(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 117724706 B
(45)授权公告日 2024.05.03

(21)申请号 202410171210.0

(22)申请日 2024.02.06

(65)同一申请的已公布的文献号
申请公布号 CN 117724706 A

(43)申请公布日 2024.03.19

(73)专利权人 湖南盛鼎科技发展有限责任公司
地址 410205 湖南省长沙市高新开发区尖山路18号长沙中电软件园二期A8栋

(72)发明人 王怀采 王先红 李修庆

(74)专利代理机构 长沙致为远航知识产权代理事务所(普通合伙) 43280
专利代理师 罗霞

(51)Int.Cl.
G06F 8/34(2018.01)
G06F 9/48(2006.01)
G06F 9/54(2006.01)

(56)对比文件
CN 113535837 A,2021.10.22
CN 115495221 A,2022.12.20
CN 117149873 A,2023.12.01
WO 2023082681 A1,2023.05.19
CN 106557457 A,2017.04.05
CN 111400352 A,2020.07.10
CN 111597005 A,2020.08.28
CN 113052322 A,2021.06.29
CN 115687468 A,2023.02.03
CN 116796015 A,2023.09.22
EP 3049916 A1,2016.08.03
US 2004078105 A1,2004.04.22
US 2008059563 A1,2008.03.06
WO 2022056735 A1,2022.03.24

审查员 王瑞丽

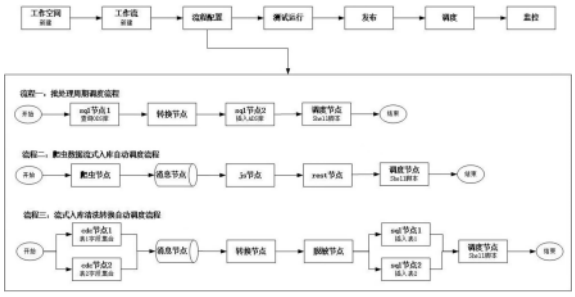
权利要求书3页 说明书17页 附图3页

(54)发明名称

批流一体流程化实时处理异构平台海量数据的方法及系统

(57)摘要

本发明公开一种批流一体流程化实时处理异构平台海量数据的方法及系统,包括如下步骤:进行环境部署,并对基础数据初始化;新建工作空间,并在所述工作空间内新建工作流,流程配置和节点配置,并发布所述工作流至所述作业调度服务模块;所述作业调度服务模块获取要调度的所述工作流的详细信息,并按照所述工作流的消息节点进行作业调度分组,将所述调度作业发布到批流一体计算平台;工作流服务模块提供了可视化的图形组件;依赖于消息服务组件的异步传输机制,实现高效的大数据处理和实时数据处理;实现从数据获取到数据输出的整个过程的全流程化;采用变化数据捕捉方案,实时地实现数据的增量加载,从而实现异构平台下海量数据高效处理。



CN 117724706 B

1.一种批流一体流程化实时处理异构平台海量数据的方法,其特征在于,包括如下步骤:

进行环境部署,包括源数据库、目标数据库、消息中间件模块、工作流服务模块、批流一体计算平台和作业调度服务模块,其中:所述工作流服务模块包括可视化的图形组件,所述消息中间件模块包括用于异步传输的消息服务组件,并对基础数据初始化;

通过所述工作流服务模块新建工作空间,并在所述工作空间内新建工作流,对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块,所述工作流实现从数据获取到数据输出的整个过程的全流程化;

所述作业调度服务模块通过所述工作流的ID获取要调度的所述工作流及所有节点的详细信息;

根据所述工作流的详细信息,判断是否有消息节点;

若是,则按照消息节点进行作业调度分组,所述消息节点前的为第一调度作业,所述第一调度作业通过变化数据捕捉机制,以流的形式实时采集所述源数据库变化的数据,然后发送给所述消息中间件模块形成消息队列,所述消息节点后的为第二调度作业,所述第二调度作业通过从消息队列中获取源数据,根据后续节点对所述源数据处理和入目标数据库;

若否,则整个所述工作流分组为一个所述调度作业;

读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,所述批流一体计算平台执行各个所述调度作业;其中,所述批流一体计算平台包括用于执行工作流中各个阶段的计算引擎。

2.如权利要求1所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,所述工作流包括sql节点、变化数据捕捉节点、消息节点、转换节点、脱敏节点、调度节点、爬虫节点、js节点和rest节点中的一个或者多个;

所述sql节点用于执行标准的sql;

所述变化数据捕捉节点用于利用capturedatachange机制,以流的形式实时采集变化的数据;

所述消息节点的属性包括节点名称、类型、输出格式、数据源、租户、命名空间和主题;

所述转换节点用于支持正则表达式,以及常用的字符串操作函数;

所述脱敏节点用来处理敏感字段,所述敏感字段包括手机号、身份证号、银行卡号、IP地址和姓名;

所述调度节点用于指定流程运行的环境和调度参数;

所述爬虫节点用于定义要爬取的网站url,登录验证信息;

所述js节点用于定义javascript函数;

所述rest节点用于直接调用restfulapi。

3.如权利要求1所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,所述对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块的步骤包括:

在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流;

并发布所述工作流至所述作业调度服务模块。

4.如权利要求3所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

在工作流画布界面,拖拽一个开始节点;

拖拽一个变化数据捕捉节点,第一连接线从所述开始节点指向所述变化数据捕捉节点;

拖拽另一个所述变化数据捕捉节点,第二连接线从开始节点指向另一个所述变化数据捕捉节点;

拖拽一个消息节点,第三连接线从前面两个所述变化数据捕捉节点指向所述消息节点;

拖拽一个转换节点,第四连接线从所述消息节点指向所述转换节点;

拖拽一个脱敏节点,第五连接线从所述转换节点指向所述脱敏节点;

拖拽一个sql节点,第六连接线从所述脱敏节点指向所述sql节点;

拖拽另一个所述sql节点,第七连接线从脱敏节点指向另一个所述sql节点;

拖拽一个调度节点,第八连接线从前两个所述sql节点同时指向所述调度节点;

点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

5.如权利要求3所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

在工作流画布界面,拖拽一个开始节点;

拖拽一个sql节点,第九连接线从所述开始节点指向所述sql节点;

拖拽一个转换节点,第十连接线从所述sql节点指向所述转换节点;

拖拽另一个sql节点,第十一连接线从所述转换节点指向另一个所述sql节点;

拖拽一个调度节点,第十二连接线从另一个所述sql节点指向所述调度节点;

点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

6.如权利要求3所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

在工作流画布界面,拖拽一个开始节点;

拖拽一个爬虫节点,第十三连接线从所述开始节点指向所述爬虫节点;

拖拽一个消息节点,第十四连接线从所述爬虫节点指向所述消息节点;

拖拽一个js节点,第十五连接线从所述消息节点指向所述js节点;

拖拽一个rest节点,第十六连接线从所述js节点指向所述rest节点;

拖拽一个调度节点,第十七连接线从所述rest节点指向所述调度节点;

点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

7.如权利要求1-6中任一所述的批流一体流程化实时处理异构平台海量数据的方法,其特征在于,读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,所述批流一体计算平台执行各个所述调度作业;其中,所述批流一体计算平台包括用于执行工作流中各个阶段的计算引擎的步骤之后,还包括如下步骤:

