuils ou des quotas...
La visualisation des informations est assurée par le composant HeatMap.

Autres spécificités de la distribution MapR :

Apache Cascading

Cascading est un framework Java dédié à Hadoop. Il permet à un développeur Java de retrouver ses marques (JUnit, Spring, etc...) et de manipuler les concepts d'Hadoop avec un langage de haut niveau sans en connaître les API.

Apache Drill

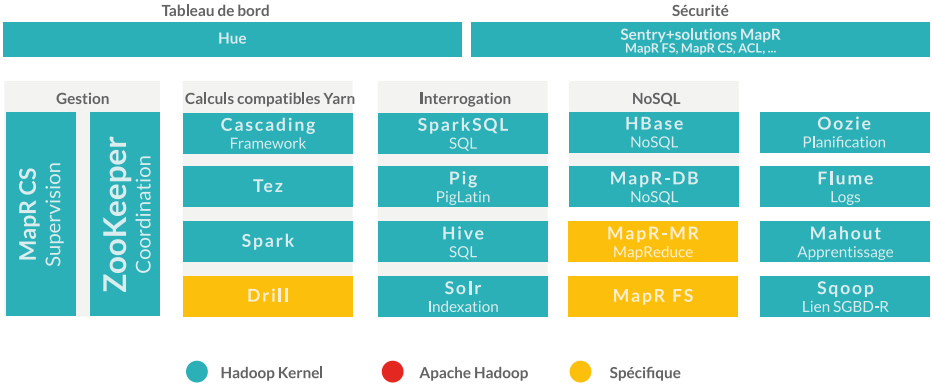
MapReduce a la réputation d'être puissant mais complexe à manipuler (il faut en maîtriser l'API).
De plus, il est impossible de redéfinir les requêtes à la volée.
Drill vient compléter MapReduce et se présente sous la forme d'une API permettant de créer plus rapidement des requêtes en se basant sur le modèle SQL.
Utiliser la norme SQL plutôt qu'une nouvelle API, c'est donc le choix de la capitalisation fait par Drill.

Sécurité

La Sécurité dans Hadoop MapR s'appuie sur Apache Sentry mais aussi sur un ensemble de composants maison.
Sentry s'intègre avec Hive et Impala.
La sécurité de bas niveau est intégrée au File System maison (chiffrement, ...) alors que la gestion des rôles est intégrée au système de gestion MapR Control System.

10.3.3. Vision d'ensemble de la distribution

MapR (M3)



10.4. Apache BigTop

10.4.1. Présentation

Initiative très récente, le projet BigTop vise à apporter une distribution complète et homogène des composants Apache Hadoop.

Il n'est plus nécessaire de télécharger les composants unitairement.

Actuellement en version 1.0 depuis septembre 2015.

10.4.2. Contenu de la distribution

1. BigTop Groovy
2. BigTop jsvc (Apache common)
3. BigTop Tomcat
4. BigTop utils
5. Apache Crunch
6. Apache DataFu
7. Apache Flume
8. Apache Giraph
9. Apache Hadoop
10. Apache Hbase
11. Apache Hive
12. Apache Hue
13. Apache Ignite
14. Apache Kafka
15. Apache Mahout
16. Apache Oozie
17. Apache Phoenix
18. Apache Pig
19. Apache Solr
20. Apache Spark
21. Apache Sqoop 1 & 2
22. Apache Tachyon
23. Apache Tez
24. Apache ZooKeeper

10.4.3. Ecosystème

Puppet : Une grande partie des outils de packaging et de déploiement du projet Apache BigTop utilise Puppet pour le lancement et la configuration du cluster.

Groovy : Groovy est le langage de prédilection pour l'écriture de tests au sein de BigTop.

Maven : Utilisé pour la construction du projet BigTop.

Jenkins : Utilisé pour piloter la chaîne de build et garantir l'exécution des tests unitaires au sein du projet.

9.4.4 Plateformes disponibles

Apache BigTop est disponible sous la forme de package deb et rpm. Seules les distributions Linux sont donc supportées : Debian, Ubuntu, CentOS, Fedora, openSUSE...

