

IPPON

Digital . Technologies . Hosting

ECOSYSTÈME HADOOP

Par CHRISTOPHE PARAGEAUD
MANAGER TECHNIQUE

NOV
2013



 www.ippon.fr - <http://blog.ippon.fr>

 [#hadoopippon](https://twitter.com/hadoopippon)


SOMMAIRE

1 A propos d'Ippon Technologies	4
1.1 Nos Solutions.....	4
1.2 Notre offre « Data Integration ».....	4
1.3 Nous contacter.....	5
2 Licence	5
3 Présentation de l'étude	5
3.1 BigData.....	5
3.2 Hadoop.....	6
4 Présentation Hadoop	6
4.1 Les 4 V du BIG DATA.....	6
4.1.1 Volume.....	6
4.1.2 Variété.....	6
4.1.3 Vélocité.....	6
4.1.4 Variabilité.....	7
4.2 Hadoop versus SGBDR.....	7
5 Le cœur : Hadoop kernel	8
5.1 HDFS (Hadoop Distributed File System).....	8
5.1.1 Alternatives.....	9
5.2 MapReduce.....	9
5.2.1 Alternative.....	9
5.3 Format de stockage.....	9
5.3.1 Zoom sur Apache Avro.....	12
5.3.2 Synthèse.....	12
6 Les extensions	13
6.1 Requêtage des données : Hive (Facebook).....	13
6.2 Scripting sur les données : Pig (Yahoo).....	13
6.3 Intégration SGBD-R : Sqoop (Cloudera).....	13
6.4 Ordonnanceur : Apache Oozie (Yahoo).....	13
6.5 Gestion des clusters Hadoop.....	13
6.5.1 Clustering.....	13
6.5.2 Supervision.....	14
6.6 Autres composants Hadoop.....	14
6.6.1 Apache Flume (Cloudera).....	14
6.6.2 Apache Mahout.....	14
6.6.3 Apache Drill (MapR).....	14
6.6.4 Apache HCatalog (HortonWorks).....	15
6.6.5 Apache Tez (HortonWorks).....	15
7 Vue d'ensemble de la plate forme Hadoop	15

8 Les distributions	17
8.1 HortonWorks	17
8.1.1 Composants de la plate forme HDP (HortonWorks Data Platform)	17
8.1.2 Présentation	17
8.1.3 Vision d'ensemble de la distribution	17
8.1.4 Déploiement de la plate forme	18
8.2 Cloudera	18
8.2.1 Présentation	18
8.2.2 Composants de la plate forme CDH (Cloudera's Distribution including Apache Hadoop)	18
8.2.3 Vision d'ensemble de la distribution	19
8.2.4 Déploiement de la plate forme	19
8.3 MapR	19
8.3.1 Présentation	20
8.3.2 Contenu de la distribution MapR M3	20
8.3.3 Vision d'ensemble de la distribution	21
8.3.4 Déploiement de la plate forme	21
9 Les cas d'utilisations d'Hadoop	22
9.1 Audit/Qualités des données	22
9.2 Audience d'un site (Analyse des logs)	23
9.3 Sécurité : Analyse du trafic d'un site	24
10 Conclusion	25
10.1 Comment choisir une solution Hadoop ?	25
10.1.1 Cloudera	25
10.1.2 MapR	25
10.1.3 HortonWorks	26
10.2 Récapitulatif	26
10.3 En savoir plus	26
12 Remerciements	26

1 A PROPOS D'IPPON TECHNOLOGIES

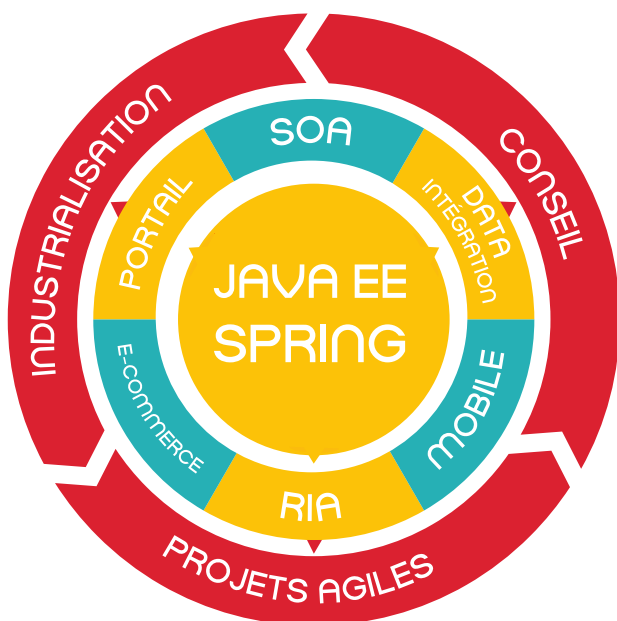
Ippon Technologies est une société de services informatiques spécialisée dans les technologies Java EE. Société fondée début 2002 sous la double impulsion de Sportifs de Haut Niveau et de Managers issus de grandes sociétés de conseil en technologies, Ippon Technologies s'est développé autour de l'expertise nécessaire à la conception et l'intégration de solutions logicielles à la pointe des nouvelles technologies de l'entreprise, et notamment sur la plate-forme Java EE. Notre force passe avant tout par la richesse de notre potentiel humain et notre positionnement original sur le marché de l'externalisation. L'ensemble de nos consultants sont des professionnels aguerris dans leur secteur et ont à leur actif des expériences déterminantes pour le développement des entreprises.

Ippon Technologies intervient dans les domaines de l'industrie, de la distribution, des télécommunications, des services, de la banque et de la finance.

1.1 Nos Solutions

Expert historique des développements spécifiques Java EE, Ippon Technologies a fait évoluer ses compétences vers un ensemble cohérent d'offres fonctionnelles et techniques.

Ces solutions mises en place par Ippon Technologies permettent d'apporter une réponse de choix aux problématiques d'entreprise suivantes :



Rénover, par la refonte d'applications stratégiques vers les nouvelles technologies et les nouveaux usages.

Fédérer, par la mise en place de portails d'entreprise collaboratifs et mobiles.

Intégrer, en facilitant l'échange d'information entre les systèmes à l'aide des technologies d'EAI, d'ETL et d'ESB.

Diffuser, en facilitant l'accès à l'information et aux données par la mise en place d'interfaces riches et mobiles.

1.2 Notre offre « Big data » et Ippon Hosting

Sans données de qualité, les plus belles applications ne servent à rien.

C'est fort de ce constat qu'Ippon Technologies a toujours disposé au sein de ses équipes de spécialistes des données (Architectes et DBAs Oracle et PostgreSQL, spécialistes ETL ou grilles de données) qui complètent à merveille nos équipes applicatives.

Cette offre « Data Integration » couvre aujourd'hui un large spectre de solutions en plus des bases : ETL, MDM, grilles de données, reporting, NoSQL,...

