Analysez des données en batch

Réaliser des traitements distribués de données en temps réel
Créer et visualiser des métriques à partir de données générées en temps réel
- Créer une plateforme distribuée de flux de messages en temps réel

Sommaire

01

PRESENTATION DU PROJET

Le contexte

Nous avons accès a une grande quantité de données, provenant de plusieurs sources Wikipédia en libre accès. Nous allons interrogrer ces données pour identifier les contributeurs les plus importants, à l'édition francophone de Wikipédia, sur différents sujets.

Le contexte

Nos sources de données sont:

- **l'historique des contributions à chaque article de wikipédia français**.

Document xml de 77.7Go (frwiki-20191020-stub-meta-history.xml)

Le contexte

- **la base de données MySQL contenant les liens entre tous les articles**.

Document sql de 11.6Go (frwiki-20191020-pagelinks.sql)

Le contexte

Ces données brutes seront chargées sur un ***système de fichier HDFS***. Puis seront sérialisées à l'aide d'***AVRO***.
Ensuite, nous réaliserons des ***requêtes Spark SQL*** pour nous permettre d'obtenir les résultats voulus.

Le contexte

Nous souhaitons obtenir *les contributeurs les plus importants* pour ces domaines:

- **Cinéma surréaliste.**
- **Anthropologie marxiste.**
- **Guitare.**

02

CHOIX TECHNIQUES

Technologies

Méthodologie

Nous allons stocker les données dans un **Data Lake** qui contiendra les données brutes et nos données sérialisées.

Pour le stockage de ces données, nous utiliserons un système de données distribué qui permettra de passer à l' échelle si cela devient nécessaire. Nous choisissons **HDFS**.

Pour ce qui est de la sérialisation des données, nous opterons pour le format **AVRO** qui permet de stocker les données de manière compacte tout en embarquant le schéma des données. Dans ce projet, le fichier xml regroupant toutes les révisions de pages sera sérialisé, le fichier sql sera traité par un script dans l'application Spark.

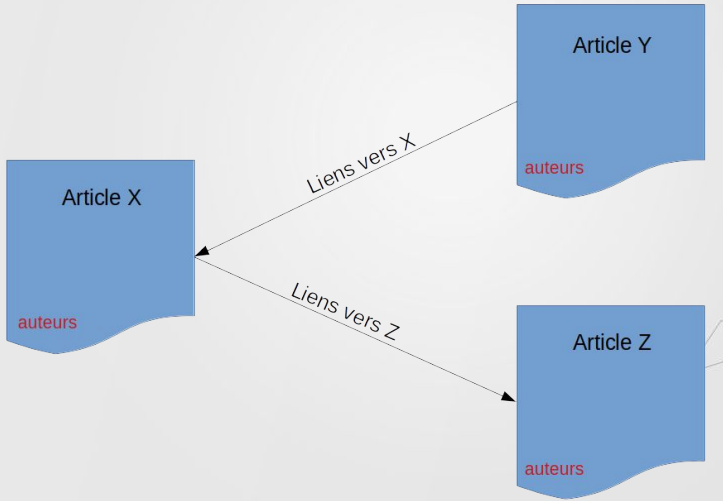
La recherche concerne les pages Wikipédia Français dont l'id a été récupérée:

- Cinéma surréaliste : idPage = **2785024**
- Anthropologie marxiste: idPage = **1590508**
- Guitare: idPage = **3356698**

Technologies

- **Méthodologie**

Nous réaliserons des requêtes pour récupérer les principaux auteurs pour chaque article X tel que décrit sur ce graphe.



Technologies

HDFS

- **Hadoop** est un framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées (au niveau du stockage des données et de leur traitement) et échelonnables (scalables) permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Le noyau d'Hadoop est constitué d'une partie de stockage : **HDFS (Hadoop Distributed File System)**, et d'une partie de traitement appelée MapReduce.

  HDFS est un système de fichiers distribué, extensible et portable développé par Hadoop à partir du GoogleFS. Écrit en Java, il a été conçu pour stocker de très gros volumes de données sur un grand nombre de machines équipées de disques durs. Il permet l'abstraction de l'architecture physique de stockage, afin de manipuler un système de fichiers distribué comme s'il s'agissait d'un disque dur unique.

