

# Architecture Big Data complète

- Distribuer des algorithmes sur un cluster de calcul de manière optimale
  - Indexer des données avec NoSQL
- Stocker des quantités massives de données hétérogènes
  - Traiter des flux de données en temps réel
- Concevoir une architecture de stockage et de traitement de données distribuée adaptée aux besoins métiers
  - Créer une plateforme distribuée de flux de messages en temps réel

# Sommaire

- 01 • Présentation du **projet**.
- 02 • **Architecture** et **choix techniques**.
  - Contexte, enjeux
  - Architecture.
  - Technologies utilisées.
  - Résultats obtenus.
- 03 • **Scénarios** de gestion des erreurs et des pannes.
- 04 • **Conclusion**.



# 01

## PRESENTATION DU PROJET

# Le contexte

Nous devons déployer une solution complète d'analyse de données pour créer un top 10 des sujets les plus tendance sur Twitter heure par heure.



---

# Le contexte

Nous allons créer un outil qui permet de lister les dix hashtags les plus fréquents pour chaque heure.

---

---

# Le contexte

L'application permettra d'avoir la liste désirée pour l'heure souhaitée sur une durée de 24H.


---

# Projet

## ■ Cahier des charges simplifié

1. Collecte des données
2. Stockage dans des structures adaptées
3. Traitement au coup par coup et en temps réel
4. Solutions pour améliorer la robustesse de l'architecture globale et de chacun de ses composants





# 02

## ARCHITECTURE ET CHOIX TECHNIQUES

---



# Architecture

## ■ API Twitter

Chiffres Twitter 2019:

- Nombre de tweets envoyés par seconde : 6 000 environ (<https://www.blogdumoderateur.com/chiffres-twitter>)

L'API Twitter utilisée nous permet de diffuser environ 1% de tous les nouveaux tweets publics au fur et à mesure qu'ils se produisent (<https://developer.twitter.com/en/docs/labs/sampled-stream/overview>). Ce qui fait environ 60 tweets/seconde.

Le script utilisé a calculé une vélocité de : 84.35 tweet/sec Max. durant l'enregistrement 42.5H (tweets contenant des hashtags).

Les observations lors de l'enregistrement montrent une taille de 1Mo:

- pour env. 47000 hashtags enregistrés (données à analyser).
- Pour environ 3700 tweets enregistrés (données brutes)

13Mo de données à analyser -> 0.30 Mo/H -> 0.085 Ko/sec

580 Mo de données brutes -> 13.6 Mo/H -> 3,8 Ko/sec

Notre application devra analyser 24H de contenu soit environ 335 Mo.

# Architecture

## ■ Proposition répondant au cahier des charges

1. Recueil des hashtags avec un script écrit en Python qui fait appel à l'API Twitter. **[Producer]**
2. Envoi des données au travers de topics Kafka. **Kafka** remplira le rôle de file de messages
3. Le pipeline **Storm** injectera les données brutes dans notre **Data Lake** (Cluster **HDFS**) et enregistrera les données en temps réel dans une base de données **Elasticsearch**
4. Le **Data Lake** fournira les données pour le traitement en batch par une application **Spark**
5. Les données sur Elasticsearch seront indexées de manière à obtenir deux vues distinctes, une vue batch et une vue temps réel
6. Les données de la vue temps réel qui ne sont plus utiles seront supprimées
7. Enfin, une application **Spark** permettra d'obtenir le résultat souhaité en indiquant la date et l'heure voulu. Elle assurera le choix de la vue à utiliser.

## ■ Points traités

Collecte des données ✓

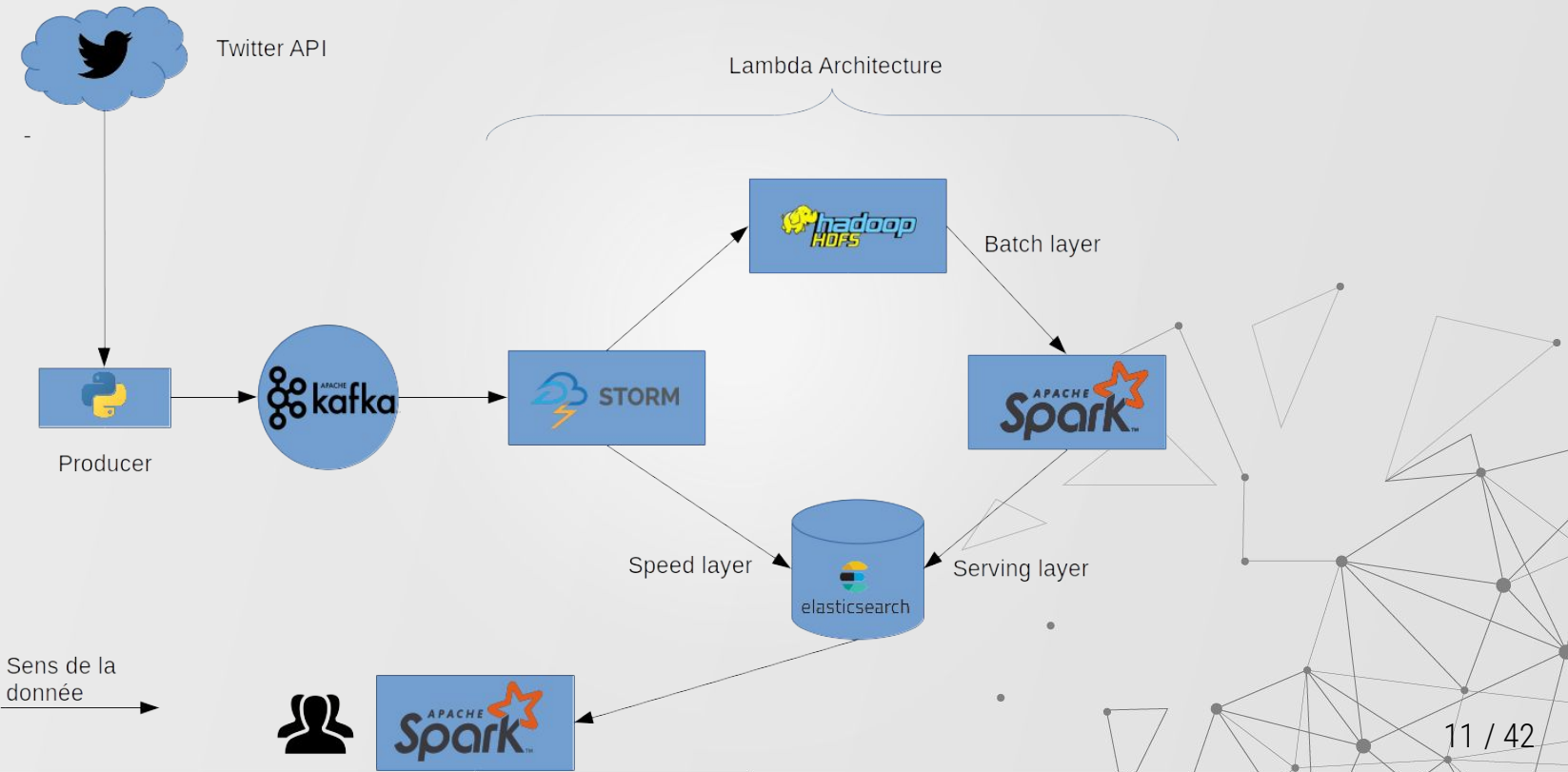
Stockage dans des structures adaptées ✓

Traitement au coup par coup et en temps réel ✓

Solutions pour améliorer la robustesse de l'architecture globale et de chacun de ses composants [Solutions décrites plus tard dans le document]

# Architecture

## Architecture proposée



# Technologies

## SCRIPT PRODUCER



Le script [hashtagsCollect.py](#) permettra de recueillir une partie des tweets (env 1%) en anglais émis sur la plateforme Twitter.

