文章编号：2096-1472(2019)-12-44-03
DOI:10.19644/j.cnki.issn2096-1472.2019.12.013

大数据分析下分布式数据流处理技术研究

刘 琴

(青海民族大学计算机学院，青海 西宁 810007)

摘 要：由于数据流的不稳定性，将数据流查询安排在固定节点上就会造成分布式数据流处理技术很难对计算资源实现较高的处理效率，基于此，提出大数据分析下分布式数据流处理技术研究。具体流程是数据收集、历史数据的存储和查询、Storm实时处理、智能索引、数据模型的建立。根据实验结果可知，本文提出的大数据分析下分布式数据流处理技术与传统技术相比，在数据流的处理效率上占有较大优势，一般维持在75%以上，能够大大节省处理时间。

关键词：大数据；分布式；数据流处理技术；处理效率
中图分类号：TP333 **文献标识码**：A

Research on Distributed Data Flow Processing Technology under Big Data Analysis

LIU Qin

(*School of Computer Science*,*Qinghai Nationalities University*,*Xining* 810007,*China*)

Abstract: Because of the instability of data flow,it is difficult for distributed data flow processing technology to achieve high processing efficiency for computing resources by arranging data flow query on fixed nodes.For this reason,this paper proposes the research of distributed data flow processing technology under big data analysis.The specific process is data collection,historical data storage and query,storm real-time processing,intelligent index,data model building.According to the experimental results,compared with the traditional technology,the distributed data flow processing technology proposed in this paper has a greater advantage in the efficiency of data flow processing,generally maintained at more than 75%,which can greatly save processing time.

Keywords:big data;distributed;data flow processing technology;processing efficiency

1 引言(Introduction)

近几年，随着信息技术与计算机技术的迅猛发展及其大规模推广应用，越来越多的客户逐渐加入互联网世界中，全球范围数据总量也呈现出了爆炸式增长趋势[1]。面对如此庞大的数据，传统的数据处理技术已经远远不能满足人们对数据应用的需求。主要是由于传统数据处理技术以处理器为核心，利用数据移动对其进行一系列的操作。在大数据分析背景下，因为数据总量过于庞大，给数据移动也带来了诸多不便，所以迫切需要采取以数据为中心的处理模式，以此降低因数据移动而带来的一系列开销。除此之外，传统的数据处理技术满足不了大规模数据流的处理，无法提供充足的存储空间，因而面向大规模数据流的处理技术已发展为一项全新的挑战。

对此，在大数据分析环境下，提出新型的分布式数据流处理技术。此技术主要是利用基于Map Reduce的Hadoop系统处理构架实现的[2]，因Hadoop系统在常规环境下的处理策略是先进先出(FIFO)，此类处理策略在运行任务时，以数据流到达时间点为依据进行相关处理，基本不将数据的本地性和集聚拓扑构造列入考虑范围之内，可避免因为任务等待时间过长、资源使用率较低、没有考虑任务优先级别等原因导致的紧急作业无法得到优先处理的问题，所以针对基于Map Reduce的分布式数据流处理技术研究将会推动大数据分析下Hadoop系统的广泛应用。

2 大数据分析下分布式数据流处理技术(Distributed data flow processing technology under big data analysis)

2.1 数据收集

海量数据是产生大数据的基本条件，然而数据的收集就成了大数据分析的基础[3]。日志数据采集处于流数据集中一个很大比例，大部分公司的业务平台每一天都将产生数量

庞大的零散数据，将这些业务日志数据集中进行收集并加以整合，用来满足客户在线和离线情况下能够同时使用。需考虑日志收集的基本特点是：可靠性能高、实用性强、可扩展性强。“分散收集、统一处理”是目前主流的日志处理的技术手段。日志收集也变成了分布式日志数据处理的基础和前提。只有在完成日志的实时收集和整合后，才能继续跟踪日志之后的相关操作。

2.2 历史数据的存储和查询

有关分布式数据库的历史数据存储和ORM技术的联系，与传统数据库存储区别较大，分布式数据库的数据在硬盘中支持混合手段(依据行或列）进行混合存储和管理[4]。因为列存储构造对数据查询、整理和分析类操作具有一定的优势，所以在运行分析管理系统等大数据分析背景下混合存储时可以获得较好的应用效果。而传统数据存储主要根据数据大小进行优先分配进行存储，存在存储不佳的问题。有关混合存储的优势主要表现在以下几方面：

首先具有更高的灵活性：对数据进行混合存储可根据列或行分别进行，每张表或表分区可以被管理员按照现实需求或数据格式的不一样进行直接操作处理，选择不同的存储和压缩方法。这种方式能够在一定程度上有效提高系统整体配置的灵活性，具体如图1所示。

