PROJET BIGDATA 2020-2021: ETUDIER L'EVOLUTION DE LA PANDEMIE COVID19 VIA SON IMPACT MEDIA

INF728 – BASES DE DONNÉES NON RELATIONNELLES : NoSQL

LÉVÊQUE Florian – PARTIMBENE Vincent – ROSE Louis – SOBHANI Armand

PLAN

Présentation du projet et des objectifs

Architecture

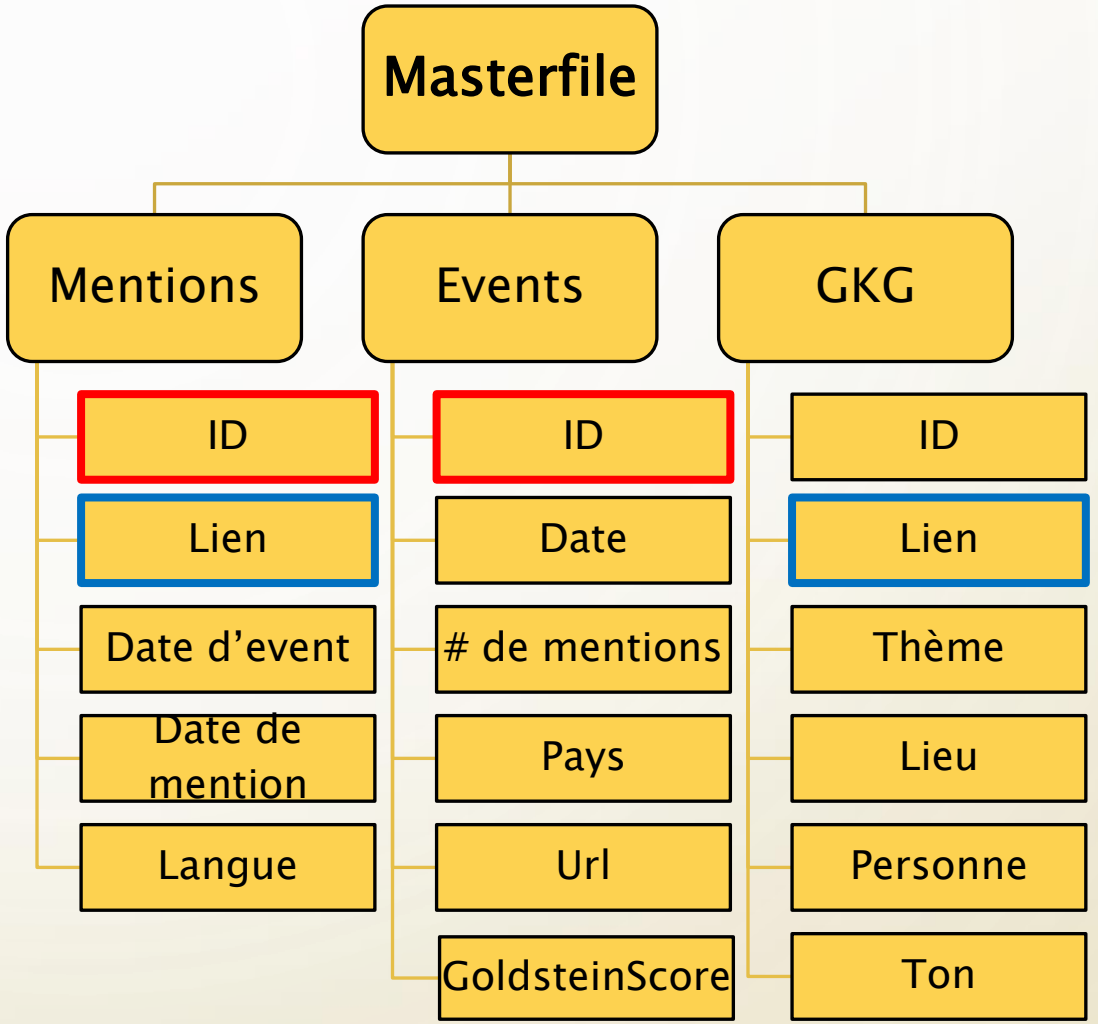
Requêtes

Performances et limites

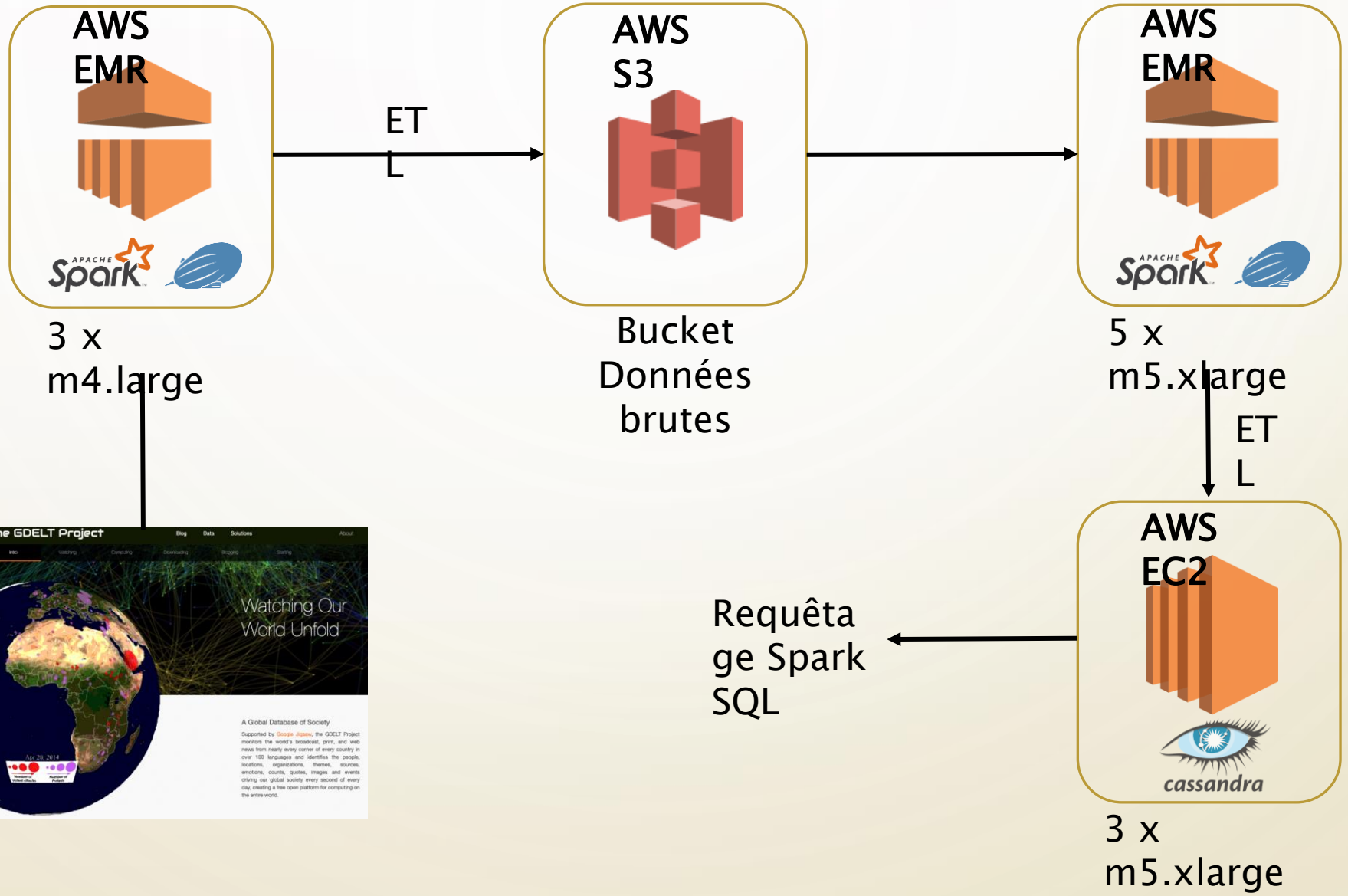
Discussions et ouvertures

PRÉSENTATION DU PROJET ET DES OBJECTIFS

1ère étape : compréhension des tables et des colonnes nécessaires aux jointures.



