

# Hive over Hbase 框架查询性能研究

叶奇明 陈俊宇

(广东石油化工学院 理学院 广东 茂名 525000)

摘要:通过 Hive 与 Hbase 两者本身对外的 API 接口互相进行通信,搭建了一个 Hive over Hbase 的整合框架方案。实验证明,基于“Hbase 存储入口 + Hive 查询入口”的 Hive over Hbase 框架比 Hive 或 Hbase 原生访问具有更高效的数据库查询速度,且随着数据量的增加,查询性能更加明显。

关键词:Hive; Hbase; Hive over Hbase; 查询性能

中图分类号:TP391.1

文献标识码:A

文章编号:2095-2562(2018)06-0036-03

Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具,可以进行数据提取转化加载(ETL),是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 查询操作过程严格遵守 Hadoop MapReduce 的作业执行模型,Hive 将用户的 HiveQL 语句通过解释器转换为 MapReduce 作业提交到 Hadoop 集群上,实现查询操作。Hbase 则是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统,利用 Hadoop HDFS 作为其文件存储系统;利用 Hadoop MapReduce 来处理 HBase 中的海量数据<sup>[1]</sup>。面对日益庞大的海量数据,大多数学者对 Hive 和 Hbase 数据查询效率更多侧重于 HiveQL 语句的优化研究、map 和 reduce 数量设置和 jvm 重用<sup>[2]</sup>;有的学者也通过对 Hbase 和其他操作组件 lealone、phoenix 整合来提升查询性能。比起孤立的两个方案,Hbase 和 Hive 整合的方案有更低的 cpu 负载、磁盘的使用率也一直处于低负载状态<sup>[3]</sup>,且稳定性更好。这种“Hbase 存储入口 + Hive 查询入口”的 Hive over Hbase 整合方案比采用 Hive 提供的基于 MapReduce 计算框架的 HiveQL 查询语句,能更好地提高数据抽取、转化和加载的效率<sup>[4-6]</sup>。

## 1 Hive 和 Hbase 的相关理论

### 1.1 Hbase 查询流程

Hbase 使用 MemStore 和 StoreFile 存储对表进行更新。数据在更新时,先写入 HLog 和 MemStore。MemStore 中的数据是排序的,当 MemStore 累计到一定阈值时,就会创建一个新的 MemStore,并且将老的 MemStore 添加到 Flush 队列,由单独的线程 Flush 到磁盘上,成为一个 StoreFile。同时系统会在 Zookeeper 中记录一个 CheckPoint,表示这个时刻之前的数据变更已经持久化了。当系统出现意外导致 MemStore 中的数据丢失时,使用 HLog 来恢复 CheckPoint 之后的数据。StoreFile 是只读的,因此,Hbase 的更新其实是不断追加的操作。当一个 Store 中的 StoreFile 达到一定阈值后,就会进行一次合并操作,将对同一个 key 的修改合并到一起,形成一个大的 StoreFile。当 StoreFile 的大小达到一定阈值后,又会对 StoreFile 进行切分操作,等分为两个 StoreFile。在 Hbase 查询过程中,通常通过使用 bloomfilter 和 mapfile\_index\_interval、增大内存、修改 java 虚拟机属性、增大 RPC 数量等方式来提高查询速度。

Hbase 查询操作流程如下:(1) Client 访问 Zookeeper,查找 -ROOT- 表,获取 .META. 表信息。(2) 从 .META. 表查找,获取存放目标数据的 Region 信息,从而找到对应的 RegionServer。(3) 通过 Region-

收稿日期:2018-05-30;修回日期:2018-10-16

作者简介:叶奇明(1977—),男,湖北大冶人,硕士,副教授,主要研究方向为传感器网络、数据挖掘。

Server 获取需要查找的数据。(4) RegionServer 的内存分为 MemStore 和 BlockCache 两部分,MemStore 主要用于写数据,BlockCache 主要用于读数据。读请求先到 MemStore 中查数据,查不到就到 BlockCache 中查,再查不到就会到 StoreFile 上读,并把读的结果放入 BlockCache。

### 1.2 Hive 查询流程

Hive 查询过程中,通常用 Map join、优化设计模型、避免数据倾斜、减少 job 数、设置合理的 map reduce 的 task 数、小文件合并等方式来提高其查询效率。

