

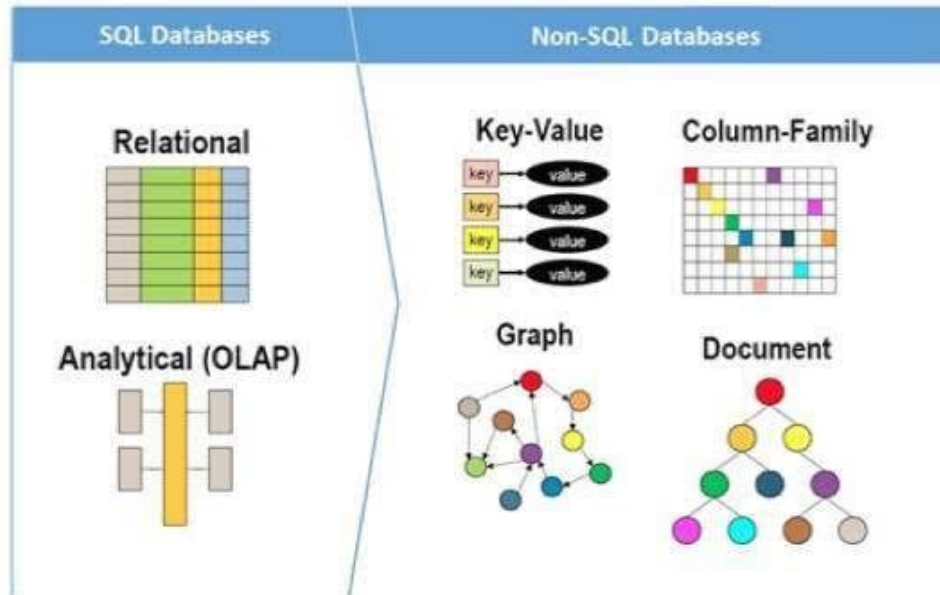
HBase



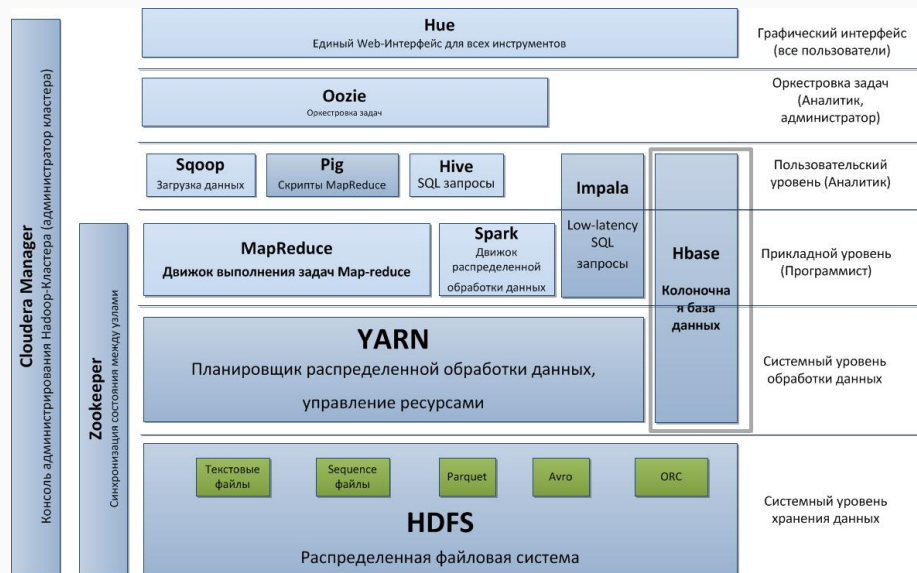
NoSQL Базы данных

- Гибкость
- Масштабируемость
- Высокая производительность
- Широкие функциональные возможности

noSQL: “Not Only SQL”



HBase в экосистеме Hadoop



Особенности HBase

- Распределенная база данных:
 - Работает на кластере серверов
 - Легко горизонтально масштабируется
- NoSQL база данных:
 - Не предоставляет SQL-доступ(напрямую)
 - Не предоставляет реляционной модели

Особенности HBase

- Column-Oriented хранилище данных
 - нет фиксированной структуры колонок
 - произвольное число колонок для ключа
- Спроектирована для поддержки больших таблиц
 - Миллиарды строк и миллионы колонок
- Поддержка произвольных операций чтения/записи

