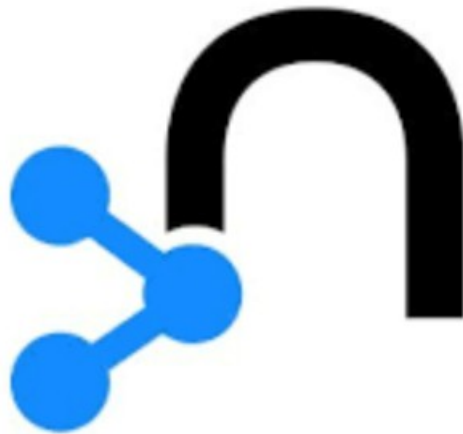


Synthèse de présentation :

Veille technologique.



NEO4J

Introduction :

Neo4j est un système de gestion de base de données au code source libre basé sur les graphes, développé en Java par la société suédo-américaine Neo technology. Le produit existe depuis 2000, la version 1.0 est sortie en février 2010. Elle prend en charge le modèle ACID (Atomicité, Cohérence, Isolation et Durabilité) et assure un cluster ((groupe de ressource)). Chaque transaction est fiable et unique au sein de cette base. L'un des points mis en avant par Neo Technology est la possibilité d'exécuter plusieurs millions de requêtes par seconde sur des relations entre nœuds.

Neo4J intègre son propre langage de requêtage, le Cypher. Il peut être appelé grâce à l'API REST fournie, ou via l'une des différentes implémentations disponibles pour les principaux langages, dont Python

Elle se veut une BD de graphe rapide, fiable, évolutive et entièrement automatisable avec le Cloud.

I. Comment ça fonctionne ?

Le modèle de Neo4j :

Il utilise des graphes contenant des labels et propriétés ce qui laisse la possibilité d'associer différemment des objets du graphes (configurer la taille des traits, la couleur pour différencier les données, etc).

A) GRAPHE.

Un graphe est un ensemble d'objets stockés sous forme de nœuds et liés entre eux par des relations qui sont nommées.

Les entités sont représentées par des nœuds configurables par les propriétés et associables par l'usage des labels.

Les propriétés sont le seul moyen de différencier les entités (nœuds). Contrairement à un SGBDR comme MariaDb, pas besoin d'identifiant unique pour chacune des entités car ce sont les labels et les propriétés qui font la différence.

Les labels servent ainsi à créer des groupes ou des catégories de nœuds pour décrire l'objet du nœud et à filtrer les recherches lors de requêtes.

Voici la représentation au format GUI d'un graphe. Sur cet exemple nous avons stocké deux objets dans la base de données: un objet de type Staff qui représente un employé, et un objet de type Company qui représente une entreprise.

Je souhaite également enregistrer le fait que le Staff travaille pour la Company, nous avons besoin de créer une relation WORK_IN pour spécifier ce qui les lie ensemble et dans quel sens. Nous pouvons voir que l'objet Xavier travaille pour Auchan.

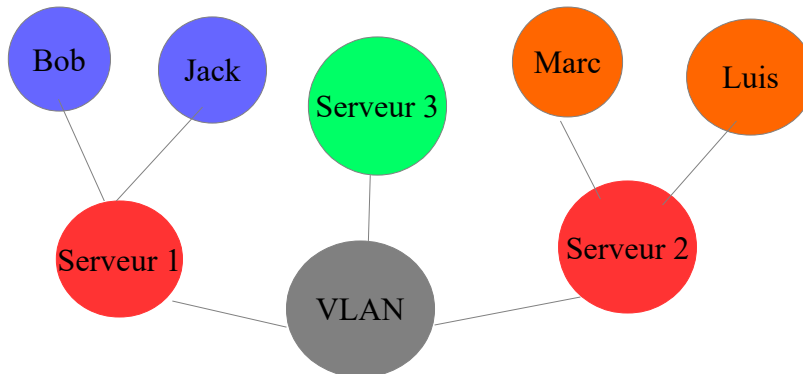
Nous entendons par graphe orienté le fait que les relations aient un sens. Dans le cas présent, de l'objet de type `Staff` vers l'objet `Company`.



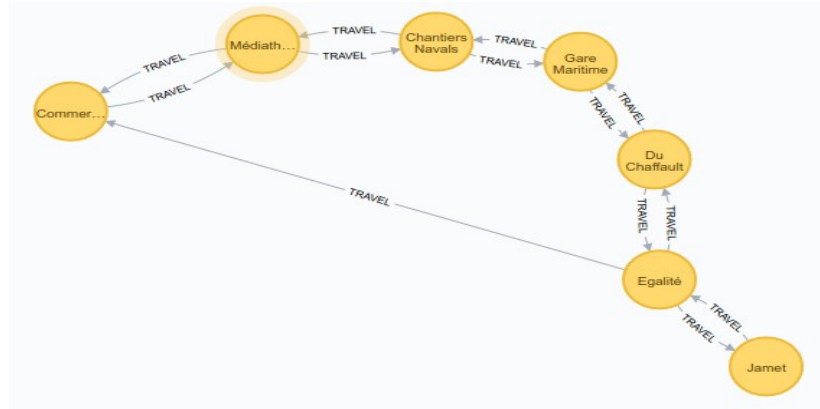
```
CREATE (Theo:Staff {title:'Theo', joined:2014, job: 'Developer', city: 'Nantes'})
CREATE (Jon:Staff {title:'Jonathan', joined:2013, job: 'Project Manager', city: 'Lille'})
CREATE (Synbioz:Company {name:'Synbioz', born:2007})
MATCH (n:Staff)
MATCH (m:Company {name:'Synbioz'})
CREATE (n)-[:WORK_IN]->(m)
```

