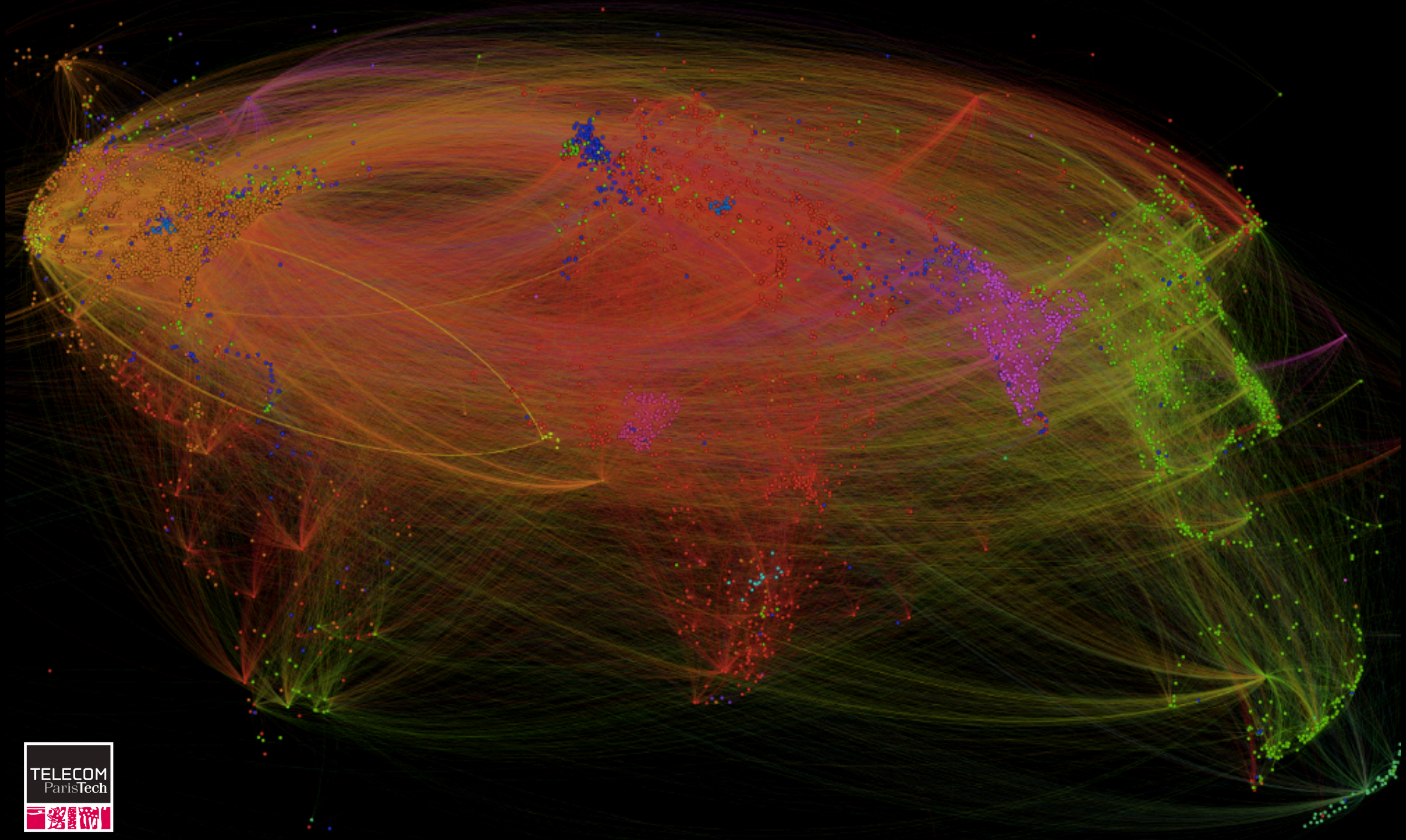
TELECOM PARISTECH - 2019

NOSQL - GDELT



TELECOM ParisTech

GDELT

SUJET

PROBLÉMATIQUE

- Proposer un système de stockage distribué, resilient et performant sur AWS pour les données de GDELT.
  Afficher :
  - le nombre d'articles / événements pour chaque (jour, pays de l'événement, langue de l'article)
  - pour un acteur(pays/organisation …) ⇒ afcher les evenements qui y font reference
  - les sujets (acteurs) qui ont eu le plus d'articles positifs/negatifs (mois, pays, langue de l'article)
  - acteurs/pays/organisations qui divisent le plus

