

La Compacité De La Big Data Et La Programmation Orientée Graphe.

Taoufik Douhi

Mourad BECHROURI et Islam Eddine HADI.

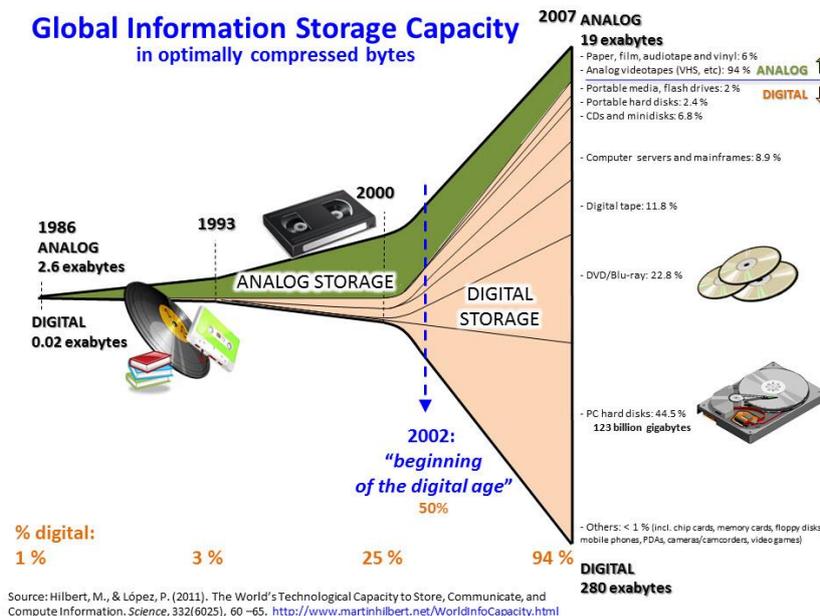
Résumé: Ce travail discute les extrémités et les échecs de la base de données relationnelle volumineuse (Big Data), et les remèdes par la modélisation orientée graphe qui plus d'avantage sur la conception et la programmation en se basant sur les principes de compacité de la Big Data, notamment la localisation par sous-graphes et les groupes denses.

Mots Clés: Mega données, base de données relationnelle, SQL, Graphe, Neo4j, Cypher, Compacité.

AMS Classifications : 68Mxx, 68Nxx, 68Pxx.

I. Introduction générale

Le phénomène méga données, données massives, base de données volumineuses où Big Data commençait en 1993 à noyauter silencieusement aux différents systèmes et organismes, possèdent une part du marché mondiale de la prise de décision et la planification de l'action, ainsi dans nombreux domaines de recherche scientifiques ; Électronique : La théorie du traitement se signale, Médecine : Biochimie Générique et Moléculaire, Marketing : Étude des comportements de consommateurs, l'observation astronomique, les télécommunications et réseau de capteurs ... [1] L'analyse et le traitement de mégadonnée originale, qui ont été collecté, ranger et stocker ou parfois migrer depuis une ou plusieurs autres architectures, pose un défi significatif, un goulot d'étranglement pendant la durée d'exploit.



Notamment un gros volume de données collecté, norme de tailles parfaitement hétérogènes, d'autre part la mesure de la rapidité avec laquelle un paquet d'informations arrivent aux organisations. Ce sont les trois caractéristiques majeures du phénomène, le volume, la variété et la vitesse.

L'internet des objets :

Selon l'Organisation des Nations Unies (ONU), plus de données ont été créées en 2011 que dans toute l'histoire de l'humanité [5], Ces données sont générées lorsque l'accès au réseau internet est devenu possible pour les objets tels que : téléphone, montre, lunette ... Ce qui a marqué la notion "internet des objets", un écosystème numérique désormais globale [6], permettant ainsi le développement de certains procédés, principalement adopté par le domaine industriel et informatique :

« Project Glass », les lunettes Google

À réalité augmentée, permettant de superposer à une image du monde réel des éléments virtuels comme un plan une photo ou une fiche récupérée sur l'internet. [7]



« **Wing** », est un petit appareil qui va évaluer l'état de santé des personnes qui souffrent d'asthme. Il va les aider à détecter s'ils risquent de souffrir d'une crise d'asthme. [8]

« **Maison Intelligentes** » : la gestion des flux (eau, énergie, aliments,

information...) et la sécurité pourraient être pour tout ou partie gérés par un système informatique, auto-apprenant, en interaction avec les besoins des occupants, éventuellement en utilisant des énergies et des ressources moins nuisantes pour l'environnement, la maison s'adapterait aux habitudes et aux goûts de ses habitants et invités



(éventuellement malvoyants, handicapés, âgés, malades, etc.), grâce à un profilage de ces derniers, communiqué au système gérant la maison. Certains imaginent aussi une maison intelligente et autonome pour ses besoins en eau, thermies, frigories ou électricité, capable « de détecter d'elle-même, des dysfonctionnements ou des changements de paramètres susceptibles de présenter un gaspillage d'énergie ou un danger » [9] Plusieurs autres domaines sont particulièrement impactés, comme le logement (économies d'énergie ...), La mobilité (Systèmes d'aide à la conduite, coordination entre les véhicules ...) [6].

