Computer Science and Application 计算机科学与应用, 2021, 11(5), 1503-1511
Published Online May 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.115154

Hans汉斯

基于Hadoop的产品大数据分布式存储优化

王耐东，王雅君*，张昕晨，程胜明

大连工业大学机械工程与自动化学院，辽宁 大连

Email: 19042085201329@xy.dlpu.edu.cn, *wangyj@dlpu.edu.cn, 20042095136471@xy.dlpu.edu.cn, 20042085500444@xy.dlpu.edu.cn

收稿日期：2021年4月25日；录用日期：2021年5月20日；发布日期：2021年5月27日

摘　要

研究产品相关大数据资源组织存储与检索查询技术，提出在Hadoop平台基础上对产品大数据资源进行分块存储。基于MapReduce并行架构模型，提出多副本一致性Hash数据存储算法，算法充分考虑了数据的相关性和时空属性，并优化了Hadoop平台的数据划分策略和数据块规格调整。通过对数据的优化存储布局，采用多源并行连接检索方法和多通道数据融合特征提取技术实现产品大数据信息检索，提高了数据资源管理效率。实验表明和标准Hadoop方案比较，多源并行连接数据检索的执行时间为其31.9%。

关键词

产品大数据，Hadoop平台，数据存储优化，数据检索

Storage Optimization of Product Big Data Based on Hadoop Platform

Naidong Wang, Yajun Wang*, Xinchen Zhang, Shengming Cheng

School of Mechanical Engineering, Dalian Polytechnic University, Dalian Liaoning

Email: 19042085201329@xy.dlpu.edu.cn, *wangyj@dlpu.edu.cn, 20042095136471@xy.dlpu.edu.cn, 20042085500444@xy.dlpu.edu.cn

Received: Apr. 25th, 2021; accepted: May 20th, 2021; published: May 27th, 2021

Abstract

A blocking storage layout optimization method based on Hadoop was proposed. A multi-copy consistency hash algorithm based on data correlation and spatial and temporal properties was used.

*通讯作者。

文章引用: 王耐东, 王雅君, 张昕晨, 程胜明. 基于 Hadoop 的产品大数据分布式存储优化[J]. 计算机科学与应用, 2021, 11(5): 1503-1511. DOI: 10.12677/csa.2021.115154

Data distribution strategy and block size adjustment were studied based on Hadoop. A multi-data source map join query algorithm and a multichannel data fusion feature extraction algorithm based on data-optimised storage were designed for the big data resources of products according to the MapReduce parallel framework. Practical verifications show that the execution time of multi-data source parallel retrieval was only 31.9% of the time of the standard Hadoop scheme.

Keywords

Product Big Data, Hadoop Platform, Data Storage Optimization, Data Retrieval

1. 引言

计算机的发展和网络通信技术日趋成熟，数据规模的增长在给人们带来便利生活的同时也让从大量数据中汲取有用信息变得困难，如何从中检索到有用数据是目前需要面对的重要问题[1] [2] [3]。其中有产品相关的数据资源包含生产车间监测视频图像及产品相关数据及文档、物料跟踪数据、加工数据、生产流通数据等，其存在着数据资源规模大，种类多，来源不同且分散分布的特点[4] [5] [6]。传统的分布式数据库受数据库存储能力限制，存在着架构存储能力有限，对数据的管理与发布支持相对较弱，管理效率低的问题[7] [8] [9]。

目前，针对传统分布式数据库存在的问题，庞书杰[10]提出了一种基于 Hash 的关联规则并行优化算法(HP-AR)，通过对数据库统计频繁项集部分的并行处理结合辅助 Hash 表简化挖掘过程满足了面对大规模数据集时挖掘隐藏关联规则的需求。潘俊辉等学者[11]针对基本算法 Apriori 的改进，提出了一种基于压缩矩阵的优化算法，该算法使用 MapReduce 计算模型对数据库进行分块，之后对数据库的关联规则的挖掘结果进行合并，得出频繁项集。Aisha Siddiqa 等学者[12]为了评估不同存储架构的性能，使用 Brewer 的 CAP 定理比较和分析了现有方法，提出了一种定义明确的大数据存储技术分类法。

