基于 Spark SQL 的数据查询与索引优化系统研究

陈春茹 [1]
CHEN Chunru

摘 要 随着大数据及云计算技术、移动场景应用数据量的迅猛发展，对于动态场景下的时态大数据查询与处理分析，成为不同企事业单位高吞吐量、低延迟数据管理关注的重要方向。基于 Apache Spark 分布式计算框架，搭建起涵盖 Spark SQL 解析器、Catalyst 查询优化器、Data Frame 查询接口、Hive 数据仓库、RDD（resilient distributed datasets）数据库等组件的大数据查询分析系统，针对海量的半结构化、非结构化时态数据，基于 Spark SQL 内核的 Parser 组件拓展时态查询的范围，使其支持特定索引创建、删除与内存读入管理的关键字，将本地分区建立的时态索引打包为 IndexRDD 数据集，利用局部时态索引模型展开含有 K 个时态对象的数据查询，快速扫描、查询与定位相应的数据项位置，进而提升时态数据查询的容错性、执行性能。

关键词 Spark SQL 组件；数据查询与索引优化；系统

doi：10.3969/j.issn.1672-9528.2024.07.036

0 引言

传统关系型数据库是基于网络存储器、后台服务器等硬件的数据管理模式，对于数据集群查询与计算分析的耗时长、扩展性不足，难以满足海量大数据的查询、索引分析和存储需求。为满足大量非结构数据、业务迁移数据的分布式检索需求，提出基于 Spark SQL 的时态查询与索引扩展系统，使用 SQL 解析器查询读入与添加词法出现的时间段、时间点，通过全局过滤判断查询点、查询区间是否落在数据集内，在 Spark SQL 内核中引入局部时态索引模型、索引管理器进行海量时态数据的分布式索引与查询，通过集群内存的列存储完成索引数据存储，使得系统的大数据查询与索引可保持高吞吐量、高容错性与低时延 [1]。

1 基于集群计算的 Spark SQL 时态查询系统的关键技术

1.1 Apache Spark 框架技术

Apache Spark 是在 Hadoop MapReduce 框架分布式、可扩展、容错处理等优势的基础上，改进与开发出的更高效的分布式大数据处理引擎，被广泛应用于各类大数据处理场景。基于 Java、Scala 通用编程语言，建立起涵盖 Driver application main 运行逻辑、spark task 运行进程、client 客户机等结构的 Apache Spark 大数据分析系统，在系统框架内设置丰富的数据源 API 接口，可为 Spark Streaming 流式计算、Spark SQL 查询解析器、MLib 机器学习库、GraphX 图形计算库等组件的连接提供支持（如图 1 所示）[2]。相比于 Hadoop 框架的数据查询处理而言，基于 Apache Spark 框架的内存任务处理速率要快 10 ～ 100 倍，并能够使用 80 多种操作符的内置集合进行数据查询操作 [3]。

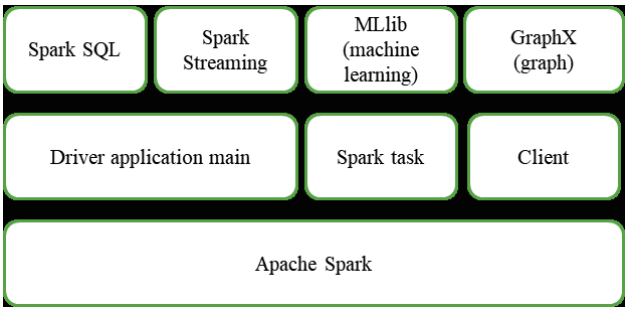


图 1 Apache Spark 框架

1.2 Catalyst 翻译引擎技术

Spark SQL 框架内置的 SQL 解析器、Catalyst 查询优化器等组件，是大数据查询与索引的最核心组件，Catalyst 翻译引擎负责将 SQL 解析过程，转换翻译为能够被认知的底层分布式计算任务，SQL 解析器负责任务计划的执行与优化。

