

## 个人资料



chlaws

访问：390905次  
积分：5245  
等级：  
排名：第4578名

原创：107篇 转载：21篇  
译文：1篇 评论：233条

## 文章搜索

## 文章分类

apache (10)  
C++学习 (12)  
Linux/Unix Program (20)  
Linux/Unix shell (12)  
oracle (2)  
原创项目源码 (13)  
异步IO (3)  
性能优化 (4)  
技术分析 (47)  
数据结构和算法 (14)  
总结回顾 (7)  
编译调试 (12)  
hadoop系列 (14)  
nginx (7)  
nosql (3)  
项目经验 (24)  
redis (3)  
lucence (1)  
tigase (1)  
lua (5)  
zeromq (4)  
storm (5)  
MapReduce 1.2.1源码分析 (4)

## 文章存档

2014年08月 (1)  
2014年07月 (1)

【评论送书】我的世界、架构师、OpenStack Python 创意编程活动 CSDN日报20170510 ——《如何撰写一篇受人欢迎的博客》

## HBase -ROOT-和.META.表结构(region定位原理)

标签：[hbase](#) [-ROOT-](#) [.META.](#) [Region定位](#)

2013-11-24 13:50 7946人阅读 评论(1) 收藏 举报

分类：[hadoop系列 \(13\)](#)

在Hbase中，大部分的操作都是在RegionServer完成的，Client端想要插入，删除，查询数据都需要到RegionServer。什么叫相应的RegionServer？就是管理你要操作的那个Region的RegionServer。Client本身并不知道哪个RegionServer管理哪个Region，那么它是如何找到相应的RegionServer的？本文就是在研究揭秘这个过程。

在前面的文章“HBase存储架构”中我们已经讨论了HBase基本的存储架构。在此基础上我们引入两个特殊的概念：-ROOT-和.META.。这是什么？它们是HBase的两张内置表，从存储结构和操作方法的角度来说，它们和其他HBase的表没有任何区别，你可以认为这就是两张普通的表，对于普通表的操作对它们都适用。它们与众不同的地方是HBase用它们来存贮一个重要的系统信息——Region的分布情况以及每个Region的详细信息。

好了，既然我们前面说到-ROOT-和.META.可以被看作是两张普通的表，那么它们和其他表一样就应该有自己的表结构。没错，它们有自己的表结构，并且这两张表的表结构是相同的，在分析源码之后我将这个表结构大致的画了出来：

## -ROOT-和.META.表结构

RowKey	info			history
	regioninfo	server	serverstartrcode	
TableName, StartKey, TimeStamp	StartKey, EndKey, Family List { Family, BloomFilter, Compress, TTL, InMemory, BlockSize, BlockCache }	address		

我们来仔细分析一下这个结构，每条Row记录了一个Region的信息。

首先是RowKey，RowKey由三部分组成：TableName, StartKey 和 TimeStamp。RowKey存储的内容我们又称之为Region的Name。哦，还记得吗？我们在前面的文章中提到的，用来存放Region的文件夹的名字是RegionName的Hash值，因为RegionName可能包含某些非法字符。现在你应该知道为什么RegionName会包含非法字符了吧，因为StartKey是被允许包含任何值的。将组成RowKey的三个部分用逗号连接就构成了整个RowKey，这里TimeStamp使用十进制的数字字符串来表示的。这里有一个RowKey的例子：

## Java代码

```
01. Table1, RK10000, 12345678
```

然后是表中最主要的Family：info，info里面包含三个Column：regioninfo就是Region的详细信息，包括StartKey, EndKey 以及每个Region的RegionServer的地址。

所以当Region被拆分、合并或者重新分配的时候，都需要来修改这张表。到目前为止我们已经学习了必须的背景知识，下面我们要正式开始介绍。打算用一个假想的例子来学习这个过程，因此我先构建了假想的-ROOT-表。我们先来看.META.表，假设HBase中只有两张用户表：Table1和Table2。Region，因此在.META.表中有很多条Row用来记录这些Region。而Table1中只有两条Row用来记录。这个表的内容看上去是这个样子：

## .META.行记录结构



迷你仓



2014年05月 (1)  
2014年04月 (2)  
2014年01月 (1)

展开

#### 阅读排行

我的2012-分享我的四个  
libevent-2.0.21笔记 (18535)  
我的2011-分享我的四个  
解决客户端通过zookeeper (14014)  
优化hbase的查询操作-大 (12430)  
lua 类与继承 (10311)  
MapReduce源码分析之! (10024)  
使用hbase自带工具测试 (9291)  
短作业优先算法-SJF (8643)  
lua 编码转码url (8589)  
短作业优先算法-SJF (8522)  
lua 编码转码url (8465)

