HBase2 入门介绍

背景

> RDBMS的不足：为了访问超大规模的数据集，很多厂商提供了复制和分区解决办法，让数据库能够从单节点上扩展出去，但是这些附加的技术都属于“事后”的解决办法，而且非常难维护。并且这些解决办法常常要牺牲一些重要的RDBMS特性。但是执行连接、复杂查询、触发器、视图以及外键约束这些功能要么开销大，要么不能使用。
>
> HBase的优点：HBase自底向上地进行构建，能够简单地通过增加节点来达到线性扩展，HBase并不是关系型数据库，它不支持SQL。

基本介绍

简介

HBase是bigtable的开源java版本。是建立在hdfs之上，提供高可靠、高性能、列存储、可伸缩、实时读写nosql的数据库系统。

简介：

- 它介于nosql和RDBMS之间，仅能通过主键（row key）和主键的range来检索数据，仅支持单行事务。
- 主要用来存储结构化和半结构化的松散数据，HBase查询数据功能很简单，不支持join操作，不支持复杂的事务。
- HBase中支持的数据类型：byte[]
- 与hadoop一样，HBase目标主要依靠横向发展，通过不断增加廉价的商用服务器来增加计算和存储能力。

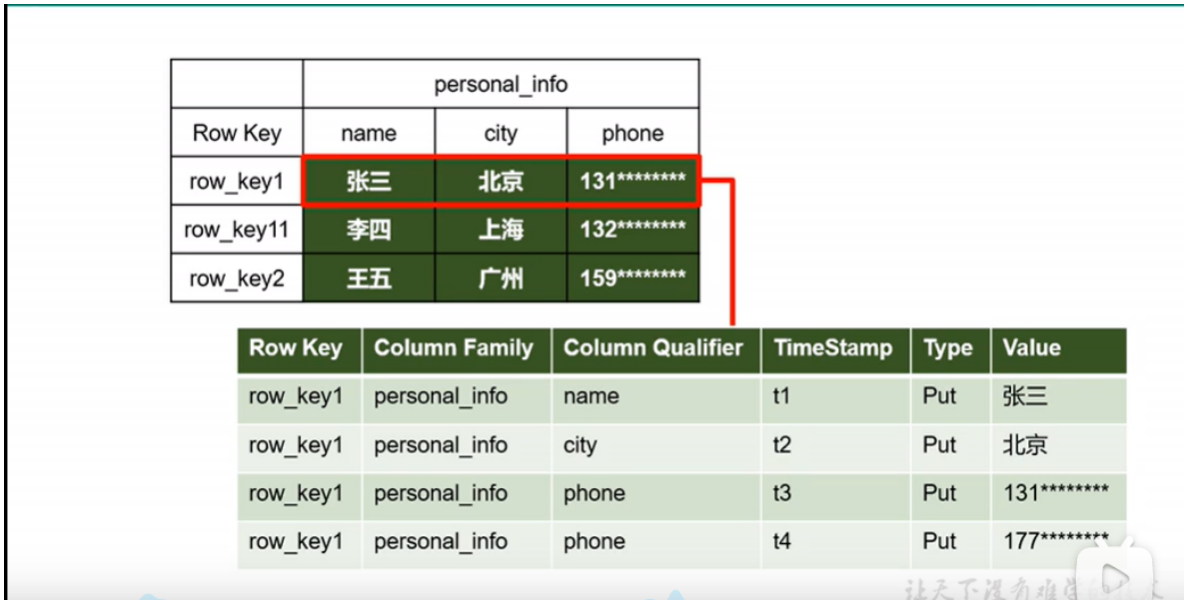
传统数据表

HBase的表模型

逻辑结构

- Row key 采用字典序排列

物理结构



HBase与Hadoop的关系

HDFS

- 为分布式存储提供文件系统
- 针对存储大尺寸的文件进行优化，不适用对HDFS上的文件进行随机读写
- 直接使用文件
- 数据模型不灵活
- 使用文件系统和处理框架
- 优化一次写入，多次读取的方式

HBase

- 提供表状的面向列的数据存储
- 针对表状数据的随机读写进行优化
- 使用key-value操作数据
- 提供灵活的数据模型
- 使用表状存储，支持MapReduce，依赖HDFS
- 优化了多次读，以及多次写

