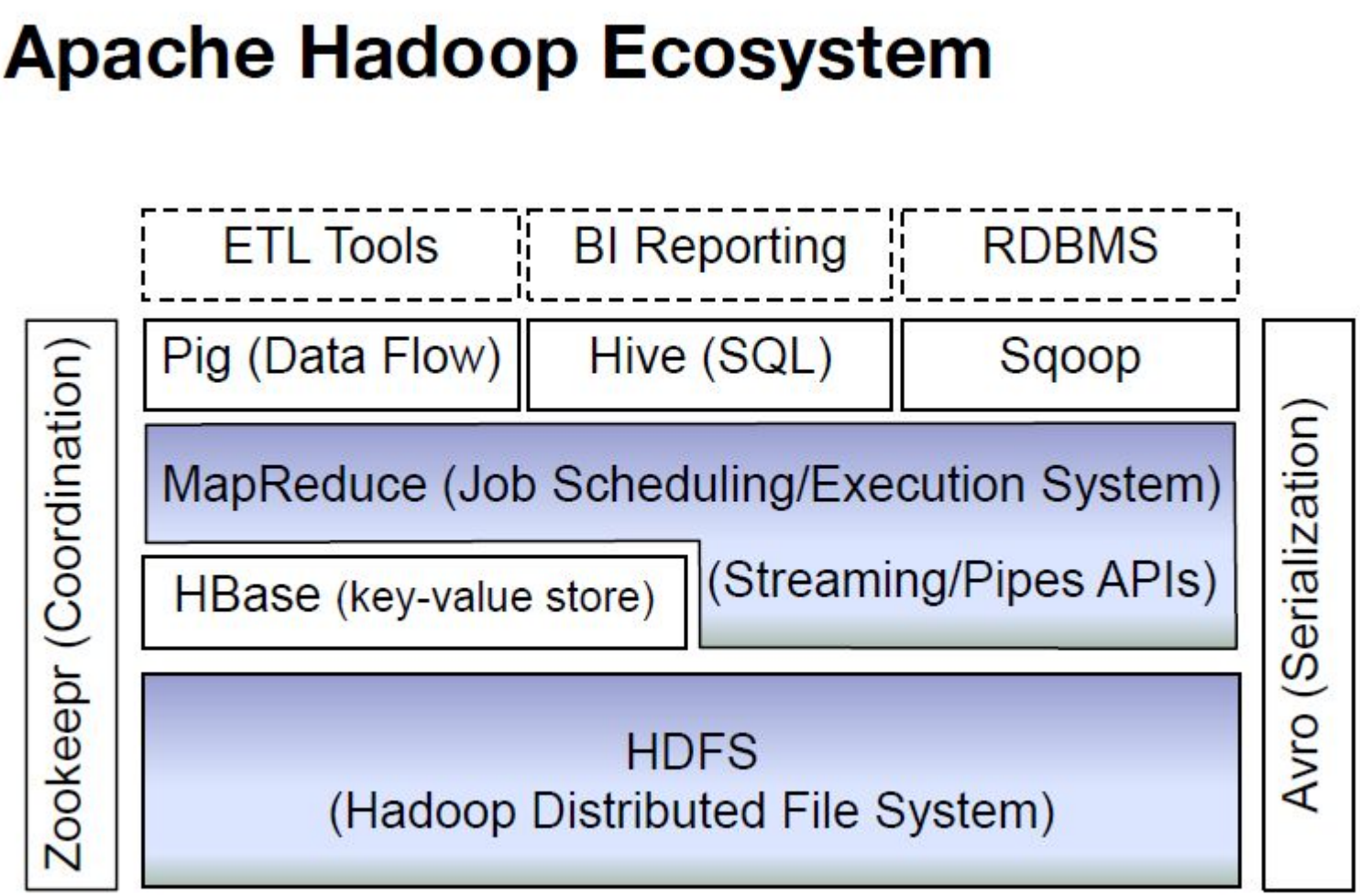
Hbase介绍

UC优视 郑梓力

目录

- 简介
- 数据模型
- 物理储存
- 系统架构
- 操作
- 特点

Hbase简介——Hadoop的生态系统



Hbase简介

- HBase，全称为Hadoop Hbase，是一个分布式的、多版本的、面向列的开源数据库
- HBase是Google Bigtable的开源实现

数据模型——概念视图

RowKey	Time Stamp	Column Family:article		Column Family:author	
		列	值	列	值
rowkey1	t4	article:title	HBase in Action	author:name	lee
	t2	article:title	HBase	author:nickname	nicholas
	t4	article:content	HBase is the hadoop database.		
rowkey2	t1	...	...	...	...

- 表由行构成
- 每行都对应一个row key，且包含多个列
- 每个列都属于某个列族

数据模型——基本概念

- row key
  - 行键，表中的行根据行的键值进行排序，数据按照Row key的字典序排序存储
- column family
  - 列族， HBase表中的每个列都归属于某个列族
- column qualifier
  - 列修饰符，通过列族:列修饰符来指定列
  - 客户端随时可以把列添加到列族
  - 每行的列可以不同
- timestamp
  - 版本号，表示数据的版本
  - 默认为数据插入时的时间戳，可自定义
  - 多版本的数据按照时间倒序排序，最新的数据版本排在最前面

数据模型——多维映射表

- 表内的值通过三个键唯一索引
- rowKey (ASC) + column(ASC) + timestamp(DESC) => value

```
SortedMap（
    RowKey，List（
        SortedMap（
            Column，List（
                Value，Timestamp
                ）
            ）
        ）
）
```

