Projet GDELT

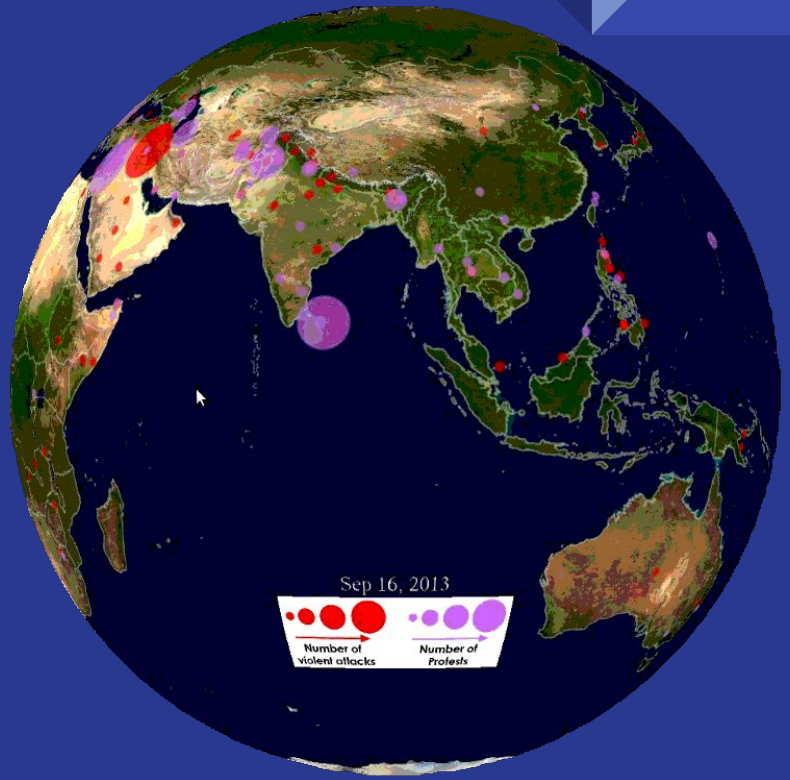
Aghmari Imane

Richard Vincent

Di Wu Léa

Savouré Gaël

El Attaoui Farid



TELECOM Paris

IP PARIS

Sommaire

1. Choix d’architecture
2. Modélisation des requêtes
3. Démonstration
4. Volumétrie
5. Pistes d’optimisation

TELECOM Paris

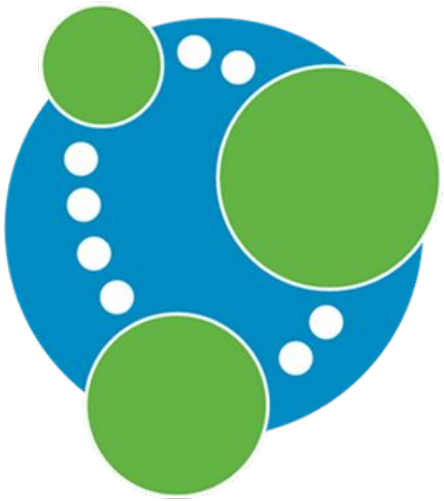
IP PARIS

Sommaire

TELECOM Paris

IP PARIS

ARCHITECTURE



neo4j

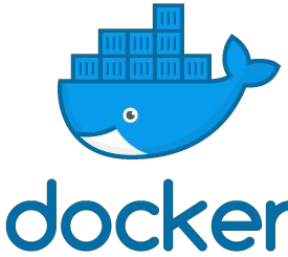
Pourquoi Cassandra ?

Avantages

- Très accessible (Availability)
- Très évolutif (Scalability)
- Gros flux Entrant
- Consistance modifiable

Inconvénients

- Peu de flexibilité sur le langage
  - Spark
- Besoin de définir un schéma
  - Préparation en amont



avec

Pourquoi docker ?

- Facilité de déploiement.
- Minimisation du paramétrage de cassandra.

