一种分布式大数据管理系统的设计与实现

陈海燕
(华东政法大学计算机科学与技术系 上海 201620)

摘 要 随着云计算、物联网、移动互联网等技术的飞速发展,海量数据在这些崭新的领域迅猛地生长着,大数据作为一项颠覆性技术,为处理海量数据提供了无限可能。而传统的关系型数据库的不再适用,导致了分布式数据库NoSQL的应运而生。针对大数据领域面临的种种现实难题,设计并实现了一种基于Hadoop和NoSQL的新型分布式大数据管理系统(DBDMS),其提供大数据的实时采集、检索以及永久存储的功能。实验表明,DBDMS可以显著提高大数据处理能力,适用于海量日志备份和检索、海量网络报文抓取和分析等领域。

关键词 大数据,分布式,数据存储,数据检索,Hadoop,NoSQL
中图法分类号 TP391 **文献标识码** A

Design and Realization of Distributed Big Data Management System

CHEN Hai-yan
(Department of Computer Science and Technology, East China University of Political Science and Law, Shanghai 201620, China)

Abstract With the pretty rapid development of cloud computing, internet of things, mobile internet, and other technologies, mass data grows in those areas in violent speed. Big Data provides a possibility of handling mass data, which acts as a subversive technique. By the way, traditional relation database is no more effective of mass data that causes distributed database NoSQL to appear and evolve. Facing with actual and various difficulties, we designed and realized a new distributed big data management system (DBDMS), which is based on Hadoop and NoSQL techniques, and it provides big data real-time collection, search and permanent storage. Proved by some experiment, DBDMS can enhance the processing capacity of mass data, and very fits for mass log backup and retrieval, mass network packet grab and analysis, and other applied areas.

Keywords Big data, Distributed, Data storage, Data query, Hadoop, NoSQL

1 引言

维基百科这样定义"大数据":所涉及的数据量规模巨大到无法通过人工,在合理时间内达到截取、管理、处理,并整理成为人类所能解读的信息,在总数据量相同的情况下,与个别分析独立的小型数据集(data set)相比,将各个小型数据集合并后进行分析可得出许多额外的信息和数据关系性,可用来察觉商业趋势、判定研究质量、避免疾病扩散、打击犯罪或测定实时交通路况等,因此大数据又称巨量数据或海量数据。其4V特点分别为:Volume(大量)、Velocity(高速)、Variety(多样)、Value(价值)。在这个信息爆炸的时代,物联网[1]、云计算[2]、移动互联网[3]等新技术层出不穷,而与之密不可分的大数据技术的出现无疑让人兴奋,其带来无限的机遇与挑战。本文针对大数据检索效率低下的问题,设计并实现的一种新型的分布式大数据管理系统DBDMS(Distributed Big Data Management System),将大数据的检索效率有效提高,适用于海量日志备份和检索、海量实时数据分析、海量网络报文抓取和分析等大数据实时处理的应用场景。

2 DBDMS系统的架构

大数据,顾名思义对数据的采集[4]、存储[5]和检索[6]的效率都有非常高的需求,通常情况下,数据的采集效率可达MB/s或GB/s,而数据的存储可达TB甚至PB量级,传统的关系型数据库由于一致性约束已经很难满足如此高强度的需求。对此Google提出了BigTable数据存储系统,此后大量的非关系型数据库NoSQL应运而生,它们共同的特点在于基于Key-Value进行读写操作,但缺少对多列检索和多表联合统计分析等复杂操作的支持;而另一方面Hadoop使用MapReduce加速数据检索,但由于其缓存机制所限,不支持大量数据的快速采集和检索,导致整体效率很低。为了解决这些问题,我们基于Hadoop[7,8]和NoSQL[9]两种技术实现了一种新型的分布式大数据管理系统DBDMS(Distributed Big Data Management System)。它包括5个主要的部分:中央控制集群、大数据采集集群、大数据检索集群、大数据永久存储集群以及高速以太网连接模块,主要解决海量数据的高速采集和检索以及分布式存储等问题,DBDMS系统架构如图1所示。