Une estimation qui a été faite par l'institut Montagne, décrit qu'entre 30 et 212 milliards d'objets pourrait être connectée d'ici 2020.

Le Volume :

Une explosion quantitative de données numériques, parfois redondante nécessite la mise en disposition des espaces de stockage physique 'Data Center' performant aussi très coûteux, comme : 'cloud computing', 'Super Calculateurs Hybrides', 'Systèmes de fichiers Distribués'. [3]

Exemple : Domaine de psychologie.

D'après le système de codage des expressions faciales élaborer par Paul Ekman, psychologue américain, chacun de nous contient 10 000 mimiques, et chaque mimique a le groupe d'action de muscle impliquée, notons qu'il existe 43 muscles faciaux. [4]

Donc pour pouvoir identifier les terroristes avant qu'ils passent à l'action il nous faut un espace de stockage exprimer en téraoctet $To(1 To=1024 Go)$ ou pétaoctet $Po(1 Po=1024 To)$ dans un supermarché à titre d'exemple.

La Variété :

L'information n'admet pas un seul format, ainsi qu'elle provient d'une ou plusieurs source, d'une façon structuré ou non structuré, ceux qui a mener a impliquée des nouveaux enjeux des systèmes d'informations, stocker l'information la ranger d'une façon qu'on peut l'exploiter en respectant certain règles de sociabilité du Hardware. Fessant appelle aussi cette fois aux théorèmes fondamentale mathématique du domaine théorie des graphes dans les majorités des systèmes gérés.

Exemple : Réseaux sociaux

Twitter a créé FlockDB pour gérer des réseaux Graph larges, et peu profonds. Le projet Cayley de Google s'est directement inspiré de la base de données Graph qui propulse son Knowledge Graph, la base de connaissances reliée à son moteur de recherche. Facebook utilise de son côté Apache Giraph, construit pour assurer une grande évolutivité.

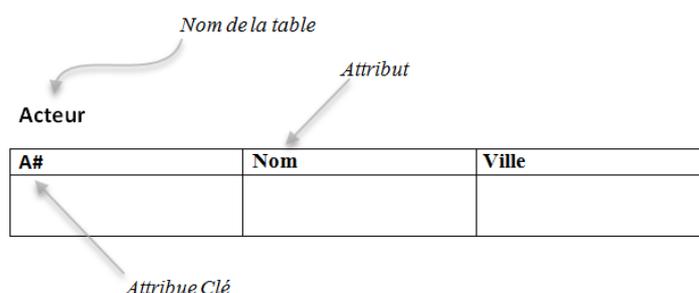
La Vitesse :

C'est la mesure de la rapidité avec laquelle un **paquets** arrivent aux organisation, qui ont prévus a améliorer des protocoles de gestion des tsunamis de flux d'information dynamique, a fin de pouvoir exécuter des réactions(Algorithme) parmi d'autre, et anticiper certain mesure(Allocation dynamique) a titre d'optimisation.

II. Base de données Relationnelle

A. Principes fondamentaux du modèle relationnel.

La table est une forme simple et parlante pour rassembler des données ou représenter des informations. Par exemple, la table ci dessous est conçue pour recueillir des informations sur les acteurs d'un film, Elle porte un nom écrit en majuscule, ACTEUR les noms de colonnes, désignent les attributs des Acteurs. Dans notre table, ce sont le numéro d'acteur "A#", le nom d'acteur "Nom" et son domicile «Ville». Chaque attribut (attribute) représente une propriété dont les valeurs entrées dans la table appartiennent à un domaine prédéfini (domain). Dans la table ACTEUR, l'attribut A# Permet d'identifier chaque acteur de manière unique. En vertu de cette propriété, nous déclarons que le numéro d'acteur est une Clé. L'attribut Ville prendra comme valeurs les noms des localités, et l'attribut Nom, les noms des acteurs.



Introduisons maintenant les données du personnel dans la table ACTEUR ligne par ligne, comme indiqué dans la figure 1-2. Nous observons que certaines valeurs apparaissent plusieurs fois dans la table. Ainsi, le lieu de domicile Shirebrook est présent deux fois dans la table ACTEUR. Cette double apparition traduit un fait important : les acteurs Neeson et Statham habitent tous les deux à Shirebrook. Dans la table ACTEUR, non seulement les noms des localités mais aussi ceux des acteurs, peuvent apparaître plusieurs fois. Pour cette raison, l'attribut clé A# est indispensable pour identifier chaque acteur dans la table de manière unique.