本文针对目前存在的数据管理效率低、检索速度慢等问题基于 Hadoop 平台，结合分布式、分层结构的存储优化和并行处理等技术，提出了一种多副本一致性 Hash 数据存储算法，将数据集中的数据按照相关性以及时空属性进行分块处理，提高了数据处理的效率。同时在 Hadoop MapReduce 并行框架的基础上，设计了一种多源并行连接数据检索算法，实验结果表明，同传统 Hadoop 方案相比，多源并行连接数据检索算法的运行速度明显加快。

2. 产品大数据存储优化研究

2.1. 数据存储及数据分布策略

基于数据相关性的多副本一致性 Hash 数据存储算法(Multi-copy Consistency Hash Algorithm Based on Data Correlation, CMCHA)，进行 Hadoop 的数据布局优化，优化技术路线：尽可能集中存储相关联的数据，数据检索和分析时在映射阶段完成主要工作，使由映射端到约减端数据通信负载消耗降低，系统整体数据检索和分析性能得到提高。每种跟踪过程数据的类型和格式不同，可将数据的具体采集位置和时间作为数据检索和分析时的关键字。

通常 Hadoop 平台将数据存储为 3 个副本，一份在本地，一份在同机架内不同节点上，一份在不同机架的某一节点上。为减少整体数据传输带宽消耗和数据读取时间延时，HDFS 让读取应用程序读取距离它最近的副本数据。

存储算法考虑如下 3 方面的相关性：数据采集地点相关性、数据采集时间相关性和自定义数据相关性。利用一致性 Hash 算法，按照采集地点编号对数据副本 1 进行 Hash 映射；按照采集时间戳对数据副本 2 进行 Hash 映射；相关系数作为跟踪过程数据的一个重要属性，按照自定义相关系数对数据副本 3 进行 Hash 映射，实现不同的数据查询和数据分析需求。根据应用系统需要自定义数据相关性，给相关系数赋值，算法设计过程中构建配置流程如图 1 所示的 Hash 环。

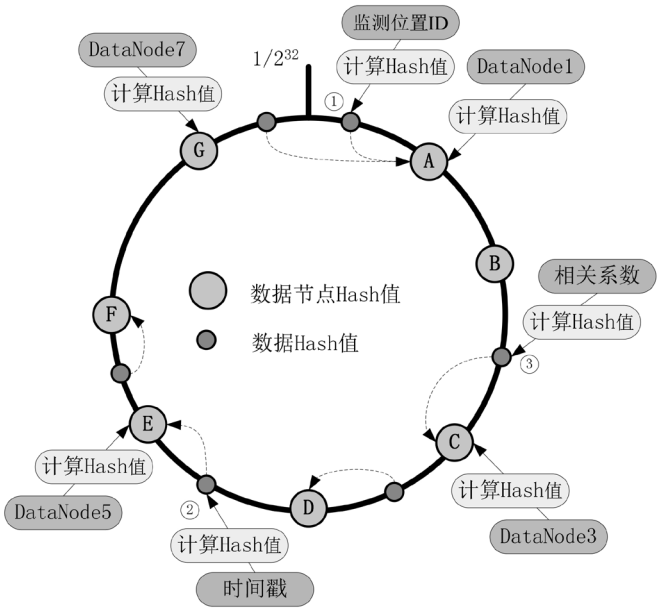

Figure 1. CMCHA flow
图 1. CMCHA 算法流程

步骤 1：通过配置文件预定义跟踪过程数据的相关系数以及冗余的副本数量，定义冗余副本数量为 3；

步骤 2：计算集群中每个数据节点的 Hash 值，配置到 0~2^{32} 的 Hash 环区间上；

步骤 3：基于跟踪过程数据的时间及空间属性和相关系数计算数据的 Hash 值。根据数据来源位置 ID，在云平台下对第 1 份副本数据①，计算 Hash 值 1，映射到 Hash 环上；对第 2 份数据②，根据跟踪过程数据的采集时间戳，计算 Hash 值 2，并映射到 Hash 环上。对第 3 份数据③，根据数据的相关系数计算其 Hash 值 3，并映射到 Hash 环上。可配置大于 3 的副本数量，交替按照这 3 种方式计算其 Hash 值 i，并依次映射到 Hash 环上，满足更高的数据存储可靠性；

