DOI:10.14075/j.jgg.2017.S3.001
文章编号:1671-5942(2017)S3-0001-08

国家地震数据灾备中心大数据存储平台建设方案与研究

柴旭超[1] 王文青[1] 王丹宁[1] 何 松[2] 王庆良[1]

1 中国地震局第二监测中心(国家地震数据灾备中心),西安市西影路 316 号,710054
2 陕西云集华海信息技术有限公司,西安市高新三路财富中心二期 B 座,710075

摘 要:针对国家地震数据灾备中心海量地震波形数据的存储与恢复需求,设计提出一种基于 Kafka 消息队列的分布式集群处理模型,对地震波形数据的 HBase 读写效率进行优化。首先,对比大数据存储架构与传统磁盘阵列存储架构的特点;其次,实现地震波形数据解析及周期性截断,通过 Spark Streaming 集群以记录形式写入 HBase;最后,对比测试大数据平台数据处理过程与传统存储架构读写性能。结果表明:本设计实现的 HBase 地震波形数据处理架构读写速度比传统存储架构有显著提升,验证了本设计的有效性与实用性。

关键词:Kafka;SparkStreaming;HBase;地震波形数据;Key-Value;一致性
中图分类号:P315 **文献标识码**:A

地震观测波形数据来源于台网台站测震传感器的采集与转发,并通过 Jopens 软件逐步汇聚形成实时流数据[1],最终将实时流数据形成归档文件写入磁盘阵列是目前地震波形数据存储的主要形式。国家地震数据灾备中心(简称灾备中心)主要任务之一就是针对此两种形式的测震数据进行备份:1)测震波形数据实时流数据;2)传统磁盘阵列提供的 NAS 文件数据。

大数据时代,针对不同维度数据的数据挖掘工作可以有力地推动行业的进步。随着地震观测手段的提升及观测网络的加密,地震观测数据量与日俱增,对文件形式存储的测震数据进行数据挖掘或关联性分析需要极大的系统资源,因此需要将文件形式的测震数据以数据库的形式进行存储以符合数据应用需求。然而,传统地震数据存储于关系型数据库中,例如前兆数据传统关系二维表数据存储,无法解决地震大数据带来的读写及计算等需求,如表连接运算的性能瓶颈、数据表底层存储 I/O 瓶颈等。因此,测震数据的存储有必要选取一种有利于进行海量数据挖掘分析的大数据存储架构。Google 在 2006 年发表论文阐述 HDFS 和 BigTable 后,人们越来越多地使用分布式文件系统与 NoSQL 分布式数据库相结合的形式代替关系型数据库管理系统(RDBMS)[2],例如,Apache 社区的 HBase(Hadoop Database)[3]、Facebook 的 Cassandra、Amazon 的 Dynamo 等,均使用了列式数据库 Key-Value 数据模型,其中 Apache 社区的 HBase 使用最为广泛,目前已广泛应用于腾讯、阿里及百度等互联网行业,解决海量数据存储与高速查询业务。HBase 同样也是解决地震大数据存储读写效率及可扩展性、分析挖掘及分析预报的有效手段,并在行业内已有相关应用[4]。

HBase 是构建在 Hadooop 之上的一种主流分布式的、可扩展的、面向列的数据存储,是一种可快速、实时随机数据访问的 NoSQL 数据库[5]。目前,将地震波形数据以 Key-Value 形式写入 HBase 分布式数据库的理论模型已在地震行业领域得到应用,表明利用 HBase 列式数据库存储地震波形数据能够有效提高此类数据存储效率。刘坚等[6]已对 MySQL 与 HBase 地震数据存储结构化、非结构化数据进行了对比测试,证明了 HBase 存储地震数据无论在存储效率还是查询效率方面都有极其显著的提升。然而,这些方法尚未针对海量测震数据的 HBase 入库提出解决方案,从而无法利用集群优势、集群并发处理实现测震数据的高速写入。为解决此问题,本文首先介绍灾备中心选取分布式

收稿日期:2017-09-08
项目来源:国家地震数据灾备中心项目(152038000000150003);中国地震局地震科技星火计划青年项目(XH16056Y)。
第一作者简介:柴旭超,助理工程师,主要从事信息化及国家地震数据灾备中心数据存储及服务工作,E-mail:chai_cx@126.com。
通讯作者:王庆良,博士,研究员,主要从事三维地壳运动特征及其动力学研究,E-mail:wangql63@163.com。

存储系统的策略与依据,并引入大数据分布式处理架构的Kafka消息系统,设计提出基于Kafka消息队列及Spark Streaming实时计算框架的测震波形数据处理架构,将测震波形数据从传统磁盘阵列或实时流数据解析成Key-Value形式写入HBase分布式数据库,解决将地震波形数据写入分布式数据库的速度瓶颈问题,从而提升此类数据的解析、存储及查询效率。

