Extraction du schéma d'une BD NoSQL orientée documents

Amal Ait Brahim *, Rabah Tighilt Ferhat * et Gilles Zurfluh *

*Institut de Recherche en Informatique de Toulouse (IRIT)
Université Toulouse Capitole - 2 Rue du Doyen-Gabriel-Marty - 31042 Toulouse - France
<prenom.nom>@gmail.com
http://www.ut-capitole.fr/

Résumé. Au cours de ces dernières années, l'utilisation des systèmes NoSQL pour stocker et exploiter les bases de données (BD) massives n'a cessé de s'accentuer. Ces systèmes apportent une grande souplesse d'utilisation notamment grâce à leur propriété « schema less ». Cependant, le schéma de la BD NoSQL est nécessaire pour permettre aux utilisateurs (notamment les décideurs) d'exprimer leurs requêtes. Dans ce qui suit, la notion de « BD NoSQL » est utilisée pour désigner une BD gérée par un SGBD NoSQL. L'objectif de cet article est d'extraire le schéma d'une BD NoSQL orientée documents. Pour cela, nous proposons un processus incrémental d'élaboration du schéma pendant l'exploitation de la BD.

1 Introduction

Au cours de ces dernières années, la nécessité d'exploiter les données qui sont générées et accumulées par les dispositifs informatiques n'a cessé de s'accentuer. Ceci est à cause du volume, qui peut dépasser plusieurs téraoctets, et de la variété de ces données qui sont qualifiées de complexes. De plus, ces données sont souvent saisies à très haute fréquence et doivent donc être filtrées et agrégées en temps réel pour éviter une saturation inutile de l'espace de stockage. Ces problématiques sont désignées par l'expression « Big Data » Chen et Zhang (2014) et caractérisées par la règle dite des « 3V » Laney (2001). Il s'agit du Volume (des masses des données considérables à gérer), de la Variété (des données complexes) et de la Vélocité (en référence à la collecte et au traitement en temps-réel de ces données). Les approches classiques basées principalement sur le paradigme relationnel, ne peuvent pas répondre à ces objectifs et exigent de nouvelles approches de stockage et de manipulation des données.

Regroupées sous le terme NoSQL (Not Only SQL) (Han et al., 2011), ces approches permettent une plus grande adaptabilité dans des contextes fortement distribués, ainsi qu'une gestion performante des données complexes Darmont et al. (2007). Les BD NoSQL proposent une nouvelle façon de gérer des données. Elles sont classées en quatre catégories : orientée clé-valeur, orientée documents, orientée colonnes et orientée graphes.

La plupart des BD NoSQL sont caractérisées par l'absence du schéma de données lors de la création. Dans les SGBD relationnels, tous les éléments du schéma sont fixés à l'avance ;

autrement dit, le nom et le type des attributs de toutes les lignes d'une table, sont spécifiés avant la saisie des données. Cependant, les BD NoSQL ne nécessitent pas un schéma rigide pour le stockage des données. La propriété « schema less » apparait dans de nombreux systèmes NoSQL tels que MongoDB, CouchDB, HBase et Neo4j (notons cependant qu'elle est absente dans Cassandra et Redis par exemple). Cette propriété offre une souplesse indéniable en permettant de stocker dans des tables des lignes ne respectant pas un schéma unique. Par exemple, l'ajout de nouveaux attributs dans une ligne existante se fait sans modifier les autres lignes de même type préalablement stockées.

Le schéma d'une base de données est un élément de connaissance essentiel pour la manipulation des données. En effet, la connaissance du schéma de la base s'avère nécessaire, voire indispensable, pour le traitement des requêtes où sont mentionnés les noms des tables et les noms des attributs.

Dans cet article, nous proposons une approche permettant d'extraire le schéma d'une BD NoSQL orientée documents qui vérifie la propriété « schema less ». Le but est de fournir le schéma aux utilisateurs (notamment les décideurs) afin de formuler leurs requêtes d'interrogation. Pour ce faire, nous préconisons un processus d'élaboration du schéma tout au long de l'exploitation de la BD NoSQL.

2 Etude de cas

Pour motiver et illustrer nos travaux, nous utilisons le cas des dossiers médicaux partagés (DMP [1]) inspiré de l'application mise en place actuellement par la CPAM [2] (Caisse Primaire d'Assurance Maladie) dans plusieurs départements français.