LES DONNÉES

Global Event ID

GKGRecordID

Mentions

- *Date of mention*
- *Language of mention*
- *Type of mention*
- *Tone of mention*

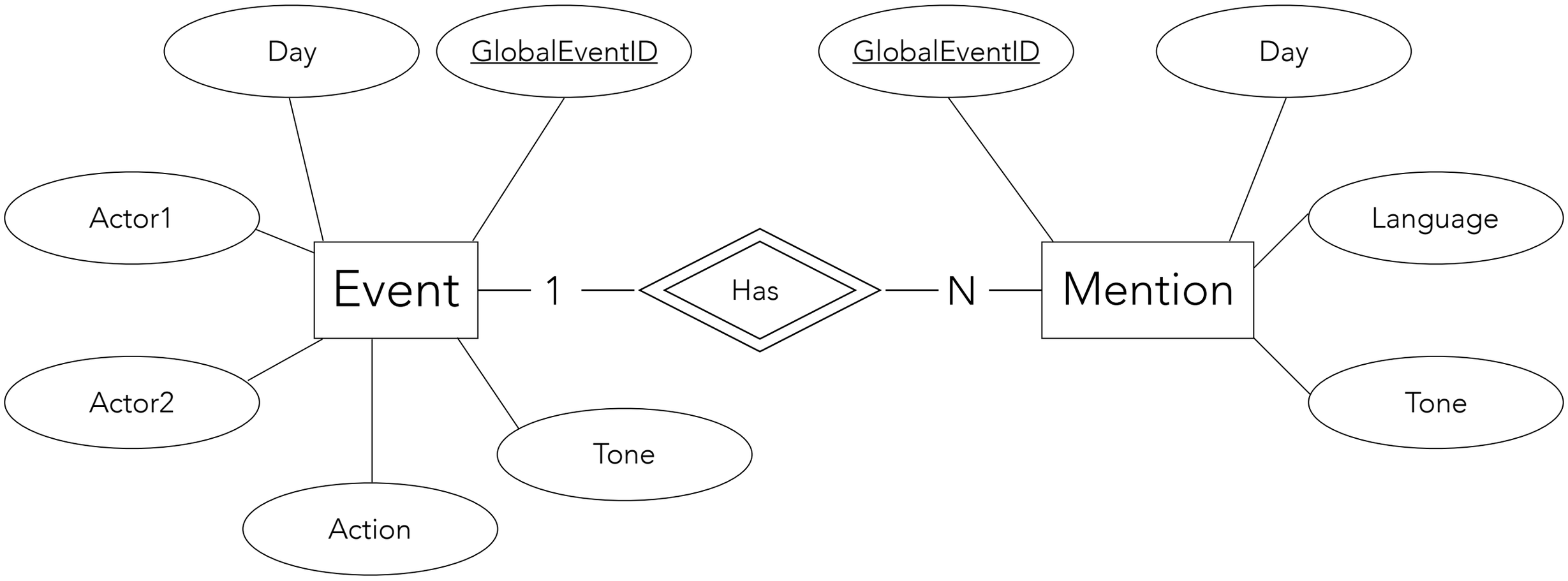
Event

- *Date of event*
- *Actor 1 Code + Country*
- *Actor 1 Name*
- *Actor 2 Code + Country*
- *Actor 2 Name*
- *Number of Articles*
- *Average Tone*
- *Action Country*
- *Action Longitude*
- *Action Latitude*

