Synthèse Big Data

Sacré Christopher

Table des matières

Introduction au Big Data

L'ère de l'information et du Big Data

De nos jours, on a de plus en plus de données, Il faut dès lors trouver un moyen de les traiter. De nouvelles méthodes apparaissent afin de nous y aider (Machine Learning, …).

En soit l'humain d'aujourd'hui est beaucoup plus social (dû aux réseaux sociaux) et plus mobile (Toujours connecté à l'aide de son smartphone notamment). De plus nos données sont stockées un peu n'importe où dû notamment à l'utilisation des clouds.

L'émergence des réseaux sociaux

On remarque qu'au fil du temps sont apparus divers réseaux sociaux et également l'ampleur que ceux-ci ont acquis dans notre quotidien. Leur évolution est sans limite et a pris une telle ampleur que de nombreuses marques (93 %), sociétés, … l'utilisent au niveau de leur marketing.

Social Business

Le social business au contraire de son nom n'a aucune vocation sociale. En soit le social business permet à une entreprise classique de continuer à fournir son service mais d'une nouvelle manière. On va tenter d'inclure les parties prenantes au sein du processus (clients, employés, fournisseurs, …). On tentera de créer une communauté autour du produit comme c'est par exemple le cas avec le crowdfunding.

On remarquera d'ailleurs qu'il existe deux types d'échange :

- Mono directionnel : l'entreprise produit, le client quant à lui achète.
- Multi directionnel : il s'agit de l'échange mono directionnel auquel on ajoute la présence du client, on va notamment lui demander son avis, des suggestions (On va faire en sorte qu'il ait ce qu'il aimerait avoir).

On va tenter de faire participer tout le monde dans l'entreprise (sondage de l'opinion). Avant on proposait juste un produit sans savoir si celui ci allait fonctionner.

Consom'acteur

Dans le temps, un consommateur était surtout passif. De nos jours les gens souhaitent être de plus en plus actif, ils achètent notamment en fonction de la valeur du produit et portant particulièrement importance à la réputation de la marque. En soit, ils agissent de plus en plus en tant que citoyen responsable qu'en tant que mouton.

Le consommateur peut de nos jours favoriser nos activités dans le cas où celles-ci soient en accord avec ces valeurs. Mais il peut aussi dans le cas où nos valeurs ne sont pas en adéquation avec les siennes, boycotter nos activités.

On remarquera plusieurs catégories de consommateurs :

- Influencé : ce consommateur ci va sur base de ce qu'il a pu lire ailleurs (sur internet, …) et sur base de l'avis d'autres gens aiguillé son choix d'achat.
- Influenceur : va donner son avis sur un produit et partagé celui ici avec d'autres personnes (Par exemple : dénoncer un produit non fonctionnel ou encore donner un avis positif). On remarquera deux types d'avis : les avis positifs et les avis négatifs. Mais il faudra dès lors faire attention aux faux avis potentiels ainsi qu'aux avis truqués (il faut également avoir conscience qu'il y a une plus grande tendance à avoir des avis négatifs).

Klout.com est une entreprise qui analyse les réseaux sociaux (ce que les gens publient, leurs avis) et donnent un score à chaque personne, il s'agira du score d'influence de la personne (La puissance de ces avis, la fréquence de ceux-ci). A la suite de la récolte de ces données, elle va ensuite les envoyer aux entreprises.

Analyse des réseaux sociaux

Il s'agit en soit d'une des disciplines du Big Data même si elle peut prendre énormément de formes différentes. Le but va être de tirer le maximum d'info (ou du moins les informations pertinentes), comme par exemple : les ressentis, les thèmes clés, les tendances, les sites et contributeurs influents, …

Cela va notamment permettre de définir de nouvelles stratégies de marques ou commerciales.

Le Mobile

L'importance du mobile dans nos quotidiens est sans contestes. Les mobiles sont même plus répandus que les téléphones fixes (dû à leur coût moindre à l'installation, de sa mobilité, de son côté pratique ou encore de son coût moindre en infrastructure).

On remarquera également son utilisation dans des opérations commerciales (M-Commerce, Paiements mobiles, …). Mais cela est très dur à quantifié. Le mobile impacte même les transactions en magasin. Cela ouvre donc forcément de nombreuses opportunités même si celle-ci ne sont pas forcément bonnes.

On remarquera notamment : la vente en ligne, SMS et mails business, confirmation de rendez vous, encodage d'informations au travail, scan de code QR.

Internet of things (IOT)

De nos jours l'internet of things commence à devenir de plus en plus important et est en expansion croissante. De nombreux objets de notre quotidien commencent à être connectés (Montres connectées, voiture connectée, frigo connecté, …).

Combiner

Il s'agit de l'utilisation de données venant de différentes sources afin de fournir des solutions pouvant intéresser. La combinaison d'informations est l'étape logique suivant la collecte de ces dernières.

On remarquera notamment les smart cities ou en bon français les villes connectées, dans lesquelles on a notamment installé des capteurs qui sont disponibles à tout le monde afin que ceux qui en voient une utilité puissent s'en servir.