Le script enverra 2 formats de données sur 2 topics différents:

- Un topic "**p5\_raw\_tweet**" qui recevra toutes les données nécessaires de chaque tweet. Ceci permettra de rechercher la données si une erreur d'algorithme, de calcul est faite. Ce sont des données brutes sous format JSON.
- Un topic "**p5\_tweet\_hashtags**" qui récoltera tous les hashtags employés lors de l'enregistrement. Ces données seront analysées en streaming et en batch (depuis notre master dataset).

Le script sera supervisé par **supervisord** permettant un redémarrage en cas d'erreur et donc un débit de données stable.

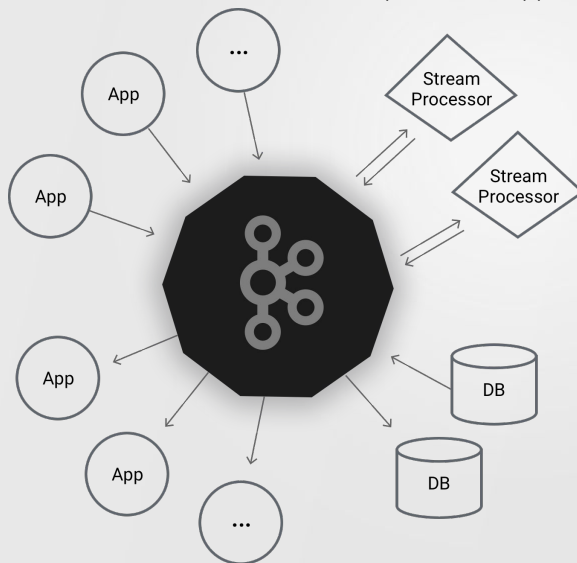
# Technologies

APACHE KAFKA



- **Apache Kafka** est un projet à code source ouvert d'agent de messages développé par l'Apache Software Foundation et écrit en Scala. Il permet de fournir un système unifié, en temps réel à latence faible pour la manipulation de flux de données.

Kafka nous permettra d'obtenir des données de manière fiable pour notre application temps réel.

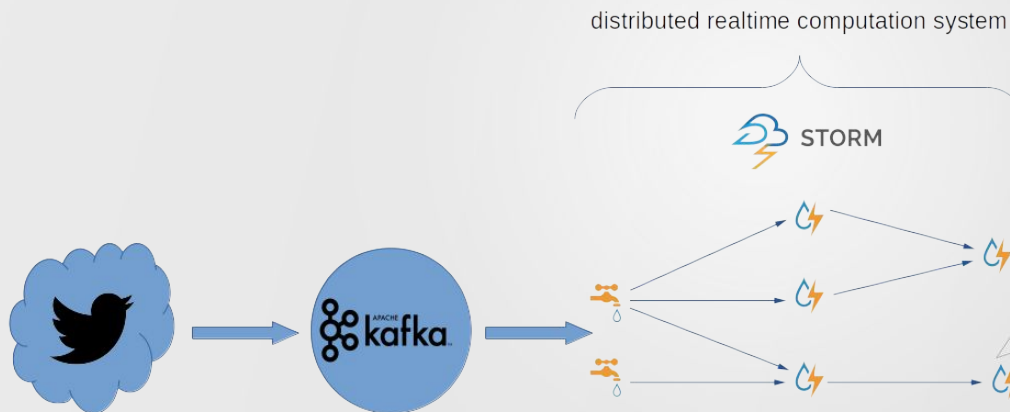


# Technologies

APACHE STORM



- **Apache Storm** est rapide: un benchmark l'a cadencé à plus d'un million de tuples traités par seconde par nœud (site officiel). Il est évolutif, tolérant aux pannes, garantit que les données seront traitées et est facile à configurer et à utiliser. Il s'intègre aux technologies de mise en file d'attente telle que Kafka.



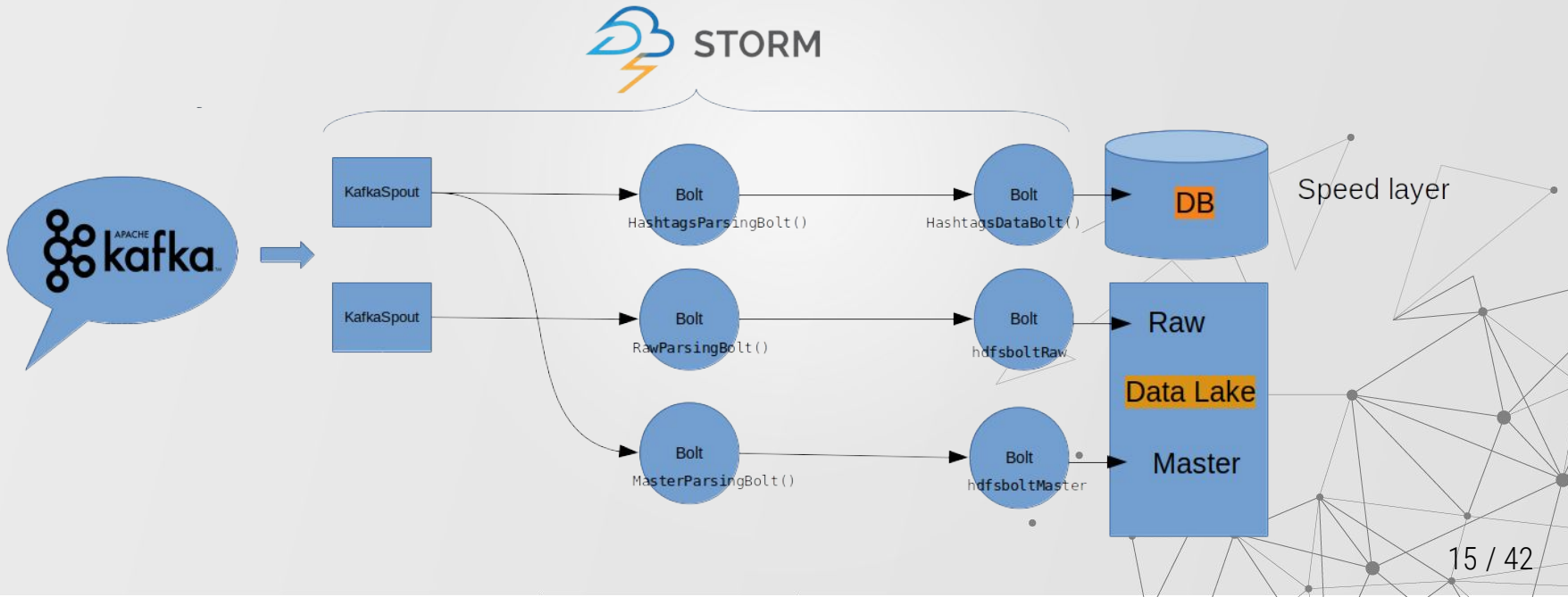
# Technologies

APACHE STORM



■ Topologie

[code](#)

