Modélisation de la dynamique des territoires : méta-données et lacs de données dédiés à l'information spatiale

Rodrique Kafando, Rémy Decoupes, Lucile Sautot, Maguelonne Teisseire

Modélisation de la dynamique des territoires : méta-données et lacs de données dédiés à l'information spatiale

Rodrique Kafando [1,3], **Rémy Decoupes** [1], **Lucile Sautot** [2], **Maguelonne Teisseire** [1]

1. *TETIS, Univ Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier, France*
   prenom.nom@inrae.fr
2. *AgroParisTech, Montpellier, France*
   lucile.sautot@agroparistech.fr
3. *Montpellier Méditerranée Métropole, France*

RÉSUMÉ. *La gestion efficace d'un lac de données nécessite un système de gestion de méta-données performant. De nombreux travaux se sont penchés sur cet aspect en proposant des solutions. Néanmoins, peu de travaux se sont intéressés aux lacs de données dédiés aux informations spatiales. Pourtant, cette dimension géographique est fondamentale dès lors que l'on souhaite explorer les différentes trajectoires de projets d'aménagement au sein d'un même territoire. Dans cet article, nous nous intéressons tout particulièrement à la mise en oeuvre d'un lac de données pour la métropole de Montpellier. La solution conceptuelle proposée s'adosse à la norme ISO 19115 pour décrire des méta-données spatiales qui est étendue dans le cadre des lacs de données. L'implémentation basée sur HDFS et GeoNetwork est présentée et discutée. Le code source est également mis à disposition de la communauté.*

ABSTRACT. *Data lake management requires an efficient metadata management system. Some works have already addressed this aspect in order to describe the datasets recorded and ensure their proper use. However, little work has been done on data lake dedicated to spatial information. However, geographical dimension is fundamental when we wish to explore the different trajectories of development projects within a territory. In this article, we are particularly interested in the implementation of a data lake for Montpellier metropolis. The proposed conceptual solution is based on the ISO 19115 standard to describe extended spatial metadata within the context of data lakes. The implementation based on HDFS and GeoNetwork is presented and discussed.*

MOTS-CLÉS : *Lac de données spatial, Données hétérogènes, Dynamique Territoriale*

KEYWORDS: *Spatial data lake, Heterogeneous data, Territorial Dynamic*

1. Introduction

Selon (Albino *et al*., 2015), les villes intelligentes sont définies comme des villes équipées en hautes technologies, qui connectent les habitants, les informations et les éléments urbains afin de créer une ville durable, un contexte économique compétitif et innovant, et une meilleure qualité de vie. Ces dernières années, nous avons constaté une croissance exponentielle des nouvelles technologies et des services associés en lien avec les villes intelligentes (Kitchin, 2014 ; Al Nuaimi *et al*., 2015). L'ensemble de ces services génèrent un grand volume de données, qui caractérisent, d'un point de vue global, l'évolution et le comportement du territoire.
Dans un tel contexte, les travaux présentés sont issus de la collaboration d'un laboratoire de recherche pluridisciplinaire avec Montpellier Méditerranée Métropole (3M). Le principal besoin exprimé par les utilisateurs est d'arriver à explorer sémantiquement de grandes quantités de données disponibles au sein de leur organisation. Parmi ces données, certaines sont produites par les citoyens, d'autres par les différents services de la métropole et des municipalités associées (transport, tourisme, etc.). Il est donc difficile d'avoir une vue d'ensemble sur les informations à disposition.

Le principal inconvénient des outils existants est la difficulté pour les utilisateurs d'explorer de manière flexible un ensemble de données massives et hétérogènes. Plus précisément, les entrepôts de données (Devlin, Cote, 1996) sont trop rigides pour permettre aux utilisateurs de construire de nouvelles analyses qui n'auraient pas été prévues (Madera, Laurent, 2016). Pour résoudre ce problème, les lacs de données (Dixon, 2010) représentent un nouveau mode de gestion des données, avec un stockage total ou partiel des éléments associés (données et méta-données). Pour ces nouveaux systèmes, dont la théorisation est récente, il y a eu peu de travaux méthodologiques sur la conception de ces infrastructures de données, considérant qu'elles requièrent essentiellement des compétences techniques. Confrontés à la mise en place d'un lac de données en conditions réelles dans le contexte d'une ville intelligente, nous sommes amenés à contredire cette hypothèse. Dans cet article, nous nous intéresserons ainsi à la conception et à l'implémentation d'un lac de données, en partant d'une masse de données hétérogènes avec une forte composante spatiale provenant de notre cas d'étude. En nous inspirant de travaux précédents sur la normalisation de données spatiales (ISO/TC 211, 2019) et sur les lacs de données (Ravat, Zhao, 2019 ; Madera, Laurent, 2016 ; Sawadogo *et al*., 2019), nous développons une méthodologie de conception dédiée. Le code développé de notre proposition est mis à disposition de la communauté. Nous démontrons que les infrastructures type lac de données ne sont pas réservées aux experts mais peuvent être proposées à d'autres utilisateurs à la condition de leur fournir une interface adaptée.

L'article est organisé comme suit. Dans la Section 2, nous présentons les définitions et les travaux relatifs à la conception de lac de données et à la gestion de l'information spatiale. Les données utilisées dans le cadre de la collaboration avec 3M sont détaillées dans la Section 3. La méthodologie proposée est présentée dans la Section 4, suivie par une description de l'implémentation dans la Section 5. La Section 6 conclut ce travail avec la discussion des résultats et la présentation de travaux à venir sur une solution de lac de données spatiales.