步骤 4：确定数据的存储位置，根据数据 Hash 值和数据节点 Hash 值在 CMCHA 算法配置流程图中按顺时针方向将数据映射到距离其最近的节点(如将数据①映射到节点 A 上)；

步骤 5：如果节点空间不足或在映射过程出现异常，则跳过该节点寻找下一个存放节点。

2.2. 数据存储优化研究

按照所属大文件，所有分块数据存储为一个文件，分块数据基于 Hadoop 分布式存储调度策略，被分散存放在不同的分布式存储节点上，每个分块数据设置相应的存储副本率，为便于数据检索该存储策略

另外定义和维护分块数据的索引键名。

每个大文件包含的每个分块数据通过<key, value>记录形式存储到 HDFS 中，记为<Blk-ID, Data>，数据类型为<int, byte[]>，Blk-ID 表示数据分块顺序号，Data 表示数据分块的二进制数据，通过给定的 Blk-ID 可得到对应数据分块的二进制字节数据。大文件数据分块存储方法如图 2。

HDFS 的设计目标是存储大文件，其数据块规格默认为 64 MB，远大于 512B 的物理磁盘的块大小。HDFS 文件访问时间主要包括系统寻址时间和数据传输时间，文件传输效率 η_{effect} 计算公式如下：

$$\eta_{effect}=\frac{t_t}{t_t+t_s}=1-\frac{t_s}{\frac{S_{block}}{v}+t_s} \tag{1}$$

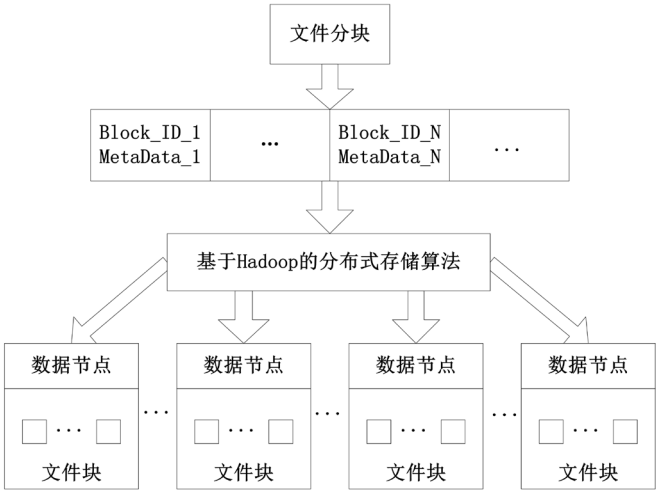

Figure 2. Block storage process of large file
图 2. 大文件分块存储流程

其中，t_t 表示数据传输时间，$t_t=\frac{S_{block}}{v}$；t_s 表示系统寻址时间；S_{block} 表示数据块规格；v 表示数据传输速度。

从(1)可看出 η_{effect} 小于 1。通常在数据分布和索引方法确定情况下，t_s 和 v 是确定的值，要提高 η_{effect} 应增加 S_{block}。在 HDFS 中，通过 dfs.block.size 参数设置数据块 S_{block} 的规格。如果规格设置过大会降低系统负载均衡性，在调整数据块的规格时应综合考虑进入系统的数据规模、数据传输率和负载均衡性。

