GeekBrains

BigData. Введение в экосистему Hadoop.

Урок 7. NoSQL

На уроке рассматриваем популярные базы данных, разбираем как устроены NoSQL решения на базе Cassandra и HBase.

Оглавление

- Оглавление

Теоретическая часть

- Строчные форматы
- Колоночные форматы
- Бинарные форматы

Практическая часть

Домашнее задание

Используемая литература

Теоретическая часть

Cassandra

Apache Cassandra - это распределенное, отказоустойчивое, масштабируемое, колонко-ориетированное хранилище данных. Хранилище само позаботится о проблемах наличия единой точки отказа (single point of failure), отказа серверов и о распределении данных между узлами кластера (cluster node). При чем, как в случае размещения серверов в одном центре обработки данных (data center), так и в конфигурации со многими центрами обработки данных, разделенных расстояниями и, соответственно, сетевыми задержками.

В терминологии кассандры приложение работает с пространством ключей (keyspace), что соответствует понятию схемы базы данных (database schema) в реляционной модели. В этом пространстве ключей могут находиться несколько колоночных семейств (column family), что соответствует понятию реляционной таблицы. В свою очередь, колоночные семейства содержат колонки (column), которые объединяются при помощи ключа (row key) в записи (row). Колонка состоит из трех частей: имени (column name), метки времени (timestamp) и значения (value). Колонки в пределах записи упорядочены. В отличие от реляционной БД, никаких ограничений на то, чтобы записи (а в терминах БД это строки) содержали колонки с такими же именами как и в других записях — нет. Колоночные семейства могут быть нескольких видов, но в этой статье мы будем опускать эту детализацию. Также в последних версиях кассандры появилась возможность выполнять запросы определения и изменения данных (DDL, DML) при помощи языка CQL[1], а также создавать вторичные индексы (secondary indices).

HBase

Apache HBase — это нереляционная, распределенная база данных с открытым исходным кодом, созданная по аналогии с BigTable от Google и написаная на Java. Она разработана как часть проекта Hadoop (входящего в состав Apache Software Foundation) и запускается на кластере HDFS (Hadoop Distributed Filesystem), предоставляя схожие с BigTable возможности. То есть, она обеспечивает отказоустойчивый способ хранения больших объемов разреженных данных. HBase линейно масштабируется для обработки огромных наборов данных с миллиардами строк и миллионов столбцов и легко объединяет источники данных, которые используют множество различных структур и схем. HBase интегрирован с Hadoop и без проблем работает вместе с другими механизмами доступа к данным через YARN.

BigTable - это дизайн для таблиц, для которых существует 2 принципа:

Принцип 1 - на всю таблицу есть одно индексное поле, называемое row key (аналог primary key).

Принцип 2 - данные во всех остальных полях не индексируются, таблица может иметь сколько угодно полей, добавление нового поля - затрагивает только отдельные row.

Практическая часть

Cassandra

1. Подключаемся к Cassandra на worker-2:

   /cassandra/bin/cqlsh 10.0.0.18

2. Создаем пространство ключей:

   CREATE KEYSPACE lesson7

   WITH REPLICATION = {

   'class' : 'SimpleStrategy', 'replication_factor' : 1 }

3. Создаем таблицу и вставляем значения:

   CREATE TABLE animals

   (id int,

   name text,

   size text,

   primary key (id));

   insert into animals (id, name, size)

   values (3, 'Deer', 'Big');

4. Проверяем как работает фильтрация:

   select * from animals

   where id = 3 and name = '12321';

5. Сравниваем удаление и вставку пустого значения:

   delete id from animals where id = 1;

```
insert into animals (id, name, size)
values (3, null, null);
```

HBase

1. Подключаемся к HBase

   ```
   hbase shell
   create_namespace 'lesson7'
   create 'lesson7:animals', 'name', 'size'
   ```

2. Вставляем значения:

   ```
   put 'lesson7:animals', '3', 'name', 'Deer'
   put 'lesson7:animals', '3', 'size', 'Big'
   put 'lesson7:animals', '5', 'name', 'Snake'
   put 'lesson7:animals', '3', 'name', 'Doe'
   ```

3. Удаляем значение:

   ```
   delete 'lesson7:animals', '5'
   ```

4. Делаем запрос к созданной таблице:

   ```
   get  'lesson7:animals', '5'
   ```

Домашнее задание

1. Подключить к Cassandra
2. Создать таблицы
3. Вставить записи
4. Изучить особенности работы where
5. Подключиться к HBase
6. Создать таблицы и вставить значения
7. Изучить особенности хранения данных

Задачи со * предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. https://ru.bmstu.wiki/Apache_Cassandra
2. https://ru.bmstu.wiki/Apache_HBase