

基于 Hadoop 平台的地铁 NCC 数据中心方案研究

朱东升^{1,2}, 徐石明^{1,2}, 李天阳^{1,2}, 叶剑斌^{1,2}, 王玉祥³

(1. 南瑞集团公司(国网电力科学研究院), 南京 210003; 2. 国电南瑞科技股份有限公司, 南京 210061;

3. 江苏祥华科技有限公司, 东台 224200)

摘要: 线网指挥中心是地铁管理控制中的承上启下的重要环节, 接入线网各线路的控制中心, 实时的监测线网整体的运营状况; 并通过与上级政府交通管理部门系统的对接, 将相关数据提交至交通管理系统, 配合交通管理部门的工作; 数据中心是线网指挥中心的环节, 存储线网所有线路的生产数据和业务数据; 目前线网指挥中心的数据中心都是基于传统的数据仓库的架构来创建的, 而随着线路的不断增多, 数据的种类和数量都在不断增加, 传统数仓暴露出成本高, 扩容难, 维护困难的缺点; 该方案提出的基于 Hadoop 大数据平台的数据中心建设方案, 极大地降低了使用和维护成本, Hadoop 生态圈包含了各类的组件模块, 能够解决建设和使用中遇到的各类问题, 提高了数据中心的性能。

关键词: 大数据; Hadoop; 线网指挥中心

Research of Hadoop Platform for Metro Data Center Construction

Zhu Dongsheng^{1,2}, Xu Shiming^{1,2}, Li Tianyang^{1,2}, Ye Jianbin^{1,2}, Wang Yuxiang³

(1. NARI Group Corporation (State Grid Electric Power Research Institute), Nanjing 210003, China;

2. NARI Technology Co., Ltd., Nanjing 210061, China;

3. Jiangsu Xiang Hua Technology Co., Ltd., Dongtai 224200, China)

Abstract: The network control center, which access to the line operation control center, plays an important role in the metro management and control by the way of real time monitoring the overall operation status of the metro network. And also submit the relevant data to the traffic management system so as to coordinate with the work of the government traffic management department. The data center, which stores the overall network production data and business data, is the core of the network control center. At present, the data center of the network control center constructed based on the traditional data warehouse architecture. With the number of the line and data increasing, the traditional warehouse exposed high cost, difficult expansion and maintenance and other shortcomings. This paper proposes the data center construction scheme based on the Hadoop big data platform, which greatly reduces the use and maintenance costs. The Hadoop ecosystem contains various modules, can solve all kinds of problems encountered in the construction and use of the data center, as the result, improves the performance of data center.

Keywords: bigdata; Hadoop; network control center

0 引言

随着国内地铁投资规模的不断扩大, 各大城市的地铁规模已逐步形成网络化的趋势, 新的业务需求也随之产生, 地铁运营管理也日益复杂, 建设线网层级的指挥中心已成为各大地铁城市的迫切需求。北京是国内最早建设线网指挥中心的城市, 线网指挥中心负责协调各运营主体, 具有监视, 运行协调, 应急指挥, 信息共享的功能, 但由于定位和功能不够清晰, 系统功能也有不成熟的地方, 后续北京又进行了二期工程建设, 进行了系统扩容, 其中基于 teradata 的数据仓库建设的数据中心, 是数据仓库在国内地铁线网指挥中心的的首次应用。广州, 深圳也进行了线网指挥中心工程的建设, 两者都利用了数据仓库建设了线网层级的数据中心, 实现生产管理指标分析和运营评估功能。南京, 成都, 西安等城市的线网指挥中心的建设方

案中包含了数据中心的建设。可见, 数据中心已经成为各大地铁城市建设线网指挥中心的重要环节。但由于数据仓库价格昂贵, 数据容量限制等因素, 地铁建设与运营方也在寻找一种新的价格低廉, 扩容方便, 性能稳定的技术代替数据仓库。本文介绍一种基于 Hadoop 大数据平台的数据中心建设方案, 实现了如计算、分析、展示及存储等基础服务和相应的指标查询, 客流分析等专业服务。