B) EXEMPLE d'USAGE :

- Gestion de réseau/analyse d'impact : avoir la main sur son système et identifier en temps réel le client affecté par une maintenance ou une panne ;



- Logistique : calculer le meilleur chemin pour livrer un client ;



- Social, collaboration : rechercher très facilement qui sont les amis de mes amis ;

Parmi les grands noms, on citera : Viadeo, ebay, National Geographic, Adobe, Cisco, Deutsche Telekom, Walmart, Glowbl, Career Builder, Meetic, SFR, Voyages-SNCF, etc. Les cas d'usage sont nombreux : dans l'énergie, les services financiers (pour la détection de fraudes), la gestion des ressources humaines, l'informatique, les jeux, la logistique (pour le routage de colis, par exemple), mais aussi la distribution (pour la recommandation de produits) et le transport (pour le calcul d'itinéraires).

II) Comparatif.

A) Les avantages et les inconvénients :

Avantages :

- **Open Source**
- **Facile à apprendre les commandes du langage de requête CYPHER**
- **Représente des données semi-structurées très facilement**
- **Haute disponibilité pour les applications temps réel des grandes entreprises**
- **Réglage simplifié**
- **Régulièrement mis à jour**

Inconvénients :

- **Ne prend pas en change le SQL**

B) Benchmark des SGBDs :

Voici le top 20 des bases de type Graphes (le site DB-engine inclut également ici certaines bases de type multi-modèles) :

Rank			DBMS	Database Model	Score		
Jan 2020	Dec 2019	Jan 2019			Jan 2020	Dec 2019	Jan 2019
1.	1.	1.	Neo4j	Graph	51.66	+1.10	+4.86
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	31.51	+0.07	+7.12
3.	4.	4.	ArangoDB	Multi-model	5.21	+0.33	+0.92
4.	3.	3.	OrientDB	Multi-model	5.11	+0.18	-0.63
5.	5.	5.	Virtuoso	Multi-model	2.65	+0.01	+0.00
6.	6.	6.	JanusGraph	Graph	1.77	+0.02	+0.51
7.	7.	7.	Amazon Neptune	Multi-model	1.73	+0.16	+0.67
8.	8.	10.	GraphDB	Multi-model	1.14	-0.01	+0.33
9.	11.	11.	Dgraph	Graph	1.04	+0.09	+0.39
10.	9.	8.	Giraph	Graph	1.02	-0.02	+0.02
11.	10.	13.	TigerGraph	Graph	1.00	+0.04	+0.41
12.	12.	9.	AllegroGraph	Multi-model	0.86	-0.02	-0.05
13.	13.	12.	Stardog	Multi-model	0.81	+0.03	+0.18
14.	14.	18.	FaunaDB	Multi-model	0.80	+0.11	+0.50
15.	15.	15.	Blazegraph	Multi-model	0.64	+0.02	+0.17
16.	16.	14.	Graph Engine	Multi-model	0.58	0.00	+0.01
17.	17.	17.	InfiniteGraph	Graph	0.39	0.00	+0.02
18.	19.	27.	GRAKN.AI	Multi-model	0.28	+0.05	+0.19
19.	18.	19.	FlockDB	Graph	0.27	-0.01	+0.04
20.	20.	21.	HyperGraphDB	Graph	0.19	0.00	+0.02

Dans ce top 20 des BD de type graphe les plus utilisés, Neo4J est la plus utilisée. Elle est restée pendant 2 ans au sommet battant Microsoft Azure Cosmos DB et ArangoDB.

Pour voir l'efficacité de neo4j, analysons sa performance contre MySQL.

