

基于数据湖的海量异构实时处理技术的研究与应用

叶浩 张承琪 张亚威 申小军 杨孟新

(中国联合网络通信有限公司软件研究院 济南 250100)

摘要 按照数字中国“2522”的整体框架要求,需持续夯实数据资源体系,充分发挥数据作为新生产要素的关键作用,在新一代数字科技的支撑和引领下,以数据为关键要素,以价值释放为核心,以数据赋能为主线,推动数字化转型,通过“上云用数赋智”行动,实现高质量的发展。随着企业业务的发展,企业数据数据量不断膨胀,在当前环境下,流批数据复用场景越来越多。传统架构使用数据仓库来处理批计算场景,使用流平台处理流计算场景。相对传统架构,数据湖可以一份存储、批流两用,实现近实时的处理能力,直接进行高效的数据分析。文中提出了一种基于数据湖的数据处理方案,在解决流式处理痛点的同时,实时批流一体处理模式。相对于传统的数据处理模式,流式数据处理有着更高的处理效率和成本控制能力。

关键词: 湖仓一体;Flink;计算引擎

中图分类号 TP274.2

Research and Application of Multi-source Heterogeneous Data Integration Based on Flink Computing Engine

YE Hao,ZHANG Chenqi,ZHANG Yawei,SHEN Xiaojun and YANG Mengxin

(China United Network Communications Corporation Limited,Jinan 250100,China)

Abstract According to the overall framework requirements of Digital China's "2522", it is necessary to continuously consolidate the data resource system, give full play to the key role of data as a new production factor, and under the support and guidance of the new generation of digital technology, take data as the key element, release value as the core, take data empower as the main line, promote digital transformation, and achieve high-quality development through the "cloud with data intelligence" action. With the development of enterprise business, the amount of enterprise data data continues to expand. In the current environment, there are more and more reuse scenarios for streaming batch data. Traditional architectures use data warehouses to handle batch computing scenarios, and use streaming platforms to handle streaming computing scenarios. Compared with traditional architectures, data lakes can be used for both storage and batch streaming, achieving near-real-time processing capabilities and directly performing efficient data analytics. This paper proposes a data processing solution based on data lakes, which not only solves the pain points of streaming processing, but also integrates real-time batch and stream processing. Compared with traditional data processing models, streaming data processing has higher processing efficiency and cost control capabilities.

Key word Integrated lake and warehouse,Flink,Computing engine

0 引言

随着各行业数字化转型的不断深入以及技术的不断发展,传统的离线大数据模式越来越无法满足新兴业务的发展需求,实时化的大数据业务,成为企业深入挖掘数据价值的必经之路。目前,数据湖与实时计算技术快速发展,基于实时能力的批流一体处理模式、湖仓融合的架构等,也在各企业中得到了应用与实践,本文提出了一种解决方案,以实

现海量异构流式数据的快速开发以及批流一体处理能力,其主要优势如下。

(1)数据存储在数据湖提供流处理及批处理的接口;计算接口上统一流批SQL;实现了存储及计算意义上的流批一体化,降低了用户的学习、使用成本。

(2)对于时间要求不高的任务,摆脱传统Kafka源流处理平台,避免数据冗余。

(3)依托数据湖能力实现了数据全生命周期管理,降低

作者简介:叶浩(1990—),本科,中级工程师,研究方向为大数据、分布式架构设计;张承琪(1989—),硕士,高级工程师,研究方向为分布式架构设计、实时数据处理技术;张亚威(1988—),本科,中级工程师,研究方向为项目管理、分布式架构设计;杨孟新(1996—),硕士,初级工程师,研究方向为大数据;申小军(1996—),硕士,中级工程师,研究方向为大数据研发。

数据运维复杂度。

(4)在 Flink SQL 基础上进一步抽象、封装,实现了任务配置的界面化、低代码化。不需要借助 SQL,通过傻瓜式的操作即可完成 Flink 任务配置。

(5)实现集群式、多节点服务, Flink 任务在配置之初,需要指定运行时所用的集群。当启动或停止任务时,路由端会将请求分发到子平台节点,并由子平台节点负责实际启停任务,实现了多集群环境下的 Flink 任务管理调度,且具备集群可扩展性。

1 批流一体整体处理架构

本文提出了一种基于数据湖的流批一体数据处理系统,该系统包括数据采集模块、数据管理模块以及数据处理模块。这种模块化的架构使系统各部分功能清晰,易于维护和扩展。

1.1 批流一体数据处理架构概述

流批一体处理技术^[1]结合了实时流处理和离线批处理的优点,具有灵活的处理能力和高效的计算性能。它能同时处理实时流数据和历史批量数据,满足不同时间要求下的数据处理需求。流批一体处理技术通常基于流式计算引擎和批处理引擎。本文采用的批流一体数据处理系统架构如图 1 所示,具体架构说明如下。

