MASTER MIAGE 2ème année
Université Paris Nanterre

Mémoire de fin d'études

Méthodes pour bien choisir son architecture Big Data en fonction de son processus métier

Auteur :
Ludwig SIMON

Tuteur :
Mcf. Emmanuel HYON

Février 2019 — Juin 2019

Remerciements

Remerciements

Résumé

Résumé

Motivations

Le Big Data est un domaine très vaste, il y a une multitude d'outils pour effectuer les "mêmes" tâches. Il est donc très difficile de faire le bon choix lorsqu'on se lance dans ce domaine.

Objectifs

Dans ce mémoire, nous allons dans un premier temps rappeler brièvement ce qu'est le Big Data et quand il est vraiment utile de l'utiliser. Dans un second temps nous allons devoir établir des critères permettant d'évaluer quelle catégorie d'outil convient le mieux et par la suite quel outil conviendra le mieux. Pour le choix de l'outil on utilisera en plus de critères de cas d'utilisation des benchmarks afin de récupérer des informations sur la consommation de ressources et de temps pour chaque outil. Ensuite, à l'aide de ces critères définit précédemment, nous allons définir des méthodes d'analyse afin de sélectionner les outils adéquat. Et pour finir nous allons appliquer nos méthodes sur un cas concret afin de vérifier l'efficacité de notre solution.

Sommaire

Introduction

Le Big Data

Chapitre 1

Comment définir les critères de choix

Afin de définir des critères pour choisir l'architecture la plus adaptée pour notre cas d'utilisation, il va falloir identifier les différentes architectures existantes pour répondre à un besoin dans le domaine du Big Data.

1.1 Architecture Réactive

1.2 Architecture Répartie

1.3 Traitement des données

Après avoir récupérer des données, nous devons passer à l'étape du traitements des données. Celui à plusieurs rôles, en effet il peut servir à formater les données, leurs apportés de la cohérence en les combinant à des données déjà présentent. Et pour finir les rediriger vers le stockage souhaité. Le traitement des données peut se faire de deux manière différentes. La première solution est le traitement par Batch, et la seconde est le traitement en temps réel [1]. Chacune possède ses avantages et inconvénients, nous allons voir ça plus en détails.

1.3.1 Batch

Le traitement par Batch, consiste à traiter un important volume de données à un instant T. Le traitement par batch est surtout utilisé dans les cas ou nous avons des données stockés de manière journalière, et que nous avons besoin de tout traités en fin de journée. Il n'est pas rare de voir des tâches de traitements par Batch s'exécuter dans la nuit, étant donnée que l'on traite une masse de donnée importante, on sollicite la machine pendant une longue période. Réaliser ce traitement durant des périodes creuses permet de largement diminuer l'impact sur l'utilisation de la plateforme.

1.3.2 Temps Réel

Un traitement de données est considéré comme étant en temps réel si il s'effectue en une seconde ou moins après la réception de la donnée. Il peut être de deux types, soit des micro batchs soit en streaming.

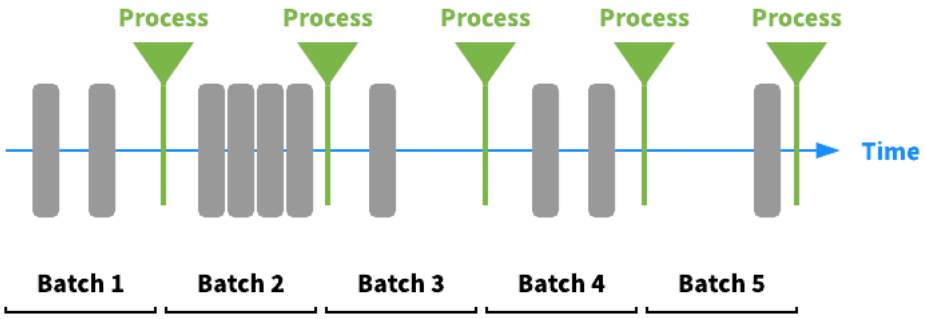

Figure 1.1 – Schéma du traitement par batch

1.3.2.1 Micro-Batch

Le traitement par micro batch est basé sur le même principe que le traitement par batch, à l'exception qu'il s'exécute beaucoup plus régulièrement (Toutes les secondes ou moins) et que le nombre de données à traiter est donc significativement plus faible. Le micro batch est surtout utilisé dans les cas où notre système ne peut pas directement réagir lorsqu'une donnée arrive, on va donc récupérer les données très régulièrement afin de garantir un traitement en temps réel ou du moins dans le délai le plus bref possible.