  Une architecture de machines HDFS (aussi appelée cluster HDFS) repose sur deux types de composants majeurs :

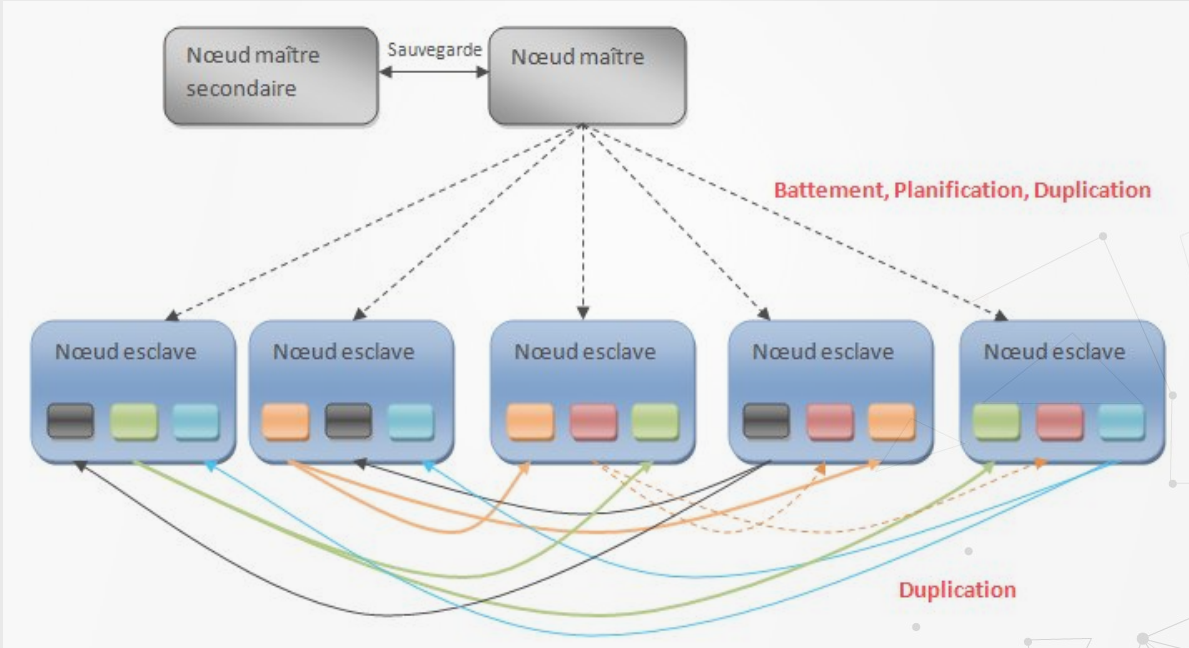
  NameNode : Ce composant gère l'espace de noms, l'arborescence du système de fichiers et les métadonnées des fichiers et des répertoires. Il centralise la localisation des blocs de données répartis dans le cluster. Il est unique mais dispose d'une instance secondaire qui gère l'historique des modifications dans le système de fichiers (rôle de backup). Ce NameNode secondaire permet la continuité du fonctionnement du cluster Hadoop en cas de panne du NameNode d'origine.

  DataNode : nœud de données, ce composant stocke et restitue les blocs de données.

Technologies

HDFS

- **Principe de fonctionnement HDFS**



Technologies

HDFS

- **Organisation des données**

  Organisation hiérarchique des fichiers dans HDFS pour ce projet:

```
/data/
  frwiki/
    20191020/
      raw/
        .snapshot/
        frwiki-20191020-pagelinks.sql
        frwiki-20191020-stub-meta-history.xml
      master/
        history.avsc
        full/
          .snapshot/
          frwiki-20191020-stub-meta-history_{numDoc}_{idPage}.avro
          .....
```

Technologies

HDFS

- **Import des données**

./bin/hdfs dfs -copyFromLocal
/local_data.xml
/data/frwiki/20191020/**raw**/frwiki-20191020-stub-meta-history.xml

./bin/hdfs dfs -copyFromLocal
/local_data.sql
/data/frwiki/20191020/**raw**/frwiki-20191020-pagelinks.sql

Technologies

APACHE AVRO

- **Apache AVRO** est un framework de sérialisation de données élaboré au sein du projet Apache Hadoop. Il utilise JSON pour la définition des types de données, et sérialise les données dans un format binaire plus compact.