En fonction des exigences de nos clients, nous intervenons en conseil (architecture, choix de solutions), accompagnement de projet et industrialisation (audits, tests, performances,...).

Nous accompagnons sur la partie grille de données des projets e-commerce ou transactionnels majeurs qui requièrent des niveaux de performance très élevés que ce soit en temps de réponse ou garantie de disponibilité.

1.3 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 :

PARIS

01 46 12 48 48

43-47 avenue de la
Grande Armée
75116 PARIS

TOULOUSE

05 34 51 23 60

Immeuble Burolines II
2 ter, rue Marcel Doret
31700 Blagnac

NANTES

02 40 48 28 06

1 Rue Du Guesclin
44019 Nantes

BORDEAUX

05 35 54 62 26

61 cours de l'Intendance
33000 Bordeaux

Posez aussi vos questions sur Twitter à l'aide du hashtag : #hadoopippon 

2 LICENCE

Ce document vous est fourni sous licence Creative Commons Attribution Share Alike.

Vous êtes libres de reproduire, distribuer et communiquer cette création au public selon les conditions suivantes :

- Paternité. Vous devez citer le nom des auteurs originaux mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre.
- A chaque réutilisation ou distribution de cette création, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition sous licence identique Creative Commons Share Alike.
- Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits sur cette œuvre.
- Rien dans ce contrat ne diminue ou ne restreint le droit moral de l'auteur ou des auteurs.

3 PRÉSENTATION DE L'ÉTUDE

3.1 BigData

Le terme BigData reflète l'augmentation exponentielle de la volumétrie des données d'un SI que l'on a pu connaître ces dernières années.

En plus de cette augmentation, les nouveaux challenges auxquels sont confrontés les entreprises sont la vitesse et le coût du traitement de ces données.

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 monde relationnel des bases de données traditionnelles pour un monde 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).

3.2 Hadoop

En 2004, Google a publié un article présentant son algorithme de calcul à grande échelle, MapReduce, ainsi que son système de fichiers 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 :

1. Ce sont des technologies jeunes.
2. Beaucoup de buzz et de communication de sociétés qui veulent prendre le train Big Data en marche.
3. Des raccourcis sont souvent employés (non MapReduce ou un équivalent n'est pas suffisant pour parler d'Hadoop).
4. 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, HCatalog, Oozie, Pig, Sqoop, ...

Ces solutions sont des projets Apache et donc disponibles mais l'intérêt d'un package complet est évident : compatibilité entre les composants, simplicité d'installation, support, ...

Dans ce livre blanc 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 :

- **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) comme Intel Distribution for Hadoop ou bien Greenplum (Pivotal HD).

4 PRÉSENTATION HADOOP

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

4.1 Les 4 V du BIG DATA

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

4.1.1 Volume

Les entreprises font face à une augmentation exponentielle des données (plusieurs milliers de Tera Octets) Les technologies actuelles (BI, Data Mining) n'ont pas été pensées pour de telles volumétries.

4.1.2 Variété

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, ...).

NB : 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.

4.1.3 Vitesse

Dans certains cas, l'accès et le partage des données doit se faire en temps réel (on verra par la suite que ce n'est pas toujours vrai pour Hadoop).

4.1.4 Variabilité

La solution doit être évolutive car les formats de données ne sont pas actuellement connus (voir par exemple comment le format JSON a supplanté XML très rapidement).

4.2 Hadoop versus SGBDR

Parce qu'au sein d'un SI, un seul type 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 :

	SGBDR	Hadoop
Volumétrie	Giga octets	Péta octets
Type d'accès	Temps réel et batch	Batch
Mises à jour	Lecture et écriture multiples	Une seule écriture, lectures multiples
Structure	Schéma pré défini	Schéma évolutif
Intégrité	Forte	Faible
Scalabilité	Non linéaire	Linéaire

5 LE COEUR : HADOOP KERNEL

Hadoop est le framework le plus utilisé actuellement pour manipuler et faire du Big Data.

Apache Hadoop est un framework qui va permettre le traitement de données massives sur un cluster allant d'une à plusieurs centaines de machines. C'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).

C'est lui qui 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 nœuds (principe d'un nœud primaire et de nœuds secondaires).

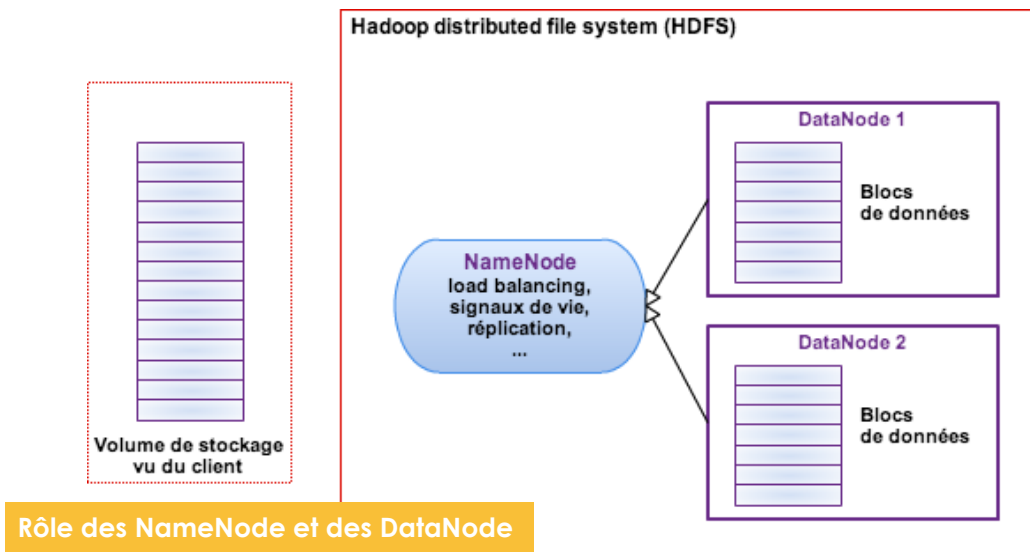
5.1 HDFS (Hadoop Distributed File System)

HDFS est un système de fichiers Java utilisé pour stocker des données structurées ou non sur un ensemble de serveurs distribués.

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

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

1. Node (Master/slave) : Dans une architecture Hadoop chaque membre pouvant traiter des données est appelé Node (Nœud).
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. Il est responsable de la localisation des données dans le cluster
4. Les autres nœuds, stockant les données, sont des slaves appelés DataNode.
5. Bien qu'il puisse y avoir plusieurs NameNode, la "promotion" doit se faire manuellement (Hadoop 2.0, actuellement en version alpha, introduit un failover automatisé).
6. Le NameNode est donc un Single Point Of Failure (SPOF) dans un cluster Hadoop.



