

Architecture répartie à large échelle pour le traitement parallèle de requêtes de biodiversité

Ndiouma Bame¹ Hubert Naacke², Idrissa Sarr³, Samba Ndiaye¹

¹Département de
mathématique-informatique
UCAD
Dakar SENEGAL
ndioubame@yahoo.fr
samba.ndiaye@ucad.edu.sn

²LIP6
UPMC Sorbonne
Universités
France
Hubert.Naacke@lip6.fr

³LARIM
UQO
Canada
idrissa.sarr@lip6.fr

RÉSUMÉ.

Le portail du GBIF qui contient une description de la plupart des collections de données de la biodiversité mondiale, est confronté à des problèmes de disponibilité et d'expressivité limitée des requêtes liés à un nombre d'utilisateurs grandissant et manifestant sans cesse de nouveaux besoins. Pour faire face à ces problèmes, nous envisageons une solution qui passe à l'échelle avec un coût relativement faible. Dans cette perspective, nous proposons une architecture décentralisée et non intrusive pour interroger les données du GBIF en s'appuyant sur une infrastructure Cloud. Nous définissons une stratégie de répartition dynamique des données, adaptée au contexte du GBIF. Nous démontrons la faisabilité de notre approche par l'implémentation d'un prototype exécutant des requêtes jusqu'ici non supportées par le GBIF.

ABSTRACT.

The GBIF portal that contains a description of most of the global biodiversity data faces two problems, namely the data availability and a poor expressiveness of queries mainly due to a growing number of users which keep expressing new needs. To deal with these problems, we envision a scalable and relatively low cost solution. With this in mind, we propose a non-invasive and decentralized architecture for processing GBIF queries over a cloud infrastructure. We define a dynamic strategy for data distribution that fits the GBIF requirements. We demonstrate the feasibility of our solution by a prototype implementation which allows for processing extra query types, up to now unsupported by the GBIF portal.

MOTS-CLÉS : masses de données, réplication et distribution de données, nuage informatique, cloud computing, traitement des requêtes, GBIF.

KEYWORDS: big data, data replication and distribution, cloud computing, query processing, GBIF.

1. Introduction

Le GBIF est un consortium international visant à fédérer les données de biodiversité à l'échelle mondiale [1][3]. Il est reconnu comme étant la référence, pour les données primaires de biodiversité, sur laquelle s'appuient les autres initiatives (e.g., LifeWatch, GEOBON). La base de données du GBIF est complétée continuellement par les correspondants nationaux du consortium. Elle contient aujourd'hui plus de 300 millions d'enregistrements. Sa taille croît continuellement, et atteindra prochainement plusieurs téraoctets. Avec un nombre croissant de fournisseurs qui ajoutent de nouvelles données et d'utilisateurs qui interrogent la base, l'infrastructure actuelle peine à servir toutes les demandes en un temps raisonnable. Ce qui pose un réel problème de disponibilité des données tout en empêchant un usage réellement interactif des données du GBIF. D'autre part, les infrastructures informatiques sont en pleine évolution : les nuages informatique (*cloud*) sont omniprésents et permettent d'accéder à des ressources quasi-illimitées de stockage et de calcul [10]. Ceci incite à concevoir de nouvelles solutions pour la gestion de gros volumes de données, garantissant des accès rapides et un coût relativement abordable en fonction de la charge applicative [16] [8] [9]. De plus, le GBIF ne supporte, à ce jour, que le traitement de quelques requêtes prédéfinies, bien que les usagers ne cessent d'exprimer de nouveaux besoins pour d'autres types de requêtes plus complexes. Le premier objectif de cet article est d'augmenter l'expressivité des requêtes proposées aux utilisateurs. Le deuxième objectif est d'apporter plus de disponibilité au GBIF, en proposant une solution garantissant qu'une requête est traitée efficacement bien que le nombre et la complexité des requêtes augmentent.

L'article est organisé comme suit : la section 2 présente le contexte applicatif du GBIF. La section 3 identifie les limites de la solution actuelle. La section 4 propose une architecture décentralisée et flexible pour le traitement des requêtes et montre une instantiation de cette architecture sur une infrastructure de type Cloud.

