

基于 FPGA 的高效数据过滤技术研究

陈宝远,张秀芝,梁 状

(哈尔滨理工大学 测控技术与仪器黑龙江省高校重点实验室,黑龙江 哈尔滨 150080)

摘 要: 为了提高 Hadoop 的数据处理效率,在 Hadoop 集群中加入 FPGA 对 Hadoop 分布式文件系统中存储的数据进行过滤,这样 MapReduce 程序只需要对过滤后的数据进行自定义的操作,从而避免了扫描全部数据.实验表明处理海量的数据时,FPGA 能够显著地过滤掉无用数据,从而提高数据处理效率.

关键词: 数据过滤;FPGA;海量数据;Hadoop

中图分类号: TN4

文献标识码: A

文章编号: 1000—7180(2017)12—0130—04

Research on Efficient Data Filtering Based on FPGA

CHEN Bao-yuan,ZHANG Xiu-zhi,LIANG Zhuang

(The Higher Education Key Laboratory for Measuring & Control Technology and Instrumentations of Heilongjiang Province,Harbin University of Science & Technology,Harbin 150080,China)

Abstract: In order to increase the efficiency of data processing, FPGA was injected into Hadoop cluster. FPGA will filter the data stored in HDFS(Hadoop Distributed File System). In this way MapReduce just need to process the filtered data. A series experiment shows that FPGA could filter useless data and increase the efficiency of data processing.

Key words: data filtering; FPGA; massive data; hadoop

1 引言

信息时代,人类每时每刻都会产出海量的数据,怎么从海量数据中提取有价值的信息变得更加重要.现今企业的成功和社会的进步很大程度上依赖于数据以及对数据的分析^[1].互联网广告业的飞速发展就是得益于对大数据的成功分析,通过对用户网上行为数据进行分析,互联网广告公司可以做到定向投送广告,从而能够最大化广告的收益.

近年来,FPGA^[2]凭借其良好的并行计算能力及可重构等特性,被许多研究人员用于加速数据库系统的查询操作^[3].Andreas^[4]、Glacier^[5]等人利用FPGA设计出了针对数据库查询操作的加速器.但是传统的关系型数据库已经无法应对海量数据的处理需求,人们迫切需要新的技术对海量数据进行存储和分析.目前,源于Google的Apache Hadoop^[6]

已经成为PB量级数据处理的成熟技术和实施标准.Hadoop主要由两部分组成,HDFS用于将海量的数据存储在分布式的集群上,MapReduce^[7]是一个编程框架用于对HDFS中的数据进行分析处理.但是Hadoop采用扫描的方式对HDFS^[8]中存储的数据进行处理,这种数据处理方式的效率很低.如果能够提前对数据进行过滤,从而避免数据扫描,将会极大提高Hadoop处理海量数据的效率,从而降低人们面对海量数据时消耗的财力和人力^[9].针对Hadoop效率低下的问题,本文设计了一种基于FPGA的数据过滤系统,该系统可以避免Hadoop扫描全部的数据,从而提高数据处理效率.

2 系统设计与实现

Hadoop集群由一个主节点和多个从节点组成,主节点负责任务分发及集群管理,从节点负责任

务执行及数据存储. 当 MapReduce 程序执行时,每个从节点都会读取本地的数据,在内存经过处理后,通过网络进行传输交互,最终得到结果数据. 本文在 Hadoop 系统的基础上设计实现了一种基于 FPGA 的数据过滤系统,从节点中存储的数据经过 FPGA 过滤后才会进入内存,然后进行网络传输,这样就大大降低了数据处理和传输效率,避免了处理和传输无用数据的消耗. 系统结构如图 1 所示,本系统由一个主节点和若干个从节点组成,每个从节点上都配置一个 FPGA.

