

文章编号: 2095-2163(2022)02-0054-05

中图分类号: TP393.0

文献标志码: A

基于张衡一号卫星 HBase 数据库的入库方法研究

杨旭明, 李 忠, 李锦文, 贾 娟, 杨百一, 张富志

(防灾科技学院 应急管理学院, 河北 三河 065201)

摘要: 海量卫星监测数据不断增大,为提高 HBase 数据库的入库效率,本文以张衡一号卫星空间电场探测仪 ULF 频段数据为例,进行基于 HBase 数据库的入库方法研究,将 HBase 提供的多种入库方法进行对比实验,提出了将 HBase API 方法和 MapReduce Bulk Load 方法结合的方案,来满足不同数据量的入库需求,为张衡一号卫星的海量数据入库提供了技术支撑,为后续基于 HBase 的大数据存储方案的设计奠定了基础。

关键词: 卫星; 海量数据; HBase; 入库方法; Bulk Load

Research on HBase database entry method based on Zhangheng-1 satellite

YANG Xuming, LI Zhong, LI Jinwen, JIA Juan, YANG Baiyi, ZHANG Fuzhi

(Institute of Disaster Prevention, Institute of Emergency Management, Sanhe Hebei 065201, China)

[Abstract] With the increasing of massive satellite monitoring data, in order to improve the efficiency of HBase database entry, this paper takes the ULF frequency band data of Zhang Heng-1 satellite space electric field detector as an example to study the entry method based on HBase database, and carries out comparative experimental research on various entry methods provided by HBase. A scheme combining HBase API method and MapReduce Bulk Load method is proposed to meet the storage requirements of different amounts of data, which provides technical support for the massive data storage of Zhang Heng-1 satellite and lays a foundation for the subsequent design of HBase-based big data storage schemes.

[Key words] satellite; massive data; HBase; storage method; bulk load

0 引言

张衡一号卫星从 2018 年在轨至今,其搭载的载荷观测数据量已达几百 TB^[1],各个载荷观测数据会经过一系列的处理、流程以及相应的算法进行存储和科学研究,会涉及针对海量数据的存储、访问以及入库方面的不同需求。

HBase 是一个分布式的、面向列的开源数据库^[2],可以随机的、实时的访问大数据。HBase 具有高性能、高可靠性、列存储、可伸缩、实时读写等特性^[3],可以满足张衡一号卫星海量数据的高效存储与读取需求,但是怎样有效的将卫星海量数据导入到 HBase 数据库是当前迫切需要解决的问题。HBase 有多种导入数据的方法,最直接的方法就是调用 HBase 的 API,用 Put 方法插入数据;另外一种是通过 MapReduce 编程模型的 Map 函数从分布式

文件系统(HDFS)中加载数据,再通过其 Reduce 函数直接生成 Put 对象后写入 HBase。但是这两种方法都是使用了较为传统的 Put 方法来存储数据,十分耗时,同时也没有合理利用软硬件,会让 HBase 频繁地进行刷写(Flush)、合并(Compact)、拆分(Split)等操作,因此需要消耗较大的 CPU 和网络资源,会导致 HBase 集群中运行 HRegionServer 服务从节点服务器(Region Server)的压力非常大。而 Bulk Load 方式则无需进行刷写、合并、拆分等过程,不占用 HBase 的分片存储单元(Region)资源,也不会产生巨量的写入操作,只需要较少的 CPU 和网络资源,就能极大的提高写入效率,并降低对 Region Server 节点的写入压力。张衡一号卫星海量空间电场数据入库时,Bulk Load 则为其提供了一种全新有效的方法。

基金项目: 张衡一号卫星电场数据的大数据平台研发(2021057)。

作者简介: 杨旭明(1990-),男,硕士研究生,主要研究方向:大数据技术、云计算应用技术、数据分析;李 忠(1966-),男,博士,教授,主要研究方向:人工智能、大数据技术、灾害信息处理;李锦文(1997-),女,硕士研究生,主要研究方向:灾害信息处理;贾 娟(1995-),女,硕士研究生,主要研究方向:灾害信息处理;杨百一(1997-),男,硕士研究生,主要研究方向:数据挖掘;张富志(1995-),男,硕士研究生,主要研究方向:灾害信息处理。