3. 数据多源并行连接检索

产品数据跟踪管理系统对在线监测的多个监测点以及相关参数进行综合检索，查询条件是监测位置 ID、采样时间或位置和时间联合条件等。检索内容包括位置信息(数据采集点设备名称、设备运行时间、采集位置等)、环境信息(生产车间的温度、湿度、气压等)、生产数据(捕捞时间、捕捞批次、数量等)等多源数据，需要将不同来源的数据进行数据连接。如在产品加工过程质量控制参数的综合检索过程中需要连接 3 个数据文件：1) 加工过程数据文件(表 1)，其中采样批次即为产品批次码；2) 质量控制参数检测数据文件(表 2)；3) 检测环境文件(表 3)，其中检测位置编码代表“车间－工段－班组－工位”。按时间进行的综合查询生成在 2020 年 3 月 14 日 9:00~9:20 的综合检测结果数据，形成质量检测数据列表，包括位置信息和环境信息。此过程需要将 3 个数据文件按照查询条件进行连接，形成满足综合查询要求的查询结果数据列表，如表 4。

Table 1. Processing data file
表 1. 加工过程数据文件

位置 ID	采集时间	采样批次
DL082	2020-03-14 9:08	202003140103100
DL083	2020-03-14 9:18	202003140103300
DL081	2020-03-14 9:10	202003140103200

Table 2. Quality inspection data file
表 2. 质量参数检测数据文件

位置	采集时间	采样批次	采样信息
DL083	2020-03-14 9:10	202003140103200	31.9/7.58/50.2
DL081	2020-03-14 9:00	202003140103300	32.5/7.55/50.5
DL082	2020-03-14 9:08	202003140103100	32.2/7.57/50.4
DL082	2020-03-17 9:00	202003140103100	11.7/10.1/62.2

Table 3. Detection position data file
表 3. 检测环境数据文件

位置 ID	检测位置	采集时间	温度℃	湿度
DL081	ZZ01-I-02-B	2017-03-14 8:00	39	65
DL082	ZZ01-I-02-A1	2017-03-14 8:10	38	63
DL083	ZZ02-II-01-A1	2017-03-14 8:11	38	59

Table 4. Results of date join
表 4. 数据连接结果

位置 ID	检测位置	采集时间	温度/℃	湿度	采样批次	采样信息
DL081	ZZ01-A-026-B	2017-03-14 9:00	39	65	201703140103300	32.5/7.55/50.5
DL082	ZZ01-A-026-A1	2017-03-14 9:08	38	63	201703140103100	32.2/7.57/50.4
DL083	ZZ02-B-017-A1	2017-03-14 9:10	38	59	201703140103200	31.9/7.58/50.2

按照数据检索需求和数据格式描述，设计并行过滤连接检索算法，算法在映射端执行，设计的主要依据是为节省网络流量传输，提高检索效率，过滤和连接在映射过程进行，避免要执行的检索操作在约减过程进行。为使数据连接时所需数据聚集到同一个数据节点，采用基于数据相关性的多副本一致性 Hash 算法进行数据分布。算法流程：1) 根据检索条件过滤掉不符合检索条件的数据；2) 根据连接检索需求，确定数据连接的组键(group key)：检测位置 ID、时间戳或相关系数；3) 用数据文件名作为标签，标记各数据源的各个记录；4) 将相同属性值的记录根据连接组键划分到一组，按照检索条件进行数据连接。