Un dossier médical partagé contient les données de santé et le parcours de soins d'une personne (assuré social). Il est créé à la naissance de la personne et se clôt lors de son décès. Il permet au médecin traitant et aux autres professionnels de santé (médecins, infirmiers, dentistes, protection civile, etc.), d'accéder et de partager les informations médicales concernant cette personne. Il doit permettre aussi aux médecins de visualiser l'historique médical d'une personne. En outre, le DMP permet aux gestionnaires de la CPAM d'analyser les données des patients ou de les confronter avec celles d'autres malades, ceci dans le but d'avoir une meilleure connaissance du domaine de la santé.

Les données collectées dans le cadre de la mise en place du DMP, présentent les caractéristiques généralement admises pour les Big Data (les 3 V). En effet, à terme, le DMP doit prendre en charge la gestion des données médicales de plus de 60 millions de Français. Par conséquent, le volume des données recueillies quotidiennement auprès des professionnels de santé, peut atteindre plusieurs téraoctets ; si nous supposons que le DMP stocke jusqu'à 1 Mo de données par personne (ce qui est peu), sa taille globale atteindra 60 To. D'autre part, le DMP peut contenir des données de formats variés (ordonnances, textes explicatifs, comptes rendus, radiographies, scintigraphies, etc.), des tableaux de grande dimension (enregistrement

1. https ://www.mon-dmp.fr/
2. https ://www.gouvernement.fr/guide-victimes/caisse-primaire-d-assurance-maladie-cpam

de mesures issues des capteurs comme la température et le pouls). Ces données peuvent varier d'un patient à un autre selon ses caractéristiques physiologiques et biologiques. Enfin, certaines données sont produites en flux continu par des capteurs ; elles doivent être traitées quasiment en temps réel car elles peuvent s'intégrer dans des processus sensibles au temps (mesures franchissant un seuil qui impliqueraient l'intervention d'un praticien en urgence par exemple). Nous avons donc stocké l'ensemble de ces données dans un système NoSQL.

Pour l'implantation de notre cas d'étude, nous avons utilisé un cluster de 3 machines. Chaque machine est de type Intel Core i5, 8 Go de RAM et 2 To de disque. L'une de ces machines est configurée pour agir comme maitre et les autres comme esclaves.

3 Etat de l'art

Dans l'article Klettke et al. (2015), les auteurs proposent un processus d'extraction du schéma d'une BD NoSQL orientée documents. Le processus préconisé fournit un schéma en format JSON pour une collection de documents. Il utilise un graphe comme niveau intermédiaire pour élaborer le schéma de la BD NoSQL.

D'autre part, dans l'industrie, plusieurs solutions d'intégration des Big Data telles que Drill d'Apache [3], CloudMdsQL [4] et BigIntegrator ont été proposées pour l'extraction du schéma d'une BD NoSQL Bondiombouy (2015).

Par exemple, Apache Drill est un moteur d'interrogation de BD. Il permet de visualiser et d'interroger des données stockées dans des systèmes relationnels, des systèmes NoSQL et des systèmes de gestion de fichiers tels que HDFS d'Hadoop et Amazon Harrison (2015). Apache Drill peut joindre dans la même requête des données provenant de plusieurs sources. Pour l'interrogation des BD NoSQL « schema less », Drill peut fournir dynamiquement le schéma avant le traitement de la requête. Pour cela, avant de formuler sa requête, l'utilisateur doit effectuer une suite d'opérations à l'aide d'un script shell de Drill. Cependant, Drill affiche les noms des collections de MongoDB mais ne mentionne pas les champs. Par conséquent, l'utilisateur ne peut pas exprimer des requêtes précises.

À travers cet état de l'art, Il apparait que peu de travaux ont étudié l'extraction du schéma d'une BD NoSQL de type « schema less ». En effet dans l'article Klettke et al. (2015), la construction du schéma de la BD s'applique uniquement pour une seule collection de documents. Or, l'expression des requêtes nécessite d'élaborer un schéma global de la BD NoSQL.