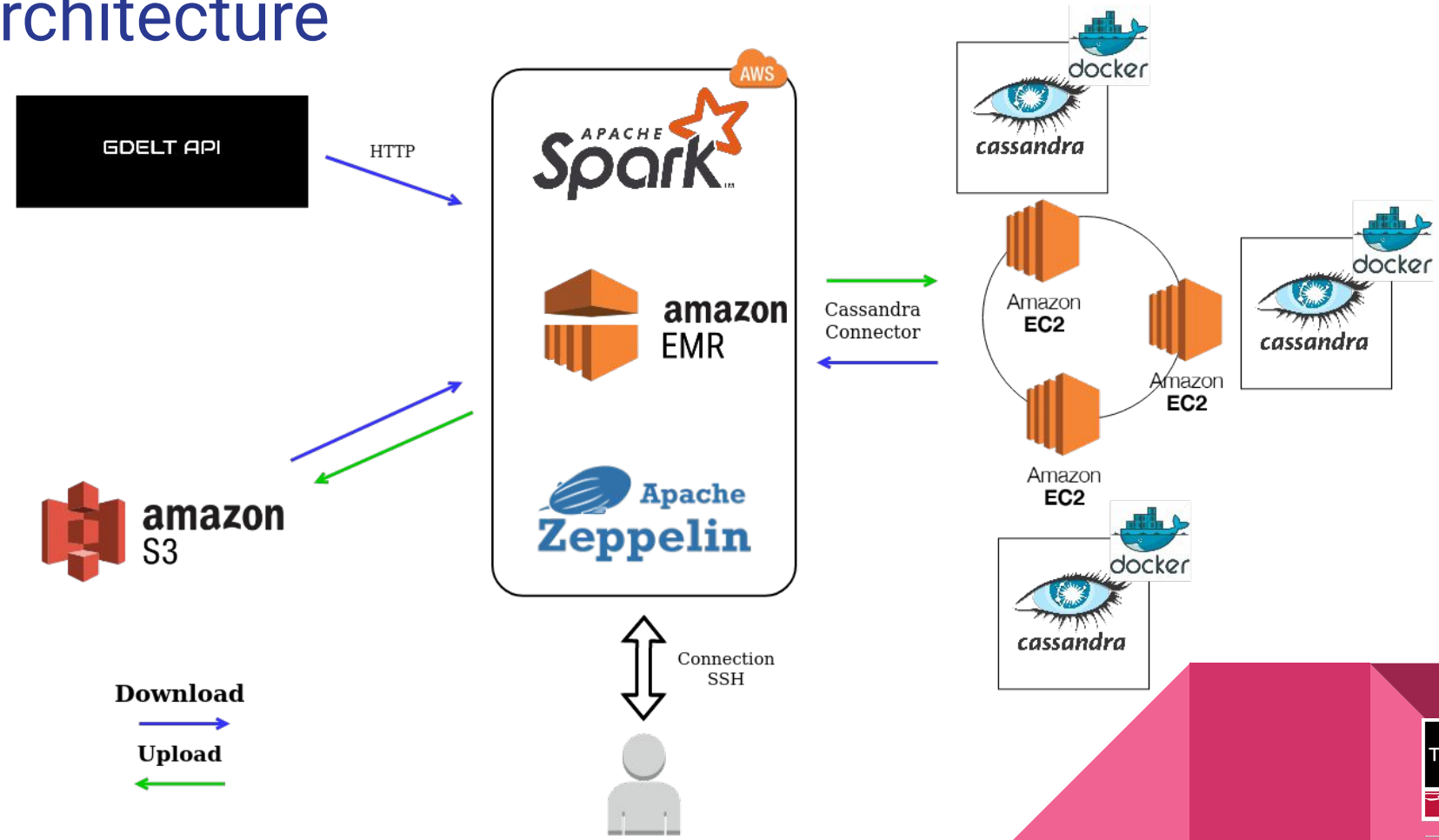
Paramètres du cluster

- **Vnodes**: 256 (default)
- **Snitch**: Ec2Snitch
- **Replication Strategy**:
  NetworkTopologyStrategy
- **Consistency Level**:
  Write: LOCAL_QUORUM
  Read: LOCAL_ONE
- **Number of nodes:** 3

TELECOM Paris

IP PARIS

Architecture



Déploiement automatique

Avantages:

- Déploiement simple pour chaque personne de l'équipe.
- Minimisation des coûts.
- Personnalisation (taille du cluster, type d'instance, …)