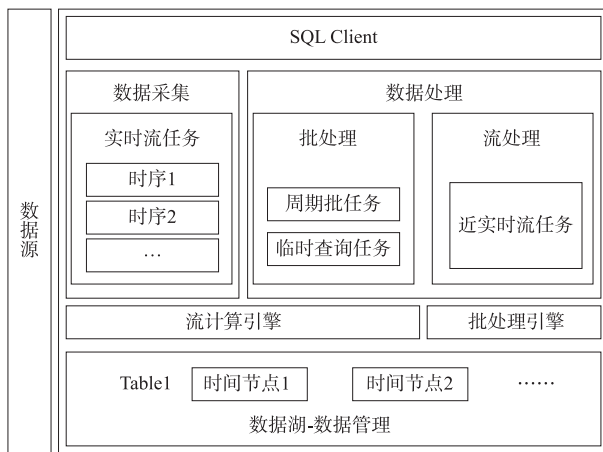


图 1 整体架构图

1.1.1 数据存储层

(1)数据湖:作为系统的存储层,存储着原始数据和处理后的数据。数据湖具备流式读取能力和快照查询能力。

(2)数据管理组件:部署在主机上,负责管理数据表结构和生命周期。

1.1.2 任务提交模块

Web SQL Client:在主机上部署 Web 客户端,用户通过 Web 界面提交各种任务。

1.1.3 任务管理模块

任务管理组件:记录用户提交的任务信息,包括任务名称、计算类型、任务 SQL 及执行周期。用户可通过客户端将

所有查询 SQL 提交至相应的计算组件。

1.1.4 计算模块

(1)流计算组件:接收用户提交的流处理 SQL 任务,并将其提交至 YARN 集群执行。

(2)批计算组件:部署在集群中,用于接收用户提交的批处理 SQL 任务。可以启动多个实例对应多个不同租户,并在 YARN 集群中常驻,限定资源使用上限。

1.1.5 权限管理模块

权限管理插件:部署在主机上,所有查询必须通过该插件进行鉴权。若用户无表访问权限,则拒绝查询。

1.2 多源异构数据采集

数据采集流程支持多种类型的数据源^[2],包括 Kafka、文本文件、数据库等,具体采集流程如下。

(1)用户在创建采集流程时,需选定数据源类型为有界或无界数据源,并配置流程所需资源,编辑并采集好 SQL 后,提交至服务器任务管理组件。

(2)任务管理组件收到请求后解析提交信息,将任务配置、存储到数据库中,并将任务提交至流计算引擎。

(3)流计算引擎接收请求后,将任务提交至计算集群执行任务,并将提交状态返回给任务管理组件。任务管理组件接收到提交状态后,将状态返回给用户,若提交失败则会解析失败原因,并向用户返回报错信息。

(4)集群采集流任务按照业务规则处理数据,然后根据时间顺序进行数据聚合,并将聚合后的数据写入数据湖。

(5)数据湖接收到一个批次的数据后,会生成一个快照。快照包含时间节点信息,然后将元数据信息存入元数据库中。

1.3 数据批流处理

数据处理流程包括数据流处理及批处理,具体如下。

(1)用户创建处理任务,并配置任务开始的批次、流程等所需的资源及数据输出类型,编辑好 SQL 后提交至服务器 SQL 解析组件。输出类型包括 Kafka、数据湖等。

(2)SQL 解析组件将解析提交的 SQL,分辨任务是流处理任务、批处理任务还是临时查询任务,并将任务提交给任务管理组件。

(3)任务管理组件收到请求后开始解析提交的信息,并将任务配置存储到数据库中,再根据任务类型将任务提交至流计算或批计算引擎。

(4)若任务为流计算任务,则将任务提交至计算集群执行任务,并将提交状态返回给任务管理组件。任务管理组件接收到提交状态后,将状态返回给用户,若提交失败则会解析失败原因,并向用户返回报错信息。

(5)流计算引擎会依据快照时间顺序依次读取并处理数据湖中的每一批次的数据,最后根据用户要求,将结果下发至指定的存储地点。此外,流处理任务会每隔一段时间(用户可配置)记录数据处理的批次,任务重启时可以从记录的批次重新启动,避免重复计算^[3]。

(6)若任务为周期批计算任务,在任务管理组件接收请求后,则会记录任务 SQL、任务周期及任务配置信息。任务管理组件还会记录每一次批计算结束时对应数据的数据湖快照时间节点,在下一个执行周期中,任务管理组件会提交 SQL 执行时间,限制计算数据的范围(上次结束时间的批次及当前时间节点批次之间的所有数据)。

(7)批处理计算引擎接收到任务后,将读取时间范围内的所有数据,完成数据处理后,任务管理组件会记录当前时间至最后处理数据的时间节点。若中途报错,则向用户返回报错信息。