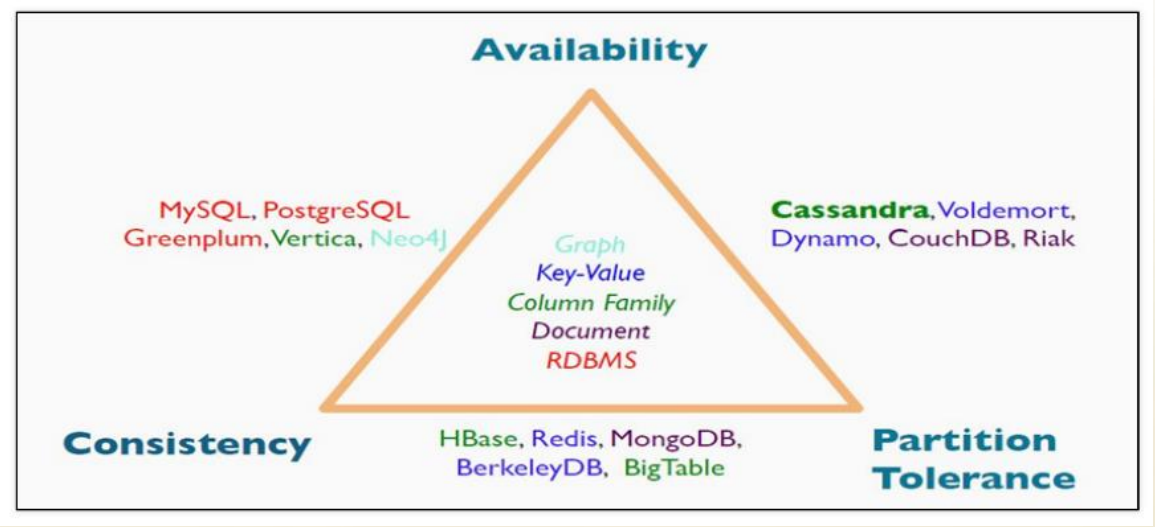
ARCHITECTURE



QUELQUES POINTS : CASSANDRA VS MONGODB

Cassandra semble la meilleure option pour :

- High Availability – temps de réponse très rapide
- Rapidité d'écriture et
  de scalability (plusieurs nœuds pouvant écrire)
- Possibilité de faire des requêtes proches du langage SQL



Source : https://phoenixnap.it/kb/cassandra-vs-mongodb

CASSANDRA PRÉCISIONS :

- 3 nœuds Cassandra répartis dans des régions différentes (us-east-1a, us-east-1b, us-east-1c).
- Configuration : Passage de SimpleSnitch ⇒ à Ec2Snitch.
- Choix d'un réplica factor de 3.
- Utilisation du degré de consistency initial pour lecture et écriture: One.
- SimpleStrategy pour le déploiement des réplicas.

Une telle configuration permet de perdre 2 nœuds sur les trois tout en maintenant les opérations de lecture et d'écriture.

RF	Used CL	Number of allowed simultaneous failed nodes without compromising HA
3	ONE/LOCAL_ONE	2 nodes
3	QUORUM/LOCAL_QUORUM	1 node
5	ONE/LOCAL_ONE	4 nodes
5	QUORUM/LOCAL_QUORUM	2 nodes

- Source : http://www.doanduyhai.com/blog/?p=13216

REQUÊTES - 1

```
// Champs de la table events récupérés :
val Events_DF_bis = Events_tmp.select(
    $"value".getItem(0).as("globalevent_id"),
    $"value".getItem(1).as("year_month_day"),
    $"value".getItem(2).as("year_month"),
    $"value".getItem(3).as("year"),
    $"value".getItem(31).as("num_mention"),
    $"value".getItem(53).as("country"),
    $"value".getItem(60).as("source_url")
    )

// Champs de la table Mentions récupérés :
val Mentions_DF_bis = Mentions_tmp.select(
    $"value".getItem(0).as("globalevent_id"),
    $"value".getItem(1).as("event_time_date"),
    $"value".getItem(2).as("mention_time_date"),
    $"value".getItem(5).as("mention_identifier"),
    $"value".getItem(14).as("article_language")
    )

// Champs de la table Gkg récupérés :
val Gkg_DF_bis = Gkg_tmp.select(
    $"value".getItem(0).as("gkg_record_id"),
    //$"value".getItem(1).as("DATE"),
    $"value".getItem(3).as("source_common_name"),
    $"value".getItem(4).as("document_identifier"),
    $"value".getItem(7).as("themes"),
    $"value".getItem(9).as("locations"),
    $"value".getItem(11).as("persons"),
    $"value".getItem(15).as("tone")
    )
```

GKG filtrage

- Theme sur CORONAVIRUS

JOIN Mention & GKG

MentionIdentifier

DocumentIdentifier

- Langue

JOIN with Event *on GlobalEventId*

- Pays
- Jour

COUNT(Globaleventid) → # Articles

COUNT (DISTINCT Globaleventid) → # d'events

WHERE *YearMonthDay* = 'jour' **AND** *Country* = 'pays' **AND** *ArticleLanguage* = 'langue'