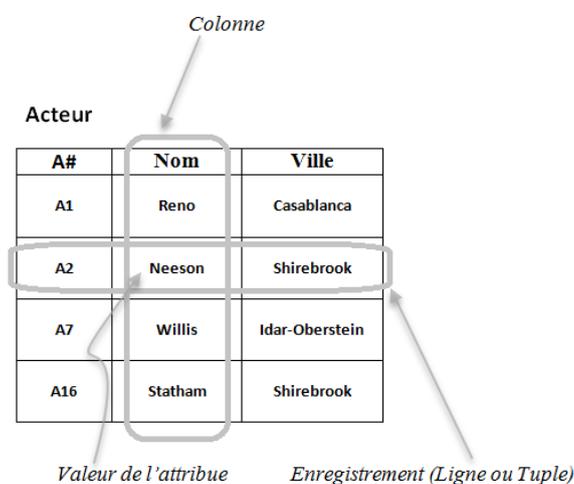


Figure 1-2

Définition d'une table

Une table ou relation (table, relation, en anglais) est un ensemble de tuples représenté sous la forme tabulaire et ayant les propriétés suivantes :

1. Chaque table porte un nom unique.
2. À l'intérieur de la table, le nom de chaque attribut est unique et désigne une colonne avec des propriétés spécifiques.
3. Une table peut contenir un nombre quelconque d'attributs, l'ordre des colonnes dans la table est indifférent.
4. L'un des attributs ou une combinaison d'attributs identifie de façon unique chaque tuple dans la table et sera la clé primaire.
5. Une table peut contenir un nombre quelconque de tuples, l'ordre des tuples dans la table est indifférent.

Selon cette définition, dans un modèle relationnel (relational model, en anglais) chaque table est vue comme un ensemble non ordonné de tuples. En vertu de cette notion ensembliste, un tuple ne peut donc apparaître qu'une seule fois dans une table. [10]

B. SQL, langage normalisé au niveau international.

Comme nous l'avons déjà vu, un modèle relationnel représente les informations sous la forme tabulaire. À chaque table est associé un ensemble de tuples ou d'enregistrements de même type. Ce concept d'ensemble permet d'implémenter des opérations ensemblistes pour interroger et manipuler les bases de données. Ainsi, le résultat d'une opération de sélection est un ensemble ; en d'autres termes, chaque interrogation de la base de données génère un résultat sous la forme d'une table. Si la table consultée ne contient aucun tuple répondant aux propriétés requises, le résultat obtenu par l'utilisateur est une table vide. Les opérations de mise à jour portent également sur une table ou un groupe de tables spécifiques. [10]

➤ C'est à la fois :

- langage de définition de données (LDD, en anglais DDL Data Definition Language) :
- II. C'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle, ainsi que d'en modifier ou en supprimer.
- Un langage de manipulation de données (LMD, ou en anglais DML, Data Manipulation Language) :
- III. Cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle.
- langage de contrôle de données (LCD, ou en anglais DCL, Data Control Language) :
- IV. Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données.

C. Exemple.

Films		
idFilm	Titre	Réalisateur
1	Matrix	Réalisateur de Matrix
2	La soupe aux choux	Réalisateur de la soupe aux choux

Comment peut-on créer cette table avec MySQL :

Table films:

```
CREATE TABLE films (idFilm INT not null AUTO_INCREMENT, Titre VARCHAR (255) not null ,
Réalisateur VARCHAR (255) not null , PRIMARY KEY (idFilm))
```

- Insérer des données :

```
INSERT INTO films (Titre,Réalisateur) VALUES ('Matrix','Réalisateur de Matrix')
INSERT INTO films (Titre,Réalisateur) VALUES ('La soupe aux choux','Réalisateur de la soupe aux choux')
```

- Supprimer ces données :

```
DELETE FROM films where idFilm=2
DELETE FROM films where Titre='La soupe aux choux'
```

- Modifier des données:

```
UPDATE films SET Réalisateur='John Woo' where id=1
UPDATE films SET Réalisateur='John Woo' where Réalisateur='Réalisateur de Matrix'
```

- Traiter les données:

```
SELECT COUNT(*) FROM films
```

```
SELECT titre FROM films
```

[11]

III. Bases de données Orientées Graphes

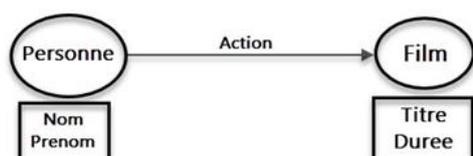
3.1 Principes fondamentaux du modèle Orientée graphe.

