

河海大学

2017 届本科优秀毕业
设计（论文）摘要汇编

物联网工程 学院（系）

二〇一七年七月

（加盖院章）

时序大数据可视化分析平台研究与设计

(姓名: 陶友贤 学号: 1362810213)

(专业: 计算机科学与技术 指导老师: 牟艳)

摘要: 可视化分析平台对大数据的处理分析越来越重要。本文针对时序大数据可视化分析的需求, 分析出现有的可视化分析工具操作繁琐、重复、耗时, 以及在时序大数据(即随着时间序列变化特征的数据)场景下存储和查询性能下降时的问题。为此, 借助开源的 Zenvisage 可视化分析平台, 提出了基于 Spark 的时序大数据分析方案, 该方案以 HDFS 进行文件存储、Parquet 列存储数据压缩、分布式内存引擎 Spark 为底层计算框架, 并将此方案集成于 Zenvisage 的可视化分析平台底层中, 通过系统测试和对比实验评估本文研究内容的有效性和可用性。经测试证明, 该系统能够满足用户交互式分析的需求, 优化了 Zenvisage 平台在时序大数据场景下的存储和查询性能, 运行稳定、功能完整、可扩展性强。

关键字: 时序大数据; 交互式分析; 可视化; Zenvisage; HDFS; 列存储; Spark

Abstract: Visual analysis platform for the process and analysis on large data is becoming increasingly important. This paper analyzes the demand for visualization analysis of large time series data(ie, data that changes with time series), and some visual analysis tools sounds cumbersome, repetitive, time-consuming, and time-consuming large data (ie, data with time series variation) problem. For this, with the open source Zenvisage visual analysis platform, we propose a large data analysis scheme based on Spark. The program is stored in HDFS for file storage, Parquet columns store data compression, distributed memory engine Spark as the underlying computing framework, which is integrated into the bottom of Zenvisage's visual analysis platform. It proves that the system can meet the users' needs of interactive analysis, optimize the storage and query performance of Zenvisage platform in the scene of large time series data, and it is stable, functional and extensible.

Key words: Large Time Series Data; Interactive Analysis; Visualization; Zenvisage; HDFS; Column-store; Spark

1. 绪论

如今的交互式可视化分析工具, 如 Tableau^[1]、Spotfire^[2], 他们的出现给数据挖掘和数据科学的民主化铺平了道路, 并被广泛使用。通过这些工具, 标准的数据分析方法一般遵循以下操作: 数据分析师首先选择可视化分析工具, 再将数据集加载到工具中, 通过可视化来检查, 分析结果, 然后重复该过程, 直到找到满足期望的视图。

以上现有的交互式分析工具普遍是基于数据层面进行设计和实现的, 表面上看是“一劳永逸”、直观的分析方案, 充分利用了用户自己的偏好设置, 支持用户修改查询条件, 却忽略了在充分提高用户体验时加重了用户的检查、试错工作。为了在数据集中找到所需的期望, 数据人员可能需要密集型地手工检查集合中的每个可视化, 这是一个繁琐、重复、耗时的循环过程^[3]。此外, 对于一些复杂的数据探索, 如: 在单个数据集中找到给定属性的上升趋势

的视图，这很难通过单个 SQL 的编写实现，一定程度上对分析人员的编程能力要求也很高。

针对重复耗时的手工试验和相对复杂的查询这两个对数据本身进行操作时存在的问题，我们借助开源系统 Zenvisage 平台，自动识别相关条件下的可视化期望，一定程度上实现了轻松的数据探索过程^[4]。针对其在时序大数据的应用场景下存储和查询策略的问题，本文设计并实现基于 Spark 的时序大数据分析方案，即通过 HDFS 分布式文件系统实现数据存储以及 Spark SQL 内存计算引擎的查询方案，并根据不同属性数据集设计出适应性读取存储策略；同时，通过 Parquet 列存储技术进一步提高存储和查询性能，有效地满足了不同数据集分析时系统底层高吞吐、高并发的查询分析，以达到稳定集成于 Zenvisage 平台的底层设计的目的。

2. 基于 Spark 的时序大数据分析方案

针对于海量数据的基于 Spark 的时序大数据分析方案，并详细介绍了对 Zenvisage 的优化设计，包括 HDFS 列存储和 Spark SQL 查询策略，以及其中的查询结果缓存优化和其他相关改进部分。

2.1 问题定义与分析

基于 Spark 的时序大数据分析方案中的操作流程可大致描述为：

1. 将样例数据集的字段说明文件 (*.txt) 提交至 HDFS 文件系统上，并对文件内容进行解析；
2. 将样例数据集 (*.csv) 提交至 HDFS 文件系统上，并根据 1 中的内容解析结果对各个数据集进行 Parquet 列存储转化，可以删除源文件；
3. 选定数据集，且 SQL 查询提交后，通过 Spark SQL 对 2 中的 Parquet 列存储文件进行查询，返回查询结果 List；
4. 根据 List 结果集构造可视化。

如下图 1 架构图所示，可以看出此方案在大数据场景下的三大技术点：第一点在于各个数据集的数据模型设计，由数据集文件 (*.csv) 和数据字段说明文件 (*.txt) 组成；第二点在于 Parquet 列存储转化；第三点在于 Spark SQL 读取结果集。

