尚硅谷大数据技术之HBase基础

（作者：尽际）

官网：<http://www.atguigu.com>

# HBase的起源

HBase的原型是Google的BigTable论文，受到了该论文思想的启发，目前作为Hadoop的子项目来开发维护，用于支持结构化的数据存储。

官方网站：<http://hbase.apache.org>

\* 2006年Google发表BigTable白皮书

\* 2006年开始开发HBase

\* 2008年北京成功开奥运会，程序员默默地将HBase弄成了Hadoop的子项目

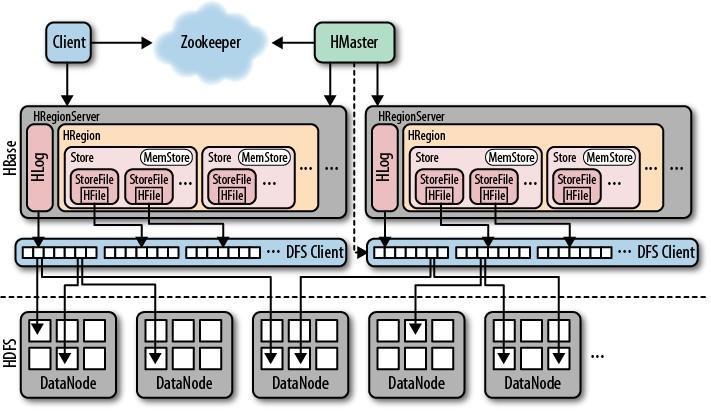
\* 2010年HBase成为Apache顶级项目

\* 现在很多公司二次开发出了很多发行版本，你也开始使用了

# 二、基于Hadoop的HBase架构C:\Users\61661\AppData\Local\Temp\1501410678(1).png

HBase内置有zookeeper，但一般我们会有其他的Zookeeper集群来监管master和regionserver，Zookeeper通过选举，保证任何时候，集群中只有一个活跃的HMaster，HMaster与HRegionServer 启动时会向ZooKeeper注册，存储所有HRegion的寻址入口，实时监控HRegionserver的上线和下线信息。并实时通知给HMaster，存储HBase的schema和table元数据，默认情况下，HBase 管理ZooKeeper 实例，Zookeeper的引入使得HMaster不再是单点故障。一般情况下会启动两个HMaster，非Active的HMaster会定期的和Active HMaster通信以获取其最新状态，从而保证它是实时更新的，因而如果启动了多个HMaster反而增加了Active HMaster的负担。

一个RegionServer可以包含多个HRegion，每个HRegion维护一个HLog，和多个HFiles以及其对应的MemStore。RegionServer运行于DataNode上，数量可以与DatNode数量一致，请参考如下架构图：



# 三、RDBMS与HBase的对比

## 3.1、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

## 3.2、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、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，而取决于机器数量

\* HBase不支持ACID（Atomicity、Consistency、Isolation、Durability）

\* 适合结构化数据和非结构化数据

\* 一般都是分布式的

\* HBase不支持事务

\* 不支持Join

# 四、HBase特征简要

### 4.1、自动故障处理和负载均衡

HBase运行在HDFS上，所以HBase中的数据以多副本形式存放，数据也服从分布式存放，数据的恢复也可以得到保障。另外，HMaster和RegionServer也是多副本的。

### 4.2、自动分区

HBase表是由分布在多个RegionServer中的region组成的，这些RegionServer又分布在不同的DataNode上，如果一个region增长到了一个阈值，为了负载均衡和减少IO，HBase可以自动或手动干预的将region切分为更小的region，也称之为subregion。

### 4.3、集成Hadoop/HDFS

虽然HBase也可以运行在其他的分布式文件系统之上，但是与HDFS结合非常之方便，而且HDFS也非常之流行。

### 4.4、实时随机大数据访问

HBase采用log-structured merge-tree作为内部数据存储架构，这种架构会周期性地将小文件合并成大文件以减少磁盘访问同时减少NameNode压力。