通讯作者: 李 忠 Email: lizhong@cidp.edu.cn

收稿日期: 2021-10-26

1 HBase 数据库入库方法介绍

1.1 HBase 物理存储模型

HBase 物理存储模型,如图 1 所示。HBase 中主节点(Master)上运行 Hmaster 服务,其作用是维护整个集群的负载均衡、为 Region Server 节点分配 Region、维护集群的元数据信息等;Region Server 节点上运行 HRegionServer 程序,其作用是管理一系列的 Region、处理来自客户端的读写请求等。每个 Region 又由多个 HBase 存储基本单元(Store)构成,每个 Store 与 HBase 的列族是一一对应的;Store 由内存存储(MemStore)和存储文件(StoreFile)组成,MemStore 是写缓存,当其大小达到特定阈值后就会触发刷写操作生成存储文件(StoreFile),每个 StoreFile 会以 HBase 的物理文件(HFile)格式存储在 HDFS 中。

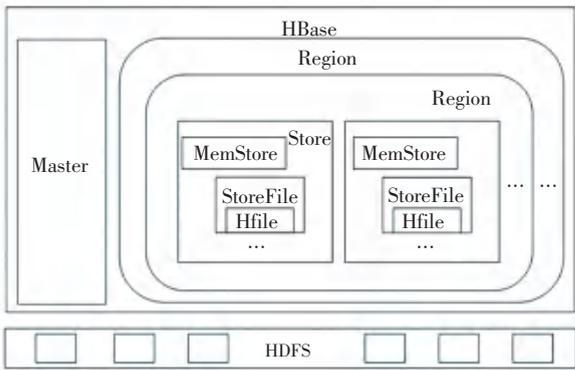


图 1 HBase 物理存储模型

Fig. 1 HBase physical storage model

1.2 HBase API 方法

客户端的插入数据操作其实就是远程过程调用协议(RPC)请求,当数据到达 Region Server 时,会被默认先写入到预写日志(Write Ahead Log,以下简称 WAL)即 HLog 中;再将数据写入到相应 Region 的 MemStore 中,当达到 MemStore 的特定阈值之后,触发刷写(Flush)操作,将数据刷新到 StoreFile,StoreFile 则以 HFile 格式存到 HDFS 中,此时的 Flush 操作引起的写操作会瞬间堵塞用户。当 StoreFile 数量达到特定的阈值时,会触发 Compact 操作,多个 StoreFile 会被合并为一个大的 StoreFile,该过程包含了大量的网络和硬盘 I/O。当 StoreFile 过大并达到相应阈值后又会被 Split 操作拆分。HBase 客户端 API 写入数据的原理图,如图 2 所示。

从通过 HBase API 方法写入数据的流程可知,客户端在 Put 数据时,需要频繁的与存储数据的

Region Server 通信,当一次写入大量数据时,就会占用 Region Server 的大量资源,就会大大影响对该 Region Server 上存储表的操作。因此,当大批量数据写入时,会导致效率极其低下,从而影响 HBase 集群节点的稳定性。

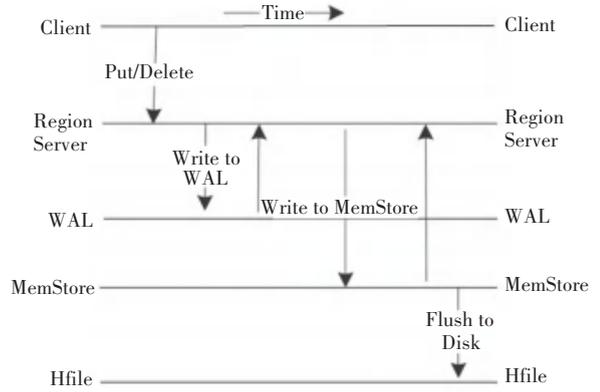


图 2 HBase API 写入数据原理图

Fig. 2 HBase API data writing schematic diagram

1.3 Bulk Load 方法

通过 HBase 物理存储模型的原理可知,HBase 数据是以 HFile 文件结构在 HDFS 中存储的。因此,可以先将数据生成 HFile 结构文件,再把生成的 HFile 文件加载到 HBase 中。Bulk Load 批量加载方法的原理就是通过使用 MapReduce 作业以 HBase 内部数据格式输出表数据,直接将生成的 StoreFiles 加载到正在运行的集群中,从而快速完成海量数据的入库,该过程绕过了 Put 操作中写数据的路径包括 WAL、MemStore、Flush 等,不会占用 Region 资源,也不会产生大量的写入 I/O,因此使用 Bulk Load 比通过 HBase API 加载占用更少的 CPU 和网络资源。

利用 Bulk Load 进行 HBase 批量数据加载的过程主要分为两个步骤:使用 MapReduce Job 准备数据和完成数据加载。Bulk Load 方法加载数据过程如图 3 所示。

1.3.1 使用 MapReduce 生成数据

Bulk Load 批量加载的第一步是使用官方提供的 ImportTSV 工具将 HDFS 中 TSV 格式数据转化为 HBase 数据文件 HFile,或者使用相应的类编写 MapReduce 作业程序生成 HFile。这种输出格式以 HBase 的内部存储格式写出数据,以便后续可以高效地将数据加载到集群中。

