**Hbase实战**

一些参考网址

cloudera下载

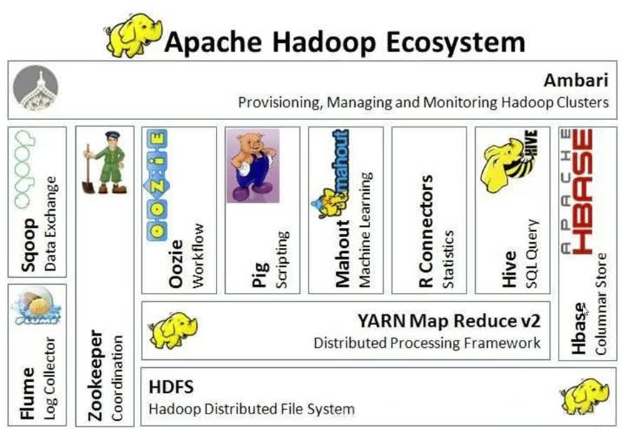
<http://archive-primary.cloudera.com/cdh5/cdh/5/>

spark官方文档

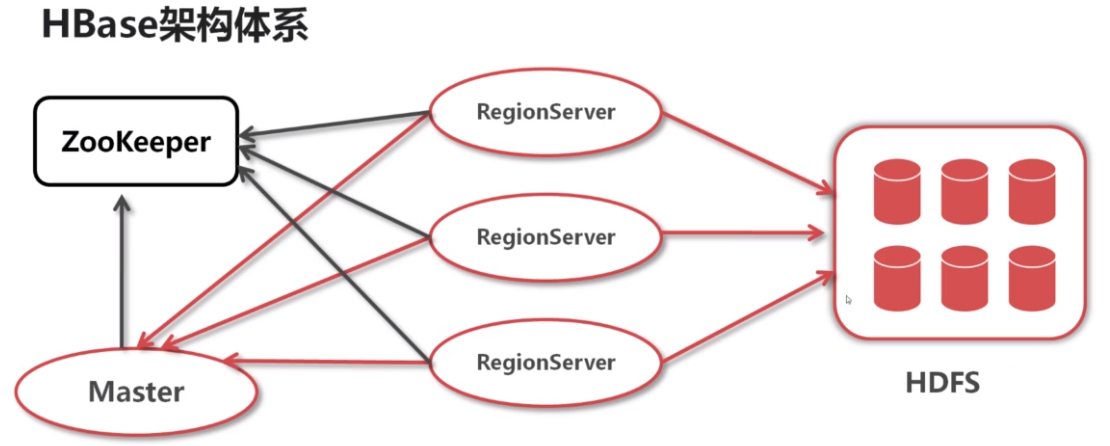
<http://spark.apache.org/docs/>

# 1 HBase入门

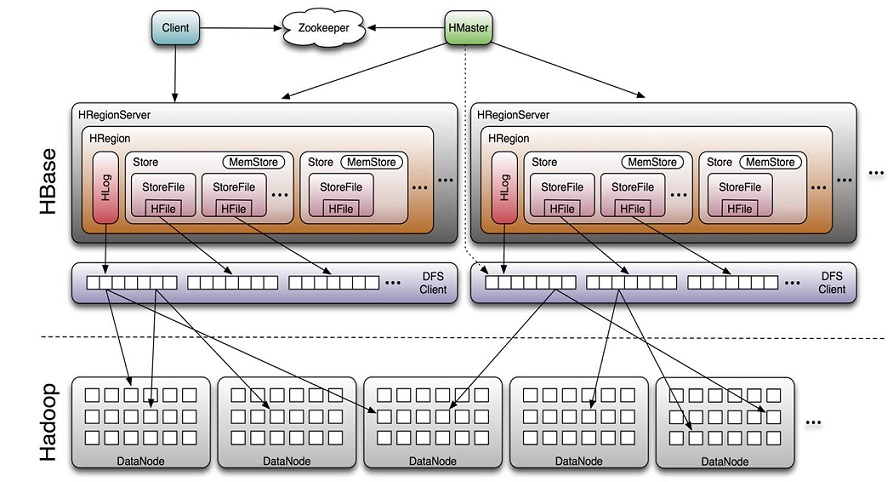
## HBase安装实践



### 1 HBase架构



两大主要进程：Master和RegionServer



Client主要功能

\* 使用HBase的RPC机制与HMaster和HRegionServer进行通信

\* 管理类操作，Client与HMaster进行RPC

\* 读写类操作，Client与HRegionServer进行RPC

Zookeeper主要功能

\* 通过选举，保证任何时候，集群中只有一个master，Master与RegionServers 启动时会向ZooKeeper注册

\* 实时监控Region server的上线和下线信息,并实时通知给Master

\* 存贮所有Region的寻址入口和HBase的schema和table元数据

\* zookeeper实现HBase主从节点的failover（故障切换）

HMaster主要功能

\* 管理HRegionServer，实现其负载均衡，在HRegion split时分配新的HRegion；在HRegionServer退出时迁移其内的HRegion到其他HRegionServer上

