BIG DATA CHALLENGE

Groupe Aquila

Jin Liu, Colin Cheoux-Damas, Alexandre Giraud, Tianyuan Liu

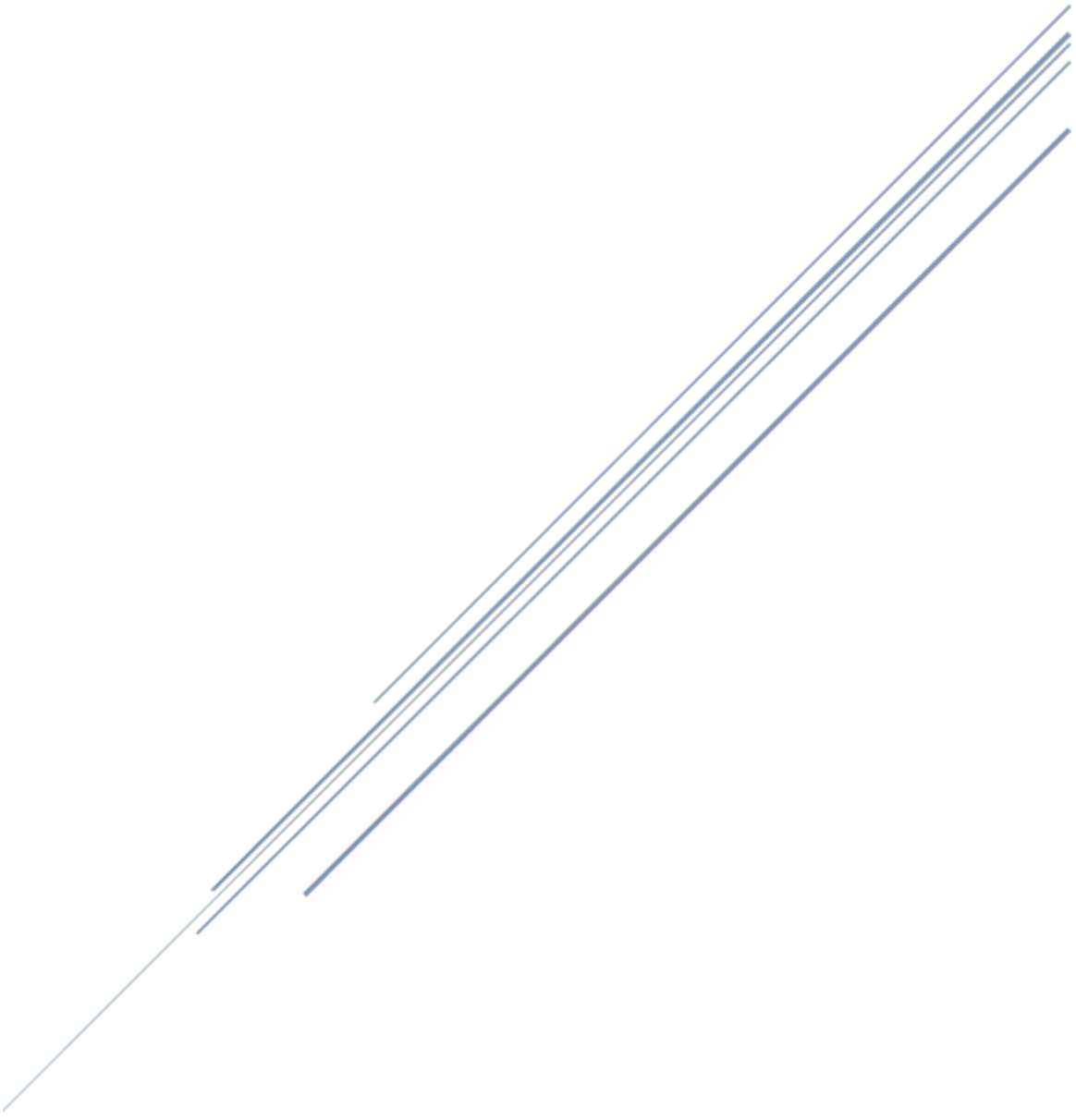


Table des matières

Introduction

Face à l'explosion du volume d'informations, le Big Data vise à proposer une alternative aux solutions traditionnelles de bases de données et d'analyse (serveur SQL, plateforme de Business Intelligence...). Confrontés très tôt à des problématiques de très gros volumes, les géants du web, au premier rang desquels Yahoo (mais aussi Google et Facebook), ont été les premiers à déployer ce type de technologies. Selon le Gartner, le Big Data (en français mégadonnées ou "Grandes données") regroupe une famille d'outils qui répondent à une triple problématiques :
un Volume de données important à traiter, une grande Variété d'informations (en provenance de plusieurs sources, non-structurées, structurées, Opendata...), et un certain niveau de Vélocité à atteindre - c'est-à-dire de fréquence de création, collecte, traitement/analyse et partage de ces données.
Ce challenge va permettre de faire une étude comparative de différents outils utilisés en Big Data.

Description du challenge

a) Contexte

Le challenge à l'origine de ce dossier se repose sur l'analyse des tweets et des utilisateurs de Twitter. La première question à se poser est donc qu'est-ce que Twitter?

Twitter est un service de microblogage ou microblogging, qui permet à ses utilisateurs de bloguer grâce à de courts messages, des « tweets ». Outre cette concision imposée, la principale différence entre Twitter et un blog traditionnel réside dans le fait que Twitter n'invite pas les lecteurs à commenter les messages postés. La promesse d'origine de Twitter, « What are you doing? », le définit comme un service permettant de raconter ce qu'on fait au moment où on le fait.

Twitter compte 313 millions d'utilisateurs actifs par mois avec 500 millions de tweets envoyés par jour et est disponible en plus de 40 langues.