为了将海量地震波形数据高速写入HBase分布式数据,本文首先针对测震数据设计了HBase分布式数据库结构,按年建表并由〈台网〉〈台站〉〈通道〉〈开始时间〉组成RowKey,确定了测震数据HBase存储格式。其次,实现了对miniSEED地震波形数据的Key-Value解析:1)通过C语言实现对测震数据的周期性截断与msgKey-msg解析;2)将msgKey-msg格式解析成为Key-Value格式进行HBase入库。再次,为了进一步提升将地震波形数据写入HBase的效率,与地震行业当前存储方案[6]相比,笔者引入Kafka环节[7],针对地震波形数据设计提出了基于Kafka生产消费模型与Spark Streaming实时计算框架的数据入库流程,实现了集群并发处理地震波形数据的目的,有效提升了将测震数据并发写入HBase的速度。最后,通过在灾备中心大数据存储计算集群上对本设计提出的测震数据解析存储架构进行入库性能测试,并与单机将测震数据写入HBase的模式进行对比,从而验证了基于Kafka生产消费模型的测震数据存储方案可以有效提升将测震数据写入HBase的效率。

1 地震波形数据大数据存储架构与传统存储架构的对比

由于观测手段及数据采集能力的进步,地震观测数据无论从采样率和采样精度上都有大幅提高,数据量激增,使用传统磁盘阵列的数据存储手段在存储效率、可扩展性、可靠性以及性价比等方面逐渐无法满足需求,例如采用磁盘阵列RAID技术仅允许文件系统冗余条件下的磁盘数量失效,而无法规避节点级故障。地震数据Hadoop的出现解决了大数据时代下的海量地震数据存储和处理问题,是一种支持分布式存储和计算的框架体系,其核心组件HDFS分布式文件系统可有效支撑海量地震数据的读写能力,不仅可支持节点级系统故障,更是在可允许磁盘失效数量上较磁盘阵列存储有了极大的提升[8]。HDFS分布式存储系统已经作为大数据存储的事实标准,用于海量文件数据的在线存储[9]。

在分布式存储文件系统中,数据的冗余是保证系统可靠性,提高数据高可用性的基本因素[10]。在同一文件系统中,针对一个数据文件通过在跨节点存储其实例即可实现数据的有效冗余,在出现部分数据失效情况下重构原始数据,从而达到可用性的目的。分布式文件系统的冗余策略均需确定两点:一是建冗余数据策略,二是重构数据策略。当前,主流分布式文件系统使用的冗余策略是复制和纠删码:前者采用副本机制,跨机架存储数据文件以实现高可用冗余能力;而后者首先将进入文件系统的数据条带化,然后根据冗余比计算出冗余条带个数[11],最后保存在不同节点中。由于数据存在冗余,系统能够容忍某些节点的失效,从而在不可靠的节点集合中获得某种程度的系统可靠性。然而,当系统的规模扩大且需要长期可靠运行时,对系统数据冗余的可靠性进行理论分析显得尤为重要。已有大量文献[12]分析了不同冗余策略的数据可用性、分布式存储系统的节点失效和修复过程,从理论上计算出存储系统失效的概率、可靠运行的时间、所需的副本数目、系统的生命周期等,从而证明了HDFS分布式文件系统纠删码技术无论在系统可用性、存储空间利用率、系统能耗等方面均优于副本机制,如表1所示。

如MySQL、ORACLE等关系型数据库存在扩展困难、维护复杂等问题,且无法处理上亿甚至上千亿的数据规模。而HBase作为一个高可靠性、高性能、面向列、可伸缩的分布式数据库[13],其底层文件系统使用HDFS,可通过增加节点对系统进行线性扩展,在处理海量地震数据时可提供大规模并发能力。HBase是Key-Value形式的数据库,HBase中的表具有如下特点:1)体量大:一个表可以有上亿行,上百万列,对于海量时间序列的地震观测数据可采用记录形式进行存

表1 灾备中心存储系统对比

Tab.1 Features of storage system of NSDBC

	存储利用率/%	可扩展能力	数据读取性能	数据写入性能	数据重构速度	是否支持节点故障	是否支持HDFS接口
磁盘阵列(RAID5)	80	低(PB级)	低	中	低	否	否
分布式存储(3副本)	33	高(EB级)	高	低	中	是	是
分布式存储(8+2纠删码)	80	高(EB级)	高	高	高	是	是

储;2)面向列:面向列(族)的存储和权限控制,列(族)独立检索;3)稀疏表设计:对于为空(null)的列,并不占用存储空间,因此表可以设计得非常稀疏;4)每条记录的数据可以有多个版本,默认情况下版本号自动分配(时间戳),对于修改数据操作具有可操作性与可回溯性;5)HBase 中的数据都是字节,没有类型。基于以上特点,再考虑 HBase 自身可扩展性及执行效率等优势,灾备中心数据存储结构选择 HBase 分布式数据库与基于磁盘阵列纠删码的 HDFS 文件系统组合策略。

