

HBase

Вопросы проектирования базы данных

“Узкие” или “Широкие” таблицы

- “Широкая” таблица означает большое количество столбцов
 - Плюсы : атомарное изменение данных, одна запись лежит на одном узле
 - Минусы : нет способа быстро выбрать требуемые данные без scan
- “Узкая” таблица подразумевает использование большого количества записей
 - Плюсы и минусы прямо противоположны плюсам и минусам широкой таблицы

Примеры

- Хранение email
- “Широкая” таблица :

`<userId> : <colfam> : <messageId> : <timestamp> : <email-message>`

`12345 : data : 5fc38314-e290-ae5da5fc375d : 1307097848 : "Hi Lars, ..."`

`12345 : data : 725aae5f-d72e-f90f3f070419 : 1307099848 : "Welcome, and ..."`

`12345 : data : cc6775b3-f249-c6dd2b1a7467 : 1307101848 : "To Whom It ..."`

`12345 : data : dcbee495-6d5e-6ed48124632c : 1307103848 : "Hi, how are ..."`

- Узкая таблица

`<userId>-<messageId> : <colfam> : <qualifier> : <timestamp> : <email-message>`

`12345-5fc38314-e290-ae5da5fc375d : data : : 1307097848 : "Hi Lars, ..."`

`12345-725aae5f-d72e-f90f3f070419 : data : : 1307099848 : "Welcome, and ..."`

`12345-cc6775b3-f249-c6dd2b1a7467 : data : : 1307101848 : "To Whom It ..."`

`12345-dcbee495-6d5e-6ed48124632c : data : : 1307103848 : "Hi, how are ..."`

Partial key scans

- Пример составного ключа:

<userId>-<messageld>

<userId>-<date>-<messageld>

- Позволяет выбирать данные заданной группы

<userId>-<messageld>

Выбираем email пользователя 12345

12345-000... по 12346-999...

<userId>-<date>-<messageld>

Выбираем email пользователя 12345 написанные 01.01.2013

12345-2013.01.01-000... по 12345-2013.01.02-000

Проблема монотонных ключей

- Для многих предметных областей характерно монотонное увеличение ключа, например при логировании событий
- Результат: все записи будут сконцентрированы на одном сервере
 - Неравномерная загрузка кластера
 - Теряется весь смысл nosql

Варианты решения : “соль”

- Добавление префикса(“Соль”) в ключ

```
byte prefix = (byte) (Long.hashCode(timestamp) %  
<number of region servers>);  
  
byte[] rowkey = Bytes.add(Bytes.toBytes(prefix),  
Bytes.toBytes(timestamp));
```
- Минусы : чтобы получить выборку данных по диапазону ключа, потребуется сделать запрос к каждому узлу кластера (сгенерировав руками все варианты префиксов) и потом объединить результаты

Изменение порядка столбцов

- Применяем составной ключ в котором сначала идет поле с нормальным распределением
- Минусы : неудобно делать выборку по интервалу значений основного ключа

Использование случайного ключа

- В данном случае мы используем функцию (например md5) от столбца
- Плюсы : равномерное распределение данных по кластеру
- Минусы : невозможно осуществить сканирование интервала

Сравнение подходов