在大数据分析系统导入元数据后，使用 SQL 解析器访问半结构化、非结构化数据文件，通过 Schema 类构造函数生成 Data frame 数据描述（结构元）信息，使用 SQL antlr 解析器对输入的元数据自动生成语法树；定义作用于一个对象中所有元素的操作为 visitor（访问者）模式，使用 visitor 模式将 antlr 解析器生成语法树替换为 Catalyst 查询优化器的计划

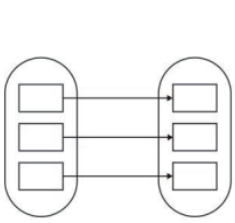
1. 山西金融职业学院 山西太原 030008

树，将计划树的每个节点与 Spark 执行引擎的元数据信息相匹配，设定为元数据与计划树的物理匹配计划 [4]。

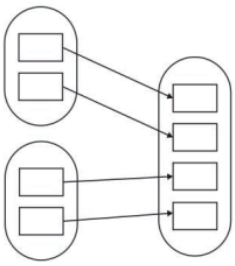
1.3 RDD 数据库技术

RDD 数据库为数据并行存储、分区控制的工作流结构，包含多个只读的分区、每个分区用于记录部分的数据集合，支持数据映射、过滤、合并、运算及存储的任务执行，其中数据映射、过滤、合并、运算等的转换操作在同一 RDD 数据库内执行，数据存储操作在另一 RDD 数据库内执行。但利用 RDD 数据库的数据转换过程并不能立即得到运算结果，只是记录下海量数据集的处理流程、处理后对应的数据集合 [5]。

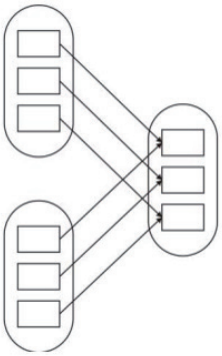
同时，RDD 数据库结构自身的容错机制，包括数据检查、数据记录与更新机制，能够为海量化数据表的依赖关系 RDD 分区、数据分析与恢复提供支持。借助数据中心网络连接，Spark RDD 数据库支持粗粒度的数据转换，每一 RDD 只记录单个分区块的数据映射、过滤、合并、运算操作，随后创建另一 RDD 记录数据转换过程、重建的数据信息，因而当某一 RDD 分区的缓存数据丢失时，可通过 RDD 数据库的容错机制恢复和还原丢失的分区数据，不同 RDD 分区之间的依赖关系如图 2 所示 [6]。



（a）RDD 分区输入数据映射，过滤的依赖关系：一对一



（b）RDD 分区输入数据联接依赖关系：一对一



（c）具有共分区输入数据联接依赖关系：多对一

图 2 RDD 数据库结构

以上用于数据转换的 RDD 分区存在着窄依赖关系，即每个子 RDD 分区对应一个父 RDD 分区，例如数据映射、过滤等操作；*n* 个子 RDD 分区对应着一个父 RDD 分区，例如数据联接操作，之后需调用 Spark map 算子执行数据索引操作，并将数据索引执行结果返回到驱动程序。

1.4 Hive 数据仓库技术

Hive 数据仓库为 Spark SQL 数据查询的映射工具，基于 HiveQL 查询语言可从不同数据源中抽取、转换和加载数据，设置索引管理的数据库表、数据库分表，将处理后的数据分类至不同的表中进行管理。通过 Hive 数据仓库的查询和分析功能，可轻松地获取用户数据浏览、调用的偏好信息，结合机器学习算法可实现更加智能化的数据分析和挖掘 [7-8]。

1.5 内存列存储技术

内存列存储为 Spark SQL 大数据分析系统的最优存储方式，相比于传统的 JVM 对象存储方式，基于数据表分区的内存列存储所占用空间更小，读取数据吞吐量更高。JVM 存储的每个数据对象都要耗费几十字节空间、含有百兆级别的数据集要耗费 2 GB 以上的数据空间，而 Hive 支持下的内存列存储，会将数据映射、过滤与合并后的数组执行序列化，将每一列数组对应于一个 JVM 对象，并对待存储的数组作出列长度编码、字典编码，使得列聚合数组占用的内存更小，查询分析效率更高 [9]。