GKG

- *Date*
- *Theme*
- *Collection*
- *Persons*

MODÈLE CONCEPTUEL

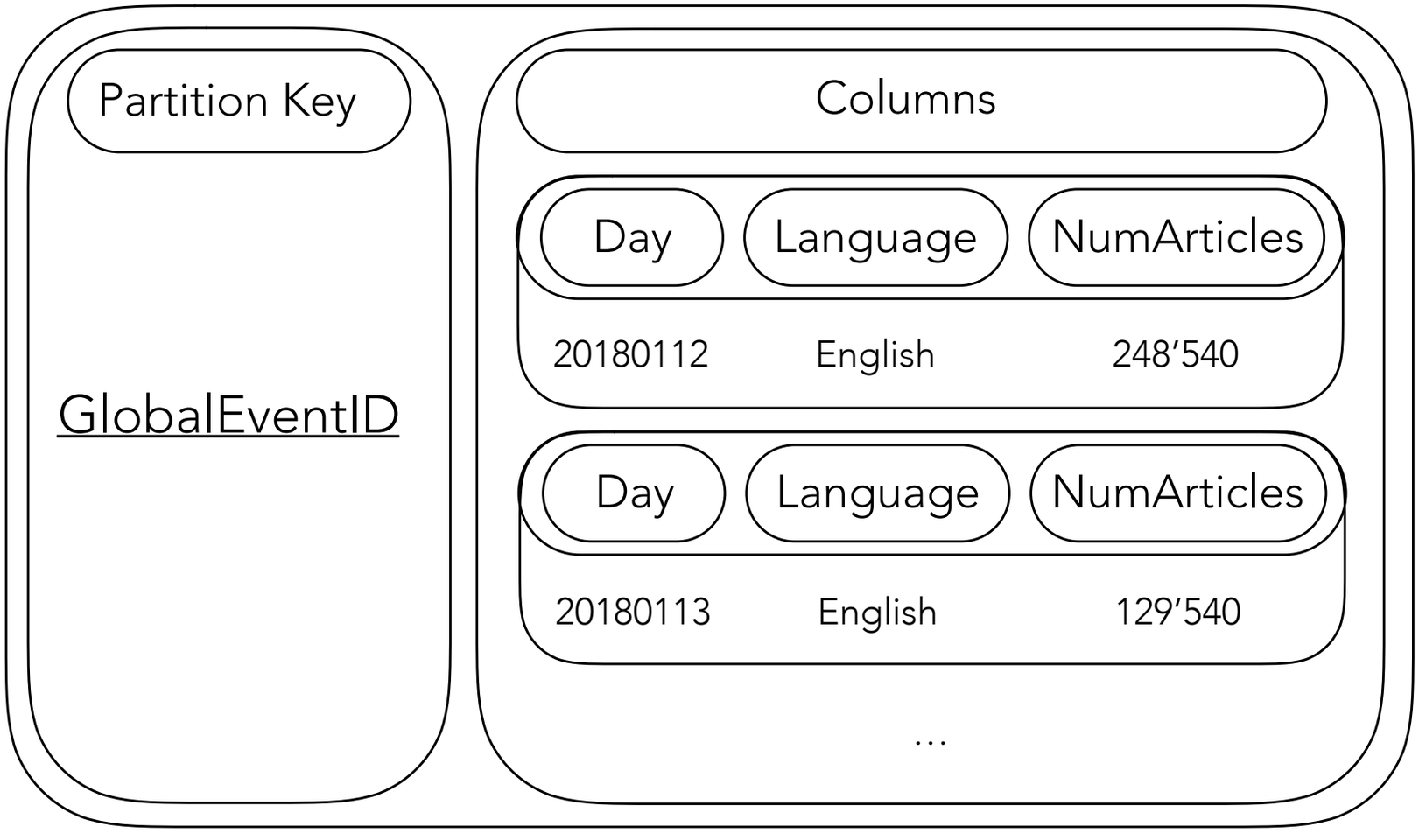