Ces caractéristiques en font une source de données exceptionnelle pour une étude comparative d'outils qui se réclament capable d'exploiter des données issues du Big Data.

L'objectif de ce challenge est donc d'identifier les forces et les faiblesses de certains de ces outils.

Chaque groupe a dû ainsi choisir une solution permettant de réaliser des analyses sur un ensemble de tweets et d'utilisateurs. Cette solution devait être capable de récupérer des données directement d'une API twitter, de les stocker dans une base de données et enfin de réaliser des analyses sur les données stockées.

b) Choix de la solution

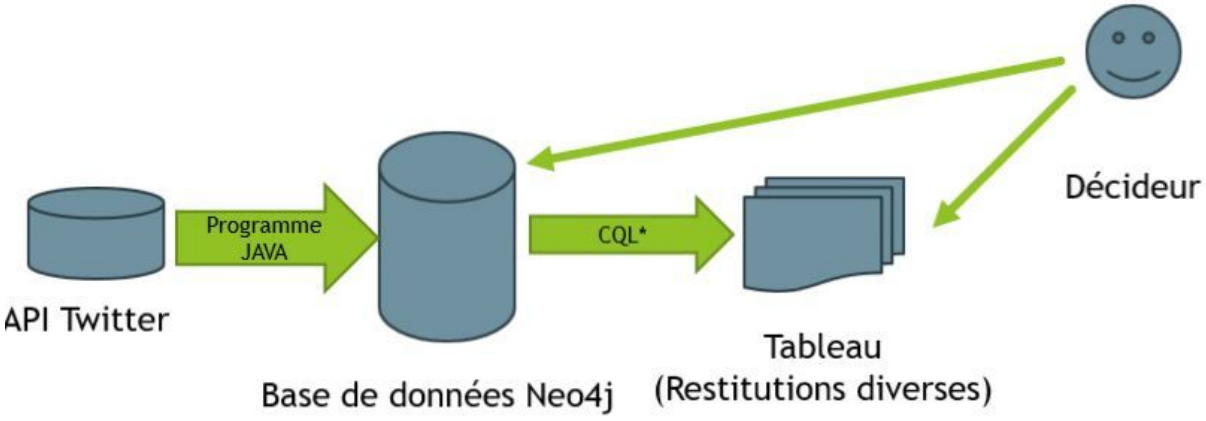
Au début de ce challenge, il a fallu se décider sur le type de base de données à utiliser. Cette base de données devrait être capable de gérer une grande quantité de données et devrait être adaptée à une analyse de Twitter.
Un choix s'est donc naturellement détachée. En effet, les bases de données graphes sont particulièrement compatible avec les réseaux sociaux.

La solution choisie ici est donc centrée autour du SGBD neo4j: un SGBD qui respecte les propriétés ACID habituellement présentes dans les bases de données relationnelles. Ce qui veut dire qu'elle intègre les notions d'atomicité (les transactions se font dans leur intégralité ou pas du tout), de cohérence (chaque transaction fait passer le système d'un état valide à un autre état valide), d'isolation (toute transaction s'exécute indépendamment des autres) et de durabilité (une fois une transaction terminée, elle demeure enregistrée). Les données peuvent être stockées de différentes façons ainsi Neo4j peut stocker des données sous forme de noeuds et de liens. Chaque noeud possédant des attributs.
De plus, Neo4j met à la disposition de l'utilisateur un langage de requêtage puissant: les requêtes CYPHER. Ces requêtes permettent de faire de nombreuses analyses basées sur des sous graphes ou des tableaux.
Deux plugins sont aussi installable : "Graph Algorithm" qui répertorie de nombreux algorithmes applicables aux graphes comme l'algorithme de plus court chemin ou des algorithme de calcul de degré de centralité et "APOC" qui contient une liste de procédures permettant d'implémenter certaines fonctionnalités qu'il serait difficile d'implémenter seulement avec des requêtes CYPHER.

c) Implémentation de la solution

La première phase d'implémentation de la solution, fut le remplissage de la base de données.

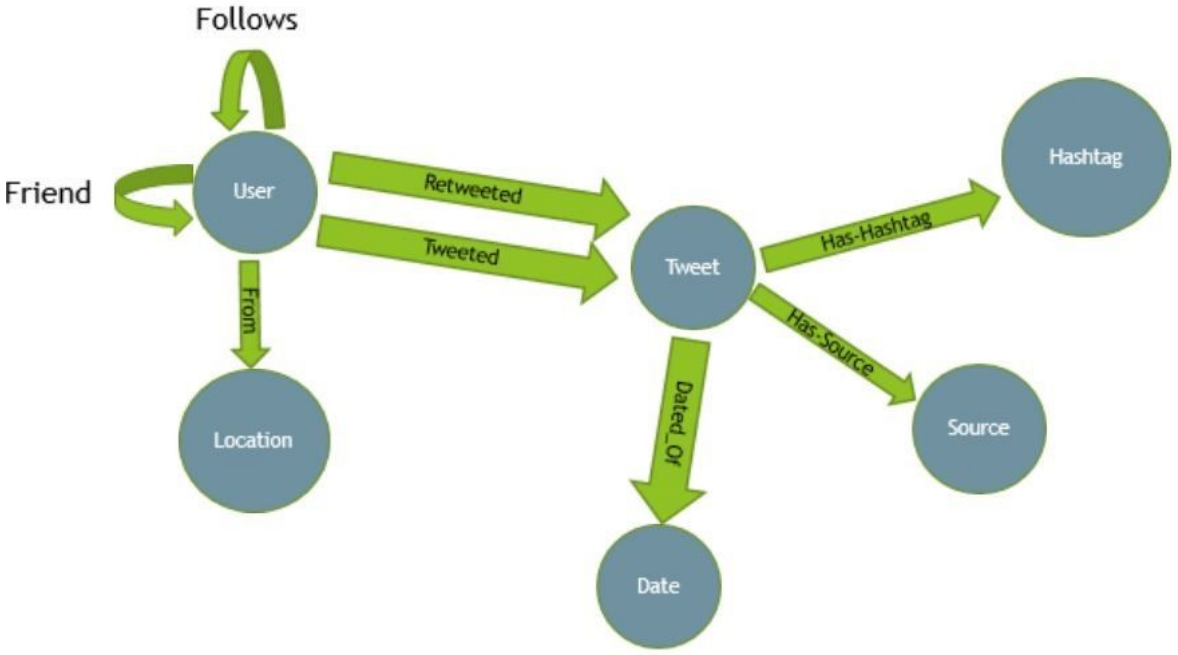
Pour cela, il a fallu extraire des données directement de l'API Twitter à l'aide d'un programme JAVA écrit sur l'IDE NetBeans puis les charger dans la base de donnée dans le format voulu. Tableau se greffe ensuite sur la bases de données, donnant l'architecture suivante:



* CQL: CYPHER Query Language

1. Processus ETL

Avant de se lancer à la découverte des processus ETL utilisés il est nécessaire de se pencher sur l'architecture qui a été retenue pour l'établissement de la base de donnée. Cette architecture a été représentée dans le schéma ci-dessous:



En premier lieu, le programme cible un compte, pour ce challenge, le compte choisi est celui d'Emmanuel Macron. Ensuite, le programme extrait des données dans un ordre précis en partant du compte originel et en se propageant aux amis de celui-ci:

Les premières données extraites sont les données concernant les utilisateurs: leur identifiant unique, leur nom, leur description, leur nombre d'amis, leur nombre de followers, leur localisation, leur nombre de favoris ainsi que le statut de vérification du compte (est ce que l'identité de l'utilisateur a été vérifiée).

Pour chaque utilisateur un certain nombre de données concernant les tweets est récupéré: leur identifiant, l'identifiant de l'utilisateur qui les a émis, leur contenu, la date d'émission du tweet, le nombre de fois que chaque tweet a été retweeté, le nombre de favoris, ainsi que la latitude et la longitude de l'utilisateur au moment de l'émission si l'utilisateur a laissé la géolocalisation lors de l'émission du tweet.

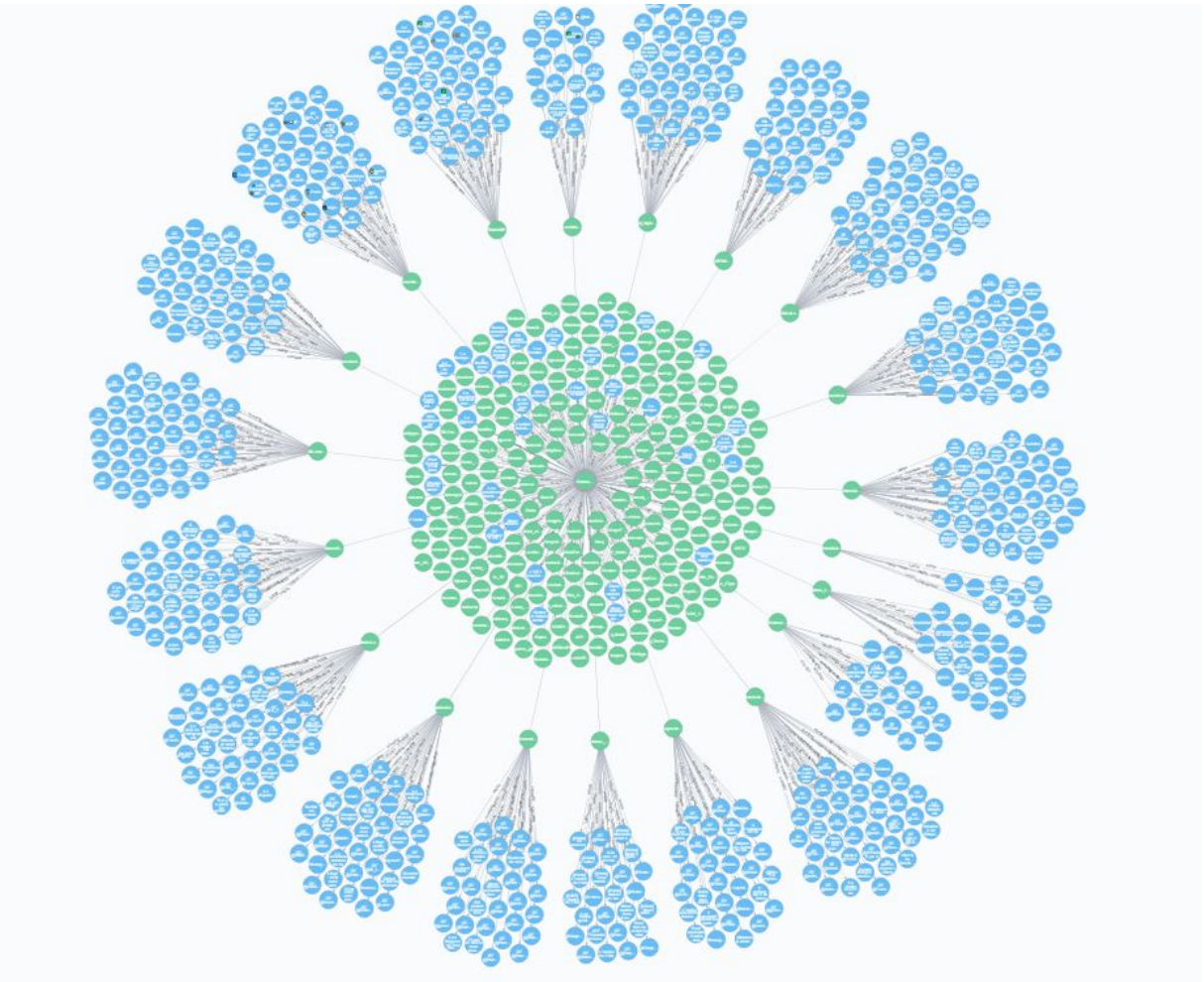
L'ensemble des hashtags rattachés à ces tweets est enfin récupéré.