2. Le portail du GBIF : architecture et requêtes

Le portail du GBIF fournit un accès électronique aux collections de données primaires sur la biodiversité [5]. Le portail intègre la description de la plupart des collections existant à travers le monde. L'informatisation des collections (recensement, numérisation, standardisation) est supervisée par les correspondants nationaux.

2.1. Architecture du portail du GBIF

L'architecture du portail du GBIF est fondée sur les *nœuds nationaux*. Un nœud national dispose de ressources informatiques dédiées à l'intégration des données nationales au

sein du portail. Le portail réalise deux services distincts : premièrement, l'intégration des données et, deuxièmement, l'interrogation des données intégrées. (i) L'architecture pour l'intégration des données dans la base centrale du portail est composée des nœuds nationaux et du nœud central du portail. Un protocole spécifie l'envoi des données vers le nœud central. (ii) L'architecture pour l'interrogation des données est actuellement centralisée ; toutes les requêtes sont traitées par le nœud central. On constate que les nœuds nationaux ne sont pas impliqués dans l'interrogation des données du portail (quelques nœuds nationaux proposent l'accès à des données supplémentaires non présentes dans le portail). Intuitivement, il semble que les nœuds nationaux pourraient participer davantage à l'interrogation des données afin d'améliorer le service d'interrogation offert par le portail. Cette intuition, à l'origine de notre travail, nous a motivé à définir une architecture décentralisée pour l'interrogation des données du portail. Cela nécessite tout d'abord d'étudier les fonctionnalités du service d'interrogation en précisant sur quel schéma de données les requêtes sont exprimées, et quelles sont les requêtes possibles.

2.2 Schéma des données

Nous décrivons brièvement le schéma des données du portail du GBIF. La base contient un ensemble de collections. Une collection contient un ensemble d'*occurrences*. Une occurrence décrit l'observation d'un spécimen. Une occurrence contient de nombreux champs taxinomiques ainsi qu'un géo-référencement. Une occurrence constitue l'unité d'information élémentaire de la base de données. Par ailleurs, la provenance des collections est décrite : une collection appartient à un fournisseur, lui-même rattaché à un pays. En termes de taille, la base de données contient plus de 300 millions d'occurrences issues de 1200 collections et 360 fournisseurs [3]. La taille totale de la base est de plusieurs centaines de giga octets.

2.3 Requêtes supportées par le portail

Le portail du GBIF permet de rechercher des occurrences de spécimens. Le portail propose deux types de requêtes : (i) une requête par navigation selon un pays, une collection ou une structure taxinomique ; (ii) une requête conjonctive multi critères portant sur un nombre fini de champs (nom scientifique, date, localisation géographique, etc..). Le résultat d'une requête a une structure fixe : c'est un ensemble d'occurrences avec tous leurs champs. Le portail ne supporte aucun autre type de requête, à l'heure actuelle.

Par ailleurs, le portail propose deux types d'interfaces pour l'interrogation : une interface de type IHM (interface homme machine à travers un navigateur) et une interface programmatique permettant d'automatiser l'interrogation en invoquant un service web. Le service web peut être ainsi utilisé directement par de nombreuses applications tierces qui étudient divers aspects de la biodiversité. La facilité d'accès

automatisé accroît fortement le nombre de requêtes que doit traiter le portail, ce qui accentue le besoin pour une solution de traitement de requêtes plus efficace.

3. Problèmes liés au fonctionnement actuel du GBIF

Les problèmes identifiés dans le fonctionnement actuel du GBIF peuvent être résumés en deux points : (i) une limitation de l'expressivité des requêtes; (ii) une architecture centralisée qui peut être source de congestion et ne permet pas une utilisation efficace de toutes les ressources associées aux GBIF. Nous détaillons dans les deux sous-sections suivantes ces deux problèmes avant de proposer dans la section suivante une solution à ceux-ci.