10.5. Offres cloud

Hadoop est disponible sur de nombreuses offres cloud.
Si le cloud est utilisé à des fins de POC ou de développement, il est impératif de pouvoir capitaliser sur ces phases et donc de bénéficier d'offres compatibles avec la distribution retenue en interne.

	AMAZON WEB SERVICES	MICROSOFT AZURE
NOM	Amazon Elastic MapReduce (Amazon EMR)	HDInsight
SOLUTIONS HADOOP	Hive, Pig, Hbase, Impala, R, Mahout, Ganglia, Spark et Shark, Accumulo, Sqoop, HCatalog	Tez, Pig, Hive, HCatalog, Hbase, Sqoop, Oozie, Templeton, Ambari, Zookeeper, Mahout
DISTRIBUTION	Apache Hadoop, Hortonworks, MapR, Cloudera	Apache Hadoop, Hortonworks, MapR, Cloudera

	GOOGLE CLOUD	RACKSPACE
NOM	BigQuery et Hadoop pour Google Cloud Platform	Cloud Big Data Platform
SOLUTIONS HADOOP	Hive, Pig, Spark et Shark	Hive, Pig, MapReduce, Tez, YARN, Flume, Sqoop, Oozie, Storm, Kafka, Falcon, Spark
DISTRIBUTION	Apache Hadoop, MapR, Hortonworks	Hortonworks

11 LES CAS D'UTILISATIONS D'HADOOP

Les cas d'utilisation d'Hadoop dans une entreprise vont correspondre aux cas actuels d'utilisation de la technologie SQL pour le stockage et l'interrogation des données et les ETL pour la partie traitements, le tout avec :

- Une volumétrie accrue.
- Données hétérogènes.
- Volonté d'exploitation de ces données (Data Mining, BI).

Outre la volumétrie des données, une solution de type Hadoop va permettre d'améliorer la scalabilité de la solution et donc de pouvoir faire face à des augmentations de volumétrie sans dégradation importante des performances.

11.1. Audit/Qualité des données

Historiquement, le Système d'Information d'une entreprise va cumuler des données sous des formats variables (car provenants de sources différentes) et pas toujours correctes (évolutions des règles de validation, système de saisie défaillant, ...).

Elles ont donc besoin d'être consolidées avant d'être exploitées :

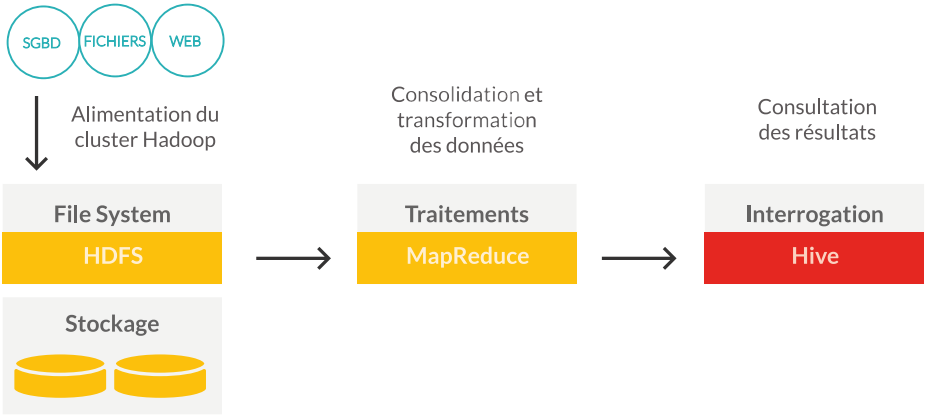
- Uniformisation du format des données.
- Correction ou purges des données incorrectes.

Voici les différences phases du processus :

1. Insertion des données dans HDFS.
2. Définition des règles de gestion des données (en fonction de la source, du type de données, de leur ancienneté, ...).
3. Traitement parallélisé des données.
4. Écriture du résultat dans un système d'analyse (base SQL, système de fichiers).