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

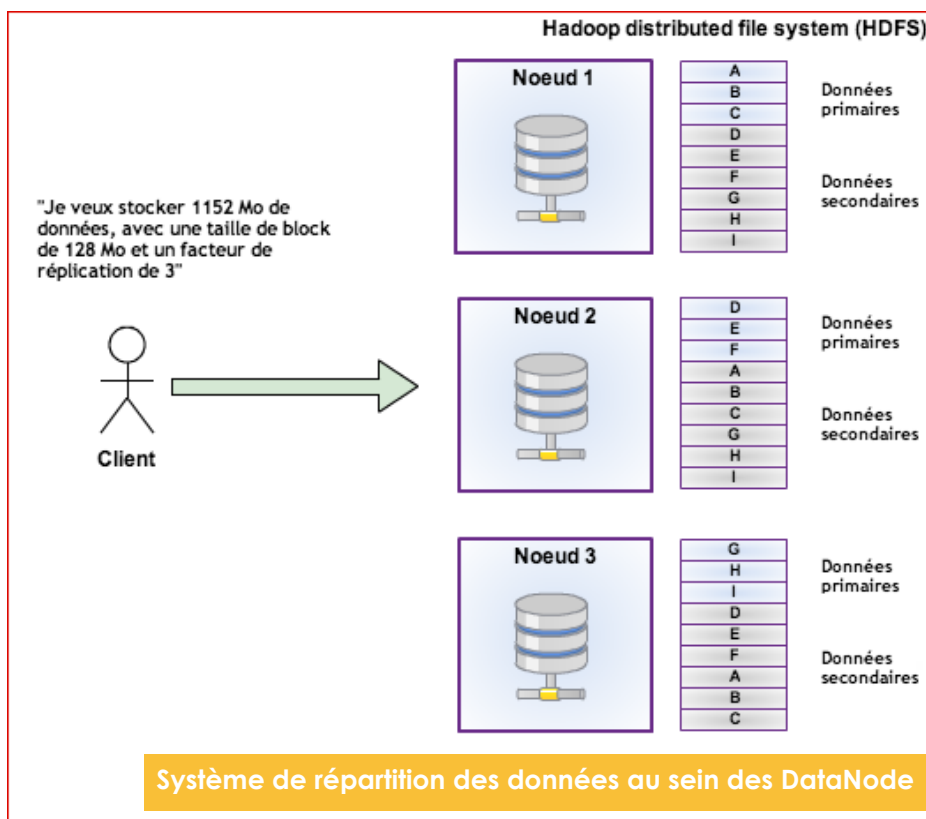
- **Blocksize** : Taille unitaire de stockage (généralement 64 Mo ou 128 Mo). C'est à dire qu'un fichier de 1 Go (et une taille de block de 128 Mo) sera divisé en 8 blocks.
- **Replication factor** : C'est le nombre de copies d'une données devant être réparties sur les différents noeuds du cluster (souvent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

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.

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).



5.1.1 Alternatives

5.1.1.1 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.

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. (Exemple : comment le format JJson a supplanté XML très rapidement).

Contrairement à HDFS, HBase permet de gérer les accès aléatoires read/write pour des applications de type temps réel.

5.1.1.2 MapR

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.

5.1.1.3 Cassandra (Apache)

Cassandra est une base de données orientée colonnes, développée sous l'impulsion de Facebook (qui l'a abandonné 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 meilleur 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.

5.1.1.4 Offre Cloud

Le cloud est un complément idéal au monde Hadoop, en offrant des possibilités de stockage et de traitement extensibles.

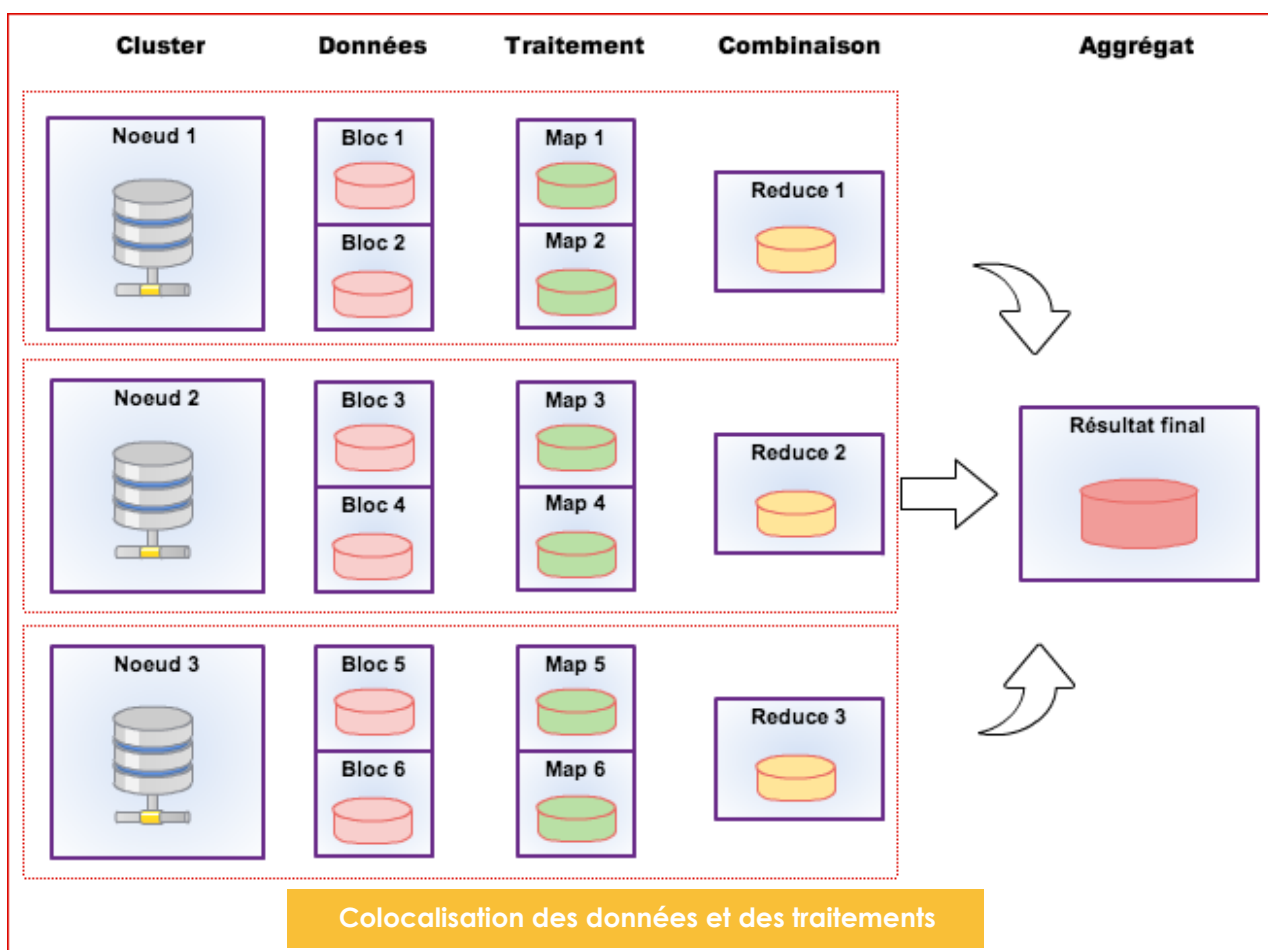
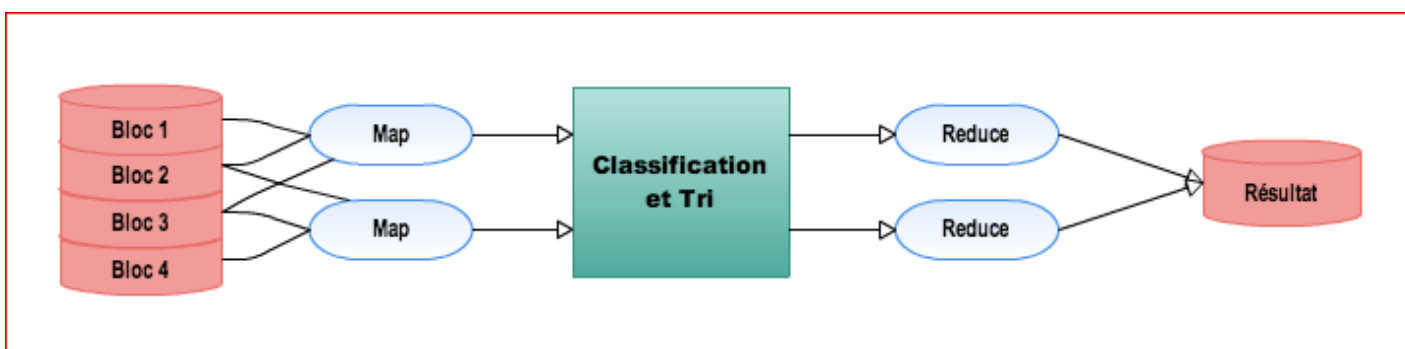
Il est donc possible d'utiliser un système de fichiers situé dans le cloud pour le stockage des données et l'exécution des traitements.