### 4.5、MapReduce

HBase内建支持MapReduce框架，更加方便快速，并行的处理数据。

### 4.6、Java API

HBase提供原声的Java API支持，方便开发。

### 4.7、横向扩展

HBase支持横向扩展，这就意味着如果现有服务器硬件性能出现瓶颈，不需要停掉现有集群提升硬件配置，而只需要在现有的正在运行的集群中添加新的机器节点即可，而且新的RegionServer一旦建立完毕，集群会开始重新调整。

### 4.8、列存储

HBase是面向列存储的，每个列都单独存储，所以在HBase中列是连续存储的，而行不是。

### 4.9、HBase Shell

HBase提供了交互式命令行工具可以进行创建表、添加数据、扫描数据、删除数据等操作和其他一些管理命令。

# 五、HBase在集群中的定位

HBase一种是作为存储的分布式文件系统，另一种是作为数据处理模型的MR框架。因为日常开发人员比较熟练的是结构化的数据进行处理，但是在HDFS直接存储的文件往往不具有结构化，所以催生出了HBase在HDFS上的操作。如果需要查询数据，只需要通过键值便可以成功访问。

# 六、HBase内部存储架构

HBase是由row key，column family，column和cell组成，row key确定唯一的一行，column family由若干column组成，column是表的字段，cell存储了实际的值或数据。

# 七、HBase与Hadoop

## 7.1、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，不需要对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

## 7.2、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

# 八、HBase的优缺点

## 8.1、优点

\* 方便高效的压缩数据

\* 支持快速数据检索

\* 管理和配置简单，支持横向扩展，所以非常容易扩展

\* 聚合查询性能非常高

\* 可高效地进行分区，提供自动分区机制把大的region切分成小的subregion

## 8.2、缺点

\* 对JOIN以及多表合并数据的查询性能不好

\* 更新过程中有大量的写入和删除操作，需要频繁合并和分裂，降低存储效率

\* 对关系模型支持不好，分区和索引模式设计比较困难。

# 九、HBase的环境角色

## 9.1、HMaster

### 9.1.1、功能描述

\* 监控RegionServer

\* 处理RegionServer故障转移

\* 处理元数据的变更

\* 处理region的分配或移除

\* 在空闲时间进行数据的负载均衡

\* 通过Zookeeper发布自己的位置给客户端

## 9.2、RegionServer

### 9.2.1、功能描述

\* 负责存储HBase的实际数据

\* 处理分配给它的Region

\* 刷新缓存到HDFS

\* 维护HLog

\* 执行压缩

\* 负责处理Region分片

### 9.2.2、内含组件

**\* 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）。如果机器突然原地爆炸，把数据保存在内存中会引起数据丢失，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。

**\* 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

**\* MemStore**

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

## 9.3、Zookeeper

HMaster与HRegionServer 启动时会向ZooKeeper注册，存储所有HRegion的寻址入口，实时监控HRegionserver的上线和下线信息。并实时通知给HMaster，存储HBase的schema和table元数据，默认情况下，HBase 管理ZooKeeper 实例，Zookeeper的引入使得HMaster不再是单点故障。一般情况下会启动两个HMaster，非Active的HMaster会定期的和Active HMaster通信以获取其最新状态，从而保证它是实时更新的，因而如果启动了多个HMaster反而增加了Active HMaster的负担。