2. Etat de l'art

Plusieurs systèmes de gestion et de stockage de données ont émergé pour supporter le Big Data (McAfee *et al.*, 2012). Parmi eux, nous pouvons citer les bases de données NoSQL (Not Only SQL) (Bruchez, 2015), les entrepôts de données (Kimball, Ross, 2011 ; Phipps, Davis, 2002) et les lacs de données (Russom, 2017 ; Hai *et al.*, 2016).

2.1. Entrepôts de données et lac de données

Les entrepôts de données ont été conçus comme une optimisation des bases de données relationnelles pour l'exécution de requêtes analytiques et sont utilisés comme support à la prise de décision dans les organisations. Les modèles conceptuels des entrepôts de données sont basés sur les concepts suivants : les faits (et mesures), les dimensions, les hiérarchies et les membres (Kimball, Ross, 2013). De fait, concevoir un entrepôt de données revient à définir l'espace des tableaux croisés possibles, qui vont être construits par les utilisateurs pour explorer les données. Les entrepôts de données permettent une exploration facilitée de volumineux jeux de données par les utilisateurs. Mais la mise en place d'un entrepôt de données implique une normalisation des données entrantes issues de sources variées (cette normalisation pouvant être automatisée via un ETL). Malgré quelques propositions intéressantes (voir par exemple (Oukid *et al.*, 2016) et (Minati *et al.*, 2006)), l'intégration de documents et d'images satellitaires dans le même entrepôt de données reste une tâche complexe. Ainsi, la mise en place d'un entrepôt de données nécessite un long processus de préparation des données.

La définition d'un lac de données a été proposée par (Dixon, 2010). Une comparaison détaillée avec les entrepôts de données a été réalisée dans (Madera, Laurent, 2016) puis reprise dans (Sawadogo *et al.*, 2019). Les lacs de données sont une solution récente qui a été développée pour répondre à la gestion des Big Data pour lesquelles les entrepôts de données montraient quelques faiblesses. Le principal problème rencontré avec les entrepôts de données est la gestion de données de natures hétérogènes. Un lac de données est une structure de stockage de données massives, qui intègre les données en provenance de différentes sources dans leur format natif, sans qu'il soit nécessaire de réaliser un traitement (Russom, 2017 ; Hai *et al.*, 2016). Selon (Sawadogo *et al.*, 2019). Un lac de données est un système évolutif de stockage et d'analyse de données, stockées dans leur format natif, destiné

à des spécialistes tels que des statisticiens, des analystes et des "data scientists". Les principales composantes et caractéristiques des lacs de données sont :

– un catalogue de méta-données qui facilite l'accès aux données et en assure la qualité,

– des outils de gestion des données,

– l'accessibilité aux utilisateurs,

– l'évolution possible des données,

– l'ingestion des données de toute nature,

– une organisation logique et physique.

Etant une nouvelle technologie Big Data, les lacs de données ont été étudiés dans de nombreux articles. En raison de leur capacité à gérer de larges volumes de données, structurées et non structurées, une étude exploratoire a été réalisée pour mieux comprendre l'utilisation des lacs de données dans le contexte industriel (Llave, 2018). Dans (Giudice *et al.*, 2019 ; Mehmood *et al.*, 2019), de nouvelles architectures de lac de données ont été conçues afin d'extraire des informations pertinentes d'un ensemble de données hétérogènes, en se basant sur les sources de ces données. Dans (Quix *et al.*, 2016), un système de gestion de méta-données générique et extensible pour les lacs de données (Generic and Extensible Metadata Management System for Data Lakes, GEMMS) a été développé, en premier lieu pour extraire des méta-données des sources et en second lieu, pour enrichir les sources de données en utilisant des informations sémantiques venant à la fois des données et des méta-données. De nombreux système de gestion de méta-données ont été ainsi proposés par la communauté, mais il reste encore des défis à relever dont en particulier la mise en lien sémantique des données (Nargesian *et al.*, 2019).

En nous basant sur ces travaux, nous définissons un lac de données comme une structure de stockage composée de jeux de données, ayant les caractéristiques précédemment citées et celles décrites dans la Section 4.

2.2. Information géographique

Plusieurs définitions ont été proposées pour le concept de territoire selon le domaine étudié. Dans (Moine, 2006), le territoire est considéré comme étant un système complexe et évolutif qui associe un ensemble d'acteurs, d'une part, et d'autre part, l'espace géographique que ces acteurs utilisent, développent et gouvernent. Dans (Simone *et al.*, 2018), les auteurs quant à eux, considèrent que le territoire est un ensemble composé de trois dimensions : l'espace géographique, le temps et les relations sociales. Ils définissent le territoire comme étant un système complexe situé dans un espace géographique spécifique émergeant de la co-évolution d'un ensemble de processus hétérogènes (anthropologico-culturel, relationnel, cognitif et économico-productif) qui caractérise cet espace d'une manière unique et non répétitive.

Tout en prenant en compte les définitions proposées dans l'état de l'art, nous considérons que le territoire est :

– un ensemble d'acteurs physiques et/ou juridiques. Physique dans le sens où il est habité par un ou plusieurs groupes de personnes interagissant les uns avec les autres, et juridique au sens où il est composé de plusieurs organisations politiques, économiques, etc.

– décrit par un ensemble d'informations géographiques, à savoir des entités spatiales, thématiques et temporelles qui interagissent entre elles. Ces informations évoluant dans le temps et dans l'espace.