数据模型——物理视图

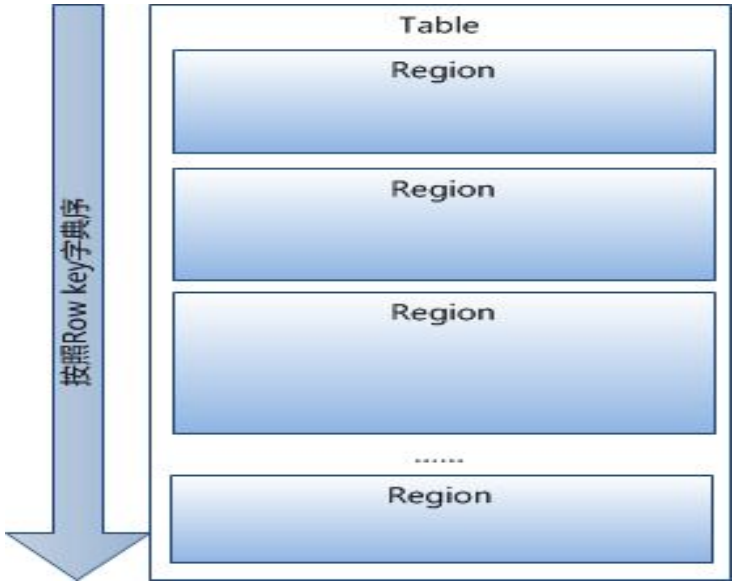
- 面向列
- 稀疏

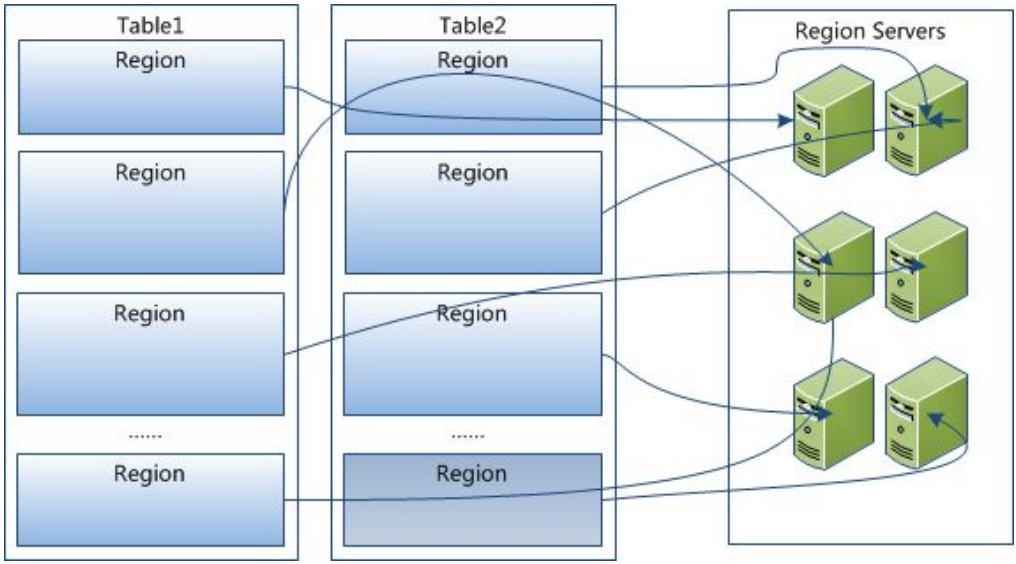
RowKey	Time Stamp	Column Family:article	
		列	值
rowkey1	t4	article:title	HBase in Action
	t4	article:content	HBase is the hadoop database.