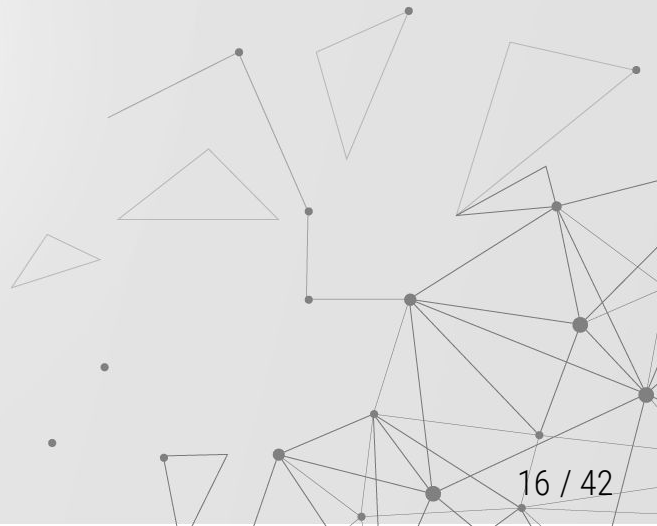


# Technologies

## HDFS

- **Hadoop HDFS** est un système de fichiers distribué sur plusieurs serveurs, et chaque nœud stocke une partie du système fichier. Pour éviter le risque de perdre des données, chaque donnée est stockée à trois emplacements. 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.

Cette architecture de machines HDFS (aussi appelée cluster HDFS) nous permettra de stocker nos données et constituera notre **Data Lake**.





# Technologies

## HDFS



### ■ Organisation des données

**/data/ twitter/**

**raw/**

.snapshot/

[files max 1MB].txt > enregistrement des tweets sous format JSON:

```
{"hashtags":["Win","MonsoonValley","competition"],"user_id":"1706886714","user_name":"Tim  
Holden","id_str":"1208035707996004357","created_at":"Fri Dec 20 14:45:31 +0000 2019","text":"RT @MonsoonValleyUK: #Win 3 bottles of fine  
#MonsoonValley wine! Just RT this tweet and follow @MonsoonValleyUK to enter! #competition T&u2026","user_followers":109}
```

.....

**master/**

**hashtagsOnly/**

.snapshot/

[files max 1MB].txt > enregistrement des hashtags sous format TXT: "timestamp|hashtag"

.....

**full/**

# Technologies

## ELASTIC SEARCH



elasticsearch

- **Elasticsearch** est un serveur utilisant Lucene pour l'indexation et la recherche des données. Nos données seront indexées dans ES en temps réel avec STORM, après analyse avec SPARK.



STORM



# Technologies

## ELASTIC SEARCH



elasticsearch

### ■ Données indexées:

```
{
  "_index": "batch_layer_view",
  "_type": "_doc",
  "_id": "dbmMlW8Br6X9jx6_dMH-",
  "_version": 1,
  "_score": 0,
  "_source": {
    "count": 771,
    "hashtags": "FightBackSana",
    "ts": 1578758400
  }
}
```

```
{
  "_index": "speed_layer_view",
  "_type": "_doc",
  "_id": "vbldlW8Br6X9jx6_WIOV",
  "_version": 1,
  "_score": 0,
  "_source": {
    "speed": {
      "timestamp": 1578755313,
      "hashtags": "OKCThunder"
    }
  }
}
```

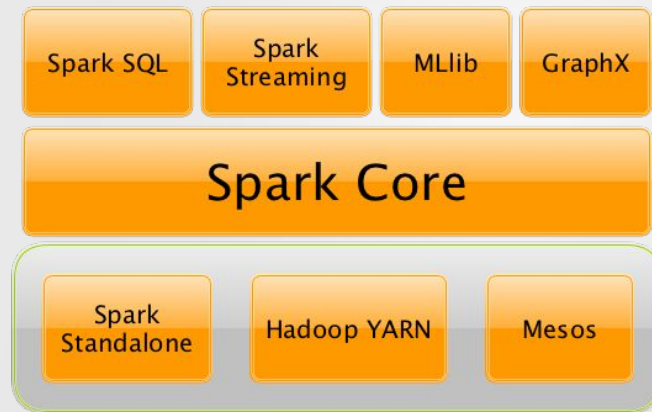
```
1 GET batch_layer_view/_mapping
2 {
3   "batch_layer_view" : {
4     "mappings" : {
5       "properties" : {
6         "count" : {
7           "type" : "long"
8         },
9         "hashtags" : {
10          "type" : "text",
11          "fields" : {
12            "keyword" : {
13              "type" : "keyword",
14              "ignore_above" : 256
15            }
16          }
17        },
18        "ts" : {
19          "type" : "long"
20        }
21      }
22    }
23  }
24 }
```

```
1 GET speed_layer_view/_mapping
2 {
3   "speed_layer_view" : {
4     "mappings" : {
5       "properties" : {
6         "speed" : {
7           "properties" : {
8             "hashtags" : {
9               "type" : "text",
10              "fields" : {
11                "keyword" : {
12                  "type" : "keyword",
13                  "ignore_above" : 256
14                }
15              }
16            },
17            "timestamp" : {
18              "type" : "long"
19            }
20          }
21        }
22      }
23    }
24 }
```

# Technologies



- Apache Spark est un moteur d'analyse unifié pour le traitement de données à grande échelle. Apache Spark atteint des performances élevées pour les traitements en batch et streaming.  
Spark SQL nous permettra d'interroger des données structurées dans nos programmes Spark



# Technologies

## APPLICATION TRAITEMENT BATCH



### ■ Code de l'application de traitement en batch

[script\\_batch.py](#)

Pour ce projet, l'application doit être lancée avec un package (`--packages org.elasticsearch:elasticsearch-hadoop:7.5.0`).

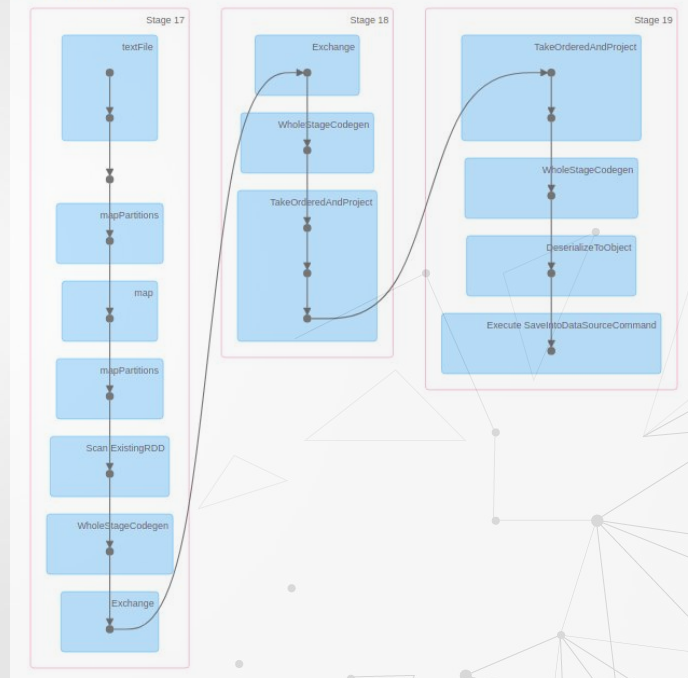
La variable `ts_start` du script doit être définie avec le premier timestamp soit pour ce projet 1578747600.