2　HBase 数据库存储设计

为了将海量测震波形数据以 NoSQL 形式进行存储,本研究提出针对测震数据的 HBase 分布式数据库 RowKey 格式,将测震数据以 Key-Value 形式进行入库。

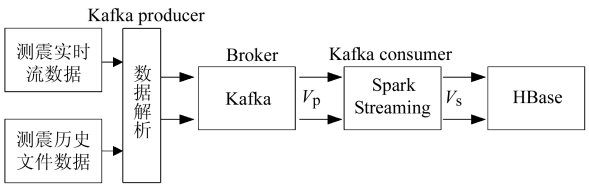

图 1　数据解析存储整体架构

Fig. 1　The data analysis stores the overall structure diagram

如图 1 所示,针对测震波形文件以及实时流数据进行解析形成的相同格式数据,本研究均采用 HBase 列式数据库进行存储。HBase 的 Key-Value 数据结构由关键字 RowKey 和列簇 Value 组成。其中 RowKey 作为数据设计的基础,类似关系型数据库中的主键,将决定数据查询和分析的效率;而 Value 可以看作与 Rowkey 对应的数据体部分,由列名(Columnfamily:column name)和列簇组成。

为了实现测震波形数据高速写入 HBase 分布式数据库的目的,本文引入 Kafka 生产消费模型对测震波形数据进行并发处理。Kafka 是一种高吞吐量的分布式发布订阅消息系统,它可以处理消费者系统中的所有动作流数据,目的是通过 Hadoop 的并行加载机制来统一线上和离线的消息处理,也是为了通过集群来提供对数据的实时处理能力。将地震波形数据的实时数据流或归档文件解析成指定格式数据,通过 Kafka 消息系统并行转发至用户端 Spark Streaming 作为其处理对象,将极大提高 Spark 集群并发数据处理能力[14]。

数据解析在进入 Kafka 消息系统前完成:输入为文件数据或实时流数据(图 1),经过对配置的读取及数据的处理,形成如表 2 所示的指定格式 msgKey:msg 的数据发送至 Kafka 消息系统。本文记录数据完整标识和缺失数的主要目的在于灾备中心对测震数据的接收质量统计与 HBase 入库记录。

表 2　Kafka 获取 Producer 信息格式

Tab. 2　Producer of Kafka information format

msgKey	台网_台站_通道_设备 ID_开始时间_采样率_数据完整标识_缺失数
msg	纯数据块部分

地震波形数据业务逻辑通过台网、台站、通道、开始时间进行查询,本文将此 4 项组合成 Rowkey,而具体数据存储在 Value 中。根据地震波形数据的读写需求特性,以及对数据的安全性考虑,灾备中心将测震数据按年进行分表。如表 3 所示,WaveForm_head 数据表存储地震波形数据文件头信息,WaveForm_YEAR 数据表按年份存储数据体测震数据值,表名设计规则为后 4 位 YEAR 为年份。

表 3　HBase 数据库设计

Tab. 3　HBase database design

表名	解释
WaveForm_head	测震数据文件头信息存储表
WaveForm_YEAR	测震数据年表

根据上述测 HBase 设计规则,由〈台网〉〈台站〉〈通道〉〈开始时间〉组成的 RowKey 在不同年份测震数据表中可根据需求进行组合查询,比如要查询某个台站某个通道某个时间段数据,就可以指定起始 RowKey 为〈台站〉〈通道〉〈开始时间〉,终止 RowKey 为〈台站〉〈通道〉〈结束时间〉,其中结束时间可根据测震数据采样率计算获得。

数据存储在 Spark Streaming 获取 Kafka 消费者信息后进行:由 Spark 集群对 Kafka 消费者信息进行处理形成 Key-Value 格式数据写入 HBase。HBase 存储 RowKey 及 Value 字段及格式如表 4 所示,如前文所述设定台网、台站、通道及开始时间为 RowKey 的 4 个字段,将 Kafka 生产者信息中的 msgKey 其他字段解析并追加至 Value 数据块中。

表 4　HBase 存储 Key-Value 数据格式

Tab. 4　Key-Value data format of HBase

RowKey	台网_台站_通道_开始时间
Value	纯数据块部分,并追加设备 ID、开始时间、采样率、数据完整标识、缺失数等信息

3　基于 Kafka 模型的地震波形数据处理 HBase 读写原理与实现

为了将解析后 Key-Value 格式的测震数据高速写入 HBase 分布式数据库,本研究引入大数据

处理架构的Kafka消息队列模型，提出基于Kafka生产消费模型的测震数据入库方案，通过内存实时计算框架Spark Streaming实现测震数据的集群并发高速入库。本研究采用HBase的Java客户端进行数据出库处理，同时对数据进入HBase前后做数据一致性校验。