	t2	article:title	HBase

RowKey	Time Stamp	Column Family:author	
		列	值
rowkey1	t3	author:name	lee
	t2	author:nickname	nicholas

物理储存——Region

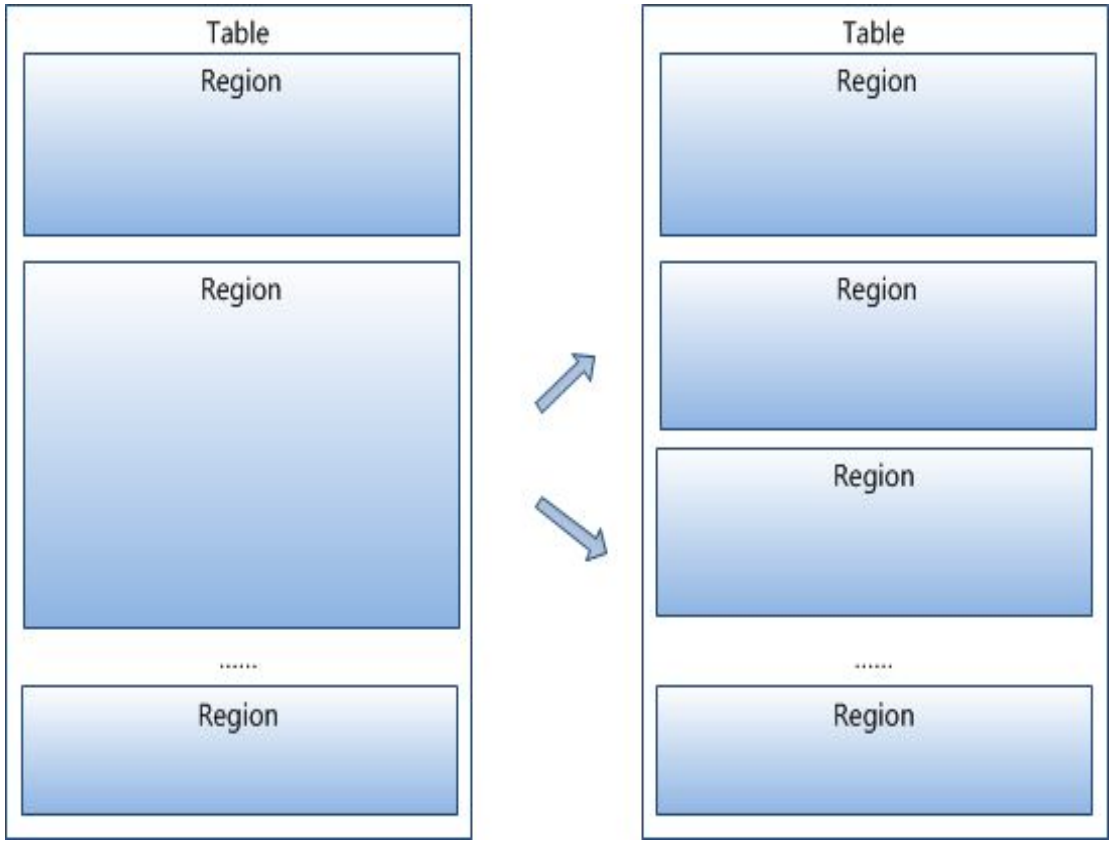
- Table中的所有行都按照row key的字典序排列
- Table 在行的方向上分割为多个Region
- Region保存在Region Server上