Les graphes sont présents à chaque étape de la vie d'un logiciel, de la conception à l'exécution en passant par le déploiement. Avant de concevoir un logiciel, le développeur passe par une phase de modélisation, durant laquelle il détaille les différentes composantes des objets qui seront mis en œuvre dans son code. Ces objets disposent de propriétés, de méthodes, et ils dépendent les uns des autres. L'UML est une norme qui sert à visualiser cette modélisation à l'aide de boîtes (pour les objets) et de flèches (pour les relations entre ces derniers). Les interconnexions entre ces objets pouvant former des cycles, nous sommes bien face à une modélisation de type graphe. [12]

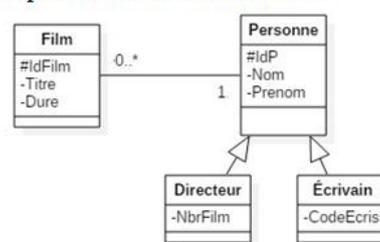
La structure en graphe modélise des données très complexes, de façon simple et intuitive. Il n'y a pas de distinction entre les données et les relations. On explore en profondeur, afin de mettre en évidence des corrélations et des relations complexes et souvent implicites. On ne se limite plus à des résumés statistiques ou des prédictions de tendances. La « découverte » d'informations est mise en avant par une exploration active de la donnée et de son contexte. La fouille de données classique consiste généralement à calculer des statistiques descriptives le long des nombreuses dimensions des tableaux de données, ou à faire des projections et avoir des indicateurs qui résument certaines informations. Cependant, certains cas ne se prêtent pas naturellement à cette forme de questionnement, comme identifier un enchaînement d'actions, un groupe d'éléments associés, des relations entre des données qui sont a priori dans des tables séparées.

Exemple :

Représentation Graphique :



Représentation Relationnelle :



3.2 La base de données Neo4j, et son langage de requête Cypher.

Développé en Java par la société suédo-américaine Neo technology, permet de représenter les données en tant qu'objets reliés par un ensemble de relations, chaque objet possédant ses propres propriétés. La base de données de graphes, permet au développeur de commencer directement le codage, les données stockées dans la base assurant un parallélisme direct avec les données elles-mêmes. En d'autres termes, à mesure que l'organisation des données se peaufinera, les programmes suivront au fur et à mesure. [12]

Cypher est un langage informatique de requête orienté graphe utilisé par Neo4j. Il se veut simple et efficace dans la formulation des requêtes d'interrogation et de mise à jour des bases de données orientées graphe. Cypher est relativement simple dans sa construction syntaxique mais reste néanmoins un langage efficace. [13]

3.3 Exemple.

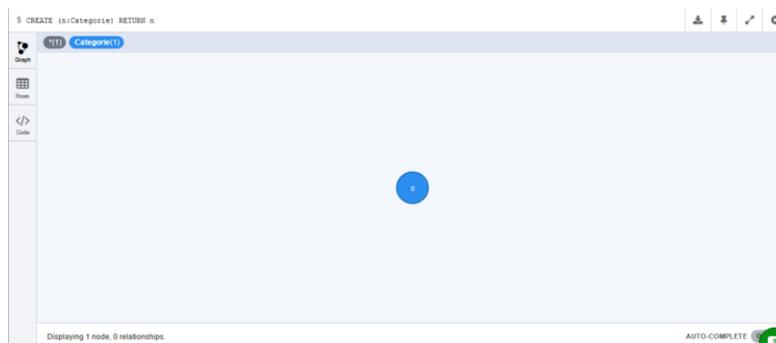
3.4.1 Insérons un nœud :

```

    $ CREATE (n:Catégorie) RETURN n
  
```



a-1) Résultat :



3.4.2 Insérer de la donnée sur un nœud

```
MATCH (n:Catégorie)
SET n.titre="Base de données", n.titreCourt="BDD"
RETURN n.titre, n.titreCourt
```

En exécutant cette requête, un nouveau panneau s'affichera mais il sera vide. En effet, comme la requête retourne les données du nœud (identifiées par les clés titre et titreCourt) et non des éléments de graphe, il est nécessaire de basculer en mode Données pour voir apparaître le tableau suivant :

\$ MATCH (n:Catégorie) SET n.titre="Base de données", n.titreCourt="BDD" RETURN n.titre, n.titreCourt	
n.titre	n.titreCourt
Base de données	BDD

3.4.3 Insérer une relation

Créons un deuxième nœud en exécutant la requête suivante :

```
CREATE (n:Produit {nom:"Neo4j", site:"http://neo4j.com"}) RETURN n
```

Puis créons une relation de type EST entre ce nœud et le précédent en utilisant la requête ci-dessous :

```
MATCH (m:Produit), (n:Catégorie)
CREATE (m)-[:EST]->n
RETURN n,m
```

La requête ci-dessus passe par deux étapes : retrouver les éléments de graphe à associer puis créer la relation entre les éléments retrouvés.