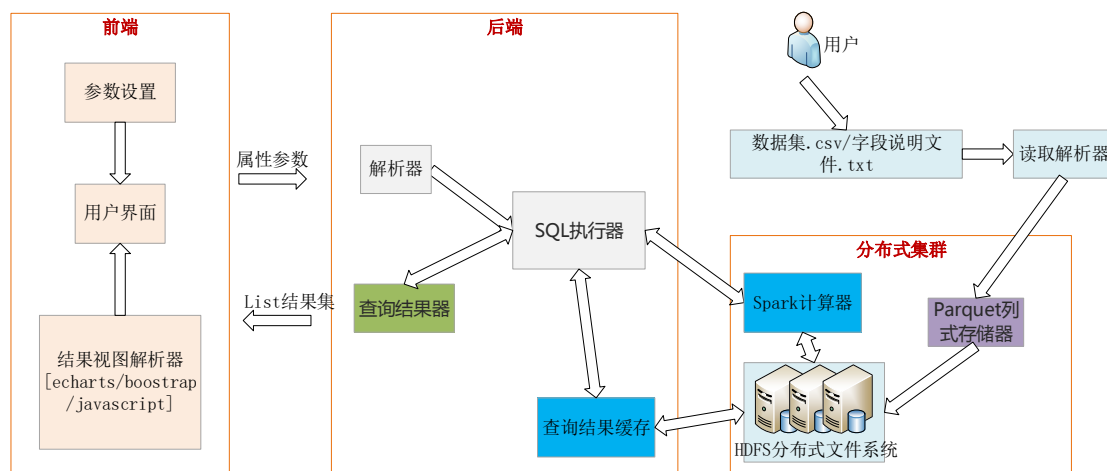


图 1 基于 Spark 的时序大数据分析方案架构图

2.2 数据模型设计

针对海量数据,本文在大数据场景下研究的技术点之一在于对各个数据集进行数据模型设计,使得在进行 Parquet 列存储时可以对数据的各字段类型进行标明,便于 Spark SQL 在查询时可以直接仿照关系型数据库的 SQL 查询。

2.2.1 文件结构

1) 源数据集

每个数据集含*.txt 和*.csv 两个文件,前者是字段说明文件,后者是样例数据集文件。此处需要对字段说明文件进行统一的文件结构进行设计,以数据集的字段说明文件为例,如下表 1 所示。

表 1 字段说明文件结构设计表

数据集名称	详细信息				
Weather.txt	location:string ,F,F,T, month:int ,T,F,F dayofyear:int ,T,F,F year:int ,T,F,F temperature:float ,F,T,F,				
字段说明文件	字段名称	字段类型	X 属性	Y 属性	Z 属性
*.txt	name	dataType	xAxisColumns	yAxisColumns	zAxisColumns

由此表 3-1 可知,将关系型数据库中的数据表字段通过文本进行说明设计,也便于程序

理解和用户设计。此处，在字段说明文件结构设计中需要确定各字段的轴属性，在应用程序界面上的“属性选择区”得以支持用户修改查询条件。

2) HDFS 文件目录

在 Hadoop 集群上，关于数据集在 HDFS 文件系统上的存储结构也需要一定的设计，便于程序进行更好地查询和存储操作。其中，本文研究方案中设计的 HDFS 文件系统上的目录结构如下图 2 所示。