Особенности HBase

- Основана на идеях Google BigTable
 - <https://static.googleusercontent.com/media/research.google.com/ru/archive/bigtable-osdi06.pdf>
- BigTable поверх GFS => HBase поверх HDFS
- Масштабируемость с помощью шардирования
- Автоматический fail-over
- Простой Java API
- Интеграция с MapReduce

Использование HBase

- Большие объемы данных
- Паттерн доступа к данным:
 - Выборка значений по заданному ключу
 - Последовательный скан в диапазоне ключей
- Свободная схема данных
 - Строки могут существенно отличаться по своей структуре
 - В схеме может быть множество колонок и большинство из них будет null

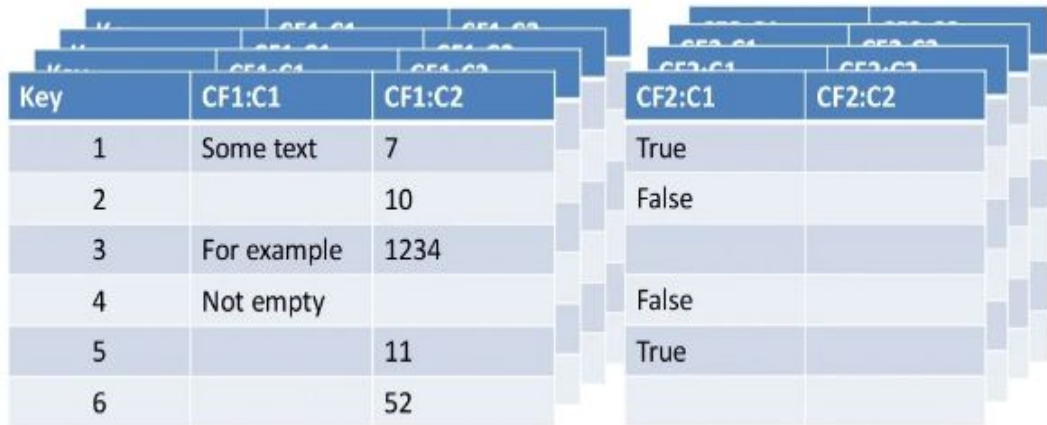
Когда не стоит использовать HBase

- Традиционный доступ к данным в стиле РБД
 - Приложения с транзакциями
 - Реляционная аналитика('group by', 'join', 'where column like' и т.д.)
- Полнотекстовый поиск

HBase Column Families

Column Family описывает общие свойства колонок:

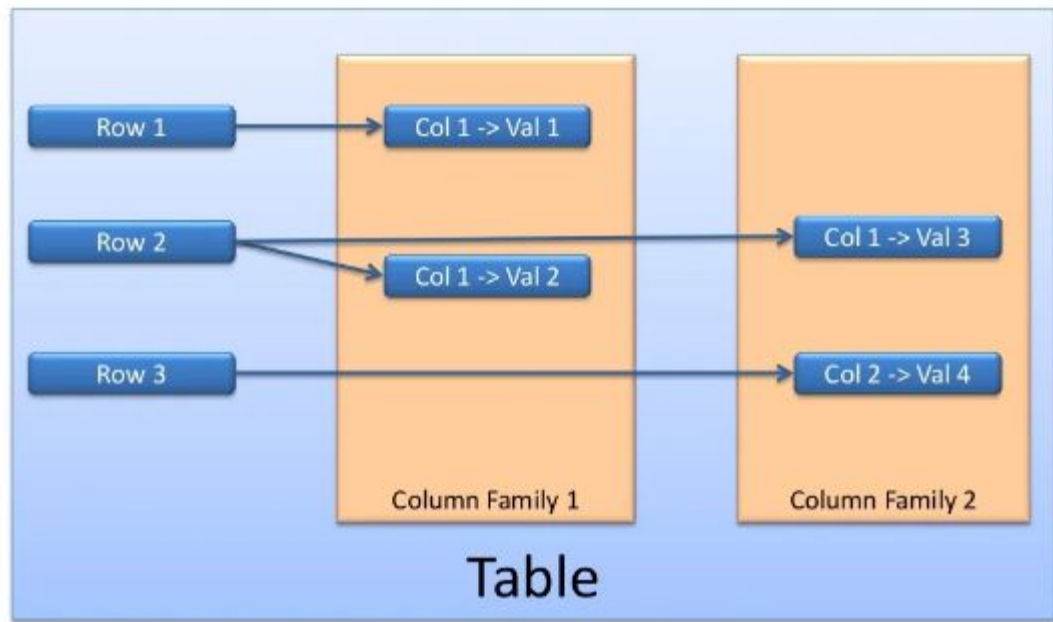
- Сжатие
- Количество версий данных
- Время жизни(Time To Live)
- Опция хранения только в памяти (In-memory)
- Хранится в отдельном файле(HFile/StoreFile)