4 Contribution

Nos travaux visent à extraire le schéma d'une BD NoSQL orientée documents qui est de type « schema less ». Pour ceci, deux démarches sont possibles comme le montre la figure 1. La première, dite « à froid », extrait le schéma d'une BD NoSQL existante et la deuxième, dite « à chaud », consiste à construire le schéma au fur et à mesure de l'exploitation de la BD NoSQL.

3. https ://drill.apache.org/
4. http ://cloudmdsql.gforge.inria.fr/

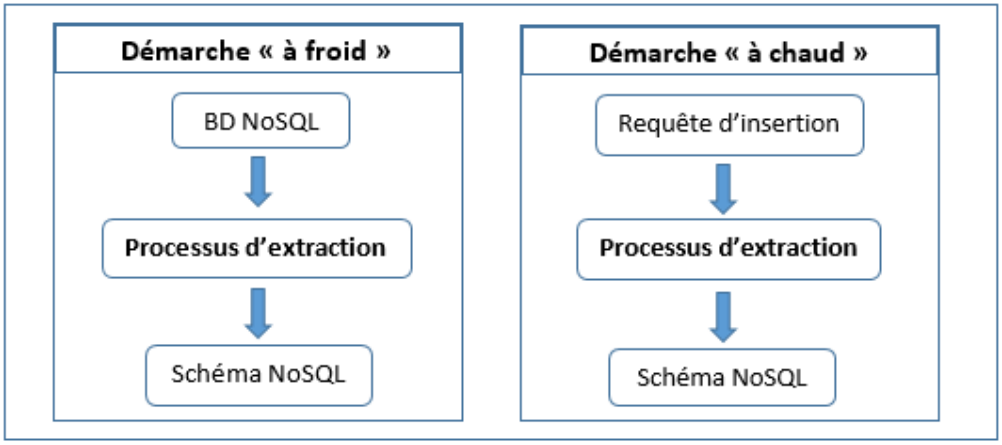

FIG. 1 – *Démarches d'extraction du schéma d'une BD NoSQL.*

Nous avons choisi l'extraction « à chaud » pour répondre aux besoins de l'application du DMP. En effet, le schéma de données est élaboré de manière incrémentale, au fur et à mesure de la saisie des données dans la BD NoSQL ; il est ainsi disponible à tout moment pour permettre aux utilisateurs l'écriture de requêtes.

Nous utiliserons dans ce qui suit le vocabulaire spécifique au SGBD NoSQL MongoDB [5]. Il s'agit d'un SGBD NoSQL orienté documents conçu pour offrir de hautes performances en lecture et en écriture ainsi qu'une mise à l'échelle automatique de la base de données. Ce système stocke les données en format BSON (Binary JSON) qui est une représentation textuelle de données en clé/valeur. La clé est un identificateur unique associé à un agrégat de champs que l'on nomme document Bruchez (2015).

Un document est toujours identifié par une clé, notée _id, et consiste en un ensemble de champs composés d'un nom et d'une valeur. La valeur d'un champ peut être atomique (nombres, chaînes de caractères, dates et booléens) ou complexe (i.e. inclut d'autres documents). Les documents sont regroupés dans des collections ; chacune d'elles peut contenir des documents avec des structures différentes (i.e. les documents d'une collection ne possèdent pas forcément les mêmes champs) MongoDB (2018).

Nous proposons un processus d'extraction du schéma à chaud qui recalcule le schéma à chaque fois que l'utilisateur soumet une requête d'insertion. Notons qu'il serait nécessaire de prendre en compte également les requêtes de suppression et de modification. Dans cet article, nous nous restreignons aux requêtes d'insertion. Notre processus procède comme suit :

— Il extrait les noms des champs ainsi que le nom de la collection tels qu'ils sont mentionnés dans la requête,

5. https ://www.mongodb.com/

— Il met à jour le schéma en rajoutant les champs extraits précédemment dans la collection correspondante.

4.1 Formalisation

Dans cette section, nous proposons une formalisation des éléments en entrée et en sortie de notre processus : la requête d'insertion et le schéma d'une BD NoSQL orientée documents.

4.1.1 Formalisation de la requête d'insertion

Une requête d'insertion R^i est définie par (N^c, A), où :
— $R^i.N^c$ est le nom de la collection,
— $R^i.A$ = $\{a_1, ..., a_n\}$ est un ensemble d'attributs, où $\forall\ j \in [1..n]$, a_j est défini par (N, T, V), avec :
  — $a_j.N$est le nom de l'attribut,
  — $a_j.T$est le type de l'attribut,
  — $a_j.V$est la valeur de l'attribut. Elle peut être atomique ou complexe.