Solutions supportées :

- **Amazon S3,**
- **Kosmix's CloudStore,**
- **IBM GPFS (General Parallel File System).**

5.2 MapReduce

A l'origine crée par Google pour son outil de recherche web. C'est un framework qui permet le décomposition d'une requête importante en un ensemble de requêtes plus petites qui produiront 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.



5.2.1 Alternative

5.2.1.1 YARN (HortonWorks)

YARN (Yet-Another-Resource-Negotiator) est aussi appelé MapReduce 2.0, ce n'est pas une refonte mais une évolution du framework MapReduce.

YARN apporte une séparation claire entre les problématiques suivantes :

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

YARN est compatible avec les anciennes versions de MapReduce (il faut simplement recompiler le code).

On retrouve cette évolution dans la version Hadoop 2.0+ chez Apache.

5.3 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 noeuds.

Les formats supportés sont :

- **Format texte de type CSV,**
- **Json,**
- **Binaire (Hadoop Serialization),**
- **Binaire (Avro, Thrift, ...).**

Les formats Texte et Json vont présenter l'avantage d'être compréhensibles par des humains mais l'inconvénient d'être peu optimisé pour le stockage et les performances.

Les formats binaires vont au contraire présenter l'avantage d'être plus performants.

5.3.1 Zoom sur Apache Avro

Apache Avro est un système de sérialisation de données qui à été conçu par Doug Cutting comme indépendant des langages.

Il existe une implémentation pour les langages suivants : C, C++, C#, Java, PHP, Python et Ruby.

De plus le fait de ne pas être lié à un langage (et à ses types) permet une évolution indépendante du type de données supportée 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.

5.3.2 Synthèse

Format	Avantages	Inconvénients
Format texte	Lisible par les humains	Simpliste (difficile de représenter les objets liés entre eux) Performances Peu flexible
JSON	Flexible, Lisible par les humains	Performances
Hadoop Serialization	Performances	Pas de compatibilité ascendante prévue
Binaire (Thrift, Avro, ...).	Performances Flexible	Encore un nouveau format !

6 LES EXTENSIONS

6.1 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 SQL sur un cluster Hadoop en vue d'analyser et d'agréger les données. Le langage SQL 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.

6.2 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.

Contrairement à Hive, Pig ne dispose pas d'interface web.

6.3 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/vers Hadoop et Hive.

Pour la manipulation des données Sqoop utilise MapReduce et des drivers JDBC.

6.4 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.**

6.5 Gestion des clusters Hadoop

6.5.1 Clustering

6.5.1.1 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 HBase.

6.5.2 Supervision

6.5.2.1 Apache Ambari (HortonWorks)

Ambari est un projet d'incubation Apache initié par HortonWorks et 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,**
- **Des informations issues de Ganglia et de Nagios,**
- **De l'exécution des jobs,**
- **Des métriques de chaque machine et du cluster.**

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, 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 outils annoncés sont :

- **Hadoop,**
- **HDFS,**
- **MapReduce,**
- **Hive, HCatalog,**
- **Oozie,**
- **HBase,**
- **Ganglia, Nagios.**

6.6 Autres composants Hadoop

6.6.1 Apache Flume (Cloudera)

Flume est une solution de collecte et d'agrégation de fichiers 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.

Alternatives : Apache Chukwa.

6.6.2 Apache Mahout

Apache Mahout est un projet de la fondation Apache visant à créer des implémentations d'algorithmes d'apprentissage automatiques et de dataminings.

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.

6.6.3 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.

6.6.4 Apache HCatalog (HortonWorks)

HCatalog permet l'interopérabilité d'un cluster de données Hadoop avec des systèmes externes. HCatalog est un service de management de tables et de schéma des données Hadoop :

- Permet d'attaquer les données HDFS via des schémas de type tables de données en lecture/écriture.
- Permet d'opérer sur des données issues de MapReduce, Pig ou Hive.