ImportTSV 工具适用于很多实际情况,但是需要手动指定所有的列名,在面对需要导入的数据中存在很多列名的情况时,该方法显得尤为笨重。因此

很多情况下高需求用户会通过编写 MapReduce 作业生成数据。

1.3.2 完成数据加载

将 MapReduce 生成的目标数据文件加载到 HBase 中。此工具通过遍历准备好的 HFile 数据文

件,确定每个文件所属的 Region,通过相应的 Region Server 将 HFile 数据文件加载到存储目录。如果在整个过程中的某个阶段 Region 边界发生改变,则将 HFile 数据文件自动拆分为和 Region 新边界对应的部分。

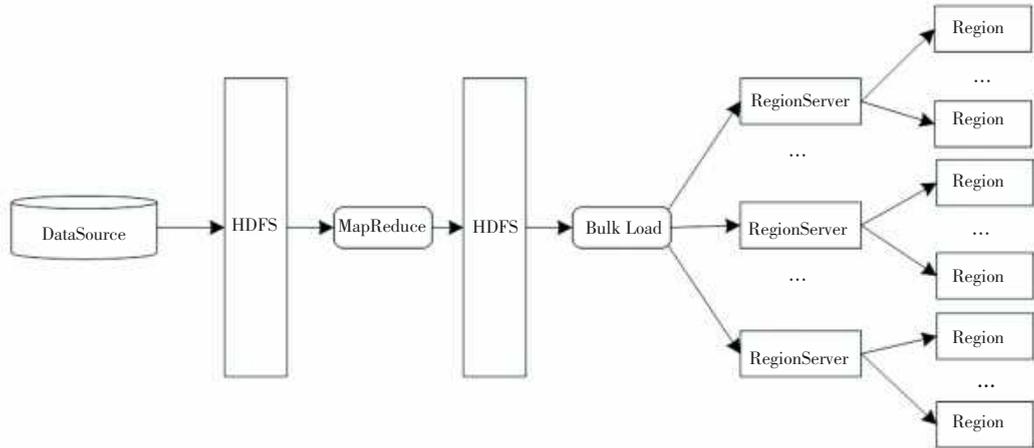


图3 Bulk Load 方法加载数据过程

Fig. 3 The process of data loading of Bulk Load method

2 HBase API 和 Bulk Load 方法对比实验

本文以张衡一号监测卫星空间电场数据中的 ULF 频段数据作为数据来源。张衡一号卫星空间电场数据现有的存储方式为 H5 格式文件,每个文件包括波形数据和功率谱数据,单个文件数据量大,严重制约了数据的存储、访问、查询等效率,为此选择 HBase 数据库作为存储数据库。由于数据涉及卫星编号、轨道号、工作模式、时间、地理经度、地理纬度、地磁经度、地磁高度、频率等等很多的字段,因此选择使用 HBase API 方法和编写 MapReduce 作业的 Bulk Load 方法进行入库实验对比。

2.1 实验环境

为对比两种 HBase 入库方法在实际生产环境中的效果,本文使用 Docker 容器技术搭建了由 5 台服务器节点组成 Hadoop 高可用集群,每个节点的硬件配置为 4 核 CPU、16 G 内存和 300 G 硬盘。每个节点安装 Centos7 操作系统和 JDK1.8 环境,并按照表 1 中的角色分配情况在相应节点上安装 Hadoop3.2.2、ZooKeeper3.6.2 和 HBase2.2.7 软件。测试数据选用张衡一号卫星 2018 年 8 月 1 日中的部分空间电场 ULF 频段数据。开发语言选择 Java 语言和 Python 语言。

表 1 Hadoop 集群角色分配

Tab. 1 Hadoop Cluster role assignment

节点名称角色	节点 H1	节点 H2	节点 H3	节点 H4	节点 H5
ZooKeeper	是	是	否	否	否
NameNode	是	是	否	否	否
DataNode	否	否	是	是	是
JournalNode	否	否	是	是	否
ZKFC	是	是	否	否	否
ResourceManager	是	是	否	否	否
NodeManager	否	否	是	是	是
Hmaster	是	是	否	否	否
HRegionServer	否	否	是	是	是
JobHistory	是	否	否	否	否