2 基于 Spark SQL 框架的大数据分析系统架构

在时态大数据查询与索引过程中，外部访问用户可通过创建 Spark Shell 命令、执行 Spark 应用程序等两种方式，使用 SQL antlr 解析器解析时间关键字，索引不同时态下的分区数据。因而大数据查询与索引优化系统包括 SQL 客户端层、Catalyst 翻译引擎层、并行计算层、分布式索引存储层、Spark 内核层的层级，不同层级分别负责接收用户查询语句，以及提供执行词法或语法解析优化、分布式全文检索、分布式索引与存储等功能服务 [10]。

2.1 SQL 客户端

SQL 客户端为接收用户输入 SQL 语句、语句提交模块，利用 Spark SQL-Shell 命令行工具与 Spark SQL 进行交互并提交 SQL 查询语句，将 SQL 语句提交给翻译引擎。同时，通过 Spark deploy-mode 程序部署模式、Spark-conf 程序属性、executor memory 内存大小等的常用参数配置，方便完成“insert into _temp(Id, name, area_id)”临时表创建、数据加载和处理操作 [11]。

2.2 SQL 翻译引擎层

SQL Catalyst 翻译引擎层级是将 SQL 语句翻译成执行计划（ANTLR）树的模块结构，属于 Spark SQL 的翻译解析与调度核心，负责将 SQL 语句转换为逻辑查询计划、物理查询计划并进行翻译解析。其中，逻辑查询计划表示未被解析的语法树（AST），之后根据 Catalyst Analyzer 内置的多条

语义规则，包括 Resolve Relations 表目录关系设置、Resolve References 逻辑计划子节点替换、CTE Substitution 语法解析等的规则来解析 ANTLR 语法树，遍历整个语法树对树上的每个节点进行元数据类型绑定、函数绑定与校验，通过利用局部时态索引模型、Optimizer 逻辑查询优化器将解析完成的 AST 语法树转换为物理匹配计划，将不同数据表以内存列的形式存储至分布式集群之中 [12]。

2.3 并行计算层

并行计算层将 SQL 翻译引擎层的全文检索操作以多任务并行方式运行，包括全文索引检索、数据源对接等执行流程。如在“select [Key], [Data] From BigTable where Data<10000”的查询脚本执行中，被执行脚本的 SQL 语句筛选条件为“where Data<10000”，这就需要 SQL Server 调用网络服务器的 $\{CPU_1, CPU_2, \cdots, CPU_n\}$ 等多个线程，设置并行计算的 MAXDOP 系统参数、指定每个运算符的最大并行计算数，基于数据源对接模块建立各分区表数据的索引，作出多线程的数据表并行索引扫描，可在保证系统线程自动添加或移除、正常数据吞吐率的前提下达到性能最优 [13]。

2.4 分布式索引存储层

分布式索引存储是以 HDFS（hadoop distributed file system）分布式文件系统为通用硬件，执行数据表分区、分片索引存储的指定列存储操作。针对索引关键词的复杂度设定索引分片的列长度编码、字典编码，按照“Index_ name、Talbe_ name、Columnl&Column2”等表数据的分类模式，作出索引分区的指定列存储、全量存储，全量存储适用于短时间内获取到的查询结果存储，指定列存储适用于有限空间的海量化数据存储，可实现成百上千的索引数据存储并返回后台查询结果。

2.5 局部时态索引模型

基于局部时态索引模型建立不同时间的分区数据集对应索引，设置数据集上下的分区边界为（B_l, B_r）、某一时间段中值 timemid 对应着线段树索引的根节点值，小于中值的时间段数据集合称为上边界集合 S_l，大于中值的时间段数据集合称为下边界集合 S_r，第三类为中间集合 S_{mid} [14]。

在用户输入 SQL 时态数据查询语句后，通过 SQL antlr 解析器将 SQL 语句转化为抽象语法树，随后使用二叉线段树索引模型比较线段树节点值、中值 timemid 大小，小于中值的时间段数据集合 Sl 进入左子树，大于中值的时间段数据集合 S_r 进入右子树。设定线段树叶节点的数量为 $n=\left[\sqrt{N}\right]$，那么 n 个叶节点的时间间隔长度为 $time_{mid}/n$，之后按照设定的时间间隔，由不同叶节点对局部 RDD 数据集作出线性遍历，将 where 子句移入查询块进行数据过滤与索引，查询出服务器存在的特定数据项，如果存在已建立的索引，则读入查询结果，否则将得到的结果数据集执行物理计划优化。