作业监控步骤,其中:所述作业监控步骤包括查看调度日志、监控各个作业的运行情况和资源耗用情况。

8.一种批流一体流程化实时处理异构平台海量数据的系统,用于实现权利要求1-7中任一项所述的一种批流一体流程化实时处理异构平台海量数据的方法,其特征在于,包括源数据库、消息中间件模块、工作流服务模块、作业调度服务模块、批流一体计算平台和目标数据库;

所述源数据库用于存储各业务系统数据;

所述消息中间件模块用于支持主题的动态创建、消费与生产、异步实现数据流输入与输出;

所述工作流服务模块用于以工作流的方式,图形化配置各计算节点,并将配置好的流程发布到作业调度平台;

所述作业调度服务模块用于批处理和流处理作业调度;

所述批流一体计算平台用于运行在开源调度集群框架Flink或Spark计算平台上的自动化引擎,同时用于批处理和流处理计算;

所述目标数据库用于作为数据仓库或目标业务数据库。

批流一体流程化实时处理异构平台海量数据的方法及系统

技术领域

[0001] 本发明涉及批流一体实时处理异构数据的技术领域,具体涉及一种批流一体流程化实时处理异构平台海量数据的方法及系统。

背景技术

[0002] 在当代信息社会中,数据的处理和分析已经成为了一个重要的议题,大量的数据处理需求推动了数据处理技术不断的发展,出现了很多解决海量数据处理的软件方案。然而,当前的处理方案往往存在着数据处理不能实时、配置复杂、处理过程不规范,不适用于异构平台的问题。因此,如何解决异构平台下海量数据高效处理的问题成为了数据处理技术的新的研究方向。

发明内容

[0003] 本发明的主要目的是提供一种批流一体流程化实时处理异构平台海量数据的方法及系统及设备,旨在解决现有异构平台下海量数据高效处理的问题。

[0004] 为实现上述目的,本发明提出的批流一体流程化实时处理异构平台海量数据的方法,包括如下步骤:

[0005] 进行环境部署,包括源数据库、目标数据库、消息中间件模块、工作流服务模块、批流一体计算平台和作业调度服务模块,其中:所述工作流服务模块包括可视化的图形组件,所述消息中间件模块包括用于异步传输的消息服务组件,并对基础数据初始化;

[0006] 通过所述工作流服务模块新建工作空间,并在所述工作空间内新建工作流,对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块;

[0007] 所述作业调度服务模块获取要调度的所述工作流的详细信息,并按照所述工作流的消息节点进行作业调度分组,其中:每个分组都是一个独立的调度作业;

[0008] 读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,所述批流一体计算平台执行各个所述调度作业;其中,所述批流一体计算平台包括用于执行工作流中各个阶段的计算引擎。

[0009] 优选地,所述作业调度服务模块获取要调度的所述工作流的详细信息,并按照所述工作流的消息节点类型进行作业调度分组的步骤,包括如下步骤:

[0010] 所述作业调度服务模块通过所述工作流的ID获取要调度的所述工作流及所有节点的详细信息;

[0011] 根据所述工作流的详细信息,判断是否有消息节点;

[0012] 若是,则按照消息节点进行作业调度分组,所述消息节点前的为第一调度作业,所述消息节点后的为第二调度作业;

[0013] 若否,则整个所述工作流分组为一个所述调度作业。

[0014] 优选地,所述按照消息节点进行作业调度分组,所述消息节点前的为第一调度作业,所述消息节点后的为第二调度作业的步骤之后包括:

[0015] 读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,

[0016] 所述批流一体计算平台同时执行所述第一调度作业和第二调度作业,其中:所述第一调度作业通过变化数据捕捉机制,以流的形式实时采集变化的数据,然后发送给消息队列;所述第二调度作业通过从消息队列中获取源数据,根据后续节点对所述源数据处理和入库。

[0017] 优选地,所述工作流包括sql节点、变化数据捕捉节点、消息节点、转换节点、脱敏节点、调度节点、爬虫节点、js节点和rest节点中的一个或者多个;

[0018] 所述sql节点用于执行标准的sql;

[0019] 所述变化数据捕捉节点用于利用capturedatachange机制,以流的形式实时采集变化的数据;

[0020] 所述消息节点的属性包括节点名称、类型、输出格式、数据源、租户、命名空间和主题;

[0021] 所述转换节点用于支持正则表达式,以及常用的字符串操作函数;

[0022] 所述脱敏节点用来处理敏感字段,所述敏感字段包括手机号、身份证号、银行卡号、IP地址和姓名;

[0023] 所述调度节点用于指定流程运行的环境和调度参数;

[0024] 所述爬虫节点用于定义要爬取的网站url,登录验证信息;

[0025] 所述js节点用于定义javascript函数;

[0026] 所述rest节点用于直接调用restfulapi。

[0027] 优选地,所述对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块的步骤包括:

[0028] 在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流;

[0029] 并发布所述工作流至所述作业调度服务模块。

[0030] 优选地,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

[0031] 在工作流画布界面,拖拽一个开始节点;

[0032] 拖拽一个变化数据捕捉节点,第一连接线从所述开始节点指向所述变化数据捕捉节点;

[0033] 拖拽另一个所述变化数据捕捉节点,第二连接线从开始节点指向另一个所述变化数据捕捉节点;

[0034] 拖拽一个消息节点,第三连接线从前面两个所述变化数据捕捉节点指向所述消息节点;

[0035] 拖拽一个转换节点,第四连接线从所述消息节点指向所述转换节点;

[0036] 拖拽一个脱敏节点,第五连接线从所述转换节点指向所述脱敏节点;

[0037] 拖拽一个sql节点,第六连接线从所述脱敏节点指向所述sql节点;

[0038] 拖拽另一个所述sql节点,第七连接线从脱敏节点指向另一个所述sql节点;

[0039] 拖拽一个调度节点,第八连接线从前两个所述sql节点同时指向所述调度节点;

[0040] 点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0041] 优选地,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

[0042] 在工作流画布界面,拖拽一个开始节点;

[0043] 拖拽一个sql节点,第九连接线从所述开始节点指向所述sql节点;

[0044] 拖拽一个转换节点,第十连接线从所述sql节点指向所述转换节点;

[0045] 拖拽另一个sql节点,第十一连接线从所述转换节点指向另一个所述sql节点;

[0046] 拖拽一个调度节点,第十二连接线从另一个所述sql节点指向所述调度节点;

[0047] 点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0048] 优选地,所述在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流的步骤,包括如下步骤:

[0049] 在工作流画布界面,拖拽一个开始节点;

[0050] 拖拽一个爬虫节点,第十三连接线从所述开始节点指向所述爬虫节点;

[0051] 拖拽一个消息节点,第十四连接线从所述爬虫节点指向所述消息节点;

[0052] 拖拽一个js节点,第十五连接线从所述消息节点指向所述js节点;

[0053] 拖拽一个rest节点,第十六连接线从所述js节点指向所述rest节点;

[0054] 拖拽一个调度节点,第十七连接线从所述rest节点指向所述调度节点;

[0055] 点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0056] 优选地,读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,所述批流一体计算平台执行各个所述调度作业;其中,所述批流一体计算平台包括用于执行工作流中各个阶段的计算引擎的步骤之后,还包括如下步骤:

[0057] 作业监控步骤,其中:所述作业监控步骤包括查看调度日志、监控各个作业的运行情况和资源耗用情况。

[0058] 一种批流一体流程化实时处理异构平台海量数据的系统,包括源数据库、消息中间件模块、工作流服务模块、作业调度服务模块、批流一体计算平台和目标数据库;