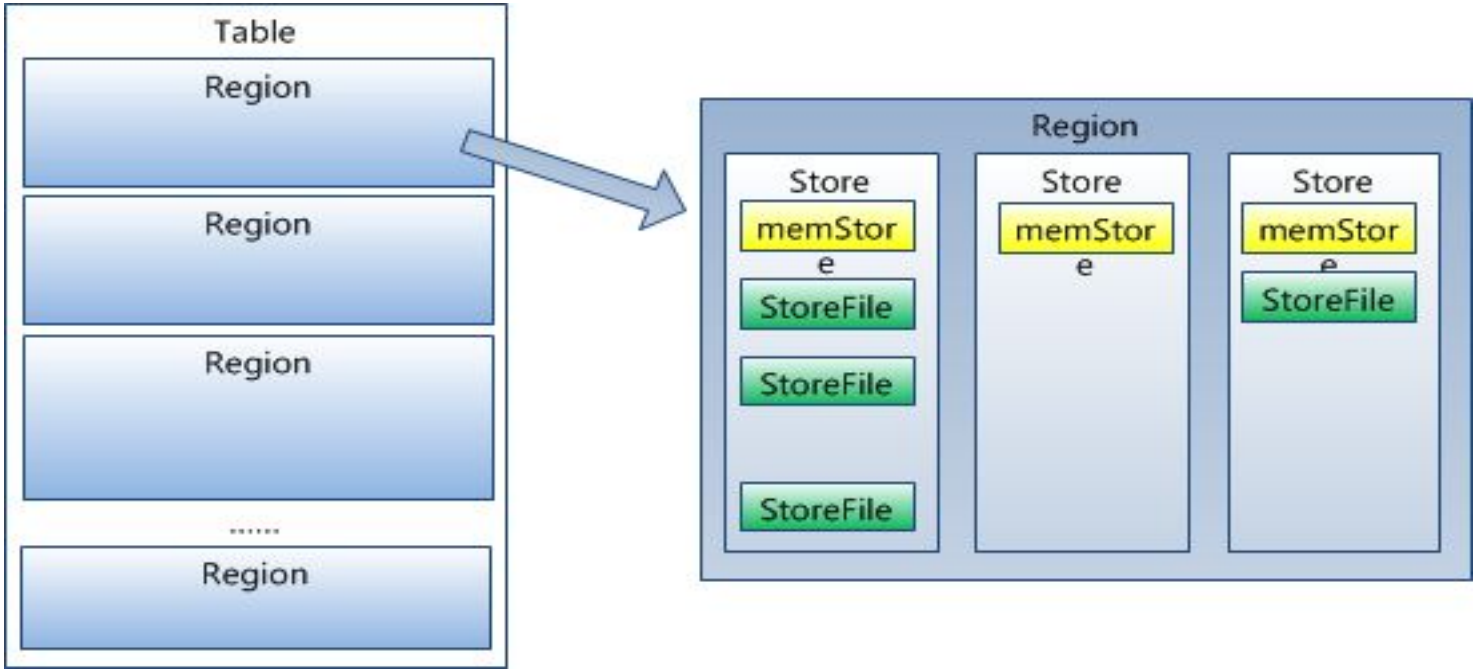
物理储存——Region

- 默认情况下，每个表一开始只有一个Region；
- 随着数据不断插入表，Region不断增大，当增大到一个阀值时，Region就会分裂为两个新的Region；