本文受国家社会科学基金项目(06BFX051),上海高校选拔培养优秀青年教师科研专项基金(hzf05046),华东政法大学校级科研项目(09HZK014)资助。
陈海燕(1978-),男,硕士生,讲师,主要研究方向为数据挖掘、人工智能、信息安全、数据库,E-mail:tom_chy@163.com。

下面详述每个部分的组成和作用。

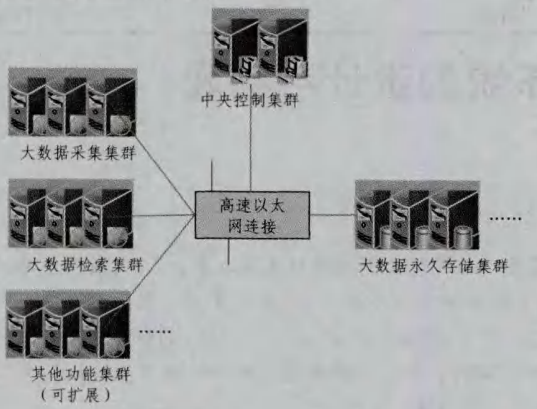

图1 DBDMS系统架构

2.1 中央控制集群

它在整个系统的运行过程中起管理控制作用，如获取用户检索请求并实施检索过程，实时监控各部分的状态并处理异常情况，触发、取消及更新特定集群的任务，优化分配网络连接资源等，目的是保证整个系统稳定、高效、持续、健壮地运行。

2.2 大数据采集集群

它相当于整个系统的内部入口，以进程为执行单元，在多台机器上同时开启多个并发的数据采集功能模块，以提高整个系统的数据采集效率；同时在每台机器上开启缓存，在中央控制集群的协调下，周期性地将缓存数据写入永久存储集群以永久保存。

2.3 大数据检索集群

它相当于整个系统和用户的交互接口，通过系统自定义的查询语句或命令，向中央控制集群提交查询请求。中央控制集群根据整个系统以及各部分的状态信息查找索引快，通知永久存储集群进行查询操作，并将查询结果汇总后返回给数据检索集群，通过视图及界面的方式呈现给用户查看。

2.4 大数据永久存储集群

它是整个系统的数据“仓库”，永久保存所有时期的数据，并通过数据采集集群的周期性写入同步周期性更新，采用数据集方式存储，更有利于分类整理海量数据，并提高查询效率。

2.5 其他功能集群

它为DBDMS预留的一些可编程接口，方便以后根据新的用户需求向整个系统增加新的功能集群。

3 DBDMS系统的数据结构与核心算法

3.1 DBDMS系统的数据结构

DBDMS使用表(Table)作为主要数据结构组织数据，以记录为数据存储单元，每个记录包含多个字段，每张表都包含一个特殊的“描述文件”来管理整张表的信息，如表结构、表数据类型等，此文件分类存储在中央控制集群上，当用户提交查询请求时，中央控制集群会根据表描述文件处理查询。

针对每张表，DBDMS通过改进标准SQL语言，实现了一种专用数据查询与分析语言D-SQL[10]，其语法保留了标准SQL的格式，具体例子如下所示：

create table tb1：创建一张表

drop table tb1：删除一张表

select * from tb1 where name =“DBDMS”：查询符合特定条件的记录

insert into tb1 values (DBDMS)：插入一条记录

delete from tb1 where name=“DBDMS”：删除一条记录

update tb1 set name=“DBDMS” where id=“1”：更新一条记录

在大数据永久存储集群上，数据以列的方式组织，所有字段按照字典顺序来排序，并根据不同类型保存到不同的位置，当达到一定容量时开始以文件为单位保存，这个文件称为数据块(Data Block)，这里的“块”就作为数据采集、检索和存储的一个基本单位或称最小单位，每个块还拥有一个“块索引”，用来存储块内数据信息。DBDMS使用时间戳为分块方式，建立基于时间戳的B树存储结构对数据块进行分类管理，同时将块索引存储在中央控制集群上，以进一步提高加速数据采集和查询的效率。