A noter que l'écriture des résultats dans un système HDFS, couplé à une utilisation de Hive permet d'exploiter au format SQL les données consolidées.

Audit/qualité des données



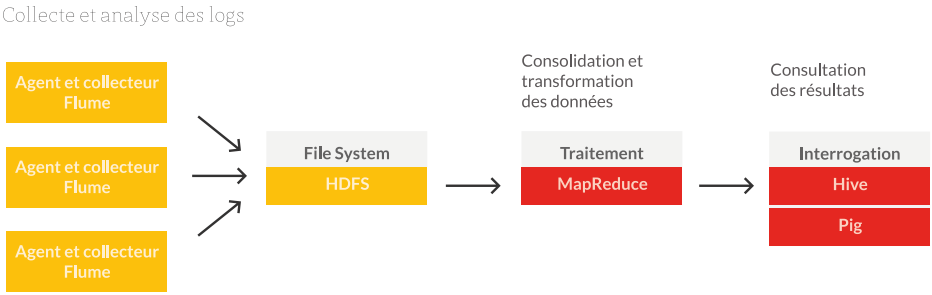
11.2. Audience d'un site (Analyse des logs)

Les sites marchands ont un trafic très important et un besoin de mesures d'audience et de comportement très réactif de manière à prendre des décisions rapidement pour l'animation de ce site.
Cela génère un volume de données particulièrement important et se pose le problème de la performances des systèmes d'analyse de l'audience d'un site lorsqu'ils sont synchrones.
C'est pourquoi Hadoop est particulièrement préconisé dans le cas d'analyse de logs.

Même si il n'y a pas de standard d'utilisation (car cela dépend des informations présentes dans les logs), on retrouve généralement les indicateurs suivants :

- Le parcours type,
- Le temps passé sur chacune des pages,
- Les produits les plus consultés,
- Les erreurs rencontrées,
- Analyse des tentatives de fraude,
- ...

Il existe d'ailleurs des composants spécifiques dédiés à l'analyse des logs comme Apache Flume, voire des éditeurs spécialisés comme Cloudera ainsi que des offres SaaS comme Loggly.

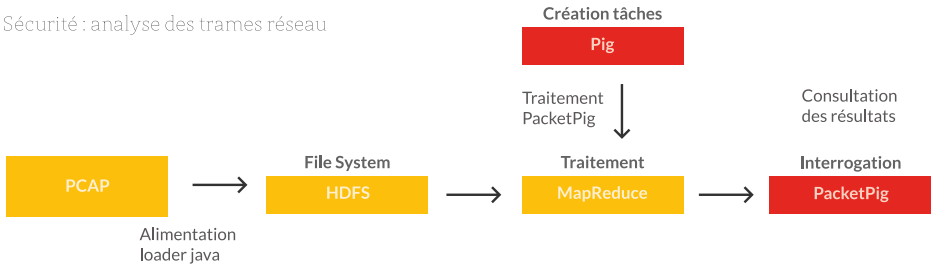


11.3. Sécurité : Analyse du trafic d'un site

Toujours dans le cas de sites web publics, Hadoop peut être utilisé pour le monitoring de la sécurité des réseaux et la détection des intrusions en capturant les paquets IP.
Il existe un projet Packetpig basé sur Apache Pig, de la société Packetloop.
C'est en fait un analyseur de données issues de pcap (librairie réseau) adapté pour Hadoop.
Grâce à des loaders Java spécifiques, chaque paquet est stocké sur le cluster HDFS en vue d'une analyse par le système Hadoop.
En décorrélant ainsi le flux réseau en temps réel et l'analyse, ce système est quasiment sans impact sur les performances.
De plus, l'utilisation d'Hadoop permet de s'affranchir des volumétries importantes inhérentes à l'analyse des trames du réseau et de conserver l'historique des échanges.

Parmi les possibilités offertes, on trouve :

- Faille de type « zero day »,
- Scanning des ports,
- Tentatives d'intrusion,
- Paquets inhabituels (commandes jamais vues auparavant sur le réseau)...