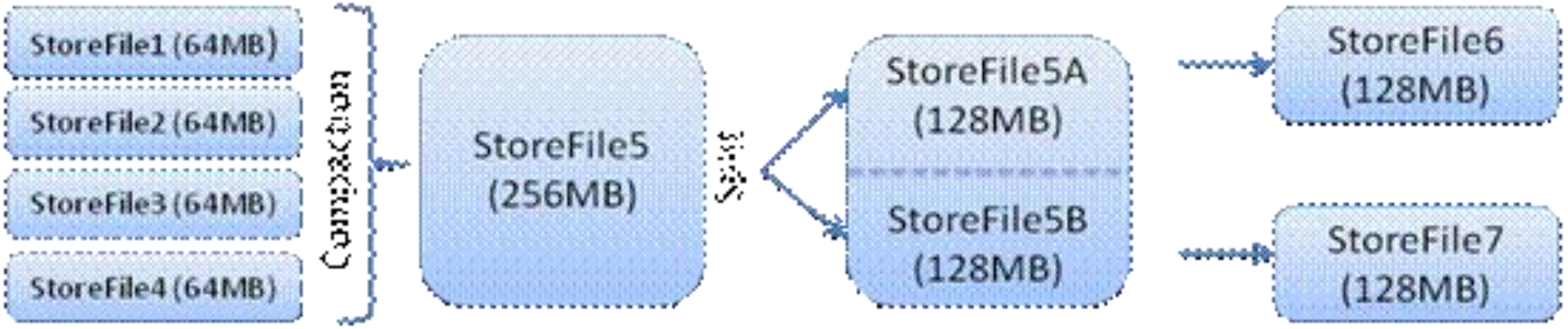
物理储存——HStore

- Region由若干个HStore组成，每个HStore储存一个列族的数据
- HStore由MemStore和StoreFiles构成
- MemStore位于内存
- StoreFile以HFile的格式保存在HDFS上



物理储存——HStore

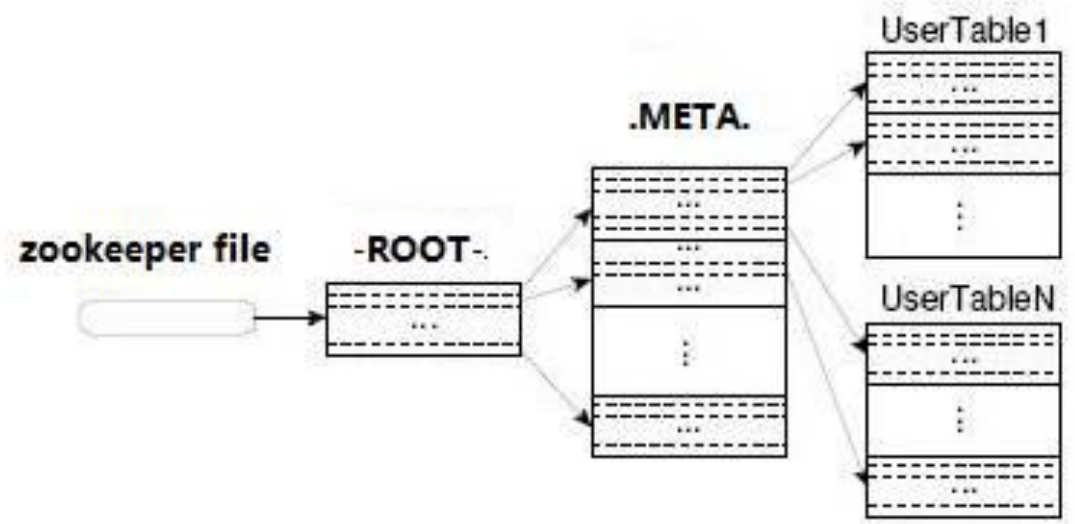
- Flush： 用户写入的数据首先会放入MemStore，当MemStore满了以后会Flush成一个StoreFile
- Compact： 当StoreFile文件数量增长到一定阈值，会触发Compact合并操作，将多个StoreFiles合并成一个StoreFile，合并过程中会进行版本合并和数据删除
- Split：当单个StoreFile大小超过一定阈值后，会触发Split操作，同时把当前Region Split成2个Region；父Region会下线，新Split出的2个孩子Region会被HMaster分配到相应的HRegionServer上