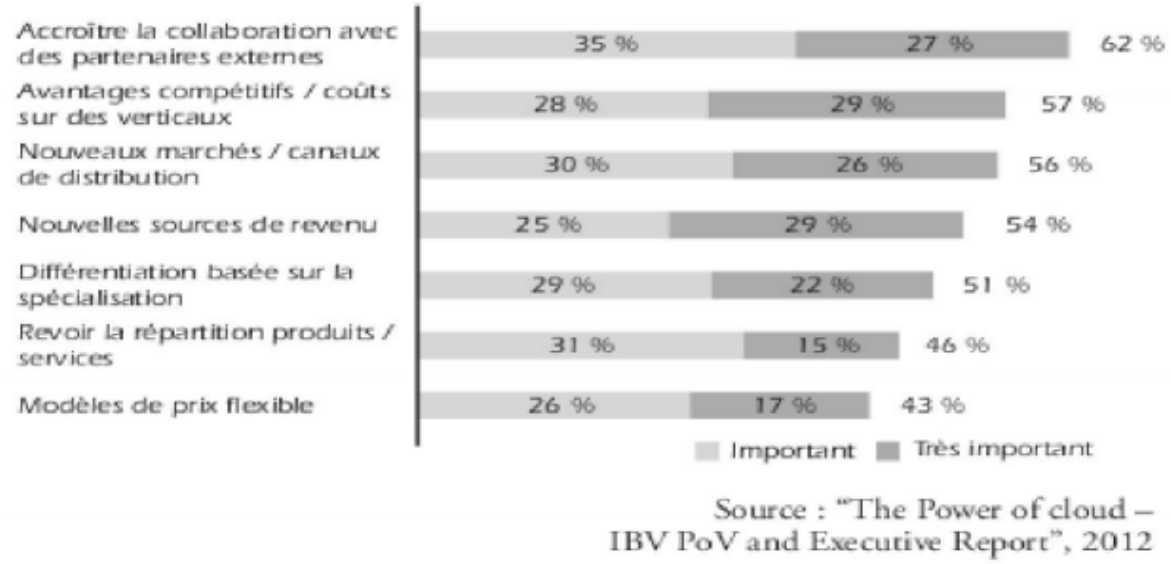
Cloud

Le cloud est de nos jours présents dans de nombreux aspects de notre vie, les systèmes de sécurité, nos balances, nos montres, nos messageries, voici de nombreux systèmes utilisant le cloud.

On distinguera d'ailleurs deux types de cloud :

- Les clouds publics : qui sont comme leur nom l'indique accessible à tous.
- Les clouds privés : les machines nous sont dès lors dédiées et nous appartiennent.

Changement de paradigme



Source : "The Power of cloud – IBV PoV and Executive Report", 2012

De nos jours la fluidité de l'information est une chose très importante. L'information doit pouvoir circuler facilement entre les différentes entreprises, leurs partenaires, leurs employés, leurs administrations, leurs clients, …

On va dès lors devoir utiliser le code pour fluidifier ce flux d'informations. Un point important est la sécurité et le sérieux du fournisseur de ce flux.

Contextualisation

En soit le cloud permet la collecte, le stockage et l'analyse des données. La combinaison du cloud et du Big data offre surtout des services contextuels et qui donc vont varier en fonction du contexte.

L'open data est le fait de mettre les données accessibles à tout le monde, de les rendre publiques. Dès le moment où nous sommes les personnes qui générons ces données, pourquoi certaines sociétés pourraient / devraient garder ces données privées. Le souci dans tout cela est que malgré l'anonymisation des

données, on pourrait dresser un profil très précis d'une personne en particulier ou même d'un échantillon de personnes en regroupant / combinant ces données.

Les 3V

Il s'agit du pilier du Big Data. En soit le Big Data n'est pas l'analyse de gros volume de données traditionnels. Le Big Data est la combinaison de ces données (volumineuses ou non) afin de décider d'une action.

Dû à cette augmentation du nombre de données, les bases de données traditionnelles doivent être revues car elle n'arrive pas à traiter cet amas de données. De nouveaux outils sont donc à notre disposition (Hadoop, …).

La business intelligence (BI) quant à elle effectue un travail de recherche de causalité. On va tenter à l'aide de la BI de trouver les causes de la situation actuelle.

Volume

Anciennement on interrogeait un échantillon de la population afin d'effectuer des statistiques (on procédait à un échantillonnage). Mais certains problèmes s'en dégageaient : la façon de poser la question pouvait influencer les réponses, on pouvait réfléchir avant de donner notre réponse et de plus il s'agissait des données dites mais non pensées.

Le Big Data quant à lui travaille sur l'entièreté de la population. On peut donc espérer que cela est plus fiable. De plus un mauvais choix de traitement peut être rattrapé en recommençant le traitement. Le plus gros avantage est qu'il travaille directement sur l'avis des gens (lorsqu'ils sont à chaud).

On va tenter un travail de prédiction plutôt qu'un travail de causalité comme le ferait la BI.

« Plus on a de données sur un phénomène et son environnement, mieux on l'appréhende. »

Vélocité

Afin que les résultats reflètent avec précision la situation actuelle il faut que les données utilisées soient les plus fraiches possibles et qu'elles soient donc toujours d'actualité.

Par exemple si nous prenons le real time biding, la vente de certains emplacements présent sur notre site : celui qui paie le plus cher pour cet emplacement de publicité aura la possibilité d'y afficher sa publicité. Mais pour cela il faut être rapide.

Les résultats doivent donc être le plus instantané possible et pour cela il vaut mieux traiter les données en temps réel mais surtout au bon moment et au bon endroit.

Comme c'est par exmple le cas avec le geo-fencing qui permettra d'afficher uniquement la bonne pub au bon moment (et donc une pub qui pourrait nous intéresser).

Variété

Afin de permettre divers types de résultats, tout deviendra donnée. Ainsi qu'il s'agisse de texte, de son, d'images on va tenter de récupérer cela afin d'y apposer un traitement.

On remarquera dès lors différents types de données :

- Structurées : il ne s'agit pas forcément de structure comme nous la voyons. Par exemple : un fichier xml possédant un fichier contraignant, …
- Semi-structurée : il s'agit du type de données le plus répandu. Par exemple : fichier xml (sans fichier contraignant, fichier de log, …)
- Non structurée : image, film, …

Un élément qui prend de l'ampleur dû à cette différence des types de données est l'intelligence artificielle, celle-ci va nous aider à traiter ces diverses données et donc de le faire plus rapidement et avec plus d'efficacité.

NoSQL

SGBD relationnelles

Même si les bases de données relationnelles offrent de nombreux avantages : l'utilisation de tables et de tuples pour structurer l'information, les liens par clés permettant de faire facilement des sélections et des jointures (à l'aide de SQL), elles sont basées sur un système à scalabilité verticale[1] et ne permettent donc malheureusement pas de faire de la scalabilité horizontale[2].

Un point également important au sujet des BSD est leur utilisation de la normalisation empêchanrt les redondances.