### Details for Job 9

Status: SUCCEEDED  
Completed Stages: 3

▶ Event Timeline  
▼ DAG Visualization

extrait Spark UI



# Technologies



## ■ Code de l'application et requêtes utilisées

[script\\_spark\\_app.py](#)

Pour ce projet, l'application doit être lancée avec un package (`--packages org.elasticsearch:elasticsearch-hadoop:7.5.0`) et une date comprise entre 2020/01/11-13:00 et 2020/01/12-12:00 avec des pas de 1H. Format à utiliser : "yyyy/mm/dd-HH:00". La date sera convertie en timestamp.

Le choix a été pris de laisser une fenêtre glissante de 2H pour la speedView (configurable mais doit être au minimum > 1H pour prendre en compte le temps de calcul en batch)

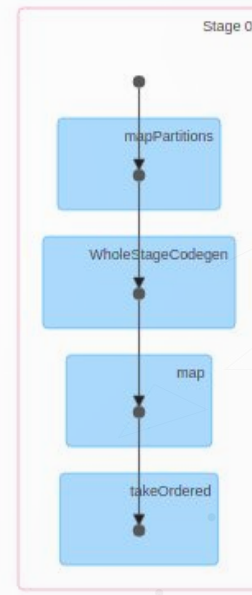
### Details for Job 0

Status: SUCCEEDED

Completed Stages: 1

▶ Event Timeline

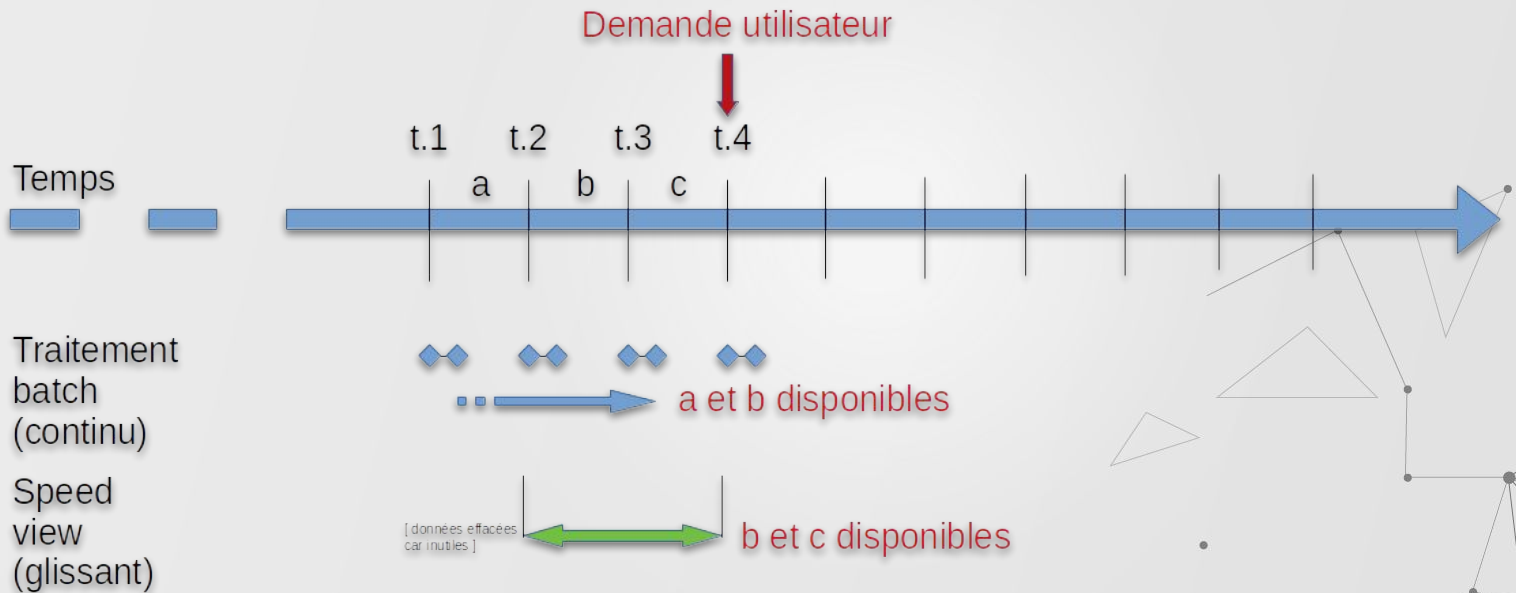
▼ DAG Visualization



# Technologies



## ■ Timeline de gestion des vues et des enregistrements



# Résultats

## APPLICATION UTILISATEUR

### ■ Lancement de l'application en ligne de commande:

~/Spark/spark-2.4.4-bin-hadoop2.7/bin/spark-submit --packages org.elasticsearch:elasticsearch-hadoop:7.5.0  
script\_spark\_app.py **2020/01/11-14:00**

```
batchView
10 most used hashtags calculated in the batchLayer
+-----+-----+-----+
|ts      |hashtags      |count|
+-----+-----+-----+
|1578751200|CONNECT_BTS   |2790 |
|1578751200|ShowStopperAsim|919  |
|1578751200|FightBackSana |877  |
|1578751200|PeoplesChoiceRashami|470 |
|1578751200|iHeartAwards  |416  |
|1578751200|OnlySidMatters|278  |
|1578751200|BestMusicVideo|198  |
|1578751200|GOT7          |166  |
|1578751200|BestFanArmy   |151  |
|1578751200|마크         |140  |
+-----+-----+-----+

took: 0.00231899999999996 seconds
```

La demande ayant un timestamp inférieur à l'ouverture de la speedView, l'application utilisera la vue batch pour obtenir le résultat.



# Résultats

## APPLICATION UTILISATEUR

### ■ Lancement de l'application en ligne de commande:

~/Spark/spark-2.4.4-bin-hadoop2.7/bin/spark-submit --packages org.elasticsearch:elasticsearch-hadoop:7.5.0  
script\_spark\_app.py **2020/01/11-15:00**