3-4-1-4. Retrouver des éléments de graphe

Vous pouvez créer d'autres relations et d'autres nœuds. Voici quelques exemples :

Exemple 1.1. Lister les 25 premiers éléments du graphe

```
MATCH (n) RETURN n LIMIT 25
```

Exemple 1.2. Retrouver tous les nœuds de type Produit

```
MATCH (n:Produit) RETURN n
```

Exemple 1.3. Retrouver le nœud de type Produit dont le nom est Neo4j

```
MATCH (n:Produit {nom:"Neo4j"}) RETURN n
```

Notez dans cet exemple que le double-clic sur un nœud permettra d'afficher ses nœuds voisins.

Exemple 1.4. Retrouver tous les nœuds connectés au nœud de type Produit dont le nom est Neo4j

```
MATCH (n:Produit {nom:"Neo4j"})--m RETURN n,m
```

Exemple 1.5. Supprimer tous les éléments du graphe

```
MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
```

[20]

IV. Extrémité atteinte par les Base de données relationnelle volumineuse et la programmation Orientée Graphe.

4.1 Extrémité atteinte par les Base de données relationnelle.

L'architecture relationnelle s'est affronté par un défi particulier, il s'agit d'enregistrer certain types de données complexes comme les images en 3D, ou des objets de données abstraites qui ne sont pas disponibles pour stocker. La solution la plus évidente est de mettre en place un système de base de données plus robuste, embaucher un administrateur de base de données pour la maintenance et acheter un logiciel spécial. Peu importe le niveau de confidentialités des informations, la taille de l'entreprise ou l'institution le cout de cette opération est géant. D'autre part, pour un système qui gèrent [1,n] catégories, elles-mêmes composées de [0,n] forums, eux-mêmes pouvant être composés de [0,n] sous-forums qui ont [0,n]adhérent, Le type de requêtes « trouver toutes les adhérent de ville X qui ont fait une action Y dans toutes les sous-forums », augmente exponentiellement la complexité, ce qui signifie grand effort analytique du Hardware et dans un temps remarquable, Et donc des latences incompatibles avec la réactivité exigée des applications en ligne. Donc Les bases de données de type relationnelles sont plus adaptées aux systèmes hiérarchiques,

4.2 Les principaux cas d'usage pour les bases de données Orientée graphe.

"Le principal cas d'usage est la recommandation" note Cédric Fauvet. "Meetic utilise une base de donnée Graph pour obtenir des recommandations pertinentes entre les membres. Par exemple pour connaître les amis d'amis qui ont le même âge. Et ce principe de filtrage collaboratif, calculé avec un algorithme de recommandation, est très difficile à utiliser avec des bases de données relationnelles". C'est avec cette technologie également qu'eBay peut optimiser ses livraisons en Californie, promises en moins de deux heures. "7 lignes de code Cypher ont suffi pour mettre en place un système où le livreur peut récupérer un nouveau colis et le livrer alors même qu'il effectue déjà une livraison" mentionne le responsable de Neo4j. "Cela permet de faire deux à trois tournées au lieu d'une seule dans le même temps".[15]

4.3 Avantages de la programmation Orienté Graphe par rapport a Relationnelle.

Expérience : MySQL vs. Neo4j

L'ensemble de données utilisé sont généré artificiellement, le graphe a 1 million de sommets et 4 millions d'arêtes.

Notons que toutes ces expériences ont été exécutées sur un MacBook Pro avec un 2,66 GHz Intel Core 2 Duo et 4Gigs de RAM à 1067 MHz DDR3. Les paquets utilisés étaient Java 1.6, MySQL JDBC 5.0.8 et Blueprints Pipes 0.1.2.

Parcours :

Le parcours qui a été évalué sur chaque base de données a commencé à partir de certains sommets racine et émané n-étapes. Il n'y a pas de tri, pas de distinction, etc. Les deux seules variables pour les expériences sont la longueur du traversal et le sommet de la racine pour démarrer le parcours. Dans MySQL, les 5 requêtes suivantes indiquent des trajectoires de longueur 1 à 5. Notez que le "?" Est un paramètre variable de la requête qui indique le vertex de la racine.

```
SELECT a.inV FROM graph as a WHERE a.outV=?
```

```
SELECT b.inV FROM graph as a, graph as b WHERE a.inV=b.outV AND a.outV=?
```

```
SELECT c.inV FROM graph as a, graph as b, graph as c WHERE a.inV=b.outV AND b.inV=c.outV AND a.outV=?
```

```
SELECT d.inV FROM graph as a, graph as b, graph as c, graph as d WHERE a.inV=b.outV AND b.inV=c.outV AND c.inV=d.outV AND a.outV=?
```

```
SELECT e.inV FROM graph as a, graph as b, graph as c, graph as d, graph as e WHERE a.inV=b.outV AND b.inV=c.outV AND c.inV=d.outV AND d.inV=e.outV AND a.outV=?
```

Résultats :

Pour MySQL et Neo4j, les résultats de la requête (SQL et Pipes) ont été itérés. Ainsi, tous les résultats ont été récupérés pour chaque requête.