Pionniers du NoSQL

Dû à ce besoin de scalabilité horizontale et l'impossibilité de mettre cela en œuvre avec une base de données relationnelles classique est « né » le NoSQL qui permettait d'apporter une autre solution. Il faudra tout de même noter que NoSQL ne veut pas dire plus de SQL non, il s'agit plus de bases de données n'utilisant pas seulement du SQL (Not Only SQL).

Base de données orienté clefs – valeurs

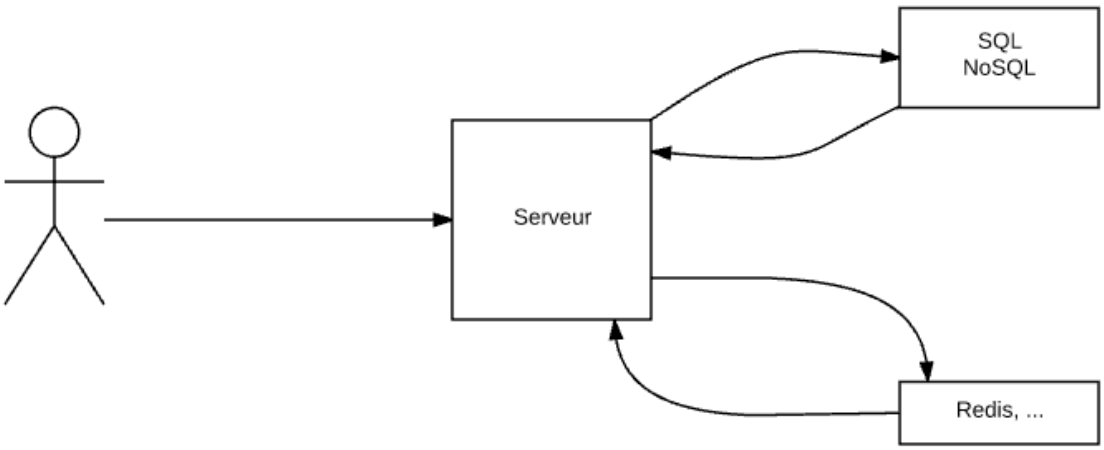
Il s'agit de bases de données tentant de stocker de petits formats de données (Chaines de caractères, tableaux associatifs, listes, ensembles, …). Celle-ci sera stockée en RAM afin d'être très rapide.

1 On augmente les capacités d'une machine en particulier (on est limité technologiquement)

2 On augmente le nombre de machines physiques mais virtuellement on a l'impression d'en avoir une seule.

Ce type de bases de données ne permet ni la scalabilité horizontale, ni de stocker de grandes quantités de données. De plus en cas de panne, une partie de nos données se verra perdue. En contrepartie, ce type de bases de données a de très bonnes performances, permet des opérations atomiques ainsi que des recherches en O(1) et est donc parfaitement adapté par exemple pour les sessions.

En fait les bases de données orienté clefs – valeurs (comme Redis) sont surtout utiles si combinés avec une autre base de données plus conséquentes. Ainsi celle orienté clefs – valeurs pourra contenir les informations les plus importantes demandant un traitement rapide tandis que l'autre se chargera de contenir l'entièreté des informations.



En conclusion, les bases de données orientées clefs – valeurs ont surtout permis d'apporter comme solution de meilleures performances comparer aux anciennes bases de données relationnelles un peu lente.

Un exemple de ce type de bases de données, est Redis, une base de données écrite en C possédant de très hautes performances et se basant sur l'utilisation de la RAM, de l'évènementiel et un principe de maitre – esclave (master -slave) dans lequel un maitre va rediriger les requêtes vers les esclaves (le maitre n'effectuera donc aucun traitement).

Bases de données orientées documents

On tentera de stocker au sein de notre base de données des collections, ou des documents. Mais un document pourrait très bien ne pas avoir de schéma et donc on pourrait avoir plusieurs types de documents différents dans une même collection. En soit on remarquera qu'il n'y a plus de type de tuples mais bien de valeurs. Ce type de bases de données est « Schema Less ». Chaque objet sera généralement un JSON (ou un BJSON, du Binary JSON).

On peut stocker ce que l'on veut, où on le veut, et quand on le veut (IDS automatique). Il faudra par contre faire attention à garder un peu d'autodiscipline.

L’avantage de ce type de bases de données est que l’on peut y stocker ce que l’on veut ajouter à cela le fait qu’elles utilisent du JSON (ou BJSON), il y a beaucoup moins de conversions, et donc beaucoup moins de pertes et également de meilleures performances. On parlera notamment de Full Stack JS (Notre application sera entièrement en JavaScript du Front end à la base de données) ainsi que de Sharding (découpe de l’application en plusieurs morceaux de données).

Les désavantages sont cependant cette absence de relations entre les documents et donc pas de jointures. Dû à cela une application demandant de nombreuses relations entre les données ne sera pas adaptée avec ce type de base de données. En soit une base de données orienté documents peut s’apparenter à une dénormalisation (Et il nous faudra dès lors reporter chacune des modifications dans le cadre d’une occurrence).

Exemple de telles bases de données : MongoDb, Couchdb

Bases de données orientées colonnes

Cela pourrait ± s’apparenter à une table classique possédant des colonnes dynamiques. La plupart des requêtes pouvant s’appliquer sur ce type de bases de données seront des requêtes simplistes et nécessiteront un index.

Les colonnes ne seront pas forcément gardées sur le disque (on ne gardera que les données que l’on a besoin). Certaines cellules pourront être vides, sauf qu’elles ne seront pas dans le tuples et donc ne réserveront pas cet espace en mémoire. On tentera donc d’avoir un gain de place. Ce type de bases de données est prévu (à la base) pour une scalabilité horizontale.

Bases de données orientées graphes

Ce type de bases de données sera surtout constituées de nœuds et d’arcs afin de former un graphe. Ce type de bases de données est surtout pratique lors de la modélisation des réseaux sociaux. L’utilisation de l’algorithme de Dijkstra au sein de ce type de bases de données est donc relativement importante.

Exemple : Neo4j, Orient DB.