[0059] 所述源数据库用于存储各业务系统数据;

[0060] 所述消息中间件模块用于支持主题的动态创建、消费与生产、异步实现数据流输入与输出;

[0061] 所述工作流服务模块用于以工作流的方式,图形化配置各计算节点,并将配置好的流程发布到作业调度平台;

[0062] 所述作业调度服务模块用于批处理和流处理作业调度;

[0063] 所述批流一体计算平台用于运行在开源调度集群框架Flink或Spark计算平台上的自动化引擎,同时用于批处理和流处理计算;

[0064] 所述目标数据库用于作为数据仓库或目标业务数据库。

[0065] 本发明的技术方案中,

[0066] 1、工作流服务模块提供了可视化的图形组件,通过拖拉拽、点选和设置属性等方式来完成数据处理各过程的配置,开发人员不需要手动编写代码,而是利用图形UI和复用接口来完成功能;

[0067] 2、依赖于消息服务组件的异步传输机制,将批处理和流处理技术进行深度整合,实现高效的大数据处理和实时数据处理;

[0068] 3、数据开发处理的各个过程,包括数据同步、提取、清洗、转换、脱敏、执行、入库等,均定义成可配置的流程节点,各节点之间有序连接构成一个可执行的工作流,实现从数据获取到数据输出的整个过程的全流程化;

[0069] 4、通过作业调度服务模块进行触发或者实时调度,可以采用变化数据捕捉(CDC,Change Data Capture)方案,通过不断监控原始数据系统的更改,提取,转换并将它们分发到目标数据库,可近乎实时地实现数据的增量加载。

[0070] 从而实现异构平台下海量数据高效处理。

附图说明

[0071] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图示出的结构获得其他的附图。

[0072] 图1为本发明批流一体流程化实时处理异构平台海量数据的方法流程示意图。

[0073] 图2为本发明批流一体流程化实时处理异构平台海量数据的系统结构示意图。

[0074] 图3为本发明批流一体流程化实时处理异构平台海量数据的方法的流程结构示意图。

[0075] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0076] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明的一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0077] 另外,在本发明中如涉及“第一”、“第二”等的描述仅用于描述目的,而不能理解为指示或暗示其相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。在本发明的描述中,“多个”的含义是至少两个,例如两个,三个等,除非另有明确具体的限定。

[0078] 在本发明中,除非另有明确的规定和限定,术语“连接”、“固定”等应做广义理解,例如,“固定”可以是固定连接,也可以是可拆卸连接,或成一体;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通或两个元件的相互作用关系,除非另有明确的限定。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本发明中的具体含义。

[0079] 另外,本发明各个实施例之间的技术方案可以相互结合,但是必须是以本领域普通技术人员能够实现为基础,当技术方案的结合出现相互矛盾或无法实现时应当认为这种技术方案的结合不存在,也不在本发明要求的保护范围之内。

[0080] 请参照图1-图3,本发明提出一种批流一体流程化实时处理异构平台海量数据的方法,包括如下步骤:

[0081] S100,进行环境部署,包括源数据库、目标数据库、消息中间件模块、工作流服务模

块、批流一体计算平台和作业调度服务模块,其中:所述工作流服务模块包括可视化的图形组件,所述消息中间件模块包括用于异步传输的消息服务组件,并对基础数据初始化;

[0082] S200,通过所述工作流服务模块新建工作空间,并在所述工作空间内新建工作流,对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块;

[0083] S300,所述作业调度服务模块获取要调度的所述工作流的详细信息,并按照所述工作流的消息节点进行作业调度分组,其中:每个分组都是一个独立的调度作业;

[0084] S400,读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体计算平台,所述批流一体计算平台执行各个所述调度作业;其中,所述批流一体计算平台包括用于执行工作流中各个阶段的计算引擎。

[0085] 本发明的技术方案中:

[0086] 1、工作流服务模块提供了可视化的图形组件,通过拖拉拽、点选和设置属性等方式来完成数据处理各过程的配置,开发人员不需要手动编写代码,而是利用图形UI和复用接口来完成功能;

[0087] 2、依赖于消息服务组件的异步传输机制,将批处理和流处理技术进行深度整合,实现高效的大数据处理和实时数据处理;

[0088] 3、数据开发处理的各个过程,包括数据同步、提取、清洗、转换、脱敏、执行、入库等,均定义成可配置的流程节点,各节点之间有序连接构成一个可执行的工作流,实现从数据获取到数据输出的整个过程的全流程化;

[0089] 4、通过作业调度服务模块进行触发或者实时调度,可以采用变化数据捕捉(CDC,Change Data Capture)方案,通过不断监控原始数据系统的更改,提取,转换并将它们分发到目标数据库,可近乎实时地实现数据的增量加载。

[0090] 从而实现异构平台下海量数据高效处理。

[0091] 具体的,系统各方参与主体,通过以下六部分组成:工作流服务,作业调度服务,批流一体计算平台,源数据库,目标数据库,消息中间件。

[0092] 其中工作流服务提供了可视化的界面配置,支持新建工作空间、新建工作流、以拖拉拽的图形化方式组装流程节点、测试运行、发布工作流等;作业调度服务提供批流一体化处理作业调度,对应于上图中的“发布”环节。

[0093] 批流一体计算平台是一套运行在Flink或Spark大数据集群上的计算引擎,用于执行工作流中的各个节点,可以同时处理实时和离线数据,通过对数据处理的优化和动态资源分配等方式,充分利用集群资源,提供灵活、实时、高效的服务。

[0094] 在本发明的另一实施方式中,所述S300的步骤,包括如下步骤:

[0095] S310,所述作业调度服务模块通过所述工作流的ID获取要调度的所述工作流及所有节点的详细信息;

[0096] S320,根据所述工作流的详细信息,判断是否有消息节点;

[0097] 若是,则执行S330,按照消息节点进行作业调度分组,所述消息节点前的为第一调度作业,所述消息节点后的为第二调度作业;

[0098] 若否,则执行S340,则整个所述工作流分组为一个所述调度作业。

[0099] 在本发明的又一实施方式中,所述S330的步骤之后,包括:

[0100] S331,读取所述工作流的调度节点的配置信息,将所述调度作业发布到批流一体

计算平台，

[0101] S332,所述批流一体计算平台同时执行所述第一调度作业和第二调度作业,其中:所述第一调度作业通过变化数据捕捉机制,以流的形式实时采集变化的数据,然后发送给消息队列;所述第二调度作业通过从消息队列中获取源数据,根据后续节点对所述源数据处理和入库。

[0102] 在本发明的又一实施方式中，

[0103] S33110,流程配置:在工作流服务的可视化界面中,拖拽各处理节点,配置好流程三,即“流式入库清洗转换自动调度流程”。

[0104] S33111,工作流发布:点击“发布”按钮,将工作流发布到作业调度服务。

[0105] S33113,作业调度服务:由工作流的id获取流程信息。按消息节点进行作业调度分组,每个分组都是一个独立的作业;如果没有消息节点,则整个流程就是一个调度作业;读取调度节点的配置信息,将作业调度发布到批流一体计算平台。

[0106] S33114,批流一体计算平台:同时执行上面的二个调度作业。其中消息节点前的调度作业,通过变化数据捕捉机制,以流的形式实时采集变化的数据,然后发送给消息队列。