1.3.2.2 Streaming

Le traitement en streaming s'appuie sur l'architecture réactive. En effet, contrairement aux traitements par batch et micro batch, ici on ne vas pas récupérer des données de temps en temps. Dès qu'une donnée arrive on va la récupérer et la traiter immédiatement. De par son fonctionnement le traitement en streaming ne nécessite pas de stockage en amont contrairement aux autres type de traitement.

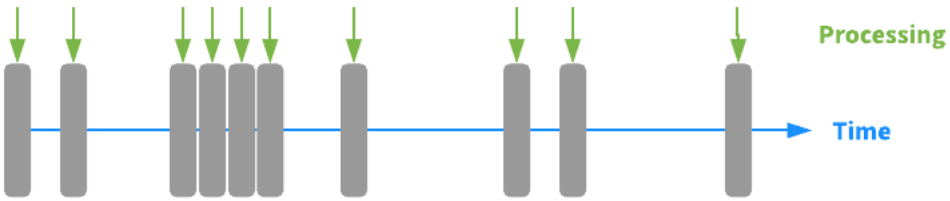

Figure 1.2 – Schéma du traitement en streaming

De manière générale, on peut pas forcément se permettre de n'utiliser que l'un de ces deux type de traitement de données, nous avons régulièrement besoin de les utiliser en même temps. Des architectures ont justement été pensées pour cela [2].

1.3.3 Architecture Lambda

L'architecture Lambda a été crée dans le but de pouvoir à la fois traiter des données en batch et en streaming. Plus précisément, l'architecture lambda est composé de deux couches afin de gérer à la fois les batch et le streaming. Une couche est dédié pour traiter les données par batch, pour ensuite les rendre disponible (Ce qui est appelé une "view"). L'autre couche, s'occupe du streaming. Et à chaque donnée traitée, le résultat est mis à disposition. Grâce à ce fonctionnement lorsque l'on souhaite faire une requête, elle va pouvoir prendre en compte nos données récupérées par batch et par streaming.

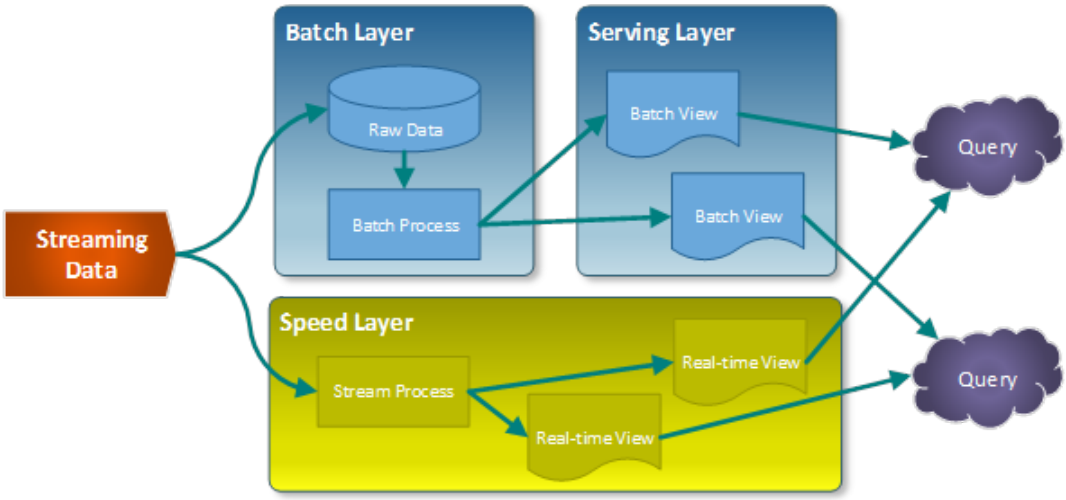

Figure 1.3 – Schéma de l'architecture Lambda

1.3.4 Architecture Kappa

L'architecture Kappa, se veut plus "simple". Le but étant de traiter toutes les données en streaming. Cela a pour conséquence de n'avoir qu'une seule couche contrairement aux deux nécessaire pour l'architecture Lambda. Mais cela ne permet pas de remplacer l'architecture Lambda, en effet il faut que les données que l'on récupère en Batch et la manière dont elles sont stockées, permettent de les traiter en streaming par la suite.

