RethinkDB et son langage d'interrogation

Projet du cours NFE204

Préparation du certificat de spécialisation Analyste de données massives

Rodolphe CHAZELLE

Année 2015-2016

Table des matières

Introduction

Dans le cadre du projet NFE204, j'ai décidé d'étudier le langage de requête du système No SQL RethinkDB. Développé à partir de 2009, RethinkDB est un système de gestion de base de données distribuées orienté documents sous licence libre. Ce système temps réel offre de nombreux avantages. En plus de la possibilité de visualiser en temps réel les modifications réalisées sur la base de données, RethinkDB dispose de son propre langage de requête : ReQL.

ReQL permet de manipuler avec aisance les données grâce à un système chainable d'instructions. Simple à utiliser, celui-ci il s'intègre facilement dans un environnement de programmation grâce à son package. Chose rare en système distribué, il permet de réaliser des jointures entre des tables.

Ce projet sera composé de trois parties. La première sera consacrée à la présentation du système RethinkDB. Nous verrons son histoire et décrirons ses principales caractéristiques (scalabilité, cohérence, partitionnement, reprise sur pannes etc.). La seconde partie introduira les données des événements en France, que nous utiliserons pour illustrer les principes énoncés. Nous présenterons les principales manipulations dans RethinkDB permettant d'importer les données et de configurer le système. Enfin, dans la dernière partie, nous aborderons les possibilités offertes par le langage de requête ReQL (filtre, agrégation, manipulations de bases, jointure etc.). Nous étudierons sa syntaxe en réalisant des requêtes sur la base de données constituée.

Logo de la dernière version de RethinkDB

I. Présentation et caractéristiques du systéme RethinkDB

a) Présentation

RethinkDB est un système de gestion de bases de données distribuées orienté documents qui permet de stocker des documents JSON. Il a été développé à partir de 2009 par une société du même nom. L'entreprise est domiciliée au 156 E.Dana St. Mountain View, CA 94041 aux Etats-Unis.

Distribué librement, le système et sa documentation sont disponibles à l'adresse suivante :

https://www.rethinkdb.com

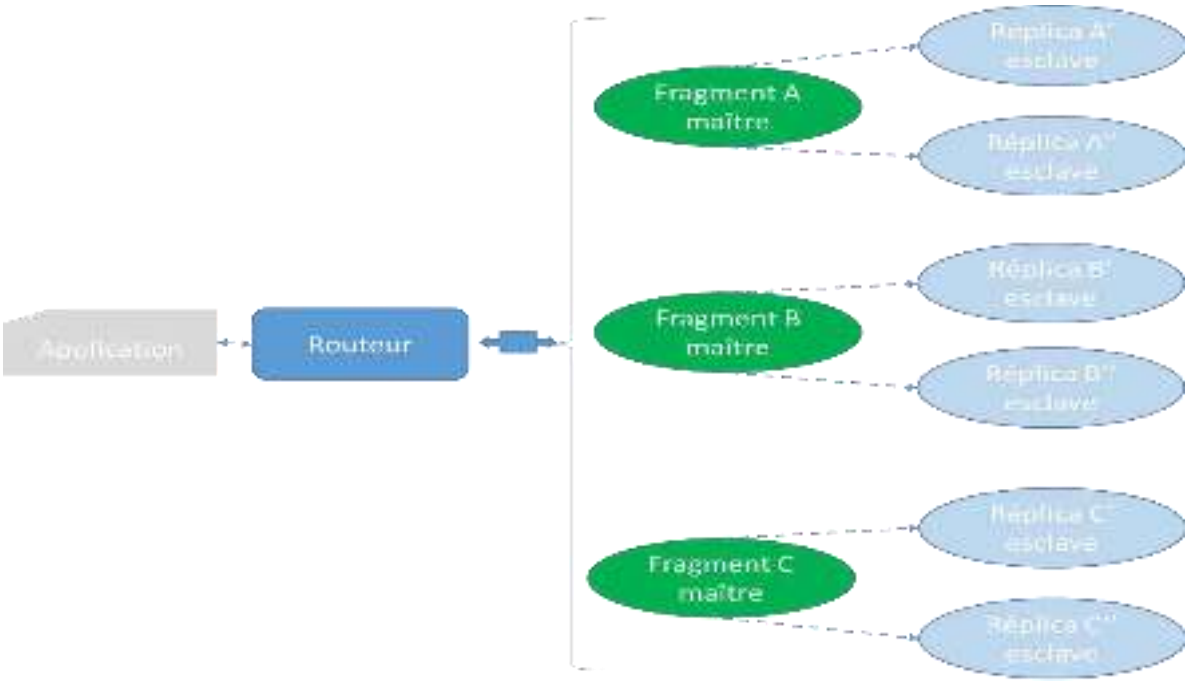
L'installation se réalise facilement et la documentation proposée est claire.