Conclusion

Chaque type de base de données à sa propre utilisation. Il faudra dès lors se demander quels sont les liens utiles ?

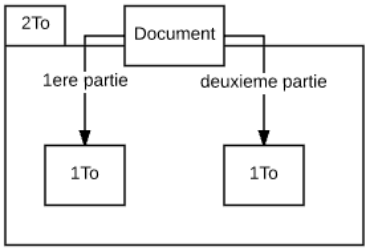
Le système NoSQL ne permettant plus de transactions (triggers) ni de requêtes compliquées (jointures, GROUP BY, …) et ne permettant pas la normalisation, les redondances seront donc permises et fortements utilisées.

Afin de nous aider dans notre choix on effectuera celui ci à l’envers. On choisira le type de base de données après avoir étudié les données (Même si cela voudrait dire dand certaines circonstances de devoir le faire en cours de route).

Hadoop

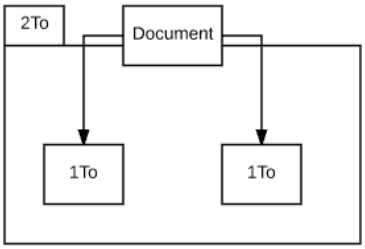
Avant de nous attaquer regardons ensemble les différentes manières dont nous pourrions améliorer nos systèmes de stockages actuels :

1. RAID 0



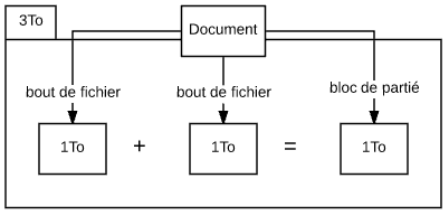
On va effectuer diviser le document en deux afin que de diviser la tâche en deux. Cela va permettre de meilleures performances en écriture / lecture mais en contrepartie si l'un des disques dur lâche, on perdra automatiquement les données de l'autre car elles deviendront illisibles.

2. RAID 1



Il s'agit en soit du « RAID sécurité » on va écrire un document en double. On va donc avoir une réplication parfaite des deux disques durs. En soit il s'agit d'un système de backup automatisé.

3. RAID 5



On va sauvegarder une partie d'un document au sein d'un disque et l'autre partie au sein d'un autre mais on va également garder un bloc de parité qui permettra de savoir dans le cas où l'un des disques lâche, le contenu du disque défectueux.

En soit on remarquera deux types de RAID : les RAID orienté matériel qui sont plus rapide mais ne connaissent pas le contenu des disques et les raids logiciels qui sont plus flexibles et qui travaille avec l'OS et au niveau des partitions. Le choix de tel ou tel type de RAID dépend des cas d'utilisation.

Introduction

Dû à la production de plus en plus massive de données et à cette combinaison des 3V du Big Data les systèmes actuellement mis en place ne permettaient pas une gestion efficace des fichiers.

C'est donc pour cela que voient le jour des systèmes distribués comme le HDFS (Hadoop Distributed File System) ainsi que le mapReduce. Hadoop permet notamment d'utiliser un environnement distribué (on pourra donc séparer le travail sur différentes machines) et de traiter de grandes quantités de données.

Dû à sa nature il faudra faire attention Hadoop n'a de l'intérêt que dans un système de grande taille et n'est donc pas du tout adapté pour les petits fichiers.

HDFS

Il s'agit d'un système de fichiers (un fichier étant un ensemble de blocs), reliant le nom du fichier à une liste de blocs. Ce système de fichiers possédera

également des permissions ainsi que des répertoires comme un système de fichiers classique (Unix).

Les grandes différences de l'HDFS est qu'il n'est pas lié au noyau et donc est portable. En soit il s'agit d'une application Java (et donc est « virtuel ») et pourra créer un nouveau système de fichiers sur n'importe quelle machine, il s'agit donc d'une application externe de montage.

De plus celui-ci est distribué (et ne possède donc pas de limite de taille) et on pourra donc stocker les données sur plusieurs serveurs. A cela vient s'ajouter sa taille de blocs plus élevées que sur un système de fichiers « classique » celle-ci dépassera les 64Mo on n'y stockera donc que des gros fichiers. A l'aide de cela on gaspillera moins de place.

HDFS permettra également la réplication des blocs. Dû au fait que chaque bloc est distribué, ceux-ci peuvent donc se retrouver sur n'importe lequel des serveurs et pourraient donc être répliqués sur des serveurs différents. (Le facteur par défaut de réplication au sein de HDFS est 3).

HDFS – Namenode

Il s'agit du service central (en soit du maître). C'est lui qui possède la connaissance de l'état du système de fichiers (Tant les métas données que les répertoires ou les droits). Il possède également la connaissance des datanodes de notre système. Il connait donc l'entièreté du système.

Il a également un rôle de load balancing[3]. Il assurera le rôle de chef d'orchestre et devra répartir équitablement les tâches.

Lors du démarrage d'un namenode celui-ci chargera la position des blocs (Safe-Mode). Ceux-ci ne seront d'ailleurs accessibles qu'en lecture seule. De plus ce procédé se passera uniquement en mémoire ce qui est donc très couteux (150 bytes par fichiers).

Afin de pouvoir fonctionner, il va utiliser deux fichiers :

-fsimage_xxx : il s'agit de l'image du système de fichier (file system image) : celle ci permet de connaître la position de tous les blocs.

- edits_xxx : ce fichier permettra de répertorier toutes les modifications du file system (droits, emplacement et tailles des fichiers).

=> Ces informations ci seront chargées en mémoire (Il faudra aller vite pour garder les performances, et dès lors cela nécéssitera beaucoup de RAM).

Le problème avec cela est le fait que le namenode devient dès lors un « single point of failure ». Une solution à cela est l'utilisation d'un deuxième namenode. Pour cela ce namenode devra être un réplica du premier et devra donc vérifier constamment l'état du premier. De plus il ne sera pas toujours facile pour le deuxième namenode de savoir quand prendre le relais.