3.1. Besoins d'autres requêtes

En plus des requêtes prédéfinies, les usagers ont exprimés leurs besoins pour d'autres types de requêtes plus expressives, mais qui ne sont pas supportées par le portail actuel du GBIF. Généralement, les usagers ont besoin de combiner les résultats de plusieurs requêtes simples (supportées par le portail) pour effectuer des opérations ensemblistes non supportées par le portail telles que l'union, l'intersection, la différence entre plusieurs ensembles d'occurrences. Les usagers ont aussi besoin d'agréger et de dénombrer des occurrences selon des critères spatiaux (e.g. superposer les distributions géographiques de deux espèces). Par exemple, des usagers ont besoin des requêtes suivantes :

R1) Déterminer l'ensemble des fournisseurs qui partagent des informations sur une espèce de nom n : Pour chaque fournisseur afficher son nom et le nombre d'occurrences de n qu'il fournit. Trier l'ensemble par ordre croissant du nombre d'occurrences de n .

R2) Déterminer les espèces de nom n existant dans chacun des cinq continents avec au moins m occurrences n dans chaque continent, et qui sont partagées par au moins p fournisseurs différents dans ce même continent.

De manière générale, l'ensemble des requêtes dont ont besoin les utilisateurs est inclus dans l'ensemble des requêtes pouvant être exprimées en SQL sur la base du GBIF. Toutefois, un utilisateur a généralement besoin d'accéder à une portion relativement faible des données ; nous exploiterons par la suite cette propriété pour optimiser la fragmentation des données en fonction des requêtes.

3.2. Passage à l'échelle des données et des traitements

Le portail du GBIF fournit un accès centralisé à toutes les données intégrées par le GBIF. Plusieurs problèmes se posent pour faire face à la demande croissante pour stocker et interroger toujours plus de données.

- **Gérer la croissance des données** : Le nombre d'enregistrements accessibles dans le portail est passé de 163 millions en décembre 2008 à 317 millions en décembre 2011. La croissance du nombre d'enregistrement s'accroît à l'heure actuelle pour atteindre prochainement le milliard d'enregistrements.

-**Gérer la croissance des requêtes** : Le nombre de requêtes par jour s'accroît à cause du nombre croissant d'utilisateurs (+25% entre janvier et décembre 2009 [2]), et de l'accès automatisé depuis des applications tierces. De plus, dès lors que le portail complètera ses fonctionnalités de requêtes pour mieux satisfaire les besoins des utilisateurs, le portail devra traiter des requêtes de plus en plus complexes.

-**Améliorer la disponibilité des données** La disponibilité du portail est sa capacité à répondre à toutes les requêtes en un temps raisonnable. Les freins à la disponibilité sont doubles : d'une part, la capacité de traitement de requêtes du portail ne lui permet pas de répondre à toutes les demandes suffisamment rapidement. D'autre part, certaines données, (bien que déjà informatisées sur un nœud national) n'ont pas encore achevé le processus d'intégration dans le portail et ne sont pas disponibles au moment où l'utilisateur pose sa requête.

4. Proposition d'une solution décentralisée et non intrusive

Nous proposons une solution pour répartir le traitement des requêtes et les données interrogées sur plusieurs machines afin d'allouer plus de ressources de stockage et de calcul. L'objectif étant de satisfaire les demandes croissantes des utilisateurs tout en évitant la concentration des demandes qui est une source de congestion.

4.1. Méthode de résolution

Nous répartissons les données et le traitement des requêtes, selon les principes suivants :

- (i) Fragmenter dynamiquement les données, à la demande, en fonction des requêtes non connues a priori.
- (ii) Répliquer et distribuer les fragments de données sur plusieurs serveurs de données, afin de paralléliser davantage les requêtes : parallélisme inter-requête pour améliorer le débit d'exécution des requêtes simples accédant à des fragments disjoints, et parallélisme intra-requête pour accélérer le traitement des requêtes complexes.
- (iii) Distribuer les données à proximité des utilisateurs pour réduire le coût de transfert du résultat des requêtes.
- (iv) Dissocier les deux étapes d'intégration et d'exploitation. La mise à jour des données et l'intégration de nouvelles données se font en dehors des périodes d'exploitation, afin de ne pas ralentir le traitement des requêtes. Ainsi, les requêtes accèdent à un cliché des

données (*snapshot*) suffisamment récent pour convenir aux utilisateurs. Dans le cas de l'analyse de données de biodiversité, un cliché demeure récent pendant plusieurs jours.

(v) Concevoir un service de requêtes qui relaie et complète le portail actuel, sans intrusion. Des fonctionnalités sont ajoutées pour évaluer les requêtes complexes de manière décentralisée.