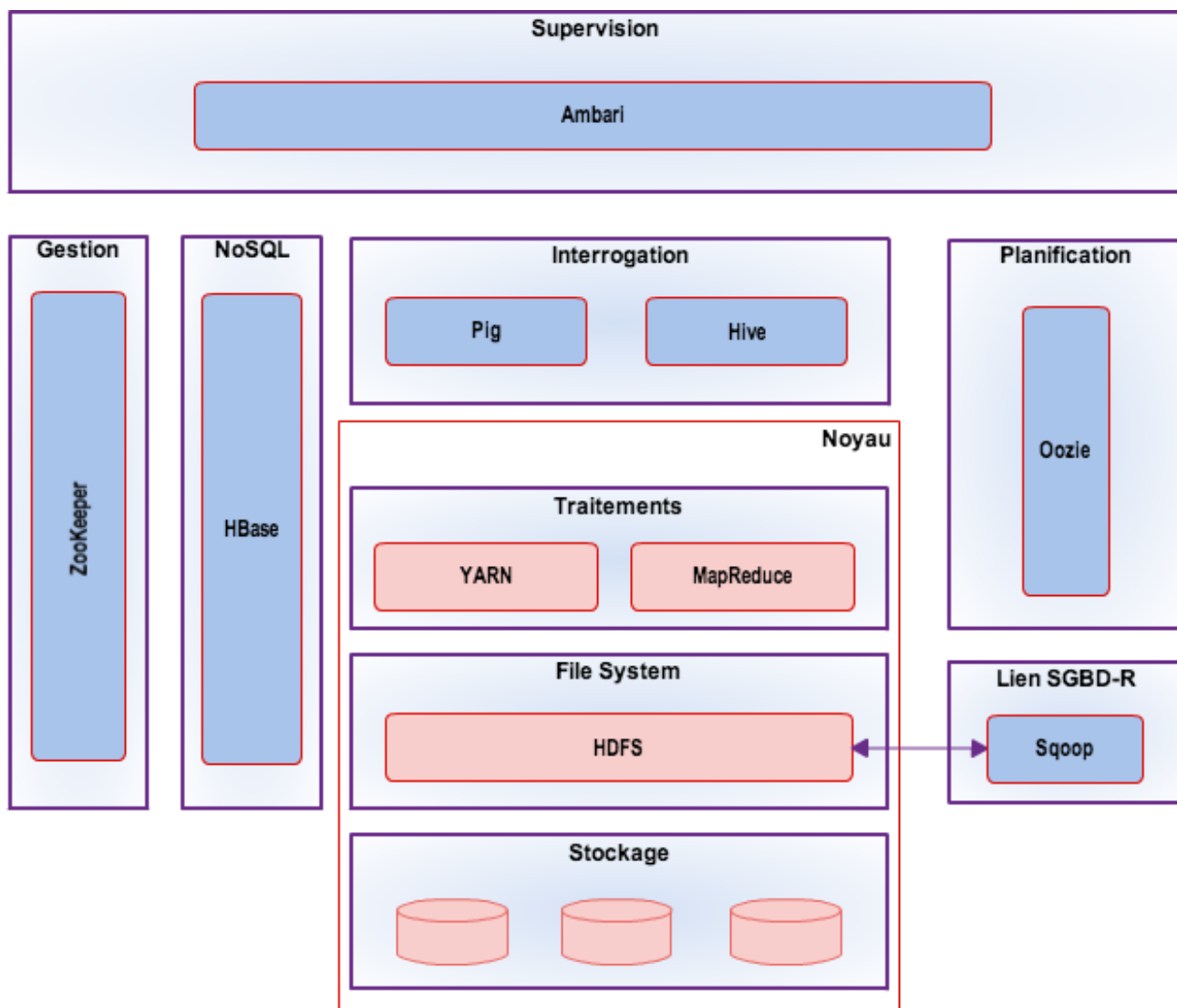
6.6.5 Apache Tez (HortonWorks)

Tez est un nouveau framework en incubation chez Apache.

Utilisant YARN il remplace MapReduce afin de fournir des requêtes dites "temps réel". La faible latence est en effet un pré requis à l'exploration interactive des données stockées sur un cluster Hadoop.

C'est un concurrent d'Apache Drill (MapR) et de Cloudera Impala.

7 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

8 LES DISTRIBUTIONS

8.1 HortonWorks

8.1.1 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 uniquement du support et des formations.

Cette distribution est la plus conforme à la plate forme Hadoop d'Apache et HortonWorks est un gros contributeur Hadoop.

Parmi les projets reversés il y a :

- **YARN,**
- **HCatalog,**
- **Ambari,**

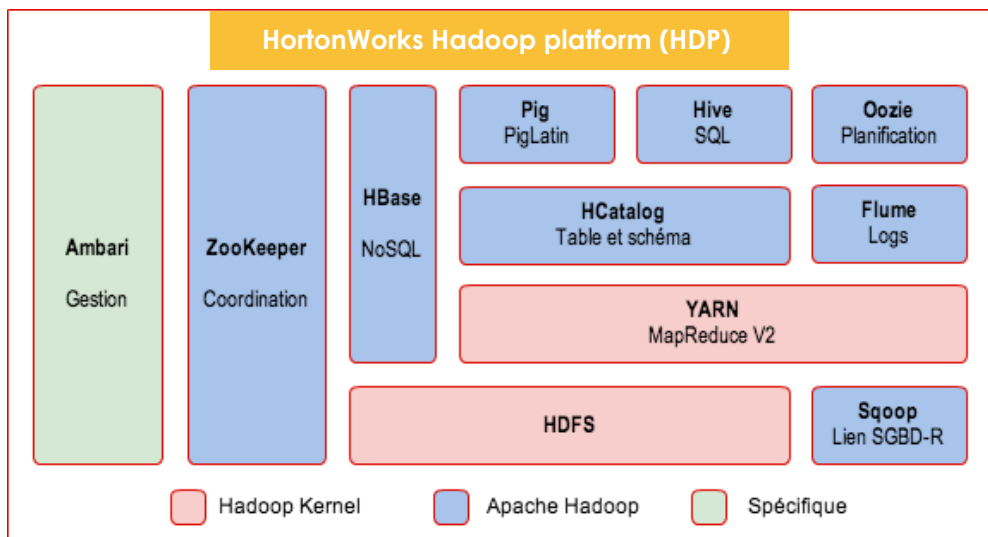
...

8.1.2 Composants de la plate forme HDP (HortonWorks Data Platform)

Les éléments suivants composent la plate forme HortonWorks :

1. Cœur Hadoop (HDFS/MapReduce).
2. NoSQL (Apache HBase).
3. Méta-données (Apache HCatalog).
4. Plate forme de script (Apache Pig).
5. Requêtage (Apache Hive).
6. Planification (Apache Oozie).
7. Coordination (Apache ZooKeeper).
8. Gestion et supervision (Apache Ambari).
9. Services d'intégration (HCatalog APIs, WebHDFS, Talend Open Studio for Big Data, Apache Sqoop).
10. Gestion distribuée des logs (Apache Flume).
11. Apprentissage (Apache Mahout).

8.1.3 Vision d'ensemble de la distribution



8.1.4 Déploiement de la plate forme

8.1.4.1 Machine Virtuelle prête à l'emploi

HortonWorks met à disposition une machine virtuelle où sont pré installés les composants de la plate forme Hadoop.