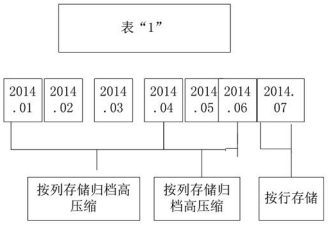

图1 混合存储示意图

Fig.1 Schematic diagram of hybrid storage

其次可以大大提高其响应速度：在对语句进行查询时，传统的行存数据库必须从硬盘上将整行数据全部提取出来，而列存储只能够读取所需数列，不读取其他列的数据[5]。这种方式能够有效降低I/O的运营成本，提高数据查询功能和响应速度。

最后在高扩展性，分布式数据库独特的存储格式可以将列数据细划为“数据包”的格式。不管一个表的内存有多大，数据库只会对有关的数据包进行标准操作，其性能并不会随着数据量的增多而降低，如此表数据就能够实现较高的可扩展性。

ORM(Object Relation Mapping）也叫作对象——关系直接式映射，是针对面向对象的软件开发手段而出现的。它主要作用于程序对象至关系数据库内数据的直接映射，ORM的引进对于数据库数据的相关操作有非常重要的作用，使数据处理与查询能够更加方便和迅速。ORM常常用于数据的长久性工作，比较常用的ORM技术主要包括OJB、MFC-OODE、Hibernate、PDO、TJDO等。本文主要采取MFC-OODE进行常规性的数据存储与查询处理，数据库内的表信息在该系统中均是以类对象的方式存在，而对于系统中在处理后出现的历史数据，可利用定时器进行精准控制，当数据流超时或数据流缓冲区内存被占满时，可以一次性将存储数据即内存中的类对象信息全部导入数据库中方便数据查询。利用该机制能够将多个数据流一次性全部导入数据库内，大大节省了相关数据表和数据库的处理时间，有效地降低了I/O出现并发冲突的可能性，提升了系统的处理能力。有关数据查询方面，数据查询所提取到的数据能够从系统硬盘中获得，也能够从数据库中获得，这是由于系统利用定时器对历史数据进行存储操作，如果是对最近存储的数据信息进行相关查询，则可以直接从硬盘内中提取，这能够大幅度提高查询效率，同时也省去了对数据库和数据表的一系列操作步骤节省操作时间。

2.3 Storm实时处理

在大数据分析的影响下，将数据实时进行收集与整合，使其变成有效数据流之后，想要尽可能快的得到应用系统实时需要数据结果，则数据解析系统一定要对原始数据迅速地完成一系列实时处理[6]。处理时，一台服务器不能在短暂的时间内满足整个系统能够计算超大量数据的需求，主要是因为考虑到业务水准和数据上升幅度的原因，需要数据处理系统具有很强的扩展性。storm最开始是通过twitter开发来扩源，以分布式实时数据处理系统为基础，在twitter、Yahoo等众多著名的互联网公司得到了大力的推广和应用。具有良好的可扩展和容错性，并能够获得次秒级的延迟，适用于延迟较低的应用环境[7]。它的主要组成部分是：

(1)Nimbus是数据集群的重要节点，主要负责数据集群的资源管理、任务分配。

(2)Supervisor主要用作接收Nimbus分配的任务，能够实时结束系统工作管理的进度。

(3)Zookeeper是storm主要应用的外部组件，提供Supervisor和Nimbus之间的协调服务，Nimbus和Supervisor的资源任务管理状况都存储在Zookeeper内。storm组合的数据流包括Topoloy(拓扑)，与Hadoop上的Map Reduce任务类似，节点间的数据流动方向形成了标准形式下运行的数据处理逻辑；Tuple(消息元组)，也就是最小的消息处理和上传单元，每个Tuple均为不可逆的数据消息组；Spout(喷嘴)，主要的职责是把从storm外部获取的数据转换成内部数据组件，并上传初始数据到Tuple；Bolt(螺栓)的职责是接收来自Spout或者上个流程的Bolt的Tuple内传送的信息，在内部进行简单的数据转换和计算以后，会产生很多的输出Tuple数据流，再把它发送到别的Bolt，互相合作进而实现更为复杂的计算逻辑。

2.4 智能索引

智能索引和以往数据库在行数据上建立数据细粒度索引的技术比较，分布式数据库的智能索引是一种建立在数据流基础上的数据粗粒度索引。每一个数据流在完成数据加载后就会自动建立，其中包含数据过滤信息和整合信息[8]。粗粒度的智能索引包含了预存数据之间互相依存关系的高级信息，可以精准描绘和识别出数据流的实际需求，有效完成复杂多表区的连接和子查询问题。表中所有列完成自动建立后，无须用户自行建立和人工维护。这就使得智能索引对数据存储空间的占用较低，具备较高的扩展性，可以在使用索引后不会发生数据膨胀。后续数据流构建索引的速度也不会受到前面数据流的影响，加快索引构建速度。