Pour minimiser le temps d'exécution du programme, il a été décidé de fusionner les noeuds identiques au fur et à mesure de l'exécution du programme JAVA. Cela permet de soulager grandement la machine lors du chargement des données dans la base en réduisant le volume de données présent dans la base.

Un ensemble d'analyse a été réalisé en s'appuyant sur la base de données neo4j. Les premières analyses ont été effectuées directement sur le navigateur neo4j qui permet une bonne visualisation de graphe et de tableaux à l'aide de requêtes CYPHER. Il est nécessaire de préciser que la liste d'analyses possibles est infinie: il suffit d'avoir des requêtes correspondant à ces analyses.

2. Analyses réalisées sur neo4j

Certaines analyses ont été effectuées directement sur le navigateur neo4j qui permet une visualisation de la base de données sous forme de graphes. Un exemple de sous graphe a été fourni ci-après: il représente le réseau d'Emmanuel Macron en France (parmis les utilisateurs qui ont bien déclaré leur localisation en France).



Les tweets sont symbolisés par des disques bleus et les utilisateurs par des disques verts.

La requête CYPHER associée à ce graphe est la suivante:

```
MATCH (u:User)--(l:Location) WHERE toLower(l.Location) =~ '.*france.*' WITH collect(u) as users
MATCH (us:User)--(t:Tweet) WHERE us in users RETURN *
```

3. Analyses réalisées à l'aide du logiciel tableau

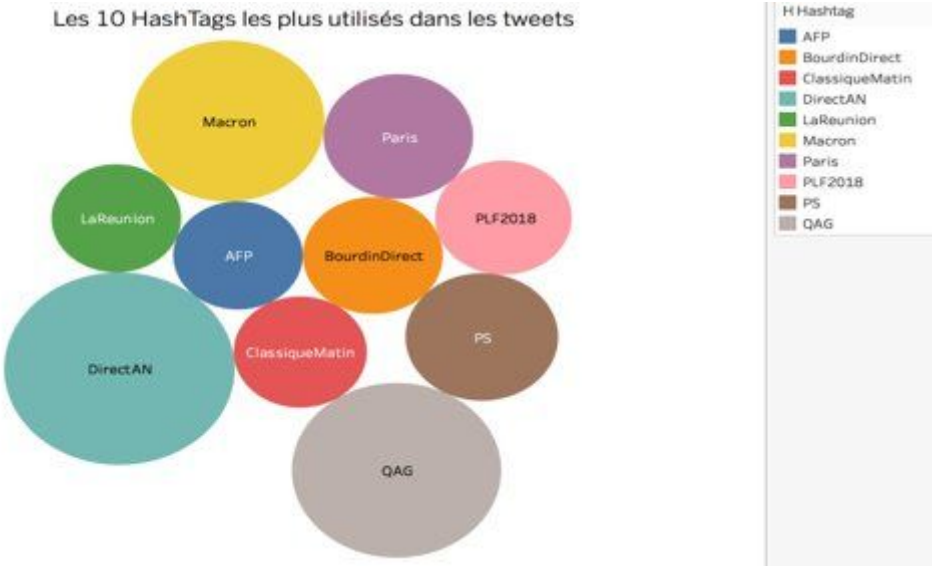
Le logiciel tableau a aussi été connecté à la base de données neo4j dans l'optique de faire des représentations supplémentaires. Le résultat est récupéré dans la base de données à l'aide d'une requête CYPHER pour chaque restitution, il suffit ensuite de choisir la forme de la restitution. Douze restitutions ont été réalisées sur ce logiciel dans le cadre de ce challenge:

a) Les 10 HashTags les plus utilisés dans les tweets:

La requête suivante a été utilisée:

```
MATCH (t:Tweet)-[:HAS_HASHTAG]->(h:Hashtag)
RETURN h,count(h) AS  nb ORDER BY nb DESC LIMIT 10
```

Le résultat observé dans la base de données fut le suivant:



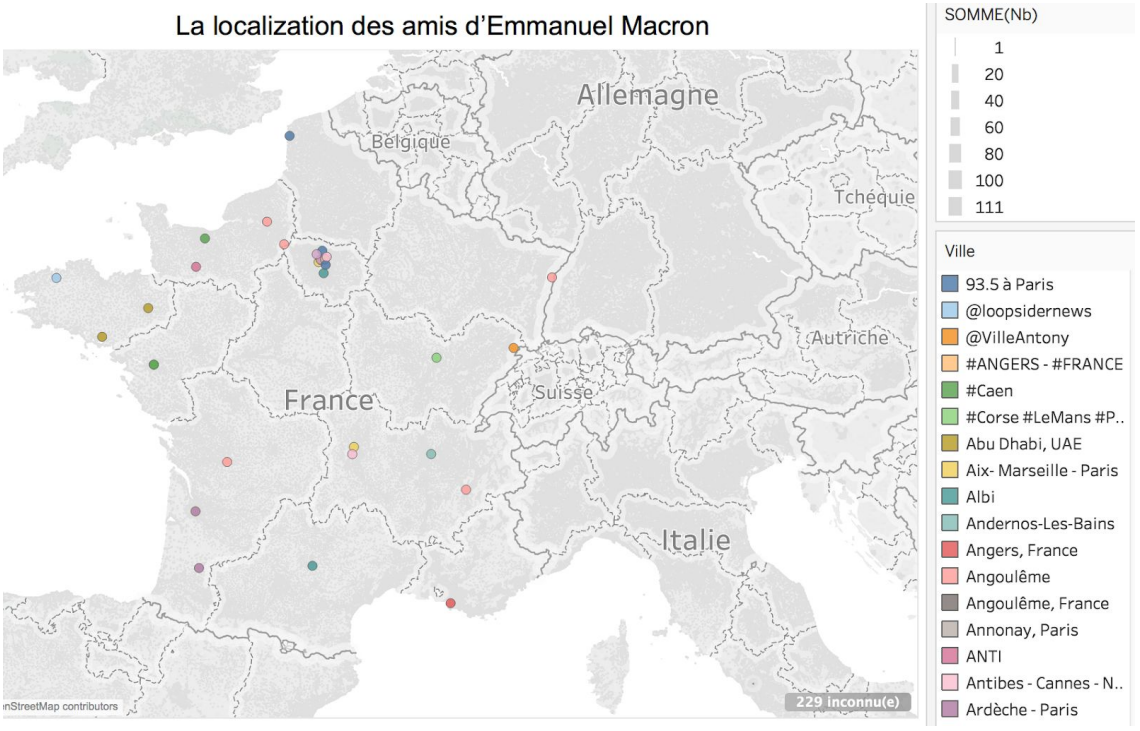
b) La localisation des amis d'Emmanuel Macron