3.1 波形数据解析

在测震波形数据产生过程中，由传感器直接采集发送的实时流数据和已经归档的历史文件数据，本研究针对实时流及历史文件波形数据进行读取解析处理后形成如前文所述的msgKey-msg格式的数据发送至图1所示数据处理系统。

文件解析过程如图2所示，对历史文件的处理，按照各台站的时间升序方式，处理所有历史文件，从而达到波形数据的时间连续；解析数据成为msgKey:msg格式，按照固定时间周期，将数据信息组装成Kafka消息生产者信息，发布到Kafka系统。

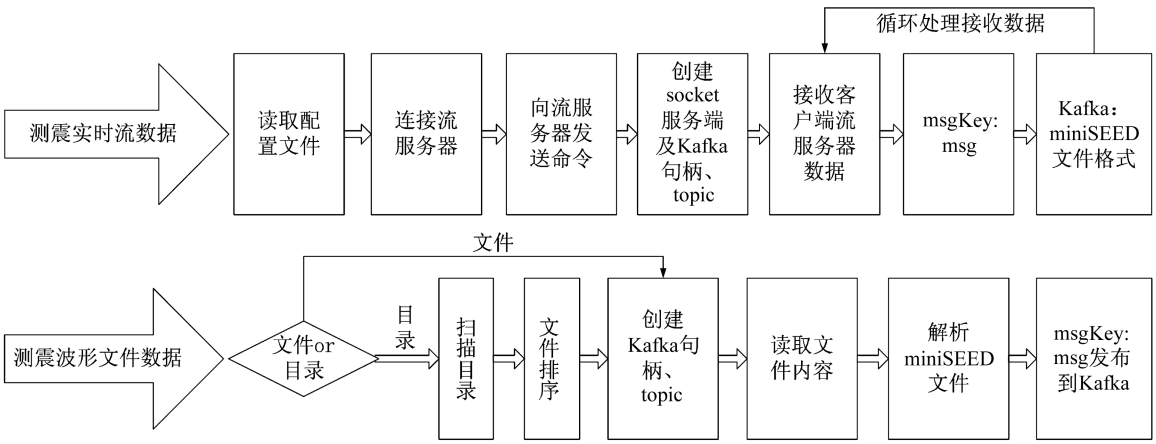

图2 数据解析流程

Fig. 2 Data analysis flow

由图2可看出，数据流解析程序是流服务器的telnet客户端，接收数据的socket服务端，Kafka消息的发布者；①作为流服务器的telnet客户端，程序会向流服务器发送流服务器特有的telnet命令，请求数据；②作为接收数据的socket服务端，在本地某个端口进行监听，等待流服务器连接并接收流数据；③作为Kafka消息的发布者，将接收到的流数据，按照不同台站不同通道的时间先后顺序，组装成固定时间周期的Kafka消息，发送到Kafka。

3.2 Spark Streaming数据存储流程

如图3所示，Spark Streaming作为消费者主动从Kafka消息队列获取msgKey:msg信息。Kafka产生消费者信息后进入Spark Streaming分布式集群处理环节，Spark节点对Kafka消息进行解析形成Key-Value格式数据写入HBase数据库，从而完成对测震波形数据的存储流程。

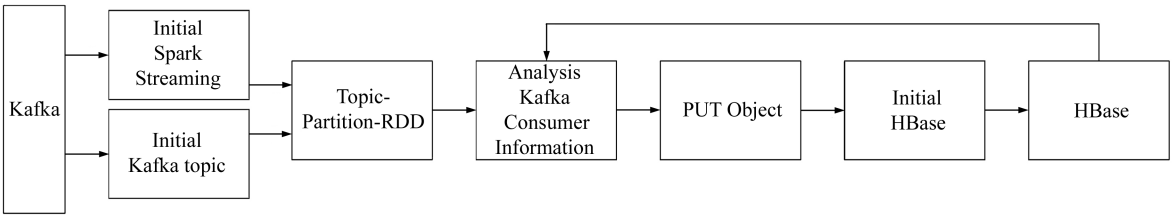

图3 Spark Streaming数据存储架构

Fig. 3 Spark streaming data storage architecture diagram

存储流程主要涉及Spark Streaming、Kafka Consumer及HBase等环节；Spark Streaming是构建在Spark上处理Stream数据的框架，将Stream数据切分成小的时间片，以类似batch的方式处理小部分数据；使用Spark的低延迟处理引擎(100 ms+)，用于实时数据处理；Spark的基础数据模型RDD，即弹性分布式数据集，适合分布式数据计算处理，而RDD的血统机制(RDD之间的依赖关系)使Spark具有高容错，重新计算部分失败任务的RDD的功能；利用Kafka的消费者组，指定每个Kafka的topic的消费者个数，即同时消费topic消息的线程数，可快速消费topic里的消息；将消费到的消息进行分区，各区进行处理消息，生成HBase存储数据的List〈Put〉对象，