数据进行优化存储分布之后进入数据连接映射阶段，此阶段在本地节点进行相应任务操作，结果传输到 HDFS，数据的优化分布及映射端连接模式流程如图 3 所示。

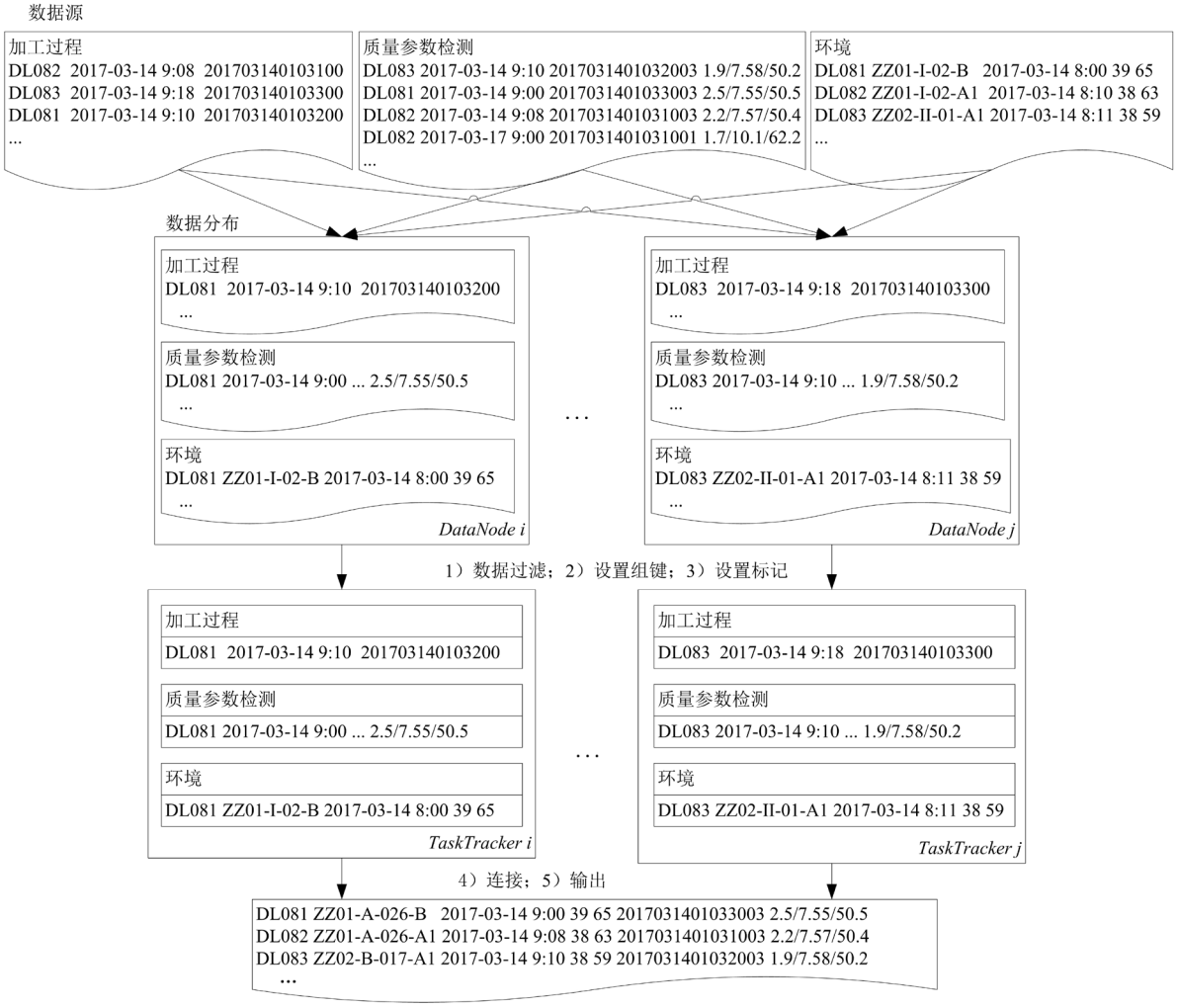

Figure 3. Optimized data distribution and map data join mode
图 3. 数据的优化分布及映射端数据连接模式

4. 算例验证

4.1. Hadoop 平台建设

采用 10 节点即 10 台服务器建设 Hadoop 集群平台，指定集群中一个节点为 NameNode，指定另一台不同的节点为 JobTracker，均是主控节点。余下节点为客户端，作为 DataNode 也作为 TaskTracker。操作系统采用 Windows；部署：虚拟机软件 Vmvare；Vmvare 安装好一台 Windows 虚拟机后，导出或者克隆出另外两台虚拟机，连接为桥连，确保虚拟机和主机 ip 地址在同一个 ip 段内，可以相互通信。设置数据块规格为 64 MB，对应 4 个 CPU 内核，各计算节点都分配 4 个任务网格，其中 3 个为映射计算任务网格，1 个为约减计算任务网格。对集群的整体数据传输性能进行基准测试。