3.2 DBDMS系统的核心算法

3.2.1 系统数据查询算法

Step1 用户提交检索请求给数据检索集群。

Step2 中央控制集群获取数据检索集群发来的用户请求信息，尝试迅速定位目标集群。

Step3 中央控制集群首先在自身的块索引缓存中查找，如果有目标集群块索引信息，直接向该集群发送查询命令；如果没有，则封装查询命令和目标集群信息，广播至数据存储永久存储集群。

Step4 数据永久存储集群根据查询条件检索自身数据库，如果找到则返回结果给中央控制集群，否则继续广播，直到找到结果或返回查询失败消息。

Step5 中央控制集群将查询结果返回给数据检索集群，进而返回给用户，算法的具体流程如图2所示。

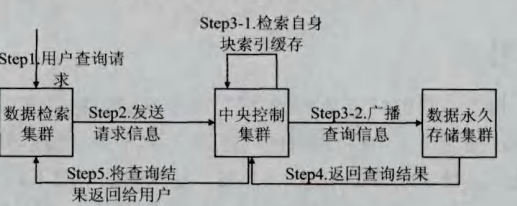

图2 系统数据查询算法

3.2.2 块索引查询算法

中央控制集群在接收到用户查询请求后，会解析并重新生成优化的查询条件，再根据优化过的并发查询算法，在数据永久存储集群中检索目标数据。优化过的并发查询算法如图3所示。

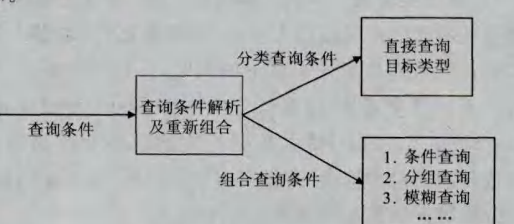

图3 块索引查询算法

在查询条件解析及重新组合模块中，用户的查询请求被解析并重新组合成如下两种查询条件：

Type1　如果包含块索引分类信息，则直接找到缓存中相应的块索引，并迅速发送查询信息给数据永久存储集群，监听并等待返回目标数据。

Type2　如果不包含块索引分类信息，则常规情况下可将查询条件分为：条件查询，如包含关系运算或逻辑运算；分组查询，如包含按某一特定元素分组或某种聚合函数计算结果分组；模糊查询，如根据某一特定元素的某种特殊关联等。此处提供可编程接口以便今后支持更多种类的查询条件。

块索引查询可以采用并发算法，利用并发算法的速度优势可大大提高查询效率。这里不再赘述。

4　DBDMS 系统实验结果及分析

为了测试 DBDMS 系统的实际性能，我们选取某电商网站域名访问日志作为被测对象，检索不同时间段、不同数量条件下的日志记录，具体实验环境的部署如表 1 所列。

表 1　DBDMS 测试环境及配置

环境	配置
中央控制集群	Intel Xeon E5450 3.0G，16GB 内存 ×1
数据采集集群	Intel Xeon E5440 2.83G，16GB 内存，1TB 硬盘 ×2
数据检索集群	Intel Xeon E5440 2.83G，16GB 内存 ×1
数据存储集群	Intel Xeon E5440 2.83G，16GB 内存，10TB 硬盘 ×4
以太网连接	1000Base-T

DBDMS 持续运行近两个月，每天经由数据采集集群进入存储集群的日志记录大约有 5 亿条，共保存日志记录 300 多亿条，占用数据永久存储集群的硬盘空间大约为 20TB。

图 4 给出了一段时间内的数据检索效率，这里以 2014.03.01 — 2014.03.02 一天之内的数据为例，共存储了 530386937 条日志记录，检索域名为 http://www.xxx.com，检索效率与时间段的关系基本呈线性增长，在 200s 左右返回检索结果，可见 DBDMS 的检索效率远高于传统关系型数据库。