[0107] S33115,消息节点后的调度作业,从消息队列中获取源数据;通过转换节点执行转换函数,再根据脱敏节点的配置对数据进行脱敏,然后设置sql语句中对应的值,处理动态参数,最后执行sql,完成该条记录的入库。

[0108] S33116,作业监控:监控各个作业的运行情况。

[0109] 在本发明的又一实施方式中,所述工作流包括sql节点、变化数据捕捉节点、消息节点、转换节点、脱敏节点、调度节点、爬虫节点、js节点和rest节点中的一个或者多个;

[0110] 所述sql节点用于执行标准的sql;

[0111] 所述变化数据捕捉节点用于利用capturedatachange机制,以流的形式实时采集变化的数据;

[0112] 所述消息节点的属性包括节点名称、类型、输出格式、数据源、租户、命名空间和主题;

[0113] 所述转换节点用于支持正则表达式,以及常用的字符串操作函数;

[0114] 所述脱敏节点用来处理敏感字段,所述敏感字段包括手机号、身份证号、银行卡号、IP地址和姓名;

[0115] 所述调度节点用于指定流程运行的环境和调度参数;

[0116] 所述爬虫节点用于定义要爬取的网站url,登录验证信息;

[0117] 所述js节点用于定义javascript函数;

[0118] 所述rest节点用于直接调用restfulapi。

[0119] 具体的,其中sql节点可执行标准的sql,其节点属性主要有节点名称、类型、数据源、内容等字段。数据源可以是各类关系型、非关系型数据库,包括url、用户名、密码等。内容可以定义标准的sql语句,同时也包括上一节点输出的sql语句;支持全参数化,即所有查询sql的结果集字段均可作为下一节点的同名参数;支持内置参数如当前登录人、当前时间、当前登录人机构等等;一个节点支持多条sql;sql节点输出格式可以为sql/json/xml/csv等,json格式如:{t_sale:{delete:{id:100,name:'zhangsan',age:23}}}

[0120] 变化数据捕捉节点是利用capture data change机制,以流的形式实时采集变化

的数据。其节点属性主要有节点名称、类型、数据源、内容等字段。数据源同sql节点;内容属性定义要同步的表,以及表中的字段。变化数据捕捉节点获取数据后可以sql/json/csv/xml等方式输出数据,json格式如:{t_sale名:{insert:{id:100,name:'zhangsan'},update{},delete:{}}}

[0121] 消息节点的属性有节点名称、类型、输出格式、数据源、租户、命名空间、主题等。数据源可以选择Kafka/RabbitMQ/pulsar/JRocket等各类消息中间件;输出格式支持json/xml/txt/csv/sql等。

[0122] 转换节点支持正则表达式,以及常用的字符串操作函数,如替换(replace)、截取(substr)、拼接(concat)、大写(toUpper)、小写(tolower)等等。如:replace("(t_+)","ods_$1")。

[0123] 脱敏节点用来处理手机号、身份证号、银行卡号、IP地址、姓名等敏感字段;支持的脱敏算法包括假名、HASH、掩盖、字符替换、区间变换、取整、置空等几种方式。节点属性"内容"指定了一个列表,包括要脱敏的字段及对应的脱敏算法。

[0124] 调度节点用来指定流程运行的环境和调度参数,其属性"内容"字段支持写shell脚本,"调度参数"字段支持cron表达式,由"分 时 日 月 周 年"6位字符和4个通配符", - * /"构成。

[0125] 爬虫节点可以定义要爬取的网站url,登录验证信息,可以是json/xml/csv等。

[0126] js节点可以定义javascript函数,对上一节点输出的数据进行更加复杂的处理。

[0127] rest节点可以直接调用restful api,支持get/put/post/delete等操作,如"PUT /db_es1/_doc { ? }",其中"?"代表前一节点输出的json串。

[0128] 在本发明的又一实施方式中,所述S200步骤中对所述工作流进行流程配置和节点配置,并发布所述工作流至所述作业调度服务模块的步骤包括:

[0129] S210,在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成工作流;

[0130] S220,并发布所述工作流至所述作业调度服务模块。

[0131] 在本发明的又一实施方式中,所述S210的步骤,包括如下步骤:

[0132] S21010,在工作流画布界面,拖拽一个开始节点;

[0133] S21011,拖拽一个变化数据捕捉节点,第一连接线从所述开始节点指向所述变化数据捕捉节点;

[0134] S21012,拖拽另一个所述变化数据捕捉节点,第二连接线从开始节点指向另一个所述变化数据捕捉节点;

[0135] S21013,拖拽一个消息节点,第三连接线从前面两个所述变化数据捕捉节点指向所述消息节点;

[0136] S21014,拖拽一个转换节点,第四连接线从所述消息节点指向所述转换节点;

[0137] S21015,拖拽一个脱敏节点,第五连接线从所述转换节点指向所述脱敏节点;

[0138] S21016,拖拽一个sql节点,第六连接线从所述脱敏节点指向所述sql节点;

[0139] S21017,拖拽另一个所述sql节点,第七连接线从脱敏节点指向另一个所述sql节点;

[0140] S21018,拖拽一个调度节点,第八连接线从前两个所述sql节点同时指向所述调度

节点;

[0141] S21019,点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0142] 在本发明的又一实施方式中,所述S210的步骤,包括如下步骤:

[0143] S21020,在工作流画布界面,拖拽一个开始节点;

[0144] S21021,拖拽一个sql节点,第九连接线从所述开始节点指向所述sql节点;

[0145] S21022,拖拽一个转换节点,第十连接线从所述sql节点指向所述转换节点;

[0146] S21023,拖拽另一个sql节点,第十一连接线从所述转换节点指向另一个所述sql节点;

[0147] S21024,拖拽一个调度节点,第十二连接线从另一个所述sql节点指向所述调度节点;

[0148] S21025,点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0149] 具体的,批处理周期调度流程。

[0150] 在本发明的又一实施方式中,所述S210的步骤,包括如下步骤:

[0151] S21030,在工作流画布界面,拖拽一个开始节点;

[0152] S21031,拖拽一个爬虫节点,第十三连接线从所述开始节点指向所述爬虫节点;

[0153] S21032,拖拽一个消息节点,第十四连接线从所述爬虫节点指向所述消息节点;

[0154] S21033,拖拽一个js节点,第十五连接线从所述消息节点指向所述js节点;

[0155] S21034,拖拽一个rest节点,第十六连接线从所述js节点指向所述rest节点;

[0156] S21035,拖拽一个调度节点,第十七连接线从所述rest节点指向所述调度节点;

[0157] S21036,点击画布中的保存按钮,保存整个工作流的配置,形成工作流。

[0158] 具体的,为爬虫数据流式入库自动调度流程。

[0159] 在本发明的又一实施方式中,S400的步骤之后,还包括如下步骤:

[0160] S500,作业监控步骤,其中:所述作业监控步骤包括查看调度日志、监控各个作业的运行情况和资源耗用情况。

[0161] 具体的,部署运行自动化可调度可监控:借助优秀的任务调度框架,轻松实现部署运行的自动化和运行全过程监控,并方便的扩展到大型集群,支持高度并行的数据处理任务。

[0162] 在本发明的又一实施方式中,所述S100的步骤,包括:

[0163] S110,进行环境部署,基于Flink大数据计算平台,搭建集群环境,一个节点作为JobMaster,2个节点作为WorkerMaster。下载并部署flink变化数据捕捉驱动包,以及mysql和sqlserver驱动,将批流一体计算引擎部署到目录形成批流一体计算平台;

[0164] S120,选用pulsar服务器,搭建ZooKeeper集群、Bookkeeper集群和Broker集群,并配置好了相应的命名空间和主题,形成消息中间件模块;

[0165] S130,建立源数据库、目标数据库、工作流服务模块和作业调度服务模块;

[0166] S140,对基础数据初始化,包括数据源管理、业务主题、数仓分层、用户模块、机构模块、角色模块和权限模块的设置。

[0167] 请参照图2,本发明还包括一种批流一体流程化实时处理异构平台海量数据的系统,包括源数据库、消息中间件模块、工作流服务模块、作业调度服务模块、批流一体计算平台和目标数据库;

[0168] 所述源数据库用于存储各业务系统数据;

[0169] 所述消息中间件模块用于支持主题的动态创建、消费与生产、异步实现数据流输入与输出;

[0170] 所述工作流服务模块用于以工作流的方式,图形化配置各计算节点,并将配置好的流程发布到作业调度平台;

[0171] 所述作业调度服务模块用于批处理和流处理作业调度;

[0172] 所述批流一体计算平台用于运行在开源调度集群框架Flink或Spark计算平台上的自动化引擎,同时用于批处理和流处理计算;

[0173] 所述目标数据库用于作为数据仓库或目标业务数据库。

[0174] 具体的,1、源数据库:存储各业务系统数据,可以是各种关系型、文档型数据库,如Oracle、DB2、Mysql、Sqlserver、MongonDB等等;

[0175] 2、消息中间件:支持主题的动态创建,消费与生产,异步实现数据流输入与输出,如Kafka/RabbitMQ/pulsar/JRocket等;

[0176] 3、工作流服务:以工作流的方式,图形化配置各计算节点,并将配置好的流程发布到作业调度平台。

[0177] 4、作业调度服务:用于批处理和流处理作业调度,支持集群调度,多个实例可以协同工作,以实现高可用、可伸缩的作业调度解决方案。推荐选用优秀的开源调度框架如xxl-job,quartz等。

[0178] 5、批流一体计算平台:运行在开源调度集群框架Flink或Spark计算平台上的自动化引擎,同时用于批处理和流处理计算,支持可视化监控,支持三种部署方式。

[0179] a)独立发送端:部署在源数据库网络侧,同步源数据到消息服务器,适用于源数据库不暴漏ip地址和端口的情况,增加数据安全性。

[0180] b)独立接收端:部署在目标数据库网络侧,从消息服务器接收数据并处理入库。

[0181] c)二端合一:部署在开源调度集群框架Flink或Spark计算平台上,同时作为发送端和接收端,适用于源数据库和目标数据库网络连通的场景。

[0182] 6、目标数据库,可以是数据仓库,也可以是目标业务数据库。对于关系型数据,可以使用Oracle、DB2、Mysql、Sqlserver等;对于海量数据,可以使用ElasticSearch、Hadoop生态、MongoDB等等。

[0183] 在本发明的又一实施方式中,以实时同步sqlserver(一种关系型数据库系统)的经营主体数据,经转换处理后到mysql(关系型数据库管理系统)业务场景为例,描述一下具体实施过程。

[0184] 第一步:环境部署。包括工作流服务,作业调度服务,批流一体计算平台,源数据库,目标数据库,消息中间件。

[0185] 1.1、工作流服务。前后端分离,部署好工作流的前端配置系统和后端微服务。

[0186] 1.2、作业调度服务。基于quartz集群调度框架进行二次开发,部署好前端作业配置界面,后端微服务。

[0187] 1.3、批流一体计算平台。基于Flink大数据计算平台,搭建集群环境,一个节点作为JobMaster,2个节点作为WorkerMaster。下载并部署flink变化数据捕捉驱动包,以及mysql和sqlserver驱动。将批流一体计算引擎部署到目录“/home/program/engin/

BatchFlowIntegrateEngine.jar”。

[0188] 1.4、源数据库。提供经营主体的业务数据库,包括二张表基本信息t_basic和销售情况t_sale,使用sqlServer数据库db_subject。

[0189] 1.5、目标数据库。存储清选转换后的目标数据,使用mysql数据仓库,贴源层db_subject。

[0190] 1.6、消息中间件。选用pulsar服务器,搭建ZooKeeper集群、Bookkeeper集群和Broker集群,并配置好了相应的命名空间,主题topic_subject。

[0191] 第二步:基础数据初始化。包括数据源管理、业务主题、数仓分层、用户、机构、角色、权限等模块设置。其中数据源支持DM(达梦)、Mysql、PostgreSQL、SqlServer、Oracle、Db2、Redis、OSS、MongoDB、Elasticsearch、Pulsar、Kafka、Hbase、HDFS等数据源,支持连接模式包括参数模式和连接串模式(jdbcURL)。

[0192] 第三步:工作流配置与发布。全流程化配置经营主体数据的开发需求,并发布结果。

[0193] 3.1:新建工作空间,按业务维度创建一个工作空间,指定项目名称、描述、业务类型。

[0194] 3.2:新建工作流。在指定的工作空间下创建流程,属性字段有流程名称、描述、工作流级别、调度类型(流处理,批处理,周期调度)。

[0195] 3.3:流程配置。在工作流画布界面,拖拽各节点,组成有序连接的有向无环图,最终形成如下工作流图:

[0196] 3.31:拖拽一个开始节点;

[0197] 3.32:拖拽一个变化数据捕捉节点,连接线从开始节点指向自己。节点属性配置中,数据源选择预先配置好的“sqlServer数据库db_subject”,

[0198] 内容为该数据库下的基本信息表和要同步的字段:

[0199] “table:t_basic,fields:[name,address,tel,email,capital,lar,uscc]”。

[0200] 3.33:拖拽一个变化数据捕捉节点,连接线从开始节点指向自己。节点属性配置中,数据源选择同上一步,内容为销售情况表和字段:

[0201] “t_sale:[product,amount,date,price,buyer,desc]”。

[0202] 3.34:拖拽一个消息节点,连接线从前面2个变化数据捕捉节点指向自己。节点属性配置中,数据源选择预先配置好的“pulsar消息中间件”,命名空间为default,主题为topic_subject,输出格式选择json。

[0203] 3.35:拖拽一个转换节点,连接线从消息节点指向自己。节点属性配置中,内容为“replace(“(t_+)”,“ods_$1”)”。

[0204] 3.36:拖拽一个脱敏节点,连接线从转换节点指向自己。节点属性配置中,内容为“[tel:rep,email:cov,lar:pseu,uscc:rep,buyer:pseu]”。

[0205] 3.37:拖拽一个sql节点,连接线从脱敏节点指向自己。节点属性配置中,数据源选择预先配置好的“mysql数据库db_subject”,内容为“insertintoods_t_basic(name,address,tel,email,capital,lar,uscc)values(?,?,?,?,?,?,?)”。

[0206] 3.38:拖拽一个sql节点,连接线从脱敏节点指向自己。节点属性配置中,数据源选择预先配置好的“mysql数据库db_subject”,内容为:

[0207] “insertintoods_t_sale(product,amount,date,price,buyer,desc)values(?,?,?,?,?,?)”。