Hive 查询步骤说明如下:(1) UI 调用 executeQuery 接口,发送 HQL 查询语句给 Driver。(2) Driver 为查询创建会话句柄,并将查询发送给 compiler 以生成执行计划。(3) compiler 从 metastore 获取相关的元数据定义。(4) 元数据用于对查询树中的表达式进行类型检查,以及基于查询谓词调整分区,生成计划。(5) 由 compiler 生成的执行计划是阶段的 DAG,每个阶段都会涉及到 Map/Reduce job,元数据的操作或者 HDFS 文件的操作。在 Map/Reduce 阶段,执行计划包含 Map 操作树(在 Mappers 上执行)和 Reduce 操作树(在 Reducers 上执行),Execution Engine 将各个阶段提交个适当的组件执行。(6) 在每个任务( mapper/reducer)中,表或者中间输出相关的反序列化器从 HDFS 读取行,并通过相关的操作树进行传递。一旦这些输出产生,将通过序列化器生成临时的 HDFS 文件(这个只发生在只有 Map 没有 reduce 的情况),生成的 HDFS 临时文件在后续的 Map/Reduce 阶段执行。对于 DML 操作,临时文件最终移动到表的位置。对于查询,临时文件的内容由 Execution Engine 直接从 HDFS 读取,作为 Driver Fetch API 的一部分。

### 1.3 Hive over Hbase 整合

Hive 与 Hbase 利用两者本身对外的 API 来实现整合,主要是靠 HBaseStorageHandler 进行通信。利用 HBaseStorageHandler,Hive 可以获取到 Hive 表对应的 Hbase 表名、列簇和列、InputFormat 和 OutputFormat 类,以及创建和删除 Hbase 表等。

Hive 访问 Hbase 中表数据,实质上是通过 MapReduce 读取 Hbase 表数据<sup>[7]</sup>。其实现是在 MR 中使用 HiveHBaseTableInputFormat 来完成对 Hbase 表的切分,获取 RecordReader 对象来读取数据。对 Hbase 表的切分原则是一个 Region 切分成一个 Split,即表中有多少个 Regions,MR 中就有多个 Map。读取 Hbase 表数据是通过构建 Scanner,对表进行全表扫描。如果有过滤条件,则转化为 Filter,当过滤条件为 rowkey 时,则转化为对 rowkey 的过滤;Scanner 通过 RPC 调用 RegionServer 的 next() 来获取数据(见图1)。两者整合成功后,Hive 和 Hbase 中记录会同步增加。

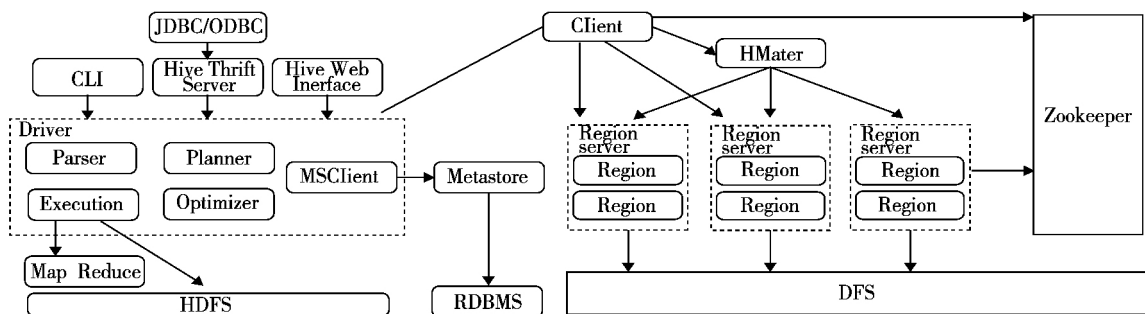


图1 Hive over Hbase 整合框架结构

## 2 实验环境和过程

硬件环境采用联想天逸 510S 商用台式办公电脑标准配置,软件环境配置如下:hadoop-2.5.2.tar.gz (6 nodes 8G mem/node,Centos7);hbase-0.98.9-hadoop2-bin.tar.gz;hive-0.13.1-bin.tar.gz。

以搜狗搜索日志分析系统部分时间段的数据为实验样本,根据实验的要求分别截取 1000,3000,7000,12000 行数据(head-行数 sogou.utf8 > sogou.demo),分别迁移入 Hive 和 Hbase,主要步骤有:创建外部表、创建扩展 4 个字段(年、月、日、小时)数据的外部表、创建带分区的表、灌入数据到 Hive。

由于 Hive over Hbase 插入数据时采用了 MapReduce 的策略算法(见图2),故插入数据到 Hive 的同时

也向 Hbase 写入。测试灌入到 Hive 中的数据是否也存在于 Hbase 中,如果 Hbase 中找到对应数据,则 Hive over Hbase 框架搭建成功,首先分别对 Hive 和 Hbase 数据原生访问,其次在 Hive over Hbase 框架基础上通过 Hive 查询入口(见图 3)测试查询性能,并对比三种测试方式的查询效率。