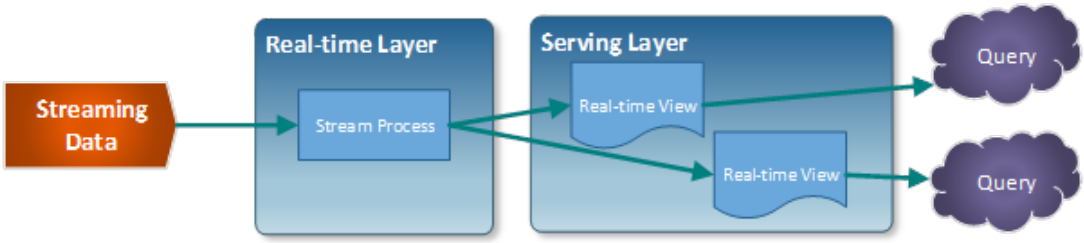

Figure 1.4 – Schéma de l'architecture Kappa

1.4 Stockage des données

Chapitre 2

Analyse des solutions logicielles existantes

Une fois nos critères définit, il faut décomposer le Big Data qui est un domaine très vaste, en plusieurs catégories [3]. Une fois cette décomposition effectué, on va pouvoir s'intéresser aux outils existant permettant de réaliser chacune des tâches. On va noter leurs points faibles et leurs point forts ainsi que la manière dont ils sont censés être utilisés.

2.1 Ingestion/Extraction de données

La première catégorie, qui est aussi la première étape d'une architecture Big Data, c'est le récupération de données. Plus précisément comment nous allons récupérer des données, soit via des requêtes sur des sources externes, soit des sources externes nous envoie directement des données.

2.1.1 Avec système de messaging

2.1.1.1 Kafka

2.1.1.2 ActiveMQ

2.1.1.3 RabbitMQ

2.1.2 Sans système de messaging

2.2 Traitement des données

Une fois les données reçu, des traitements sont nécessaire afin de pouvoir stocker les données au format souhaité ou bien pour faire un tri des données utiles.

2.2.1 Visuel

2.2.1.1 Apache Nifi

2.2.1.2 Talend

2.2.2 Non visuel

2.2.2.1 Streaming

Spark Streaming

2.2.2.2 Micro Batch

Outil

2.2.2.3 Batch

Spark

MapReduce

2.3 Stockage des données

Une partie très importante du Big Data est le stockage des nombreuses données que l'ont reçoit. Il existe énormément de manières différentes de stocker des données selon la manière dont nous voulons les utiliser par la suite.

2.3.1 Time Series

2.3.1.1 OpenTSDB

2.3.1.2 InfluxDB

2.3.2 Graph

2.3.2.1 Neo4j

2.3.2.2 JanusGraph

2.3.3 Données Structurées

2.3.3.1 Hive

2.3.3.2 Phoenix Framework

2.3.4 Clé-Valeur

2.3.4.1 HBase

2.3.4.2 Redis

2.3.5 Index

2.3.5.1 ElasticSearch

2.3.5.2 Apache Solr

2.4 Requêtage

2.4.1 Visuel

2.4.1.1 Kibana

2.4.1.2 Banana

2.4.1.3 Grafana

2.4.1.4 Tableau

2.4.2 Non visuel

2.4.2.1 OLAP

Outil

2.4.2.2 OLTP

Chapitre 3

Méthodes d'analyse pour choisir la bonne architecture

3.1 Choisir le type d'outil

3.2 Choisir l'outil

Chapitre 4

Implémentation : Exemple avec un processus métier

4.1 Définition du processus métier

4.2 Application des méthodes sur le processus métier

4.3 Évaluation du résultat

Bibliographie

[1] Laura Shiff. *Real Time vs Batch Processing vs Stream Processing : What's The Difference ?* url : https://www.bmc.com/blogs/batch-processing-stream-processing-real-time/.

[2] Laura Shiff. *From Lambda to Kappa : A Guide on Real-time Big Data Architectures*. url : https://www.talend.com/blog/2017/08/28/lambda-kappa-real-time-big-data-architectures/.

[3] Christophe Parageaud. *Big Data, panorama des solutions*. url : https://blog.ippon.fr/2016/03/31/big-data-panorama-des-solutions-2016/.

Table des matières

Table des figures