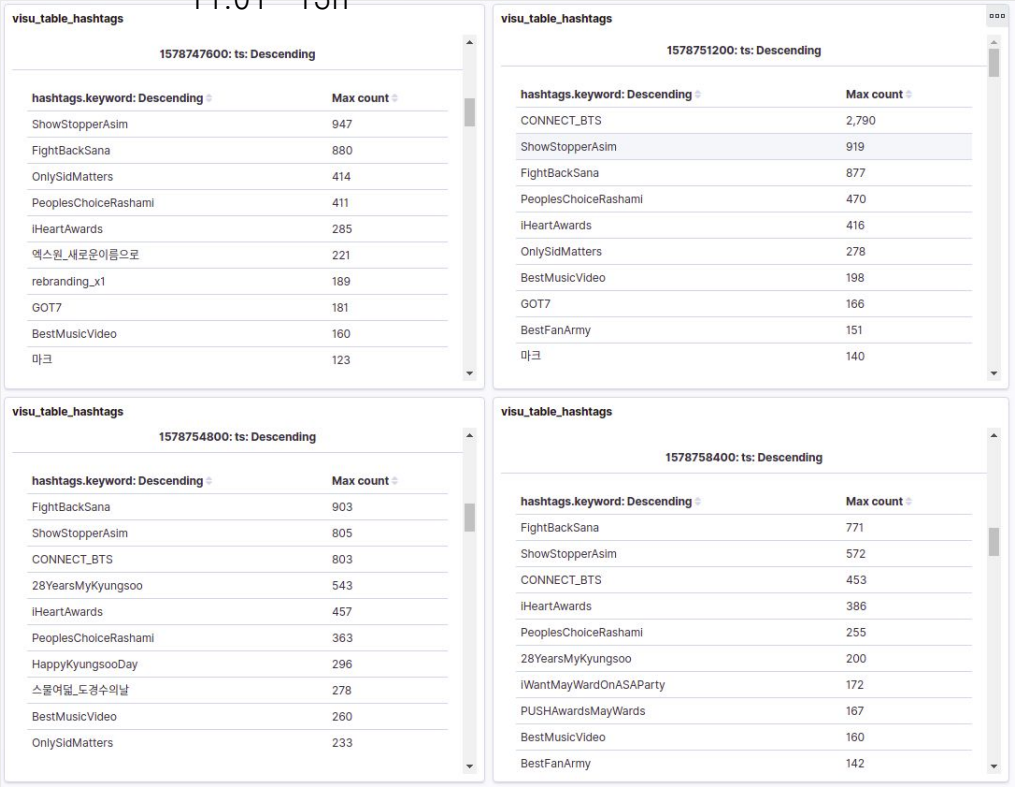
```
speedView
10 most used hashtags calculated in the speedLayer
+-----+-----+-----+
|count|hashtags          |ts          |
+-----+-----+-----+
|907  |FightBackSana     |1578754800|
|810  |ShowStopperAsim   |1578754800|
|804  |CONNECT_BTS       |1578754800|
|543  |28YearsMyKyungsoo |1578754800|
|460  |iHeartAwards       |1578754800|
|364  |PeoplesChoiceRashami |1578754800|
|296  |HappyKyungsooDay   |1578754800|
|278  |스물여덟_도경수의날 |1578754800|
|261  |BestMusicVideo     |1578754800|
|233  |OnlySidMatters     |1578754800|
+-----+-----+-----+
took: 0.004941000000000084 seconds
```

Ici, la demande a un timestamp compris dans l'ouverture de la speedView, l'application l'utilisera pour obtenir le résultat.