GDELT

APPROCHE

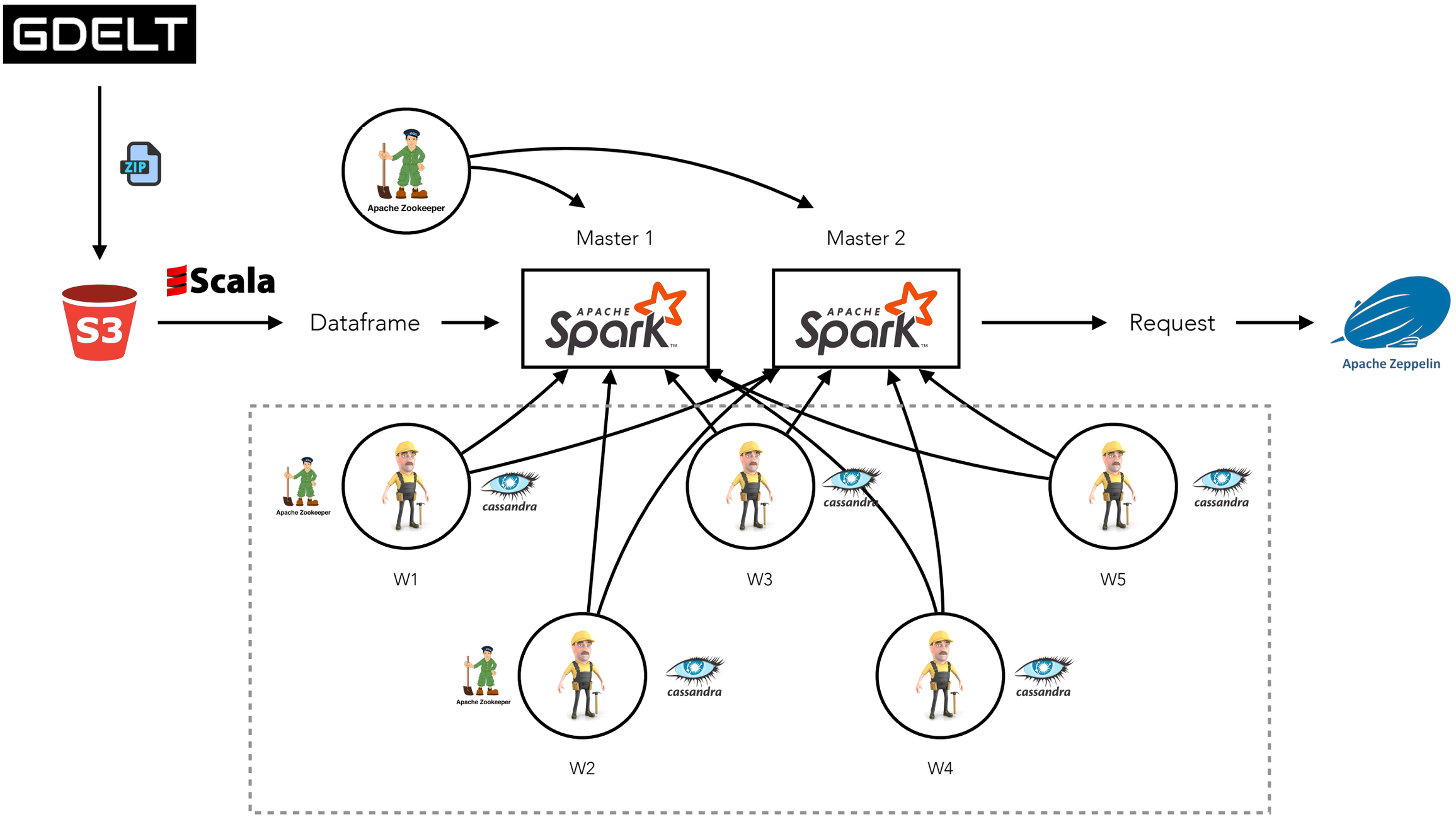
BASE DE DONNÉES

Exemple :