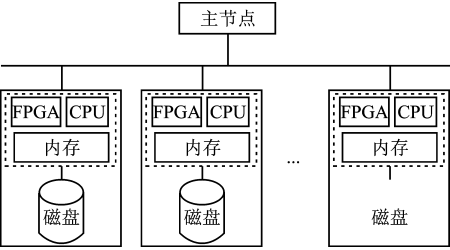


图 1 系统结构示意图

图 1 中主节点负责生成查询计划,查询计划里包含查询操作在何处执行,以及访问数据所涉及到的物理地址,然后通过网络将查询计划交给从节点执行. 查询计划执行过程中,映射(Projection)和选择(Filter)操作由 FPGA 通过 SATA 读取外存数据执行,得到的结果存入内存/中间结果临时文件中(通过内存和硬盘的缓存管理调度),接下来通过 Hadoop 实现其他操作,比如连接(Join)和聚集(Aggregation)操作. 当使用 MapReduce 程序对 HDFS 中的数据进行处理时,为了使用 FPGA,会适当地改变 MapReduce 任务的处理逻辑. 因为 Map 任务对原数据进行处理后得到的中间结果才会交给 Reduce 任务处理,所以为了避免对原数据的扫描处理,只需要改变 Map 任务的处理流程. 查询计划执行过程中,选择和映射操作交给 FPGA 执行. FPGA 对磁盘中存储的原始数据进行选择和映射操作后,将中间结果存入内存,然后交给 Reduce 任务进行其他操作,得到最终的结果. 每一个从节点上数据的过滤流程如图 2 所示.

处理数据的时候,系统会将查询条件中的过滤条件以及需要处理数据的元数据信息传递给 FPGA, FPGA 会根据元数据信息读取 HDFS 中对应的数据,并根据过滤条件对原数据的列、行进行映射和选择,然后将过滤后的数据作为中间结果存储到内存中,这些数据接下来会由 CPU 进行处理或者通过网络进行传输.

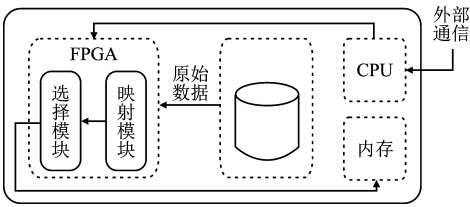


图 2 数据过滤流程

3 FPGA 上的数据过滤

近年来,许多领域的学者对如何利用 FPGA 提高海量数据的处理效率进行了大量的研究^[10],比如数据挖掘^[11]、图像处理^[12-13]以及其他数据密集型的科学领域. 这些领域的操作都是建立在海量数据基础上的,为了提高处理效率,使用 FPGA 对原数据进行过滤,从而对过滤后的数据进行接下来的操作. FPGA 在本文设计的系统中负责数据的过滤,主要包括以下两个模块.

3.1 映射模块

FPGA 对磁盘中的原数据进行处理之前,上层逻辑会传递给它一些列过滤操作所需要的参数,其中就包括查询条件中包含的列信息. 映射模块会标记查询条件中出现的列,即对每一个需要被映射的列生成一个标志信息, FPGA 的映射模块只会将被标记的列进行输出. 系统中的映射模块相当于 SQL 中的 projection 操作.

3.1.1 映射向量

处理数据时系统首先会从主节点读取需要处理的数据对应的元数据信息,元数据中存储着待处理数据的一些统计信息,比如存储位置、数据大小、字段数及每个字段的类型等信息. 然后主节点会根据元数据和用户输入的程序生成一个映射向量(1, 1, 1, 0...),向量中的元素个数等于待处理数据的字段数,向量中的 1 代表用户需要处理该列的数据,0 代表用户不需要处理改列的数据. 比如待处理数据中包含 5 个字段 c1、c2、c3、c4、c5,用户输入的程序中需要处理的字段包含 c1、c4,那么最终生成的向量为(1, 0, 0, 1, 0). 该向量会分发给每个从节点上的 FPGA.

3.1.2 映射过程

FPGA 收到 CPU 传给它的映射向量后,会根据该向量对原数据进行映射操作. 映射过程见图 3. 映射模块的输入时 HDFS 中存储的原始数据和主节点计算得到的映射向量,输出是经过映射之后的数据. 映射向量相当于开关,只有为 1 的列才会

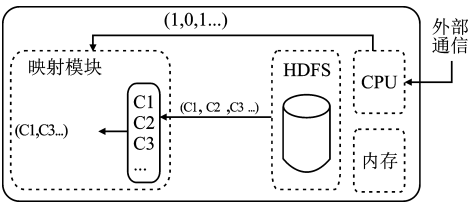


图 3 数据映射过程

被输出,其他的列则会被过滤掉,该模块支持的最大列数为 16.

3.2 选择模块

这个模块的功能是对原数据中的每一行数据进行过滤,相当于 SQL 中的 where 语句的功能. 每当一行数据满足查询条件时,此模块就发出逻辑高信号,否则发出逻辑低信号. 如果查询条件中没有对行进行过滤,则此模块会持续发出逻辑高信号. 选择模块的数据处理流程如图 4 所示.