[0208] 3.39:拖拽一个调度节点,连接线从二个sql节点同时指向自己。

[0209] 内容属性为:

[0210] “$FLINK_HOME/bin/flinkrun-ccom.xxx.Main/home/program/engin/BatchFlowIntegrateEngine.jar”,调度参数为“00256?2023”,即当天2023年6月25日凌晨0点0分开始执行,因为是流处理,所以只执行一次。

[0211] 3.310:拖拽一个结束节点,连接线从调度节点指向自己。

[0212] 3.311:点击画布中的保存按钮,保存整个工作流的配置。

[0213] 3.4:发布工作流。在工作流画布界面,点击“发布”按钮,将工作流发布到作业调度服务。

[0214] 3.41:根据当前工作流id,查询工作流及所有节点的信息,组装成json串。

[0215] {

[0216] id:1,

[0217] links:[10,[11,12],13,14,15,[16,17],18,20],

[0218] nodes:[

[0219] {

[0220] id:10,

[0221] type:'begin'

[0222] },

[0223] {

[0224] id:11,

[0225] type:'变化数据捕捉',

[0226] name:'变化数据捕捉-basicnode',

[0227] datasource:'sqlserver-db_subject',

[0228] content:'table:t_basic,fields:[name,address,tel,email,capital,lar,uscc]'

[0229] },

[0230] {

[0231] id:12,

[0232] type:'变化数据捕捉',

[0233] name:'变化数据捕捉-salenode',

[0234] datasource:'sqlserver-db_subject',

[0235] content:'table:t_sale,fields:[product,amount,date,price,buyer,desc]'

[0236] },

[0237] {

[0238] id:13,

[0239] type:'mq',

[0240] name:'messagequeuenode',
[0241] datasource:'pulsar',
[0242] namespace:'default',
[0243] topic:'topic_subject',
[0244] output:'json'
[0245] },
[0246] {
[0247] id:14,
[0248] type:'translate',
[0249] name:'translatenode',
[0250] content:'replaceAll("(t_+)","ods_$1")'
[0251] },
[0252] {
[0253] id:15,
[0254] type:'desensitize',
[0255] name:'desensitizenode',
[0256] content:'[tel:rep,email:cov,lar:pseu,uscc:rep,buyer:pseu]'
[0257] },
[0258] {
[0259] id:16,
[0260] type:'sql',
[0261] name:'sqlbasicnode',
[0262] datasource:'mysql-db_subject',
[0263] content:'insertintoods_t_basic(name,address,tel,email,capital,lar,uscc)values(?,?,?,?,?,?,?)'
[0264] },
[0265] {
[0266] id:17,
[0267] type:'sql',
[0268] name:'sqlsalenode',
[0269] datasource:'mysql-db_subject',
[0270] content:'insertintoods_t_sale(product,amount,date,price,buyer,desc)values(?,?,?,?,?,?)'
[0271] },
[0272] {
[0273] id:18,
[0274] type:'schedule',
[0275] name:'schedulnode',
[0276] cron:'00256?2023',

[0277] content:'$FLINK_HOME/bin/flinkrun-ccom.xxx.MainBatchFlowIntegrateEngine.jar'

[0278] },

[0279] {

[0280] id:20,

[0281] type:'end'

[0282] }

[0283]]

[0284] }

[0285] 3.42:将json串作为输入参数,传给“作业调度服务”的后台api,进行作业调度。

[0286] 第四步:作业调度。

[0287] 4.1:接收参数:获取要调度的工作流的详细信息。

[0288] 4.2:作业调度分组:按消息节点类型”type==mq”分组,将调度任务分为二个组。消息节点前的一个为作业一,命名为“schedul_变化数据捕捉”;消息节点后的一个为作业二,命名为“schedul_sql”。

[0289] 4.3:读取作业调度配置:读取调度节点的配置信息,得到cron表达式所配置的执行周期,以及内容属性中配置的调度命令。

[0290] 4.4:将作业调度发布到批流一体计算平台:执行调度命令,在批流一体计算平台同时启动调度作业“schedul_变化数据捕捉”和“schedul_sql”。部分业务算法如下:

[0291] publicclassScheduleJobimplementsSerializable{

[0292] privateLongid;

[0293] //任务名称

[0294] privateStringjobName;

[0295] //调度任务类名

[0296] privateStringclassName;

[0297] //bean名称

[0298] privateStringbeanName;

[0299] //方法名称

[0300] privateStringmethodName;

[0301] //执行参数

[0302] privateStringparams;

[0303] //cron表达式

[0304] privateStringcronExpression;

[0305] //任务状态0:正常1:暂停

[0306] privateIntegerstatus;

[0307] ...

[0308] }

[0309] publicclassScheduleServiceImplimplementsScheduleService{

[0310] @Override

```
[0311] publicvoidcreateJob(ScheduleJobscheduleJob){
[0312]          //构建job
[0313]          JobDetailjobDetail=JobBuilder.newJob(
[0314] ScheduleJobBean.class).withIdentity(getJobKey(scheduleJob.getId()))
.build();          //构建cron
[0315]      CronScheduleBuilderscheduleBuilder=CronScheduleBuilder.cronSched
ule(
[0316] scheduleJob.getCronExpression()).withMisfireHandlingInstructionDoNot
hing();
[0317]          //根据cron,构建一个CronTrigger
[0318]  CronTriggertrigger=TriggerBuilder.newTrigger().withIdentity
(getTriggerKey(scheduleJob.getId())).
[0319]                          withSchedule(scheduleBuilder).build();
[0320]          //放入参数,运行时的方法可以获取
[0321]   jobDetail.getJobDataMap().put(JOB_PARAM_KEY,JSON.toJSONString
(scheduleJob));
[0322]          scheduler.scheduleJob(jobDetail,trigger);
[0323]          //暂停任务(定时任务状态0:正常1:暂停)
[0324]  if(scheduleJob.getStatus()!=null&&scheduleJob.getStatus()==
ScheduleStatus.PAUSE.getValue()){
[0325]                  pauseJob(scheduleJob.getId());
[0326]          }
[0327] }
[0328]      publicvoidrunJob(ScheduleJobscheduleJob){
[0329]          JobDataMapdataMap=newJobDataMap();
[0330]          dataMap.put(JOB_PARAM_KEY,JSON.toJSONString(scheduleJob));
[0331]           scheduler.triggerJob(getJobKey(scheduleJob.getId()),
dataMap);
[0332]          scheduler.start();
[0333] }
[0334] }
[0335] a.schedul_变化数据捕捉作业传递参数为要执行的完整链路及节点信息:
[0336] {
[0337] id:1,
[0338] links:[10,[11,12],13,20],
[0339] nodes:[{id:10,...},{id:11,...},...]
[0340] }
[0341] b.schedul_sql作业传递参数为要执行的完整链路及节点信息:
[0342] {
```

[0343] id:1,

[0344] links:[10,13,14,15,[16,17],18,20],

[0345] nodes:[{id:10,...},{id:13,...},...]