C'est l'idéal pour l'apprentissage de la plate forme mais incompatible avec les exigences de production ou même celles d'un POC.

8.1.4.2 Installation automatique avec Ambari

En plus de la gestion du cluster, Ambari permet le déploiement de l'ensemble des composants Hadoop de manière centralisée.

8.1.4.3 Installation manuelle avec Linux RPM

HortonWorks met à disposition des packages RPM.

En utilisant le principe des RPM Linux il est possible d'installer les composants HDP manuellement.

8.2 Cloudera

8.2.1 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 essentiellement pour la gestion du cluster.

A noter aussi que la version d'Apache Hadoop distribuée est la dernière version stable complétée de patches 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).

8.2.2 Composants de la plate forme CDH (Cloudera's Distribution including Apache Hadoop)

Composants Apache :

- **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 (log) dans Hadoop,
- **ZooKeeper** : Service de coordination pour les applications distribuées,
- **Mahout** : Framework d'apprentissage et de datamining pour Hadoop.

Composants d'origine Cloudera :

- **Hadoop Common**: Un ensemble d'utilitaires,
- **Hue** : SDK permettant de développer des interfaces utilisateur pour les applications Hadoop,
- **Whirr** : Bibliothèques et scripts pour l'exécution d'Hadoop et de services liés dans le cloud.

Composants non Apache Hadoop :

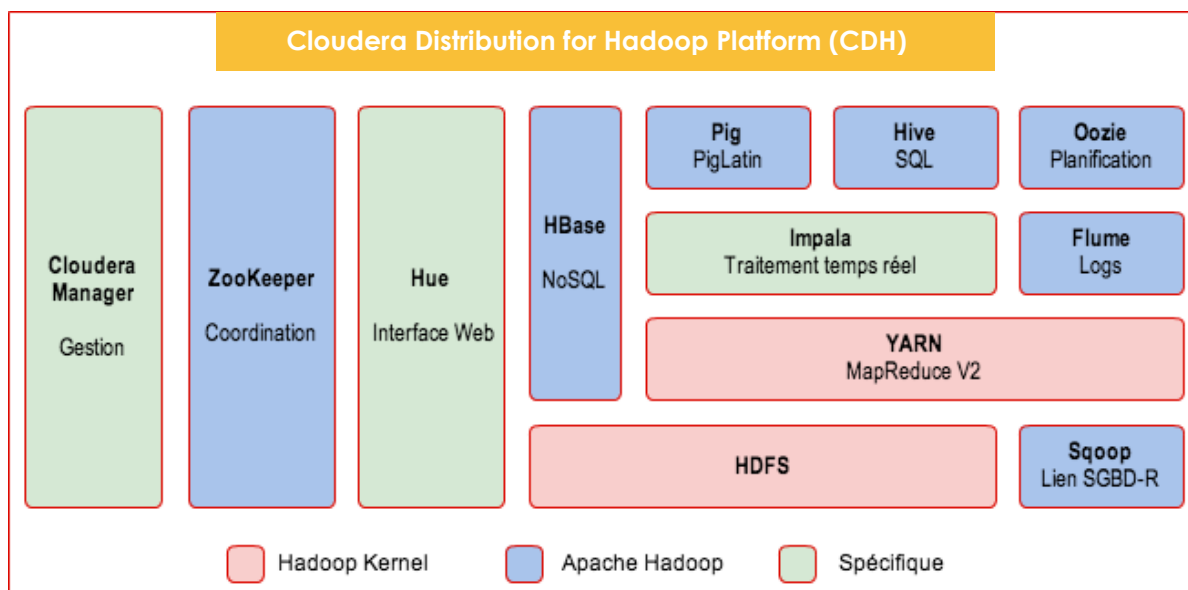
Cloudera Impala : Moteur temps réel de requêtage SQL parallélisé de données stockées dans HDFS ou HBase. Contrairement à Hive de Hadoop, Impala n'utilise pas le framework MapReduce qui exige que les résultats de recherche soient écrits sur le disque, ce qui lui permet d'exécuter les requêtes plus rapidement.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

Cloudera Manager : 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 :

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

8.2.3 Vision d'ensemble de la distribution



8.2.4 Déploiement de la plate forme

8.2.4.1 Automatique avec Cloudera Manager

Cloudera Manager permet l'installation des composants de la plate forme sur une machine (y compris distante).

Cloudera Manager permet la configuration centralisée des composants du cluster.

Enfin Cloudera Manager permet de finaliser l'installation en vérifiant le bon fonctionnement de chacun des composants.

8.2.4.2 Manuel avec les packages

Récupération des archives tarball (tgz) contenant la distribution.

Configuration et installation à l'aide des scripts fournis.

8.3 MapR

8.3.1 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. Ils proposent ainsi leur propre système de fichiers distribués ainsi que leur 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 :

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

8.3.2 Contenu de la distribution MapR M3

Composants Apache :

- **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 :

1. Système plus adapté au mode read/write que HDFS.
2. MapR intègre un serveur NFS (Network File System) pour l'intégration au SI de l'entreprise.
3. Simplification de mise en oeuvre (surcouche du File System de l'OS et non remplacement comme HDFS).
4. Plus de Single Point Of Failure.

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

Avantages :

1. MapR annonce de meilleures performances.
2. 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 aussi bien 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 Vaidya

Hadoop Vaidya est un outil d'analyse des performances des jobs MapReduce.

Son principe de fonctionnement est basé sur des règles qu'il confronte aux statistiques d'exécution des jobs et aux fichiers de configuration.

Le rapport est produit au format XML.

- Apache Drill

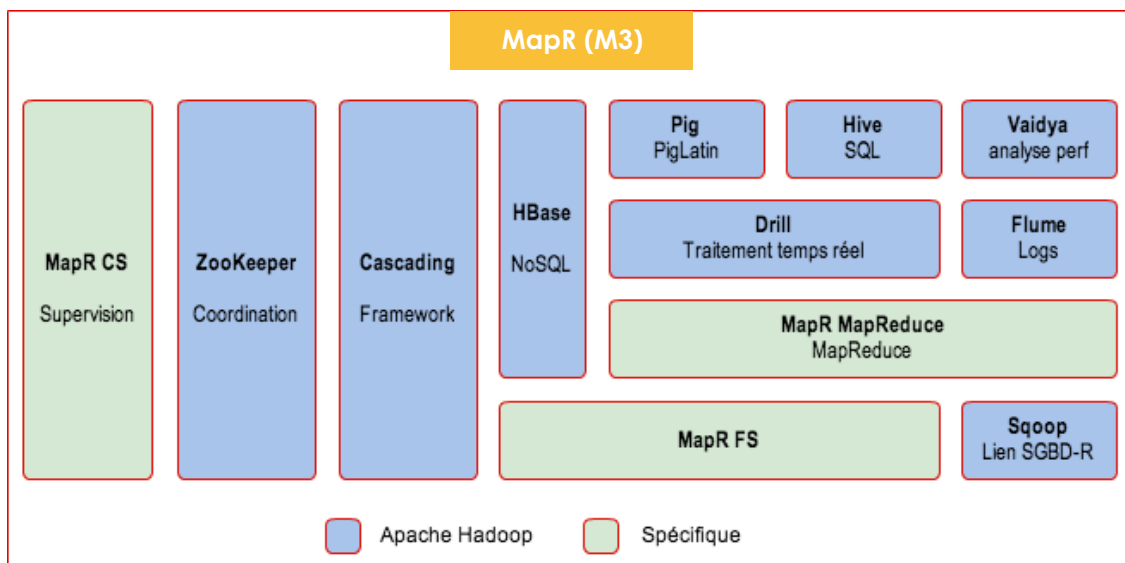
MapReduce a la réputation d'être puissant mais complexe à manipuler (il faut en maîtriser l'API).