存储到 HBase 对应表中。

3.3 HBase 数据出库流程

为了验证灾备中心分布式存储平台的数据输出效率,本研究对 HBase 分布式数据库出库以及数据一致性等方面进行测试。

3.3.1 波形数据 HBase 出库流程

灾备中心地震波形数据以 Key-Value 记录形式存储于分布式数据库 HBase 中,Java 客户端通过 HBase 静态查询接口设置接口相关参数(表 5)进行 HBase 连接初始化,进入数据出库流程(图 4),最终写入指定存储介质。其中,HBase 查询接口返回类型统一为 ResponseBean,返回的 ResponseBean 的 dataObj 类型均为 String,String 为生成文件名称。

表 5 HBase 接口格式

Tab. 5 Interface format of HBase

接口参数	参数顺序	参数描述
HBaseUtil hBaseUtil	1	连接 HBase 工具类
String filepath	2	导出文件路径
String net_stations	3	台网与台站对应
String channels	4	通道,可以是 3 个通道的任意组合 (BHE、BHZ、BHN)
String startTime	5	查询开始时间
String endTime	6	查询结束时间
WaveFormType	7	查询波形数据类型,枚举类:A//测震数据,B//强震数据,…
int fileType	8	导出文件类型:0-seed 文件,1-miniseed 文件,…

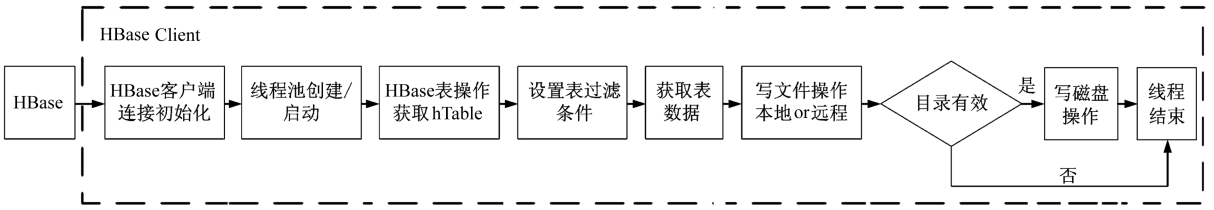

图 4 HBase 数据出库流程

Fig. 4 Data from HBase Outbound flow

首先,设置 HBase 初始化连接,同时设置 HBase 客户端参数,台网、台站、通道及时间解析条件判断;其次,集群创建线程池并启动;再次,管理 hTable 获取 HBase 表操作,通过设置表过滤条件获得表数据;最后进行写文件操作:根据 HBase 客户端具体写盘需求,将出库数据写入本地分布式存储或磁盘阵列等存储介质,或通过远程挂载存储路径进行数据写盘操作,从而实现灾备中心测震波形数据远程恢复功能。

3.3.2 HBase 数据出库查询一致性

测震波形数据实时流从台网中心至西安 HBase 分布式数据库,再从 HBase 中导出。数据出库查询一致性检测主要针对北京原始实时流数据形成的 miniSEED 文件与西安 HBase 查询出库形成的 miniSEED 文件进行对比,验证二者明文字符的一致性。

首先,准备北京 miniSEED 文件作为原始文件;其次,西安分布式数据库 HBase 客户端设置与北京相同文件的查询条件,将所查询出的 miniSEED 文件作为对比文件写入磁盘;再次,利用 3.1 节方法解析原始文件与对比文件,并进一步解析成文本格式;最后,进行文本一致性校验统计。由于 HBase 分布式数据库对于重复记录会进行删除并标记,因此将对于实时流数据中重复的数据记录进行去重处理。

4 对比测试讨论与结论

4.1 测试平台环境与数据集

本研究首先在不同规模数据集、不同集群规模下对基于 Kafka 模型的数据解析写入 HBase 的平均运行时间进行统计对比,并对比单服务器直接将数据写入 HBase 的效率。其次,测试从 HBase 中读取数据生成 miniSEED 文件的速度,并对比此 miniSEED 数据与解析前源文件的文件一致性。测试系统部署于曙光公司开源 HBase 与 HDFS 上,且参数未做深度调优。

为了验证本研究提出搭建的基于 Kafka 生产消费模型的地震波形数据存储与查询系统的有效性与实用性,已将此系统模型运用于备份中心项目测震波形数据存储与查询平台的建设。该集群测试环境共 8 个计算节点,地震波形数据解析程序(起用 8 个进程)、Kafka 组件及 Spark Streaming(多线程并发)均部署于计算节点上。集群节点配置参数如表 6 所示。