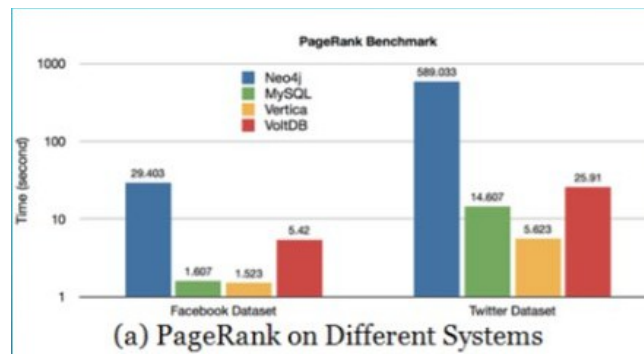
Une base de données de graphes devrait être beaucoup plus rapide qu'une base de données relationnelle en **parcours de graphes**. Par exemple, dans un réseau social, retrouver tous les amis des amis d'un utilisateur. Encore plus pour les amis d'amis d'amis. Nous pouvons voir ici MySQL et Neo4j avec une base de données de 1 000 000 d'utilisateurs présenté par couche.

Le temps d'exécution est en secondes, pour 1 000 utilisateurs.

Profondeur	Temps d'exécution – MySQL	Temps d'exécution - Neo4j
2	0.016	0.010
3	30.267	0.168
4	1,543.505	1.359
5	Pas fini en 1 heure	2.132

Pour les simples requêtes d'amis d'amis, Neo4j est 60% plus rapide que MySQL. Pour les amis des amis des amis, Neo est 180 fois plus rapide. Et pour la requête de profondeur quatre, Neo4j est 1 135 fois plus rapide. Et MySQL s'étouffe avec la requête de profondeur 5.

HELAS...



Le PageRank est un critère mathématique permettant de mesurer la popularité d'une page sur le web. Plus il est élevé, plus la page est censée être populaire (et donc intéressante pour l'internaute). Depuis [l'origine de Google](#), le PageRank (que l'on abrège en PR) est l'un des éléments pris en compte par l'algorithme de classement... mais ce n'est pas le seul !!! Il en existe plusieurs centaines...

Le PageRank est également utilisé par le moteur de recherche Google dans tout son processus. Ainsi, un site ayant une ou plusieurs page au PageRank élevé aura beaucoup d'avantages (il sera indexé plus souvent, et donc ses mises à jour seront rapidement disponibles dans Google).

On voit ici qu'il est plus efficace d'utiliser MySQL pour l'algorithme de popularité d'une page web.

III) Nouveautés.

A) Neo4J Aura

Le service Aura, qui représente la version de Neo4j disponible en cloud public depuis 2020.

la gestion de larges volumes de données comportant des milliards de nœuds et de relations pour des requêtes ultra-rapides et le chiffrement de bout-en-bout, l'isolation VPC intégrée et le contrôle d'accès basé sur les rôles, ainsi qu'au niveau des nœuds, pour une sécurité maximale des données.

Les entreprises bénéficieront également d'une tarification transparente, simple et à la demande basée sur la consommation réelle. En adoptant cette édition Cloud, les DSI profitent d'une meilleure fiabilité pour les applications critiques : conformité ACID (atomicité, cohérence, isolation et durabilité), architecture auto-réparatrice garantissant une disponibilité de service de 99,95%, mises à jour et correctifs appliqués sans interruption...

Ils bénéficient aussi bien sûr d'un accompagnement personnalisé premium avec une réponse aux incidents de sévérité 1 en une heure et un accès à la plus grande

communauté d'experts de graphes, garantissant aux utilisateurs tout ce dont ils ont besoin pour leur prochaine application révolutionnaire.

B) Nouvelles fonctionnalités

En ce mois de Février 2022, Neo4j a développé de nouvelles fonctionnalités.

La première s'agit de Neo4j DATA Importer qui dispose d'une amélioration de l'importation des données avec une interface utilisateur sans code qui est capable de charger des données de fichiers plats comme CSV ou TSV avec 1 million de lignes. La possibilité d'esquisser un modèle de graphique et de changer les données d'entrée sur sa structure et ses propriétés. Et la réduction du temps de chargement des fichiers.

Une seconde nouveauté intéressante constituent les requêtes de chiffrement qui sont dotées d'une meilleure performance lorsqu'elles sont paramétrées. Le navigateur d'aujourd'hui vous permet de déclarer des paramètres avec la commande `:param` pour un seul paramètre, ou la commande `:params` pour déclarer un objet de paramètres.

Dans cette version, Neo4J facilite la construction d'un objet `params` dans l'éditeur. Lorsque la requête que vous essayez d'exécuter a des paramètres non définis, vous verrez maintenant un *lien d'aide sous l'erreur* pour vous aider à définir les paramètres. Cliquer dessus vous créera une commande `:params` avec des paramètres préexistants déjà définis et des espaces réservés pour que vous remplissiez les paramètres manquants.

Conclusion :

Neo 4J est un outil puissant et ludique qui permet de mieux comprendre les problématiques du BigData et de mieux pouvoir appréhender ce domaine. De plus, il est intéressant d'utiliser Neo4J car constamment en amélioration. Bien qu'attrayante, elle n'est pas toujours à privilégier face à une base relationnelle. Tout va dépendre du contexte de votre application et de sa structure de données.

Il existe d'autres bases de données orientées graphe, comme OrientDB et ArangoDB, pour ne citer que les plus connues.