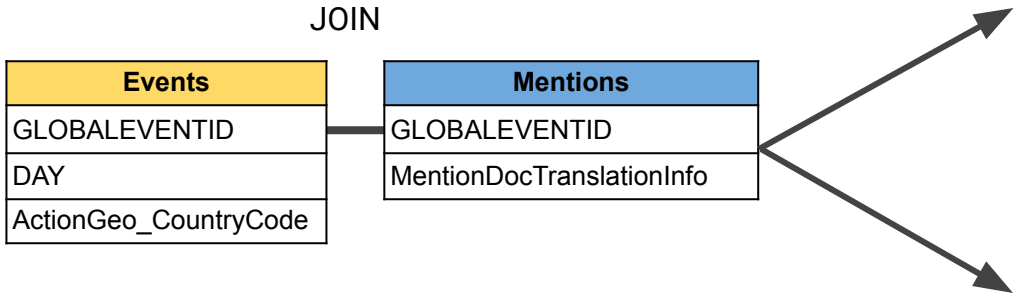
Sommaire

TELECOM Paris

IP PARIS

Query 1 - Modélisation

- Afficher le nombre d'articles/évènements qu'il y a eu pour chaque triplet (jour, pays de l'évènement, langue de l'article).



GROUP BY

DAY, COUNTRY, LANGUAGE

COUNT

EVENTS

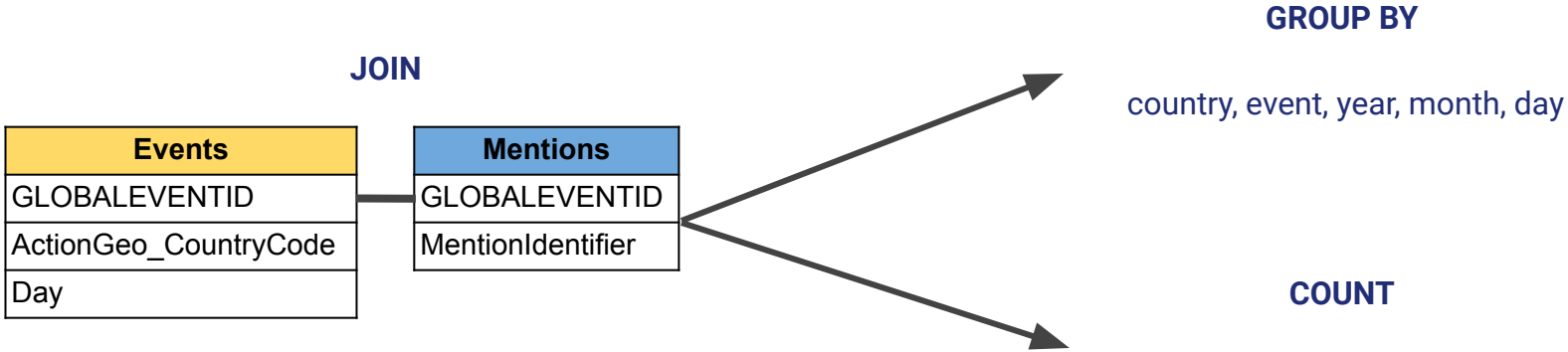
Query 1 - Table dans Cassandra

```
CREATE TABLE IF NOT EXISTS gdelt.country_events (
                date text,
                country text,
                language text,
                count_events int,
                PRIMARY KEY (date, country, language)
            );
```

event_by_day	
Date	K
Country	C
Language	C
Count_events	

Query 2 - Modélisation

- Afficher tous les évènements qui se sont déroulés dans un pays donné en paramètre.
- Trier de manière décroissante par le nombre de mentions, veiller à permettre une agrégation par jour/mois/année.



Query 2 - Table dans Cassandra

```
CREATE TABLE IF NOT EXISTS gdelt.country_events (
  country text,
  year int,
  month int,
  day int,
  event int,
  num_mentions int,
  PRIMARY KEY (country, year, month, day)
);
```

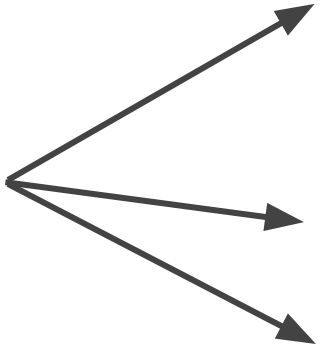
country_events	
country	(K)
year	(C)
month	(C)
day	(C)
event	
num_mentions	

ORDER BY : number of mentions (DESC)

Query 3 - Modélisation

Pour une source de données passée en paramètre (gkg.SourceCommonName) afficher les thèmes, personnes, lieux dont les articles de cette source parlent ainsi que le nombre d'articles et le ton moyen des articles (pour chaque thème/personne/lieu); permettre une agrégation par jour/mois/année.