La requête suivante a été utilisée:

```
MATCH (:User{Username:'EmmanuelMacron'})-[:FRIEND]-(ff:User)-[:FROM]-(loc:Location) WHERE loc.Location <>""  RETURN DISTINCT loc.Location AS Ville
```

Pour cette requête une restitution sous forme de carte était la plus judicieuse.

Le résultat observé dans la base de données fut le suivant:



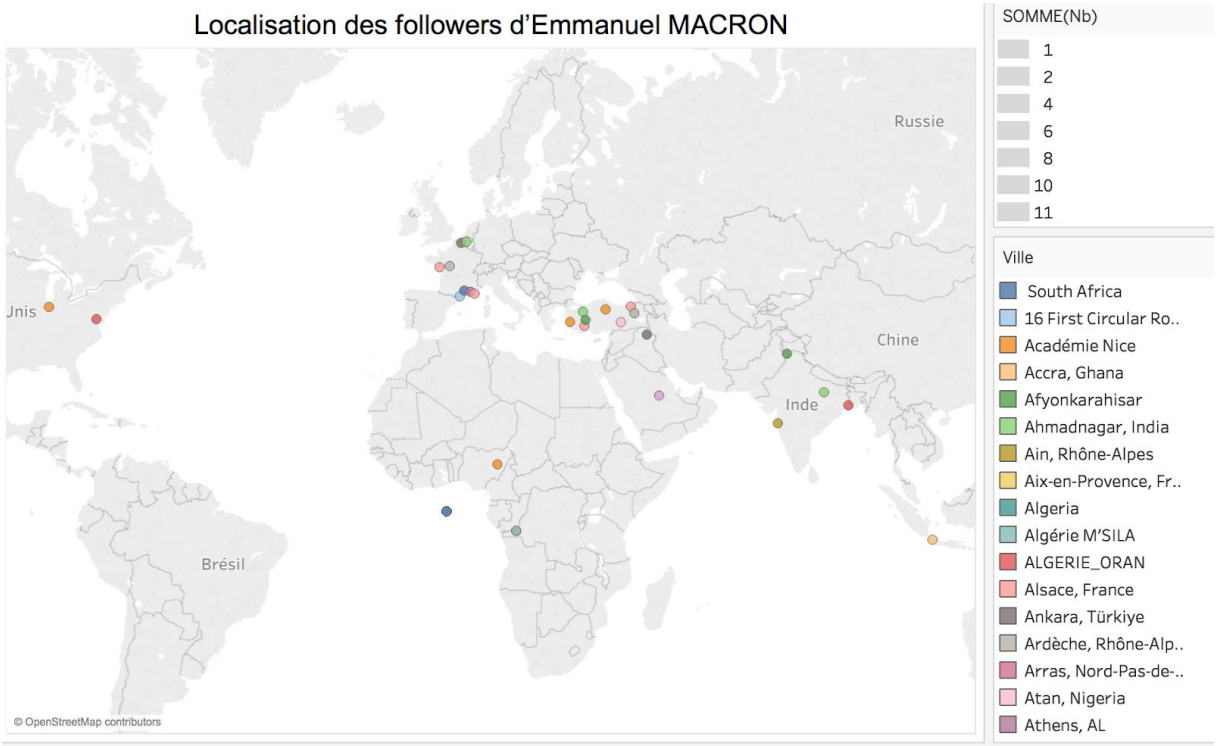
c) La localisation des followers d'Emmanuel Macron

La requête suivante a été utilisée:

```
MATCH (u1:User{Username:'EmmanuelMacron'})<-[:FOLLOWS]-(u2:User)-[:FROM]->(loc:Location)
WHERE loc.Location <>''
RETURN DISTINCT loc.Location AS ville,count(loc.Location) AS nb
```

Pour cette requête une restitution sous forme de carte était la plus judicieuse.

