## https://images2018.cnblogs.com/blog/1228818/201804/1228818-20180402125111282-1966599087.png

## Hbase角色介绍：

1个RegionServer中只记录一份WAL日志，HLog，管理多个Region（table），一个Region中管理多个Store，一个store对应Hbase表中的一个列簇

一个Store中含有一个MemStore，多个HFILE文件

### ZooKeeper作用：

1. Hmaster HA

2. Hbase元数据入口

3. 监控HRegionServer状态，通知Master处理（包括Region拆分）

### Hmaster主要作用：

1.为RegionServer分配Region

2.维护集群的负载均衡

3.维护集群的元数据信息（ZK）

4.发现失效的RegionServer，并将失效的RS进行故障转移到正常的RegionServer

Storefile中的MemStore恢复 ，通过HLog恢复

### HRegionServer作用：

1.处理Client的读写请求

2.负责与底层HDFS的交互，存储数据到HDFS

3.执行Region变大之后的拆分动作

4.负责Storefile的合并工作

### HDFS作用：

1.存储物理文件HFILE

2.存储Hlog

## Hbase Shell 操作：

### 1. 进入Hbase客户端Shell

bin/hbase shell

### 2. 查看帮助命令

help

### 3. 查看当前数据库中有哪些表

list

### 4. 创建表，列簇

create 'student','info'

### 5. 插入数据 （数据存储使用byte数组）

put 'student','1001','info:sex','male'

put 'student','1001','info:sex','female'

put 'student','1001','info:age','19'

put 'student','1001','info:age','29'

put 'student','1002','info:sex','male'

put 'student','1002','info:sex','female'

put 'student','1002','info:name','xintong'

put 'student','1002','info:age','10'

put 'student','1002','info:age','50'

### 6. 查看表 全表扫描

scan 'student' === scan "student"

--从指定行开始查询

scan 'student',{STARTROW=>'1002'}

-- 指定开始结束行查询

scan 'student',{STARTROW=>'1002',STOPROW=>'1002'}

### 7. 查看单行数据

get 'student','1001'

get 'student','1002'

### 8. 查看单行数据,指定列簇和列

get 'student','1002','info:name'

get 'student','1002','info:sex'

### 9. 查看表的行数

count 'student'

### 10.表描述

describe 'student'

### 11. 修改版本数

alter 'student',{NAME=>'info',VERSIONS => '3'}

### 12.查看多个版本数据

get 'student','1002',{COLUMN=>'info:sex',VERSIONS=>2}

### 13. 删除列簇中某一列cell

delete 'student','1001','info:sex'

### 14. 删除ROWKEY对应所有数据

deleteall 'student','1001'

### 15. 清空表

truncate 'student'

### 16. 删除表操作

#### 1. 让表下线，变为不可用

disable 'student'

#### 2. 执行删除操作

drop 'student'

list查看表是否被删除

### 17. 命名空间（类似于数据库的概念）

--命名空间用来管理表

list\_namespace

--创建命名空间

create\_namespace 'B6'

--指定命名空间下创建表名

create 'B6:staff','info'

--删除命名空间 ， Drop the named namespace. The namespace must be empty.

drop\_namespace 'B6'