# Résultats

## Visualisations

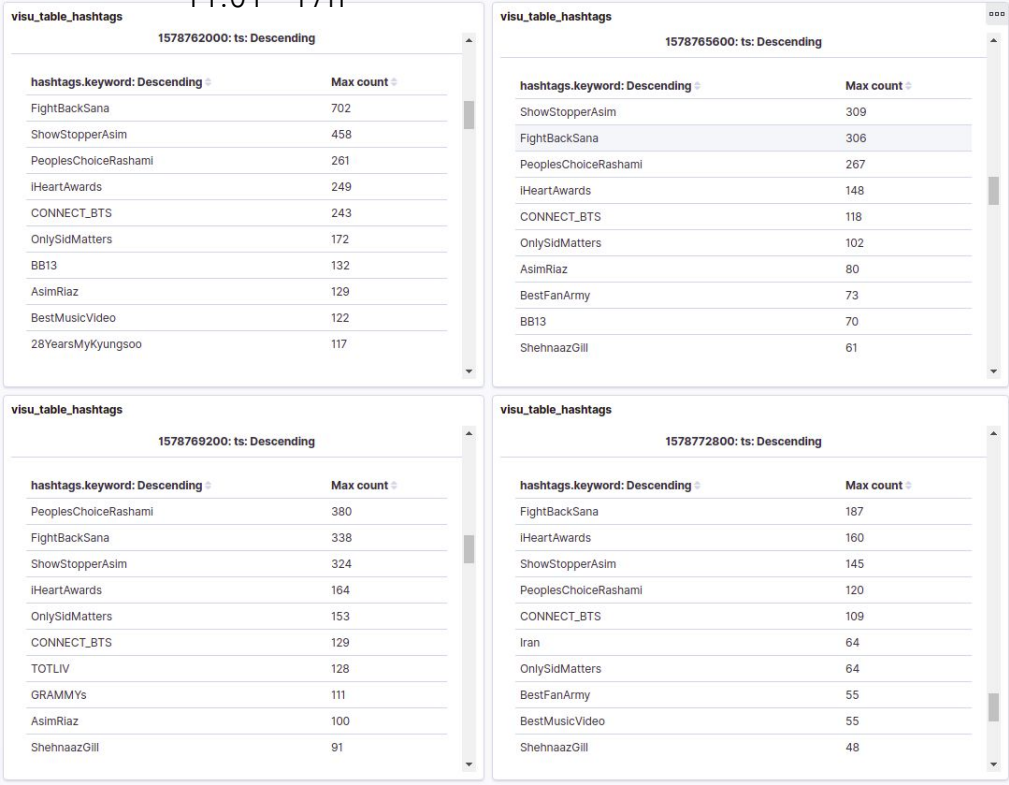
11.01 - 13h



# Résultats

## Visualisations

11.01 - 17h



# Résultats

## Visualisations

11.01 - 21h

<div><div>visu_table_hashtags</div><div>1578776400: ts: Descending</div><table><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr><tr><td>iHeartAwards</td><td>148</td></tr><tr><td>ShowStopperAsim</td><td>126</td></tr><tr><td>PeoplesChoiceRashami</td><td>93</td></tr><tr><td>CONNECT_BTS</td><td>88</td></tr><tr><td>FightBackSana</td><td>85</td></tr><tr><td>Iran</td><td>62</td></tr><tr><td>BREAKING</td><td>56</td></tr><tr><td>BestMusicVideo</td><td>54</td></tr><tr><td>iWantMayWardOnASAParty</td><td>54</td></tr><tr><td>PUSHAwardsMayWards</td><td>49</td></tr></table></div>	hashtags.keyword: Descending	Max count	iHeartAwards	148	ShowStopperAsim	126	PeoplesChoiceRashami	93	CONNECT_BTS	88	FightBackSana	85	Iran	62	BREAKING	56	BestMusicVideo	54	iWantMayWardOnASAParty	54	PUSHAwardsMayWards	49	<div><div>visu_table_hashtags</div><div>1578780000: ts: Descending</div><table><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr><tr><td>iHeartAwards</td><td>230</td></tr><tr><td>CONNECT_BTS</td><td>89</td></tr><tr><td>FightBackSana</td><td>88</td></tr><tr><td>BestMusicVideo</td><td>80</td></tr><tr><td>IDOL600M</td><td>68</td></tr><tr><td>BestLyrics</td><td>65</td></tr><tr><td>Iran</td><td>65</td></tr><tr><td>IranProtests</td><td>63</td></tr><tr><td>ShowStopperAsim</td><td>60</td></tr><tr><td>MINvsSF</td><td>58</td></tr></table></div>	hashtags.keyword: Descending	Max count	iHeartAwards	230	CONNECT_BTS	89	FightBackSana	88	BestMusicVideo	80	IDOL600M	68	BestLyrics	65	Iran	65	IranProtests	63	ShowStopperAsim	60	MINvsSF	58
hashtags.keyword: Descending	Max count																																												
iHeartAwards	148																																												
ShowStopperAsim	126																																												
PeoplesChoiceRashami	93																																												
CONNECT_BTS	88																																												
FightBackSana	85																																												
Iran	62																																												
BREAKING	56																																												
BestMusicVideo	54																																												
iWantMayWardOnASAParty	54																																												
PUSHAwardsMayWards	49																																												
hashtags.keyword: Descending	Max count																																												
iHeartAwards	230																																												
CONNECT_BTS	89																																												
FightBackSana	88																																												
BestMusicVideo	80																																												
IDOL600M	68																																												
BestLyrics	65																																												
Iran	65																																												
IranProtests	63																																												
ShowStopperAsim	60																																												
MINvsSF	58																																												
<div><div>visu_table_hashtags</div><div>1578783600: ts: Descending</div><table><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr><tr><td>iHeartAwards</td><td>273</td></tr><tr><td>BestMusicVideo</td><td>132</td></tr><tr><td>CONNECT_BTS</td><td>107</td></tr><tr><td>iWantMayWardOnASAParty</td><td>90</td></tr><tr><td>FightBackSana</td><td>87</td></tr><tr><td>PUSHAwardsMayWards</td><td>87</td></tr><tr><td>KillThisLove</td><td>74</td></tr><tr><td>BestFanArmy</td><td>72</td></tr><tr><td>BoyWithLuv</td><td>72</td></tr><tr><td>IranProtests</td><td>70</td></tr></table></div>	hashtags.keyword: Descending	Max count	iHeartAwards	273	BestMusicVideo	132	CONNECT_BTS	107	iWantMayWardOnASAParty	90	FightBackSana	87	PUSHAwardsMayWards	87	KillThisLove	74	BestFanArmy	72	BoyWithLuv	72	IranProtests	70	<div><div>visu_table_hashtags</div><div>1578787200: ts: Descending</div><table><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr><tr><td>iHeartAwards</td><td>256</td></tr><tr><td>FightBackSana</td><td>148</td></tr><tr><td>iWantMayWardOnASAParty</td><td>120</td></tr><tr><td>CONNECT_BTS</td><td>113</td></tr><tr><td>BTS</td><td>105</td></tr><tr><td>PUSHAwardsMayWards</td><td>105</td></tr><tr><td>BestMusicVideo</td><td>88</td></tr><tr><td>28YearsMyKyungsoo</td><td>79</td></tr><tr><td>BestFanArmy</td><td>72</td></tr><tr><td>BoyWithLuv</td><td>60</td></tr></table></div>	hashtags.keyword: Descending	Max count	iHeartAwards	256	FightBackSana	148	iWantMayWardOnASAParty	120	CONNECT_BTS	113	BTS	105	PUSHAwardsMayWards	105	BestMusicVideo	88	28YearsMyKyungsoo	79	BestFanArmy	72	BoyWithLuv	60
hashtags.keyword: Descending	Max count																																												
iHeartAwards	273																																												
BestMusicVideo	132																																												
CONNECT_BTS	107																																												
iWantMayWardOnASAParty	90																																												
FightBackSana	87																																												
PUSHAwardsMayWards	87																																												
KillThisLove	74																																												
BestFanArmy	72																																												
BoyWithLuv	72																																												
IranProtests	70																																												
hashtags.keyword: Descending	Max count																																												
iHeartAwards	256																																												
FightBackSana	148																																												
iWantMayWardOnASAParty	120																																												
CONNECT_BTS	113																																												
BTS	105																																												
PUSHAwardsMayWards	105																																												
BestMusicVideo	88																																												
28YearsMyKyungsoo	79																																												
BestFanArmy	72																																												
BoyWithLuv	60																																												

# Résultats

## Visualisations

12.01 - 01h

visu\_table\_hashtags

1578790800: ts: Descending

hashtags.keyword: Descending	Max count
IHeartAwards	239
TENvsBAL	145
FightBackSana	140
IWantMayWardOnASAParty	121
BestMusicVideo	108
PUSHAwardsMayWards	108
CONNECT_BTS	101
28YearsMyKyungsoo	83
Titans	83
BestFanArmy	67

visu\_table\_hashtags

1578794400: ts: Descending

hashtags.keyword: Descending	Max count
IHeartAwards	218
FightBackSana	180
IWantMayWardOnASAParty	131
PUSHAwardsMayWards	127
BestMusicVideo	125
MeatEaters_Are_Killer	125
SarileruNeekevvaru	109
KillThisLove	83
CONNECT_BTS	82
ShowStopperAsim	79

visu\_table\_hashtags

1578798000: ts: Descending

hashtags.keyword: Descending	Max count
TENvsBAL	298
IHeartAwards	231
FightBackSana	186
MeatEaters_Are_Killer	147
BestMusicVideo	143
IWantMayWardOnASAParty	139
PUSHAwardsMayWards	133
Titans	133
SarileruNeekevvaru	116
KillThisLove	97

visu\_table\_hashtags

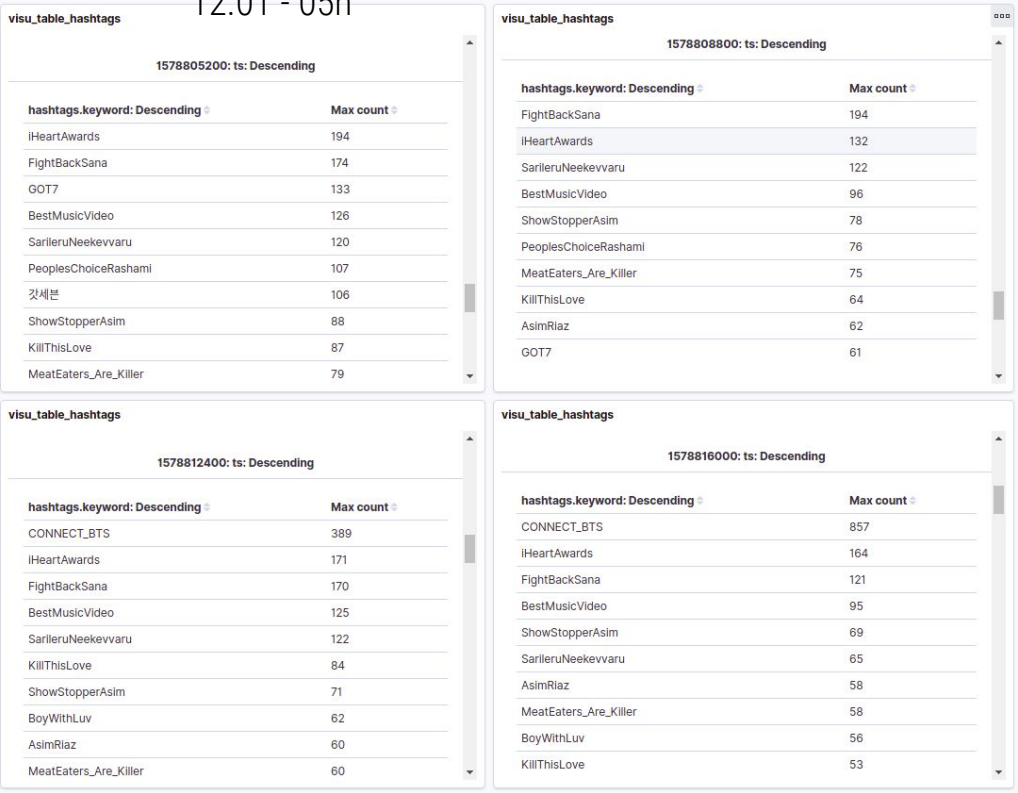
1578801600: ts: Descending

hashtags.keyword: Descending	Max count
IHeartAwards	211
FightBackSana	165
SarileruNeekevvaru	142
Titans	140
BestMusicVideo	135
王嘉爾	121
작은	121
TEAMWANG	120
jacksonwang	120
微博之夜	120