\* 监控集群中所有HRegionServer的状态(通过Heartbeat和监听ZooKeeper中的状态)

\* 处理schema更新请求 (创建、删除、修改Table的定义）

HRegionServer功能：

\* Region server维护Master分配给它的region，处理对这些region的IO请求

\* Region server负责切分在运行过程中变得过大的region

总结：

\* client访问hbase上数据的过程并不需要master参与（寻址访问zookeeper，数据读写访问regione server），master仅仅维护者table和region的元数据信息，负载很低。

\* HRegion所处理的数据尽量和数据所在的DataNode在一起，实现数据的本地化

补充：Schema是什么？？

在SQL环境下，schema就是数据库对象的集合，所谓的数据库对象也就是常说的表，索引，视图，存储过程等。

### HBase表结构

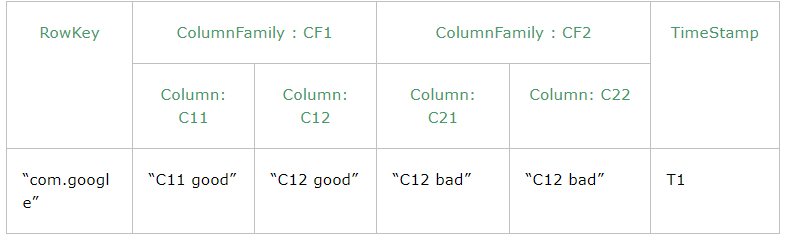
参考文档：<https://blog.csdn.net/qq_24908345/article/details/53230169>

#### 设计模式

HBase的模式结构包括**表、Rowkey,列族，timestamp**

列簇不超过5个，每个列簇中列数无限制，列只有插入数据后就存在

#### 表结构



Hbase最基本的单位是列（column)。一列或多列形成一行（row），并由唯一的行键（row key)来确定存储。

HBase表结构

建表时，不需要限定表中的字段，只需要指定若干个列族。

插入数据时，列族中可以存储任意多个列（以kv对的形式存储， 列名-列值）

一个value可以有多个版本，通过版本号来区分（时间戳）--默认三个

要查询某一个具体字段的值，需要指定的坐标：表名-->行键-->列族（ColumnFamily):列名（Qualifier)

Rowkey-行键（主键）

行键是字节数组，任何字符串都可以作为行键。

表中的行键进行排序，数据按照Rowkey的字节（byte order) 排序存储

所有的对表的访问都要通过行键（单个Rowkey访问，或Rowkey范围访问，全表扫描)

Column family(列族）

列族必须定义表时给出，但也可以动态添加列族。

列族可以有一个或者多个列成员

数据按列族分开存储，HBase是根据列族分开存储（每个列族对应一个Store）

TimeStamp(时间戳）

每个Cell可能有多个版本，他们之间用时间戳区分。

Cell(单元格）

Cell由行键，列族：限定符，时间戳唯一决定。

Cell中的数据是没有类型的，全部以字节码形式存储

每个单元格保存着同一份数据的多个版本

不同时间版本的数据按照时间顺序倒序排序

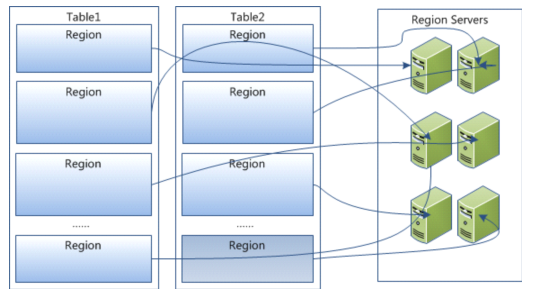
Region(区域）

HBase自动把表水平（按row）划分成多个区域（region)，每个region会保存一个表里面某段连续的数据

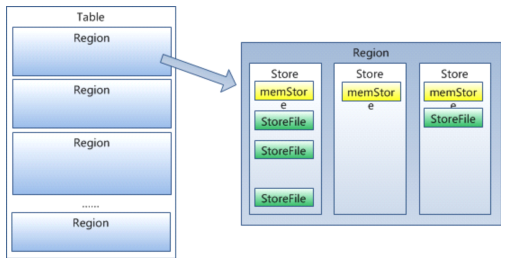
HRegion是HBase中分布式存储和负载均衡的最小单元，最小单元表示不同的HRegion可以分布在不同的HRegionServer上。但一个HRegion不会拆分到多个server上。