表6 计算节点配置信息

Tab.6 Major configuration of single node

属性	配置信息
CPU	10 Core Intel Xeon E5-2650v3 2.3GHz×2
Memory	128GB DDR4
Network	Bandwidth 10Gb/s
OS	CentOS 6.5
JVM Version	Java 1.6.0
Hadoop Version	Hadoop 2.6
HBase Version	HBase 1.0
Kafka Version	0.9.00
Spark Version	1.5.0
ZooKeeper Version	ZooKeeper 3.4.6

4.1.1 灾备中心测震数据HBase入库性能测试

入库测试采用2014年陕西台网台站的miniSEED波形数据,包括相关地震台网台站观测数据文件3 118个,约107 GB。笔者对历史数据以$t=10$ min为截断时间间隔,解析后完成HBase读写性能的测试。数据文件按指定截断时间进行Key-Value形式解析并存储到Hbase,然后从Hbase中查询指定字段信息。测试单节点及不同集群规模下miniSEED文件数据入库、查询的平均运行时间,以及不同集群规模相对单节点数据处理的加速比Speedup,

$$\text{Speedup}=V_c/V_s \tag{1}$$

式中,V_c为集群并发处理速度,V_s表示单节点单进程数据处理速度。

查询测试(不写磁盘)在单节点和集群对HBase入库完成后进行,以验证HBase客户端对出库性能的影响。采用不同配置和性能的客户端查询已写入HBase的波形数据的台网SN、台站HEYT的3个通道的数据。查询客户端Client A为2路服务器,与计算节点配置相同;Client B为4路服务器,硬件配置如表7所示,软件参数及配置与计算节点相同。

表7 Client B主要硬件配置信息

Tab.7 Major hardware configuration of Client B

属性	配置信息
CPU	10 Core Intel Xeon E7-4830v2 2.2GHz×4
Memory	256GB DDR3

4.1.2 灾备中心HBase出库性能测试

HBase出库性能测试针对已入库的测震数据进行抽样选取2017年1月1日~8月23日所有台站3个通道共6 351 446 MB(≈6.1 TB)数据,设置线程数为20执行jar包中的类方法,格式如下:

java -classpath (jar包路径)(jar包中要执行的类)(指定查询写入参数)

指定台网、台站、通道、集群线程数,以及输出文件格式与路径等出库参数;使用任意客户端主机A或B调用HBase数据接口执行此过程出库与写磁盘操作;分别写入灾备中心HDFS分布式存储文件系统、NFS存储磁盘阵列和北京NFS存储磁盘阵列进行对比测试。

4.1.3 灾备中心测震波形数据一致性测试

数据一致性测试的原始数据采用北京台网中心所提供的miniSEED抽样数据:随机从全国33个台网中各选取一个台站的BHE通道一天(2017年1月5日)的miniSEED数据作为原始数据,共33个文件,总量357 MB。再从西安HBase读取相应的miniSEED文件进行对比,此过程通过HBase出库测试执行的类方法,匹配查询条件得到与原始数据对应的33个文件,对比方法如表8举例所示。

表8 数据一致性对比方法举例

Tab.8 Example of data consistency comparison method

台站	数据量(北京/kB)	数据量(西安/kB)	数据量不一致原因	数据一致性
JS/WX	10 081	10 080	Before:重复2条记录	源数据有重复数据块,除去重复数据块后一致:10 081-(2×512)/1 024=10 080
JX/SHC	12 391	12 391	无	一致
LN/YKO	9 686	9 661	Before:重复50条记录	源数据有重复数据块,除去重复数据块后一致:9 686-(50×512)/1 024=9 661

4.2 测试结果及讨论分析

4.2.1 灾备中心测震数据HBase入库性能测试结果与分析

如表9所示,在单计算节点单进程和8个计算节点的集群规模下,对1~100 GB测震数据进行数据解析入库以及查询测试对比,记录5次结果的算术平均,最后对比每轮数据处理平均运行时间,并计算加速比。再通过不同性能的客户端对8个节点组成的HBase分布式数据库查询台网SN、台站HEYT的3个通道的数据,同样每组参数测试5次并以其算术平均值记录时间。

由表9可以看出,处理1 GB数据时单节点平均运行时间1 307 s,8个计算节点并行处理平均运行时间260 s,加速比Speedup=5.027;当数据集规模增加时,加速比逐渐上升:当处理100 GB数据时单节点平均运行时间为39 806 s,而8个计算节点并行处理的平均运行时间4 989 s,加速比Speedup=7.979。从测试结果可以看出,

HBase 入库流程的平均运行时间随着数据量的增大呈线性增长,Kafka 产生的消费者信息进入 Spark Streaming 后可在前者提交下一次 buffer 前完成入库流程;随着待处理的数据量的增大,集群环境下的数据处理性能增长且加速比逐渐上升;而查询性能主要与 HBase 自身提供接口性能有关,因此查询效率取决于 HBase 集群性能,与查询的客户端没有直接关系,而每轮测试所产生的微小耗时波动主要来源于 Spark 与 HBase 的初始化工作读取配置和环境变量的不同。