4.2. Architecture et fonctionnement du service de requêtes

La figure 1 décrit l'architecture proposée. Le *service de requêtes complexes* est associé à plusieurs machines P_i participantes, chacune capable de traiter une requête au moyen d'un serveur de données relationnelles BD_i . Le service de requêtes est le point d'entrée pour les *utilisateurs*, il contient l'état de tous les P_i , en particulier, les données et les requêtes traitées sur chaque P_i . Avant de traiter une requête, un participant P_i invoque le *portail actuel du GBIF* pour récupérer des données dont il a besoin, et les répliquer dans BD_i . Le *dictionnaire* du portail permet à P_i de formuler correctement l'appel de service. Si la donnée requise par P_i est déjà dans la BD_j d'un autre P_j , alors le transfert s'effectue de P_j vers P_i pour éviter de solliciter le portail.

Lorsqu'un utilisateur pose une requête, le service de requêtes complexes choisit le P_i optimal en fonction du coût des transferts de données (entre le portail et P_i , puis entre P_i et l'utilisateur) et de la charge des P_i . Puis il délègue la requête au P_i choisi. Ainsi, le portail est déchargé du traitement des requêtes. Il alloue ses ressources principalement à la tâche d'intégration des données qu'il récupère auprès des nœuds nationaux.

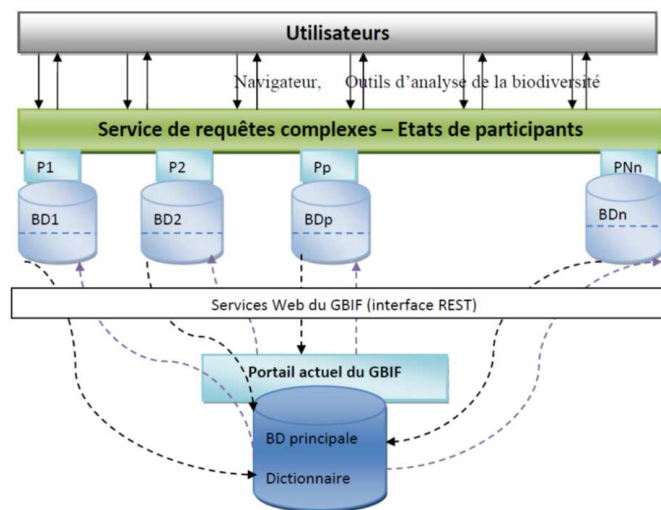


Figure 1: Architecture distribuée pour traiter les requêtes du GBIF

4.3. Instanciation de l'architecture sur un cloud

Nous avons commencé l'implémentation d'un premier prototype afin de démontrer la faisabilité de notre solution. Pour cela, nous nous sommes appuyés sur HadoopDB [12][13], un récent logiciel open source et modulaire pour le traitement efficace de requêtes SQL sur des grandes bases de données. HadoopDB est composé d'un module appelé Hive [11] [7] recevant des requêtes SQL, et de plusieurs modules Hadoop [6] chargés de calculer le résultat de la requête. Hive instancie la couche *service de requêtes complexes* de la Fig. 1. Hive découpe une requête SQL en plusieurs tâches qu'il envoie aux modules Hadoop (correspondant aux P_i de la Fig. 1). Un P_i reçoit une tâche qu'il retransforme en SQL afin de la traiter grâce à son SGBD H2 embarqué (correspondant au BD_i de la Fig. 1). Nous avons choisi le SGBD open source H2 [15] car sa mise en œuvre est facilement automatisable (déploiement aisé sur un nombre quelconque de machines), et il peut traiter toutes les requêtes en mémoire vive pour plus de rapidité. Afin de vérifier le fonctionnement de notre solution, nous avons traité les requêtes R1 et R2 présentées en section 3.1 (cf. la formulation en Hive des requêtes dans [14]).

Stratégie de répartition des données : HadoopDB requiert que les données soient initialement stockées dans les BD_i , de manière figée et définie à priori par l'administrateur du système. Cette contrainte ne convient pas à notre contexte où les données sont initialement dans le portail GBIF et où la réplication s'effectue à la demande en fonction des requêtes. Pour cela nous complétons actuellement le module P_i , afin de proposer une stratégie de répartition sélective, combinant fragmentation et réplication. La fragmentation bénéficie de l'organisation hiérarchique des données en pays, fournisseurs et collections. Lorsqu'un P_i reçoit une requête, il détermine les fragments (i.e. les collections, pays et fournisseurs) dont il a besoin pour évaluer la requête. Si nécessaire, P_i génère les appels de service appropriés pour récupérer les fragments auprès du portail du GBIF, puis P_i importe les fragments dans sa base H2.