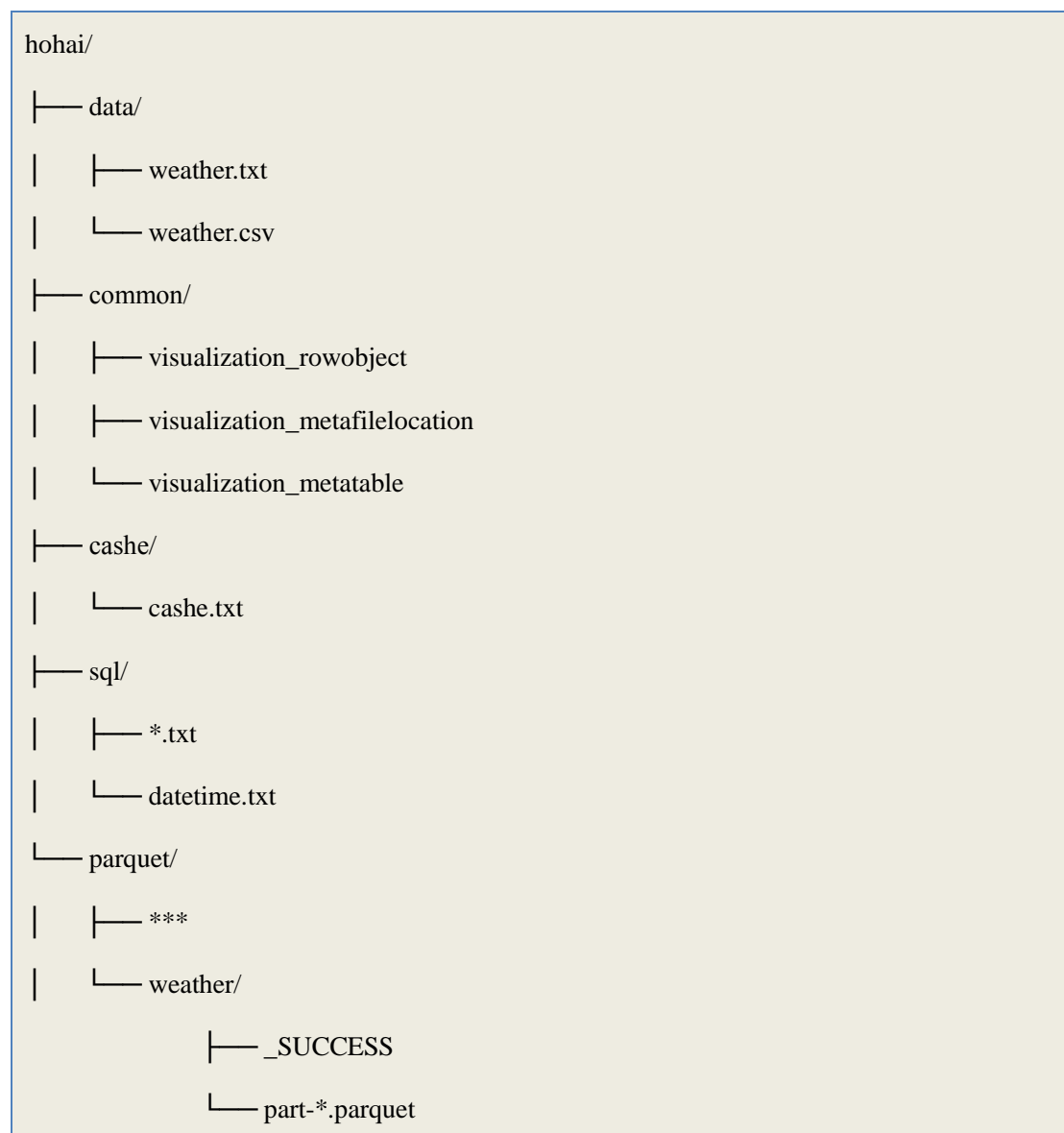


图 2 HDFS 目录结构示意图

由以上 HDFS 文件系统上的目录结构可知，data 文件夹下是上传的源数据集，parquet 文件夹下是各数据集的 Parquet 列存储转化之后的存储位置；此外，common 文件夹是为了便于系统的动态查询而设置，其下的 metafilelocation 保存了各数据集的*.txt 和*.csv 文

件目录信息，metatable 保存了各数据集的字段名称和类型信息，rowobject 保存了各数据集的字段说明文件的字段和字段类型信息。

2.3 HDFS 列存储

HDFS 列存储，即对分布式文件系统 HDFS 上的存储数据集以 Parquet 文件形式存储，使得原本在 HDFS 上的存储空间大大降低，同时在 Spark 等分布式计算框架下的查询效率更高。

2.3.1 HDFS 文件系统的部署

为了能够部署 HDFS 分布式文件系统，搭建 Hadoop 分布式集群至少需要 3 个节点，即 3 台虚拟机，每个虚拟机的运行环境为：CentOs7，单个虚拟机的配置 CPU 核数为 2 个，内存为 2G，单个节点为 20G 的磁盘空间^[5]，其部署结构图为：

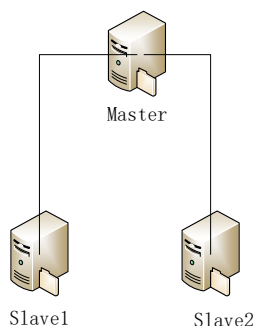


图 3 HDFS 分布式文件系统部署图

2.3.2 文件功能

本文主要在 HDFS 文件系统上实现以下接口：如实现文件目录生成、文件删除、文件读取、目录删除、显示指定目录下的所有文件、上传内容等。

2.3.2 列存储

Apache Parquet 存储引擎是大数据生态圈下的一种文件列式存储格式^[6]**错误!未找到引用源。**，其主要优点是对文件进行了双重压缩，在存储上大大减少了源数据集的存储空间，同时兼容开源大数据下的计算引擎 (Mapreduce、Spark 等)，大大降低了查询时间。

2.4 Spark SQL

类似于传统关系型数据库中的 SQL 查询语句，Spark SQL 应用的查询语句也是由投影、数据源、过滤等条件组成。在本文 Spark 查询的研究方案中，Spark 应用程序可以通过 RDD、JSON 格式数据等数据源创建 DataFrames，再将 DataFrame 注册为临时表后，就可以对该

DataFrame 执行类似 SQL 查询的操作，从而获取应用的结果集。

2.4.1 查询方案

Spark SQL 查询方案允许用户使用 SQL 语句或丰富的 DataFrame API 来查询 Spark 程序中的结构化数据，由于海量数据都是有一定结构的，因此符合应用场景要求。其中，根据数据源类型，Spark 官方提供了三种方案。

- 1) 通过 Jdbc 方式操作数据库
- 2) 对源数据集 (csv、json 等格式) 进行访问
- 3) 对 Parquet 文件访问

对于以上三种数据源的操作方案，在后文 4.2.2 节中对查询性能进行了实验设计，本文考虑到存储和查询优化选择了方案 3，系统稳定，对于 Zenvisage 在时序大数据应用场景下的瓶颈有一定的助力。

2.4.2 查询结果缓存

如前文中 2.2.1 节中 HDFS 目录结构示意图所示，cashe、sql 文件夹主要是为了实现用户的查询结果缓存。从对于用户的 SQL 查询结果访问流程可以如下图 4 所示。

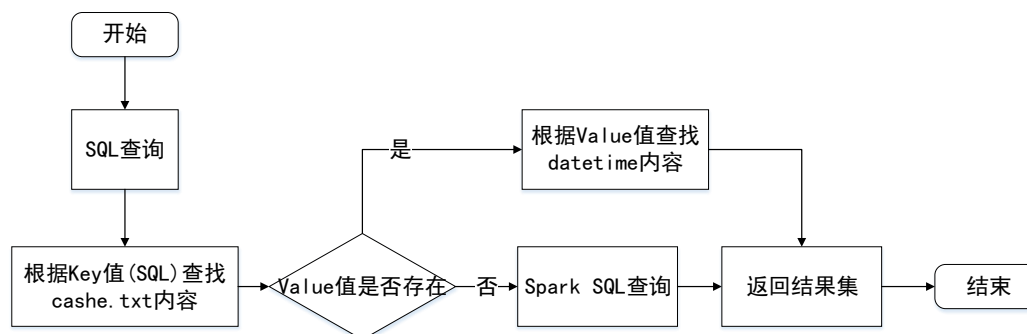


图 4 SQL 查询结果访问流程图

3. 时序大数据的可视化分析系统实现

本可视化分析系统由 SpringMVC 进行底层架构设计，前端采用支持数据自动化双向绑定、依赖注入等特性的 AngularJS 优秀框架，弥补了 HTML 在构建网页应用程序上的弊端，并且应用 Dygraph 这一开源的可交互式、可缩放的曲线表，使用户可以实现视图拖、拉、拽等功能。