1 线网指挥中心数据流及原理

本方案中, 线网指挥中心由数据源, 数据接口平台, ETL 统一管理平台, 数据平台, 应用层组成。数据源系统的数据经过数据接口平台汇总, ETL 工具将数据清洗, 并转成统一的模型存储到数据平台层, 数据平台层根据应用层的需求建立数据集市, 并通过 API, SQL 等多种方式提供数据给应用层。线网指挥中心数据流如图 1 所示。

数据接口平台采集 PSCADA, BAS, FAS, PSD, ATS 等专业的数据, 通过 ETL 平台按照业务类型存储到实时数据库和数据中心。实时数据库采用 Redis 等内存数据库, 实时的存

收稿日期: 2017-05-17; 修回日期: 2017-06-07。

作者简介: 朱东升(1987-), 男, 硕士, 工程师, 主要从事电力系统及其自动化, 轨道交通电气化及自动化, WEB 前后台框架方向的研究。

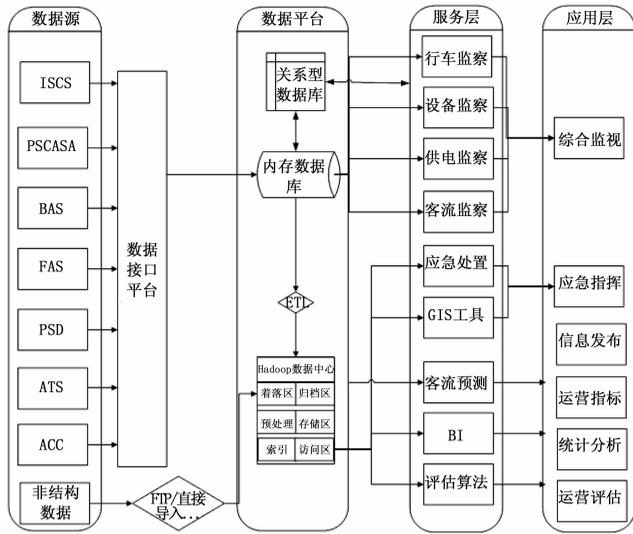


图 1 线网指挥中心数据流

储数据源系统的当前断面数据。数据中心是基于 Hadoop 大数据平台的架构建造的, 是地铁线网指挥中心的核心部分。数据中心存储线网海量的设备状态, 行车状态及客流等生产数据和相关办公系统产生的业务数据, 提供并行的计算和非结构化数据的处理能力, 实现低成本的存储和 low 延迟、高并发的查询能力, 并通过对数据的深入挖掘和分析, 建成相应的数据集市, 统一地对外部应用提供数据服务。应用层中监察类的应用如行车监察, 设备监察, 供电监察, 客流监察等功能的数据来源于实时数据库, 统计指标类如运营指标, 统计分析, 运营评估等应用的数据来源于数据中心。

2 数据中心设计

数据中心是建设地铁 NCC 系统的核心, 是相关业务系统的数据来源, 必须具有很强的数据采集, 存储, 开发, 管理和分析能力。数据中心主要由实时数据处理模块和历史数据处理模块两部分组成: 实时类数据处理模块采集, 计算, 存储设备的当前状态数据, 并根据应用层的要求提高各类接口服务供外部应用调用。历史数据处理模块存储, 分析设备的历史状态信息, 按照对应的指标体系为指标分析提供服务, 针对历史数据处理模块存储的未经处理的设备状态数据, 通过数据挖掘分析工具, 预测设备的健康状态, 为维保人员的设备预防性维修提供依据, 提高设备使用率和运营安全性。

2.1 实时数据处理

实时数据包括设备的实时状态数据, 各类报警数据等, 主要为实时监控类系统提供数据服务。按照功能模块划分, 实时数据经过“数据采集→数据接入→数据分析计算→数据输出/存储”几个环节, 提供给应用层使用。具体功能模块采用如图 2 组件实现。

实时数据通过 http 的方式发送给 flume 集群, flume 集群中配置 http source 获取实时数据, 并通过两级高可用的 flume 集群将数据转发给 kafka