# Résultats

## Visualisations

12.01 - 05h



# Résultats

## Visualisations

12.01 - 09h

<div><div>visu_table_hashtags</div><div>1578819600: ts: Descending</div><table><thead><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr></thead><tbody><tr><td>CONNECT_BTS</td><td>402</td></tr><tr><td>IHeartAwards</td><td>216</td></tr><tr><td>BestMusicVideo</td><td>127</td></tr><tr><td>FightBackSana</td><td>108</td></tr><tr><td>SarileruNeekevvaru</td><td>82</td></tr><tr><td>BoyWithLuv</td><td>80</td></tr><tr><td>ShowStopperAsim</td><td>78</td></tr><tr><td>BestFanArmy</td><td>67</td></tr><tr><td>KillThisLove</td><td>67</td></tr><tr><td>MeatEaters_Are_Killer</td><td>65</td></tr></tbody></table></div>	hashtags.keyword: Descending	Max count	CONNECT_BTS	402	IHeartAwards	216	BestMusicVideo	127	FightBackSana	108	SarileruNeekevvaru	82	BoyWithLuv	80	ShowStopperAsim	78	BestFanArmy	67	KillThisLove	67	MeatEaters_Are_Killer	65	<div><div>visu_table_hashtags</div><div>1578823200: ts: Descending</div><table><thead><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr></thead><tbody><tr><td>CONNECT_BTS</td><td>345</td></tr><tr><td>IHeartAwards</td><td>221</td></tr><tr><td>BestMusicVideo</td><td>146</td></tr><tr><td>FightBackSana</td><td>113</td></tr><tr><td>BoyWithLuv</td><td>92</td></tr><tr><td>KillThisLove</td><td>76</td></tr><tr><td>BestFanArmy</td><td>63</td></tr><tr><td>ShowStopperAsim</td><td>61</td></tr><tr><td>PakistanPartnerOfPeace</td><td>59</td></tr><tr><td>AsimRiaz</td><td>57</td></tr></tbody></table></div>	hashtags.keyword: Descending	Max count	CONNECT_BTS	345	IHeartAwards	221	BestMusicVideo	146	FightBackSana	113	BoyWithLuv	92	KillThisLove	76	BestFanArmy	63	ShowStopperAsim	61	PakistanPartnerOfPeace	59	AsimRiaz	57
hashtags.keyword: Descending	Max count																																												
CONNECT_BTS	402																																												
IHeartAwards	216																																												
BestMusicVideo	127																																												
FightBackSana	108																																												
SarileruNeekevvaru	82																																												
BoyWithLuv	80																																												
ShowStopperAsim	78																																												
BestFanArmy	67																																												
KillThisLove	67																																												
MeatEaters_Are_Killer	65																																												
hashtags.keyword: Descending	Max count																																												
CONNECT_BTS	345																																												
IHeartAwards	221																																												
BestMusicVideo	146																																												
FightBackSana	113																																												
BoyWithLuv	92																																												
KillThisLove	76																																												
BestFanArmy	63																																												
ShowStopperAsim	61																																												
PakistanPartnerOfPeace	59																																												
AsimRiaz	57																																												
<div><div>visu_table_hashtags</div><div>1578826800: ts: Descending</div><table><thead><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr></thead><tbody><tr><td>엑스원 새그룹 기다릴게</td><td>397</td></tr><tr><td>waiting_for_NEWX1</td><td>362</td></tr><tr><td>CONNECT_BTS</td><td>293</td></tr><tr><td>IHeartAwards</td><td>217</td></tr><tr><td>BestMusicVideo</td><td>136</td></tr><tr><td>KillThisLove</td><td>82</td></tr><tr><td>GOT7</td><td>79</td></tr><tr><td>BoyWithLuv</td><td>77</td></tr><tr><td>SarileruNeekevvaru</td><td>77</td></tr><tr><td>FightBackSana</td><td>76</td></tr></tbody></table></div>	hashtags.keyword: Descending	Max count	엑스원 새그룹 기다릴게	397	waiting_for_NEWX1	362	CONNECT_BTS	293	IHeartAwards	217	BestMusicVideo	136	KillThisLove	82	GOT7	79	BoyWithLuv	77	SarileruNeekevvaru	77	FightBackSana	76	<div><div>visu_table_hashtags</div><div>1578834000: ts: Descending</div><table><thead><tr><th>hashtags.keyword: Descending</th><th>Max count</th></tr></thead><tbody><tr><td>IHeartAwards</td><td>487</td></tr><tr><td>BestMusicVideo</td><td>426</td></tr><tr><td>SanaWorldWide</td><td>343</td></tr><tr><td>BoyWithLuv</td><td>327</td></tr><tr><td>Master</td><td>255</td></tr><tr><td>CONNECT_BTS</td><td>243</td></tr><tr><td>엑스원 새그룹 기다릴게</td><td>213</td></tr><tr><td>waiting_for_NEWX1</td><td>195</td></tr><tr><td>KillThisLove</td><td>141</td></tr><tr><td>BoxOfficeBaashaVIJAY</td><td>108</td></tr></tbody></table></div>	hashtags.keyword: Descending	Max count	IHeartAwards	487	BestMusicVideo	426	SanaWorldWide	343	BoyWithLuv	327	Master	255	CONNECT_BTS	243	엑스원 새그룹 기다릴게	213	waiting_for_NEWX1	195	KillThisLove	141	BoxOfficeBaashaVIJAY	108
hashtags.keyword: Descending	Max count																																												
엑스원 새그룹 기다릴게	397																																												
waiting_for_NEWX1	362																																												
CONNECT_BTS	293																																												
IHeartAwards	217																																												
BestMusicVideo	136																																												
KillThisLove	82																																												
GOT7	79																																												
BoyWithLuv	77																																												
SarileruNeekevvaru	77																																												
FightBackSana	76																																												
hashtags.keyword: Descending	Max count																																												
IHeartAwards	487																																												
BestMusicVideo	426																																												
SanaWorldWide	343																																												
BoyWithLuv	327																																												
Master	255																																												
CONNECT_BTS	243																																												
엑스원 새그룹 기다릴게	213																																												
waiting_for_NEWX1	195																																												
KillThisLove	141																																												
BoxOfficeBaashaVIJAY	108																																												

# 03

## SCENARIOS

---



# Scénarios

## Gestion des erreurs / des pannes

### ■ Apache Kafka

3 noeuds seront démarrés utilisant 3 ports différents, ceci afin de supporter la panne d'un ou 2 serveurs simultanément. Les données seront correctement répliquées sur les différents serveurs en indiquant les paramètres lors de la création des topics:

- Replication-factor 3

Pour pouvoir passer à l'échelle, nous allons devoir augmenter le nombre de consumers et donc de partitions de nos topics. Nous utiliserons ce paramètre lors de la création des topics:

- Partitions 10



# Scénarios

## Gestion des erreurs / des pannes

### ■ Apache Storm

La représentation d'une topologie sous la forme d'un DAG permet une certaine tolérance aux pannes : il suffit qu'un des nœuds signale que le traitement d'un tuple a causé un échec pour faire remonter l'erreur au spout parent et éventuellement décider de ré-émettre ce tuple.

Les tâches de traitement seront parallélisées et le passage à l'échelle est possible horizontalement.