Obtention d'un DF Spark stocké sur Cassandra et requêté en SparkSQL

```
//Question n°1 : Afficher le nombre d'articles/évènements qui parlent de COVID qu'il y a eu pour chaque triplet (jour, pays de l'évènement, langue de l'article).

z.show(spark.sql("""
                SELECT count(globalevent_id) as mentions_number, count(DISTINCT globalevent_id) as events_number
                FROM view_q1
                WHERE year_month_day = '20201002'
                AND country = 'US'
                AND article_language = 'eng'
                """)
       )
```

settings

mentions_number	events_number
105634	18257

```
CassandraConnector(sc.getConf).withSessionDo { session =>
      session.execute(
          """
            DROP KEYSPACE IF EXISTS gdelt;
          """
          )
      session.execute(
          """
            CREATE KEYSPACE IF NOT EXISTS gdelt
            WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 3 };
          """)
          session.execute(
          """
            CREATE TABLE IF NOT EXISTS gdelt.q1(
               globalevent_id text,
               mention_identifier text,
               year_month_day text,
               country text,
               article_language text,
            PRIMARY KEY ((globalevent_id, mention_identifier), year_month_day));
          """
        )
}
```

REQUÊTES – 2

JOIN Events & Mentions on *GlobalEventId*

COUNT (*GlobalEventId*) → # Mentions
WHERE *Country* = 'pays' **GROUP BY** YearMonth, Country, Globaleventid

Obtention d'un DF Spark stocké sur un cluster Cassandra et requêté en SparkSQL

```
CassandraConnector(sc.getConf).withSessionDo { session =>
      session.execute(
        """
          CREATE KEYSPACE IF NOT EXISTS gdelt
          WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 3 };
        """)
        session.execute(
        """
          CREATE TABLE IF NOT EXISTS gdelt.q2(
             globalevent_id text,
             year_month_day text,
             year_month text,
             year text,
             country text,
             mention_identifier text,
          PRIMARY KEY (mention_identifier, year_month_day));
        """
        )
}
```

```
val Pays : String = "'US'"
```

```
Pays: String = 'US'
```

```
// Question n°2 : Pour un pays donné en paramètre, affichez les évènements qui y ont eu place triées par le nombre de mentions (tri décroissant); permettez une agrégation par jour/mois/année.

z.show(spark.sql(s"""
                SELECT country, year_month, globalevent_id, count("globalevent_id") as number_mentions
                FROM view_q2
                WHERE country = $Pays
                GROUP BY year_month, country, globalevent_id
                ORDER BY count("globalevent_id") DESC
                """)
       )

/* Attention aggregation par jour à prévoir --> Number_Mentions déjà créée + sum sur cette colonne */
```

settings

country	year_month	globalevent_id	number_mentions
US	202010	950358111	454
US	202010	949992824	355

Limitation à 1 mois de données au sein de Cassandra --> Problème d'error ZLIB non résolu.
1 an de data sur S3.

REQUÊTES – 3

JOIN Mention & GKG on *MentionIdentifier* and *DocumentIdentifier*

COUNT(*SourceCommonName*) → # Articles

MEAN(Tone)

WHERE *SourceCommonName* = 'source'

GROUP BY *SourceCommonName, YearMonth, Themes, Persons, Locations*

Obtention d'un DF Spark stocké sur un cluster Cassandra et requêté en SparkSQL

```
CassandraConnector(sc.getConf).withSessionDo { session =>
      session.execute(
         """
            CREATE KEYSPACE IF NOT EXISTS gdelt
            WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 3 };
         """)
         session.execute(
         """
            CREATE TABLE IF NOT EXISTS gdelt.q3(
               gkg_record_id text,
               source_common_name text,
               document_identifier text,
               themes text,
               locations text,
               persons text,
               tone double,
               globalevent_id text,
               event_time_date text,
               mention_time_date text,
               article_language text,
               year_month_day text,
               year_month text,
               year text,
            PRIMARY KEY ((gkg_record_id, document_identifier, mention_time_date), year_month_day));
         """
        )
}
```

```
// Question n°3 : Pour une source de donnés passée en paramètre (gkg.SourceCommonName) affichez les thèmes, personnes, lieux dont les articles de cette sources parlent ainsi que le nombre d'articles et le ton moyen des articles (pour chaque th
    permettez une agrégation par jour/mois/année.

z.show(spark.sql(s"""
                SELECT source_common_name,  year_month, themes, persons, locations, count(source_common_name) as articles_number, mean(tone) as mean_tone_articles
                FROM view_q3
                WHERE source_common_name == $Source
                GROUP BY source_common_name,  year_month, themes, persons, locations
                """)
        )
```