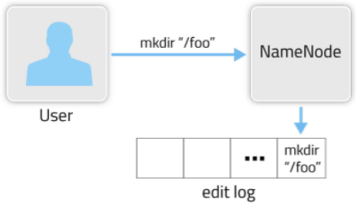
3 Permet de répartir de manière homogène les requêtes aux serveurs

HDFS – Datanode

Il s'agit en soit de ceux qui travaillent. C'est lui qui possédera les blocs de données à proprement parler. De plus ces derniers peuvent dialoguer entre eux et ne doivent donc pas forcément passer par le namenode (Cela permet au namenode de travailler le moins possible). Les datanodes se connectent au namenode dès le démarrage.

HDFS – Checkpointing

Dès que l'on fait une modification, celle-ci va se mettre dans une nouvelle photo de notre système de fichiers[4] (edits). On va ensuite remettre les deux ensembles (Le fichier log reprenant les modifications et l'ancienne photo de notre système de fichiers) afin de recréer cette image de notre système de fichiers.

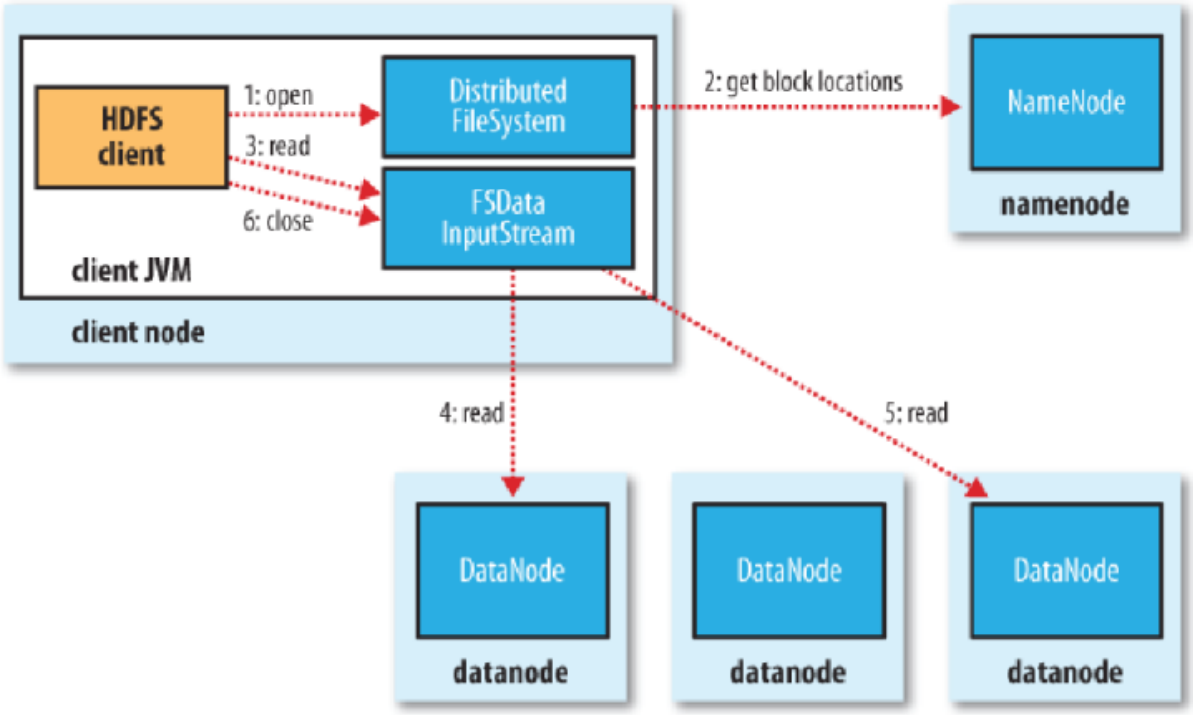


Il faudra dès lors faire attention aux photos du système de fichiers présents en mémoire (qui sont à jour) et ceux présent sur le disque (qui eux ne le sont pas). Il s'agit donc en soit d'une manière de mettre à jour notre système de fichiers.

On procédera à cette mise à jour lors du démarrage du sytème ou quand on n(utilise pas Hadoop ou via le secondary Name Node (notre name node de backup).

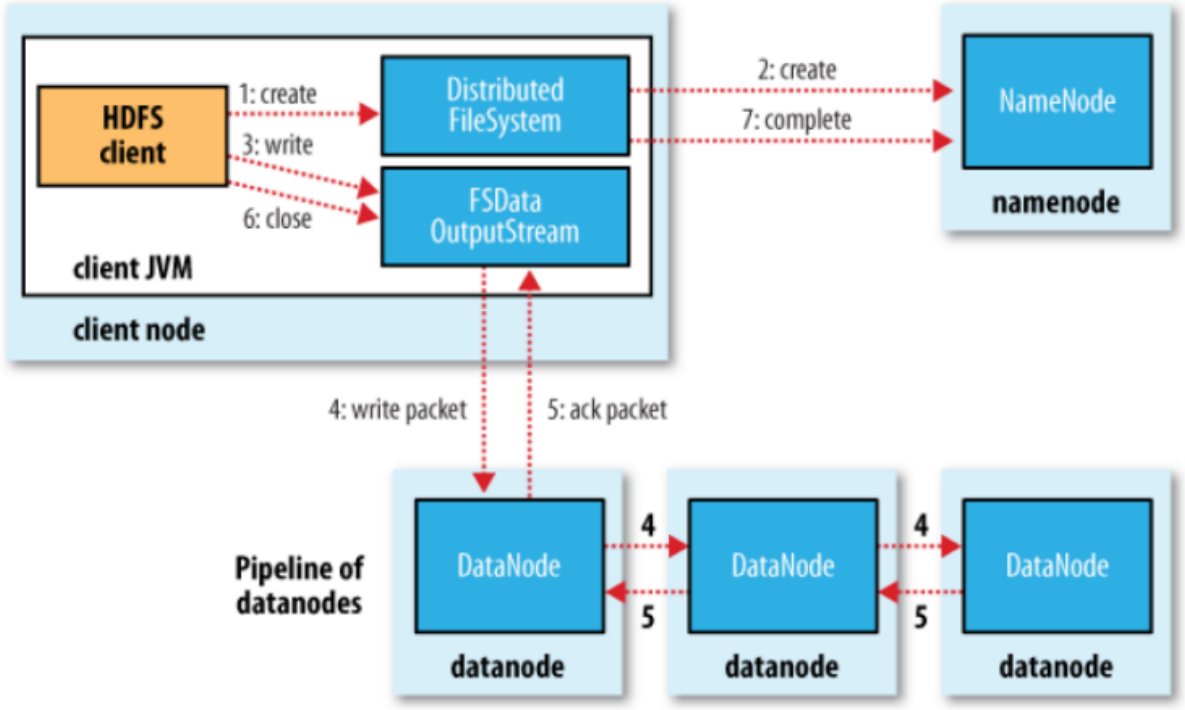
Petit apparté, si le démarrage en safe mode est aussi lent c'est également à caue de la mise à jour nécéssaire du fichier fsimage qui prend du temps.