3 时态数据查询及索引的仿真实现

3.1 实验环境设置

在网络服务器内设置 1 台 master、9 台 slave，共 10 台物理机，每台物理机参数为 Intel(R) Core(TM) i7-2600 CPU@3.4 GHz 16 GB 1 TB，在物理机中安装 XenServer8.2 虚拟化软件、Hadoop 2.7.7 分布式框架软件，以及在 Hadoop 软件平台运行 Spark、HDFS 等组件，单个 HDFS 分布式系统的块大小为 128 MB，executor memory 可用内存为 5 GB。

3.2 分布式时态数据查询

将 QueryRDD 数据集按照一定规则划分至特定的 Partition 分区中，基于 SQL 语句设置索引分区的全量索引存储、指定列索引存储策略，从 Hive 数据仓库、数据库表中索引与查找符合要求的时空数据，然后将这些数据集合并输出至 HDFS 存储端口，其中全量索引存储的执行代码如下：“create INDEX:usenet_ corpus_ M;_ Si_ test_ index;ON TABLE usenet_ corpus_ M;_ Si_ test;STRATEGY QUICKW AY”。

3.3 实验结果分析

后台服务器内存储有 3 000 000 个待测试数据形成的数据集，以 ScqucnccFilc 作为表数据存储格式，用户输入的数据表命名为 usenet_corpus_Mi_Si_test，采用 {1,2,4,8,16,32,64} 的分片方式作出 Si 数据集分片。选取 1000 个、10 000 个、100 000 个、1 000 000 个数据，形成 RDD 数据集合，基于局部时态索引模型对分片数据集作出索引查询，分析不同 RDD 数据集查询时间、有无索引的查询性能差异，实验结果如图 3 和图 4 所示 [15-16]。

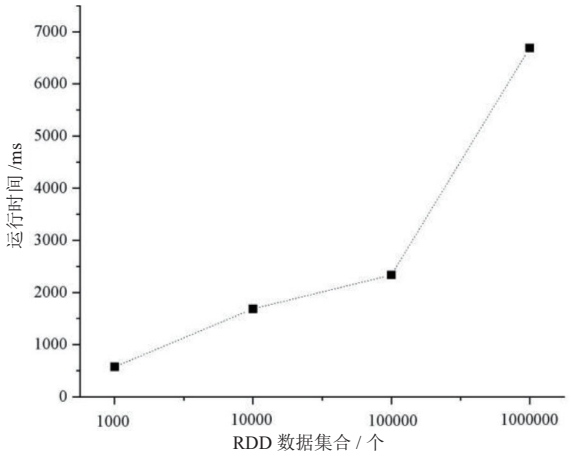


图 3 RDD 数据集查询运行时间

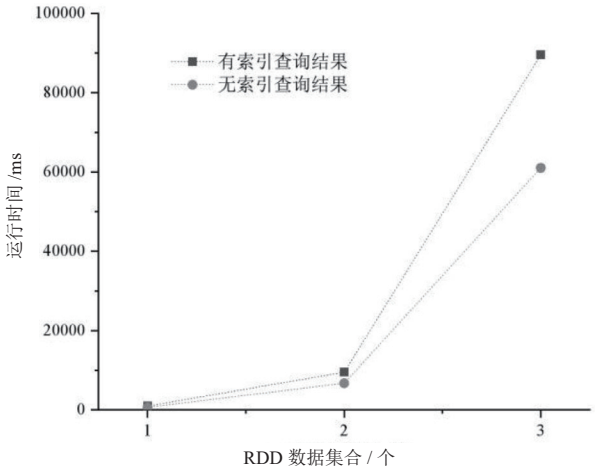


图 4 RDD 数据集有无索引的查询性能差异

从图 3、图 4 的时态数据索引查询结果可得出，基于局部时态索引模型的分片数据集索引查询，随着数据集中数据量的增多，其单位数据处理的运行时间减少，运行效率提升。相比 1000 个数据索引检索耗时 0.5 s 而言，处理 1 000 000 个数据仅需要约 6.7 s。同时，基于时间关键字、数据关键词的 Spark SQL 索引数据查询，相比于 STCode HBase. 无索引的数据查询方案（60% ～ 70%）的效果更优，索引数据查询结果准确率均稳定在 90% 以上，可被用于完成某一时态范围内的数据检索查询操作 [17]。