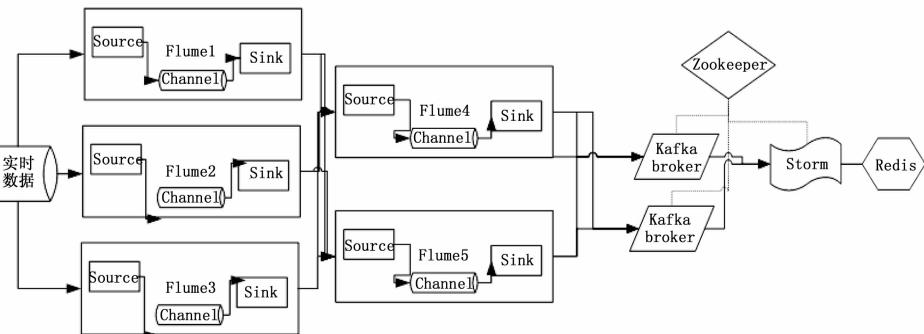


图 2 实时数据流

集群中, 解决了采集与计算速度不一致的问题。Storm 获取 kafka 传递过来的数据并分发给 storm 计算节点计算, 最终结果存放在 redis 中供外部应用程序使用。

2.1.1 实时数据采集

NCC 数据中心实时数据采集模块, 通过配置需获取的实时数据的专业, 类型等信息, 被动的接受数据源系统的数据, 数据源系统一旦设备状态数据发生变化, 立即发送给 NCC 数据中心实时数据处理模块, 保证了实时数据的及时性。数据采集模块负责从数据源系统采集数据, 使用 Flume 实现。Flume 是一种分布式, 可靠且高可用的海量日志采集, 聚合和传输系统。Flume 灵活简易的架构是基于处理流数据而创建的, 具有容错可调的可靠性和故障转移和恢复的鲁棒性。Flume 使用一个简单的可扩展数据模型, 允许应用程序在线分析。

2.1.2 实时数据接入

数据采集模块使用 kafka 消息中间件来显示, kafka 接受 flume 发送的数据并输出给 storm, 解决了 flume 数据采集与 storm 数据处理速度不一致的问题。Kafka 是一种分布式的, 基于发布/订阅的消息系统, 提供消息持久化能力, 即使对 TB 级别以上的数据也能保证访问性能。Kafka 在主题中保存消息的信息, 生产者向主题中保存数据, 消费者从主体中读取数据。

2.1.3 实时数据分析计算

数据分析计算模块使用 storm 组件实现, storm 是一个分布式, 容错性好的实时计算系统。支持 Java, ruby, python 等多语言编程并支持扩展。Storm 具有极高的容错性和水平扩展性, 可靠的消息处理机制保证了每个消息都能得到一次完整的处理, 使用的 OMQ 作为其底层消息队列保证了消息能够得到快速处理。

2.1.4 实时数据存储

实时数据主要提供给外部监察类应用使用, NCC 系统中实时数据存储采用 json 格式, 数据示例为“TagName: Value”。其中 TagName 包含设备的专业, 站点, 点号等信息。Value 中存储了设备的值 and 状态信息。根据数据的格式和特点, 实时数据采用 redis 这类 key-value 存储系统来实现。Redis 是一种高性能的 key-value 内存数据库, 提供了 java, c++, python 等客户端和编程接口, 方便系统开发, 扩大系统的兼容性。

2.2 历史数据处理

历史数据包含线网所有设备的历史状态数据, 客流数据

等，是数据中心的核心。历史数据经过预处理，建索引，按照主题存储，并通过指标体系的建立，业务主题分析和统一的业务视图，最终为外部应用提供数据服务。地铁运营需根据数据中心产生的统计分析报表获取线网地铁运营状况，客流分布，并为地铁运行时时刻表的编制提供理论依据。后续利用 BI 工具及复杂的算法，通过对数据的深入挖掘，改善地铁运营的现状，实现运营，维保的系统管理。历史数据处理流程及所用组件如图 3 所示。

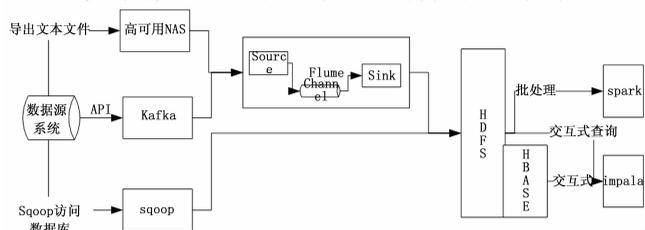


图 3 历史数据流

2.2.1 历史数据采集

本方案实现了三种数据采集方式，包括导出文本文件的方式，运行 T+1 的定时任务将数据源系统数据以文本文件的形式导出到高可用网络附属存储设备中，flume 采取批量的方式将数据写入到 hdfs 中；API 直接写入的方式，通过调用 kafka api 把数据写入到 kafka，继而经过 flume 数据存储到 hdfs 中；sqoop 定时访问数据源系统数据库批量及增量导出数据到 Hadoop 平台中。三种采集方式的配合使用，最大程度的兼容了不同的数据源系统，发挥了数据中心保存线网相应系统数据的功能，为了后续的数据挖局和分析提供了基础。