表 9　HBase 入库平均运行时间与查询耗时

Tab. 9　HBase average written time and query time-consuming

数据量/GB	入库平均运行时间/s		查询耗时/ms	
	1 node	8 nodes	Client A	Client B
1	1 307	260	1 990	1 948
10	5 579	930	2 563	2 557
20	10 364	1 656	3 293	3 133
30	17 405	2 338	4 030	4 157
40	21 182	2 926	4 771	4 723
50	23 820	3 585	5 519	5 583
60	26 744	3 964	6 269	6 192
70	29 334	4 196	6 979	6 872
80	30 597	4 305	7 671	7 509
90	37 257	4 812	7 301	7 211
100	39 806	4 989	7 943	7 730

图 5 对比了不同集群规模时的数据入库加速比。可以看出,由于解析程序进程数与集群节点个数相同,因此随着数据集规模的增大并行计算能力逐渐体现,加速比逐渐达到并发处理的最大能力。当处理 1 GB 数据时,8 节点与 4 节点集群处理加速比相对较低,分别为 5.027 和 2.714;而当数据集规模增长至 100 GB 时,8 节点与 4 节点集群处理加速比逐渐趋于系统极值,分别为7.979和 3.917。

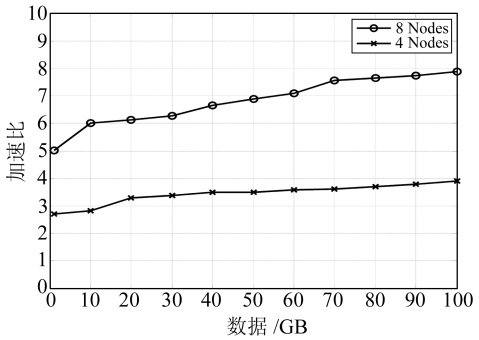

图 5　不同集群规模相对单服务器的加速比

Fig. 5　Speedup of different cluster sizes relative to single servers

4.2.2　灾备中心 HBase 出库性能结果与分析

测试结果如图 6 所示:HBase 客户端程序将所查询出的6 351 446 MB 的测震波形数据写入灾备中心本地 HDFS 分布式存储总耗时11 970 s,平均速度约 531 MB/s;写入本地磁盘阵列存储耗时72 041 s,平均速度约 88 MB/s。由此可看出,基于纠删码技术的 HDFS 分布式文件系统在写入性能方较传统磁盘阵列存储有显著提升。而将相同测试数据写入北京存储磁盘阵列总耗时635 144.6 s(>7 d),平均速度约 12 MB/s。由于西安灾备中心至北京链路带宽为 100 MB,由此可以看出西安 HBase 数据库向北京传输文件速度约 12 MB/s(约 96 MB/s),几乎可以占满行业网带宽,为本次 HBase 出库测试瓶颈。

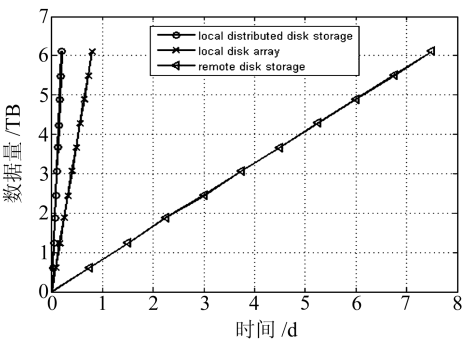

图 6　HBase 不同存储策略的出库耗时

Fig. 6　Local and remote outbound time consuming comparisons of HBase client

4.2.3　灾备中心测震波形数据一致性测试结果与分析

北京对比源数据总量为 357 M,导出后数据总量为 356 M。造成数据量不一致的原因是因为北京源数据文件中包含有重复的数据块,而将数据解析成记录写入 HBase 时,重复记录只写入一次并作出标记,因此导出文件中不存在重复记录,从而导致文件变小。对比 HBase 入库前和出库后数据量,去除 HBase 入库前重复记录后,二者数据一致。

通过搭建此基于 Kafka 集群生产消费信息的数据处理模型,该平台已进行了灾备中心测震历史数据的导入导出工作,验证了该数据解析入库出库模型是其系统成本节约以及计算业务速度提升的有效解决方案。

4.3　总结与展望

为了提升将地震波形数据写入 HBase 的效率,本文实现了对波形数据的 Key-Value 解析,并针对地震波形数据提出了基于 Kafka 生产消费模型与 Spark Streaming 实时计算框架的数据入库流程,实现了集群并发处理地震波形数据的目的。下一步,笔者将会针对地震波形数据写入 HBase 的截取时间间隔进行研究并提出优化方案。另外,将尝试在 Kafka 消息队列接收信息及 Spark Streaming 获取信息时做不同算法的数据压缩处理,从而在优化存储空间利用率的同时减少系统