#### 物理存储

Table 在行的方向上分割为多个HRegion，每个HRegion分散在不同的RegionServer中。



每个HRegion由多个Store构成，每个Store由一个memStore和0或多个StoreFile组成，每个Store保存一个Columns Family



StoreFile以HFile格式存储在HDFS中。

#### HBase与关系型数据库对比

|  |  |
| --- | --- |
| HBase | 关系型数据库 |
| 支持列动态增加  数据自动切分  高并发读写  不支持条件查询 | 不支持列动态增加  不支持数据自动切分  不支持高并发读写  支持复杂查询 |

### HBase安装部署

条件：安装hadoop、安装jdk1.7以上、安装zookeeper

版本 hbase-1.2.0-cdh5.7.0.tar

1 环境变量配置

sudo vim /etc/profile

## HBASE\_HOME

export HBASE\_HOME=/home/hadoop/app/hbase

export PATH=$PATH:$HBASE\_HOME/bin

使之生效：source /etc/profile

2 修改配置文件

conf/hbase-env.sh

添加两项：

export JAVA\_HOME=/home/hadoop/app/jdk

export HBASE\_MANAGES\_ZK=false

conf/hbase-site.xml

<configuration>

<!--hbase持久保存的目录-->

<property>

<name>hbase.rootdir</name>

<value>hdfs://hadoop1:8020/hadoop/hbase</value>

</property>

<!--是否是分布式-->

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<!--指定要连接zk的节点-->

<property>

<name>hbase.zookeeper.quorum</name>

<value>hadoop1</value>

</property>

<property>

<name>hbase.tmp.dir</name>

<value>/home/hadoop/app/hbase/data/tmp</value>

</property>

</configuration>

修改regionservers主机

conf/regionservers

hadoop1

创建hbase临时目录tmp

[hadoop@hadoop1 hbase]$ mkdir -p data/tmp

#### 常用命令

启动hbse

$ bin/start-hbase.sh

[hadoop@hadoop1 hbase]$ jps

24113 Jps

10945 Kafka

20338 NameNode

20854 NodeManager

7623 logviewer

20615 SecondaryNameNode

23880 HRegionServer

20458 DataNode

20762 ResourceManager

23739 HMaster

7373 core

10446 QuorumPeerMain

web监控

<http://hadoop1:60010/>

进入命令行

$ bin/hbase shell

创建表

hbase(main):001:0> create 'test','info'

查看所有表

hbase(main):002:0> list

查看表

hbase(main):003:0> scan 'test'

添加数据

hbase(main):005:0> put 'test','0001','info:username','hery'

0001 – rowkey

info – 列簇

username – 列名

hery – 列

hbase(main):006:0> put 'test','0001','info:age','20'

查看表具体信息

hbase(main):008:0> desc 'test'

删除表

## step1：禁用表

hbase(main):011:0> is\_enabled 'test'

### step2：删除表

hbase(main):012:0> drop 'test'

**count命令**

hbase(main):018:0> put 'test','0002','info:username','hadoop'

hbase(main):019:0> put 'test','0002','info:age','26'

hbase(main):020:0> count 'test'

**get命令**

hbase(main):021:0> get 'test','0001','info:username'

**delete命令**

hbase(main):022:0> delete 'test','0001','info:username'

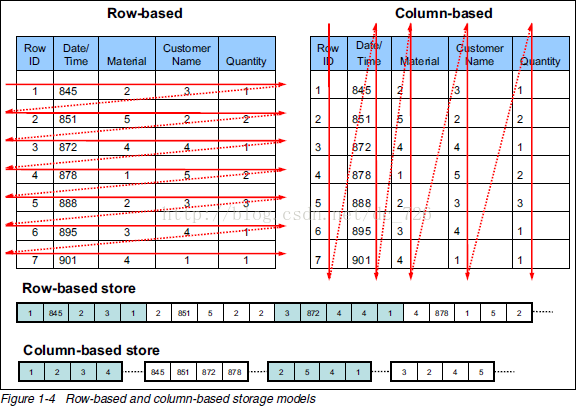
truncate命令

hbase(main):022:0> truncate 'test'

## 1.2 Hbase存储原理

### 1 行存储和列存储

列式存储是相对于传统关系型数据库的行式存储来说的。简单来说两者的区别：如何组织表：行式存储下一张表的数据都是放在一起的，但列式存储下都被分开保存了.