2.2.2 历史数据存储

按照数据功能划分，Hadoop 平台数据存储分为数据着落区，数据预处理区，索引区，数据存储区，数据归档区，外部数据访问区六个功能区。数据着落区存储原始数据，保存时间在一周左右；数据预处理区保存时间若干天，主要为统计和预处理的数据，建立 Hive/Impala 视图表；索引区建立数据索引，保留若干个月，提供 solar 搜索服务；数据存储区存储结构化化和半结构化数据，数据按照业务类型进行 Parquet+snappy 压缩，保留若干年；数据归档区按照业务类型存储永久的原始归档数据；外部数据访问区存储相应的数据库表，数据类型为统计和已处理的结构化数据，为对应的业务系统提供数据服务，同时提供统计报表查询功能。

2.2.3 历史数据计算

Hadoop 平台中提供了 5 种主要的计算框架：基于 mapreduce/spark 的批处理框架，基于 impala 的交互式 sql 查询，基于 spark streaming 的流计算，基于全文检索的 Solr 搜索引擎框架，基于 Hbase/Cassandra 的高并发实时查询。根据地铁线网控制中心的业务需求，本文采用 spark 用作批处理，定时的计算并生成相应的指标数据，spark 适用于批量数据的计算，将中间结果存放于内存当中，具有高效的迭代计算能力，从而进行批处理时更高效；同时应用层需实时查询相关指标，了解全线所有设备和客流的指标数据。本方案采用 impala 实现交互式查询功能，impala 是基于内存的大规模并行处理架构，不依赖于底层的 map/reduce 作业，交互式查询

时具有很短的响应时间，支持主流的 Hadoop 文件格式保证了应用的广泛性。

3 应用模块实现

3.1 实时监控类

实时监控类应用包括行车、供电、设备、客流及应急处置等。其中行车、供电及设备监察应用采用基于 B/S 架构的综合监控系统来实现，前端 JSP 页面嵌入 SVG 矢量图动态的展示设备当前状态，业务数据来源于数据中心的 redis 数据库，模型数据存储于监控系统本身的关系型数据库里；客流监察系统主要监测线网各个站的客流量，前端展示采用 Echart 等开源图标控件实现；应急处置系统提供了各类紧急情况的应急处置预案，当紧急情况发生时，由系统用户根据现象确定启动何种预案，执行记录会通过文字，电脑录频，语音，视频等形式保存，供后续分析研究。以南京地铁为例，本方案实现的线网指挥中心全线列车运行如图 4 所示。

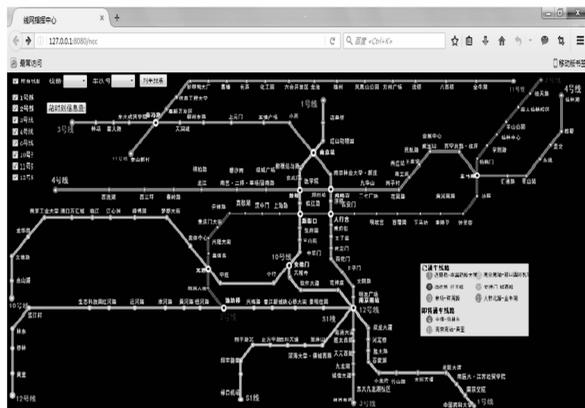


图 4 全线列车位置实时图

每条线路代表了一个 SVG 图层，用户可以通过筛选线路的方式查看全线或者某几条线路的列车运行状态，其中重要安全事故如异常火灾，站点大客流等也会显示在图中。业主管理方根据全线列车运行状态和设备状况，为紧急情况列车运行图的调整及人流疏散提供依据，提高地铁运行的安全性，保障了乘客的安全。