接下来主要是对于系统实现中的系统总体结构设计、软件结构设计等几大模块进行介绍。

3.1 系统总体结构设计

在设计本系统时，主要考虑将基于 Spark 的时序大数据分析方案集成于 Zenvisage 的可视化数据探索系统底层，其在功能上主要分为四个模块内容，如下图 5 所示。

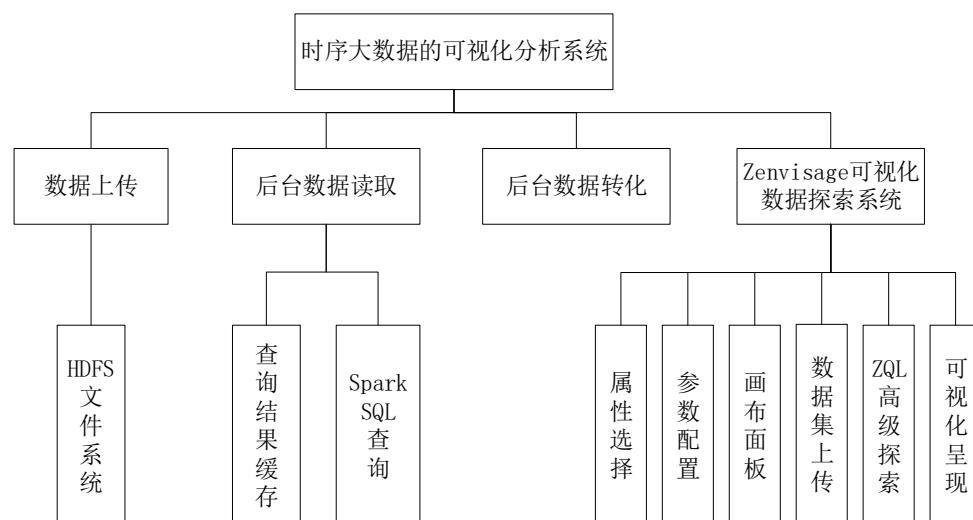


图 5 时序大数据的可视化分析系统的系统结构图

3.2 软件结构设计

本系统在功能实现上主要分为四个模块内容，分别为数据上传、后台数据读取、后台数据转化、Zenvisage 可视化数据探索。下面分别进行介绍。

3.2.1 数据上传

本系统的数据来源主要是两方面：一方面来自系统启动时本地的数据集，另一方面来自用户用于分析提交的数据集。在系统启动时，本模块需要先将本地数据集上传至 HDFS 文件系统上^[7]，并进行数据块的副本复制和文件 Parquet 转化，经过适当的处理工作后，再为系统后期需求建立临时目录和数据缓存等操作。

3.2.2 后台数据读取

本模块通过 SQL 查询读取的数据来源也主要是两方面：一方面来自 HDFS 缓存文件数据，另一方面来自 HDFS 文件系统上的 Parquet 数据集文件，这种查询缓存的设计在一定程度上减少了后台接口的访问次数和 I/O 操作。其中，具体介绍内容如前文 2.2.1 节和 2.4.3 节中的 HDFS 目录结构、查询结果访问流程图所述，其结构图如 6 所示。

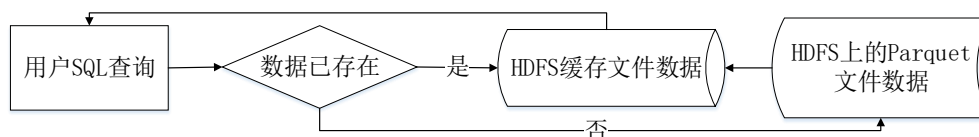


图 6 后台数据读取功能结构图

3.2.3 后台数据转化

后台数据转换模块的主要功能是负责将后台数据读取模块读取的数据文件（Json 格式文件或 Parquet 文件等）根据底层接口的设计要求对文件内容进行转换，具体可分以下几个步骤实现：

- （1）打开 HDFS 文件，进行 Load 操作读取文件内容；
- （2）如果接口请求内容是 JSON 格式文件，则读取内容为 JSON 字符串，可以将其转换为 JSON 对象^[8]，然后进行步骤 3；如果接口请求内容是 Parquet 文件，则读取结果为 DataFrame，进行步骤 4；
- （3）对转换过得 JSON 对象进行数据过滤、分类等转换工作；
- （4）如 2.4.2 节中查询方案中的操作步骤所述，将单个 DataFrame 注册为临时表，从而直接进行 SQL 语句查询，并将查询结果转化成 Json 对象。
- （5）将步骤 3、4 中转换过得 JSON 结果提交给 Zenvisage 底层处理，最后将处理结果在前端进行显示输出。

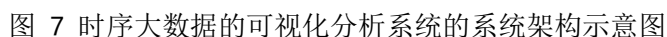
3.2.4 Zenvisage 可视化数据探索系统

由于本文研究的基础是 Zenvisage 平台进行本方案的集成，本文最终实现的时序大数据的可视化分析系统沿用了其大部分功能，并负责对后台数据转化模块传递的数据进行接收，从而满足用户或者数据人员交互式分析的需求。