|  |  |  |
| --- | --- | --- |
|  | 行存储 | 列存储 |
| 优点 | 1 数据被保存在一起  2 随机读的效率高  3 对事物的处理能力支持得非常好  （insert/update非常容易） | 1 数据分开存储，无需维护索引，存储成本低  2 按列存储，压缩效率高  3 支持并行查询  4 查询时只有涉及到的列会被读取 |
| 缺点 | 1 维护大量的索引，存储成本高  2 由于按行存储，压缩效率低。  3 不支持并行存储  4 选择时即使只涉及某几列，所有数据都会被读取。 | 1 随机查询比较慢  2 INSERT/UPDATE比较麻烦  3 选择完成时，被选择的列要重新组装 |

应用产景：

当你的核心业务是 OLTP 时，一个行式数据库，再加上优化操作，可能是个最好的选择。

当你的核心业务是 OLAP 时，一个列式数据库，绝对是更好的选择

延伸：

数据处理大致可以分成两大类：联机事务处理OLTP（on-line transaction processing）、联机分析处理OLAP（On-Line Analytical Processing）。OLTP是传统的关系型数据库的主要应用，主要是基本的、日常的事务处理，例如银行交易。OLAP是数据仓库系统的主要应用，支持复杂的分析操作，侧重决策支持，并且提供直观易懂的查询结果。

### 2 HBase列族式存储

#### 概念

列簇（多个数据列的组合），HBase表中的每个列都归属于某个列簇。

列簇是表的schame的一部分，但是列并不是

创建表时，需要给出列簇的名称，不需要给出列的名称

列名都是以列簇作为前缀

访问控制磁盘和内存的使用统计都是在列簇层面进行

HBase准确的说是列簇数据库，而不是列数据库

列簇数据库将列组织为列簇，每列都必须是某个列簇的一部分

访问数据的单元也是列

#### HBase表的组成

Table = RowKey + Family + Column + Timestamp + Value

RowKey ：HBase中用RowKey去标识唯一的一行数据，一行数据中包含多个列簇

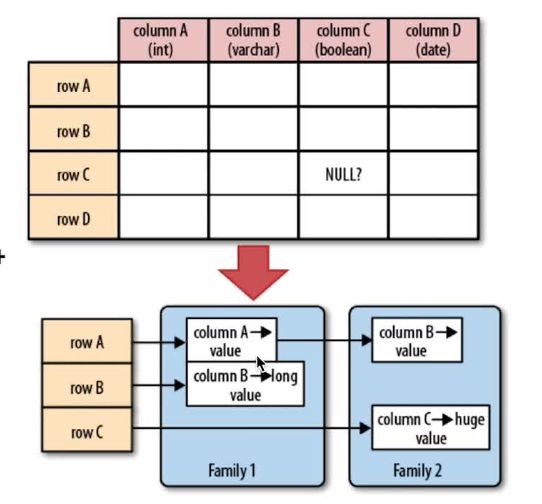
Family：多个列簇。每一列簇包含多个列

Column：列标识符。每一列数据包含了版本和值

Timestamp：版本。可以理解为时间戳，也可以理解为一个数据的版本

Value：数据值。数据本身的值

#### HBase数据存储的模式



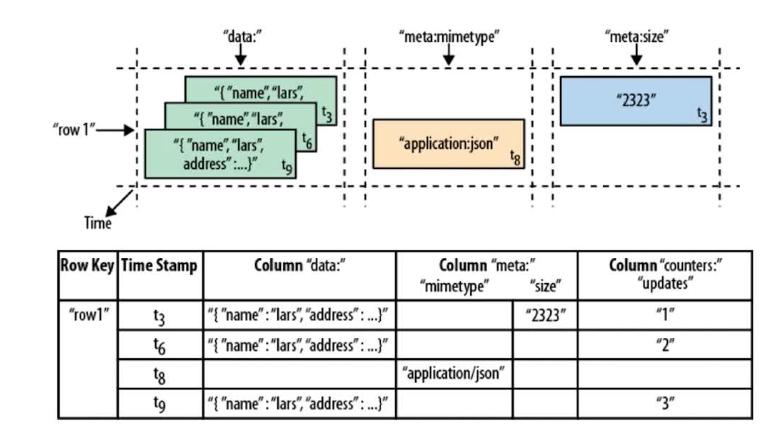
（Table, RowKey, Family, Column, Timestamp）-> Value

其实就是HBase表反过来看的样子

【重点】更抽象一点，其实HBase表数据就是Key-Value结构的

列簇式存储：列数据属性

HBase重要特性：列数据版本的概念，默认一列数据可以保存三个版本



#### 列簇式存储：数据存储原型