  Les schémas sont composés de types primitifs (null, boolean, int, long, float, double, bytes, string) ou complexes (record, enum, array, map, union, fixed).

  Les données en Avro sont stockées avec le schéma correspondant, ce qui signifie que l'article sérialisé peut être lu sans connaître le schéma à l'avance.

Technologies

APACHE AVRO

- **Schéma de sérialisation**

  Schéma Avro des données

- **Script de sérialisation**

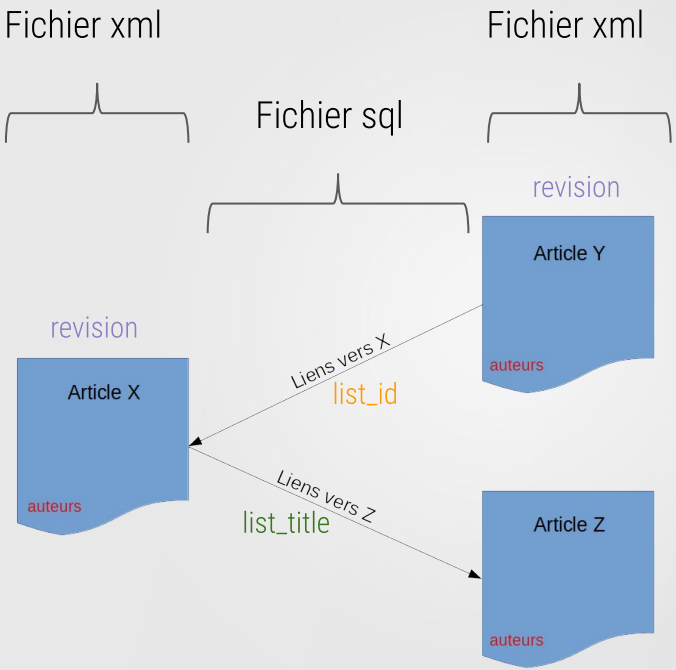
  Script serialize_history.py

03

REQUETES, RESULTATS

Requête

Schéma



Requête

Vues utilisées

Extrait revision

Extrait revision
+----------------------+--------+---+---------+
|contributor |page_id |page_title |rev_id |
+----------------------+--------+---+---------+
SimplyHugo	12394501	'Victor_Crone'	156972394
SimplyHugo	12394501	'Victor_Crone'	156972424
Huster	12394501	'Victor_Crone'	157005306
Nomarcland	12394501	'Victor_Crone'	158854423
Vysotsky	12394501	'Victor_Crone'	159423801
SimplyHugo	12394501	'Victor_Crone'	162611265
Pautard	12394501	'Victor_Crone'	163050601
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157013370
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157013451
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157013478
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157013861
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157015248
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157023325
CuriousReader	12397088	'Le_Photographe_de_Mauthausen'	157026716
Bot2Friday	12397088	'Le_Photographe_de_Mauthausen'	157231504
Alexander Doria	1450701	'Portail:Franz_Liszt/Voir_aussi'	15629531
Bregegrahf	1450701	'Portail:Franz_Liszt/Voir_aussi'	16105665