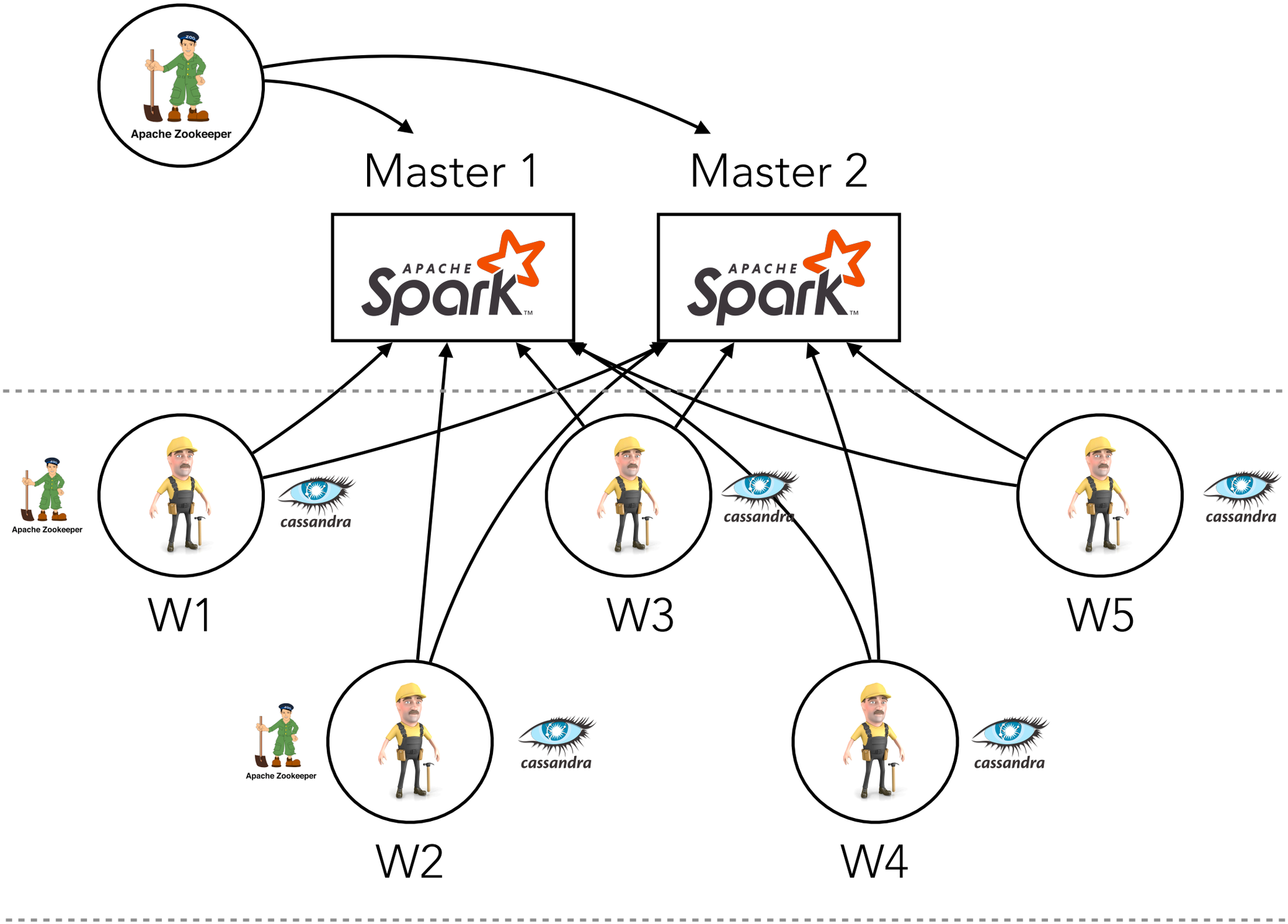


Replication Factor : **3**

ARCHITECTURE



ARCHITECTURE



DATA VISUALIZATION

- Zeppelin `z.show(spark.sql(""" SELECT * FROM q3month ORDER BY SumTone DESC LIMIT 100 """))`
- Helium Zeppelin Leaflet
- Scala + Spark