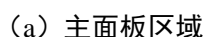
3.3 系统架构设计

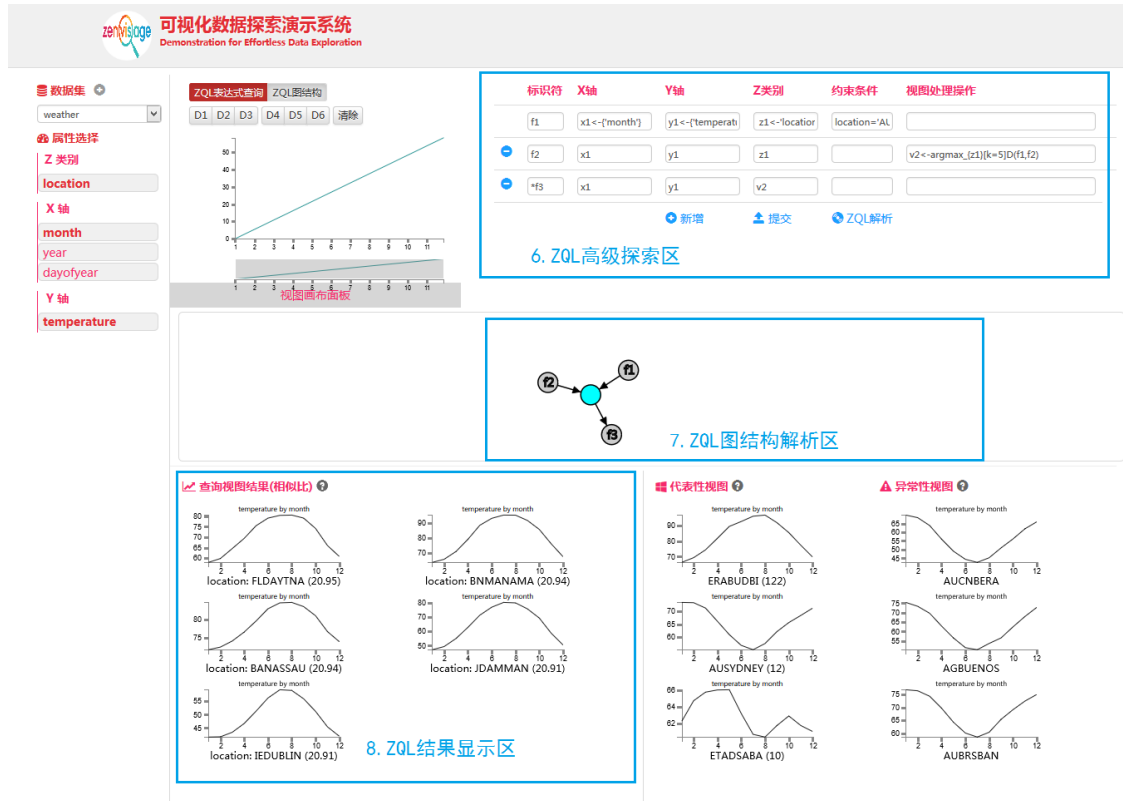
前文 2.1 节中介绍了基于 Spark 的时序大数据分析方案架构图，考，同时加上 Hadoop 集群上的 HDFS 目录设计结构，我们还需要在之前的架构图上添加系统其他功能的机制。

因此，我们将系统架构抽象成如下图 7 所示：



本文将基于 Spark 的时序大数据分析方案集成于 Zenvisage 的可视化数据探索系统的底层,并对 Zenvisage 平台进行功能上的调整,本节将对系统功能的实现效果做一个界面化的演示。如下图 8 为系统在 PC 端的运行成果图:(a)系统主面板区域;(b)ZQL 高级探索区;(c)数据源上传区。





(b) ZQL 高级探索区



(c) 数据源上传区

图 8 时序大数据的可视化分析系统的界面展示

4. 实验结果分析

通过实验测试出本文提出并实现的基于 Spark 的时序大数据分析方案的查询效率和列存储的有效性。本文首先给出了实验设计，包括集群配置、数据集以及对比方案等。然后从

数据存储的大小、数据查询性能测试实验等两个方面来评估本文研究内容的有效性。

4.1 实验设计

本文提出的基于 Spark 的时序大数据分析方案是针对海量时序数据的存储和查询的分析方案，由于海量数据的特点是不确定性数据集大小和字段数量，因此本文选择 Zenvisage 平台应用场景下有代表性的数据集；同时，考虑 Zenvisage 平台中的 jdbc 查询是在单节点的环境下，所以 SQL on Hadoop 系统 Spark 也是通过单节点来执行。基于 Spark 的时序大数据分析方案其效率实验设计将分别从集群配置中的软硬件资源、SQL 查询负载设计以及对比方案等方面介绍。

本实验的时序大数据分析方案中的数据存储部署于拥有 3 个节点的 Hadoop 分布式集群上，其中 1 个节点作为主节点，其他 2 个节点作为从节点^[9]。每个节点 CPU 核数为 2 个，内存为 2G，操作系统使用当前最新版 CentOS7。每个节点拥有 20G 的磁盘空间，且都是在虚拟机中搭建；同时，Zenvisage 分析方案中的数据存储部署于 Postgresql 中，虚拟机环境一致。此外，Java 版本为 1.8，Postgresql 版本为 9.5，Hadoop 版本为 2.7.3，与 Hadoop 集群配套的 Spark 版本为 2.1.1。

在数据集选择上，考虑到 Zenvisage 平台的应用场景的有效性，我们选择其在 Postgresql 数据库中的 weather 数据表，目前在 Github 中的数据集大小为 24.5MB。为了扩展 weather 数据集的大小，通过 Python 程序将数据规模扩展大小为 485.2MB、1.02GB、3.02GB、5.02GB、8.02GB。

在 SQL 查询负载设计上，我们考虑到 Zenvisage 平台中后台使用的 SQL 查询语句，主要在于 Group By 分组、where 条件等操作，同时参考 HAIL^[10]的实验设计中对 UserVisits 表的查询设计，将实验查询从查询字段类型、数据规模和查询效率三个维度上对 weather 数据集设计如下表 2 所示：

表 2 实验 SQL 查询设计表

名称	SQL 语句
Q1	SELECT location, month, avg(temperature) FROM weather where location = 'BRBRGTWN' GROUP BY location, month ORDER BY month
Q2	SELECT location, month, avg(temperature) FROM weather GROUP BY location, month ORDER BY month