# 十、使用场景的探讨

## 10.1、何时使用

\* 如果数据有很多列，且包含很多空字段

\* 数据包含了不定数量的列

\* 需要维护数据的版本

\* 需要很高的横向扩展性

\* 需要大量的压缩数据

\* 需要大量的I/O

一般而言数百万行的数据和频率不高的读写操作，是不需要HBase的，如果有几十亿列数据，同时在单位时间内有数以千、万记的读写操作，可以考虑HBase。

## 10.2、何时不使用

\* 数据总量不大时（比如就几个G）

\* 当需要JOIN以及关系型数据库的一些特性时

\* 如果关系型数据库可以满足需求

# 十一、HBase的安装与部署

## 10.1、Zookeeper集群的正常部署并启动

$ /opt/modules/cdh/zookeeper-3.4.5-cdh5.3.6/bin/zkServer.sh start

## 10.2、Hadoop集群的正常部署并启动

$ /opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/sbin/start-dfs.sh

$ /opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/sbin/start-yarn.sh

## 10.3、解压HBase

$ tar -zxf /opt/softwares/hbase-0.98.6-cdh5.3.6.tar.gz -C /opt/modules/cdh/

## 10.4、修改HBase配置文件

### 10,.4.1、hbase-env.sh

C:\Users\61661\AppData\Local\Temp\1501432507(1).png

C:\Users\61661\AppData\Local\Temp\1501432485(1).png

### 10.4.2、hbase-site.xml



### 10.4.3、regionservers



## 10.5、替换HBase根目录下的lib目录下的jar包，以解决兼容问题

**\* 删除原有Jar包**

$ rm -rf /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/lib/hadoop-\*$ rm -rf lib/zookeeper-3.4.6.jar

（尖叫提示：如果lib目录下的zookeeper包不匹配也需要替换）

**\* 拷贝新的Jar包**

这里涉及到的jar包大概是：

hadoop-annotations-2.5.0.jar

hadoop-auth-2.5.0-cdh5.3.6.jar

hadoop-client-2.5.0-cdh5.3.6.jar

hadoop-common-2.5.0-cdh5.3.6.jar

hadoop-hdfs-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-app-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-common-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-core-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-hs-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-hs-plugins-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-jobclient-2.5.0-cdh5.3.6.jar

hadoop-mapreduce-client-jobclient-2.5.0-cdh5.3.6-tests.jar

hadoop-mapreduce-client-shuffle-2.5.0-cdh5.3.6.jar

hadoop-yarn-api-2.5.0-cdh5.3.6.jar

hadoop-yarn-applications-distributedshell-2.5.0-cdh5.3.6.jar

hadoop-yarn-applications-unmanaged-am-launcher-2.5.0-cdh5.3.6.jar

hadoop-yarn-client-2.5.0-cdh5.3.6.jar

hadoop-yarn-common-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-applicationhistoryservice-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-common-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-nodemanager-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-resourcemanager-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-tests-2.5.0-cdh5.3.6.jar

hadoop-yarn-server-web-proxy-2.5.0-cdh5.3.6.jar

zookeeper-3.4.5-cdh5.3.6.jar

**我们可以通过find命令快速进行定位，例如我们可以执行：**

$ find /opt/modules/ -name hadoop-hdfs-2.5.0-cdh5.3.6.jar



然后将查找出来的Jar包根据指定位置复制到HBase的lib目录下，在这里我给大家整合好到一个文件夹中了，请依次执行：

$ tar -zxf /opt/softwares/CDH\_HadoopJar.tar.gz -C /opt/softwares/

$ cp -a /opt/softwares/HadoopJar/\* /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/lib/

## 10.6、将整理好的HBase安装目录scp到其他机器节点

$ scp -r /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/ \

hadoop-senior02.itguigu.com:/opt/modules/cdh/

$ scp -r /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/ \

hadoop-senior03.itguigu.com:/opt/modules/cdh/

## 10.7、将Hadoop配置文件软连接到HBase的conf目录下

**\* core-site.xml**

$ ln -s /opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop/core-site.xml /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/conf/core-site.xml

**\* hdfs-site.xml**

$ ln -s /opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop/hdfs-site.xml /opt/modules/cdh/hbase-0.98.6-cdh5.3.6/conf/hdfs-site.xml

