Rapport de Projet GMIN332 : Gestion de données complexes

Par : ALIJATE Mehdi - COUSOT Kevin - NEGROS Hadrien

15 Janvier 2014

Table des matières

1 Introduction **2**

2 Jeu de données **2**
2.1 Description . . . 2
2.2 Schéma . . . 2

3 Analyse du système **2**
3.1 Différents modèles de représentation des données . . . 2
3.1.1 Modèle relationnel . . . 2
3.1.2 Triple Store . . . 3
3.1.3 Base de données orientée graphe . . . 3
3.1.4 Base de données orientée colonnes . . . 3
3.2 Une API pour les gouverner tous : Jena . . . 3
3.3 Le langage de requete SPARQL . . . 4

4 Application **4**
4.1 Architecture . . . 4

5 Démonstration **4**
5.1 Principe . . . 4
5.2 Requêtes . . . 4

6 Discussion et Conclusion **4**

7 Sources **4**

Résumé

Accès et consultation de données provenant de différentes solutions de persistance (gros volumes de données distribuées et hétérogènes) au travers d'un démonstrateur.

1 Introduction

Les systèmes NoSQL et les technologies du Web Sémantique sont une altérnative aux SGBD classiques. Cependant, elles sont encore très loin d'être couramment utilisées, comme celles du monde relationnel "classique".
Néanmoins, de grands acteurs d'internet (*comme Facebook (Cassandra Project puis HBase), Google (BigTable), Ubuntu One (CouchDB)... etc.*) commencent à exploiter des bases de données de type NoSQL. L'avantage de celà, c'est que la majorité de ces projets est open source et sous licence libre.
Dans notre projet, nous avons voulu explorer les différentes technologies qui puissent gérer des données RDF. Nous avons donc étudié une solution basée sur le mapping de bases relationnelles (**D2RQ**), et des solutions basées sur des bases NoSQL (**TDB, Hbase et Neo4j**).
Le but de ce projet est d'exploiter différentes sources de données, gérées au travers de plusieurs systèmes de gestion de données, afin de permettre un accès et consultation de ces derniers. Ceci, via une application permettant d'interconnecter ces sources de données, basée sur l'API de Jena.

2 Jeu de données

/**TODOO à finir**

2.1 Description

Le jeu de données utilisé représente des données de vocabulaire de l'**INSEE** et des données **Geonames**. L'INSEE (Institut National de la Statistique et des Etudes Economiques) collecte, produit, analyse et diffuse des informations sur l'économie et la société françaises. A ce titre, il conduit des recensements et des enquêtes, il gère des bases de données et exploite aussi des sources administratives. Geonames est une base de données géographique gratuite et librement accessible sur Internet (www.geonames.org). La base regroupe plus de 8 millions de noms de lieux géographiques, et beaucoup d'informations autours de ces lieux (par exemple la population, la subdivision administrative, le code postal, la latitude, la longitude, l'altitude, etc.).

2.2 Schéma

//**A finir avec une description plus précise de chaque données avec un schéma d'interconnexion**

3 Analyse du système

3.1 Différents modèles de représentation des données

3.1.1 Modèle relationnel

Le modèle relationnel est le modèle de représentation le plus utilisé dans les SGBD actuellement. Il est basé sur l'algèbre relationnelle. Une BDD relationnelle est un ensemble de *tuples* groupés ensemble dans des *tables*, et un ensemble de contraintes qui, entre autre, permettent de définir des liens entre les tables. Pour pouvoir exploiter des données stockées dans une BDD relationelle en meme temps que des données stockée dans des BDD NoSQL, nous avons besoin d'une interface qui nous permettra de requêter dans un langage commun, différent donc de SQL.
Le SGBD que nous avons choisi est **MySQL**, et l'interface est **D2RQ**.

3.1.2 Triple Store

Un Triple Store est une base de donnée NoSQL conçue pour stocker des triplets RDFs. La consultation des données se fait avec le langage de requête **Sparql**. La création d'un TDB store à pour effet de creer des fichiers persistant dans un dossier précisé en entrée. Les fichiers de données RDF sont donc chargés une fois, puis l'accès aux données se fait en chargeant ce dossier dans notre application. Le seul type de donnée qu'un triple store peut stocker est le *triplet*. Contrairement au relationnel, ce modèle ne dépend donc pas d'un schéma.
Le triple store que nous avons décidé d'intégrer est **TDB**.

