

1. Статьи в блоге Cloudera про RegionSplit

- a. <https://blog.cloudera.com/apache-hbase-region-splitting-and-merging/>
- b. <https://community.cloudera.com/t5/Community-Articles/How-Region-Split-works-in-HBase/ta-p/248814>

2. Задача на планирование архитектуры БД

Главное при выполнении задания по HBase - удобная организация таблицы с данными, с использованием ключевых свойств колоночного хранения. Рассмотрим это на примере одной задачи.

Задача.

Клиент должен уметь выводить количество хитов (просмотров страниц) за указанный интервал времени в разбивке по дням, часам и по доменам. На вход клиент получает: интервал времени (с точностью до дней), тип разбиения данного интервала (по дням или часам), флаг необходимости детализации по доменам. На выходе выдает набор пар: интервал времени – количество хитов. При наличии флага детализации по доменам: интервал времени и домен – количество хитов.

Итак, надо быстро отвечать на запросы, которые будут содержать: start_date, end_date, show_domains (флаг). На выходе должна быть статистика по дням, с детализацией по часам и, возможно, доменам:

```
date  hour  hits
или
date  hour  domain  hits
```

Напомню главную цель: данные надо разложить в таблицу так, чтобы быстро отвечать на такие запросы. Посмотрим, какие могут быть варианты.

Решение 1. (стиль реляционной базы)

Таблица, колонки: date, hour, domain, hits

Тройка: (date, hour, domain) - primary key, т.е. date - не уникальное поле, но по нему есть быстрый поиск.

В HBase это можно записать в таблицу, у которой:

Key: <date> <hour> <domain>

Column family: h - одно семейство, имя семейств обычно выбирается коротким

Column: hits - в семействе одна колонка

Value: <hits> - в ячейке для каждого ключа будем записывать количество хитов.

По этой таблице реально искать нужные даты, т.к. есть поиск по префиксу ключа. Но для получения статистики надо будет прочитать много ключей (потому что доменов много). Для ускорения чтения статистики без детализации по доменам, можно ввести фиктивный домен TOTAL, куда писать сумму хитов по всем доменам для пары (date, hour). В этом решении HBase используется как простая key-value база и из всех удобств, предоставляемых семействами столбцов, не задействовано ни одного.

Напомню: таблица в HBase может иметь несколько семейств колонок - column family, cf (обычно немного), в каждом из них - колонки, которых может быть очень много. Причем для каждой записи (для каждого ключа) может быть свой набор колонок. В ячейке, заданной ключом и парой cf:column хранится значение.

Решение 2. Добавляем колонки в таблицу. В названиях колонок пусть будут домены.

Key: <date> <hour>

Column family: d - семейство колонок для доменов

Column: <domain> - в семействе столько колонок, сколько доменов для данного дня и часа

Value: <hits> - в ячейке - по-прежнему количество хитов на домен.

Для ускорения получения статистики без детализации вводим еще семейство.

Column family: t - одно семейство, для доменов

Column: total - одна колонка в семействе

Value: <hits> - количество хитов всего для дня и часа.

Выборка без детализации будет работать быстрее, потому что разные семейства колонок хранятся и читаются отдельно: для доступа к семейству t читать другое семейство не нужно (в этом - специфика колоночных баз).

Кроме того, читать будем намного меньше ключей, чем в первом варианте: количество дней в интервале start_date - end_date умноженное на 24.

Решение 3. Можно еще уменьшить число ключей, если занести часы в семейства столбцов.

Key: <date>

Column: <hour> в виде 00, 01, ...- семейства колонок для доменов

Column family: <domain> - в каждом семействе столько колонок, сколько доменов для данного дня и часа

Value: <hits> - количество хитов на домен.

И также отдельное семейство для total. Этот вариант - небольшое развитие предыдущего и имеет смысл, когда нужно читать отдельные часы. Поэтому для данной задачи пользы не будет, но схема вполне рабочая.

Решение 4. Вспомним, что в каждой ячейке есть история значений, которые маркируются timestamp. Эти временные отметки мы можем выставлять сами и хранить в них полезные данные. В этом решении будем хранить там день и час.

Key: <date>

Column family: d - семейство колонок для доменов

Column: <domain> - в семействе столько колонок, сколько доменов для данного дня

Timestamp: timestamp(<hour>)

Value: <hits> - в ячейке - количество хитов, причем каждая ячейка имеет максимальную глубину истории 24.

Итак, у довольно простой задачи есть несколько путей решения. Вариант 1 - самый очевидный, но для HBase выглядит нелепо. Вариант 2 - хороший и вполне рабочий, 3 - тоже, но в данном случае несколько избыточный. Вариант 4 интересен использованием истории значений ячейки.