2.5 数据流处理模型的建立

数据流s作为由数据元组构成的无限序列，用$s=(t_1,t_2,t_3,...)$表示。数据元组可以用$t=(key,value,TS)$表示，其中的key代表数据元组的按键；$value$代表数据元组的细分数值；TS代表数据元组的作用时间段。数据元组的key并不是唯一指定的，一般其经常用来确保数据元组路由的顺利。时间段TS主要是由一个持续递增的逻辑时钟在数据元组建立之初完成标准分配。数据元组在数据流处理过程中根据时间段有序排列。数据元组由诸多的操作符组成。一个操作符具体是以一条或多条数据流为输入$I_0=\{s_1,s_2,s_3,...,s_n\}$，处理输入数据流中的主要元组，并产生一条或多条输出数据流o_0。操作符函数用f_0表示操作符对输入数据流元组的处理程序。操作符主要有两种：无状态操作符(比如过滤和映射)和有状态操作符(比如连接和集聚)。有状态操作符的操作函数以$f_0=(t,o_0)\rightarrow(t',o')$表示，其会在新的数据元组抵达时被调取。有状态操作符保留了之前处理数据流元组的状态o_0。当一个新的数据流元组t'抵达并被操作符函数完成处理后，就会产生新的数据流元组t'，而同时其状态o_0也会随之被更换为o'，实现数据流的综合处理。

3 实验与效果分析(Experiment and effect analysis)

为了更加直观地看出大数据分析下，本文提出的分布式数据流处理技术的实际应用效果，采用本文方法与传统分布式数据流处理技术为对比，以处理效率作为标准进行实验对比分析。

3.1 实验准备

为确保实验的准确性，把这两种分布式数据流的处理技术处在同样的服务器环境中，再进行处理能力有关的实验。具体的服务器配置见表1所示。

表1 服务器配置

Tab.1 Server configuration

服务器类型	数量	配置参数
分布式消息队列服务器	3	处理器3×8核，内存128G，磁盘3×700GB/4×5B，千兆以太网
数据存储服务器	1	处理器3×6核，内存64G，磁盘3×300GB/6×5TB，千兆以太网
流计算服务器	5	处理器2×8核，内存128G，磁盘2×700GB/64×5TB，千兆以太网

在实验中使用了三个数据集(S1,S2,S3)，为了便于表述，使用Size(Si)(1≤i≤3)表示数据集的规模，Size(Si)的单位为数据条数。HadoopDB和HBase的数据文件格式不同，导致文件大小差异较大。各个数据集相关参数见表2。

表2 OceanCube实验数据流大小

Tab.2 Size of OceanCube experimental data streams

级别	数据流名称		
	S_1	S_2	S_3
Time/Ywar	3年	30年	30年
Area/1	1个	1个	5个
Depth/100m	5层	5层	10层
Size(S_i)	108条	109条	1010条

按照一定的文件格式生成所有的数据，并载入相关数据，由于数据的生成是ETL 阶段，针对处理效率进行对比。

3.2 实验结果分析

实验过程中，在相同配置环境下，通过两种不同的分布式数据流处理技术同时进行工作，分析其处理能力的变化。实验效果对比图如图2所示。

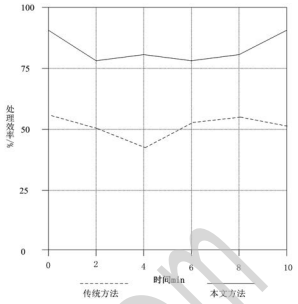

图2 实验结果对比图

Fig.2 Comparison of experimental results

根据实验结果可知，本文提出的大数据分析下分布式数据流处理技术与传统技术相比，在数据流的处理效率上占有较大优势，一般维持在75%以上，能够大大节省处理时间。

4 结论(Conclusion)

本文对大数据分析下分布式数据流处理技术研究进行分析，在分布式环境下，根据大数据反馈与分析，调整数据流的处理技术，完成本文设计。实验论证证明，本文设计的方法有效性极高。可为后续大数据分析下分布式数据流的处理方法提供理论依据。

参考文献(References)

[1] 朱蔚林,木伟民,金宗泽,等.基于MR的高可靠分布式数据流统计模型[J].计算机技术与发展,2018,28(01):6-10;16.

[2] 王春凯,孟小峰.应对倾斜数据流在线连接方法[J].软件学报,2018,29(03):869-882.

[3] 张愿.基于Spark Streaming的在线多数投票提升算法研究[J].福建电脑,2018,34(07):105-107;115.

[4] 相坤,杨建设.面向广域电网的分布式流协同处理技术研究[J].计算机与网络,2018,44(23):68-71.

[5] 郑钤.基于MapReduce模式的大数据分布式计算态势分析[J].通讯世界,2018(06):102-104.

[6] 谭亮,周静.基于Spark Streaming的实时交通数据处理平台[J].计算机系统应用,2018,27(10):133-139.

[7] 付眸,杨贺昆,吴唐美,等.基于Spark Streaming的快速视频转码方法[J].计算机应用,2018,38(12):3500-3508.

[8] 闾程豪,荆一楠,何震瀛,等.基于分布式流处理的自适应数据分发策略[J].计算机应用与软件,2018,35(08):24-30.

作者简介：

刘 琴(1976-)，女，本科，副教授.研究领域：软件工程.