Dans cette étude, nous nous focalisons principalement sur les informations géographiques produites et gérées au niveau de la Métropole de Montpellier qui est notre zone d'étude. Notre proposition est basée sur la norme ISO 19115 (ISO/TC 211, 2019) dédiée aux données spatiales (identification, étendue, qualité, contenu, référence géographique, etc.).

3. Données et utilisateurs du lac de données

Les jeux de données utilisés dans le cadre de notre étude sont constitués entre autres d'images satellites, de documents textuelles, de couches vectorielles et autres données telles que les données de transports, d'urbanisation, d'agriculture, de commerce, etc. Elles proviennent de sources différentes :

– la plate-forme opendata de la Métropole de Montpellier [1]. Elle regroupe un ensemble de données produites par la Métropole de Montpellier et qui sont mises à la disposition du grand public. La liste exhaustive des liens se retrouve dans le fichier datasources.csv présent dans le dépôt logiciel de notre implémentation (https://github.com/aidmoit/collect/blob/master/input/datasources.csv), accédé le 2020-02-19. Ces jeux de données sont publiées sous licence "Open Data Commons Open Database License" (ODbL).

– le web : nous avons constitué des corpus de données textuelles à partir du web. Les corpus sont construits en tenant compte des thématiques abordées que nous souhaitons étudier dans la phase de mise en relation.

– OpenStreetMap : nous a permis d'obtenir les étendues spatiales des lieux de notre cas d'étude (les communes de la métropole de Montpellier), accédé le 2020-02-19.

Notre solution peut être exploitée par deux types d'utilisateurs. Tout d'abord, les utilisateurs 'grand public' ou tout utilisateur, le système leur permet d'explorer et de récupérer des données présentes dans le lac à travers l'interface web offert par GeoNetwork sans avoir besoin de compétences sur l'exploitation des lac de données. Ensuite les utilisateurs expérimentés, qui en plus de l'exploration peuvent effectuer

1. Open Data 3M : http://data.montpellier3m.fr/

des traitements et des analyses directement sur le lac de données en utilisant des outils comme Apache Spark.

4. Architecture du lac de données spatiales

Dans cette section, nous proposons une vue générale de l'architecture d'un lac de données spatiales, avec pour objectif de fournir un guide pour reproduire une telle architecture. L'architecture proposée est prévue pour stocker les données produites et utilisées par la métropole de Montpellier. Ce cas d'étude implique plusieurs contraintes :

– la dimension spatiale des jeux de données et les analyses spatiales réalisées sont un élément important. Nous avons notamment besoin de stocker des images satellites.

– le système proposé doit être inter-opérable avec d'autres systèmes d'information, au niveau local, national et européen.

– les utilisateurs souhaitent explorer le lac de données pour y trouver des données pertinentes et découvrir de nouvelles connaissances.

L'architecture proposée est composée de trois parties principales : la section data, la section metadata et la section intermetadata. La section data, le coeur de la structure de stockage, est basée sur Hadoop Distributed File System (HDFS) (Shvachko *et al.*, 2010). Le choix de HDFS est motivé d'une part, par le fait qu'il permet de stocker les données dans leur format natif (contrairement au système de stockage clé-valeur), et d'autre part, de sa distributivité. Avec HDFS, il est possible détendre (parallèlement) plus facilement la capacité de stockage en cas de besoin et aussi d'effectuer des calculs distribués.

La section metadata est un catalogue de données (Lamb, Larson, 2016), qui décrit les données stockées dans le lac de données. La section intermetadata est une partie de la section metadata. Elle permet le stockage de relations entre jeux de données riches sémantiquement.

HDFS est un système performant pour le stockage de données massives et hétérogènes, mais ne peut pas être utilisé tel quel par nos utilisateurs. Les utilisateurs du lac de données ont besoin d'explorer le lac de données afin de trouver les jeux de données les plus pertinents vus leur requête, et éventuellement de découvrir de nouveaux jeux de données. Ces fonctionnalités (exploration, requêtage, découverte) sont supportées par le catalogue de données, qui offre une interface graphique simple pour accéder aux méta-données descriptives du contenu du lac de données.

Le modèle conceptuel que nous proposons est une extension de la norme ISO 19115 (ISO/TC 211, 2019). Cette norme inclut une description spatiale des données et sert de base à plusieurs profiles de méta-données (INSPIRE, Dublin Core) utilisés habituellement par les institutions publiques.

La FIGURE 1 est une vue générale du modèle conceptuel étendu proposé. Dans cette figure, la classe représentée en blanc est directement issues de la norme ISO

19115 (FIGURE 1) et les classes représentées en jaune constituent nos ajouts. Afin que les modèles restent lisibles, nous avons fait le choix de ne représenter que la classe principale de chaque package.

Dans la section data (FIGURE 2), nous définissons un lac de données comme un ensemble de ressources. Une ressource peut être un service (voir la norme ISO 19115) ou une série de données. Une série de données est composée d'un ou plusieurs jeux de données, qui partagent une caractéristique. Un jeu de données est une collection de données identifiables. Trois types de jeux de données ont été définis : document, vecteur et raster.