（尖叫提示：不要忘记其他几台机器也要做此操作）

## 10.8、启动服务

$ bin/hbase-daemon.sh start master

$ bin/hbase-daemon.sh start regionserver

或者：

$ bin/start-hbase.sh

对应的停止命令：

$ bin/stop-hbase.sh

## 10.9、查看页面

启动成功后，可以通过主机名:60010地址来访问HBase的管理页面

例如，<http://hadoop-senior01.itguigu.com:60010>

# 十二、HBase常用操作

## 12.1、进入HBase客户端命令操作界面

$ bin/hbase shell

## 12.2、查看帮助命令

hbase(main):001:0> help

## 12.3、查看当前数据库中有哪些表

hbase(main):002:0> list

## 12.4、创建一张表

hbase(main):003:0> create 'student','info'

## 12.5、向表中存储一些数据

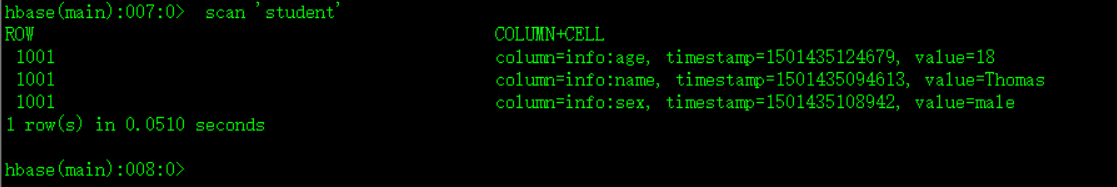
hbase(main):004:0> put 'student','1001','info:name','Thomas'

hbase(main):005:0> put 'student','1001','info:sex','male'

hbase(main):006:0> put 'student','1001','info:age','18'

## 12.6、扫描查看存储的数据

hbase(main):007:0> scan 'student'

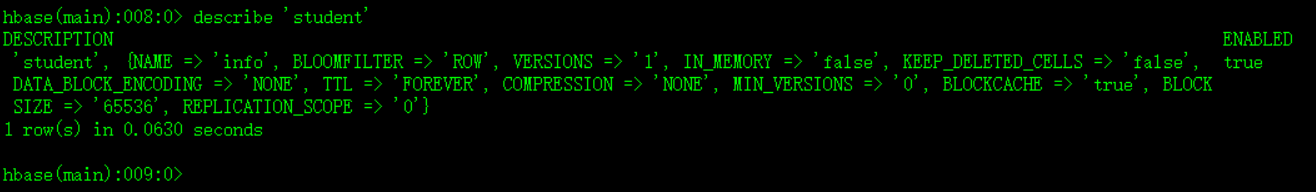


或：查看某个rowkey范围内的数据

hbase(main):014:0> scan 'student',{STARTROW => '1001',STOPROW => '1007'}

## 12.7、查看表结构

hbase(main):009:0> describe ‘student’

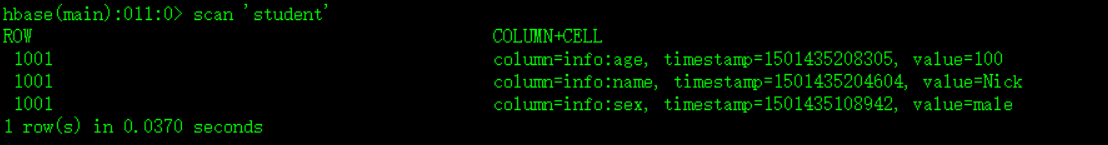


## 12.8、更新指定字段的数据

hbase(main):009:0> put 'student','1001','info:name','Nick'

hbase(main):010:0> put 'student','1001','info:age','100'

查看更新后的数据：



## 12.9、查看指定行的数据

hbase(main):012:0> get 'student','1001'

或：查看指定行指定列或列族的数据

hbase(main):013:0> get 'student','1001','info:name'

## 12.10、删除数据

### 12.10.1、删除某一个rowKey全部的数据

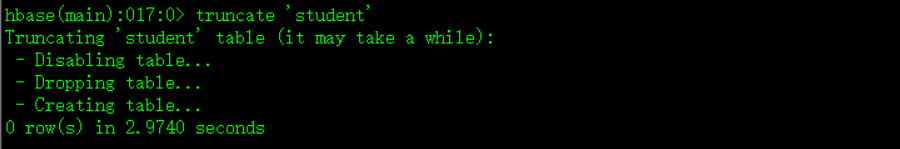
hbase(main):015:0> deleteall 'student','1001'