I/O操作。

致谢:感谢陕西云基华海公司杨天宇先生对Spark Streaming环节提供自动化测试脚本;感谢王方建研究员在灾备中心建立以及本文撰写过程中提供的建议和帮助;感谢台网中心郭凯提供原始对比数据。

参考文献

[1] 吴永权,黄文辉. JOPENS流服务与TDE-324系列地震数据采集器实时数据流接口程序的设计与实现[J]. 华南地震,2011,31(3):50-58(Wu Yongquan, Huang Wenhui. Design and Realization of Real-Time Data Stream Interface Routine between the JOPENS SSS Service and the TDE-324 Series Earthquake Data Acquisition[J]. South China Journal of Seismology, 2011,31(3):50-58)

[2] Lakshman A, Malik P. Cassandra-A Decentralized Structured Storage System[J]. ACM SIGOPS Operating Systems Review,2010,44(2):35-40

[3] HBase:Bigtable-Like Structured Storage for Hadoop Hdfs [EB/OL]. http://hadoop.apache.org/hbase/, 2010

[4] 王方建,李卫东,赵国峰. 地震观测数据平台体系架构研究[J]. 中国地震,2009,25(2):214-222(Wang Fangjian, Li Weidong, Zhao Guofeng. Research on the Architecture of Seismic Observation Data Platform[J]. Earthquake Research in China,2009, 25(2):214-222)

[5] Dean J, Ghemawat S. Map Reduce: Simplified Data Processing on Large Clusters[J]. Communications of the ACM, 2008,51(1):107-113

[6] 刘坚,李盛乐. 基于Hbase的地震大数据存储研究[J]. 大地测量与动力学,2015,35(5):890-893(Liu Jian, Li Shengle. Research of Earthquake Big Data Storage Based on Hbase[J]. Journal of Geodesy and Geodynamics, 2015,35(5): 890-893)

[7] Kafka A. A High-Throughput Distributed Messaging System[EB/OL]. http://kafka.apache.org/

[8] Dean J, Ghemawat S. MapReduce: A Flexible Data Processing Tool[J]. Communications of the ACM, 2010,53(1):72-77

[9] Apache Hadoop[EB/OL]. http://hadoop.apache.org/core

[10] Ahmed T O, Miquel M. Multidimensional Structures Dedicated to Continuous Spatiotemporal Phenomena[C]. Proceedings of the British National Conference on Databases,2005

[11] Abello A, Ferrarons J, Romero O. Building Cubes with MapReduce[C]. Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP,2011

[12] Utard G, Vernois A. Data Durability in Peer to Peer Storage Systems[C]. Proceeding of IEEE GP2PC'04, Chicago, Illinois,2004

[13] Dodge D A, Walter W R. Initial Global Seismic Cross-Correlation Results: Implications for Empirical Signal Detectors[J]. Bulletin of the Seismological Society of America, 2015(105): 240-256

[14] Neumeyer L, Robbins B, Nair A, et al. S4: Distributed Stream Computing Platform[C]. 2010 IEEE International Conference on Data Mining Workshops (ICDMW),2010

Research on Seismic Waveform Data Storage Architecture of National Seismic Data Backup Center

CHAI Xuchao[1] *WANG Wenqing*[1] *WANG Danning*[1] *HE Song*[2] *WANG Qingliang*[1]

1 National Seismic Data Backup Center, The Second Monitoring and Application Center, CEA, 316 Xiying Road, Xi'an 710054, China

2 Yunji Huahai Information Technology Co Ltd of Shaanxi, Phase II B of Wealth Center High-Tech Road, Xi'an 710075, China

Abstract: In order to satisfy the requirement of storage and recovery capacity of national seismic data backup center(NSDBC), this paper presents a distributed cluster model of seismic waveform data based on Kafka message queue to optimize the efficiency of the current HBase read and written method for seismic waveform data. Firstly, the features of big data storage architecture and traditional disk array are compared. Secondly, the seismic waveform data analysis and periodic truncation are completed to by using C program and the information is written into the HBase distributed database through the Spark Streaming cluster with Key-Value record form. Finally, the serial flow is compared with the speed of the data processing flow that presented in this paper, which is based on NSDBC distributed storage and big data computing platform. The results show that the distributed seismic waveform data processing architecture proposed in this paper has significantly improved the performance of single-machine and thread, which proves the validity and practicability of this work.

Key words: Kafka; Spark Streaming; HBase; seismic waveform data; Key-Value; consensus

Foundation support: National Seismic Data Backup Center Project , No. 152038000000150003; The Spark Youth Program of Earthquake Technology of CEA, No. XH16056Y.

About the first author: CHAI Xuchao, assistant engineer, majors in informatization and national seismic data backup center data storage and service, E-mail: chai_xc@126.com.

Corresponding author: WANG Qingliang, PhD, researcher, majors in 3D crustal movement and dynamics, E-mail: wangql163@163.com.