3.2 统计分析类

根据国内不同城市地铁运营的需求以及对轨道行业的分析，总结出客流信息类、清算信息类、票务信息类、列车运行信息类、路网车辆设备故障类、服务类、试运行类、线网指挥中心系统采集类及线网指挥中心设备运行类 9 大类指标清单。以乘客关注的服务类指标中的自动扶梯可靠度和进出站闸机可靠度两项指标为例，需统计一定时间内自动扶梯或进出站闸机实际运行时间，并得出与计划运行时间的比值，根据一定的比例确定设备的健康状态，为后续的设备评估和运维检修提供依据。

统计分析是商业智能的一部分，由于商业智能工具的专业性，本方案采用了 FineBI 商业智能软件，FineBI 是一种轻量级的 BI 工具，部署方便，升级成本低且易于维护。FineBI 基于 hadoop 平台中的表间的关联关系和完整的数据结构进行数据处理和分析，采取 cube 预处理以及并行计算的先进的数据处理模式，采用高效的智能位图索引和缓存机制，保证了前端

数据展示的响应及时。

用户通过网页配置页面, 填入所需的驱动器、URL、数据库用户名和密码及编码等信息。FineBi 同时支持各类主流的包括 mysql, oracle, sqlserver 等关系型数据库, 以及 MongoDB 等 Nosql 型数据库。

4 实验结果与分析

4.1 实验环境

Hadoop 平台创建时已经考虑到脱离对硬件的依赖, 故可搭建在从传统的 PC 机到大型的服务器等各类机器上。为了更贴近 NCC 系统生产环境, 本方案采用如今地铁各类生产系统广泛应用的主流 X86 服务器, 实验环境 Hadoop 平台的部署采用水平部署方式, 采用 X86 服务器搭建四个节点, 其中 Hadoop 集群采用三个节点, 另外一个节点配置 redis 以及关系型数据库。每个节点通过一块千兆网卡与千兆交换机相联接。网络结构如图 5 所示。

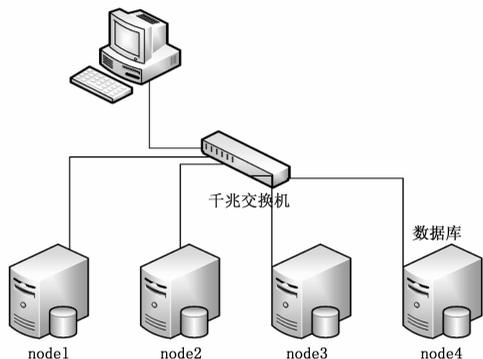


图 5 实验环境

4.2 数据准备

Hadoop 平台包含了如 TestDFSIO, mrbench, nnbench 等测试工具, TestDFSIO 侧重于测试 Hadoop 的 IO 性能, mrbench 侧重于测试任务执行效率, nnbench 侧重于测试主节点的负载性能。这三种工具能够从不同方面测试 Hadoop 的性能, 但无法与传统的数据库以及 MPP 数据仓库的测试结果形成横向对比。结合 NCC 业务类型和数据结构, 本方案采用 TPC-H 基准测试方案, 模拟各类复杂的查询和修改操作, 从而测试结果更能反应 NCC 实际应用中的数据访问性能。数据根据 TPC-H 标准工具 dbgen 生成, 数据表文件大小如表所示。

表 1 数据表文件大小

表	100G		1T	
	记录数	tbl 文件大小	记录数	tbl 文件大小
region	5 条	389Byte	5 条	389Byte
nation	25 条	2.2K	25 条	2.2k
supplier	1000000	137M	1000 万	1.4G
part	20000000	2.3G	2 亿	24G
customer	15000000	2.3G	1.5 亿	24G
partsupp	80000000	12G	8 亿	116G
orders	150000000	17G	15 亿	169G
lineitem	600037891	75G	5999989709	758G

数据采用 Hadoop 提供的 gpload 工具, 通过执行 gpload -f f 导入到 Hadoop 中, 数据导入耗时如表 2 所示。

4.3 结果分析

本实验通过分别对 100 G 和 1T 的地铁设备状态数据执行不同的 sql 语句, 模拟各类的复杂查询和并行修改操作等, 得出每个 sql 语句的执行时间, 并纵向对比 100 G 和 1T 数据下同一 sql 语句的执行效率, 得出随着数据量的增大, Hadoop 在处理效率上并没有得到明显降低的结论, 反应了 Hadoop 处理大数据的能力。测试结果如图 7 所示。