[0346] }

[0347] 第五步:批流一体计算平台。在Flink集群环境,通过批流一体计算引擎,同时执行“schedul_变化数据捕捉”和“schedul_sql”。

[0348] 5.1、执行schedul_变化数据捕捉:

[0349] 5.1.1、计算引擎调用工作流类的执行方法,输入参数为要执行的完整链路及节点信息;将节点执行路径解析成节点链表,每个节点用next()方法获取所有下一节点;按节点链路表依次执行每个节点。

[0350] 5.1.2、执行第一个“变化数据捕捉节点”:执行该链路中的节点方法,即变化数据捕捉子类的exec()方法,参数为当前节点的完整信息,可以从工作流类的参数中解析获得。执行代码逻辑简要描述如下:

```
[0351] publicvoidexecute(Nodenode,Objectparam){
[0352]     ...
[0353]     //如果节点的执行次数超过规定值,直接跳过,以防多链路节点重复执行;
[0354]     if(TimesMap.get(node)>=node.getMaxTime())continue;
[0355]     //获取节点的执行器;
[0356]     NodeExecutorne=executorMap.get(node.getType());
[0357]      //执行节点的引擎方法,获取输出参数,作为下一节点的输入参数;
Objectresult=ne.exec(param);
[0358]           List<Node>nodes=node.next();
[0359]     for(Nodend:nodes){
[0360]           execute(nd,result);
[0361]     }
[0362] };
```

[0363] 变化数据捕捉子类的exec()方法的简要逻辑如下:

```
[0364] publicObjectexecute(Objectparam){
[0365] //获取当前节点
[0366]     Nodenode=getNode(param);
[0367] //连接源数据库,利用自定义的解析类,流式同步获取t_basic表的变化的数据为
json格式;
[0368] Objectobj=toParam(SqlServerSource.<String>builder()
[0369] .hostname(node.getDataSource("hostname"))
[0370] .port(Integer.parseInt(node.getDataSource("port")))
[0371] .database(node.getDataSource("database"))
[0372] .tableList(node.getContent("table"))
[0373] .username(node.getDataSource("username"))
[0374] .password(node.getDataSource("password"))
```

[0375] .deserializer(newCustomerDeserialization())

[0376] .build());

[0377] returnobj;

[0378] }

[0379] 5.1.3、执行第二个“变化数据捕捉节点”:逻辑同上,按节点链路表依次执行每个节点。

[0380] 5.1.4、执行消息节点:第一个变化数据捕捉节点执行完后,执行其下一节点时,将输出参数作为消息节点的sink()方法的入参,将采集的数据发送到消息队列指定的主题“topic_subject”。同理,第二变化数据捕捉节点执行完后,也会执行消息节点sink()方法。

[0381] 5.1.5、因为是流处理,所以只要表t_basic或t_sale有数据变化,对应的变化数据捕捉节点就会自动触发下一节点执行,把变化的数据发送到消息节点。

[0382] 5.2、执行schedul_sql:

[0383] 5.2.1、计算引擎调用工作流类的执行方法,逻辑同上,。

[0384] 5.2.2、执行消息节点:执行消息节点的source()方法,由消息中间件的url,命名空间和主题信息,向主题订阅消息,获取到数据json串。

[0385] 5.2.3、执行转换节点:执行转换函数,如将json串中“t_”开头的表名加上前缀“ods_”。

[0386] 5.2.4、执行脱敏节点:即将tel(电话号码)、uscc(统一社会信用代码)使用替换算法替换后四个字符为“****”;lar(法定代表人)、buyer(购买方)用假名算法;email使用掩盖算法只展示后6位。

[0387] 5.2.5、执行第一个“sql节点”:获取上一节点输出的数据,通过表名快速匹配到sql节点ods_t_basic;为sql中的字段设置值,处理动态参数;再利用sql节点配置的数据源信息建立数据库连接,执行insertsql语句,将当前记录写入mysql数据库db_subject。

[0388] 5.2.6、执行第二个“sql节点”:通过表名快速匹配到sql节点ods_t_sale,其它逻辑同上。

[0389] 因为是流处理,消息节点不断获取流数据,自动触发下一节点执行。

[0390] 第六步:作业监控。

[0391] 在控制台查看调度日志;

[0392] 监控各个作业的运行情况;

[0393] 资源耗用情况等。

[0394] 一种批流一体流程化实时处理异构平台海量数据的系统,包括源数据库、消息中间件模块、工作流服务模块、作业调度服务模块、批流一体计算平台和目标数据库;

[0395] 所述源数据库用于存储各业务系统数据;

[0396] 所述消息中间件模块用于支持主题的动态创建、消费与生产、异步实现数据流输入与输出;

[0397] 所述工作流服务模块用于以工作流的方式,图形化配置各计算节点,并将配置好的流程发布到作业调度平台;

[0398] 所述作业调度服务模块用于批处理和流处理作业调度;

[0399] 所述批流一体计算平台用于运行在开源调度集群框架Flink或Spark计算平台上的自动化引擎,同时用于批处理和流处理计算;

[0400] 所述目标数据库用于作为数据仓库或目标业务数据库。

[0401] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是在本发明的构思下,利用本发明说明书及附图内容所作的等效结构变换,或直接/间接运用在其他相关的技术领域均包括在本发明的专利保护范围内。