5. Conclusion

Cet article présente une adaptation du portail GBIF dans l'optique d'améliorer les performances liées au traitement des requêtes et de permettre aux utilisateurs de définir de nouveaux besoins sur les données. Nous avons proposé l'utilisation d'une infrastructure de type Cloud afin de garantir la disponibilité du portail GBIF quand le nombre et la complexité des requêtes augmentent fortement. Nous avons identifiés les besoins des utilisateurs pour des requêtes SQL complexes portant sur des fragments des données exprimables en fonction de la hiérarchie (pays, fournisseurs, collections) des données de biodiversité. Nous avons énuméré une série de règles de conception à suivre, dans le but de résoudre le problème du passage à l'échelle du traitement des requêtes de biodiversité. Nous avons défini une architecture décentralisée pour traiter efficacement les requêtes des utilisateurs. Elle présente l'avantage de fonctionner sans

modifier le portail GBIF existant. Nous avons instancié notre architecture sur un cloud en nous appuyant sur des logiciels de gestion de données open source et récents. Nous les avons adaptés à notre contexte et améliorés en ajoutant une stratégie de fragmentation et réplication dynamique des données. Les premiers résultats obtenus montrent la faisabilité de notre solution et son efficacité pour quelques requêtes typiques de l'usage en biodiversité. Nous effectuons actuellement des expérimentations plus complètes et mesurons le coût d'exécution des requêtes et le passage à l'échelle du nombre de requêtes en fonction du nombre de nœuds alloués au système. Prochainement, nous investiguerons l'enrichissement des types de données présentes dans le GBIF pour prendre en compte une palette plus large des usages en biodiversité (e.g. la validation de modèles de niches écologiques ou la définition de bio-indicateurs).

Remerciements : Les auteurs remercient Régine Vignes-Lebbe, coordinatrice du GBIF France, pour sa présentation des problématiques de gestion de données de bio-diversité.

6. Bibliographie

- [1] D. Hobern, GBIF Biodiversity Data Architecture, *GBIF Data Access and Database Interoperability (DADI)*, 2003.
- [2] GBIF, GBIF Annual Report 2009, pages 25-42, 2010.
- [3] GBIF international www.gbif.org et GBIF France, www.gbif.fr.
- [4] H. Saarenma. Sharing and Accessing Biodiversity Data Globally through GBIF, *ESRI User Conf.*, 2005.
- [5] The GBIF Data Portal: A practical "hands-on" accessible au <http://data.gbif.org>.
- [6] Apache Hadoop <http://wiki.apache.org/hadoop>.
- [7] Hive wiki <http://wiki.apache.org/hadoop/hive>.
- [8] J. Dean, S. Ghemawat. MapReduce: simplified data processing on large clusters, *CACM*, 2008.
- [9] M. Brantner et al. Building a Database on S3, *SIGMOD*, 2008.
- [10] D. J. Abadi: Data Management in the Cloud: Limitations and Opportunities. *IEEE Data Eng. Bull.* 32(1), 2009.
- [11] A. Thusoo et al. Hive - A Petabyte Scale Data Warehouse Using Hadoop, *ICDE*, 2010.
- [12] K. Bajda-Pawlikowski, D. J. Abadi, A. Silberschatz, E. Paulson. Efficient processing of data warehousing queries in a split execution environment. *SIGMOD*, 2011.
- [13] A. Abouzied et al. HadoopDB in action: building real world applications. *SIGMOD*, 2010.
- [14] N. Bame. Traitement de requêtes pour les données du GBIF : Utiliser un Cloud pour améliorer les performances des requêtes, *Mémoire de DEA d'Informatique*, UCAD, 2011.
- [15] H2 Database Engine <http://www.h2database.com>.
- [16] D. Agrawal, S. Das, A. El Abbadi: Big data and cloud computing: current state and future opportunities. *EDBT* 2011.