Key	CF1:C1	CF1:C2	CF2:C1	CF2:C2
1	Some text	7	True	
2		10	False	
3	For example	1234		
4	Not empty		False	
5		11	True	
6		52		

HBase Column Families

- Конфигурация CF статична
 - Задается в процессе создания таблицы
 - Количество CF ограничено небольшим числом
- Колонки наоборот НЕ статичны
 - Создаются в runtime
 - Сотни тысяч для одной CF



HBase Timestamps

- Ячейки имеют несколько версий данных
 - Настраивается в конфигурации ColumnFamily
 - По-умолчанию 3
- Данные имеют timestamp
 - Задается неявно при записи
 - Явно указывается клиентом
- Версии хранятся в убывающем порядке ts
 - Последнее значение читается первым

Value = Table + RowKey + Family + Column + Timestamp

Пример таблицы

Row Key	Timestamp	CF: "Data"		CF: "Meta"		
		Url	Html	Size	Date	Log
row1	t1	Mail.Ru				Log text 1
	t2				123456	Log text 2
	t3		<html>...	1234		Log text 3
	t4		<HTML>...	2345		Log text 4
row2	t1	OK.Ru			123765	Log text 1
	t2					Log text 2

Архитектура HBase

Хранение данных таблицы

Key	Column1	Column3	TimeStamp
Row1	value1		timestamp1
		value2	timestamp2
Row2	value4		timestamp1
Row3	value1	value6	timestamp1

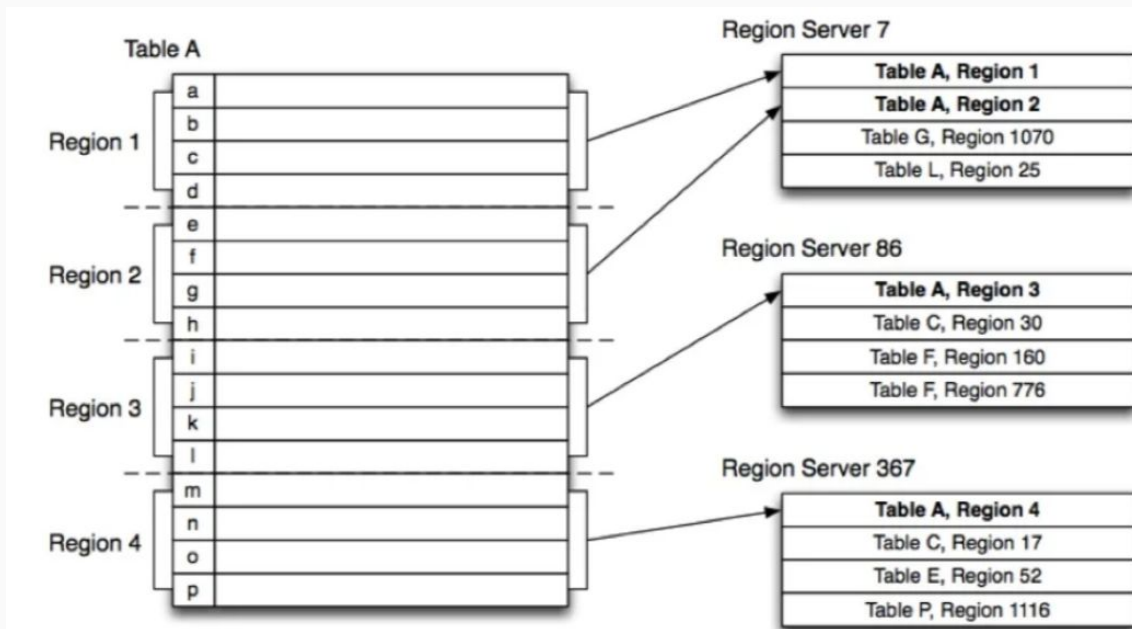
HFile

Row1:ColumnFamily:column1:timestamp1:value1
Row1:ColumnFamily:column3:timestamp2:value3
Row2:ColumnFamily:column1:timestamp1:value4
Row3:ColumnFamily:column1:timestamp1:value1
Row3:ColumnFamily:column3:timestamp1:value6