RethinkDB est un système dont la popularité ne cesse de croitre. Classé 67éme en septembre 2015 au classement DB-Engines, il est à la 46éme place à fin février 2016.

b) Mode de fonctionnement du système

RethinkDB fonctionne selon le schéma maître esclave. On choisit pour chaque table le nombre de fragments à mettre en place pour le stockage ainsi que le nombre de réplications de ces fragments.

Dans l'exemple d'une table partitionnée en trois fragments avec trois réplications par fragment. Une représentation simplifiée serait la suivante :



Tous les fragments de réplications sont associés à un réplica maître. Toutes les requêtes en lecture et écriture passent par ce nœud maître, où elles sont priorisées et évaluées.

c) Performance/Cohérence

Le rapport performance/cohérence du système est géré par les trois paramètres suivants :

En écriture

- Le paramètre write_ack permet de choisir entre une écriture synchrone ou asynchrone. Il peut prendre la valeur :

  «majority» : L'écriture est confirmée lorsque la majorité des réplicas a confirmé l'écriture. Les données sont majoritairement cohérentes.

  «single» : L'écriture est confirmée quand un seul réplica a confirmé l'écriture. Plus rapide mais risque pour la cohérence des données en cas de plantage.

En méthode de stockage

- Le paramètre durability permet de choisir le type de sauvegarde. Il possède les attributs suivants :

«hard» : les écritures sont réalisées sur le disque dur avant que la confirmation ne soit renvoyée.

«soft» : les écritures sont réalisées en RAM et par la suite sur le disque dur. Plus rapide mais en cas de panne les données n'ont pas été écrites dans la base.

En lecture

- Le paramètre read_mode peut prendre les valeurs suivantes :

  «single» : retourne la valeur dans la mémoire (pas nécessairement le disque dur) du réplica maître.

  «majority» : retourne la valeur majoritaire stockée sur les disques durs des réplicas. C'est la solution la plus longue mais qui garantit la meilleure cohérence.

  «outdated» : renvoie les valeurs qui sont en mémoire d'une sélection arbitraire de réplicas. C'est la solution la plus rapide mais qui garantit le moins la cohérence des données.

  Configuration de base : Majority, hard, single. RethinkDB préfère la prudence à la performance, excepté pour l'écriture (single à la place de majority), car le mode «majority» nécessite d'envoyer des requêtes à tous les réplicas et d'attendre une majorité de retours, ce qui a pour conséquence de dégrader significativement la performance.

d) Partitionnement

RethinkDB effectue un partitionnement par intervalles. Il utilise comme clé de partitionnement la clé primaire indiquée lors de l'import des données. Le nombre de fragments et de réplications est paramétrable pour chaque table.

e) Disponibilité du système : reprise sur panne

Tant qu'une table possède plus de la moitié des réplicas de ses fragments, alors sa disponibilité est assurée. Si un des fragments ne possède plus la moitié de ses réplicas, alors les opérations de lecture et d'écriture échoueront.

Si un réplica maître n'est plus disponible mais qu'une majorité des réplicas est disponible, alors ils élisent un nouveau réplica maître.

II. Présentation des données et import dans RethinkDB

a) Source des données

Les données sont issues du site Infolocale (http://datainfolocale.opendatasoft.com/explore/?sort=modified)

Infolocale propose divers services :

- Annonce et diffusion d'événements dans les médias (en une seule saisie, l'événement créé est publié dans les journaux et sur les sites partenaires).
- Faire connaitre son organisme (diffuser sa carte de visite dans l'annuaire local des associations de la commune).

C'est à ce premier service que nous allons nous intéresser dans le cadre du projet. Infolocale accepte de mettre à disposition ses données des événements au format JSON. Pour cela, il suffit de s'inscrire (gratuitement) et de présenter son projet par écrit. L'accés à la base de données m'a été permis suite à la présentation de ce projet.

b) Description du format des données

Le fichier est au format JSON et fait 1.2go. Les principaux champs de description d'un événement sont les suivants :

- Datasetid : constante « infolocale_evenements »
- Record_id : Clé primaire du document
- geometry
    - Coordinates : [longitude, latitude]
- Fields :
    - Categorie : catégorie de l'évènement
    - Code_insee : code INSEE de l'événement
    - Code postal : code postal de la ville
    - Commune : nom de la commune
    - Coordonnées GPS : latitude et longitude
    - Date modification : date de modification de l'annonce
    - Departement : numéro du département
    - Etat : «V»
    - Genre : genre de l'événement
    - Genre_ident : genre de l'événement en numérique
    - Jour_1 : date de début de l'événement
    - Jour_max : dernier jour de l'événement
    - Jour_min : premier jour de l'événement
    - Organisme_nom : Nom de l'organisme organisateur
    - Tarif_mention_payant : Oui ou non
    - Tarif mention_gratuit : oui ou non

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.