4 Entièreté des informations du système de fichiers




```
FileSystem fileSystem = FileSystem.get(conf);
Path path = new Path("/path/to/file.ext");
if (!fileSystem.exists(path)) {
   System.out.println("File does not exists");
   return;
}
FSDataInputStream in = fileSystem.open(path);
int numBytes = 0;
while ((numBytes = in.read(b)) > 0) {
   System.out.prinln((char)numBytes));// code to manipulate
the data which is read}
   in.close();
   out.close();
   fileSystem.close();
}
```

HDFS – Écriture




```
FileSystem fileSystem = FileSystem.get(conf);
// Check if the file already exists
Path path = new Path("/path/to/file.ext");
if (fileSystem.exists(path)) {
    System.out.println("File " + dest + " already exists");
    return;
}
// Create a new file and write data to it.
FSDataOutputStream out = fileSystem.create(path);
InputStream in = new BufferedInputStream(new FileInputStream(new File(source)));

byte[] b = new byte[1024];
int numBytes = 0;
while ((numBytes = in.read(b)) > 0) {
    out.write(b, 0, numBytes);
}
// Close all the file descripters
in.close();
out.close();
fileSystem.close();
```

MapReduce

Il s'agit d'une manière de programmer, d'un Framework d'implémentation. On tentera de programmer le plus proche possible des données (on aménera le calcul aux données). Le calcul sera ainsi effectué sur chaque machine (où sont stockées les données) et seront renvoyées au demandeur. Ce dernier s'occupera de regrouper les résultats de chacuns des « esclaves ». À l'aide de cela on va séparer la masse de travail en parallélissant celui-ci et en le distribuant.

Pour cela on va utiliser des « jobs » qui vont prendre des données en entrées, un programme mapReduce ainsi que des paramètres d'exécution (Il pourra s'agir

par exemple du temps avant le démarrage, du nombre de nœuds sur lesquels faire mes calculs, ...).

Au sein de Hadoop ceux-ci seront divisés en deux tâches : une pour map et une pour reduce.

MapReduce – Map

On va effectuer une map avec les données reçues en entrée, ou plutôt on va créer une liste de clefs – valeurs. En d'autres mots notre valeur sera notre liste de données.

Exemple :

On prend en input deux phrases :

- Hello World, Bye World
- Hello Hadoop, GoodBye Hadoop

Et on recevra en sortie :

<Hello 1>, <World 1>, <Bye 1>, <World 1>

<Hello 1>, <Hadoop 1>, <GoodBye 1>, <Hadoop 1>

```
public static class Map extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter reporter) throws IOException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            output.collect(word, one);
        }
    }
}
```

MapReduce – Combine

On va combiner les clefs identiques afin de supprimer les doublons.

Exemple :

On aura en input une liste constituée de :

<Hello 1>, <World 1>, <Bye 1>, <World 1>

<Hello 1>, <Hadoop 1>, <GoodBye 1>, <Hadoop 1>

Et on recevra en sortie :

< Bye 1>, < Hello 1>, < World 2>

< Goodbye 1>, < Hadoop 2>, < Hello, 1>

MapReduce – Reduce

On va récupérer une liste clefs – valeurs et n'en faire qu'une liste de valeurs. Il va permettre également de réduire le nombre de listes disponibles afin de n'en former qu'une.

Exemple on prend en input :

< Bye, 1> < Hello, 1> < World, 2>

< Goodbye, 1> < Hadoop, 2> < Hello, 1>

Et on recevra en output une seule liste contenant tous les éléments distincts :