↔ KeyValues

Масштабируемость HBase

- Таблица делится на регионы
- Регион - группа строк, хранящихся вместе
 - Единица шардинга
 - Динамически делится пополам
- RegionServer - демон, который управляет одним или несколькими регионами(регион принадлежит только одному RS)
- MasterServer(HMaster) - демон, который управляет всеми RS



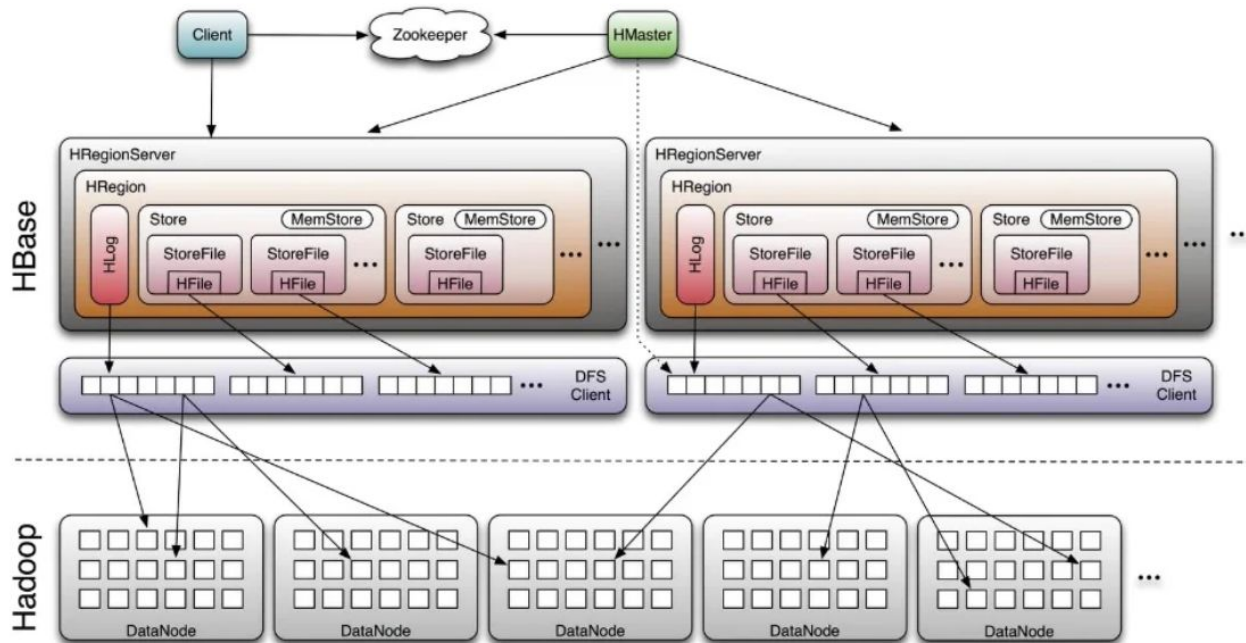
Hbase Regions

- Регион - это диапазон ключей: [Start Key; Stop Key)
 - StartKey включается в регион
 - StopKey не включается
- По-умолчанию есть только один регион
- Можно предварительно задать количество регионов
- При превышении лимита, регион разбивается на 2 части

Hbase Regions Split

- Регионы более сбалансированы по размеру
- Быстрое восстановление, если регион повредился
- Баланс нагрузки на RegionServer
- Split - быстрая операция
- На больших инсталляциях лучше производить вручную