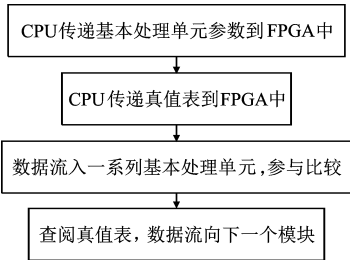


图 4 选择模块数据处理流程

3.2.1 基本处理单元

本模块的核心部件是一系列的基本处理单元,每一个基本处理单元都可以处理如下两种判断语句:

(1)形如 column θ constant 的判断语句. 其中 θ 为比较符号,包括 $=, <, <=, >, >=$; column 为参与比较的列 id; constant 为某个常数或字符串.

(2)形如 column θ column 的判断语句. 其中 θ 和上面一样为比较符号;column 为参与比较的两个列的列 id.

每当一个查询到来之后,CPU 会对查询条件进行分析,并向 FPGA 中的某个基本处理单元传递参数 (column, θ , constant, column). 该处理单元接收到参数之后,就能立即按照过滤条件对数据进行处理,并且当满足判断条件时发出信号“1”,否则发出信号“0”.

3.2.2 多核多流水

一个查询语句中可能拥有多个过滤条件,此时系统采用流水线的方式对某个查询语句进行处理^[14]. 例如,对于语句 where($col_1 > \alpha$ and $col_2 = \beta$)

or $col_3 < \gamma$,这种条件系统采用多核多流水的方式进行并行处理,每一个过滤条件的判定都是并行进行的. 如图 5 所示.

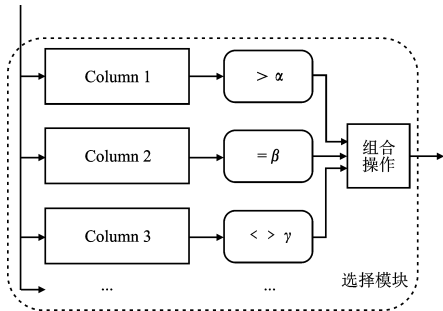


图 5 并行处理

记录流经所有的处理单元,当该条记录的某一属性的列 id 与某一处理单元的 i 列 id 一致时,这个属性就会参与该处理单元的比较. 当某个属性和某一处理单元比较之后,应继续流向后续处理单元,因为该属性有可能参与了多次比较.

3.2.3 真值表

获得每个过滤条件的判断结果之后,系统会根据 and,or 等逻辑词判断整个过滤条件的结果,这里本系统采用真值表的方法^[15]. 同样以 where($col_1 > \alpha$ and $col_2 = \beta$) or $col_3 < \gamma$ 为例,其真值表见图 6.

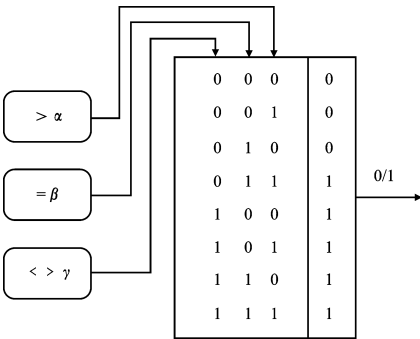


图 6 真值表

真值表是由 CPU 根据过滤条件计算出来的,然后传递给 FPGA. 当过滤条件过多时,可能导致一片 BRAM 不能存放一整张真值表,这时系统采用多重真值表的做法,即把全部的过滤条件分成若干组,每一组对应一个小的真值表,每一个真值表单独计算结果,最终汇总到一起形成一张大的真值表. 分组的方法依赖于 BRAM 的大小.

4 实验

4.1 实验环境及数据

本文使用拥有 4 个节点的机群环境:1 个主节

点,3 个从节点. 每个节点都拥有 1 个 Intel Core i5-2400 3.1 GHz 四核处理器,24 GB 内存,500 GB SATA 硬盘. 同时每个节点上都运行着 64 位的 Ubuntu 14.04 系统,并且每个系统都安装了 Hadoop 2.6.0 软件. HDFS 设置副本数为 2,默认数据块大小是 64 MB. FPGA 开发环境为 Quartus II 9.0. 实验数据由 TPC-H 生成,分别生成 1 GB、10 GB、30 GB、50 GB 的 Customer 数据, Customer 表含有 8 个字段.