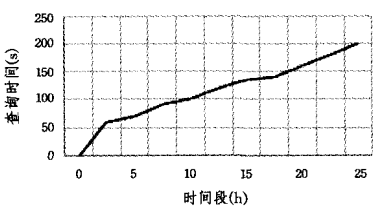

图 4　时间段与检索效率的关系

图 5 给出了检索结果集合与检索效率的关系，继续前面的测试条件，可见结果集合数量越多，检索时间越长，相应的块索引查找越复杂，也越费时。

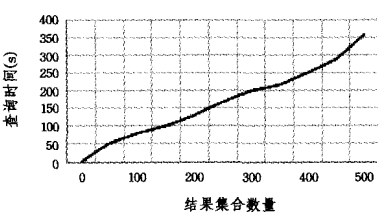

图 5　结果集合与检索效率的关系

图 6 给出了检索条件组合与检索效率的关系，继续前面的测试条件，可见随着检索条件组合越来越复杂，检索效率呈现轻微的增长趋势，但幅度并不大。

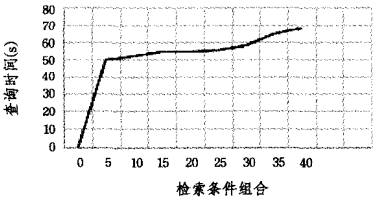

图 6　检索效率与检索条件组合的关系

综上实验结果可见，DBDMS 在处理大数据的检索时有其独特的优势，在不同的时间段、结果集合数量和检索条件组合等情况下，DBDMS 的检索效率均优于传统关系型数据库，根本原因是其采用了块索引加多线程检索机制，外加中央处理集群的控制作用。不过在结果集合数量过大时，检索效率明显降低，这将是今后进一步改进 DBDMS 系统的一个工作方向。此外，如何使 DBDMS 支持更多类型的检索模式或者现有模式是否为相对最优，有待于进一步研究。

结束语　本文针对传统关系型数据库在处理大数据时的局限和缺陷，设计并实现一种基于 Hadoop 和 NoSQL 的新型分布式大数据管理系统 DBDMS，提供大数据的实时采集、检索以及永久存储等功能。DBDMS 适用于海量数据处理的场景。大数据作为当前最热门的技术之一，我们对它的研究会随着认识和技术的长期积累而日趋深入。而接下来在 DBDMS 上的工作包括：如何提高数据采集集群的效率，如何优化块索引算法，以及如何提高更多检索条件下整个系统的效率等。

参 考 文 献

[1] Bari N, Mani G, Berkovich S. Internet of Things as a Methodological Concept[C]// Fourth International Conference on Computing for Geospatial Research and Application. 2013: 48-50

[2] Dikaiakos M D, Pallis G, Katsaros D. Cloud Computing: Distributed Internet Computing for IT and Scientific Research[C]// IEEE Internet Computing. 2009: 10-13

[3] Song Juan, Tang Shou-lian. Operator's Mobile Internet Strategy in the process of Converged Network[C]// 2010 International Conference Management and Service Science (MASS). 2010: 1-4

[4] Wu Yu-lin, Gong Guang-hong. A Fully Distributed Collection Technology for Mass Simulation Data[C]// 2013 Fifth International Conference Computational and Information Sciences (ICCIS). 2013: 1679-1683

[5] Ringel D M, Skiera B. Understanding Competition using Big Consumer Search Data[C]// 2014 47th Hawaii International Conferences, System Sciences (HICSS). 2014: 3129-3138

[6] Membrey P, Chan K C C, Demchenko Y. A Disk Based Stream Oriented Approach For Storing Big Data[C]// 2013 International Conference Collaboration Technologies and Systems (CTS). 2013: 56-64

[7] Han J, Ishii M, Makino H. A Hadoop Performance Model for Multi-Rack Clusters[C]// 2013 5th International Conference Computer Science and Information Technology (CSIT). 2013: 265-274

[8] He Chen, Weitzel D, Swanson D, et al. HOG: Distributed Hadoop MapReduce on the Grid[C]// 2012 SC Companion High Performance Computing, Networking, Storage and Analysis (SCC). 2012: 1276-1283

[9] von der Weth C, Datta A. Multiterm Keyword Search in NoSQL Systems[C]// Digital Object Identifier. 2012: 34-42

[10] Kaur K, Rani R. Modeling and Querying Data in NoSQL Databases[C]// 2013 IEEE International Conference Big Data. 2013: 1-7