Q3	SELECT location, year, avg(temperature) FROM weather where location = 'BRBRGTWN' GROUP BY location, year ORDER BY year
Q4	SELECT location, year, avg(temperature) FROM weather GROUP BY location, year ORDER BY year

其中,Q1 和 Q3、Q2 和 Q4 分别属于一类查询,前者包含 where 条件,属于对单个 location 的细粒度查询;后者对所有 location 的结果进行查询。

Q1~Q4 中 每一个查询在 24.5MB~8.02GB 的数据规模上均进行了相应实验,以更好地提现基于 Spark 的时序大数据分析方案的优化性能。

在对比实验方案中,我们使用相同的数据集、相同的 SQL 查询负载,通过数据集存储大小进行存储方案的对比,以及 Zenvisage 中的 jdbc 方式查询、本文研究方案查询以及 Postgresql 管理器查询等三方面进行查询效率对比;同时,针对现有的 Spark SQL 中不同查询方案下的时间性能对比。为了保证查询结果的准确性,每次查询执行都是在执行结果稳定时记录,且每个查询的查询执行时间都是执行三次后计算平均时间而得来。

4.2 对比方案

由上述 4.1 节中实验设计可知,本文的实验对比方案主要从存储和查询两方面来进行。

4.2.1 存储方案

本文提出的存储方案是先将数据集提交至 HDFS 文件系统上,然后进行 Parquet 列存储转化,减少了数据集在 HDFS 上的存储空间,对比 postgresql 数据库和实际数据集大小进行实验对比。

4.2.2 查询方案

对于数据集的结果查询性能是针对相同的 SQL 在不同查询方案下的效率对比,本文将设计两个实验,实验一是通过 Postgresql 管理器、jdbc 和 Spark SQL 三方面进行实验,此实验不仅可以测试出三种方式在 SQL 查询时的效率,还可以证明随着数据规模的变化来验证本文采取的大数据分析方案的有效性;实验二是针对现有的 Spark SQL 中不同查询方案下的时间性能对比,查询 SQL 查询为 Q1,选择的数据集规模分别为 24.5MB 和 485.22MB,继而验证选择 Spark SQL 对于 Parquet 列存储文件进行查询时能够带来查询性能的优化。

4.3 结果与分析

1) 存储方案实验结果

本文提出的存储方案将 HDFS 文件系统上的数据集进行 Parquet 列存储转化，对比 postgresql 数据库和实际数据集大小的实验对比如下图 9 所示。

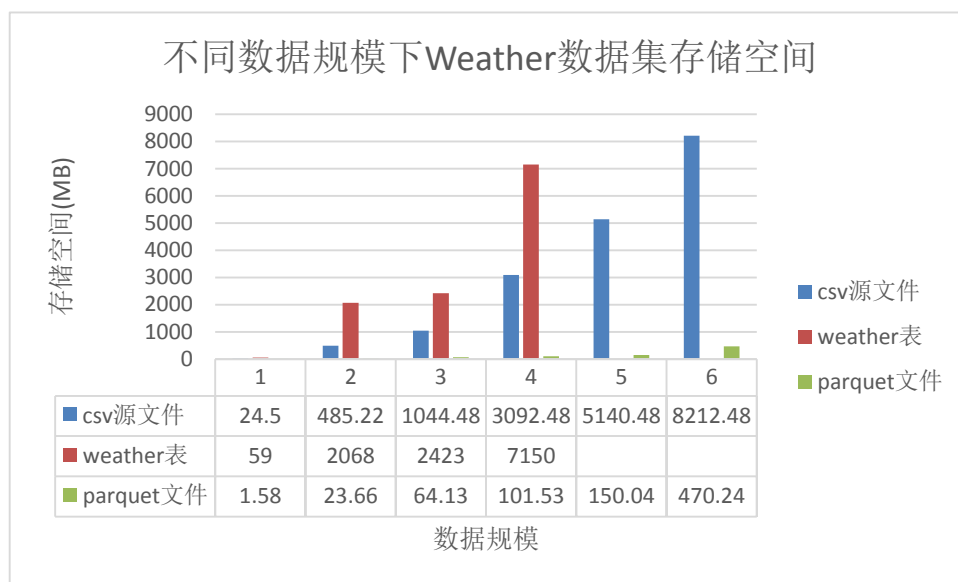


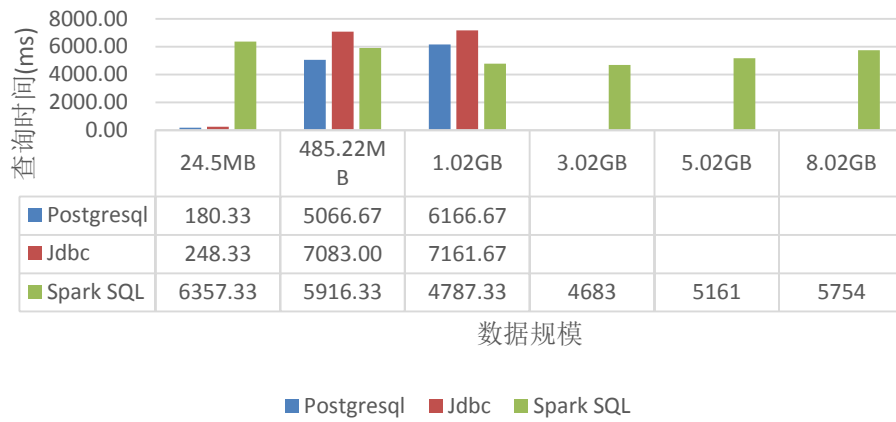
图 9 不同规模下 Weather 数据集存储空间对比实验

由上图 9 可知，在 5.02GB、8.02GB 数据规模的源数据集中，导入 Postgresql 数据库在实验中已经造成了很大的困难（这里为略），且关系数据库行存储结构占用了很大表空间；此外，HDFS 文件系统不会影响数据集大小，但在本文方案中将源数据集转成 Parquet 格式，对数据集的存储性能均有大幅度提升。