Extrait list_id

Extrait list_id
+-------+-----+-------------------+----------+
|pl_from|pl_ns|pl_title |pl_from_ns|
+-------+-----+-------------------+----------+
311	0	'Cinéma_surréaliste'	0
2904	0	'Cinéma_surréaliste'	0
3354	0	'Cinéma_surréaliste'	0
11892	0	'Cinéma_surréaliste'	0
11915	0	'Cinéma_surréaliste'	0
12072	0	'Cinéma_surréaliste'	0
13309	0	'Cinéma_surréaliste'	0
21265	0	'Cinéma_surréaliste'	0
22175	0	'Cinéma_surréaliste'	0
39293	0	'Cinéma_surréaliste'	0
56176	0	'Cinéma_surréaliste'	0
56328	0	'Cinéma_surréaliste'	0
58301	0	'Cinéma_surréaliste'	0
66284	0	'Cinéma_surréaliste'	0
67149	0	'Cinéma_surréaliste'	0
70446	0	'Cinéma_surréaliste'	0
75161	0	'Cinéma_surréaliste'	0
124599	0	'Cinéma_surréaliste'	0

Extrait list_title

Extrait list_title
+-------+-----+---+----------+
|pl_from|pl_ns|pl_title |pl_from_ns|
+-------+-----+---+----------+
2785024	0	'1928_au_cinéma'	0
2785024	0	'1962_au_cinéma'	0
2785024	0	'1965_au_cinéma'	0
2785024	0	'Aaltra'	0
2785024	0	'Ado_Kyrou'	0
2785024	0	'Alejandro_Jodorowsky'	0
2785024	0	'Alexander_Hammid'	0
2785024	0	'Alfred_Hitchcock'	0
2785024	0	'Alice_(film,_1988)'	0
2785024	0	'Anemic_cinema'	0
2785024	0	'Animation_(audiovisuel)'	0
2785024	0	'Antonin_Artaud'	0
2785024	0	'Arzach'	0
2785024	0	'Au-delà_du_réel_(film)'	0
2785024	0	'Au_poste_!'	0
2785024	0	'Avant-garde_(art)'	0
2785024	0	'Benoît_Delépine'	0
2785024	0	'Blue_Velvet'	0
2785024	0	'Brazil_(film,_1985)'	0
2785024	0	'Cache-cache_pastoral'	0
2785024	0	'Carlos_Atanes'	0

Requête

Requête pour la recherche des contributeurs

Script complet

```
Request  =  "SELECT rev.contributor contributors, COUNT(rev.contributor) quantity " \
            "FROM revision rev " \
            "WHERE rev.page_id = {} " \
                    "OR rev.page_title = {} " \
                    "OR rev.page_id in (SELECT pl_from FROM list_id) " \
                    "OR rev.page_title in (SELECT pl_title FROM list_title) " \
            "GROUP BY contributors order by quantity desc" \
            .format(pageId, pageTitle)
```

OÙ

```
df_pagelinks.filter(df_pagelinks.pl_title == pageTitle).createOrReplaceTempView("list_id")
df_pagelinks.filter(df_pagelinks.pl_from == pageId).createOrReplaceTempView("list_title")
```

Résultats

Résultats pour les différentes recherches

```
pageTitle = Guitare
pageId = 3356698
+------------+--------+
|contributors|quantity|
+------------+--------+
|Synthwave.94|    2627|
|  HerculeBot|    2283|
|      Addbot|    1831|
|  DSisyphBot|    1723|
| Le Pied-bot|    1627|
|     Phe-bot|    1434|
| OrlodrimBot|    1305|
|       Vlaam|    1278|
|      FlaBot|    1264|
|    Speculos|    1101|
+------------+--------+
Program took 2600.64 sec. to perform.
```

```
pageTitle = Cinéma_surréaliste
pageId = 2785024
+------------+--------+
|contributors|quantity|
+------------+--------+
|  Orthogaffe|      68|
|      MedBot|      61|
|    Puckstar|      54|
|    Arcane17|      46|
|     Phe-bot|      46|
|    Nataraja|      42|
|    Chicobot|      38|
|    JacquesD|      36|
|   HasharBot|      31|
|      Yuzuru|      29|
+------------+--------+
Program took 1872.47 sec. to perform.
```

```
pageTitle = Anthropologie_marxiste
pageId = 1590508
+--------------+--------+
|   contributors|quantity|
+--------------+--------+
|      Recyclage|      19|
|       Methexis|      17|
|   Bachi-bozouk|      17|
|    Titi Sitria|      15|
|         MedBot|      15|
|           PAC2|      14|
|Homo sovieticus|      13|
|    Winckelmann|      11|
|        Madpier|       8|
|     Orthogaffe|       8|
+--------------+--------+
Program took 1857.70 sec. to perform.
```

Résultats

Vérification

Les vues du fichier sql nous donnent une liste d'id de pages et une liste de titres de pages liées a notre article étudié.