4 结语

Spark SQL 数据查询与索引系统是利用 Java、Scala 通用编程语言，以 Spark SQL 查询解析器为主的时态数据查询与索引分析，通常用到 Apache Spark 框架、RDD 数据库、Hive 数据仓库、内存列存储、Catalyst 翻译引擎组件等关键技术。通过基于 Apache Spark 分布式软件框架搭建数据处理分析系统，使用二叉线段树索引模型的叶节点对局部 RDD 数据集作出线性遍历，本地分区策略、集群内列存储模式为数据集处理提供操作符，将数据查询、索引的处理结果存储至 HDFS 分布式系统之中，有助于提升时态数据查询的效率与准确率。

参考文献：

[1] 和晓军 , 孙康 . 基于 Spark 的电商用户行为分析系统 [J]. 信息技术与信息化 ,2021(11):95-97.

[2] 林子孟 , 葛欣竹 , 曹若麟 . 面向电信应急系统的 Spark 性能预测与参数调优方法探究 [J]. 电信快报 ,2020(12):26-30.

[3] 谌婧娇 . 基于 Spark 的决策树算法对航班延误预测研究 [J]. 电脑知识与技术 ,2021(4):217-219.

[4] 杨卫宁 , 邹维宝 . 基于 Spark 的出租车轨迹处理与可视化平台 [J]. 计算机系统应用 ,2020(3):64-72.

[5] 胡志宝 , 陆会明 . 基于 Spark SQL 技术的工业数据统计研究 [J]. 科学技术创新 ,2021(6):58-61.

[6] 申毅杰 , 曾丹 , 熊劲 . 基于收益模型的 Spark SQL 数据重用机制 [J]. 计算机研究与发展 ,2020(2):318-332.

[7] 齐超 , 崔然 . 基于递归随机搜索算法的 Hadoop 平台大数据软件系统研究 [J]. 软件 ,2020,41(6):177-184.

[8] 胡晶 . 基于 Spark SQL 的海量数据实时分类查询算法的研究 [J]. 黄河科技学院学报 ,2021(5):35-38.

[9] 胡晶 . 基于 Spark SQL 结构化数据文件的推荐系统性能优化研究 [J]. 电脑与信息技术 ,2021(5):61-63.

[10] 陆赟 , 闫歌 . 利用 Spark SQL 分析传统数据源的通用步骤 [J]. 电子制作 ,2020(16):66-68.

[11] 杨彦彬 , 干祯辉 . Spark 环境下 SQL 优化的方法 [J]. 数字通信世界 , 2021(7):13-14.

[12] 宾茂梨 . 基于 Spark 的结构化数据连接查询优化策略研究 [D]. 重庆邮电大学 , 2022.

[13] 赵丽梅 , 黄小菊 , 宫学庆 . Spark 查询引擎中 Join 操作的优化 [J]. 计算机应用与软件 , 2022(8):44-50.

[14] 刘春雷 . 基于代价模型的 Spark SQL 查询优化研究 [D]. 电子科技大学 , 2016.

[15] 徐石磊 , 王雷 , 胡卉芃 , 等 . 基于分布式系统 OceanBase 的并行连接 [J]. 华东师范大学学报 (自然科学版), 2017(5):1-10.

[16] 王鹏 , 刘鹏 , 刘佳祎 . 基于 MapReduce 模型的并行处理优化策略 [J]. 电子技术与软件工程 , 2021(1):201-203。

[17] 薛慧敏 . 基于 MapReduce 的分布式云计算数据挖掘方法 [J]. 安阳师范学院学报 , 2020(5):24-27。

【作者简介】

陈春茹（1995—），女，山西运城人，硕士，助教，研究方向：计算机科学与技术、大数据、人工智能。

（收稿日期：2024-03-22）