2 分布式实时任务开发架构

在移动互联网时代,随着数据量的激增,传统的单机任务架构已无法满足生产需求。因此,分布式任务架构应运而生。相比单机任务架构,分布式任务架构适用于处理大规模数据和高并发请求,且可以通过水平扩展来提高性能和吞吐量。本文采用分布式架构进行实时任务开发,以应对移动互联网时代的挑战。

2.1 分布式任务开发架构

为实现兼容多集群、多版本 Flink 的流任务管理调度平台,本文采用以下技术方案(见图 2)。

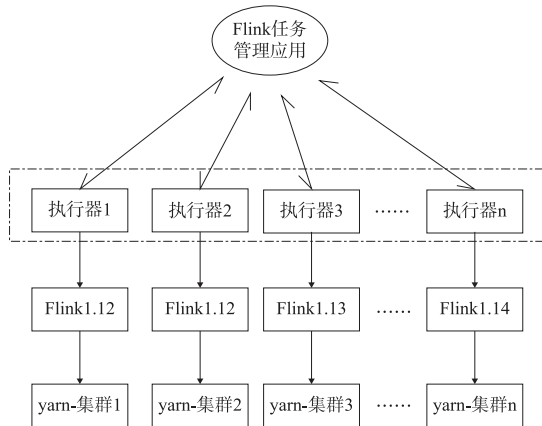


图 2 应用架构图

2.1.1 任务管理模块

任务管理模块是用户创建、管理任务的入口,包括任务配置信息、集群配置信息等,同时也负责任务的分发。在该模块配置任务执行器与集群相关信息,包括任务执行器应用的地址、对应的 YARN 地址以及 Flink 的安装目录。在该模块进行任务配置时,需指定任务使用的任务执行器。

2.1.2 任务执行器模块

任务执行器负责将任务提交到 Flink,并监控任务在 YARN 上的状态。每个 Hadoop 集群只需要部署一个任务执行器。

2.1.3 应用部署

该架构需要在主机上部署一个 Flink 任务管理应用,并在同一主机上部署一个任务执行器应用^[4]。本架构的优势在于可以为每个集群中部署所需的 Flink 版本(允许任意版

本的 Flink),保证了多版本的兼容性。

2.1.4 任务执行

用户提交任务后,任务管理平台会根据任务绑定的执行器信息,构建 HTTP 请求,并将任务分发到相应的执行器节点。执行器接收到请求后,会依次执行以下步骤。

(1)校验请求的合法性,确保请求来源可信。

(2)若请求合法,则根据请求参数查询任务配置。

(3)将任务配置的 Flink SQL 脚本写入文件,并获取文件路径,用于构建执行脚本。

(4)执行器下载任务依赖的自定义 JAR 包到本地指定目录,并记录 JAR 包路径。

(5)根据任务配置信息和 Flink 配置,构建 Flink 任务的执行脚本。

(6)执行器启动任务,将任务提交到 YARN 执行,并获取任务在 YARN 上的 application_id。

(7)执行器将任务的日志信息插入本地日志记录。

这一架构允许在不同的集群上部署不同版本的 Flink,同时通过任务管理平台统一管理任务的创建、配置和分发,使得流任务的管理和调度更加灵活和高效。

2.2 界面化及低代码开发

传统的 FlinkSQL 需要专业的人员来编写 SQL 语句^[5],并配置数据的来源与去处,普通用户难以胜任。FlinkSQL 组件化配置功能则只需要用户知道所需提取的字段,进行简单配置即可,这降低了使用门槛。同时,由于组件实现了标准化,其支持 Kafka, MySQL, hudi 等多种数据源,用户只需简单选择适合的配置选项,降低了编写 SQL 过程中出现错误的可能性,提高了生产效率。FlinkSQL 组件配置需要 4 个要素,即任务配置、源表配置、目标表配置以及源表目标表映射关系。可视化任务配置的一般流程如下。

(1)任务配置:任务的基本配置包括任务名称、自定义函数、第三方 jar 包、任务命令行参数、运行集群地址等。

(2)源端配置:源端配置包含源端类型、源端库、源表名称、字段名称、字段类型、表达式等。

(3)目标端配置:目标表配置包含目标库、目标表名称、字段名称、字段类型、主键标志等。

(4)映射关系:映射关系表示源表字段和目标表字段之间的对应关系。

(5)生成 SQL:生成 FlinkSQL 时,需先检查任务配置中是否有自定义函数,以英文分号结尾拼接;然后根据源表、目标表的配置生成 Create 建表语句;最后根据映射关系生成 insert 语句。

(6)保存 SQL:FlinkSQL 创建完成后,将任务配置、源表配置、目标表配置以及源表目标表映射关系保存到数据库中,供发起 Flink 任务时使用。

按上述向导模式创建任务后,即可随时提交、执行任务。用户在提交 FlinkSQL 任务时,程序会自动查询任务配置表,查询 Flink 执行时用到的命令行参数、checkpoint 快照路径、第三

(下转第 315 页)