La section metadata décrit les fiches de méta-données associées à chaque ressource (voir FIGURE 3). Une fiche de méta-données est composée de :

– une identification (obligatoire) qui permet la différenciation des ressources par l'utilisateur ;

– une représentation spatiale (optionnelle), un système de coordonnées de référence (optionnel) et une emprise spatiale et/ou temporelle (optionnelle). Ces trois éléments permettent de décrire la spatialité de la ressource ;

– une description du contenu de la ressource (optionnelle) ;

– une généalogie (optionnelle), qui explique comment la ressource a été obtenue ;

– un ou plusieurs liens vers des ressources associées ;

– un système de référence (optionnel), qui identifie les systèmes de références spatiaux, temporels et paramétriques utilisés par cette ressource ;

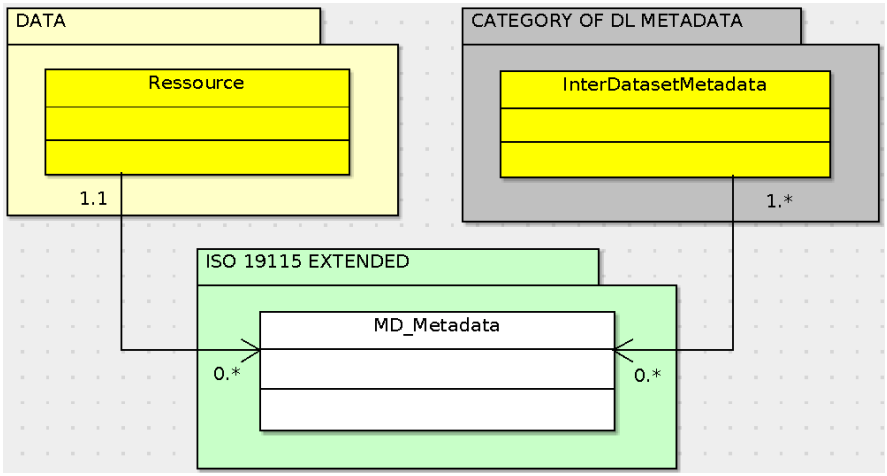
– une emprise, qui décrit l'emprise temporelle et spatiale de la ressource.

Enfin, la section intermetadata (voir FIGURE 4) décrit les relations entre les jeux de données et permet à l'utilisateur d'avoir une visibilité sur les données liées à sa requête initiale. Quatre types de relations ont été proposés, basés sur (Sawadogo *et al.*, 2019) : parenté, inclusion, similarité et regroupement thématique. Le modèle conceptuel proposé dans son ensemble permet de prendre en compte non seulement l'intégration des méta-données de données spatiales, mais aussi tout type de données stockées dans le lac.

5. Implémentation pour la Métropole de Montpellier : 3M (Montpellier Méditerranée Métropole)

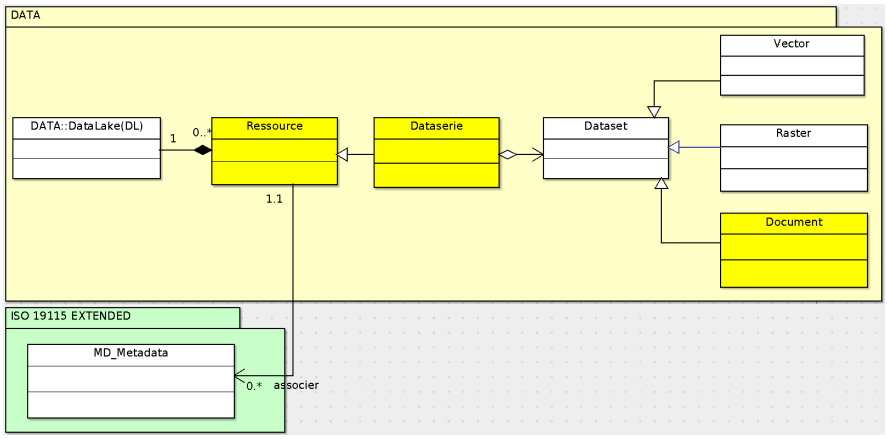
L'objectif de cette section est de présenter la mise en œuvre du lac de données pour la Métropole de Montpellier [2].

2. Montpellier Méditerranée Métropole (3M) : https://www.montpellier3m.fr/



Les classes blanches viennent de (ISO/TC 211, 2019), les classes jaunes ont été ajoutées à la norme.

Figure 1. Vue générale du modèle conceptuel proposé

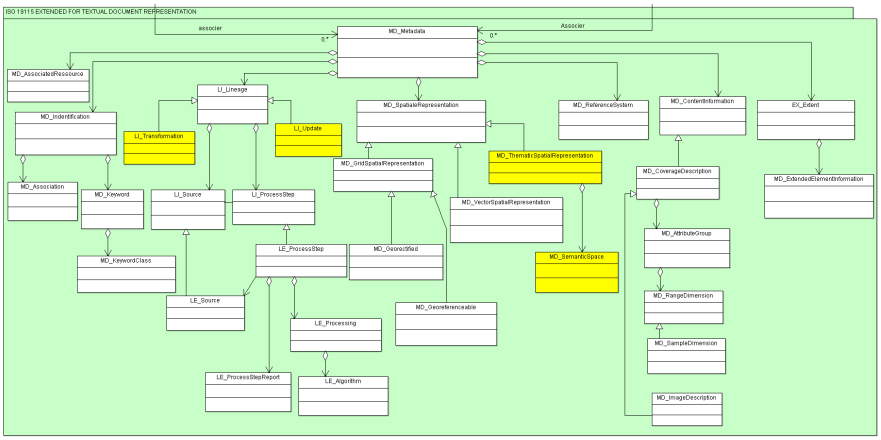


Les classes blanches viennent de (ISO/TC 211, 2019), les classes jaunes ont été ajoutées à la norme.