11.4. Data Lake

Un des cas d'utilisation particulièrement répandu d'Hadoop est le « Data Lake » qui va contenir dans un endroit unique l'ensemble des données d'une entreprise.
Le besoin est de stocker de gros volumes de données à structures variables, mais surtout dont on ne sait pas à l'avance comment elles vont être utilisées et analysées.

Il se distingue d'un entrepôt de données classique (Data Warehouse) par sa non structuration des données et son approche non dimensionnelle.

Dans un Data Lake, le format de stockage est le plus brute possible, la transformation se fait uniquement au dernier moment et dans le format le plus adapté au système utilisé (le plus souvent par le système cible lui-même).
En conservant la donnée brute, sans structure, les possibilités ultérieures d'analyse ne sont pas bridées.

Toutes les données sont stockées au même niveau (pas de hiérarchie des données), sous forme d'une multitude de fichiers distribués.
Eventuellement, des métadonnées sont ajoutées aux données afin de faciliter le requêtage et l'analyse de données.

Cette structure plate convient très bien aux données dont on décide de conserver l'historique sans forcément savoir par avance quelles analyses leur seront appliquées.

L'avantage de cette approche est de ne pas typer les données en fonction du système cible, ce qui favorise les changements de format et les changements de système d'interrogation.

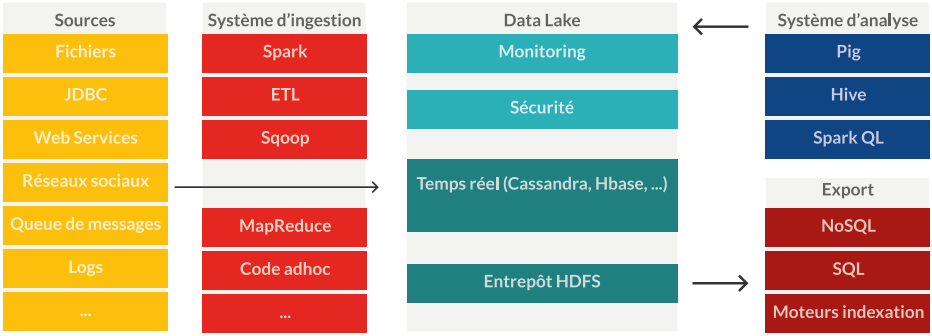
Un Data Lake est caractérisé par les éléments suivants :

Collecte de l'ensemble des données au format le plus brute possible (non structuré, non relationnel, ...).
Accessibilité très large : des utilisateurs d'horizons divers pour explorer, travailler et enrichir la donnée.
Types d'accès multiples et spécialisés : batch, interactif, recherche, ...

Le Data Lake est responsable de :
La sécurité des données
L'archivage (historisation des données)

Les systèmes cibles possibles sont :

- Apprentissage (Machine Learning),
- Moteur d'indexation,
- Applications métier,
- Base NoSQL,
- Base relationnelle,
- Data Mining,
- ...



12 CONCLUSION

12.1. Quand utiliser Hadoop ?

On estime à environ 2500 le nombre de clusters Hadoop actuellement en production dans le monde.

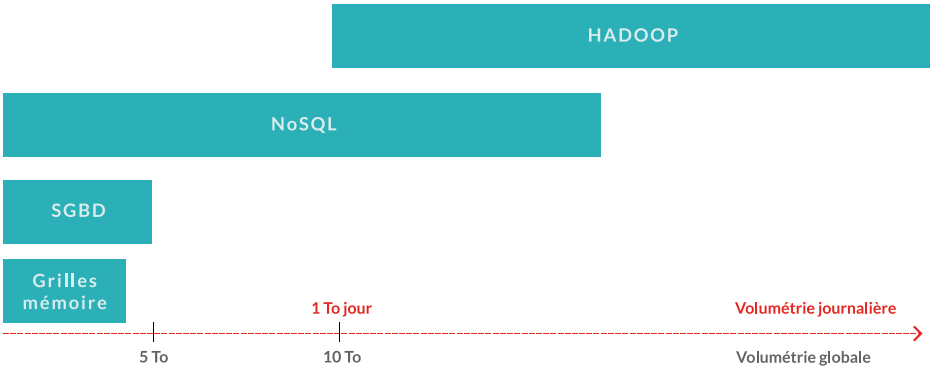
C'est assez limité et ne correspond pas pourtant à une hyper spécialisation de la plateforme, les services fournis semblant assez génériques :
Stockage distribué des données.
Traitements parallélisés.