Le résultat observé dans la base de données fut le suivant:

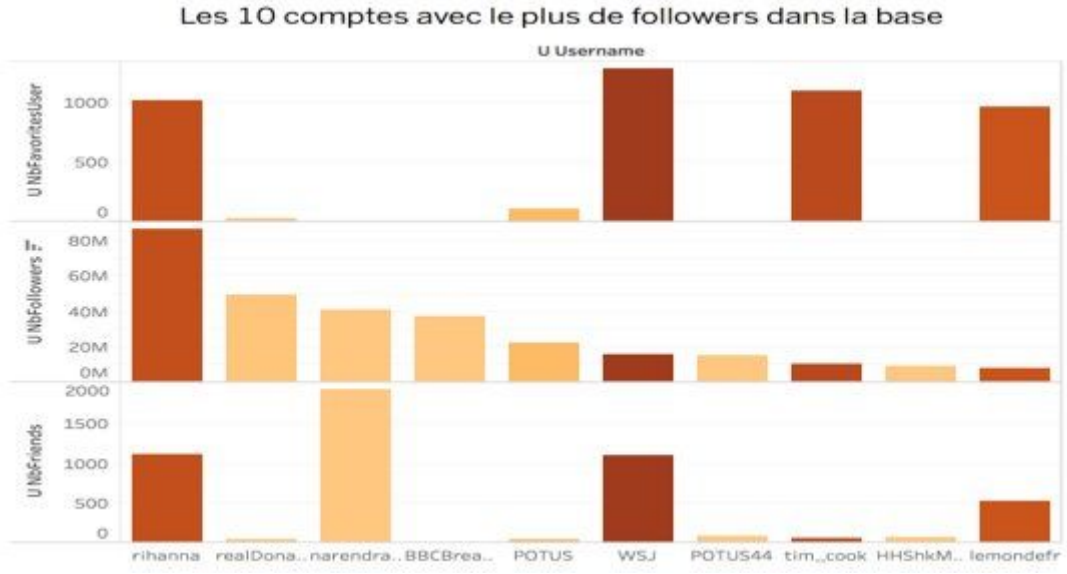


d) Les 10 comptes avec le plus de followers dans la base

La requête suivante a été utilisée:

```
MATCH (u:User)
RETURN u ORDER BY toInteger(u.NbFollowers) DESC LIMIT 10
```

Le résultat observé dans la base de données fut le suivant:



e) Les 10 comptes avec le plus d'amis dans la base

La requête suivante a été utilisée:

MATCH (u:User)
RETURN u ORDER BY toInteger(u.NbFriends) DESC LIMIT 10

Le résultat observé dans la base de données fut le suivant:

f) Les 10 tweets les plus retweetés

La requête suivante a été utilisée:

MATCH (u:User)-[:TWEETED]-(t:Tweet)
RETURN u.Username,t.TweetContent ORDER BY t.NbRetweets DESC LIMIT 10

Le résultat observé dans la base de données fut le suivant:

Les 10 tweets lee plus retweetés

U Username	T TweetContent
ChTaubira	"Les instincts ne doivent pas être nos maîtres à penser" #BondID
equipedefrance	Nouveau changement pour nos Bleus ! @LacazetteAlex remplace @_OlivierGiroud_ #FRAPDG https://..
	Victoire pour l'Equipe de France !! 2-0 ! Les buteurs ce soir : @AntoGriezmann et @_OlivierGiroud_ #F..
EtatMajorFR	Les moyens arrivés avec le BPC Tonnerre sont déployés à Saint-Martin, aux côtés de la population. htt..
franceinter	.@JDoreofficiel : "J'ai passé plusieurs mois à habiller mes chansons, je réapprends à les aimer nues" ..
gerardcollomb	⚠ Nouvelle tempête hivernale : #Eleanor va aborder notre pays dès cette nuit : 21 départements plac..
partisocialiste	[CP] #Autriche : le #PS condamne la coalition gouvernementale droite extrême-droite..
plantu	La ville de Nice et ses enfants. Le dessin du Monde de ce samedi 16 juillet 2016. https://t.co/xcirYFHvs7
	Y'EN A MARRE DE TRUMP ! : Aux États-Unis, Trump propose un "filtrage extrême" des immigrants au..
socialistesAN	Communiqué du groupe Nouvelle Gauche - Le Gouvernement met à sac la politique du logement - 03/1..

g) Les 5 comptes avec le plus de favoris

La requête suivante a été utilisée:

MATCH (t:User)
RETURN t ORDER BY t.NbFavoritesUser DESC LIMIT 5

Le résultat observé dans la base de données fut le suivant:

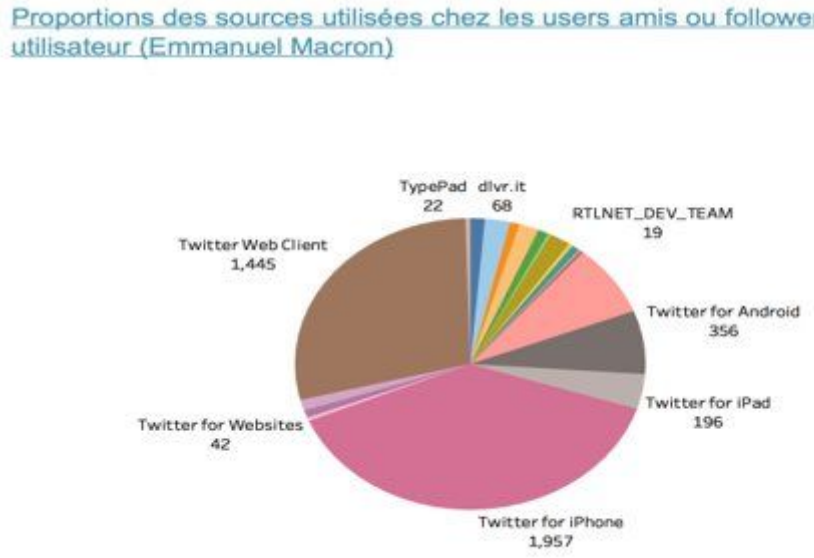
h) Proportions des sources utilisées par les amis ou followers d'Emmanuel Macron

La requête suivante a été utilisée:

```
MATCH
(s:Source)-[:HAS_SOURCE]-(t:Tweet)<-[:TWEETED]-(:User)-[:FRIEND|:FOLLOWS]-(:User{Username:'EmmanuelMacron'})
RETURN s,count(s)  AS Nb ORDER BY Nb DESC LIMIT 20
```

Cette analyse reposant sur des proportions, l'utilisation d'un diagramme en secteur a été retenue.