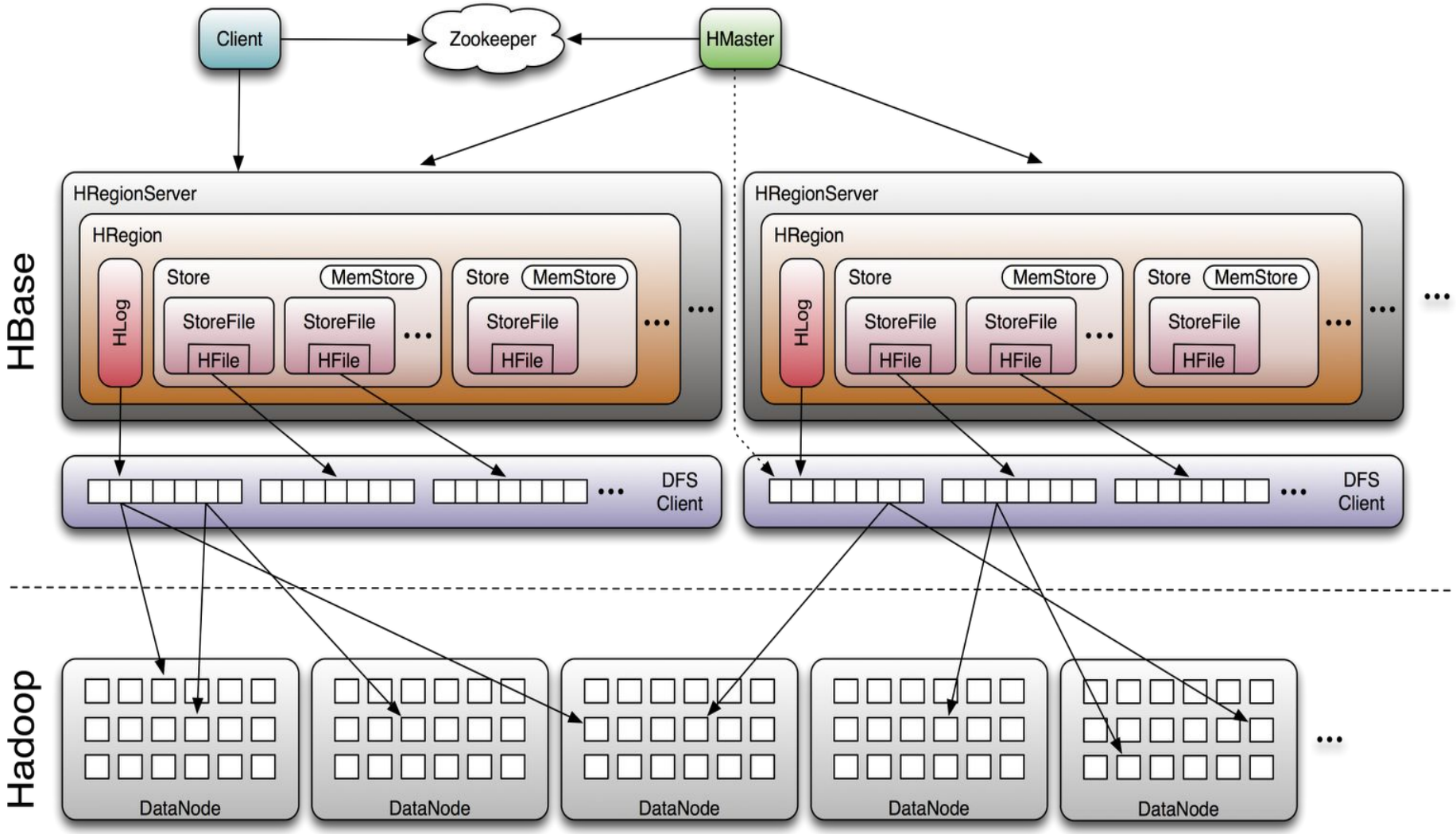
物理储存 —— -ROOT- 与 .META.