Si le système subit une panne, un redémarrage intempestif, les statistiques en cours seraient perdues. Une solution serait de stocker les statistiques récoltées dans une BDD à laquelle tous les Bolts peuvent accéder.



# Scénarios

## Gestion des erreurs / des pannes

### ■ Hadoop HDFS

Les données seront répliquées 3 fois. Le namenode est un point de défaillance unique dans l'architecture HDFS. Pour répondre a cette problématique, un Secondary Namenode sera employé.



# Scénarios

## Gestion des erreurs / des pannes

### ■ Apache Spark

Si l'application est arrêtée ou subit une panne, elle pourra être relancée pour terminer son analyse. Spark est tolérant aux pannes sans réplication. Lorsqu'un nœud du DAG (graphe acyclique orienté) devient indisponible, à cause d'une malfunction quelconque, il peut être régénéré à partir de ses nœuds parents.



# Scénarios

## Gestion des erreurs / des pannes

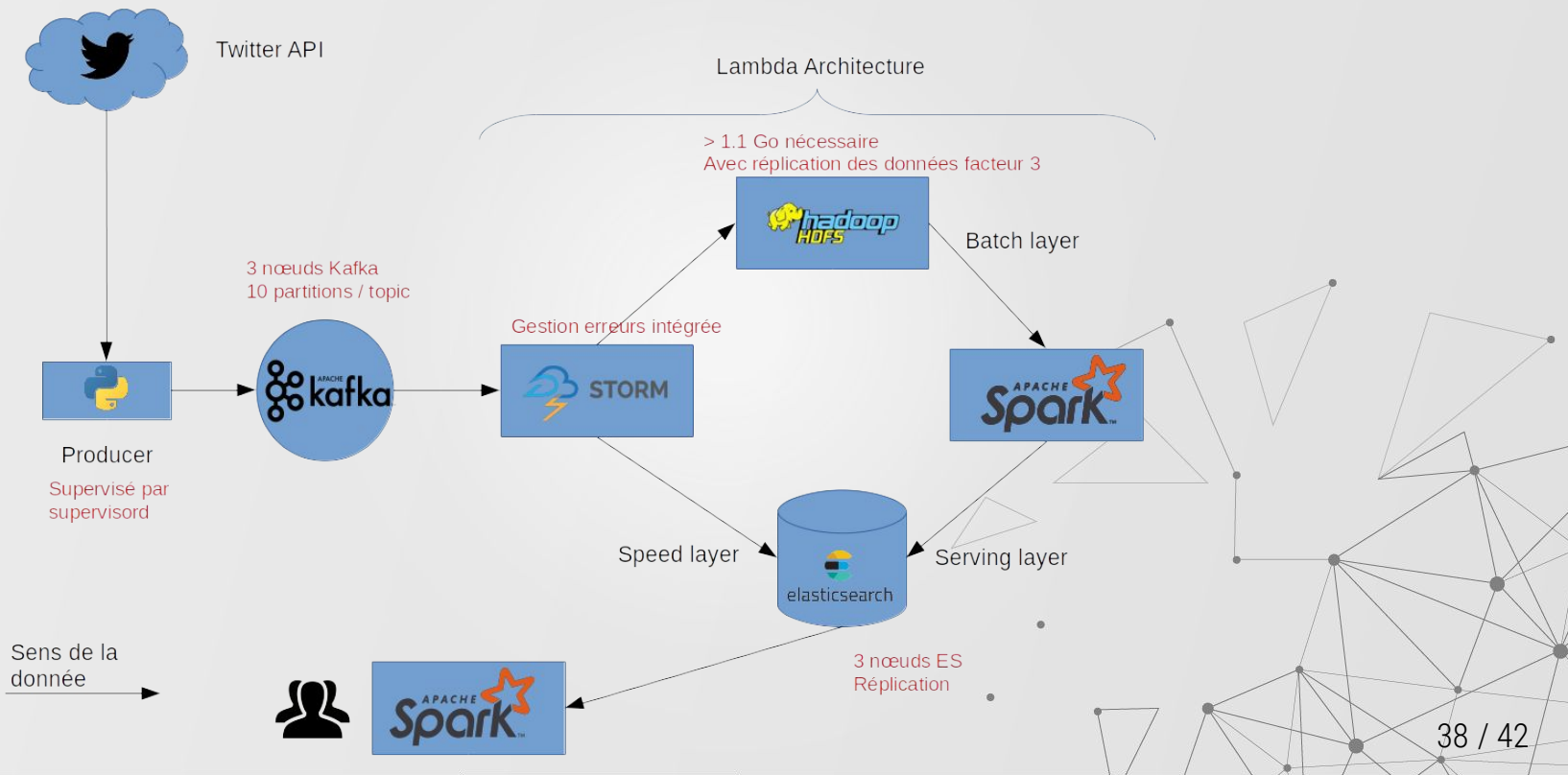
### ■ Elasticsearch

La couche de distribution effectuée par Elasticsearch permet de router les requêtes, paralléliser les traitements et répliquer les données en cas de panne. 3 noeuds seront démarrés pour la tolérance aux pannes. (nota: environ 1.2Go Ram / noeud)



# Scénarios

## Architecture améliorée



# 04

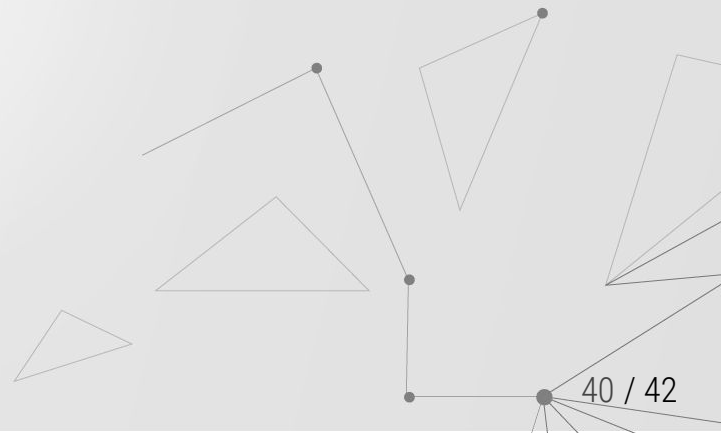
## CONCLUSION

---

# Concevoir une architecture Big Data complète

---

- Nous avons déployé une solution complète d'analyse de données pour créer un top 10 des sujets les plus tendance sur Twitter.





# Ressources

## Web

- <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>
- <https://blog.ippon.fr/2017/03/20/tamtam-bigdata-resilience-a-la-panne-des-systemes-distribues/>
- <https://le-datascientist.fr/aoache-kafka>
- <https://www.lebigdata.fr/hdfs-fonctionnement-avantages>
- <https://fr.slideshare.net/o0m65/file-format-benchmarks-avro-json-orc-parquet>
- <https://docs.microsoft.com/fr-fr/azure/hdinsight/spark/apache-spark-perf>
- <https://stackoverflow.com/questions/40590028/what-do-the-blue-blocks-in-spark-stage-dag-visualisation-ui-mean>
- <https://databricks.com/blog/2015/06/22/understanding-your-spark-application-through-visualization.html>
- <https://storm.apache.org/>
- <https://kafka.apache.org/>
- <https://spark.apache.org/>

Et bien d'autres...



# MERCI

Avez-vous des **questions**?

f2buttet@gmail.com

06.84.19.58.69