4.2 实验结果及分析

对所有的实验论文都进行了三次尝试,并取三次结果的平均值作为最终的实验结果. 实验结果数据如表 1 所示,从表中可以看出,内嵌 FPGA 进行过滤的系统在不同的数据量下的数据处理耗时都比 Hadoop 原生系统小.

表 1 实验数据		
数据量/GB	系统	耗时/s
1	原生系统	72
	FPGA	64.3
10	原生系统	425.6
	FPGA	257.4
30	原生系统	1 172
	FPGA	752
50	原生系统	1 962.2
	FPGA	1 224.4

图 7 对比了原生 Hadoop 系统和添加 FPGA 之后的系统在不同的数据量下,执行相同任务的时间开销. 从图 7 的对比来看, FPGA 系统的时间开销小于 Hadoop 系统;并且随着数据量的增加, FPGA 的优势越明显. 这是因为 FPGA 通过对数据进行过滤不仅降低了数据 I/O 开销而且减轻了 CPU 的压力,避免了对大量无关数据的处理,从而提高了数据处理效率.

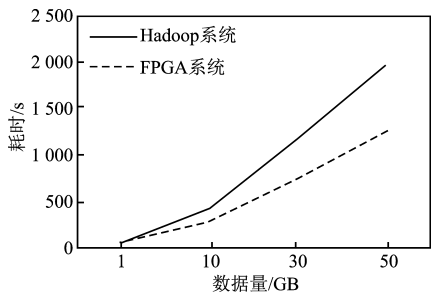


图 7 实验结果对比

5 结束语

本文借鉴了传统数据库利用 FPGA 加速查询

操作的经验,将 FPGA 和大数据处理领域的前沿技术相结合,提出并设计实现了一种基于 FPGA 的数据过滤系统,该系统为 Hadoop 集群中的每一个从节点都配置一个 FPGA,数据处理时 FPGA 会直接读取本地磁盘上的数据,数据经过 FPGA 过滤后加载到内存,然后进行后续处理或者经过网络传输到其他节点. 因为数据直接经过 FPGA 过滤,既减轻了 CPU 压力又避免了无用数据在网络之间的传输,所以能够大幅度的提高数据处理效率. 论文还通过 TPC-H 生成了标准测试数据集,然后在不同大小的数据集上验证了所提方案是有效可行的.

参考文献:

[1] 严宵凤, 张德馨. 大数据研究[J]. 计算机技术与发展, 2013,23(4): 168-172.

[2] 钱松, 周钦, 俞军. AES 算法的一种高效 FPGA 实现方法[J]. 微电子学与计算机, 2005, 22(7): 89-91.

[3] Sukhwani B, Hong M, Thoennes M. Database analytics acceleration using FPGAs[C]//International Conference on Parallel Architectures & Compilation Techniques. IEEE, 2012:411-420.

[4] Becher A, Bauer F, Ziener D, et al. Energy-Aware SQL query acceleration through FPGA-based dynamic partial reconfiguration[C]// FPL. IEEE, 2014:1-8.

[5] Mueller R, Teubner J, Alonso G. Glacier: A query-to-hardware compiler[C]//Proc. of the 2010 ACM SIGMOD Int'l conf. ACM, 2010: 1159-1162.

[6] White T. Hadoop: The definitive guide, [M]. Oreilly Media, 2015: 10-19.

[7] 江务学, 张璟, 王志明. MapReduce 并行编程架构模型研究[J]. 微电子学与计算机, 2011, 28(6): 168-170.

[8] Garcia H, Ludu A. The Google file system[C]//ACM sigops Operating System Review. ACM, 2003: 29-43.

[9] 马友忠, 孟小峰. 云数据管理索引技术研究[J]. 软件学报, 2015, 26(1): 145-166.

[10] Sukhwani, M Hong, M Thoennes. Database analytics acceleration using FPGAs[C]// International Conference on Parallel Architectures & Compilation Techniques. IEEE, 2012:411-420.

[11] YM Choi, KH So. Map-Reduce processing of K-means algorithm with FPGA-accelerated computer cluster[C]// IEEE International Conference on Application-specific System, Architectures & Processors. IEEE, 2014:9-16.