RDBMS与HBase对比

RDBMS（关系型数据库）

结构：

- 数据库以表的形式存在
- 支持FAT、NTFS、EXT、文件系统
- 使用Commit log存储日志
- 参考系统是坐标系统
- 使用主键（PK）
- 支持分区
- 使用行、列、单元格

功能：

- 支持向上扩展
- 使用SQL查询
- 面向行，即每一行都是一个连续的单元
- 数据总量依赖于服务器配置
- 具有ACID支持
- 适合结构化数据
- 传统关系型数据库一般都是中心化的
- 支持事务
- 支持Join

HBase

结构：

- 数据库以region的形式存在
- 支持HDFS文件系统
- 使用WAL（Write-Ahead Logs）存储日志
- 参考系统是Zookeeper
- 使用行键（row key）
- 支持分片
- 使用行、列族和单元格

功能

- 支持向外扩展
- 使用API和MapReduce来访问HBase表数据
- 面向列，即每一列都是一个连续的单元
- 数据总量不依赖具体某台机器
- HBase不支持ACID(Atomicity原子性、Consistency一致性、Isolation隔离性、Durability耐久性)
- 适合结构化数据和非结构化数据
- 一般都是分布式得
- HBase不支持事务
- 不支持SQL
- 不支持Join

HBase特点

海量存储

HBase适合存储PB级别的海量数据，在PB级别的数据以及采用廉价的PC存储的情况下，能在几十到百毫秒内返回数据。这与HBase的极易扩展性息息相关。

列式存储

这里的列式存储是列族存储，HBase是根据列族来存储数据的，列族下面可以有非常多的列，列族在创建表的时候必须指定。需要注意的时候，列族理论上可以很多，但实际上不要超过6个

极易扩展

两个方面： 基于上层处理能力（RegionServer）扩展，基于存储的扩展（HDFS）

通过横向添加RegionSever的机器，进行水平扩展，提升Hbase上层的处理能力，提升Hbsae服务更多Region的能力。

备注：RegionServer的作用是管理region、承接业务的访问，这个后面会详细的介绍通过横向添加Datanode的机器，进行存储层扩容，提升Hbase的数据存储能力和提升后端存储的读写能力。