< Bye, 1> < Goodbye, 1> < Hadoop, 2> < Hello, 2> < World, 2>

```
public static class Reduce extends MapReduceBase implements Reducer<Text,
IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text,
IntWritable> output, Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

MapReduce – Hadoop

On va notamment avoir trois parties importantes :

- Split

  On va séparer les différentes données au sein des différents nœuds.

- Map

  On est le plus proche possible de la donnée. Le stockage se fera par ailleurs sur le nœud.

- Reduce

  On va tout récupérer sur un seul nœud, il s'agira du nœud central, c'est lui qui s'occupera d'assembler le tout et qui récupérera les données calculées.

Les outils du Big Data

BI vs Big Data

La BI est avant tout l'étude de passé afin de prendre des décisions futures à l'instar du Big Data qui va essayer de prédire une tendance, une évolution. En soit la BI et le Big Data cohabitent ensemble.

Avec la BI va prendre ce que l'on a et on fait de notre mieux, avec le big data on va prendre ce que nous avons mais nous allons également le croiser avec des données provenant d'autres sources.

Data Warehouse

Les Data Warehouse ont été créé sur base d'une informatique décisionnelle. Les données vont ainsi être rassemblées afin de permettre leur collecte (pouvant provenir de sources différentes), leur nettoyage ainsi que leur normalisation. On tentera pour cela de garder un historique des données sauvegardées.

Dû à une collecte provenant de différentes sources cela demande de s'adapter à chaque source. En effet chaque source possède ses propres standards et normes et le format des données peut donc varier (différences d'encodages, …). De plus, le data warehouse va traiter les données en amont afin que celle-ci soient utilisables. En sot on peut dire qu'un data warehouse sera une base de données qui va regrouper toutes les autres.

Un souci avec de telles infrastructures est sa gestion et la difficulté de ses mises à jour.

Le but principal du data warehouse est d'offrir la possibilité de faire de la business intelligence (BI). Ils vont également permettre de faire du data mining (càd l'extraction de connaissances à partir de données).

Cubes multi dimensionnels

On va tenter de modéliser nos données sous forme d'un cube. Afin de parvenir à cela on va prendre plusieurs critères et on va les placer sur les différents axes dans le but d'obtenir des rapports de synthèses en temps réel.

Hadoop

Comme vu précédemment Hadoop est particulièrement adaptés pour traiter la quantité importante de données que demande le Big Data. Et il va notamment permettre le stockage de ces données (HDFS) et d'effectuer des requêtes sur celles-ci (mapReduce).

Les outils de prédictions

On va utiliser un modèle se basant sur l'historique. En fait en fonction des succès / échecs de notre prédiction on tentera de s'améliorer (si succès c'est que l'on est sur la bonne voie, si échec c'est qu'il fallait changer quelque chose au sein de notre processus) comme c'est par exemple le cas actuellement avec le machine learning.

De tels outils ouvrirait de nouvelles portes et permettraient en outre de détecter les fraudes, d'aider au diagnostic médical, de repérer des tendances sur le marché financier, … Google a par exemple réussis à prédire une épidémie de grippe plus rapidement que le système conventionnel qu'utilisent les hôpitaux et cela en analysant les requêtes de ces utilisateurs.

De plus ces outils combinent différentes sources de données afin de produire quelque chose pouvant être utilisé par exemple pour un GPS (système de guidage) performant nous allons combiner le trafic routier en temps réel, la météo ainsi que l'historique du conducteur afin de choisir l'itinéraire le plus approprié.

L'analyse en temps réels

De nombreuses applications demandent un traitement des données en temps réel. Pour cela nous allons devoir analyse le flux actuel de ces données (Stream analysis). Exemple de données demandant d'être traitées en temps réels : données GPS, réseaux sociaux, données télécoms, …

Les outils de visualisation

À l'aide de ces outils nous allons tenter de ne plus faire une approche causaliste, mais de détecter les relations entre les données. Pour cela nous allons notamment utiliser le data mining, des graphes ou encore des heatmaps (Ceux ci deviennent néanmoins obsolétes avec le Big Data).

On ne tentera pas de rechercher la causalité mais les relations entre les différents éléments.

Comment implémenter le Big Data dans l'entreprise

Les modèles de mise en place

A l'aide d'une mise en place du Big Data, on pourrait avoir des coûts plus faibles, un stockage plus important ou encore des traitements plus rapides (En grande partie à l'aide de la scalabilité horizontale).

Mais un premier choix s'impose et il nous faudra choisir le modèle à mettre en place / ce que l'on désire conserver, il faut se demander si l'on souhaite mettre plus l'accent sur la BI et les data warehouse ou sur le Big Data en tant que tel.

Modèle disruptif

Dans ce modèle ci, le Big Data sera l'élément principal du domaine décisionnel. On va dès lors mettre en place un système Hadoop, un mapReduce du machine learning ou encore de la visualisation. On va donc mettre l'accent sur une approche réactive plutôt qu'une étude à postériori.

Dans un tel type de modèle les données pourraient très bien être structurées ou non.

Modèle évolutif

Dans ce type de modèle le traitement des données se fait en amont du data warehouse. On va donc tenter de conserver l'infrastructure déjà mise en place. En fait on ne conservera qu'un nombre limité de données (les données qui nous intéressent uniquement).

Cela permettra donc un enrichissement de nos données déjà présentes, permettra une meilleure protection du data warehouse (les données étant pré traitées) et permettra d'éviter les engorgements .

Pour cela nous pouvons utiliser une société tierce afin qu'elle procède elle-même au prétraitement des données que nous souhaitons ajouter à notre data warehouse. Un exemple de telle société est The Now Factory qui va collecter des données sur les gens et qui va effectuer le traitement en amont de notre architecture conventionnelle avant de revendre ces données aux entreprises intéressées par les données.