Figure 2. Section Data

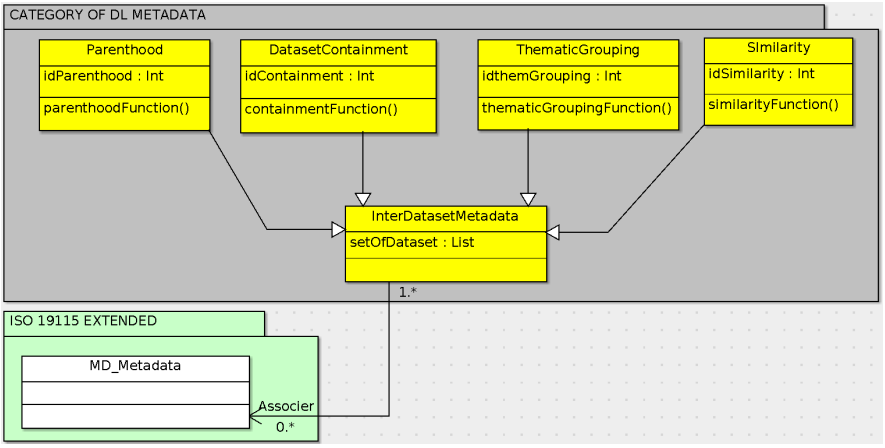
5.1. Présentation de l'infrastructure système

Comme présenté dans la précédente section, un lac de données est composé de deux sous-systèmes. La partie donnée est assurée par la mise en place d'un système de fichiers distribués. Cette partie est enrichie par un gestionnaire de méta-données qui constitue le deuxième composant du lac de données.
Dans notre implémentation, la partie données repose sur le système de fichiers distribués HDFS (Hadoop Distributed File System), utilisant la technologie du projet



Les classes blanches viennent de (ISO/TC 211, 2019), les classes jaunes ont été ajoutées à la norme.

Figure 3. Section metadata



Les classes blanches viennent de (ISO/TC 211, 2019), les classes jaunes ont été ajoutées à la norme.

Figure 4. Section intermetadata

HADOOP [3]. Notre ensemble de serveurs HDFS, appelé cluster HDFS, est constitué de trois machines ayant des rôles différents : deux nœuds de stockage (appelé datanode dans la terminologie Hadoop) et un nœud maître (appelé namenode). Ce dernier gère les interactions avec les utilisateurs ainsi que la distribution et la réplication des données sur les datanodes.

Malgré les différentes fonctionnalités offertes à l'utilisateur, le namenode HDFS ne propose ni de système d'indexation de données ni de moteur de recherche. Pour

3. Hadoop : http://hadoop.apache.org/

répondre à ce besoin, l'administrateur d'un lac de données doit avoir recourt à un outil de gestion de méta-données de type ElasticSearch, construit sur le projet Apache Lucene (Chen *et al*., 2017) et (John, Misra, 2017). Nous proposons d'utiliser l'outil GeoNetwork [4]. Cet outil open-source embarque un moteur de recherche Apache Lucene et a l'avantage d'implémenter le modèle de la norme ISO 19115. Ainsi, le serveur GeoNetwork sauvegarde les méta-données obligatoires et optionnelles, telles que décrites dans la précédente section, et conserve les liens permettant de télécharger les données stockées dans le cluster Hadoop via le namenode. Le moteur de recherche de GeoNetwork permet à l'utilisateur de faire des recherches croisées sur les trois dimensions : spatiales, temporelles et thématique. Le résultat de la recherche est une collection de jeux de données répondant à l'intersection des critères de la requête.

5.1.1. Insertion et indexation des données dans le lac de données

Comme illustrée par la FIGURE 5, l'insertion de jeux de données dans le lac de données se déroule en cinq étapes. Les deux premières étapes sont réalisées manuellement, les trois dernières étapes sont, elles, automatisées.
En effet, l'administrateur doit remplir un tableur au format CSV [étape 1]. Ses colonnes sont les méta-données telles que décrites dans la section précédente ainsi qu'un lien HDFS pour indiquer l'emplacement du jeu de données. Chaque ligne représente un jeu de données pour lequel l'administrateur doit compléter les méta-données.
Puis l'administrateur lance un programme, voir section "Accès aux logiciels et données de l'implémentation", depuis sa machine (ou un serveur du lac de données) [étape 2]. Au début de son exécution, le programme va lire [étape 3] et extraire les informations du fichier CSV. Puis le programme, télécharge les jeux de données et les insère dans le cluster HDFS [étape 4]. Enfin, le programme crée une fiche de méta-données de type ISO 19115 et l'insère dans GeoNetwork afin de bénéficier de son indexation et de son moteur de recherche [étape 5].

5.1.2. Découverte et accès aux jeux de données

L'utilisateur peut parcourir, découvrir, faire une requête et accéder aux jeux de données en utilisant le moteur de recherche de GeoNetwork. Les recherches peuvent être une combinaison de critères sur les trois dimensions :

– sémantique : basé sur les mots clés ou bien sur une recherche en texte plein sur le titre, le résumé ou la généalogie de la fiche de méta-données.

– spatialisée : en dessinant une emprise spatiale directement sur la carte afin de filtrer les jeux de données qui intersectent l'étendue géographique voulue.

– temporelle : filtrer sur les années, mois et jour.