De même le coût de la plateforme est faible et offre le meilleur ratio coût par To de données.

Cependant, il est vrai que tous les besoins n'ont pas nécessité à être traités par un cluster Hadoop.
Voici les enjeux principaux qui permettent de décider de l'utilisation d'un système Hadoop.

12.1.1. Très forte volumétrie

Plus la volumétrie est importante, plus Hadoop présente un intérêt.



12.1.2. Données immuables

Un des principes fondateurs d'Hadoop est « write once, read many».
C'est pourquoi HDFS n'autorise pas la modification de données existantes mais va plutôt créer un nouveau bloc contenant les modifications.
Il est important d'intégrer ce principe afin d'en tirer avantage : Hadoop est donc optimisé pour les données qui ne changent pas dans le temps.

12.1.3. Scalabilité

Il s'agit évidement de la scalabilité horizontale, c'est à dire la capacité d'un cluster à augmenter sa puissance avec l'ajout de machines.
Cela traduit la capacité d'un système à distribuer automatiquement les traitements et les données sur les ressources du cluster.
Avec Hadoop, on augmente donc la capacité de traitement et de stockage en ajoutant un nœud au cluster.

On peut :

- soit augmenter la capacité de stockage (disque).
- soit augmenter la capacité de traitement (CPU).

12.1.4. Coût stockage

Le coût de stockage est très faible avec Hadoop, et ce, pour les raisons suivantes :

- Solution open source avec un coût de licence contenu.
- Matériel et technologie basique (pas de RAID, OS hétérogène, …)

COÛTS COMPARÉS (HARDWARE ET SOFTWARE(	
HADOOP	$1000 par téraoctets
DATA WAREHOUSE (TERADATA, ORACLE, IBM, …(	$10000-$20000 par téraoctets
SOLUTION PROPRIÉTAIRE SQL	$5000-$10000 par téraoctets

COÛTS STOCKAGE SEULS	
SAN	$5000 par téraoctets
DISQUES LOCAUX	Moins de $1000 par téraoctets

COMPARAISON COÛTS CLOUD	
GOOGLE BIG QUERY	Traitements : $5 par téraoctets/mois
	Stockage : 20 $ par téraoctets/mois

12.2. Comment choisir une solution Hadoop ?

Les trois distributions ont une approche et un positionnement différents en ce qui concerne la vision d'une plate-forme Hadoop (open source, modèle économique...).

Le choix se portera sur l'une ou l'autre solution en fonction des exigences suivantes :

1. Solution open source (portabilité et interchangeabilité).
2. Maturité de la solution.
3. Interconnexion avec le SI.
4. Maturité interne sur les technologies Hadoop (certaines distributions étant plus simples à administrer au dépend de la facilité à configurer finement la solution).
5. Partenariats et compatibilité avec les produits satellites.

Le choix d'une distribution est d'autant plus difficile que l'avenir technologique d'Hadoop est loin d'être tout tracé et que l'écosystème Big Data est en perpétuelle évolution.

Hadoop est né afin de répondre à la problématique suivante : comment traiter des téraoctets de données simplement ?

La réponse proposée alors, un système de fichiers distribué, est arrivée à un moment où il était impossible de traiter de tels volumes de données en mémoire.

Ce n'est plus tout à fait exact :

- le coût de la RAM a fortement baissé.
- les architectures 64 bits se sont généralisées (repoussant la limite de mémoire gérable par l'OS).

Par exemple Ebay gère 250 téraoctets avec Cassandra dont une partie est en mémoire.