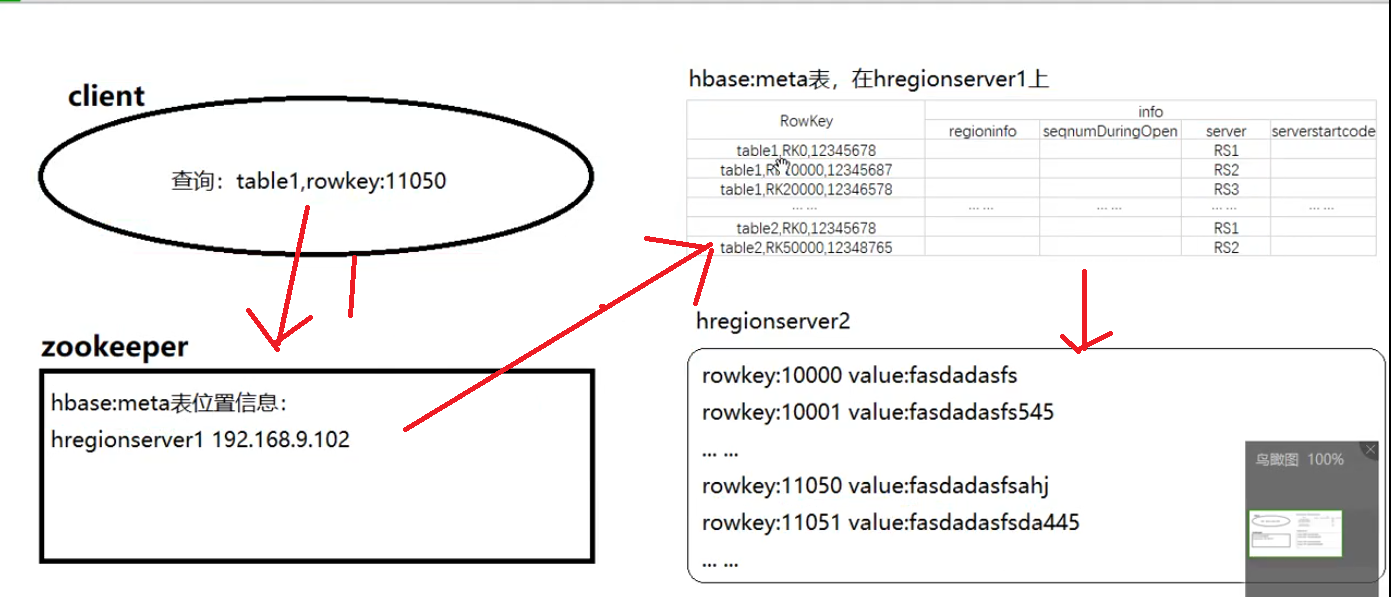
### Hbase读写流程

读流程与Hmaster挂不挂没有任何关系，

读取流程：MemStore---》blockcache---》磁盘

数据读取到，先写入BlockCache , 再返回数据

## Hbase读流程图



### 触发flush的三个机制

（RegionServer上所有MemStore都会发生flush操作）

1. Regionserver 节点上MemStore内存总和超过机器总内存40%以上，flush会阻塞客户端读写

<property>

<name>hbase.regionserver.global.memstore.size</name>

<value></value>

<description>Maximum size of all memstores in a region server before new

updates are blocked and flushes are forced. Defaults to 40% of heap (0.4).

Updates are blocked and flushes are forced until size of all memstores

in a region server hits hbase.regionserver.global.memstore.size.lower.limit.

The default value in this configuration has been intentionally left emtpy in order to

honor the old hbase.regionserver.global.memstore.upperLimit property if present.</description>

</property>

2.内存中的文件在自动刷新之前能够存活的最长时间，默认是1H，Regionserver级别

<property>

<name>hbase.regionserver.optionalcacheflushinterval</name>

<value>3600000</value>

<description>

Maximum amount of time an edit lives in memory before being automatically flushed.

Default 1 hour. Set it to 0 to disable automatic flushing.</description>

</property>

3. 针对单个hregion里memstore缓存大小达到128M，HRegion级别

<property>

<name>hbase.hregion.memstore.flush.size</name>

<value>134217728</value>

<description>

Memstore will be flushed to disk if size of the memstore

exceeds this number of bytes. Value is checked by a thread that runs

every hbase.server.thread.wakefrequency.</description>

</property>

### Compact Hfile小文件

1.一个region进行major compaction合并的周期，在这个点的时候，这个region下的所有的hfile会进行合并，默认7天合并一次小文件。major compaction 比较耗资源，建议生产关闭（设置为0），在应用空闲时间手动触发。

<property>

<name>hbase.hregion.majorcompaction</name>

<value>604800000</value>

<description>The time (in miliseconds) between 'major' compactions of all

HStoreFiles in a region. Default: Set to 7 days. Major compactions tend to

happen exactly when you need them least so enable them such that they run at

off-peak for your deploy; or, since this setting is on a periodicity that is

unlikely to match your loading, run the compactions via an external

invocation out of a cron job or some such.</description>

</property>

2.每个region对应的每个列簇对应的MemStore在flush为hfile的时候，默认情况下，超过3个hfile的时候就会对这些文件进行合并，重写为一个文件，设置个数越大可以减少触发合并的时间，但是每次合并的时间就会越长。