Le résultat observé dans la base de données fut le suivant:



i) L'évolution de l'utilisation d'un hashtag "DirectAN" en 2017

La requête suivante a été utilisée:

```
MATCH (h:Hashtag{Hashtag:"DirectAN"})<-[:HAS_HASHTAG]-
(t:Tweet)-[:DATED_OF]->(date:Date{Year:"2017"})
RETURN date,count(h) AS Nb
```

Ici, le suivi de l'évolution de l'utilisation d'un Hashtag au cours du temps nécessite l'utilisation d'une courbe.

Le résultat observé dans la base de données fut le suivant:

Cette analyse peut être recoupée avec des évènements récents pour voir leur impact sur le grand public ou même avec l'analyse des hashtags les plus utilisés pour observer des tendances sur les évènements et comparer l'importance de leurs impacts sur le public (en superposant les courbes). Cela peut aussi être utilisé pour observer le temps nécessaire à "l'oubli" d'une affaire politique par exemple.

j) Les dix hashtag les plus utilisés au mois de Décembre 2017

La requête suivante a été utilisée:

```
MATCH (h:Hashtag)-[:HAS_HASHTAG]-
(t:Tweet)-[:DATED_OF]-(date:Date{Month:"Dec",Year:"2017"}) RETURN h,count(h) as Nb ORDER BY
Nb DESC LIMIT 10
```

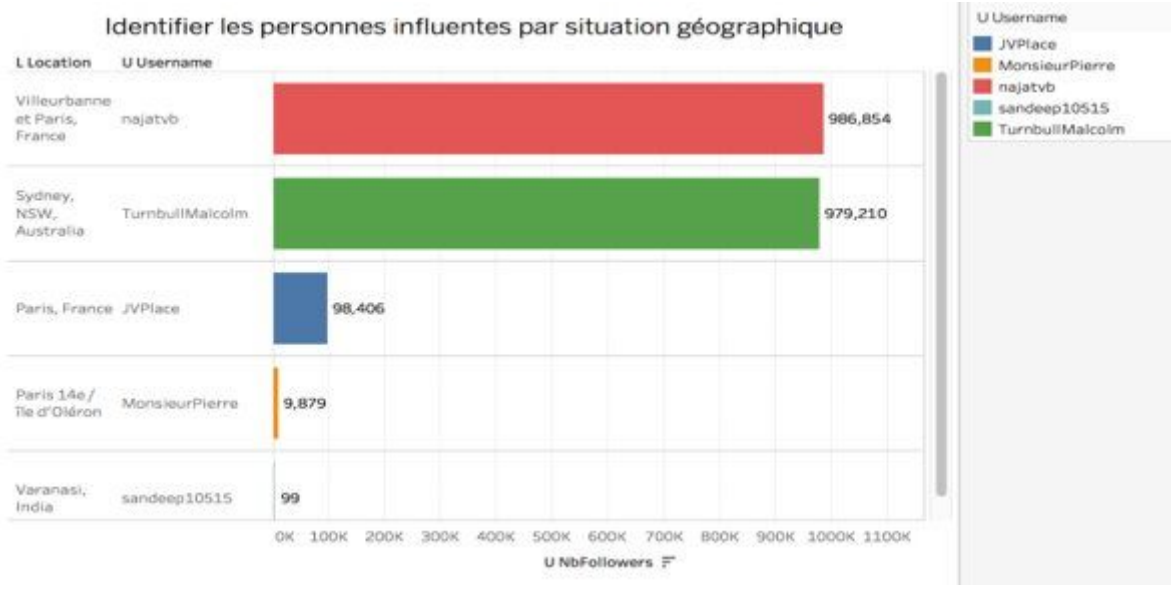
Le résultat observé dans la base de données fut le suivant:

k) Identifier les personnes influentes par situation géographique

La requête suivante a été utilisée:

```
MATCH (u:User)-[:FROM]-(l:Location)
WHERE l.Location<>''
RETURN u,l ORDER BY u.NbFollowers DESC LIMIT 5
```

Le résultat observé dans la base de données fut le suivant:

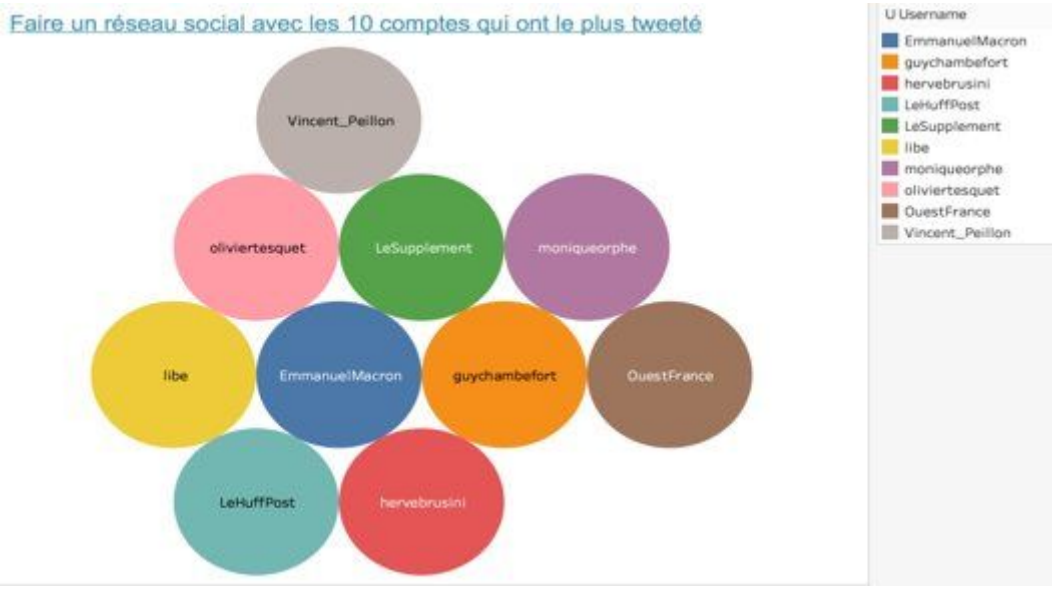


l) Les 10 comptes qui ont le plus tweeté

La requête suivante a été utilisée:

```
MATCH (u:User) - [tw:TWEETED]- (t:Tweet)
RETURN u,count(t) AS Nb ORDER BY Nb DESC LIMIT 10
```