La sécurité : sa normalisation est encore balbutiante malgré quelques initiatives comme Apache Knox/Sentry, Ranger. Chacune des distributions va avoir une solution privilégiée.

Un support direct des transactions, ce qui a toujours été un challenge très important dans le monde des données distribuées.

L'intégration avec le Système d'Information : une plate-forme Hadoop isolée et non intégrée au système d'information ne sera plus possible dans le futur (en tout cas, certains besoins exigeront une interaction plus grande).

12.2.1. Cloudera

Le vétéran, ce qui lui donne une légitimité et un nombre de clients supérieur à ses concurrents.
En terme de chiffre d'affaire, c'est le leader.
Un autre avantage est de disposer dans ses rangs de Doug Cutting, le créateur d'Hadoop.
La solution n'est pas 100 % Apache Hadoop mais reste open source (Cloudera Manager, Drill…) par contre ce ne sont pas des licences Apache.
Cloudera ne veut pas se positionner uniquement sur Apache Hadoop et se dirige vers une offre plus large :

- Solutions de type analytics.
- NoSQL.
- Remplacement de HDFS et HBase par Kudu.
- Remplacement de MapReduce par Spark.

Les principaux partenaires sont IBM, HP, Oracle, RedHat et surtout Intel.

A noter que la version gratuite (Cloudera Express) est difficilement utilisable telle quelle en production du fait des limitations des outils de management (Cloudera Manager).

12.2.2. MapR

La plus éloignée d'Apache Hadoop car elle intègre leur propre vision de MapReduce et HDFS. Après Cloudera, c'est la solution la plus mature.
C'est aussi la solution la plus simple à installer et la plus user-friendly.
Ils ont aussi un avantage en terme d'intégration avec le SI grâce à leur utilisation du File System natif compatible Linux.

Beaucoup de partenariats de haut niveau et très stratégiques sur le cloud (Amazon Elastic MapReduce et Google Compute Engine).
Les principaux partenaires sont Dataguise, Datameer, Talend, Teradata.
Tout comme pour Cloudera, la version gratuite (M3) est difficilement utilisable telle quelle en production du fait des limitations des outils de management (MCS).

12.2.3. Hortonworks

C'est la seule plate-forme 100 % Apache Hadoop.
La stratégie assumée d'Hortonworks est de se concentrer sur Hadoop. Ils sont très actifs au sein de la fondation Apache puisqu'ils en sont le principal contributeur.
Hortonworks a un positionnement différent (ils vendent quasi uniquement du service/support et pas de logiciel comme Cloudera/MapR)
Leur solution de gestion du cluster, Ambari, a beaucoup progressé et est maintenant une solution mature.
Hortonworks a signé des partenariats importants avec IBM, Microsoft, Teradata, RedHat, Talend, Pivotal et DataStax.
C'est l'une des plateformes les plus présentes sur le cloud.

Grâce à des rachats de solutions, Hortonworks est en avance en ce qui concerne l'offre de clusters Hadoop sous Docker (http://sequenceiq.com/).
Enfin, Hortonworks est la première compagnie à entrer en bourse (NASDAQ en novembre 2014).

12.2.4. Part de marché

Seul Hortonworks, du fait de son introduction en bourse en 2015, publie officiellement sur son chiffre d'affaires, ses pertes, …

Les autres distributions (MapR et Cloudera) sont plus discrètes car le sujet est sensible (on estime qu'aucun acteur n'est véritablement rentable).
Mais la volonté affichée par ces deux sociétés d'entrer en bourse au cours de l'année 2016 permettra de connaître réellement leur situation financière.

Voici cependant les estimations 2015 pour les trois acteurs :

VENDEUR	ESTIMATION REVENU 2015	CLIENTS PAYANTS	PRÉSENCE
MapR	100-150 Millions $	700	USA, Royaume-Uni, France, Allemagne, Suède, Pays-Bas, Japon, Corée du sud, Australie, Singapour, Inde
Cloudera	199 Millions $	525	USA, Chine, Japon, Royaume-Uni, France, Singapour, Corée du sud, Australie, Inde, Hongrie
Hortonworks	115 Millions $	550	USA, Corée du sud, Australie, Hongrie, Irlande, Japon, Royaume-Uni

12.2.5. Support et formations

Depuis peu, les trois éditeurs proposent un support en France.