4. GeoNetwork : application web de catalogue de données spatialisées. https://geonetwork-opensource.org/

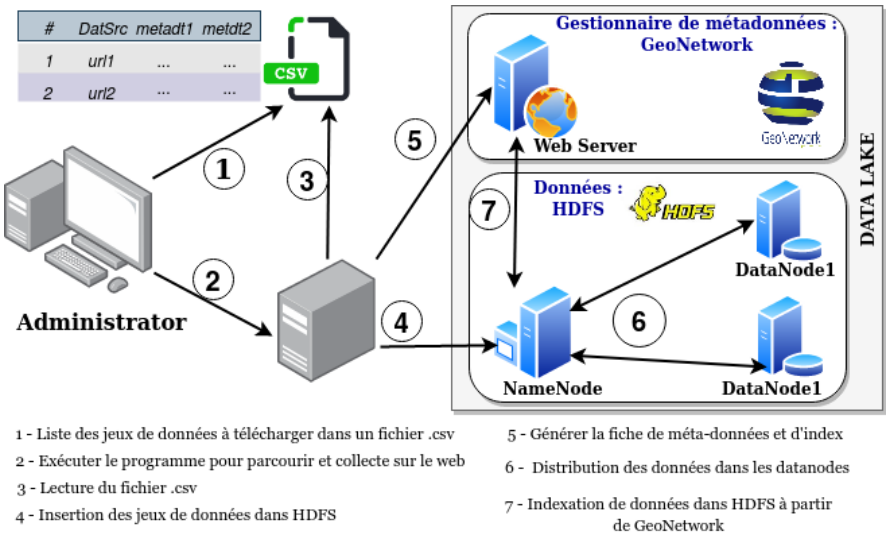


Figure 5. Insertion et indexation de jeux de données dans le lac de données

GeoNetwork retourne une collection de fiches de méta-données décrivant les jeux de données qui respectent les critères de recherche. En parcourant les jeux de données, l'utilisateur peut accéder à tous les fichiers de données stockés dans le cluster HDFS sans avoir besoin de connaître la syntaxe d'interrogation d'Hadoop (FIGURE 6).

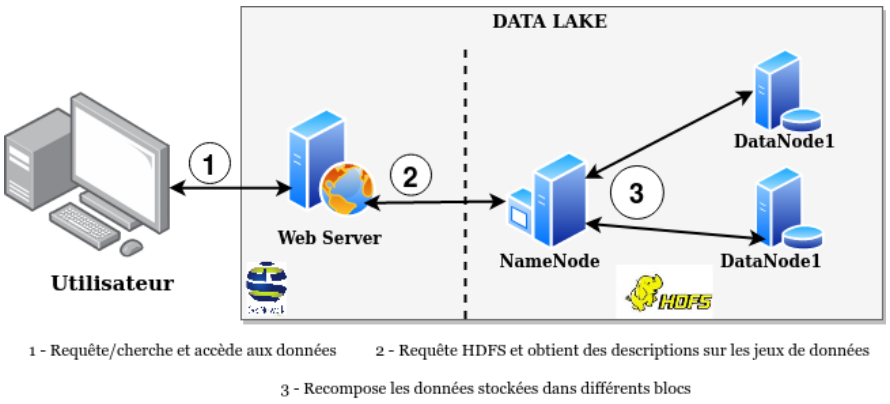


Figure 6. Recherche de jeux de données

5.2. Automatisation du déploiement du lac de données et de l'ajout de contenu

L'installation d'un cluster Hadoop peut s'avérer complexe. À des fins de reproductibilité, nous avons automatisé son installation, via les outils Vagrant [5] et Ansible [6].
Cette automatisation construit et configure quatre machines virtuelles dont les trois premières appartiennent au cluster Hadoop et la dernière héberge le serveur GeoNetwork. Ceci permet d'instancier facilement notre lac de données.
Nous avons aussi automatiser l'ajout de jeux de données ainsi que leur indexation. Cette automatisation prend, cette fois-ci, la forme d'un script python qui analyse les informations provenant d'un fichier CSV, y extrait des liens de téléchargement qui lui permettent de télécharger les jeux de données voulus. Ensuite, le script insère les fichiers de données dans le cluster HDFS sans les organiser dans une arborescence. Enfin, il crée des fiches de méta-données, les insère dans GeoNetwork afin de bénéficier de son moteur de recherche.

5.2.1. Déploiement automatique du cluster HDFS et du serveur GeoNetwork

L'ensemble du lac de données, c'est-à-dire le cluster HDFS et le serveur GeoNetwork, est déployé et maintenu grâce à l'utilisation de projets opensource notamment les suivants :

– Debian: Système d'exploitation utilisé par les 4 machines virtuelles. Nous avons utilisé la version 9 et non pas la dernière version à cause de problème de compatibilité avec la version java nécessaire à Hadoop et à GeoNetwork. La dernière version (version 10) de debian ne maintient plus cette version de java (version 9),

– VirtualBox comme hyperviseur,

– Vagrant comme système de gestion de configuration des machines virtuelles (système d'exploitation utilisé, configuration réseau, script d'installation, ...),

– Ansible comme un outil de déploiement d'application et de gestion de configuration.

Les codes sources de ce projet peuvent être retrouvés dans la section 5.4. Grâce à ces dépôts logiciels, le cluster HDFS peut être déployé et configuré en quatre commandes et le serveur GeoNetwork en une commande.