Modèle Hybride

Il s'agit d'un modèle permettant de faire un flux vers le data warehouse. En soit on va ajouter un flux de données vers le Big Data et donc tenter une intégration à la fois du Big Data et de la BI. Tout cela e tentant de conserver le modèle décisionnel déjà existant.

Ce modèle propose peu de lien entre le Big data et la BI mais permettra aux deux de cohabiter ensemble. Le Data Warehouse recevra les anciens flux, tandis que le big data s'occupera des nouveaux flux.

Il faudra cependant faire attention aux sources différentes synonymes d'incohérences.

Cela permet un bon compromis entre les deux modèles précédents car les entreprises préfèrent ne pas détruire ce qui est déjà en place. On préfèrera donc avoir un système Big Data à coté de l'infrastructure traditionnelle et mettre en place la communication entre les deux.

Compétences nécessaires

En soit trois ressources interviendront directement dans ce nouveau type de processus :

- Les données qu'elles soient internes (l'entreprise génère ces données, les produits, les possède) ou externes à l'entreprise.
- Une extraction de la valeur
- Une méthode de gestion des idées créatives pour la transformation en produits / services.

Gestion des données

En soit pour permettre une gestion de ce nouveau flux de données, il va nous falloir revoir notre département informatique et y ajouter de nouveaux modèles de traitement et de stockage (Ajout d'un système Hadoop, utilisation du NoSQL).

Il faudra dès lors faire attention à l'intégration de ce nouveau système de sa cohérence avec le système déjà existant ainsi qu'au budget qui va permettre ces modifications.

L’extraction de la valeur

Dû à ce besoin du Big Data on verra l’émergence d’un nouveau métier : les Data Scientist. Ceux-ci auront pour but de tirer quelque chose d’une masse de données. Ils auront pour cela besoin d’un profil de statisticien et de mathématicien.

Il devra cependant voir large dans l’entreprise, avoir de bonnes connaissances générales globales et tenir compte de toutes les contraintes existantes. Il devra également veiller à la maitrise des coûts et faire attention que ceux-ci ne dépassent pas un certain seuil.

Management

En soit cela permettra de mettre un état d’esprit Big Data au sein de notre entreprise. Actuellement l’état d’esprit autour de ce concept véhicule une image positive et dynamique et encourage à créer de la richesse tout cela en étant plus proche du client et à l’écoute de celui-ci.

Actuellement c’est également une branche qui manque de diplômés.

Changements organisationnels

On remarque une modification pour l’expert métier. Avant celui-ci devait se baser sur des suppositions, sur son vécu ou sur son expérience. Mais à présent, on remet cela en question et on tente d’évoluer. Le Bug Data va par exemple permettre de prédire des tendances et va diriger le métier de l’expert métier. Désormais le boulot de l’expert métier va se baser sur des données concrètes et objectives afin de prendre des décisions.

Centre d’expertise

En soit certains éléments sont nécessaires afin de permettre une bonne mise en place du Big Data au sein d’une société. Tout d’abord c’est l’assistance au démarrage ainsi que des formations afin de permettre une bonne prise en main de cet outil.

Ensuite on va créer une culture autour du Big Data et donc étoffer les données récoltées.

Un autre point crucial est le support aux utilisateurs afin que ceux-ci ne se sentent pas délaisser et puissent mieux comprendre le fonctionnement de ce nouvel outil.

Externalisation

Un autre point pouvant être intéressant est l’externalisation de nos données et donc de les fournir à d’autres sociétés (en échange de leurs données ou non). Le partage et la combinaison de données provenant de différentes sources créent également de la valeur.

Prenons l’exemple des assurances qui s’échangent leurs informations concernant leurs clients afin de mieux contrer la fraude.

Démarrage

Afin de permettre un bon démarrage du Big Data au sein d'une entreprise il faudra commencer par des sessions de travail permettant une prise en main de ce nouvel outil. Il faudra également pouvoir expliquer pourquoi cette mise en place du Big Data

- Pourquoi plus de données ? => pour permettre un résultat plus précis, plus ciblé

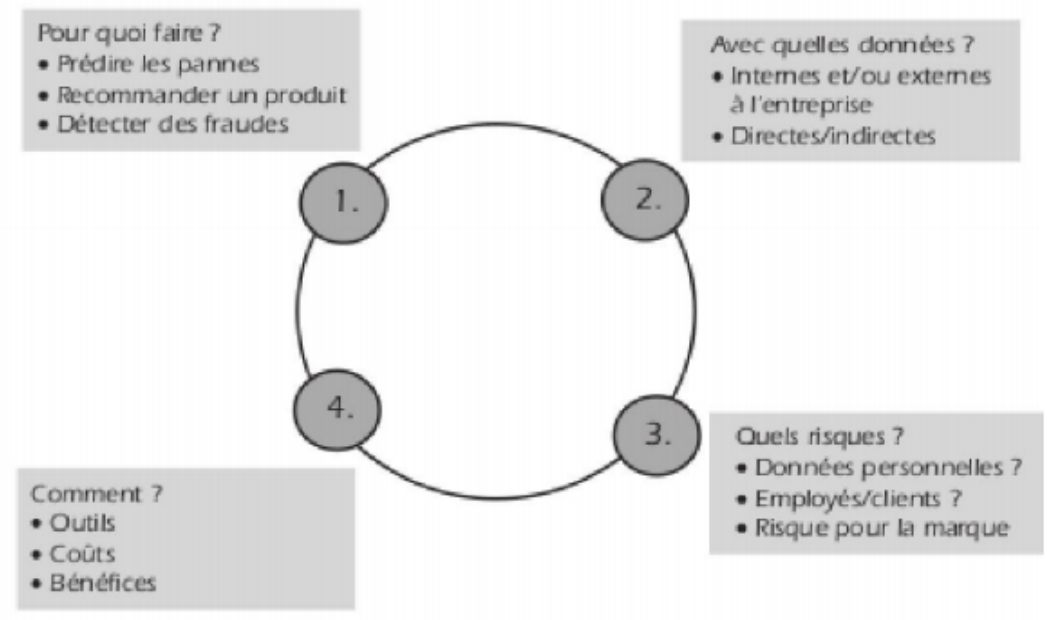
- Pourquoi des corrélations entre celles-ci ? => Pour prédire les évenements à venir

- Pourquoi du traitement en temps réel ? => Pour pouvoir réagir et s'adapter au plus vite.

Il faudra également permettre une compréhension des sources d'alimentions, et donc de données. Expliquer pourquoi utiliser à la fois des données internes et externes mais également d'expliquer le lien direct (donnée tirée de la source initiale) / indirect (donnée ne provenant pas de la source initiale) avec la source.

Il faudra également comprendre les risques relatifs à cette mise en place. Tant au niveau du personnel (La mise en place du Big Data au sein d'une société pourrait changer la structure organisationnelle). Mais également revoir la sécurité des données et comprendre que la réputation de l'entreprise pourrait vite ne pâtir.

En plus de cela afin de donner envie à la société de se lancer dans cette expérience il faudra leur donner une idée des couts qu'une telle mise en place va demander ainsi que les bénéfices que celle-ci va permettre d'apporter. Ce qui n'est pas toujours facile étant la nature même du Big Data. Il faudra donc dès lors fournir des exemples concrets mais également des retours d'expériences. On pourrait également leur proposer des outils ou une externalisation permettant de faciliter cette mise en place.



Déploiement – Intégration

Pour ce qui est du déploiement ou de l'intégration il faudra faire cela de manière itérative (procéder étape par étape) et donc développer cette culture du Big Data. Il faudra également veiller à la cohérence et l'intégration de notre solution Big Data avec les systèmes déjà existants et donc au processus global.

Intégration

Il faudra trouver un équilibre entre les canaux de ventes (avantage en ligne qui portera préjudice au magasin qui se verraient défavorisé) et limiter l'effet d'aubaine. On veut booster nos ventes et non pas faire voyager nos clients d'un canal de vente à un autre.