<property>

<name>hbase.hstore.compactionThreshold</name>

<value>3</value>

<description>

If more than this number of HStoreFiles in any one HStore

(one HStoreFile is written per flush of memstore) then a compaction

is run to rewrite all HStoreFiles files as one. Larger numbers

put off compaction but when it runs, it takes longer to complete.</description>

</property>

## Hbase与Mapreduce集成

### 查看MR需要的Hbase 依赖包

bin/hbase mapredcp

### 修改hadoop-env.sh

添加Hbase jar 路径 , Hbase 与 MR 集成需要

export HADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:/opt/app/hbase/lib/\*

## Hbase与Hive集成

### hbase 的客户端jar 拷入hive lib 目录下

cp /apps/soft/hbase-2.1.1/lib/hbase-common-2.1.1.jar /apps/soft/apache-hive-2.2.0-bin/lib/

cp /apps/soft/hbase-2.1.1/lib/hbase-server-2.1.1.jar /apps/soft/apache-hive-2.2.0-bin/lib/

cp /apps/soft/hbase-2.1.1/lib/hbase-client-2.1.1.jar /apps/soft/apache-hive-2.2.0-bin/lib/

cp /apps/soft/hbase-2.1.1/lib/hbase-protocol-2.1.1.jar /apps/soft/apache-hive-2.2.0-bin/lib/

cp /apps/soft/hbase-2.1.1/lib/hbase-it-2.1.1.jar /apps/soft/apache-hive-2.2.0-bin/lib/

### hive-site配置ZK地址

Hive需要通过ZK找到Hbase元数据

<property>

<name>hive.zookeeper.quorum</name>

<value>hadoonode1,hadoonode2,hadoonode3</value>

<description>The list of ZooKeeper servers to talk to. This is only needed for read/write locks.</description>

</property>

<property>

<name>hive.zookeeper.client.port</name>

<value>2181</value>

<description>The port of ZooKeeper servers to talk to. This is only needed for read/write locks.</description>

</property>

### HBase与hive集成

#### 第一种是创建表管理表，指定数据存储在hbase表中

1. CREATE TABLE `hive\_hbase\_emp`(

`empno` int,

`ename` string,

`job` string,

`mgr` int,

`hiredate` string,

`sal` double,

`comm` double,

`dept` int)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,info:ename,info:job,info:mgr,info:hiredate,info:sal,info:comm,info:dept")

TBLPROPERTIES ("hbase.table.name" = "hbase\_tmp\_table");

Tips：指定Hive表字段和Hbase表里字段映射关系

1. **Hive与Hbase Apache版本的不兼容性导致，使用CDH就会避免该问题**

FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. org.apache.hadoop.hbase.HTableDescriptor.addFamily(Lorg/apache/hadoop/hbase/HColumnDescriptor;)V

解决办法：需要重新将 hive-hbase-handler-0.13.1.jar重新编译，以便兼容Hbase 1.4.12

步骤1：将hive lib下和hbase下所有jar包，导入自建lib目录

步骤2：eclipse兴建java project

步骤3：编译hive 0.13.1 hbase-handler源码包，并导出，重新放入hive/lib下替换原来的hive-hbase-handler-0.13.1.jar即可

* 1. **使用CDH 5.3.6 Hive hadoop 、hbase版本重新搭建集群，试验成功**

1. Hive中创建临时表，不能将数据直接load进Hive所关联的Hbase那张表

Hbase底层存储的是byte数据，load加载的文本不能直接放到hbase目录下不能识别

CREATE TABLE `tmp\_emp`(

`empno` int,

`ename` string,

`job` string,

`mgr` int,

`hiredate` string,

`sal` double,

`comm` double,

`dept` int)

ROW FORMAT delimited fields terminated by ' ';