De plus amples informations ou instructions techniques peuvent être retrouvées dans le fichier README.md du dépôt logiciel de notre projet. Si les variables par défaut, proposé par le dépôt logiciel, sont conservées, le cluster HDFS peut être accessible de manière graphique en se connectant à son serveur web à l'adresse http://namenode:9870 (ou http://10.0.0.10:9870). D'autres informations, telles que la santé du cluster, ou l'accès aux logs ou bien encore l'accès au système de fichiers HDFS peuvent être aussi retrouvé via cette interface. Le serveur GeoNetwork est

5. Vagrant : https://www.vagrantup.com/
6. Ansible : https://www.ansible.com/

quant à lui accessible à l'adresse http://aidmoit-geonetwork:8080/geonetwork (http://10.0.0.9:8080/geonetwork).

5.2.2. Ajout de données dans le lac de données

L'ajout de données dans le lac de données a, lui aussi, été automatisé. Deux scripts, en python et en R, ont été écrits. La complexité qu'induit le développement d'un outil basé sur deux langages de programmation différents a été motivé par les couvertures de fonctionnalités des librairies de chaque langage. En effet, python offre des libraires remarquables pour interagir avec HDFS alors que R propose des modules intéressants pour gérer des fiches de méta-données compatibles ISO 19115. Afin de faciliter l'utilisation de ces deux scripts, le programme R a été encapsulé dans le code python, permettant, ainsi, à l'administrateur, de ne lancer qu'un seul programme.

Comme mentionné auparavant, l'ensemble des fichiers sources est disponible dans la section 3. L'environnement requis pour faire fonctionner ces scripts a été décrit dans le fichier "requirement.txt" présent à la racine du dépôt du logiciel. Les instructions d'installation et de lancement sont, quant à eux, présentées dans le fichier README.md.

Le script principal écrit en python opère en cinq étapes. Premièrement, il extrait les informations contenu dans le fichier datasources.csv comme le fournisseur de données, le nom du jeu de données et les mots clés associés. Le script, dans une deuxième étape, parcourt le site web du fournisseur de données afin de créer un fichier json contenant l'ensemble des liens de téléchargement des jeux de données. Ensuite, tous les fichiers constituant les jeux de données sont téléchargés, puis enregistré dans le cluster HDFS, ce qui constitue les troisième et quatrième étapes. Enfin, le script R est lancé afin de créer des fiches de méta-données au standard ISO 19139 qui sont, ensuite, ingérées par GeoNetwork.

Les fichiers de données sont facilement récupérables à partir de GeoNetwork. En effet, le namenode du cluster HDFS offre une interface de programmation de type API REST (Application Programming Interface - REpresentational state transfer) permettant une abstraction complète des commandes HDFS.

5.3. Illustration d'une recherche d'un utilisateur

L'utilisateur de notre lac de données peut créer des requêtes complexes mélangeant les trois dimensions : spatiale, temporelle et sémantique. Ces requêtes se construisent à travers l'utilisation du moteur de recherche de GeoNetwork qui offre plusieurs méthodes de composition de recherche. En effet, la requête peut être élaborée via une combinaison de recherche en texte libre (sur les trois dimensions) et/ou par mot clés (aussi sur les trois dimensions) et/ou par le dessin d'une étendue spatiale sur une carte (uniquement dimension spatiale).

5.4. Accès aux logiciels et données de l'implémentation

5.4.1. Infrastructure système

L'infrastructure du lac de données, c'est-à-dire, le cluster HDFS et le serveur GeoNetwork, est instanciable à travers l'utilisation de quatre machines virtuelles. L'installation et le lancement de ces machines ont été automatisés. Le dépôt logiciel est le suivant : https://github.com/aidmoit/ansible-deployment. Les instructions d'utilisation sont décrites dans le fichier README.md du dépôt. Le numéro de commit utilisé pour notre implémentation est le suivant :
65de950a336ee2828cdb19db976b7946649c439c
Le dépôt est publié sous la licence GPL-3.

5.4.2. Logiciel et flux de traitement

L'ensemble des logiciels pour le téléchargement des données, leurs ajouts dans le cluster HDFS et leurs descriptions dans le GeoNetwork sont orchestrés par un script python. Toutes les ressources nécessaires à son exécution sont disponibles à travers ce dépôt : https://github.com/aidmoit/collect. Le numéro de commit utilisé pour notre implémentation est le suivant : da9f63f9287a191d7e8fd24884a731bae02e1034.

Les codes sont distribués sous la licence GPL-3. Ils exploitent deux paquets R : geometa (Blondel, 2019) et geonapi (Blondel, 2018) diffusés sous licence MIT. Enfin, les scripts utilisent les données d'OpenStreetMap [7], ces données sont publiées sous licence "Open Data Commons Open Database License" (ODbL).

6. Conclusion et Perspectives

Dans cet article, nous avons présenté une nouvelle méthodologie de conception et d'implémentation de lac de données spatiales. La principale contribution est l'introduction de la dimension spatiale dans le processus de conception de lac de données, basée sur un système de méta-données géographiques. Nous avons également montré que les lacs de données peuvent être orientés vers les utilisateurs finaux, ce qui est possible en mettant en place une interface de requêtes. Nos travaux futurs seront dédiés à l'analyse et à la mise en lien des données stockées dans le lac de données. Les questions à traiter seront :

1. Comment des données hétérogènes peuvent être liées sémantiquement pour une analyse des phénomènes spatio-temporels complexes qui ont lieu sur un territoire ?

2. Quelles méthodes originales de fouille de données faut-il utiliser pour analyser des données hétérogènes massives ?

Atteindre ces objectifs nous permettra d'une part de décrire des relations riches entre les données selon les thématiques et le contexte spatio-temporel, et d'autre part de contribuer à la description de l'évolution d'un territoire. En d'autres termes, le

7. OpenStreeMap: https://www.openstreetmap.org

concept de lacs de données spatiales devient un élément central dans le dispositif des villes intelligentes.