Formation et certifications :

- **Cloudera et Hortonworks : online et aussi en France**
- **MapR : essentiellement online**

12.2.6. Synthèse

COMPOSANT	HORTONWORKS	CLOUDERA EXPRESS	MAPR M3
Système de fichiers	HDFS	HDFS	MapR-FS
MapReduce	MapReduce V2	MapReduce V2	MapR MapReduce
API Haut niveau Java	Cascading	-	Cascading
NoSQL	HBase Accumulo	Hase Base	HBase Accumulo
Métadonnées	Hive	Hive	Hive
Scripting	Pig	Pig	Pig
Data analysis	Datafu	Datafu	-
SQL on Hadoop	Hive	Hive	Hive
Ordonnanceur	Oozie	Oozie	Oozie
Coordination du cluster	Zookeeper	Zookeeper	Zookeeper
Interface SGBD-R	Sqoop 2	Sqoop 2	Sqoop 2
Collecte de logs	Flume	Flume	Flume
Machine learning	Mahout	Mahout, Oryx 2	Mahout
Hadoop UI	Hue	Hue	Hue
Requêtage	Tez (Stinger)	Impala	Drill, Impala Spark SQL, Tez
Indexation	Solr	Cloudera search (basé sur Solr	Solr
Administration	Ambari	Cloudera Manager	MapR Control System
Installation cluster	Ambari	Cloudera Manager	Pas en version gratuite
Monitoring	Ambari Metrics (Ganglia, Nagios possible)	Intégration possible avec Ganglia, Nagios	Intégration possible avec Ganglia, Nagios
Sécurité	Knox Ranger	Sentry	Sentry
Management des ressources	YARN	YARN	YARN

COMPOSANT	HORTONWORKS	CLOUDERA EXPRESS	MAPR M3
Micro batch/ Streaming	Storm Spark / Flink	Storm Spark	Storm Spark
Gouvernance	Falcon Atlas	Cloudera Navigator	Partenariats externes
Gestion des messages	Kafka	Kafka	-

12.3. Conclusion

Ce document a présenté les caractéristiques de la plateforme Hadoop ainsi que trois solutions open source répondant à cette définition.
Hadoop fournit une solution open source, scalable et à forte disponibilité pour le stockage et le traitement des données.
Il présente les forces et les faiblesses de chacun des outils et permet de déterminer si la technologie est adaptée à votre besoin.

En résumé, les avantages majeurs d'Hadoop sont :

- Prix : Hadoop est OpenSource et le coût des distributions est limité (support et licence), il permet le traitement des données de la taille de pétaoctets.
- Rapidité : Il permet le traitement parallélisé des données.
- Scalabilité : Selon vos besoins en ajoutant des nœuds.

Hadoop n'est pas une solution fermée, et de nombreux projets s'appuient sur cette base afin de l'améliorer ou proposer de nouveaux services non couverts par cette solution.
Chacune des solutions présentées est viable et même si elles répondent au même besoin, l'une d'elles pourra se révéler plus adaptée à votre contexte : il faut profiter des forces de la solution et ne pas essayer de faire rentrer le produit dans un cadre qui ne lui est pas adapté.

12.4. En savoir plus

Site web Hadoop User Group

Site web Apache Hadoop

Site web Cloudera

Comparaison des éditions Cloudera

Site web MapR

Comparaison des éditions MapR

Site web Hortonworks

Publications à l'origine d'Hadoop (Google) :

Google FS

MapReduce

BigTable

13 REMERCIEMENTS

Je tiens à remercier, pour leur aide à la rédaction de ce livre blanc, les personnes suivantes :

David MARTIN
Victoria DE BELILOVSKY
Alexis SEIGNEURIN