GDELT

DIFFICULTÉS

DIFFICULTÉS

- Problème avec EC2 la veille de la présentation

```
java.lang.NoSuchMethodError: com.amazonaws.services.s3.transfer.TransferManager.<init>(Lcom/amazonaws/services/s3/AmazonS3;Ljava/util/concurrent/ThreadPoolExecutor;)V
  at org.apache.hadoop.fs.s3a.S3AFileSystem.initialize(S3AFileSystem.java:287)
  at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:2669)
  at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:94)
  at org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:2703)
  at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:2685)
  at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:373)
  at org.apache.hadoop.fs.Path.getFileSystem(Path.java:295)
  at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.setInputPaths(FileInputFormat.java:500)
  at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.setInputPaths(FileInputFormat.java:469)
  at org.apache.spark.SparkContext$$anonfun$binaryFiles$1.apply(SparkContext.scala:921)
  at org.apache.spark.SparkContext$$anonfun$binaryFiles$1.apply(SparkContext.scala:916)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
  at org.apache.spark.SparkContext.withScope(SparkContext.scala:693)
  at org.apache.spark.SparkContext.binaryFiles(SparkContext.scala:916)
  ... 62 elided
```

- Passage sur EMR

- Préparation des données avant passage dans Cassandra à refaire

GDELT

PERFORMANCES

PERFORMANCES

- Requête d’un mois en Spark SQL :

```
z.show(spark.sql(""" SELECT * FROM q3 ORDER BY SumTone ASC LIMIT 100 """))
```

SPARK JOB FINISHED

settings

Language	ActionCountry	Month	Actor1Country	Actor1Code	SumTone
eng	US	201801	US	COP	-1365577.0567040334
eng	US	201801	US	JUD	-1037184.958702697
eng	US	201801	US	GOV	-1029943.3650900614
eng		201801		COP	-696337.9742912925
ara	IS	201801	IS	ISR	-694516.3164537457
eng	IS	201801	IS	ISR	-599651.2468628696
eng	IR	201801	IR	IRN	-476389.1384875841
eng	RS	201801	RS	RUS	-460193.67098307423

Took 10 min 48 sec. Last updated by anonymous at January 23 2019, 10:41:24 AM.

- Requête d’un mois en Cassandra :

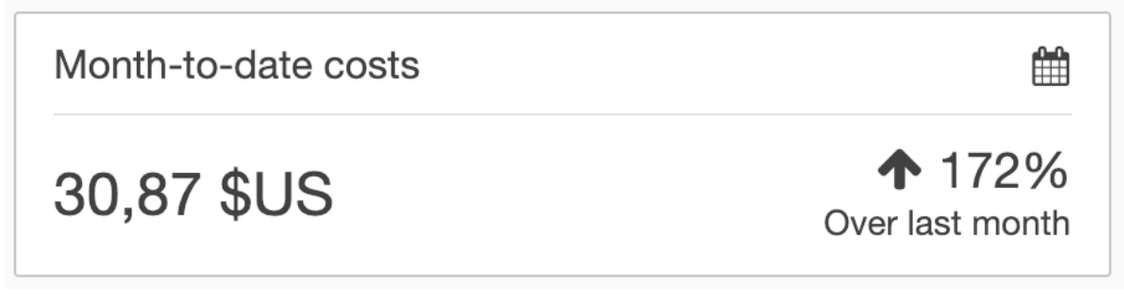
Took 10 min 48 sec. Last updated by anonymous at January 23 2019, 10:41:24 AM.