2) 查询方案实验结果

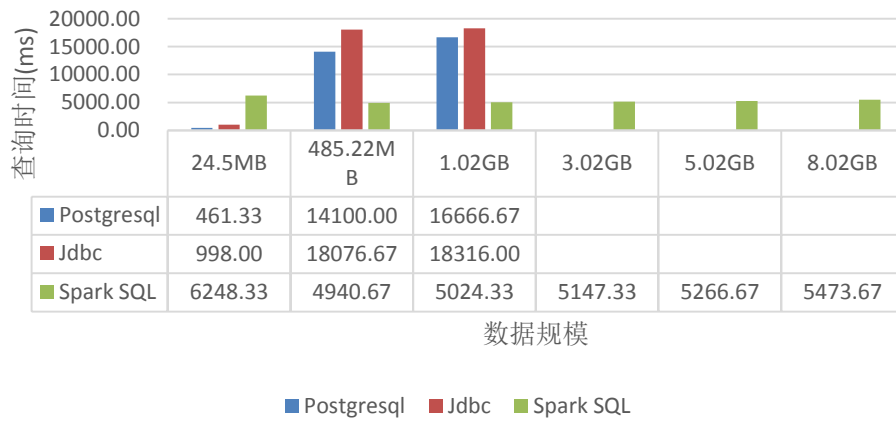
实验一是通过 Postgresql 管理器、jdbc 和 Spark SQL 三方面进行实验。实验设计如下，实验包括 Q1~Q4 这 4 条 SQL 查询语句，通过对 24.5GB~8.02GB 等不同数据规模条件下的查询性能对比实验。按照此设计实验并进行测试，实验一的测试结果如下图 10 所示。

不同数据规模的Q1查询下不同方案的查询性能对比实验

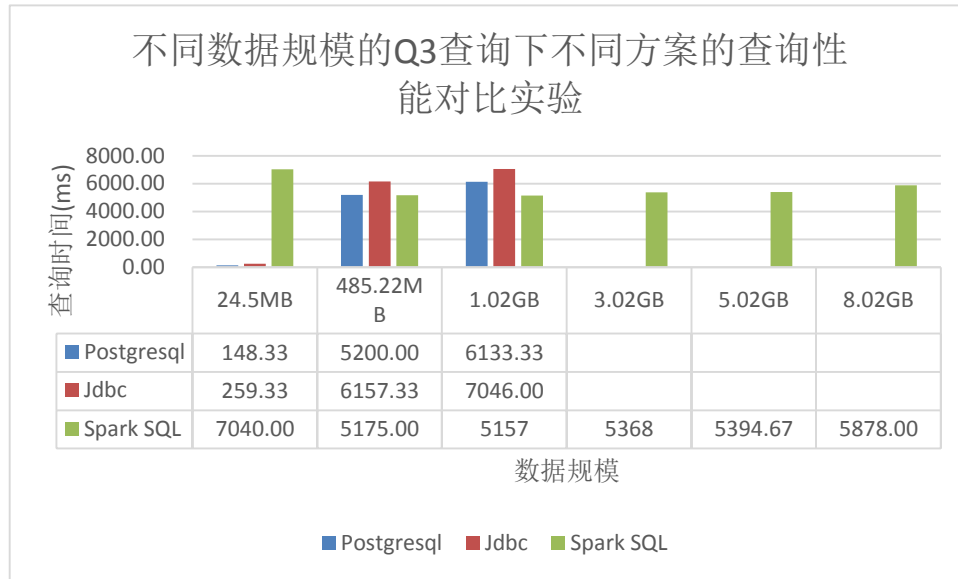


(Q1)

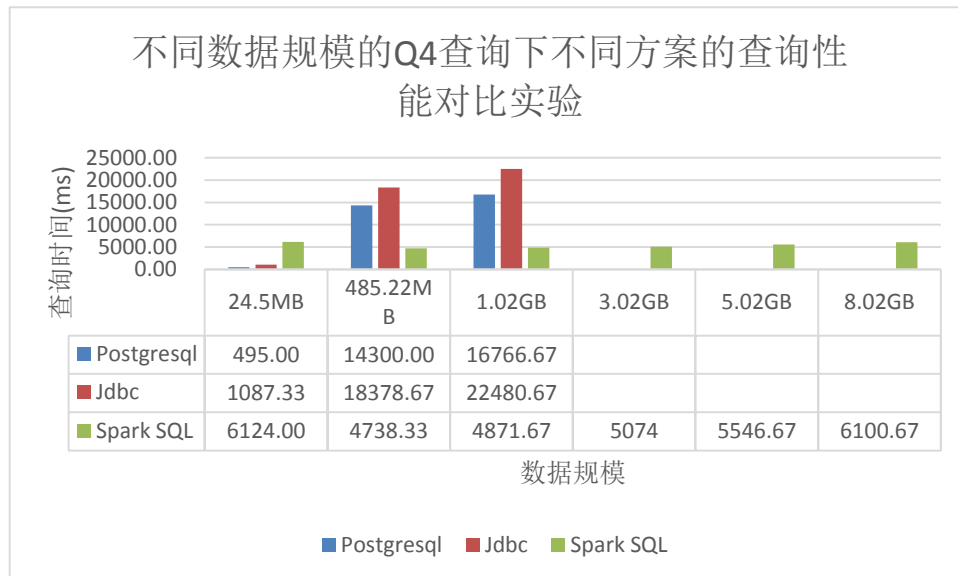
不同数据规模的Q2查询下不同方案的查询性能对比实验



(Q2)