Архитектура HBase

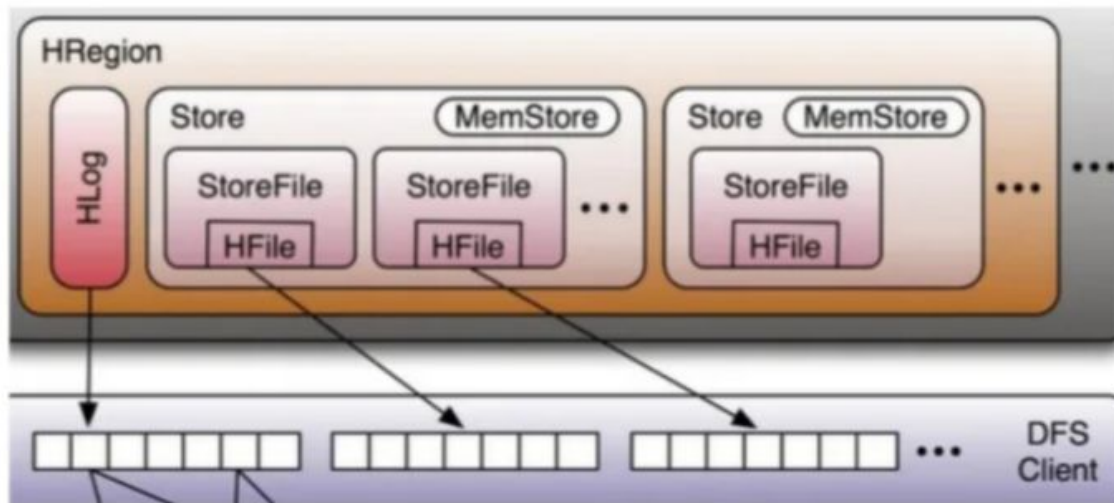


HBase Master

- **Управляет регионами:**
 - Назначает регион на RegionServer
 - Перебалансирует и для распределения нагрузки
 - Восстанавливает регион, если он становится недоступен
- **Не хранит данные**
 - Клиент взаимодействует напрямую с RS
 - Обычно не сильно нагружен
- **Отвечает за управление схемой таблиц и ее изменений**
 - Добавляет/удаляет таблицы и CF

Data Storage

- HLog - лог записи(для восстановления данных из кэша)
- MemStore - буфер на запись
- HFile - файл для хранения таблицы в HDFS
- DFS Client - взаимодействие с HDFS



Внесение изменений в данные

- В HDFS нельзя внести изменения в файл
 - Нельзя удалить key-value из HFile
 - Со временем становится много HFile'ов
- При удалении добавляется delete marker
 - Маркеры используются для фильтрации удаленных записей в runtime

Data Storage: compaction

- Minor Compaction
 - Маленькие HFile'ы объединяются в бОльшие файлы
 - Быстрая операция
 - Маркеры удаления не применяются
- Major Compaction
 - Для каждого региона все файлы в рамках одной CF объединяются в один файл
 - Используются маркеры удаления для того, чтобы не включать удаленные записи

Взаимодействие с HBase

Доступ к HBase

- Основные способы:

- HBase Shell
- Native Java API

- Другие способы:

- Avro Server
- PyHBase
- REST Server
- Thrift

Запрос данных из HBase

- По RowId или набору RowIds
 - Отдает только те строки, которые соответствуют заданным id
 - Если не заданы дополнительные критерии, то отдаются все cells из всех CF заданной таблицы(Это означает слияние множества файлов из HDFS)
- По ColumnFamily
 - Уменьшает количество файлов для загрузки
- По Timestamp/Version
 - Может пропускать полностью Hfile'ы. которые не содержат данный диапазон timestamp'ов

Запрос данных из HBase

- По Column Name/Qualifier
 - Уменьшает объем передаваемых данных
- По Value
 - Можно пропускать ячейки используя фильтры
 - Самый медленный критерий выбора данных
 - Будет проверять каждую ячейку
- Фильтры могут применяться для каждого критерия выбора
 - rows, families, columns, timestamps и values
- Можно использовать множественные критерии выбора данных

Практическое применение

- Компания Streamu применяет рассматриваемую Hbase в рамках своей соцсети новостных сайтов в реальном времени, заменив прежнюю реляционную СУБД. HBase обеспечивает хранение сотни миллионов документов, разреженных матриц, журналов и всего остального, что когда-то было сделано в SQL-системе. Благодаря кэшированию в памяти результатов запроса обеспечивается высокое быстродействие, а глубокая интеграция с экосистемой Hadoop обеспечивает надежное выполнение тысячи ежедневных заданий MapReduce, используя таблицы для анализа журналов, обработки данных о внимании и сканирования каналов.
- Facebook хранит в этой СУБД все потоковые данные, сгенерированные из различных сервисов (чаты, электронная почта, смс и пр.). Ключевыми качествами для этого примера использования являются отказоустойчивость и возможность быстрого извлечения данных с использованием техники произвольного доступа, что обеспечивает высокую производительность
- Yahoo, Twitter, NGDATA и множество других компаний по всему миру.

Практика