Remerciements

Ces travaux ont été partiellement financés par Montpellier Méditerranée Métropole et le Projet Songe (FEDER et Région Occitanie).

Bibliographie

Albino V., Berardi U., Dangelico R. M. (2015, janvier). Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, vol. 22, n° 1, p. 3–21. Consulté sur http://www.tandfonline.com/doi/full/10.1080/10630732.2014.942092

Al Nuaimi E., Al Neyadi H., Mohamed N., Al-Jaroodi J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications*, vol. 6, n° 1, p. 25.

Blondel E. (2018, août). *geonapi: R interface to geonetwork api.* Zenodo. Consulté sur https://doi.org/10.5281/zenodo.1345102

Blondel E. (2019, octobre). *geometa: Tools for Reading and WritingISO/OGC Geographic Metadata in R.* Zenodo. Consulté sur https://doi.org/10.5281/zenodo.3524348

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

Chen D., Chen Y., Brownlow B. N., Kanjamala P. P., Arredondo C. A. G., Radspinner B. L. *et al*. (2017, April). Real-time or near real-time persisting daily healthcare data into hdfs and elasticsearch index inside a big data platform. *IEEE Transactions on Industrial Informatics*, vol. 13, n° 2, p. 595-606.

Devlin B., Cote L. D. (1996). *Data warehouse: from architecture to implementation*. Addison-Wesley Longman Publishing Co., Inc.

Dixon J. (2010, octobre). *Pentaho, Hadoop, and Data Lakes.* Consulté sur https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/

Giudice P. L., Musarella L., Sofo G., Ursino D. (2019). An approach to extracting complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake. *Information Sciences*, vol. 478, p. 606–626.

Hai R., Geisler S., Quix C. (2016). Constance: An intelligent data lake system. In *Proceedings of the 2016 international conference on management of data*, p. 2097–2100.

ISO/TC 211. (2019). *Norme iso 19115-1:2014. geographic information - metadata - part 1: Fundamentals. technical report, international organization for standardization, 2019.* International Organization for Standardization.

John T., Misra P. (2017). *Data lake for enterprises: Lambda architecture for building enterprise data systems*. Packt Publishing.

Kimball R., Ross M. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons.

Kimball R., Ross M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons.

Kitchin R. (2014). The real-time city? big data and smart urbanism. *GeoJournal*, vol. 79, nº 1, p. 1–14.

Lamb I., Larson C. (2016). Shining a light on scientific data: Building a data catalog to foster data sharing and reuse. *Code4Lib Journal*, nº 32.

Llave M. R. (2018). Data lakes in business intelligence: reporting from the trenches. *Procedia computer science*, vol. 138, p. 516–524.

Madera C., Laurent A. (2016). The next information architecture evolution: the data lake wave. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*, p. 174–180. ACM.

McAfee A., Brynjolfsson E., Davenport T. H., Patil D., Barton D. (2012). Big data: the management revolution. *Harvard business review*, vol. 90, nº 10, p. 60–68.

Mehmood H., Gilman E., Cortes M., Kostakos P., Byrne A., Valta K. *et al.* (2019). Implementing big data lake for heterogeneous data sources. In *2019 ieee 35th international conference on data engineering workshops (icdew)*, p. 37–44.

Minati L., Ghielmetti F., Ciobanu V., D'Incerti L., Maccagnano C., Bizzi A. *et al.* (2006). Bio-image warehouse system: Concept and implementation of a diagnosis-based data warehouse for advanced imaging modalities in neuroradiology. *Journal of Digital Imaging*, vol. 20.

Moine A. (2006). The territory as a complex system: an operational concept for land planning and geography (le territoire comme un systeme complexe: un concept opératoire pour l'aménagement et la géographie). *Esp. Géogr*, vol. 2, nº 35, p. 115.

Nargesian F., Zhu E., Pu K. Q., Miller R. J., Arocena P. C. (2019). Data lake management: Challenges and opportunities. , vol. 12, nº 12, p. 4.

Oukid L., Boussaid O., Benblidia N., Bentayeb F. (2016). Tlabel: A new olap aggregation operator in text cubes. *International Journal of Data Warehousing and Mining*, vol. 12, nº 4, p. 54-74.

Phipps C., Davis K. C. (2002). Automating data warehouse conceptual schema design and evaluation. In *Dmdw*, vol. 2, p. 23–32.

Quix C., Hai R., Vatov I. (2016). Metadata extraction and management in data lakes with gemms. *Complex Systems Informatics and Modeling Quarterly*, nº 9, p. 67–83.

Ravat F., Zhao Y. (2019). Data Lakes: Trends and Perspectives. In *International Conference on Database and Expert Systems Applications*, p. 304–313. Springer.

Russom P. (2017). Data lakes: Purposes, practices, patterns, and platforms. *TDWI White Paper*.

Sawadogo P. N., Scholly E., Favre C., Ferey E., Loudcher S., Darmont J. (2019). Metadata systems for data lakes: models and features. In *European conference on advances in databases and information systems*, p. 440–451.

Shvachko K., Kuang H., Radia S., Chansler R. *et al.* (2010). The hadoop distributed file system. In *Msst*, vol. 10, p. 1–10.

Simone C., Barile S., Calabrese M. (2018). Managing territory and its complexity: a decision-making model based on the viable system approach (vsa). *Land use policy*, vol. 72, p. 493–502.