表 2 Hadoop 导入测试数据耗时

表	100G 耗时/s	1T 耗时 /s
region	1.11	1.28s
nation	1.23s	1.36s
supplier	5.39	22.45s
part	52.05	473.32s
customer	36.89s	350.07s
partsupp	105.78	1069.89s
orders	303.71	3046.05s
lineitem	1539.79s	15358.01s
总计	2045.95	20322.43

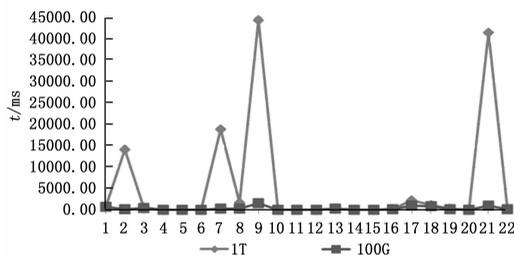


图 7 测试结果

根据测试结果可以得出, 在 100 G 的原始数据情况下, 22 条 sql 语句的执行时间都控制在 3 s 之内, 这对于 NCC 业务系统是可以接受的。1T 的原始数据情况下的测试结果也反映, 除了 4 条 sql 语句, 其它测试语句和 100 G 原始数据的运行时间几乎一致。通过分析 sql 语句 Q2, Q7, Q9, Q21, 以 Q2 语句为例:

```
Select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_comment
```

```
From part, supplier, partsupp, nation, region
where
```

```
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and p_size = 35
```

```
and p_type like '%NICKEL'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'MIDDLE EAST'
and ps_supplycost = (
```

```
select min(ps_supplycost)
from partsupp, supplier, nation, region
where
```

层次的算法优化，虽然比本设计速度占优势，但是硬件资源消耗比本设计多了 48%。相对于其他设计，本设计也取得了面积和速度的较好折中，资源消耗也有明显改善，经济性上更为突出。

表 2 FPGA 设计速度比较

有限域	文献	FPGA 型号	f(Mhz)/t(ms)	面积规模 /LUTS
GF(2 ²³³)	本设计	Xilinx Artix 7	110/2.92	9302
GF(2 ²³³)	文献[12]	Cyclone IV EP4CE115	82/4.50	6000
GF(2 ²³³)	文献[13]	XCV2000E-7	108/1.62	13799
GF(2 ²³³)	文献[14]	EP2S90F1508C	NA/2.007	15280
GF(2 ¹⁶³)	文献[15]	StratixIII-EP3SL340H1152	177.7/1.201	8500

5 结束语

本文通过研究椭圆曲线有限域的算术运算，根据 ECC 的多项式的稀疏特点，提出并行的模乘思想，并改进传统的欧几里得右移求逆算法，使其可同时完成模逆和模除运算，以此为基础设计了有效的点乘电路并进行 FPGA 原型验证。实验结果表明，当数据位宽为 233 位时，只需 2.92 ms 即可完成标量乘运算。相较于其他算法，本设计大大提高了运算速度，资源消耗较少，取得了面积和速度的较好折中，在已公开的文献中性价比较高。尽管本文采用的模多项式是三项式基，但对于其他多项式基同样适用，同时只需修改电路设计输入的参数值即可实现其他不同位宽数据的要求，保证了设计的有效性、通用性和可研究性。

参考文献:

[1] 赖忠喜, 陶东娅, 张占军. GF(2ⁿ) 域椭圆曲线密码体制中快速标量乘算法的研究 [J]. 计算机应用与软件, 2014, 31 (8): 324

(上接第 227 页)

```

p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'MIDDLE EAST'
)
order by s_acctbal desc, n_name, s_name, p_partkey limit 100;

```

Q2 语句包含复杂的如聚焦操作、子查询、排序及多表联合查询等复杂的查询逻辑，不可避免地随着数据量的增大可能会对性能造成影响。但这类复杂查询逻辑一般应用在统计分析业务系统中，根据统计系统的非实时性特点，得出测试结果同样控制在 NCC 可接受的范围中的结论。可见在大部分应用场景的情况下，Hadoop 平台都能完美地解决问题，并在数据急剧扩张的情况下都能保持良好的性能。

5 结语

不同于传统的基于 MPP 数据仓库建立的数据中心，本方案提出的基于 Hadoop 的数据中心建设方案，具有很强的技术先进性，支持结构化，半结构化，非结构化各类数据的存储和

- 326.