4.1.2 Formalisation du schéma de la BD NoSQL orientée documents

Le schéma d'une BD orientée documents BD^{od} est définie par (N, Cl), où :
— $BD^{od}.N$ est le nom de la base de données orientée documents,
— $BD^{od}.Cl$ = $\{cl_1, ..., cl_n\}$ est un ensemble de collections, où $\forall\ i \in [1..n]$, le schéma d'une collection cl_i est un couple (N^c, Fl) où :
  — $cl_i.N^c$ est le nom identifiant la collection,
  — $cl_i.Fl = Fl^a \cup Fl^{cx} \cup \{Id^{cl}\}$ est l'ensemble de champs atomiques Fl^a = $\{fl_1^a, ..., fl_r^a\}$ et complexes Fl^{cx} = $\{fl_1^{cx}, ..., fl_s^{cx}\}$ qui seront utilisés pour définir les documents de cl_i, où :
    — $\forall\ i \in [1..r]$, le schéma d'un champ atomique $fl_i^a \in Fl^a$ est un couple (N^a, T) où :
      — $fl_i^a.N^a$ est le nom identifiant le champ atomique,
      — $fl_i^a.T$ est le type de champ.
    — $\forall\ j \in [1..s]$, le schéma d'un champ complexe $fl_j^{cx} \in Fl^{cx}$ est un couple (N^{cx}, Fl') où :
      — $fl_j^{cx}.N^{cx}$ est le nom identifiant le champ complexe,
      — $fl_j^{cx}.Fl'$ est l'ensemble des champs imbriqués dans fl_j^{cx}.
    — Id^{cl} est un champ spécial qui doit exister dans chaque document de cl_i afin de l'identifier d'une manière unique dans la collection. Ce champ porte un nom fixe noté _id.

4.1.3 Règles de correspondance

Nous proposons un processus permettant de décrire la correspondance entre l'élément en entrée : requêtes d'insertion et l'élément en sotie : schéma NoSQL orienté documents. Selon le type de la valeur des attributs, atomique ou complexe, nous appliquons les règles de transformation R1 ou R2 respectivement :

R1 : Chaque attribut atomique $a_i \in R^j.A$ est transformé en un champ atomique fl^a, où :
- $fl^a.N^a = a_i.N$,
- $fl^a.T = a_i.T$,
- Le champ fl^a est ajouté par la suite à l'ensemble de champs atomiques Fl^a de la collection correspondante.

R2 : Chaque attribut complexe $a_i \in R^j.A$ est transformé en un champ complexe fl^{cx}, où :
- $fl^{cx}.N^{cx} = a_i.N$,
- Les champs de fl^{cx} correspondent exactement aux attributs imbriqués dans a_i,
- Le champ de fl^{cx} est ajouté par la suite à l'ensemble de champs complexes Fl^{cx} de la collection correspondante.

Exemples : Nous illustrons notre proposition par les requêtes ci-dessous qui consistent à insérer des documents dans la collection « Assurés » de la BD « DMP ». Ces deux requêtes donnent deux versions différentes du schéma de la BD.

Requête 1 : `db.assurés.insert({'prénom':'David', 'nom':'Ricardo'})`

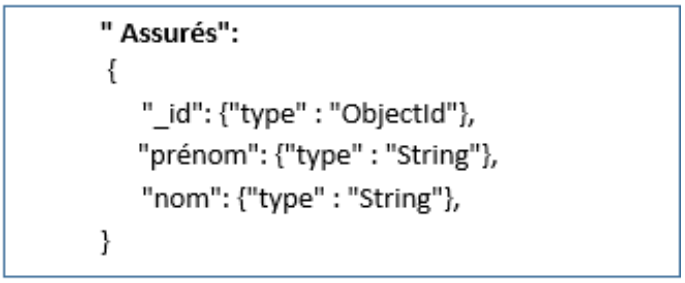

FIG. 2 – *Schéma de la BD « DMP » après l'exécution de la requête 1*

Requête 2 : `db.assurés.insert({'prénom':'Pierre', 'nom':'Torres', 'adresse': {'numRue':'14', 'nomRue':'Kabylie', 'codePostal':75015, 'ville':'Paris'}})`