3.1.3 Base de données orientée graphe

Une base de données orientée graphe est une BDD NoSQL utilisant la théorie des graphes. Les données sont donc représentées sous forme de noeuds et d'arcs représentant ces noeuds. Ce modèle est pratique pour la représentation de données, car dans beaucoup de cas les données sont facilement représentables sous forme de graphe.
La base de donnée orientée graphe que nous avons décidé d'intégré est **Neo4j**

3.1.4 Base de données orientée colonnes

3.2 Une API pour les gouverner tous : Jena

Jena est une API Java qui peut être utilisée pour créer et manipuler des données RDF . Jena possède des classes pour représenter des graphes, des ressources, des propriétés et des littéraux.

Ce projet est uni autour de l'API de Jena (Cf. Section **??**) et de ses modèles.

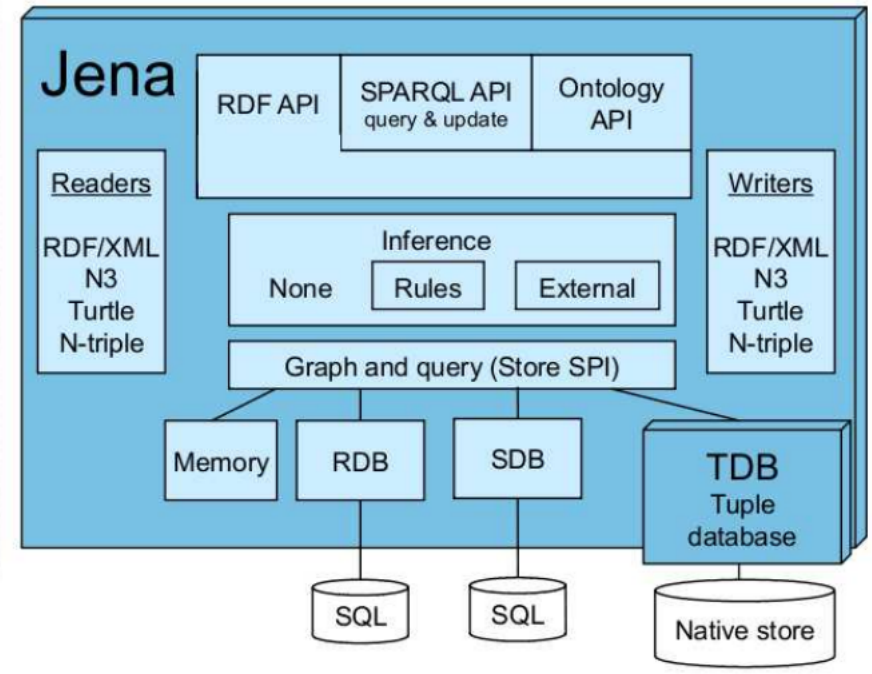

FIGURE 1 – Vue Générale de l'architecture JENA

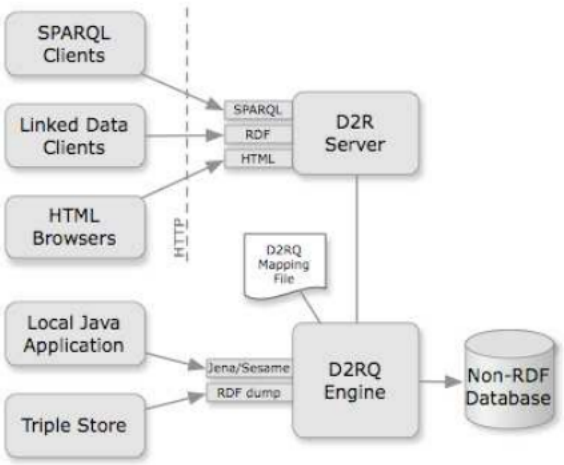

FIGURE 2 – Vue Générale de l'architecture D2RQ

3.3 Le langage de requete SPARQL

La réprésentation Model spécifique à Jena a permi l'union obtenus à partir de nos sources de données. Les requêtes SPARQL sont effectuées sur le modèle ainsi créé.
Chaque triple store définit ses propres fonctions à côté de celles de SPARQL (ARQ pour Jena). Pour gérer les requêtes SPARQL, Jena utilise sa propre librairie *ARQ*. Cette librairie définit également sa propre syntaxe, compatible notre mode de requêtage SPARQL, cela y ajoute des fonctionnalités semblables à ce qu'on peut trouver dans MySql par exemple, comme COUNT, MAX, etc, car ces fonctions ne sont pas gérées par défaut par SPARQL.

4 Application

4.1 Architecture

5 Démonstration

5.1 Principe

//TODO

5.2 Requêtes

//TODO

6 Discussion et Conclusion

//TODO

7 Sources

//TODO