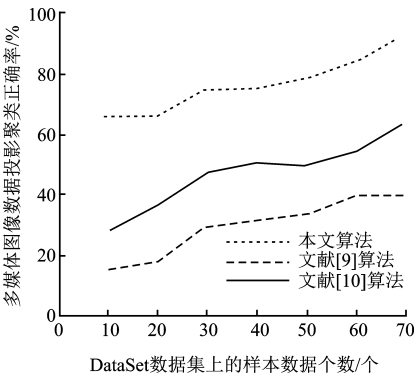


图 1 不同算法在 DataSet 数据集上的聚类准确率

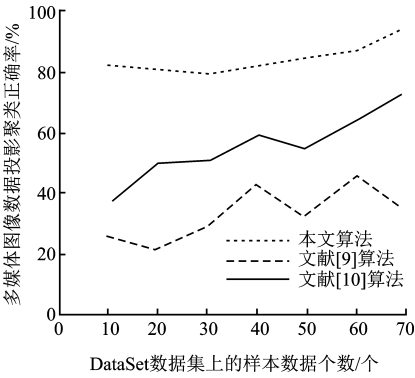


图 2 不同算法在 Iris 数据集上的聚类准确率

4 结束语

采用当前算法对多媒体图像数据投影进行聚类融合时,聚类准确率较低,且算法平均执行时间较长,为此提出一种基于粒子群的多媒体图像数据投影聚类融合优化算法.并通过实验分析证明,所提算法能够在大多数数据集上获得良好的聚类融合结果,提高了聚类准确率,达到了优化传统聚类融合算

法的目的,具有重要的研究意义.

参考文献:

[1] 吴晓璇,倪志伟,倪丽萍. 云计算环境下基于分形的聚类融合算法研究[J]. 计算机工程与应用, 2015, 51(14):1-6.

[2] 冯中慧,何亮,王栋. 基于新的成员选择方法的聚类融合算法[J]. 微电子学与计算机, 2016, 33(11):25-29.

[3] 杨威,朱福喜. 基于聚类融合的标题文本聚类方法[J]. 计算机工程与应用, 2015, 51(15):129-133.

[4] 于耕, 宗平. 基于扩展自适应融合算法的 GILS 逼近系统研究[J]. 计算机测量与控制, 2016, 24(2): 290-292.

[5] 蒋玲莉,莫志军,陈安华,等. 一种聚类优化融合故障诊断方法及其应用[J]. 中国机械工程, 2016, 27(15): 2055-2059.

[6] 郝晶晶,朱建军. 群体分类偏好下的双重语言信息融合聚类方法[J]. 控制与决策, 2015, 30(6):1044-1052.

[7] 刘蜀, 侯国超, 吉玉洁. 舰载指控系统雷达目标航迹信息融合算法研究[J]. 计算机仿真, 2016, 33(3):9-12.

[8] 慕彩红,霍利利,刘逸,等. 基于小波融合和 PCA-核模糊聚类的遥感图像变化检测[J]. 电子学报, 2015, 43(7):1375-1381.

[9] 苏卫星,朱云龙,刘芳,等. 基于改进模糊聚类的同构多传感器在线数据融合方法[J]. 信息与控制, 2015, 44(5):557-563.

[10] 马传宾,翟社平,马蒙雨,等. 一种有效的 Web 服务聚类方法[J]. 电子设计工程, 2016, 24(19):11-14.

作者简介:

陈 雅 女, (1980-), 硕士, 讲师. 研究方向为数据库、大数据. E-mail: chenya7887@163. com.

(上接第 133 页)

[12] Chenini H, JP D, Chapuis R. Parallel embedded processor architecture for FPGA-based image processing using parallel software skeletons[J]. Journal on Advances in Signal Processing, 2013(1):1-23.

[13] Zarifi T, Malek M. FPGA Implementation of image processing technique for blood samples characterization [J]. Computers and Electrical Engineering, 2014, 40(5):1750-1757.

[14] Belean B, Borda M, Bot A. FPGA based hardware architectures for interactive algorithms implementations. [C]// International Conference on Telecommunications & Signal Processing. IEEE, 2013:751-754.

[15] Becher A , Bauer F, Ziener D. Energy-aware SQL

query acceleration through FPGA-Based synaptic partial reconfiguration[C]// International Conference on Field Programmable Logic & Applications. IEEE, 2014:1-8.

作者简介:

陈宝远 男, (1970-), 副教授, 硕士生导师. 研究方向为通信工程、测试计量技术及仪器.

张秀芝(通讯作者) 女, (1990-), 硕士研究生. 研究方向为通行工程、FPGA. E-mail: xiuzhi1225@163. com.

梁 状 男, (1990-), 硕士研究生. 研究方向为集成电路设计.