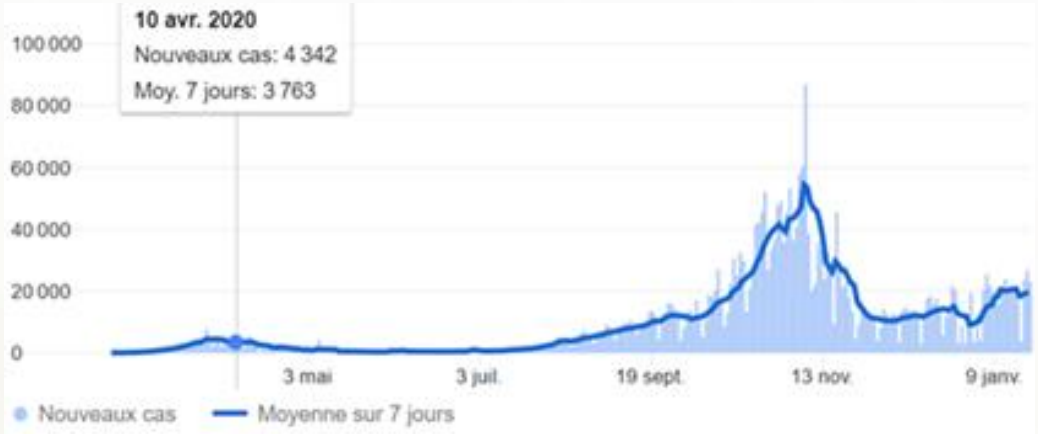
READY

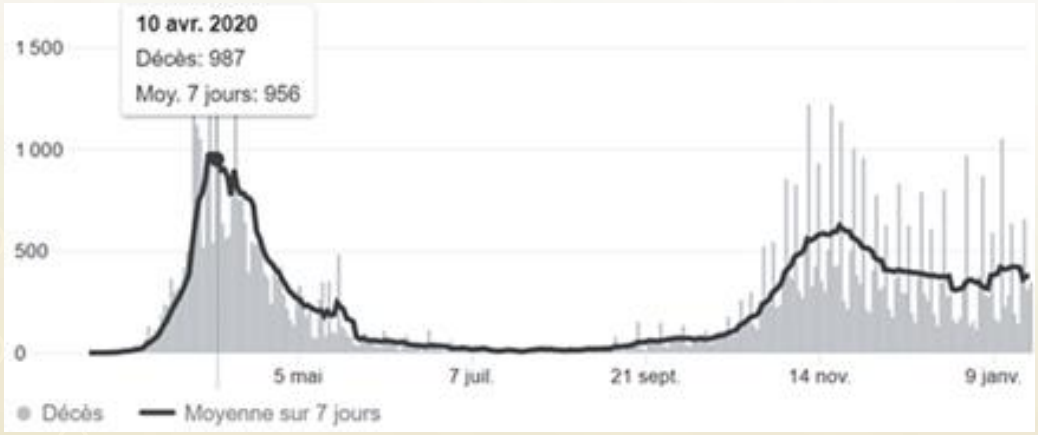
settings

source_common_name	year_month	themes	persons	locations	articles_number	mean_tone_articles
deseret.com	202010	LEADER;TAX_FNCACT;TAX_FNCACT_PRESIDENT;USPEC_POLITICS_GEN	young ford;seth ford;susan b anthony	New York, United States	2	1.502732276916504

REQUÊTES – 4 : PRÉDIRE LES VAGUES ÉPIDÉMIQUES

- <u>Problématique :</u> Quelles données prédire ? Nouveaux cas, Hospitalisations, Décès...?

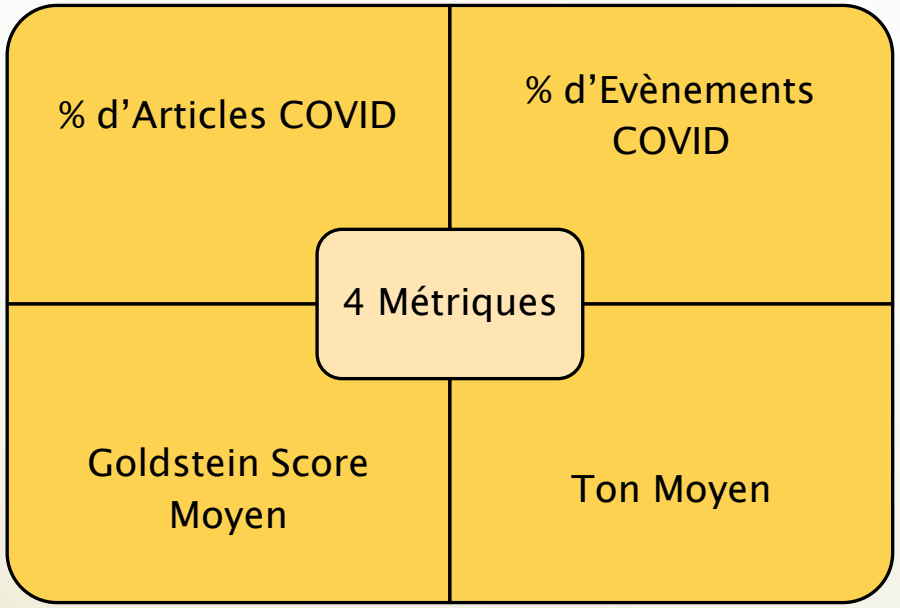




- **Bleu** : nouveaux cas. **Noir** : Décès
- Données sur la France, quotidiennes, de février 2020 à janvier 2021
- On remarque que en Mars/Avril 2020 (première vague), le nombre de Cas et de Décès ne concordent pas.
- **Analyse** : L'insuffisance des tests COVID effectués sur cette période fausse les données des nouveaux cas.
- **Conclusion** : on va utiliser les données de décès, plus fiables.