[2] 谢天艺. 素数域椭圆曲线密码 SoC 的设计与实现 [D]. 杭州: 浙江大学, 2015.

[3] 赖忠喜, 张占军, 陶东娅. 椭圆曲线底层域快速算法的研究 [J]. 计算机工程与应用, 2014, 50 (3): 67-70.

[4] 赖忠喜, 张占军. 椭圆曲线底层域快速算法的优化 [J]. 计算机工程与应用, 2015, 51 (22): 115-118.

[5] 张莉华, 蔺 莉. 一种安全高效的椭圆曲线密码抗功耗攻击算法 [J]. 测控技术, 2016, 35 (8): 118-121.

[6] 胡诗玮, 徐和根. 有限域 GF(2~(256)) 上椭圆曲线密码算法的硬件设计 [J]. 机电一体化, 2015, 21 (2): 71-75.

[7] 郑建宏, 周 亮. 椭圆曲线加密算法在卫星通信中的应用 [J]. 信息通信, 2016, 31 (02): 209-210.

[8] 石亚娟, 黄辉辉, 陈建华. 基于智能卡的动态身份认证协议 [J]. 电子技术应用, 2015, 41 (3): 97-100.

[9] 尹 恒. ECC 标量乘算法在抗边信道攻击上的应用研究 [D]. 贵阳: 贵州大学, 2015.

[10] 白永祥. 椭圆曲线密码算法在 VPN 安全握手中的应用 [J]. 电子设计工程, 2015, 23 (13): 18-20.

[11] 郭高峰, 崔强强. 基于 GF(2~m) 的椭圆曲线求逆算法的改进研究 [J]. 现代电子技术, 2014, 37 (18): 19-22.

[12] 操志辉. GF(2~m) 域上 ECC 标量乘的设计与 FPGA 实现 [D]. 武汉: 武汉理工大学, 2014.

[13] Urbano-Molano F A, Trujillo-Olaya V, Velasco-Medina J. Design of an elliptic curve crypto processor using optimal normal basis over GF(2233) [A]. 2013 IEEE Fourth Latin American Symposium on Circuits and Systems (LASCAS) [C]. Cusco: IEEE, 2013: 1-4.

[14] 田 伟, 刘爱军, 申卫昌. 基于梳状算法的椭圆曲线密码标量乘改进方案 [J]. 微电子学与计算机, 2015, 32 (5): 99-103.

[15] 王云峰, 王 静, 黄世伟. 一种两路并行调度的点乘算法硬件实现 [J]. 新型工业化, 2014, 4 (7): 20-27.

查询，支持批量，实时各类数据采集方式，在软件层面提高了数据的存储计算可靠性，同等计算能力的情况下，成本比传统的 MPP 数据仓库低廉，解决了使用数据仓库建设数据中心扩容难，成本高，维护困难的缺点。越来越多的科技企业如华为、浪潮、Cloudear 及谷歌等都加入到了 Hadoop 相关产品的研究和开发中，使得 Hadoop 生态圈的不断发展，Hadoop 产品也趋于稳定。国内北京地铁就正实施将以前基于数仓建设的数据中心迁移到 Hadoop 平台中，也证明了 Hadoop 的优势。基于 Hadoop 数据中心的数据挖掘，分析将给地铁运营带来巨大的价值，提高地铁运行的稳定性和运行效率，更好地服务乘客。

参考文献:

[1] 陈达伦, 陈荣国, 谢 炯. 基于 MPP 架构的并行空间数据库原型系统的设计与实现 [J]. 地球信息科学, 2016 (2): 151-159.

[2] 王德文. 基于云计算的电力数据中心基础架构及其关键技术 [J]. 电力系统自动化, 2016 (11): 67-71, 107.

[3] 丁泽柳, 郭得科, 申建伟, 等. 面向云计算的数据中心网络拓扑研究 [J]. 国防科技大学学报, 2011 (6): 1-6.