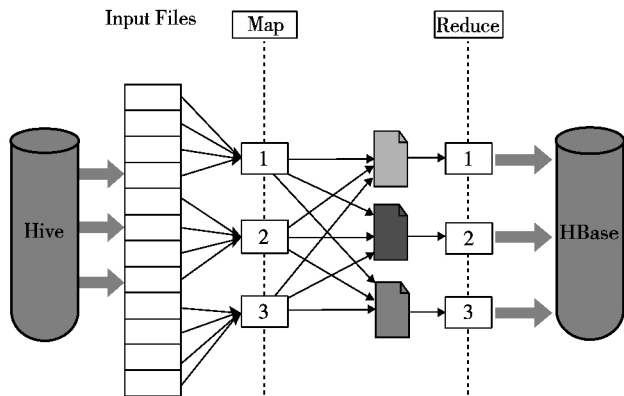


图 2 Hive 和 Hbase 策略算法

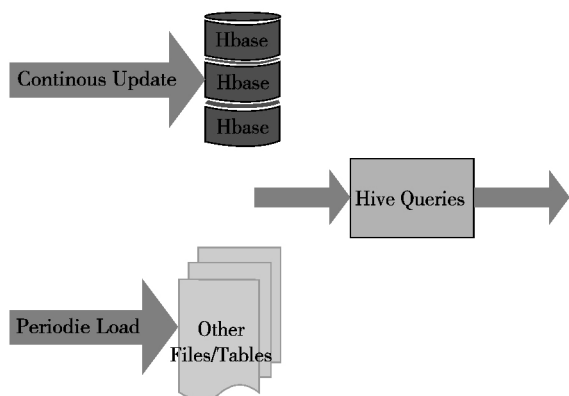


图 3 Hive over Hbase 数据访问

### 3 实验结果

对 1000 w 条样本数据设置三种查询条件 query1(根据关键词查询统计条数)、query2(根据 key 过滤统计相关记录)、query3(根据 value 过滤统计相关记录),分别对 hive on hdfs、hbase on hdfs 和 Hive over Hbase 这三种方案进行查询测试,其结果见图 4。由图 4 中的测试结果可知,三种常规查询条件下,直接通过 hive on hdfs 查询的效率优于 hbase on hdfs 查询,但基于“Hbase 存储入口 + Hive 查询入口”的 Hive over Hbase 整合方案查询效率最高。

截取不同行数样本数据,根据 query3 方式查询,其测试结果见图 5。

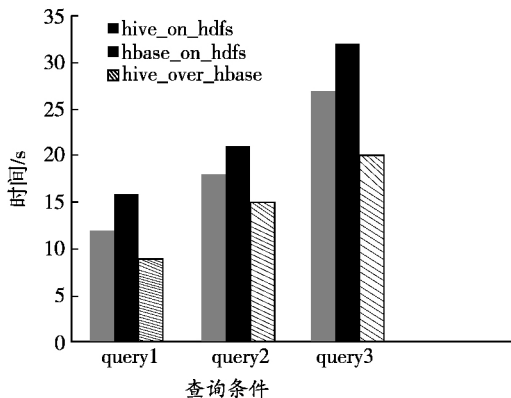


图 4 三种查询环境下查询结果

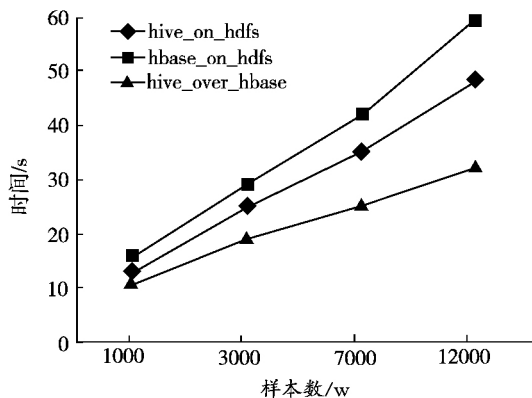


图 5 三种查询环境下不同数据量查询结果

由图 5 中的测试结果可知,基于“Hbase 存储入口 + Hive 查询入口”的 Hive over Hbase 整合方案查询时间曲线最平缓,说明该方案对数据量敏感度最低。此外,Hive over Hbase 整合方案查询随着数据量增加,数据抽取、转化和加载的效率更高,查询效率也更高,用户体验最好。