REQUÊTES - 4 - Choix des métriques pour la prédiction

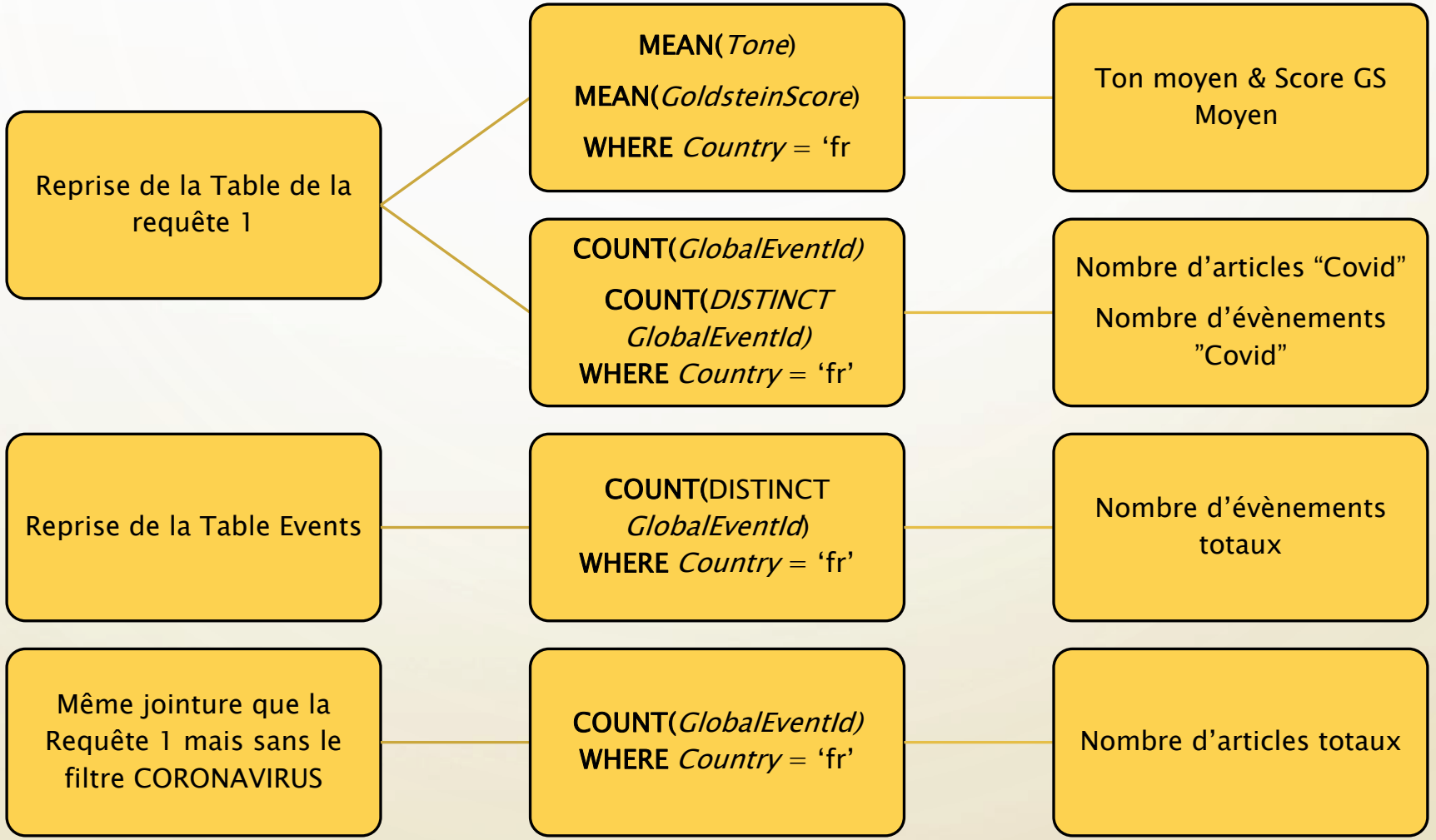
Pour prédire la courbe des décès, on va tester quatre métriques :