加载数据到临时表

load data local inpath '/opt/data/emp.txt' into table tmp\_emp;

从临时表插入数据到Hbase中

insert into table hive\_hbase\_emp select \* from tmp\_emp;

Hive中查看表

select \* from hive\_hbase\_emp ;

Hbase中查看表

scan "hbase\_tmp\_table"

#### 创建外部表，关联 hbase表(已有数据)

1 . 创建emp表，关联Hbase hbase\_tmp\_table表

CREATE EXTERNAL TABLE `emp`(

`empno` int,

`ename` string,

`job` string,

`mgr` int,

`hiredate` string,

`sal` double,

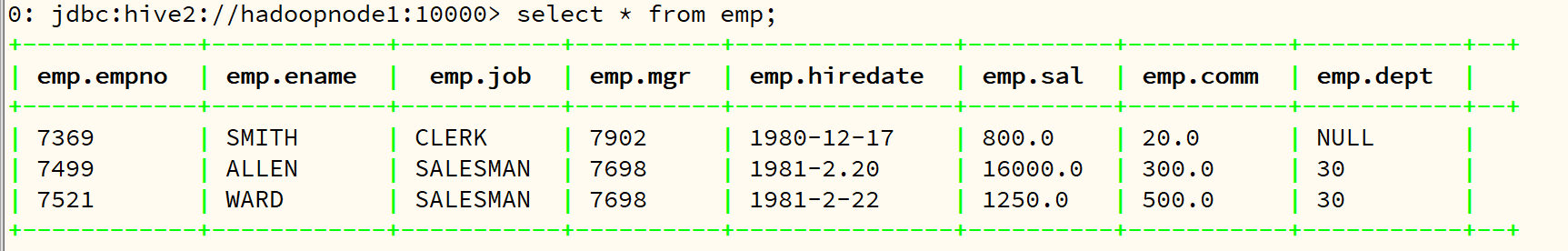
`comm` double,

`dept` int)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,info:ename,info:job,info:mgr,info:hiredate,info:sal,info:comm,info:dept")

TBLPROPERTIES ("hbase.table.name" = "hbase\_tmp\_table");



2. 创建student表，关联Hbase student表

CREATE EXTERNAL TABLE `student`(

`stuno` int,

`name` string,

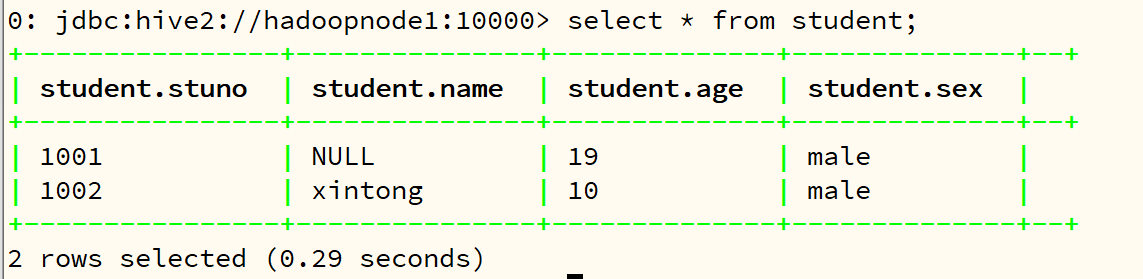
`age` int,

`sex` string)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,info:name,info:age,info:sex")

TBLPROPERTIES ("hbase.table.name" = "student");



## 预分区（数据负载均衡）

不让自动分区，自动切分策略会导致数据倾斜

1. 手动设置预分区

create "staff","info","partition1","SPLITS"=>['1001','1002','1003','1004']

2. 生成十六进制序列预分区

create 'staff1','info','partition2',{NUMREGIONS=>15,SPLITALGO=>'HexStringSplit'}

1. 按照文件中设置的规则进行分区

在Hbase目录下创建split文件

create 'staff2','info','partition1',SPLITS\_FILE=>'./split.txt'

一般一张表，在一台regionserver上最好有2-3个区

ROWKEY设计原则：唯一性、散列性、长度性

## Hbase实战微博表