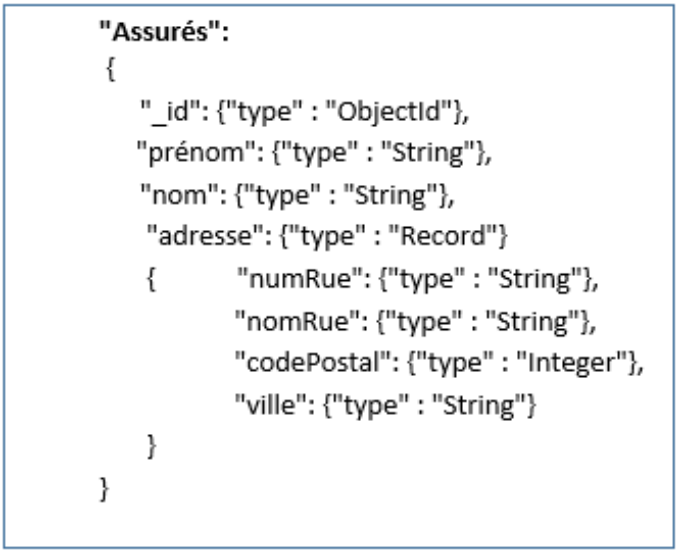

FIG. 3 – *Schéma de la BD « DMP » après l'exécution de la requête 2*

5 Conclusion

Nos travaux s'inscrivent dans le cadre de l'évolution des bases de données vers les Big Data, ceci pour prendre en compte le volume, la variété et la vélocité des données présents dans les nouvelles applications liées à la transformation digitale des entreprises. Nos études portent actuellement sur les mécanismes d'extraction du schéma d'une BD NoSQL pour permettre aux utilisateurs (notamment des décideurs) d'exprimer leurs requêtes d'interrogation.

Dans cet article, nous avons traité le processus de l'extraction du schéma d'une BD NoSQL. Pour cela, nous avons préconisé une démarche « à chaud », qui consiste à construire le schéma au fur et à mesure de l'exploitation de la BD. Notre processus d'extraction du schéma part d'une BD NoSQL vide ; le schéma est « recalculé » à chaque fois que l'utilisateur soumet une requête d'insertion.

Nous avons expérimenté notre démarche sur le cas des dossiers médicaux partagés (DMP). Le schéma de la BD NoSQL a été implanté sur le système MongoDB.

Références

Bondiombouy, C. (2015). Query processing in cloud multistore systems. In *BDA : Bases de Données Avancées*.

Bruchez, R. (2015). *Les bases de données NoSQL et le BigData : Comprendre et mettre en oeuvre*. Editions Eyrolles.

Chen, C. P. et C.-Y. Zhang (2014). Data-intensive applications, challenges, techniques and technologies : A survey on big data. *Information Sciences 275*, 314–347.

Darmont, J., O. Boussaid, J.-C. Ralaivao, et K. Aouiche (2007). An architecture framework for complex data warehouses. *arXiv preprint arXiv :0707.1534*.

Han, J., E. Haihong, G. Le, et J. Du (2011). Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pp. 363–366. IEEE.

Harrison, G. (2015). *Next Generation Databases : NoSQLand Big Data*. Apress.

Klettke, M., U. Störl, et S. Scherzinger (2015). Schema extraction and structural outlier detection for json-based nosql data stores. *Datenbanksysteme für Business, Technologie und Web (BTW 2015)*.

Laney, D. (2001). 3d data management : Controlling data volume, velocity and variety. *META group research note 6*(70), 1.

MongoDB (2018). Mongodb atlas database as a service. `https://www.mongodb.com/`. Online ; 5 July 2018.

Summary

In recent years, the use of NoSQL systems to store and exploit massive databases has been steadily increasing. These systems provide a great flexibility of use thanks to their "schema less" property. However, the schema of the NoSQL DB is necessary to allow users (especially decision-makers) to express their queries. In what follows, the notion of "NoSQL DB" is used to designate a DB managed by a NoSQL DBMS. The purpose of this article is to extract the schema from a document-oriented NoSQL DB. For this, we propose an incremental development process of the schema during the exploitation of the Database