4.2. 算法性能验证

为测试数据存储分布优化后多源连接检索查询算法的性能，将前述针对产品大数据连接算法和基于标准 Hadoop 平台的连接算法进行分析比对验证。分析使用实验室研发的“产品大数据追溯系统”中采集存储的数据集，如表 5。

Table 5. Real data set for join query
表 5. 算法验证真实数据集

文件名	副本数	文件大小	占用空间	记录数
加工过程	3	627 kB	1881 kB	1910
质量检测	3	370 GB	1110 GB	13.62 M
检测环境	3	215 MB	645 MB	4175

1) 多源并行连接检索运行时间变化趋势

选择 3 种典型连接查询条件进行基于 CMCHA 多源并行连接检索查询算法的运行测试，记录每种条件下算法的运行时间。查询结构化语言 SQL 语句的描述如表 6。

Table 6. SQL description of join query
表 6. 连接查询实验类 SQL 描述

类型	条件	查询语句
全连接	不设置	Select 位置 ID，检测位置，采集时间，温度，湿度，采样批次，采样信息 From 加工过程，质量参数检测，检测环境 Where 加工过程，位置 ID = 质量参数检测，位置 ID = 检测环境，位置 ID
位置条件连接	检测工位	Select 位置 ID，检测位置，采集时间，温度，湿度，采样批次，采样信息 From 加工过程，质量参数检测，检测环境 Where 加工过程，位置 ID = 质量参数检测，位置 ID = 检测环境，位置 ID and 位置 ID between [ID1, IDn]
时间条件连接	时间	Select 位置 ID，检测位置，采集时间，温度，湿度，采样批次，采样信息 From 加工过程，质量参数检测，检测环境 Where 加工过程，位置 ID = 质量参数检测，位置 ID = 检测环境，位置 ID and 采集时间 between [T1, Tn]

实验过程中在数据集中选取不同规模的子集，从 10 万条记录递增至数据全集(13.76 M 条)，基于 CMCHA 的多数据源并行连接检索算法运行时间变化趋势及运行时间与数据规模的关系如图 4。可以看出，应用了 CMCHA 数据存储算法优化后，数据检索运行时间随着数据规模的增长而增长平缓。由于对数据存储布局采用 CMCHA 进行了优化，且在映射过程中完成综合检索查询操作，网络通信量有效降低，保证了查询性能的稳定性。

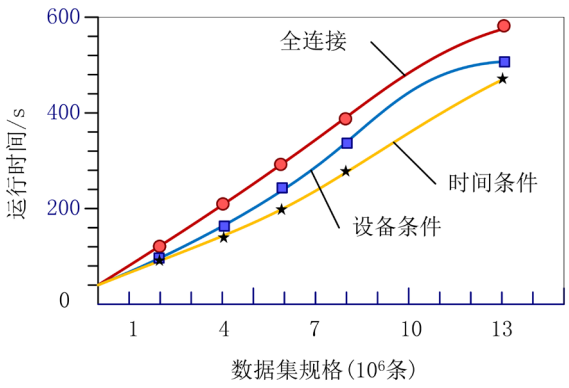

Figure 4. Execution time and variation trend of data join
图 4. 多源连接检索运行时间变化趋势

2) 数据连接检索运行时间比较

使用基于标准 Hadoop 平台的约减端连接检索处理算法和基于 CMCHA 的多源并行数据连接检索算法，针对选取的 13.76 M 条样本数据全集，分别执行全连接、以检测位置为查询条件和以时间为查询条件的连接检索操作，运行时间比较结果如图 5，后一算法的运行时间分别为前一算法运行时间的 32.9%、32.5%和 32.1%。CMCHA 算法在运行时间上远小于标准 Hadoop 算法，而且随着事务条数的增加，虽然 CMCHA 算法运行时间也在增加，但是两者的差距也在逐渐变大，当数据量逐渐越大时，CMCHA 算法的优势也越来越明显。数据存储优化布局后提高了多数据源相关数据聚集性，映射任务中的数据连接在本地就能完成，减少了映射端到约减端的数据通信，也降低了约减任务的启动对性能的影响，所以算法的运行效率明显提高。