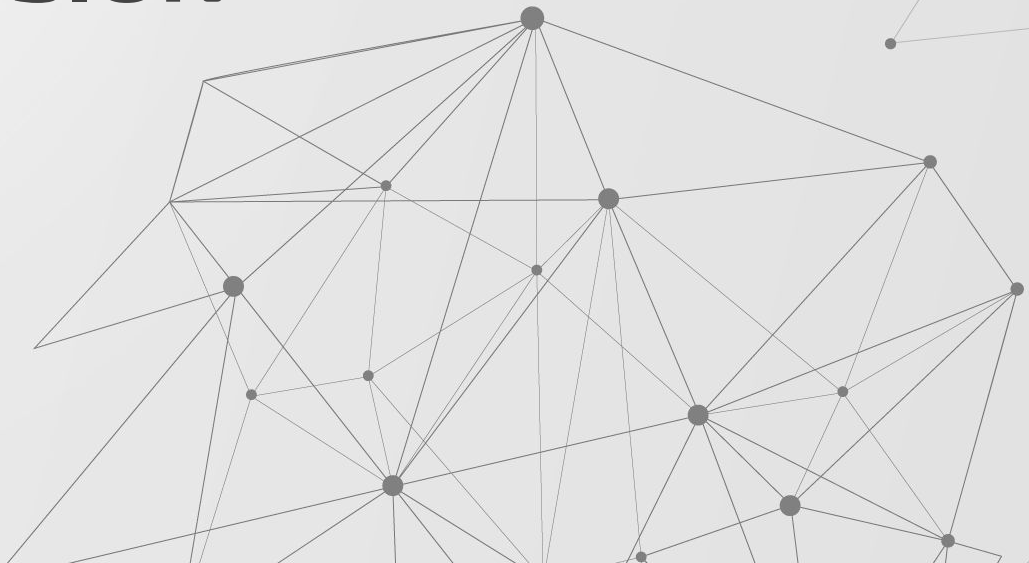
La liste de titre est passée à un script qui appelle une API MediaWiki et retourne son id, on obtient donc la liste d'id correspondante a notre liste de titre.

Les deux listes d'id sont concaténées puis passées a un script de désérialisation qui enregistre chaque page dont l'id est listé, dans un fichier json.

La vérification est possible car le fichier json généré est d'une taille beaucoup plus faible que l'original. Le nombre de contributeurs est compté par une recherche textuelle avec un éditeur de texte ou un grep en ligne de commande, puis comparé au résultat de la requête.

Une vérification avec une requête simple peut se faire également sur le fichier généré avec ce script.

04
CONCLUSION



Un projet d'analyse de données en batch

- Mettre en place des outils d'analyse de données par lot
- Concevoir un data lake
- Sérialiser des données semi-structurées avec Avro
- Représenter des structures de données complexes
- Stocker des données distribuées avec HDFS

Ressources

Web

- https://fr.wikipedia.org/
- https://www.ibm.com/developerworks/xml/library/x-hiperfparse/
- https://docs.python.org/fr/3/library/xml.etree.elementtree.html
- https://blog.ippon.fr/2016/09/26/formats-et-methodes-de-serialisation-rest/
- https://www.edureka.co/blog/apache-hadoop-hdfs-architecture
- https://docs.databricks.com/spark/latest/dataframes-datasets/index.html
- https://stackoverflow.com/questions/40557606/how-to-url-encode-in-python-3/40557716
- https://help.sap.com/viewer/50f26aa0f2044127bc5f6d5ad3d090fe/Cloud/en-US/2e64a49076b8101480abd76da746ffe7.html
- https://spark.apache.org/docs/latest/api/python/pyspark.html?highlight=flatmap

Et bien d'autres...

MERCI

Avez-vous des **questions**?

f2buttet@gmail.com
06.84.19.58.69