GDELT

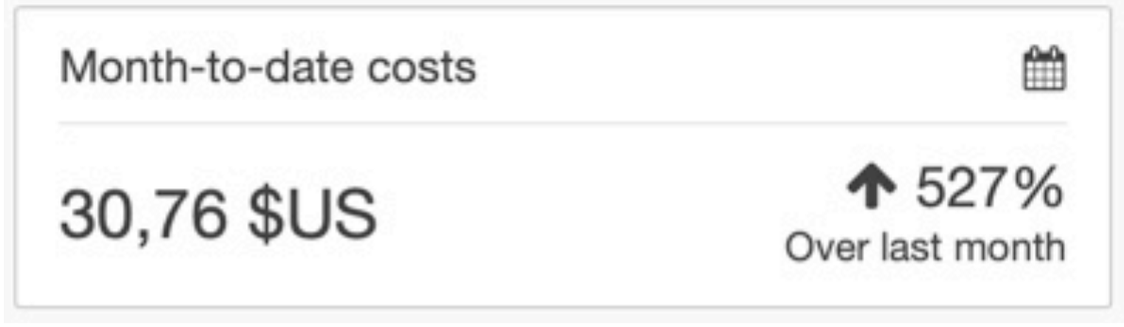
BUDGET

BUDGET

- Charger et stocker les données dans S3, pre-processing vers Cassandra :



- Architecture EC2 avant shift EMR :

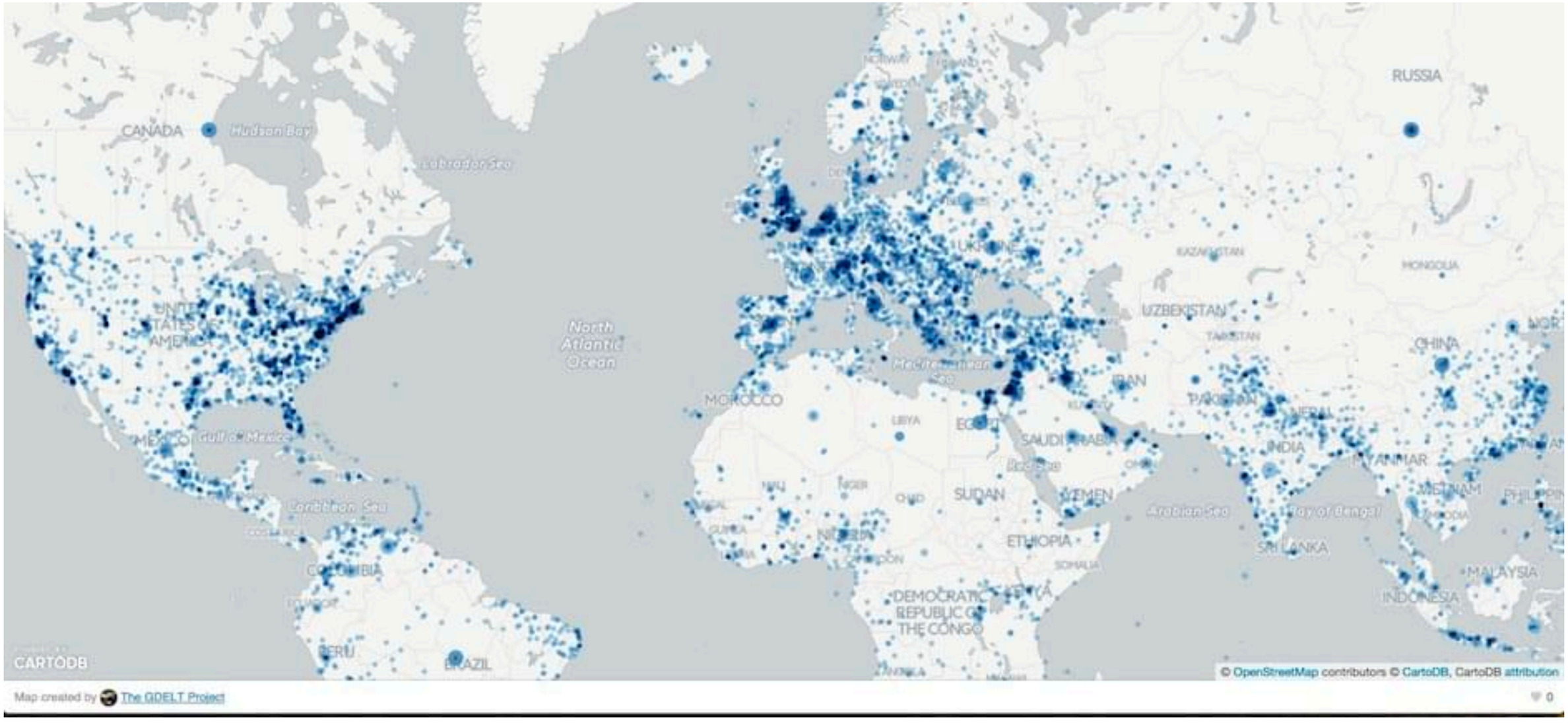


- Architecture EMR :

