链滴

常用的 hbase 命令

作者：flowaters
原文链接：https://ld246.com/article/1501562667227
来源网站：链滴

Hbase表模型

HBase中有Table和Family和Qualifier三个概念。

Table可以直观理解为表，而Family和Qualifier直观理解为列，一个Family下面可以有多个Qualifier即可以理解为，HBase中的列是二级列，其中Family是第一级列，Qualifier是第二级列。两个是父子系。

Q: 对于传统关系型数据库中的一张table，在业务转换到hbase上建模时，从性能的角度应该如何设置amily和qualifier呢?

A: 最极端的，可以每一列都设置成一个family，也可以只有一个family，但所有列都是其中的一个quaifier，那么有什么区别呢?

family越多，那么获取每一个cell数据的优势越明显，因为io和网络都减少了，而如果只有一个family那么每一次读都会读取当前rowkey的所有数据，网络和io上会有一些损失。

当然如果要获取的是固定的几列数据，那么把这几列写到一个family中比分别设置family要更好，因只需一次请求就能拿回所有数据。

首先，不同的family是在同一个region下面。而每一个family都会分配一个memstore，所以更多的family会消耗更多的内存。

其次,目前版本的hbase，在flush和compaction都是以region为单位的，也就是说当一个family达到flsh条件时，该region的所有family所属的memstore都会flush一次，即使memstore中只有很少的数也会触发flush而生成小文件。这样就增加了compaction发生的机率，而compaction也是以region单位的，这样就很容易发生compaction风暴从而降低系统的整体吞吐量。

第三，由于hfile是以family为单位的，因此对于多个family来说，数据被分散到了更多的hfile中，减了split发生的机率。这是把双刃剑。更少的split会导致该region的体积比较大，由于balance是以regin的数目而不是大小为单位来进行的，因此可能会导致balance失效。而从好的方面来说，更少的split让系统提供更加稳定的在线服务。

上述第三点的好处对于在线应用来说是明显的，而坏处我们可以通过在请求的低谷时间进行人工的spli和balance来避免掉。

因此对于写比较多的系统，如果是离线应该，我们尽量只用一个family好了，但如果是在线应用，那是应该根据应用的情况合理地分配family。

表操作

创建表

create '表名称', '列名称1','列名称2','列名称N'

eg: create 'member','member_id','address','info ‘

列出所有的表

list

描述表

describe 'member'

删除表的一个列

1. disable 'member'
2. alter 'member', {NAME=>'member_id',METHOD=>'delete'}
3. enable 'member'

删除一个表

1. disable 'member'
2. drop 'member'

查询表是否存在

exists 'member'

判断表是否enable

is_enabled 'member'

判断表是否disable

is_disabled 'member'

记录

添加记录

put '表名称', '行名称', '列名称:', '值'

put'member','scutshuxue','info:age','24'

put'member','scutshuxue','info:birthday','1987-06-17'

put'member','scutshuxue','info:company','alibaba'

put'member','scutshuxue','address:contry','china'

put'member','scutshuxue','address:province','zhejiang'

put'member','scutshuxue','address:city','hangzhou'

put'member','xiaofeng','info:birthday','1987-4-17'

put'member','xiaofeng','info:favorite','movie'

put'member','xiaofeng','info:company','alibaba'

put'member','xiaofeng','address:contry','china'

put'member','xiaofeng','address:province','guangdong'

put'member','xiaofeng','address:city','jieyang'

put'member','xiaofeng','address:town','xianqiao'

查看记录

get '表名', '行名称'

获取一个id的所有数据

get 'member','scutshuxue'

获取一个id，一个列族的所有数据

get 'member','scutshuxue','info'

获取一个id，一个列族中一个列的所有数据

get 'member','scutshuxue','info:age'

通过timestamp来获取两个版本的数据

get 'member','scutshuxue',{COLUMN=>'info:age',TIMESTAMP=>1321586238965}

get 'member','scutshuxue',{COLUMN=>'info:age',TIMESTAMP=>1321586571843}

查看表中的记录总数

count '表名称'

count 'member'

删除记录

delete '表名' ,'行名称' , '列名称'

删除id为temp的值的‘info:age’字段

delete 'member','temp','info:age'

get 'member','temp'

增加递增字段

给‘xiaofeng’这个id增加'info:age'字段，并使用counter实现递增

incr 'member','xiaofeng','info:age'

get 'member','xiaofeng','info:age'

获取当前count的值

get_counter 'member','xiaofeng','info:age'

删除整行

deleteall 'member','xiaofeng'

删除一张表

先要屏蔽该表，才能对该表进行删除

1. 第一步 disable '表名称'
2. 第二步 drop '表名称'

清空一张表

truncate 'member'

全表扫描

scan "表名称"

scan 'member'

查看某个表某个列中所有数据

scan "表名称" , ['列名称:']

更新记录

重写一遍进行覆盖

put 'member','scutshuxue','info:age' ,'99'

get 'member','scutshuxue','info:age'

查看服务器状态

status

查看版本

version

参考：

1. HBase入门实例: Table中Family和Qualifier的关系与区别