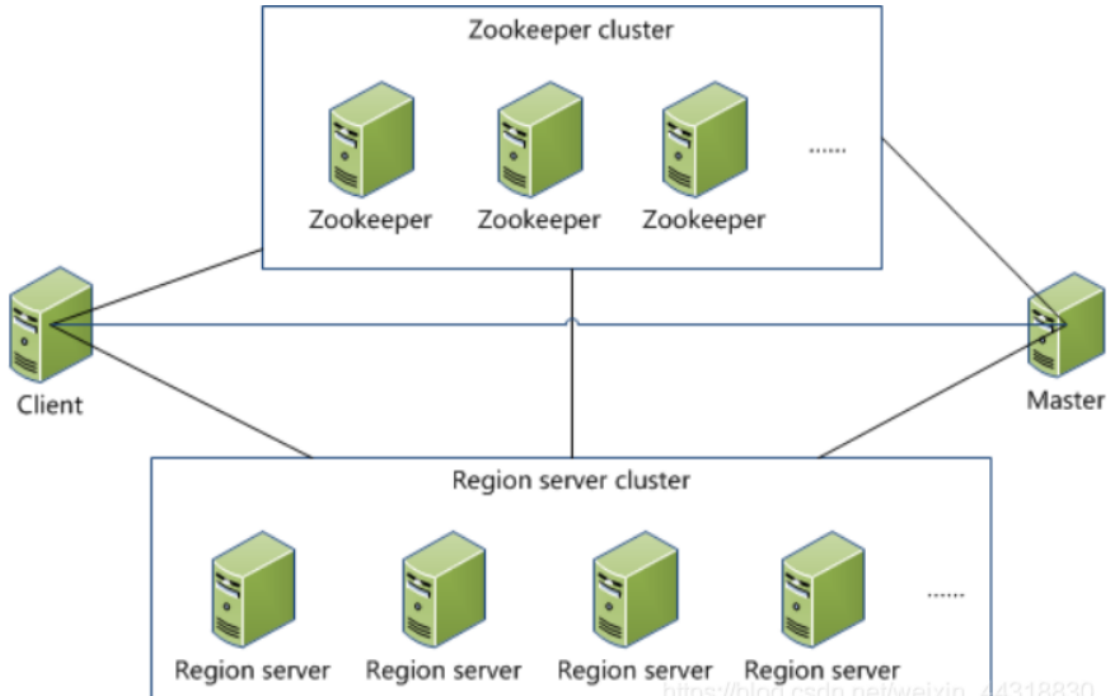
高并发

由于目前大部分使用Hbase的架构，都是采用的廉价PC，因此单个IO的延迟其实并不小，一般在几IO延迟下降并不多。能获得高并发、低延迟的服务。

稀疏

稀疏主要是针对Hbase列的灵活性，在列族中，你可以指定任意多的列，在列数据为空的情况下，是不会占用存储空间的。

HBase的基础架构

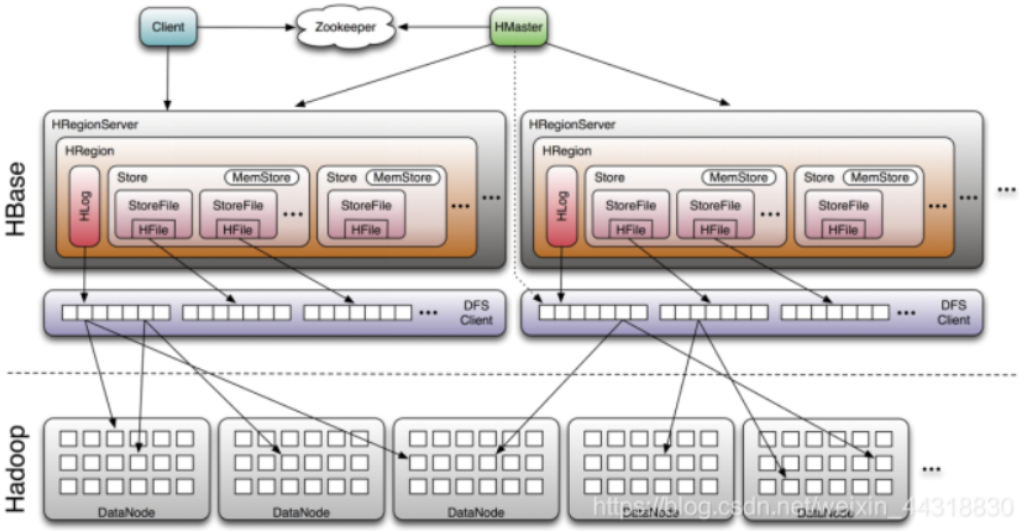


HMaster

功能：
1. 监控RegionServer
2. 处理RegionServer故障转移
3. 处理元数据的变更
4. 处理region的分配和移除
5. 在空闲时间进行数据的负载均衡
6. 通过Zookeeper发布自己的位置给客户端

RegionServer

功能：
1. 负责存储HBase的实际数据
2. 处理分配给它的Region
3. 刷新缓存到HDFS
4. 维护HLog
5. 执行压缩
6. 负责处理Region分片



Write-Ahead logs

HBase的修改记录，当对HBase读写数据的时候，数据不是直接写进磁盘，它会在内存中保留一段时间（时间以及数据量阈值可以设定）。但把数据保存在内存中可能有更高的概率引起数据丢失，为了解决这个问题，数据会先写在一个叫做Write-Ahead logfile的文件中，然后再写入内存中。所以在系统出现故障的时候，数据可以通过这个日志文件重建。

StoreFile(HFile)

这是在磁盘上保存原始数据的实际的物理文件，是实际的存储文件。

Store

HFile存储在Store中，一个Store对应HBase表中的一个列族。

MenStore

顾名思义，就是内存存储，位于内存中，用来保存当前的数据操作，所以当数据保存在WAL中之后，RegsionServer会在内存中存储键值对。

Region

Hbase表的分片，HBase表会根据RowKey值被切分成不同的region存储在RegionServer中，在一个RegionServer中可以有多个不同的region。