表 1 中 ZooKeeper 是分布式应用程序协调服务,通过其简单的架构,在分布式环境中协调和管理服务,是 Hadoop 和 HBase 的重要组件;NameNode 是 HDFS 的名称节点,主要用来保存 HDFS 的元数据信息,维护着文件系统树及整棵树内所有的文件和目录以及接收用户的操作请求;DataNode 是 HDFS 的数据节点,提供真实文件数据的存储服务;ZKFC 是 Zoo Keeper Failover Controller 的简写,是 ZooKeeper 中一个新的组件,监视和管理 NameNode 的状态;ResourceManager 是资源管理程序,负责集群中所有

资源的统一管理和分配,接收来自各个节点的资源汇报信息,并把这些信息按照一定的策略分配给各个应用程序;NodeManager 是节点管理程序,用来管理 Hadoop 集群中单个计算节点;JobHistory 是历史服务器程序,可以通过其查看已经运行完的 MapReduce 作业记录。

2.2 实验步骤

张衡一号卫星空间电场数据目前的存储格式为 H5 格式文件,因此本文先将数据从 H5 文件中提取出来,进行转换生成制表格式的 txt 文件,为后续的数据入库操作做准备。具体对比实验流程如图 4 所示。

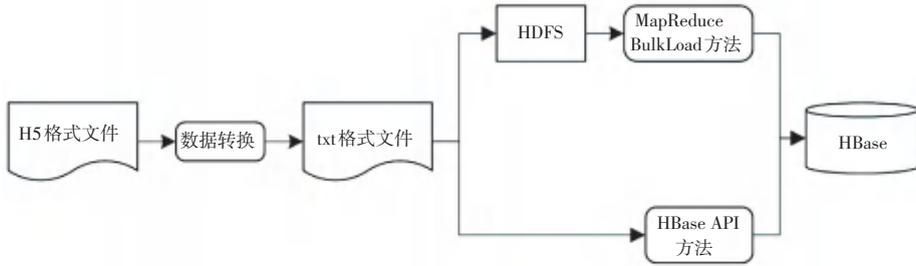


图 4 对比实验流程

Fig. 4 Flow chart of comparative experiment

(1)数据转换步骤。本实验通过 Python 语言进行编程。首先,反转 H5 文件中的时间字段数据作为 HBase 的行键前缀;再通过读取 H5 文件名称和内容中的卫星编号、载荷编码、载荷序号、数据分级编码、观测对象编码、接收站编码、版本号按顺序组合成完整的行键;最后,读取 H5 文件中的字段和数据,组合为字段:数据的样式,并把行键、字段,数据按照制表符分隔的形式写入到 txt 文件中,实现从 H5 格式文件到 txt 文件的数据转换。

数据需要一次性写入 HBase 数据库时,MapReduce Bulk Load 方式的优势是非常明显的。

(2)HBase API 入库实验。使用 Python 语言编写通过 HBase API 的入库程序,该方法比较简单,只需要通过连接 HBase 数据库,读取 txt 文件中的相应字段后存入 HBase 数据库的表中。



图 5 HBase 入库方法对比图

Fig. 5 Comparison diagram of HBase database entry methods

(3)MapReduce Bulk Load 方法入库实验。先将 txt 文件上传到 HDFS 中的相应 HBase 目录下,再使用 Java 开放语言编写自定义的 MapReduce 程序对数据进行读入与拆分操作,生成 HFile 文件,最后把生成的 HFile 文件数据加载到 HBase 中。

3 结束语

本文研究分析 HBase 数据库的入库方法、原理,使用张衡一号卫星空间电场 ULF 频段数据作为数据源,利用 HBase API 和 MapReduce Bulk Load 两种方法进行了不同数据量的入库实验。实验结果表明,当面对少量卫星数据入库需求时,可以选择 HBase API 的方式进行数据插入,这种方式简单有效,不需要编写复杂的程序;当面对大批量数据入库需求时,还是非常有必要选择编写 MapReduce 程序使用 Bulk Load 方式进行高效的入库操作。总之,HBase 的 MapReduce Bulk Load 入库方法,为张衡一号卫星的海量数据入库需求提供了解决方案,为后续基于 HBase 的大数据存储方案的设计奠定了基础,同时为科研人员利用卫星数据进行的科研分析工作提供了有效的技术支撑。

2.3 实验结果分析

本实验分别通过 HBase API 和 MapReduce Bulk Load 两种方法,执行 100、500、1 000、10 000、100 000 条数据的入库操作,统计每次入库所需时间进行对比分析,实验结果对比如图 5 所示。可以看出,当插入小于 500 条的较少数据量时,HBase API 方法的 Put 方式耗时小于 MapReduce Bulk Load 方式,略有优势;但是随着数据量的成倍增大,HBase API 方法的耗时显著增加,而 MapReduce Bulk Load 方式的耗时虽略有增长,却微乎其微,表明当大批量