### 4 结语

目前对 Hive 和 Hbase 数据查询效率的研究更多侧重于 HiveQL 语句的优化研究、map 和 reduce 数量设置和 jvm 重用,也有研究 Hbase 和其他操作组件 lealone、phoenix 整合来提升查询性能。本文搭建了“Hbase 存储入口 + Hive 查询入口”的 Hive over Hbase 整合方案,实验证明该方案既能显著提高数据抽取、转化和加载的效率,又节省系统资源开销,且稳定性较好。由于实验结论来自离线数据,对于大规模实时数据和基于行的数据,整合方案的稳定性和查询效率的优化,还有待进一步研究。

(下转第 42 页)

- [6] 黄学锋, 杨世刚. 深层低渗凝析气藏试井解释方法及其应用探讨[J]. 石油与天然气地质, 2004(3): 333-337.
- [7] 于金珠, 唐亚会. 复杂低渗气藏气井试井评价方法探讨[J]. 大庆石油地质与开发, 2008(4): 82-85.
- [8] 高树生, 熊伟, 刘先贵, 等. 低渗透砂岩气藏气体渗流机理实验研究现状及新认识[J]. 天然气工业, 2010(1): 52-55.

## Application of Wellhead Monitoring Pressure Analysis to Low Permeability Gas Well of M Gas Field

HE Wei<sup>1</sup>, WANG Wenjuan<sup>1</sup>, LI Shusong<sup>1</sup>, HAN Xin<sup>1</sup>, JIAO Qingqiong<sup>2</sup>

(1. Zhanjiang Branch of CNOOC Ltd., Zhanjiang 524057, China;

2. CNOOC Energy Tech - Drilling & Production Co., Zhanjiang 524057, China)

**Abstract:** At present, there is no effective and real-time monitoring method of determining the quality of the closed well pressure buildup testing data. Therefore, wellhead monitoring pressure analysis technique is proposed to meet this challenge by establishing real-time diagnosis curve of monitored wellhead pressure during shut-in pressure buildup, providing one method to judge the quality of shut-in pressure buildup test data and determine the shut-in time range. Wellhead monitoring pressure analysis technique mainly consists of well test design, wellhead monitoring pressure analysis and well test interpretation, which has been successfully applied in low permeability gas well M9 of M gas field, reaching reasonable expectations of accurate shut-in time range, reducing testing time and operating cost, and minimizing the influence to gas supply.

**Key words:** Low permeability gas well; Pressure buildup test; Well test interpretation; Wellhead monitoring pressure

(责任编辑: 黄容)

(上接第 38 页)

### [参考文献]

- [1] 叶文宸. 基于 hive 的性能优化方法的研究与实践[D]. 南京: 南京大学, 2011.
- [2] AZEZ H S A, KHAFAFY M H, OMARA F A. Optimizing Join in HIVE Star Schema Using Key/Facts Indexing[J]. IETE Technical Review, 2017(2): 132-144.
- [3] 马晓亮, 田丰. Hadoop 架构下原生 Hbase、Hive、Lealone、Phoenix 等各操作组件的比较[J]. 广东通信技术, 2017(3): 71-73.
- [4] 戴园园. 面向教育大数据的多维数据分析系统的研究与实现[D]. 长沙: 湖南大学, 2016.
- [5] BARBIERATO E, GRIBAUDO M, IACONO M. Performance evaluation of NoSQL big-data applications using multi-formalism models[J]. Future Generation Computer Systems, 2014, 37: 345-353.
- [6] WANG M, XING L, SUN L. MapReduce Based Heuristic Multi-Join Optimization under Hybrid Storage[J]. Journal of Frontiers of Computer Science and Technology, 2014(11): 1334-1344.
- [7] ZHAO Z C, JIANG Y. An efficient Join-Engine to the SQL query based on Hive with Hbase[J]. International Conference on Applied Science and Engineering, 2015(200): 991-996.

## Research on Query Performance of Hive over Hbase Framework

YE Qiming, CHEN Junyu

(College of Science, Guangdong University of Petrochemical Technology, Maoming 525000, China)

**Abstract:** By communicating each other with the external API interface of Hive and Hbase to build an integrated framework scheme for Hive over Hbase, the experiment proves that the Hive over Hbase framework based on the "Hbase storage entrance + hive query entry" is more efficient than the HDFS based or direct access database query. As the amount of data increases, the query performance is more obvious.

**Key words:** Hive; Hbase; Hive over Hbase; Query performance

(责任编辑: 黄容)