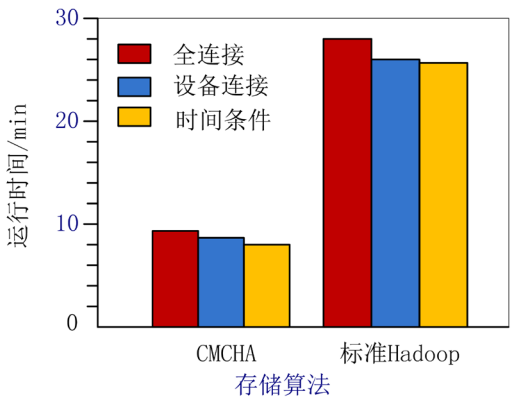

Figure 5. Execution time comparison of data join based on 2 algorithms
图 5. 两种算法多源连接运行时间比较

5. 结论

针对产品大数据资源，基于 Hadoop 平台，采用分布式、分层结构的存储优化和并行处理等技术，提出了多副本一致性 Hash 数据存储算法，按照产品主属性、相关系数和时间戳，在数据集群中按照规则聚集具有相关性的数据，提高数据处理效率。基于该算法设计了 Hadoop 平台下多源并行连接数据检索算法，测试证明通过数据的存储分布优化，算例的运行速度明显加快，和标准 Hadoop 方案比较，多源并行连接数据检索的执行时间为其 31.9%。

基金项目

辽宁省科学技术计划项目(20170540066)。

参考文献

[1] 王磊. HDFS 文件系统升级方案的研究与实现[D]: [硕士学位论文]. 南京: 南京邮电大学, 2020.

[2] 张国华, 叶苗, 王自然, 周婷婷. 大数据 Hadoop 框架核心技术对比与实现[J]. 实验室研究与探索, 2021, 40(2): 145-148+176.

[3] 王艳, 蒋义然, 刘永立. 基于 Hadoop 的大数据处理技术及发展[J]. 信息记录材料, 2020, 21(11): 146-147.

[4] 李善青, 郑彦宁, 赵辉, 等. 大数据背景下科学元数据的重要问题研究[J]. 科技管理研究, 2019, 18(1): 184-188.

[5] 李联辉, 尹冠飞, 莫蓉. 面向航空发动机装配过程的信息追溯与过程监控[J]. 计算机集成制造系统, 2018, 22(12): 2986-3000.

[6] 李青, 冯丹, 梅正朋. 飞机使用寿命周期构型管理和追溯[J]. 计算机集成制造系统, 2016, 22(2): 476-481.

[7] 王瑶. 基于 Hadoop 框架的工业物联网实验平台构建与实现[D]: [硕士学位论文]. 西安: 西安石油大学, 2020.

[8] 王丹, 张祥合, 赵浩宇. 基于元数据的信息知识组织智能检索系统设计[J]. 情报科学, 2019, 18(9): 113-116, 958.

[9] 孙文金. 基于 Hadoop 的文件存取优化的方法研究[D]: [硕士学位论文]. 沈阳: 沈阳工业大学, 2020.

[10] 庞书杰. 关联规则并行优化算法及其应用研究[D]: [硕士学位论文]. 烟台: 烟台大学, 2020.

[11] 潘俊辉, 王辉, 张强, 王浩畅. 一种在 MapReduce 下实现的 KNN 改进算法[J]. 重庆科技学院学报(自然科学版), 2021, 23(1): 70-72+95.

[12] Siddiqa, A., Karim, A. and Gani, A. (2017) Big Data Storage Technologies: A Survey. *Frontiers of Information Technology & Electronic Engineering*, **8**, 1041-1072. https://doi.org/10.1631/FITEE.1500441