GDELT

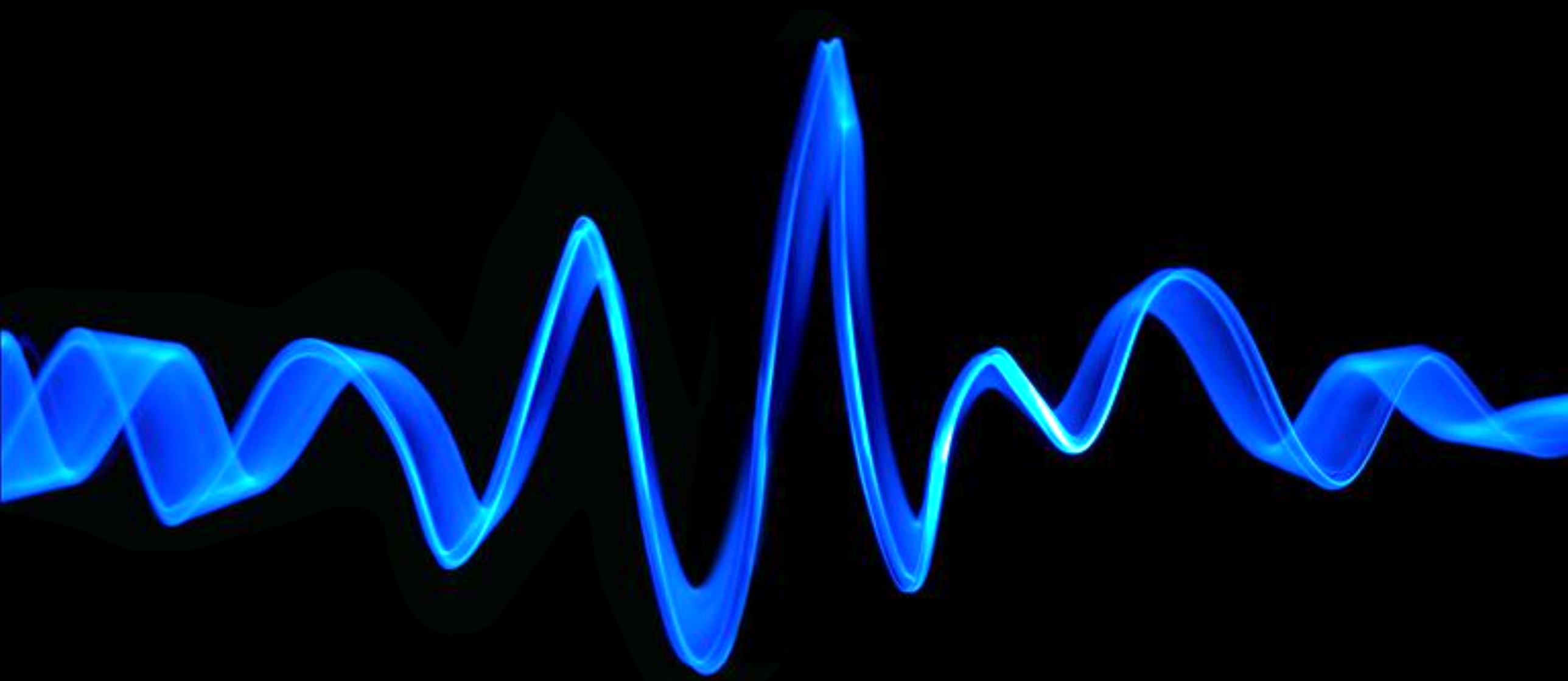
AMÉLIORATIONS

AMÉLIORATIONS POSSIBLES



TELECOM PARISTECH - 2019

GDELT - NOSQL



TELECOM ParisTech