#### 评论排行

我的大学几年- (未修改) (115)  
我的2011-分享我的四个 (27)  
我的2012-分享我的四个 (13)  
先来先服务算法-FCFS (6)



\* 程序员4月书讯: Angular来了!  
\* 程序员要拥抱变化, 聊聊Android即将支持的Java 8  
\* 彻底弄懂prepack与webpack的关系  
\* 用 TensorFlow 做个聊天机器人  
\* 分布式机器学习的集群方案介绍之HPC实现  
\* Android 音频系统: 从AudioTrack 到 AudioFlinger

#### 最新评论

HBase -ROOT和.META表结构(ShawshankLin: 请问你的hbase是版本几的?)  
CMakeup解析中文乱码及不同版小小脸庞: 不同版本的cmake,来编译程序会有区别吗?  
MapReduce源码分析之InputSplCu提: 写的很好。支持一下!  
短作业优先算法-SJF mrshen007: @u013733831:同意, 我也是想到这个情况。想请教一下, 如果是这种情况发生, sjf算法该怎么处理...  
MapReduce源码分析之架构分析RayexCui: 请问一下一个reduce task 对应一个 reducer吗?  
封装nginx的异步访问redis并生

RowKey	info			historian
	regioninfo	server	server startcode	
Table1,RK0,12345678		RS1		
Table1, RK10000,12345687		RS2		
Table1, RK20000,12346578		RS3		
*** **	*** **	*** **	*** **	*** **
Table2, RK0,12345678		RS1		
Table2, RK30000,12348765		RS2		

现在假设我们要从Table2里面插寻一条RowKey是RK10000的数据。那么我们应该遵循以下步骤:

1. 从.META表里面查询哪个Region包含这条数据。
2. 获取管理这个Region的RegionServer地址。
3. 连接这个RegionServer, 查到这条数据。

好,我们先来第一步。问题是.META也是一张普通的表,我们需要先知道哪个RegionServer管理了.META表,怎么办?有一个方法,我们把管理.META表的RegionServer的地址放到ZooKeeper上面不久行了,这样大家都知道了谁在管理.META。

貌似问题解决了,但对于这个例子我们遇到了一个新问题。因为Table1实在太大了,它的Region实在太多了,.META.为了存储这些Region信息,花费了大量的空间,自己也需要划分成多个Region。这就意味着可能有多多个RegionServer在管理.META.。怎么办?在ZooKeeper里面存储所有管理.META的RegionServer地址让Client自己去遍历?HBase并不是这么做的。

HBase的做法是用另外一个表来记录.META的Region信息,就和.META.记录用户表的Region信息一模一样。这个表就是-ROOT-表。这也解释了为什么-ROOT-和.META.拥有相同的表结构,因为他们的原理是一模一样的。

假设.META.表被分成了两个Region,那么-ROOT-的内容看上去大概是这个样子的:

#### -ROOT-行记录结构

RowKey	info			historian
	regioninfo	server	server startcode	
.META.,Table1,0,12345678,12657843		RS1		
.META.,Table2,30000,12348765,12438675		RS2		

这么一来Client端就需要先去访问-ROOT-表。所以需要知道管理-ROOT-表的RegionServer的地址。这个地址被存在ZooKeeper中。默认的路径是:

#### Java代码

```
01. /hbase/root-region-server
```

等等,如果-ROOT-表太大了,要被分成多个Region怎么办?嘿嘿,HBase此-ROOT-只会有一个Region,这个Region的信息也是被存在HBase内部。现在让我们从头来过,我们要查询Table2中RowKey是RK10000的数据。org.apache.Hadoop.hbase.client.HConnectionManager.TableServers

#### Java代码

```
01. private HRegionLocation locateRegion(final byte[] tab
02.     final byte[] row, boolean useCache) throws IO
03.     if (tableName == null || tableName.length == 0) {
04.         throw new IllegalArgumentException("table nam
05.     }
06.     if (Bytes.equals(tableName, ROOT_TABLE_NAME)) {
```



迷你仓



关闭

\_Obito: 大神 hiredis在nginx怎么编译过呀 求救

封装nginx的异步访问redis并生5  
小猴饲养员: @nishige:请问,我也想在nginx中封装异步操作,能否请教一下?

优化hbase的查询操作-大幅提升  
小小苹果323: 请问您的查询速度是多快啊?

使用hbase自带工具测试读写速2  
tan\_\_5: 群主你好,请问这个跑完,报告在哪里看的

短作业优先算法-SJF  
无畏先锋查找虎王像只猫: 兄弟,你的算法其实是错的。只是对于你的这组数据有用而已。因为你在void SJF(int len) ...