- Les % d'articles traitant du COVID
- Les % d'évènements liés au COVID
- Le Ton Moyen des Articles traitant du COVID
- Le score Goldstein moyen des évènements liés au COVID

REQUÊTES – 4 : CONSTRUCTION DES REQUÊTES

- On reprend et modifie les tables et requêtes des 3 premières questions afin de construire ces métriques :

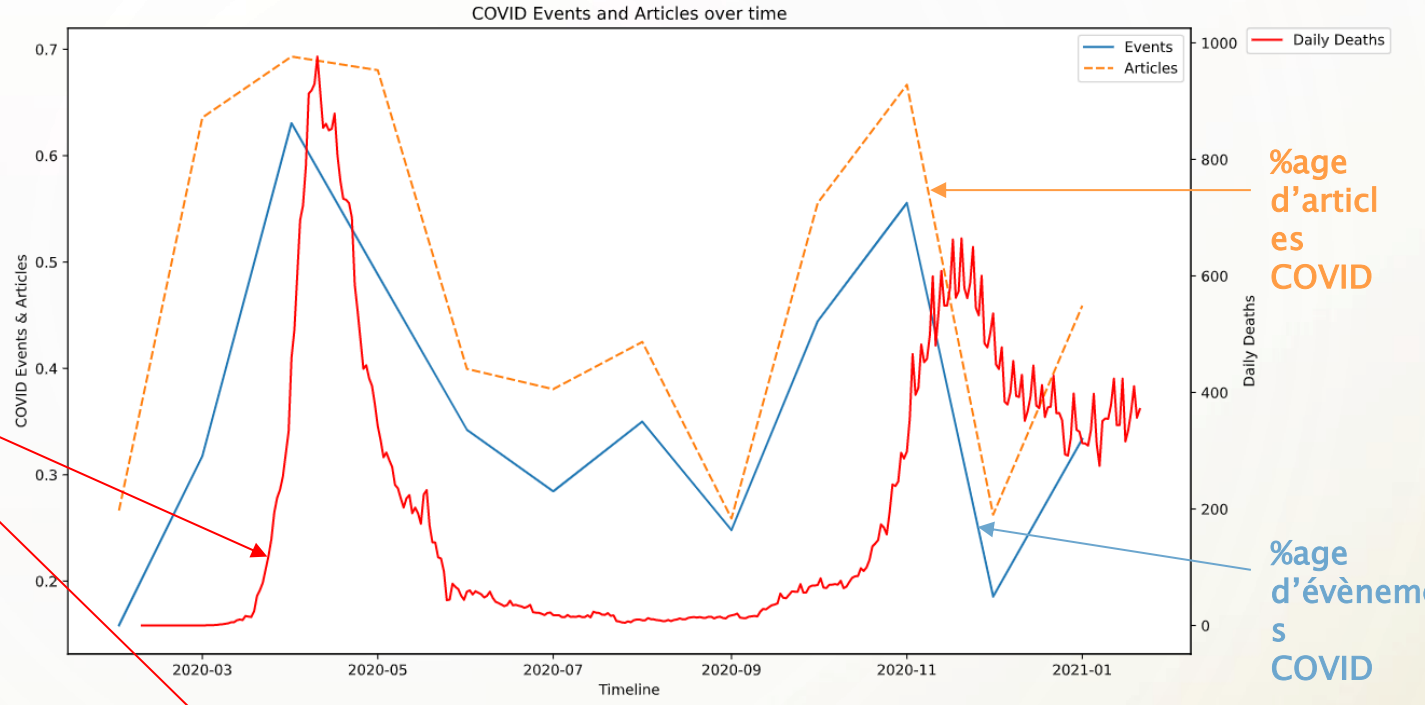


REQUÊTES – 4 : EXÉCUTION DES REQUETES DONNANT LES 4 METRIQUES

- Pour calculer ces métriques, nous avons uniquement fait du calcul en local.
- Nous nous sommes donc limités aux jours 01 et 02 de chaque mois.
- On obtient ainsi un point de données par mois, pour chaque métrique (cf slide suivante).

- Pour vérifier que cela ne fausse pas les estimations, nous avons au préalable sélectionné aléatoirement une semaine de données (fin mars), et vérifié que, pour les évènements du 25 mars, les articles associés paraissaient majoritairement dans les 2 jours suivants.
- C'est le cas : on obtient plus de 90% des articles concernés dans les 2 jours.