HBase中有两张特殊的Table，-ROOT-和.META.

- .META.：记录了用户表的Region信息，.META.可以有多个Region
- -ROOT-：记录了.META.表的Region信息，-ROOT-只有一个region
- Zookeeper中记录了-ROOT-表的location



系统架构



系统架构

- **Client**
  - 包含访问HBase的接口并维护cache来加快对HBase的访问
- **Zookeeper**
  - 保证任何时候，集群中只有一个HMaster
  - 实时监控Region Server的上线和下线信息，并实时通知给HMaster
  - 存储了-ROOT-表的位置
- **HMaster**
  - 为Region server分配region，负责Region server的负载均衡
  - 发现失效的Region server并对其上的Region进行迁移
  - 管理用户对table的增删改查操作
- **Region Server**
  - Region server维护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
  - 包含一个HLog，是预写式日志，用于灾难恢复

操作

- 对表的访问必须通过row key
  - 通过单个row key访问
  - 通过row key的range
  - 全表扫描

- HBase的写操作是锁行的
  - 每一行都是一个原子元素
  - 无论对行中任何的列进行修改，都会对行加锁

操作——交互接口

- Native Java API
  - 最常规和高效的访问方式，适合Hadoop MapReduce Job并行批处理HBase表数据
- HBase Shell
  - HBase的命令行工具，最简单的接口，适合HBase管理使用
- Thrift Gateway
  - 利用Thrift序列化技术，支持C++，PHP，Python等多种语言，适合其他异构系统在线访问HBase表数据
- REST Gateway
  - 支持REST 风格的Http API访问HBase, 解除了语言限制
- Pig
- Hive
- Spark

Hbase特点——Hbase VS. RDBMS

	HBase	RDBMS
数据类型	只有字符串（字节数组）	丰富的数据类型
数据操作	简单的增删改查	各种各样的函数，表连接
存储模式	基于列存储	基于表格结构和行存储
数据更新	更新后旧版本仍然会保留	替换
可伸缩性	轻易的进行增加节点，兼容性高	需要中间层，牺牲功能

Hbase特点——Hbase的适用场景

- 半结构化或非结构化数据
- 记录非常稀疏
- 多版本数据
- 超大数据量
- 需要高效的随机读写能力
- 不需要完整的关系数据库功能
  - 二级索引
  - join
  - 跨行/表的事务处理
  - ……

参考资料

资料	链接
淘宝Hbase介绍	http://www.searchtb.com/2011/01/understanding-hbase.html
Hbase的数据模型	http://www.cnblogs.com/NicholasLee/archive/2012/09/13/2683272.html
Hbase参考指南	http://hbase.apache.org/apache_hbase_reference_guide.pdf
Hbase简介——京东	http://wenku.baidu.com/view/9cfe96eb240c844769eaeed1.html

Thank you!

www.uc.cn