### 12.10.2、删除掉某个rowKey中某一列的数据

hbase(main):016:0> delete 'student','1001','info:sex'

## 12.11、清空表数据

hbase(main):017:0> truncate 'student'



## 12.12、删除表

首先需要先让该表为disable状态，使用命令：

hbase(main):018:0> disable 'student'然后才能drop这个表，使用命令：hbase(main):019:0> drop 'student'(尖叫提示：如果直接drop表，会报错：Drop the named table. Table must first be disabled)

## 12.13、统计一张表有多少行数据

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

# 十三、HMaster的高可用

## 13.1、确保HBase集群已正常停止

$ bin/stop-hbase.sh

## 13.2、在conf目录下创建backup-masters文件

$ touch conf/backup-masters

## 13.3、在backup-masters文件中配置高可用HMaster节点

$ echo hadoop-senior02.itguigu.com > conf/backup-masters

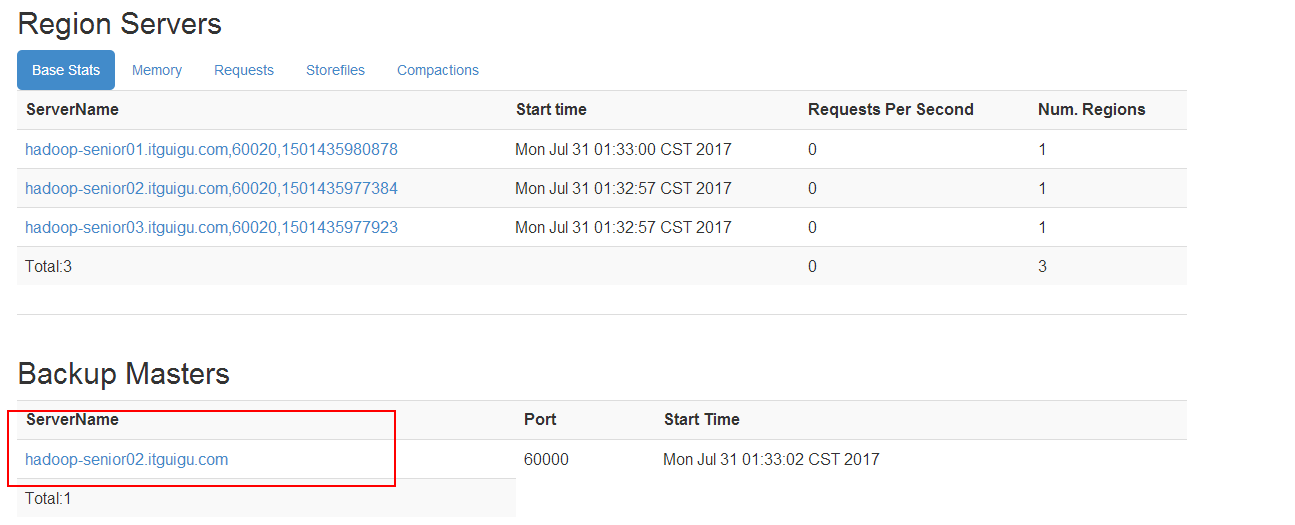
## 13.4、将整个conf目录scp到其他节点

$ scp -r conf/ hadoop-senior02.itguigu.com:/opt/modules/cdh/hbase-0.98.6-cdh5.3.6/

$ scp -r conf/ hadoop-senior03.itguigu.com:/opt/modules/cdh/hbase-0.98.6-cdh5.3.6/

## 13.5、打开页面测试

<http://hadoop-senior01.itguigu.com:60010>



最后，可以尝试关闭第一台机器的HMaster：

$ bin/hbase-daemon.sh stop master

然后查看第二台的HMaster是否会直接启用