Le résultat observé dans la base de données fut le suivant:



Forces et faiblesses de la solution

a) Forces de la solution

La grande force de cette solution est le type de base de données utilisé. En effet, comme il a été dit au début, les bases de données graph sont particulièrement efficaces pour représenter des réseaux sociaux.

L'utilisation de neo4j permet aussi de réaliser un bon nombre d'analyse facilement et rapidement et pour les analyses un peu plus complexe il est possible d'utiliser les plugins "APOC" et "Graph Algorithm" ou même un autre logiciel connecté à la base de données comme Qlik View, Qlik Sense ou tableau par exemple.

Une autre force de la solution neo4j est la possibilité d'utiliser un bon nombre de langages (Java, Python..) et logiciels (Talend) pour réaliser les processus ETL ce qui rend neo4j très accessible aux développeurs.

b) Faiblesses de la solution

Quelques faiblesses ont été répertoriées au cours du développement de cette solution:

La première, est une faiblesse commune à tous les projets en Big Data: elle nécessite une machine puissante ou même plusieurs machines en parallèle ainsi qu'une bonne connection. En effet une fois le programme d'extraction des données lancée, il faut plusieurs heures pour récupérer quelques milliers d'utilisateurs et de tweets. Un ralentissement est aussi observé avec l'augmentation du volume de données présent dans la base. L'affichage de la base de données intégrale dans le navigateur neo4j est aussi capable de mettre les machines à rude épreuve à partir d'un certain volume de données.

Une deuxième faiblesse a été observée: le rate limit imposé par Twitter. Il n'est possible de solliciter l'API Twitter qu'un faible nombre de fois toutes les 15 minutes à partir d'un compte Twitter. De la même façon le volume de données récupérable est limité. Il est arrivé de nombreuse fois que le programme d'extraction s'arrête en pleine exécution pour avoir excéder ce fameux rate limit. Il fallait alors attendre 15 minutes pour pouvoir relancer le programme (en prenant soin de corriger ce qui avait causé le problème à la base) à partir de ce compte.

Une troisième faiblesse, par rapport aux bases de données relationnelles, est le langage de requêtage. Si le langage CYPHER permet de réaliser un bon nombre de requêtes, dès que les requêtes deviennent complexes il est difficile de les réaliser, ce qui amène à regretter l'utilisation du SQL.

La quatrième faiblesse est la localisation de la base de données qui ne pouvait être réalisée qu'en local, l'option permettant de mettre cette base de données dans le cloud ne pouvant être choisie en raison d'un manque de budget lors de ce projet. Chaque membre du groupe a donc dû exécuter le programme d'extraction sur sa propre machine pour avoir accès à la base de données.

La cinquième faiblesse est la nature parcellaire de la documentation. En effet, à de nombreux endroits, des fonctions ne sont pas détaillées ou trop sommairement. Il arrive même que certaines fonctions soient déficientes et donc inexploitables.

La dernière faiblesse est la visualisation d'un gros volume de donnée: le nombre de noeuds et d'arcs présent à l'écran sont souvent illisible rendant ainsi les analyses compliquées à cause de cet effet "plat de spaghettis".

c) Comment palier à ces faiblesses?

Il est cependant possible de palier à la majorité de ces faiblesses sans trop de difficultés. En passant à des solutions payantes il est possible d'utiliser la technologie du Cloud en conjonction de neo4j, il est aussi possible de prendre les solutions entreprises qui augmente le rate limite et donc le nombre d'appels à l'API Twitter faisable toutes les 15 minutes ainsi que le volume de données récupérable en une requête.

Ensuite comme indiqué plus haut, il est possible de mettre plusieurs machines en parallèle pour augmenter leur puissance de traitements et donc les soulager lors des visualisations ou lors du chargement des données dans la base de données.

Conclusion

Ce challenge a permi de constater que même si les SGBD dédiés aux Big Data sont performants, les solutions sont encore loin d'égaler les bases de données relationnelles en terme de facilité d'utilisation et de diversités d'analyses. Cependant, le Big Data étant une problématique relativement récente, les outils vont certainement évoluer assez rapidement et s'améliorer.
De plus même avec des analyses un peu plus complexes, sur des données difficilement traitables, il est possible d'extraire des données très intéressantes. Par exemple une analyse supplémentaire possible mais qui n'a pas été réalisée dans le cadre de ce challenge: la répartition des connexions des utilisateurs dans la journée pourrait permettre de savoir à quel moment un message a le plus d'impact sur le réseau.
La solution traitées dans ce sujet serait plus adaptée au stockage de métadonnées et serait intéressante à utiliser en conjonction avec un autre type de base de données (comme une base de données document par exemple) qui contiendrait les données voulues.
Ces outils dans un futur proche permettront sûrement plus précisément qu'aujourd'hui de dessiner des tendances utiles pour les politiques (en réussissant peut-être à identifier l'ironie...), de transmettre l'information au plus grand monde et le plus rapidement possible pour les journalistes et même des analyses comportementales. Et leur utilisation va probablement se répandre de plus en plus.