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.

SQL plutôt qu'une nouvelle API, c'est donc le choix de la capitalisation fait par Drill.

8.3.3 Vision d'ensemble de la distribution



8.3.4 Déploiement de la plate forme

8.3.4.1 Machine virtuelle

MapR fournit une machine virtuelle avec un seul noeud et l'ensemble des composants installés. C'est l'idéal pour une prise en main de la plate-forme mais incompatible avec les exigences de production.

8.3.4.2 Manuel avec les packages

MapR ne fournit pas de système de déploiement Hadoop. L'installation est donc essentiellement manuelle avec des automatisations possibles.

Tout d'abord il faut récupérer les composants à installer :

1. Depuis le repository internet
2. Depuis un repository local
3. Avec des packages Debian/Linux

Après édition de la configuration il faut ensuite exécuter les scripts fournis pour installer les composants MapR sur chaque machine.

A noter que la distribution ne contient pas les composants Apache et qu'il faut les installer manuellement.

9 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 des ETL mais 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.

9.1 Audit/Qualités des données

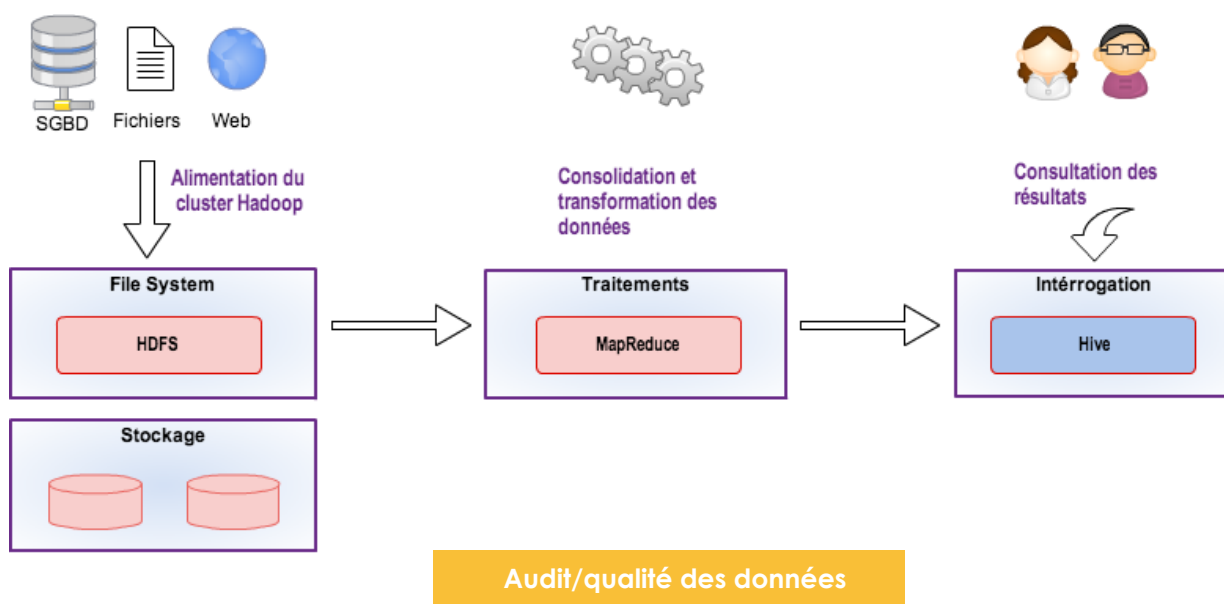
Historiquement le SI va cumuler des données sous des formats variables (car provenant de sources différentes) et pas toujours correctes (évolutions des règles de validation, système de saisie défaillant, ...) Elles ont donc besoin d'être consolidées avant d'être exploitées :

- **Uniformisation du format des données,**
- **Correction ou purges des données incorrectes**

Voici les différentes 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.



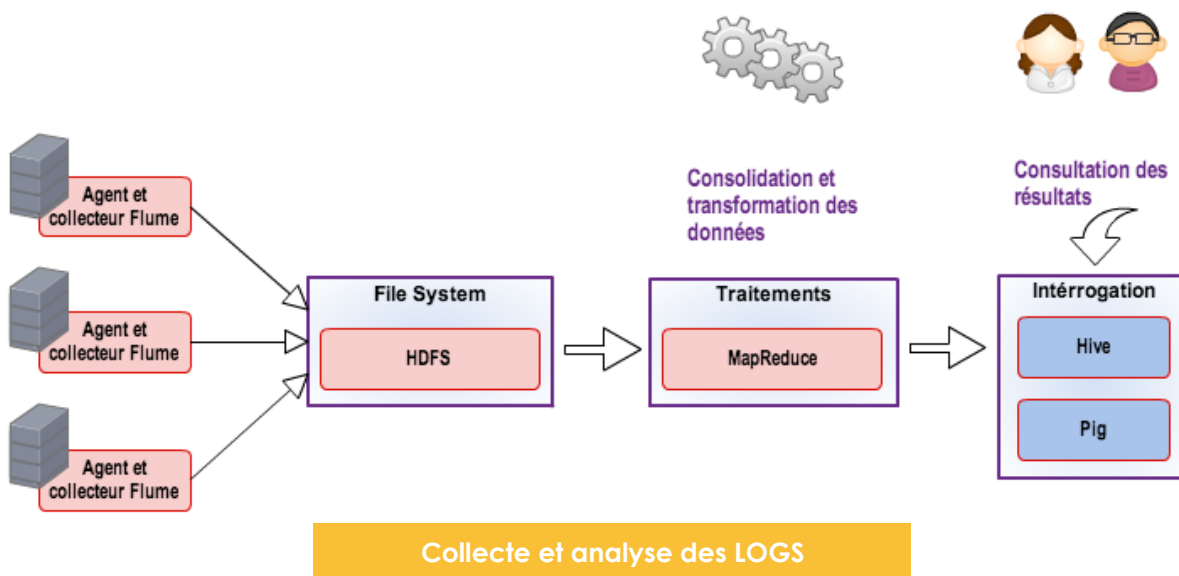
9.2 Audience d'un site (Analyse des logs)

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 log.