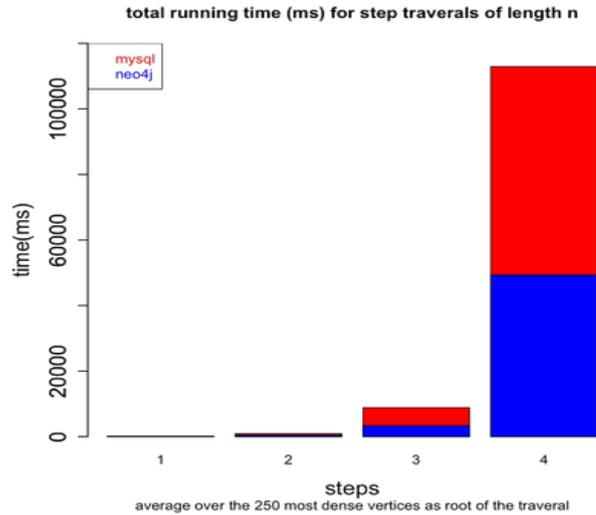
Dans MySQL, ceci a été fait comme suit.

```
while (resultSet.next()) {
    resultSet.getInt(finalColumn);
}
```

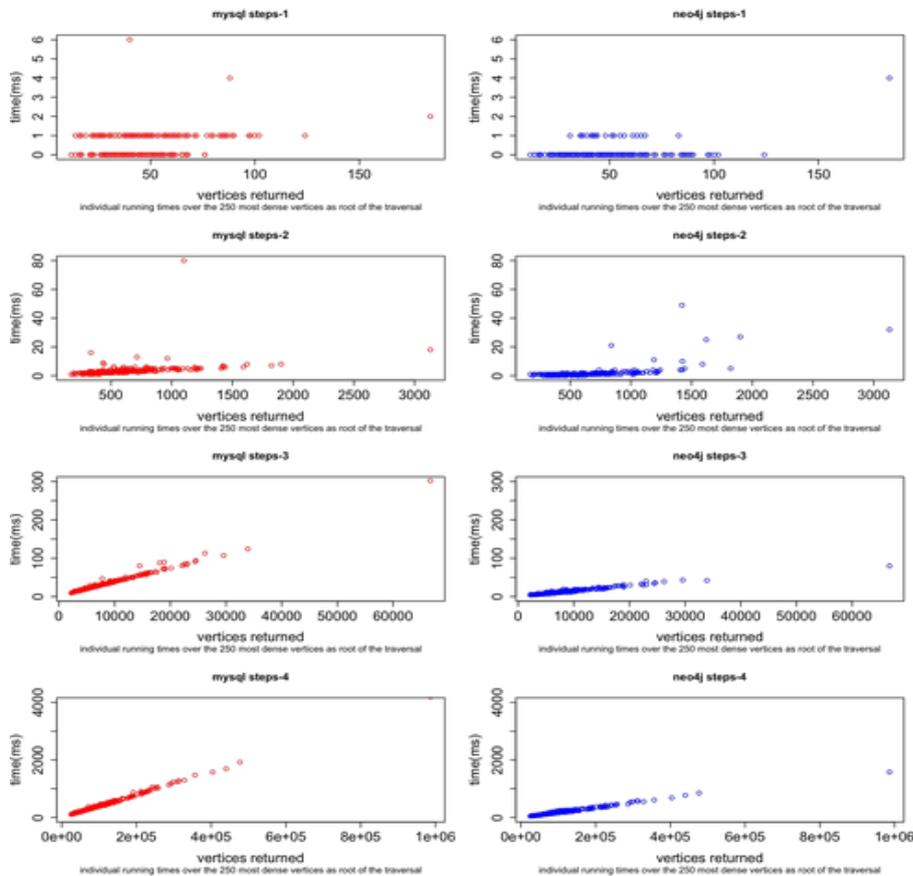
Dans Neo4j :

```
while (pipeline.hasNext()) {
    pipeline.next();
}
```

Les données brutes sont présentées ci-dessous ainsi que le nombre total de sommets renvoyés par chaque traversée, ce qui est bien sûr le même pour MySQL et Neo4j étant donné que le même ensemble de données de graphe est traité. Sachez également que les traversaux peuvent boucler et ainsi, plusieurs des mêmes sommets sont renvoyés plusieurs fois. Enfin, notez que seul Neo4j a le temps d'exécution pour un parcours de longueur 5. MySQL n'a pas terminé après avoir attendu 2 heures pour terminer. En comparaison, Neo4j a pris 14,37 minutes pour compléter un parcours de 5 étapes.