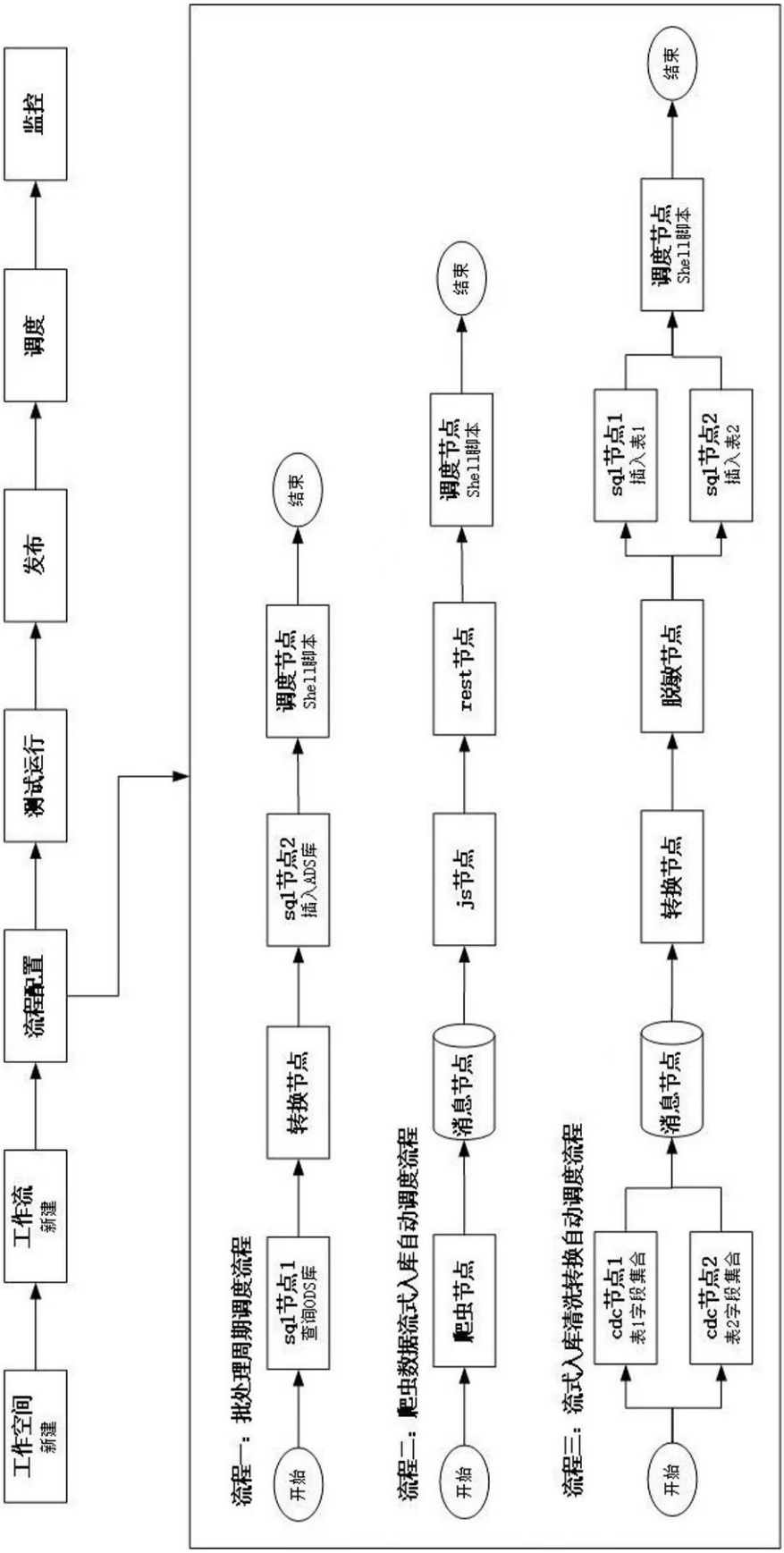

工作空间
新建
工作流
新建
流程配置
测试运行
发布
调度
监控
流程一：批处理周期调度流程
开始
sql节点1
查询ODS库
转换节点
sql节点2
插入ADS库
调度节点
Shell脚本
结束
流程二：爬虫数据流式入库自动调度流程
开始
爬虫节点
消息节点
js节点
rest节点
调度节点
Shell脚本
结束
流程三：流式入库清洗转换自动调度流程
开始
cdc节点1
表1字段集合
cdc节点2
表2字段集合
消息节点
转换节点
脱敏节点
sql节点1
插入表1
sql节点2
插入表2
调度节点
Shell脚本
结束

图 1

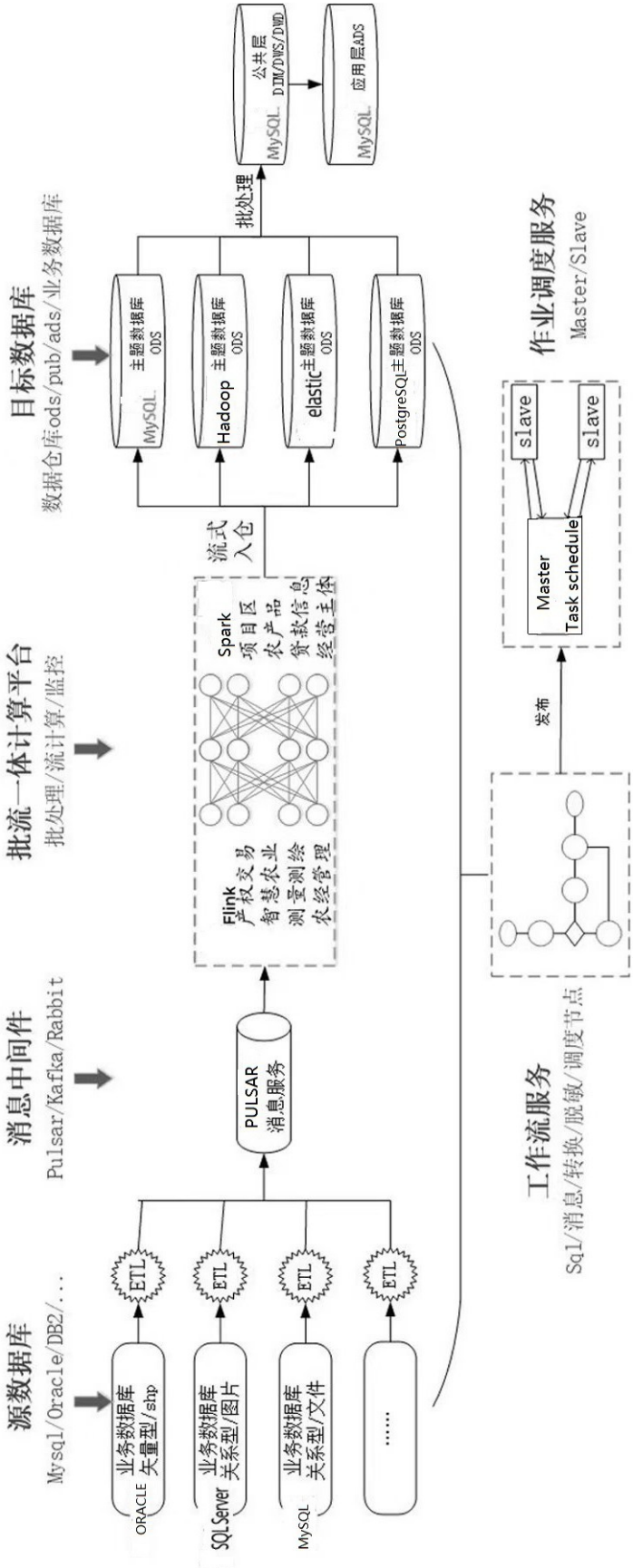

源数据库
Mysql/Oracle/DB2/...
消息中间件
Pulsar/Kafka/Rabbit
批流一体计算平台
批处理/流计算/监控
目标数据库
数据仓库ods/pub/ads/业务数据库
工作流服务
Sql/消息/转换/脱敏/调度节点
作业调度服务
Master/Slave
ORACLE
业务数据库
矢量型/shp
SQLServer
业务数据库
关系型/图片
MySQL
业务数据库
关系型/文件
.....
ETL
PULSAR
消息服务
Flink
产权交易
智慧农业
测量测绘
农经管理
Spark
项目区
农产品
贷款信息
经营主体
流式
入仓
批处理
MySQL
主题数据库
ODS
Hadoop
主题数据库
ODS
elastic
主题数据库
ODS
PostgreSQL
主题数据库
ODS
MySQL
公共层
DIM/DWS/DWD
MySQL
应用层ADS
发布
Master
Task schedule
slave
slave

图 2

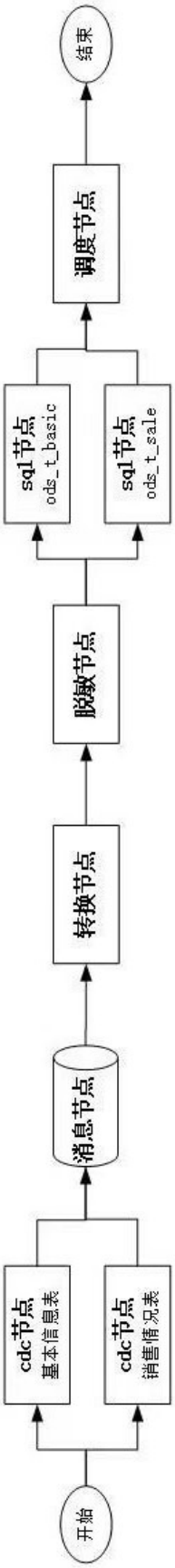

开始
cdc节点
基本信息表
cdc节点
销售情况表
消息节点
转换节点
脱敏节点
sql节点
ods_t_basic
sql节点
ods_t_sale
调度节点
结束

图 3