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 suivant :

- **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 composant spécifiques, dédiés à l'analyse des logs comme Apache Flume voire des éditeurs spécialisés comme Cloudera ainsi que des offre SAAS sur le cloud comme **Loggly**.



9.3 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 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 (bibliothèque réseau) adapté pour Hadoop.

Grâce à des loader java spécifiques chaque paquet est stocké sur le cluster HDFS en vue d'une analyse par le système Hadoop.

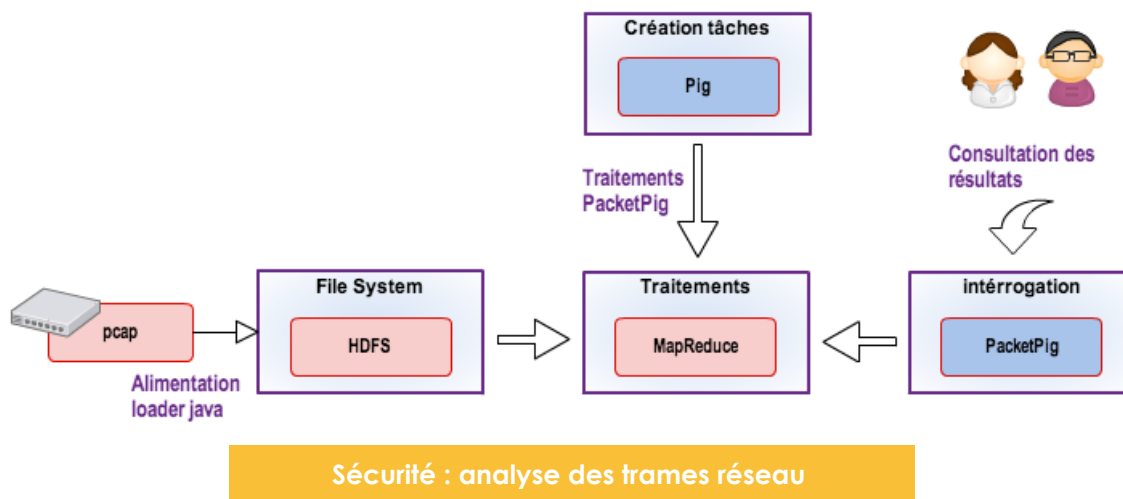
En décorrélant ainsi le flux réseau en temps réel et l'analyse, ce système est quasiment sans impact sur les performances.

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 :

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

...



10 CONCLUSION

10.1 Comment choisir une solution Hadoop ?

Les trois distributions ont une approche et un positionnement différent 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 :

1. Solution open source.
2. Maturité de la solution.
3. 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.

En effet des virages technologiques importants sont d'ores et déjà annoncés :

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

La réponse proposée alors, un système de fichiers distribués, est arrivée à un moment où il était impossible de traiter de tels volumes de données en mémoire. Maintenant le coût de la RAM a fortement baissé et avec la généralisation des architecture 64 bits ce n'est plus tout à fait exact (par exemple Ebay gère 250 Tera octets avec Cassandra dont une partie est en mémoire).

La sécurité : elle est encore balbutiante malgré quelques initiatives comme Apache Knox.

L'intégration avec le SI, 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).

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

10.1.1 Cloudera

Le vétéran ce qui lui donne une légitimité et un nombre de clients supérieur à ses concurrents.

Un autre avantage est de disposer dans ses rangs de Doug Cutting le créateur d'Hadoop.

Cloudera est très prompt à sortir les dernières versions d'Hadoop (ils ont été les premiers à sortir une distribution compatible Hadoop 2.0).

Les principaux partenaires sont IBM, HP, Oracle.

10.1.2 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 grâce à leur utilisation du File System natif.

Beaucoup de partenariats de haut niveau et très stratégiques sur le cloud (Amazon Elastic MapReduce et Google Compute Engine).

10.1.3 HortonWorks

C'est la seule plate forme 100 % Apache Hadoop.

La stratégie assumée d'HortonWorks est de se baser sur les versions stables et testées d'Apache Hadoop plutôt que sur les dernières versions.

Leur solution de gestion du cluster, Ambari, n'est pas aussi mature que la concurrence : Cloudera Manager et HeatMap.

Malgré sa relative jeunesse, HortonWorks a signé des partenariats importants avec IBM, Microsoft, Teradata et Talend. Ils ont notamment signé avec Microsoft un accord pour le déploiement de leur plate forme sur Azure.

10.2 Récapitulatif

Ce document a présenté les caractéristiques de la plateforme Hadoop ainsi que trois solutions open source répondant à cette définition.

Son objectif n'était pas une comparaison poussée des trois solutions mais une première approche visant à permettre une réflexion sur les enjeux du Big Data et d'Hadoop en particulier.

Nombre de caractéristiques n'ont pas été abordées dans ce document. Toutefois il présente les forces et les faiblesses de chacun des outils et permet de savoir si la technologie est adaptée à votre besoin.

Chaque solution présentée est viable et même si elle répond 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.

10.3 En savoir plus

Site web Hadoop User Group : <http://hugfrance.fr/>

Site web <http://www.cloudera.com/content/cloudera/en/home.html>

Site web <http://www.mapr.com/>

Site web <http://hortonworks.com/>

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

GFS :

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/fr//archive/gfs-sosp2003.pdf

Map / Reduce :

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/mapreduce-osdi04.pdf

Big Table :

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/fr//archive/bigtable-osdi06.pdf

11 REMERCIEMENTS

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

Stéphane Lagraulet
slagraulet@ippon.fr

Vincent Beretti
vberetti@ippon.fr

Jérôme Mainaud
jmainaud@ippon.fr



Digital . Technologies . Hosting

ECOSYSTÈME HADOOP

Par **CHRISTOPHE PARAGEAUD**
MANAGER TECHNIQUE

NOV
2013



www.ippon.fr - <http://blog.ippon.fr>



#hadoopippon

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 :

<p>TOULOUSE 05 34 51 23 60 Immeuble Burolines II 2 ter, rue Marcel Doret 31700 Blagnac</p>	<p>NANTES 02 40 48 28 06 1 Rue Du Guesclin 44019 Nantes</p>
<p>PARIS 01 46 12 48 48 43-47 avenue de la Grande Armée 75116 PARIS</p>	<p>BORDEAUX 05 35 54 62 26 61 cours de l'Intendance 33000 Bordeaux</p>

Posez aussi vos questions sur Twitter à l'aide du hashtag : #hadoopippon



#hadoopippon



www.ippon.fr - <http://blog.ippon.fr>