Ensuite, les points de données individuels pour MySQL et Neo4j sont présentés dans le graphique ci-dessous. Chaque point indique combien de temps il a fallu pour renvoyer n nombre de sommets pour les différentes longueurs de traversée.



Conclusion : En conclusion, étant donné un parcours d'un graphe artificiel avec des statistiques naturelles, la base de données graphique Neo4j est plus optimale que la base de données relationnelle MySQL. Cependant, aucune tentative n'a été faite pour optimiser la VM Java, les requêtes SQL, etc. Ces expériences ont été exécutées à la fois avec Neo4j et MySQL "out of the box" et avec une "syntaxe naturelle" pour les deux types de requêtes. [16]

Témoignage :

Cédric Fauvet, responsable commercial France de Neo4j.

- « Les bases de données Graph mettent en application la théorie mathématique des graphes. Ce pan des mathématiques cache de l'or. Jusqu'à présent il manquait un outillage pour mettre en application cette théorie qui est bien connue dans de nombreuses entreprises »
- "Les bases de données Graph permettent de traiter des données connectées grâce à un travail sur les jointures" explique Cédric Fauvet. "Les autres technologies de type NoSQL permettent de mieux traiter les données, pour en tirer plus de valeur. Le monde de la Business Intelligence a fortement évolué ces dernières années. Nous, nous travaillons sur la "matière noire", c'est à dire ce existe entre les données. Ce "vide" a une valeur, il s'agit de la connexion entre les données".[15]

4.4 Protocoles de Gestion.

Open Graph, est le protocole le plus populaire sur internet, c'est un protocole qui permet à une page web de devenir un objet "riche" dans un réseau social.

Plus concrètement, le protocole Open Graph est un ensemble de balises qui permet à un webmaster de donner aux principaux réseaux sociaux (Facebook, Google +, Twitter, Linked in...) des informations précises sur ses pages. Ces informations permettront aux réseaux sociaux de mieux afficher un lien sur une ces pages qui aurait été placé par leur utilisateurs.



Exemple :

Un code contenant des balisages de protocole Open Graph, pour qu'un réseau social comprenne qu'il s'agit d'un film :

<meta property="og:type" content="video.movie" />

Son lien est : <meta property="og:url" content="http://www.imdb.com/title/tt0117500/" />

```
<html prefix="og: http://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="http://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="http://ia.media-imdb.com/images/rock.jpg" />
...
</head>
...
</html>
```

[17]

V. Compacité de la base de données.

5.1 Localisation pour un accès flexible a une donnée.

Cypher est un langage déclaratif inspiré du SQL, qui se focalise sur les aspects du résultat plutôt que sur les méthodes ou les moyens d'obtenir le résultat ,afin qu'il soit lisible et expressif , Parmi les astuces de ce langage est sa méthode de mettre en place une base de donnée dans son état initiale, c'est-à-dire pouvoir se déplacer a toute secteur en profondeur du graphe(Base de données)a travers cette liaison entre les éléments ou entité nommé adjacence ou pointeur physique [18].le deuxième important astuce c'est que Neo4j a Mis en opération le mécanisme de l'interrogation de la base de données ces relations ou pointeur physique eux-mêmes pour y accéder directement a destination que nous souhaitons extraire son information, au lieu d'utiliser une ou des relations d'équivalences entre les ensembles de données(Jointure), ce qui permis d'exprimer une requête complexe de façon élégante est compacte[19].