#### 外部资料链接

啃饼的博客 (RSS)

陈硕的Blog (RSS)

#### 我的空间

我的网摘

#### 英语学习

语法学习



```
07.         synchronized (rootRegionLock) {
08.             // This block guards against two threads trying to find the root
09.             // region at the same time. One will go do the find while the
10.             // second waits. The second thread will not do find.
11.             if (!useCache || rootRegionLocation == null) {
12.                 this.rootRegionLocation = locateRootRegion();
13.             }
14.             return this.rootRegionLocation;
15.         }
16.     } else if (Bytes.equals(tableName, META_TABLE_NAME)) {
17.         return locateRegionInMeta(ROOT_TABLE_NAME, tableName, row, useCache, metaRegi
onLock);
18.     } else {
19.         // Region not in the cache - have to go to the meta RS
20.         return locateRegionInMeta(META_TABLE_NAME, tableName, row, useCache, userRegi
onLock);
21.     }
22. }
```

这是一个递归调用的过程：

#### Java代码

```
01.  获取Table2, RowKey为RK10000的RegionServer => 获取.META., RowKey为Table2, RK10000
99999999的RegionServer => 获取 -ROOT-, RowKey为.META., Table2, RK10000, 9999999999
999999999的RegionServer => 获取 -ROOT- 的RegionServer => 从ZooKeeper得到 -ROOT- 的R
er => 从 -ROOT- 表中查到RowKey最接近（小于） .META., Table2, RK10000, 9999999999999,
99999 的一条Row, 并得到.META. 的RegionServer => 从.META. 表中查到RowKey最接近（小于） Table2, RK1
0000, 999999999999999 的一条Row, 并得到Table2的RegionServer => 从Table2中查到RK10000的Row
```

到此为止Client完成了路由RegionServer的整个过程，在整个过程中使用了添加“999999999999999”后缀并查找最接近（小于）RowKey的方法。对于这个方法大家可以仔细揣摩一下，并不是很难理解。

最后提醒大家注意两件事情：

1. 在整个路由过程中并没有涉及到MasterServer，也就是说HBase日常的数据操作并不需要MasterServer，不会造成MasterServer的负担。
2. Client端并不会每次数据操作都做这个路由过程，很多数据都会被Cache起来。至于如何Cache，则不在本文的讨论范围之内。

原

顶 5 踩 0

上一篇 [storm-0.8.2源码分析之nimbus运行过程\(一\)](#)

下一篇 [迟到的2013年总结](#)

#### 我的同类文章

### hadoop系列 ( 13 )

• MapReduce源码分析之Ma... 2014-08-04 阅读 3734	• MapReduce源码分析之Ma... 2014-07-13 阅读 3525
• MapReduce源码分析之架构... 2014-04-14 阅读 2921	• MapReduce源码分析之架构... 2014-04-14 阅读 2921
• 简述thrift与应用分析 2013-07-14 阅读 6500	• hadoop几个版本对比 2013-07-14 阅读 6500
• hive部署 2012-06-30 阅读 2715	• 解决hadoop批处理性能问题 2012-06-30 阅读 2715
• 我的2011-分享我的四个项目... 2012-01-01 阅读 12429	• 优化hbase的查询操作 2012-01-01 阅读 10017

[更多文章](#)

### 迷你仓

迷你仓

迷你仓



世界十大名表排名

二手车之家

世界名表排行榜

数据分析工具

渗透测试

外汇公司

## 参考知识库



### Go知识库

2258 关注 | 979 收录



### Hbase知识库

8258 关注 | 87 收录



### Hadoop知识库

7174 关注 | 574 收录



### Java SE知识库

26067 关注 | 578 收录



### Java EE知识库

18117 关注 | 1334 收录



### Java 知识库

26464 关注 | 1476 收录



### 大型网站架构知识库

8691 关注 | 708 收录

## 猜你在找

分布式资源管理系统的前世今生，深入剖析YARN资源调 Android 操作系统 获取Root权限 原理解析

Hadoop大数据从入门到精通（行业最强，备javaee） Android的Root原理

企业级大数据架构指南（hadoop\kafka\spark\presto\ Android Root方法原理解析及Hook四 GingerBreak

搜狗郭理勇：小而美-Sogou数据库中间件Compass深度剖析 Android Root方法原理解析及Hook四 GingerBreak

阿里曾文旌：Greenplum和Hadoop对比，架构解析及技 HBase行锁原理及实现



## 查看评论

1楼 ShawshankLin 2016-09-10 11:30发表



请问你的hbase是版本几的？

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

## 核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack  
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery  
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity  
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack  
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo  
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr  
Angular Cloud Foundry Redis Scala Django Bootstrap

关闭

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



## 迷你仓