(Q3)



(Q4)

图 10 不同数据规模的 Q1~Q4 查询下不同方案的查询性能对比实验

如图 10 中所示的在 Q1~Q4 查询条件下，我们对不同数据规模下的 Weather 数据集查询结果在不同方案下进行了对比实验。在查询时间上，随着数据集的增加，各方案中的查询时间基本呈上升趋势；此外，在 24.5MB~8.02GB 的数据集上，Spark SQL 的数据查询方案在数据集更大的情况下明显优越，其中，在数据集 GB 级上时根据 5.2.1 方案中的存储无法进行查询（此处为略，按照理想推断其查询性能仍不如 Spark SQL 方案）。

实验二是针对现有的 Spark SQL 中不同查询方案下的时间性能对比，继而验证选择 Spark SQL 对于 Parquet 列存储文件进行查询时能够带来查询性能的优化。目前，Spark SQL

在对数据集进行查询时候有三种方案，通俗来说是：（a）直接针对 Postgresql 操作，这种情况下系统需要安装数据库；（b）对 HDFS 文件系统上的源数据集进行操作；（c）对 Parquet 数据集文件进行操作。按照此设计实验并进行测试，实验二的测试结果如下图 11 所示。

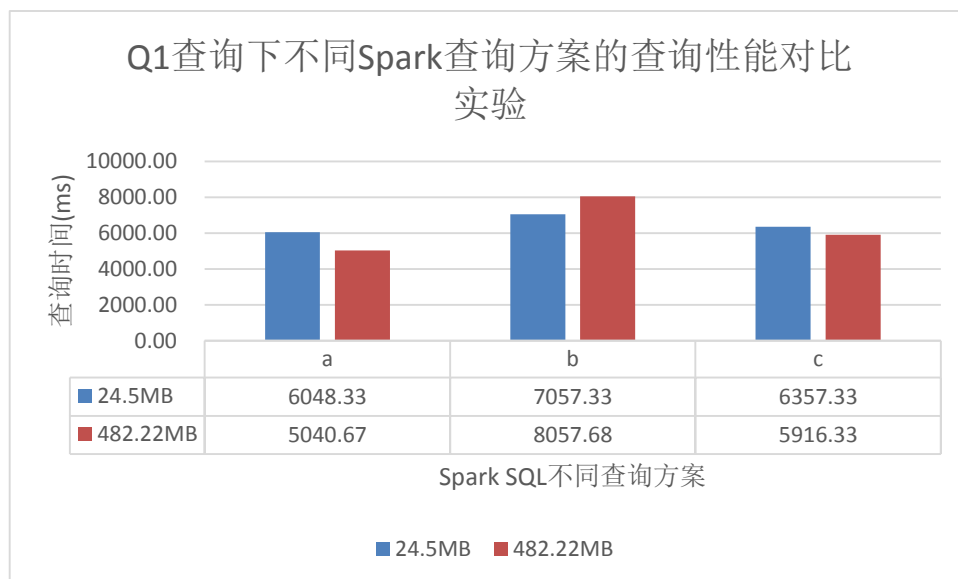


图 11 Q1 查询下不同 Spark 查询方案的查询性能对比实验

前文 2.4.2 节中详细介绍了 Spark SQL 在时序大数据场景下的三种查询方案的步骤与特性，对于（a）方案，尽管在小数据下查询效率高，在数据量增加时和（c）方案在 Q1 下查询时间差不多，但是需要提前安装 Postgresql 软件，不符合大数据下的应用场景。对于（b）方案操作下，（c）方案在不同数据规模下查询性能更高，（b）使用了 Parquet 列存储转化功能，在数据存储上有一定优越性。

5. 结束语

本课题是“时序大数据可视化分析平台研究与设计”，主要研究了时序大数据可视化分析的需求，分析了现有相关工作的不足，总结出现有的可视化分析工具操作繁琐、重复、耗时，指定可视化趋势困难，以及在时序大数据场景下存储和查询性能下降时的问题。为此，我们借助开源的 Zenvisage 可视化分析平台，提出了基于 Spark 的时序大数据分析方案，并将其集成于 Zenvisage 的可视化分析平台底层中，通过系统测试和实验结果分析验证了基于 Spark 的时序大数据分析这一方案在时序大数据场景下存储和查询性能的优化，以及其在 Zenvisage 平台下的可用性和有效性。

参考文献

- [1] Tableau public (www.tableaupublic.com/). [Online; accessed 3-March-2014].
- [2] Spotfire, <http://spotfire.com>. [Online; accessed 17-Aug-2015].
- [3] Vartak M, Rahman S, Madden S, et al. SeeDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics[C]// Proceedings VLDB Endowment, 2015:2182.
- [4] Siddiqui T, Kim A, Lee J, et al. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system[J]. Proceedings of the Vldb Endowment, 2016, 10(4):457-468.
- [5] 杜小勇, 陈跃国, 覃雄派. 大数据与 OLAP 系统[J]. 大数据, 2015(1):48-60.
- [6] Abadi D J, Madden S R, Hachem N. Column-stores vs. row-stores:how different are they really?[C]// ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, Bc, Canada, June. DBLP, 2008:967-980.
- [7] Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System[C]// MASS Storage Systems and Technologies. IEEE, 2010:1-10.
- [8] 郭庆燕, 张敏, 杨贤栋. JQuery Ajax 异步处理 JSON 数据实现气象图片的显示[J]. 计算机应用与软件, 2016, 33(6):20-22.
- [9] 何龙, 陈晋川, 杜小勇. 一种面向 HDFS 的多层索引技术[J]. 软件学报, 2017, 28(3):502-513.
- [10] Richter S, Quian, Ruiz J A, et al. Towards zero-overhead static and adaptive indexing in Hadoop[J]. The VLDB Journal, 2014, 23(3):469-494.