GKG
GKGRECORDID
SourceCommonName
Themes
V2Tone
DATE
Person
V2Locations



GROUP BY

SourceCommonName, Theme, Date

COUNT

number of articles

Sum

Tone

Query 3 - Table dans Cassandra

```
CREATE TABLE IF NOT EXISTS gdelt.article_by_theme (
    source_common_name text,
    year int,
    month int,
    day int,
    theme text,
    num_article int,
    sum_tone int,
    PRIMARY KEY (source_common_name, year, month, day, theme)
);
```

article_by_theme	
source_common_name	K
year	C
month	C
day	C
theme	C
num_article	
sum_tone	

UNION : Sur les 3 tables

Average : sum_tone / num_article

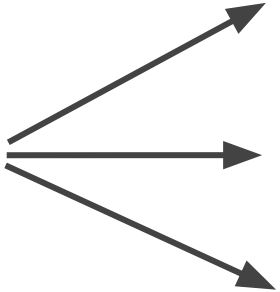
Query 4 - Modélisation

Dresser la cartographie des relations entre les pays d'après le ton des articles : pour chaque paire (pays1, pays2), calculer le nombre d'articles, le ton moyen (agrégation sur Année/Mois/Jour, filtrage par pays ou carré de coordonnées)

JOIN

Events
GLOBALEVENTID
ActionGeo_CountryCode
Day

Mentions
GLOBALEVENTID
MentionDocTone
MentionDocTranslationInfo



GROUP BY

Day, ActionGeo_CountryCode, translation_info

COUNT

GLOBALEVENTID

SUM

MentionDocTone

Query 4 - Table dans Cassandra

```
CREATE TABLE IF NOT EXISTS gdelt.country_map (
    country_code text,
    translation_info text,
    year int,
    month int,
    date int,
    num_article int,
    sum_tone float,
    PRIMARY KEY ((translation_info, country_code), year, month, day)
);
```

country_map	
country_code	K
translation_info	K
year	C
month	C
date	C
num_article	
sum_tone	

Average : sum_tone / num_article

Sommaire

TELECOM Paris

IP PARIS

Démonstration

TELECOM Paris

IP PARIS

Sommaire

1. Choix d’architecture
2. Modélisation des requêtes
3. Démonstration
4. Volumétrie
5. Pistes d’optimisation

TELECOM Paris

IP PARIS

Volumétrie:

Fichiers bruts: 425Go

	Size (Cassandra)
query1	0.8M bytes = 0.8Mo
query2	0.1M bytes = 0.1Mo
query3	45M bytes = 45Mo
query4	10.3M bytes = 10.3Mo

Budget

Utilisation de ~ 50 $

AWS Educate Starter Account

Your cloud journey has only just begun. Use your AWS Educate Starter Account to access the AWS Console and resources, and start building in the cloud!

AWS Educate Starter Account

Your account has an estimated **49** credits remaining and access will end on **Dec 12, 2020**.

Note: Clicking this button will take you to a third party site managed by Vocareum, Inc. ("Third Party Servicer"). In addition to the AWS Educate terms of service, your use of the AWS Educate Starter Account is governed by the Third Party Servicer's terms, including its Privacy Policy. AWS assumes no responsibility or liability and makes no representations or warranties regarding services provided by a Third Party Servicer.

Sommaire

1. Choix d’architecture
2. Modélisation des requêtes
3. Démonstration
4. Volumétrie
5. Pistes d’optimisation

Optimisation: 15 minutes Batch

- Implémenter des Counting Tables
  - Seulement UPDATE pas d'INSERT
  - Calcul d'agrégat direct
- Implémentation de partition key supplémentaire
- Optimisation de lecture avec Cassandra Connector

Cassandra Connector cqlsh

Percentile	Range Latency (micros)	Read Latency (micros)
50%	454.83	1629.72
75%	545.79	1629.72
95%	2346.80	2816.16
98%	2346.80	2816.16
99%	2346.80	2816.16
Min	379.02	1358.10
Max	2346.80	2816.16

Des questions?

https://github.com/vincrichard/GDELT_Project

TELECOM Paris

IP PARIS