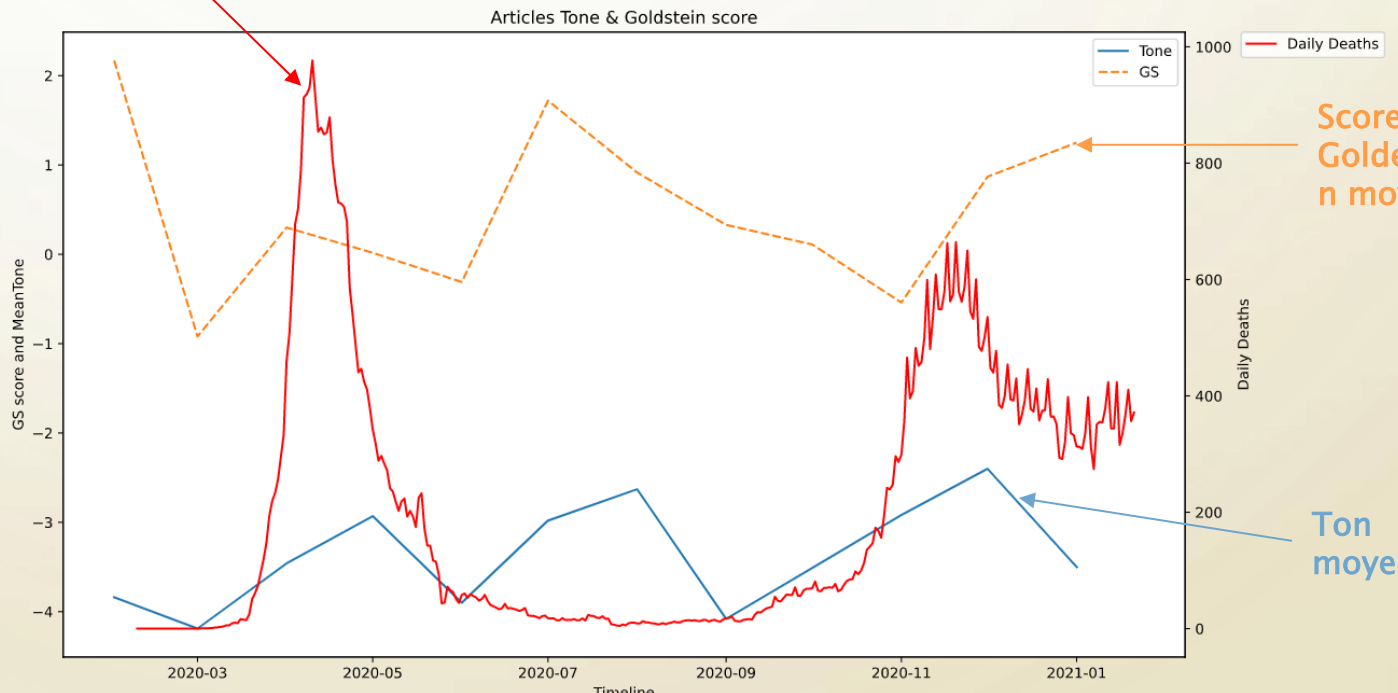
REQUÊTES – 4



Nombre de décès quotidiens

%age d'articles COVID

%age d'évènements COVID



Score GoldenStein moyen

Ton moyen

CONCLUSION :

- Le % d'articles et d'évènements liés au COVID est une bonne métrique des décès. On remarque une légère avance de phase, ce qui est positif pour prédire.
- Le score Goldstein et le MeanTone ne sont par contre pas des métriques valables.

PERFORMANCES ET LIMITES

AWS EC2 :

- 3 x m5.xlarge : 0,192 USD par heure

AWS EMR :

- 3 x m4.large : 0,10 USD par heure
- 5 x m5.xlarge : 0,192 USD par heure

Budget total : 50 USD de crédits AWS dépensés

POINTS DIFFICILES

- Comptes AWS educate : limitation du partage, besoin de modifier les tokens à chaque connexion.
- Problèmes sur les fichiers de Gdelt : error ZLIB, timeout pool. Impossibilité de télécharger certains mois.
- Besoin de profils plus orientés en informatique au sein du groupe : difficulté à mettre en place l'ensemble de l'infrastructure impliquant une réflexion limitée sur la partie NoSql …
- Par faute de temps, l'optimisation des clusters n'a pas pu être abordée

RÉSULTATS PROJETS :

- Mise en place d'une infrastructure complète permettant la gestion du projet.
- Montée en compétence de l'ensemble de l'équipe sur les technologies AWS, Zeppelin, Spark.
- Observation de patterns entre le pourcentage d'article concernant le COVID et le nombre de décès dus à cette maladie.
- Prise de conscience des barrières technologiques et des connaissances nécessaires pour le déploiement d'un projet Big Data.