Ci-joint un morceau de code du tutoriel de Neo4j :

```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})

//Création du film 'TheMatrix'

CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})

CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})

//Création des acteurs 'Keanu, Carrie'

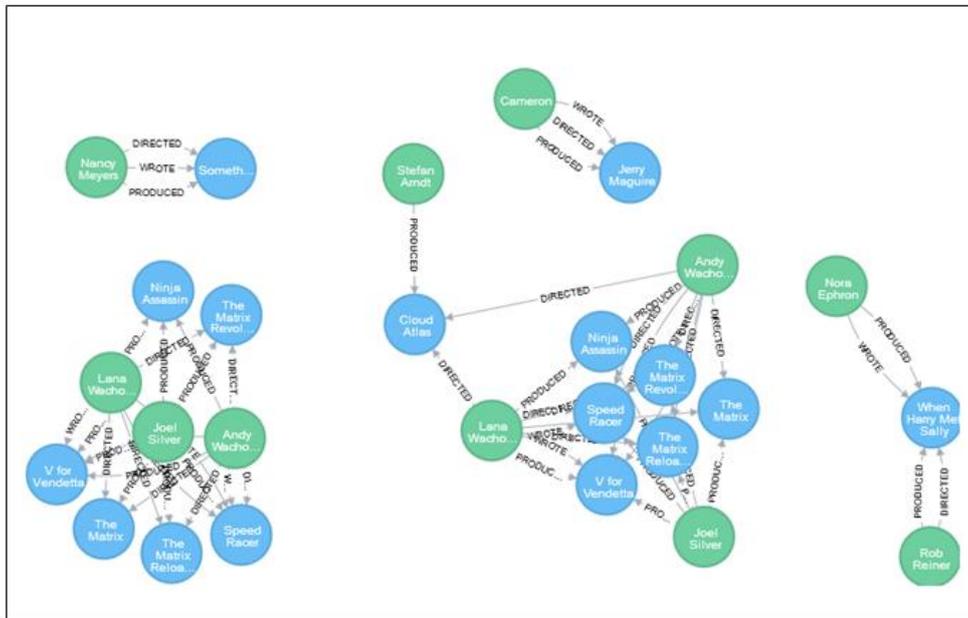
CREATE

(Keanu)-[:ACTED_IN {roles:['Neo']}]>(TheMatrix),

(Carrie)-[:ACTED_IN {roles:['Trinity']}]>(TheMatrix)

//Création des relations 'ACTED_IN', afin de pouvoir lié le film 'TheMatrix', avec les deux acteurs 'Keanu' et 'Carrie'.
```

Une visualisation d'une partie de graphe 'Movie' de ce tutoriel :



Le graphe obtenue par cette modélisation est une réunion des sous graphes ce qui donne a la base de données une compacité locale.

5.2 Densité.

Notons qu'un noyau est un nœud qui regroupe un nombre maximal d'arcs autour de lui.

L'ensemble des noyaux constituent un groupe de données dense dans le graphe.

Exemple :

Le groupe des films : { 'Ninja Assassin', 'The Matrix', }

Est un groupe dense dans le graphe 'Movie', dont chaque film est un noyau.

MATCH (n:Movie) RETURN n LIMIT 17 //



*Par ce concept de densité notre base de données peut être vue comme étant un groupe de données dense, d'où sa compacité La conception orientée graphe sur la base de données **Movie** est basée sur les deux concepts Localisation : répartition en sous-graphe, ou densité facilite la tâche pour les requêtes de traitement est rend la base de données ainsi compact.*

Bibliographie

- [1]. Big Data Analysis with Signal Processing on Graphs Aliaksei Sandryhaila, Member, IEEE and Jos'e M. F. Moura, Fellow, IEEE
- [2]. Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60–65.
- [3]. Source : https://fr.wikipedia.org/wiki/Big_data Facial Action Coding System: Investigator's Guide, Volume 1 Paul Ekman_1978.
- [4]. « Robert Kirkpatrick, Director of UN Global Observatory.org, 5 novembre 2012.
- [5]. Big data et objets connectés Faire de la France un champion de la révolution numérique RAPPORT AVRIL 2015 Page 9-14,61-68
- [6]. https://fr.wikipedia.org/wiki/Google_Glass<http://www.objetconnecte.net/wing-eviter-crise-asthme-2811>https://fr.wikipedia.org/wiki/Maison_intelligente
- [7]. Introduction pratique aux bases de données relationnelles Deuxième édition Traduit de l'allemand par : Dac Hoa Nguyen Springer page 17-24
- [8]. http://www.sqlfacile.com/apprendre_bases_de_donnees/sql_par_1_exemple_1
- [9]. :Bases de données orientées graphes avec Neo4j
- [10]. <https://fr.wikipedia.org/wiki/Neo4j>
- [11]. [https://fr.wikipedia.org/wiki/Cypher_\(langage\)](https://fr.wikipedia.org/wiki/Cypher_(langage))
- [12]. <http://www.zdnet.fr/actualites/5-points-pour-mieux-comprendre-les-bases-de-donnees-graph-39839720.htm>
- [13]. <https://dzone.com/articles/mysql-vs-neo4j-large-scale>
- [14]. Source : <http://ogp.me/>
- [15]. Onofrio Panzarino-Learning Cypher_ Write powerful and efficient queries for Neo4j with Cypher, its official query language-Packt Publishing (2014)
- [16]. Rudi Bruchez-Les bases de données NoSQL et le Big Data _ Comprendre et mettre en oeuvre-Eyrolles (2015)
- [17]. http://www.d-booker.net/HTML/neo4j/prise_en_main/connaitre/id_neo_pratique.html.